

## 階層的な分割及び適応的な階層選択による 弾性体の変形シミュレーション

今井 星<sup>†1</sup> Paulo Silva<sup>†1</sup> Yonghao Yue<sup>†1</sup>  
陳 炳 宇<sup>†2</sup> 西田 友 是<sup>†1</sup>

物体の変形シミュレーションに関してはこれまでCG分野で多くの研究がなされているが、そのほとんどが等方的で、均質な弾性体の変形のみを取り扱っている。

しかし現実には、等方的で均質な弾性体だけではなく、非均質な弾性体も多く存在する。そこで本研究では非均質な弾性体、等方的で均質な弾性体の双方を扱う。また本手法は非等方的な弾性体にも拡張できる。さらに提案法ではこうした弾性体の変形を効率良く扱うための方法を提案する。

提案法では、あらかじめ弾性体を四面体分割し、各四面体を再帰的に分割して階層構造を構築する。変形する際に部分ごとに分解能を適応的な選択をし、変化の激しい部分のみ細かい分解能で変形することで、従来法にくらべ計算量を削減することを試みる。

### Simulation of Elastic Deformation with Recursive Subdivision and Adaptive Level Selection

SEI IMAI,<sup>†1</sup> PAULO SILVA,<sup>†1</sup> YONGHAO YUE,<sup>†1</sup>  
BING-YU CHEN<sup>†2</sup> and TOMOYUKI NISHITA<sup>†1</sup>

In the Computer Graphics field, there's a substantial amount of work on simulation of elastic deformation. Most of this work targets homogeneous and isotropic objects (an object is homogeneous when its material property is independent of the location of the object, and is isotropic when its material property is independent of the orientation). However, real-world objects are not just homogeneous and isotropic. There are many objects which have an inhomogeneous and anisotropic composition. In this paper we focus on the inhomogeneous object types, and we propose an efficient simulation method for deformation.

In our method, first we tetrahedronize the target object, recursively subdivide each tetrahedron a number of times, and construct the hierarchy. In the deformation, we adaptively select a level of resolution of the simulation for each

of the original tetrahedra as a function the deformation it undergoes.

According to our results our method can help reduce the computation in the simulation of deformation.

#### 1. はじめに

物体の変形シミュレーションはコンピューターグラフィックスにおいて、例えばゴムでできたボールが壁に衝突してしゃげるといったような、弾性体同士の衝突や接触するシーンをよりリアルに表示する技術として重要である。変形に関する物理則は昔から研究されており連続体力学と呼ばれる分野でそのモデルが示されている。

CG分野ではO'Brienらにより有限要素法を用いる方法が提案された<sup>10)</sup>。O'Brienらの手法は、四面体メッシュにより物体を表現し、三次元のフックの法則を用いて、各四面体の歪みからその四面体の応力を計算する。この手法では基本的に等方的で、均質な弾性体の変形のみを取り扱っている。

しかし現実には、等方的で均質な弾性体だけではなく、植物繊維や筋繊維など非等方的で非均質な弾性体も多く存在し、このような弾性体の変形にも対応すべきである。そこで本研究では非等方的で非均質な弾性体および、等方的で均質な弾性体の双方を扱う。

提案法は以下の二つの貢献がある。第一の貢献は均質、非均質、等方的、非等方的いかににかかわらない任意の弾性体に対して変形シミュレーションを行える点である。第二の貢献は、適応的な分割と階層選択により、必要な部分のみ細かく、不必要な部分は粗いままに物体の変形を精度よく表現できる。この階層選択により更新する四面体の数を減らすことができ、結果的に総計算量を減らすことができる。

まず第一の貢献についてであるが、既存手法の多くは均質の弾性体を対象にしている。近年、非均質な弾性体に関する研究もなされ始めた。そのひとつにメッシュレスベースで非均質な弾性体の変形を実現したモデルがある<sup>14)</sup>。本手法はこのモデルとは異なり、メッシュベースなモデルである。有限要素法の既存手法の多くが非等方的な弾性体を対象にしている理由には、非等方的な弾性体の変形は計算量が多く計算コストが高いことがあげられる。

そこで階層的な分割及び適応的な階層選択手法を採用することでランタイム計算量の低減

<sup>†1</sup> 東京大学, The University of Tokyo

<sup>†2</sup> 国立台湾大学, National Taiwan University

を目指す。本手法は均質、非均質、等方的、非等方的な弾性体を包括的に扱えるモデルである。

次に第二の貢献であるが、有限要素法を用いる場合、分割数を多くすればするほど細かなめらかな変形が可能である。しかしそれと同時に計算量が増大してしまう。そこで本研究では、大きく変形する部分のみ細かな解像度でシミュレーションし、変形が小さい部分は粗い解像度でシミュレーションすることでランタイムの計算量の低減を目指す。

## 2. 関連研究

物理ベースの変形モデルはCGにおいて長い歴史がある。変形シミュレーションの歴史に関してはNealenらが調査をまとめている<sup>2)</sup>。本節では簡単に変形モデルの歴史を紹介する。

コンピューターグラフィックスの分野における変形の歴史の最初期においては質点間をバネで結ぶというMass Spring Modelが主流であった<sup>3)</sup>。このモデルでは曲げや捻りに対応すべくバネを追加している。

これまで紹介したバネモデルに比べ、より物理現象に即したシミュレーションを行うため、レギュラーグリッドに対して有限要素法を用いて数値シミュレーションするモデルがTerzopoulosらによって提案された<sup>4)5)</sup>。

Paiらはコセラ連続体理論に基づいたモデルを用いて薄い物体の大きな変形を実現した<sup>6)7)</sup>。

有限要素法はレギュラーグリッドにも適用可能であり複雑なトポロジーに対しても同様に適用可能である<sup>8)9)10)11)</sup>。これら有限要素法の手法は毎フレーム四面体や六面体、もしくは扱いやすい多面体に分割して各頂点に作用する力を計算し速度と座標を更新していく手法である。また、流動力学モデルにより、理想的な弾性体や粘弾性体、非均質物体、準非圧縮物体などを扱う方法も提案された<sup>12)</sup>。

これら有限要素法に忠実なモデルはリアルな一方で計算量が膨大である。ある程度のリアリティを維持する一方でモデルを簡略化することで計算量を減らす手法も現れ始めた。回転する変形フレームにより線形歪みを表現する手法がMullerらにより提案された<sup>13)</sup>。

既存手法では均質で等方的な弾性体を主に扱ってきた。提案法では均質、等方的という制約なしに均質、非均質、等方的、非等方的な弾性体を包括的に扱うことができなおかつ階層的な分割、および適応的な階層選択モデルを用いて変形時の計算量を減らす方法について提案する。

## 3. 物体の変形シミュレーション

提案手法を説明する前に弾性体の一般的な計算法の概略を述べる。物体は四面体の集合と

して表現されていると仮定する。変形シミュレーションの各ステップでは、各四面体の歪み(テンソル)に応じて応力がどのように作用するかを計算し、四面体の各頂点の加速度を求めて速度と位置を更新する。以下では、歪みテンソルから応力テンソルを計算する方法を概観する。以下、添字*i, j, k, l*は0, 1, 2の値をとり0, 1, 2の値がそれぞれ3次元の*x, y, z*成分に対応している。応力テンソル $\sigma_{ij}$ は歪み応力テンソル $\sigma_{ij}^e$ と粘性力テンソル $\sigma_{ij}^v$ の成分からなる。

$$\sigma_{ij} = \sigma_{ij}^e + \sigma_{ij}^v \quad (1)$$

歪み応力テンソル $\sigma_{ij}^e$ は四面体の歪みテンソル $\epsilon_{kl}$ と弾性定数(4階のテンソル) $C_{ijkl}$ を用いて

$$\sigma_{ij}^e = \sum_{k=0}^2 \sum_{l=0}^2 C_{ijkl} \epsilon_{kl} \frac{\partial \mathbf{x}}{\partial u_k} \cdot \frac{\partial \mathbf{x}}{\partial u_l} - \delta_{kl} \quad (2)$$

と表される。



図1 四面体のマテリアル座標系と世界座標系、及びその速度。

$\delta_{ij}$ は*i = j*で1、それ以外は0となるクロネッカーのデルタである。 $\mathbf{x}, \mathbf{u}$ は図1のようにそれぞれ世界座標系での座標、およびマテリアル座標系での座標を表す。ただしマテリアル座標系とは、O'Brienらの論文で定義された座標系であり<sup>10)</sup>、基準配置(ここでは外力が作用していない状態とする)における物体の各四面体の相対座標のことであり $\mathbf{u}$ で表される。 $\mathbf{x}$ はマテリアル座標 $\mathbf{u}$ が外力により変形、移動した結果の各四面体の相対座標である。変形具合を示す歪みテンソルはこのマテリアル座標 $\mathbf{u}$ と世界座標 $\mathbf{x}$ の違いから計算される。

同様に粘性応力テンソル $\sigma_{ij}^v$ も歪み速度テンソル $\nu_{ij}$ と粘性定数 $D_{ijkl}$ を用いて

$$\sigma_{ij}^v = \sum_{k=0}^2 \sum_{l=0}^2 D_{ijkl} \nu_{kl} \frac{\partial \mathbf{x}}{\partial u_k} \cdot \frac{\partial \mathbf{x}}{\partial u_l} + \frac{\partial \mathbf{x}}{\partial u_i} \cdot \frac{\partial \mathbf{x}}{\partial u_j} \quad (3)$$

と表される。

従来法では等方性の仮定から応力テンソルは

$$\sigma_{ij}^{\epsilon} \lambda \sum_{k=0}^2 \epsilon_{kk} \delta_{ij} + 2\mu \epsilon_{ij} \sigma_{ij}^{\nu} \phi \sum_{k=0}^2 \nu_{kk} \delta_{ij} + 2\psi \nu_{ij} \quad (4)$$

と 2 階のテンソルのみで表せるが, 本手法で扱う異方性のある構造は弾性定数, 粘性定数は簡略化できず 4 階のテンソルとなる.

マテリアル座標系における安定状態の四面体の座標を  $\mathbf{m}$  とするとパラメータ  $\mathbf{b}$  を用いて

$$\begin{pmatrix} \mathbf{u} \\ 1 \end{pmatrix} \begin{pmatrix} \mathbf{m}_1 & \mathbf{m}_2 & \mathbf{m}_3 & \mathbf{m}_4 \\ 1 & 1 & 1 & 1 \end{pmatrix} \mathbf{b} = M\mathbf{b} \quad (5)$$

となる. この式からパラメータ  $\mathbf{b}$  は

$$\mathbf{b} M^{-1} \begin{pmatrix} \mathbf{u} \\ 1 \end{pmatrix} \quad (6)$$

となることわかる. また変形後の座標, 速度は  $\mathbf{b}$  を代入して

$$\begin{pmatrix} \mathbf{x} \\ 1 \end{pmatrix} \begin{pmatrix} \mathbf{p}_1 & \mathbf{p}_2 & \mathbf{p}_3 & \mathbf{p}_4 \\ 1 & 1 & 1 & 1 \end{pmatrix} \mathbf{b} = P\mathbf{b} = P M^{-1} \begin{pmatrix} \mathbf{u} \\ 1 \end{pmatrix} \quad (7)$$

$$\begin{pmatrix} \dot{\mathbf{x}} \\ 1 \end{pmatrix} \begin{pmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \mathbf{v}_3 & \mathbf{v}_4 \\ 1 & 1 & 1 & 1 \end{pmatrix} \mathbf{b} = V\mathbf{b} = V M^{-1} \begin{pmatrix} \mathbf{u} \\ 1 \end{pmatrix} \quad (8)$$

となる. 両辺を  $u_i$  で偏微分して

$$\begin{pmatrix} \frac{\partial \mathbf{x}}{\partial u_i} \\ 1 \end{pmatrix} = P M^{-1} \begin{pmatrix} \frac{\partial \mathbf{u}}{\partial u_i} \\ 1 \end{pmatrix} = \begin{pmatrix} \delta_i \\ 1 \end{pmatrix} \quad (9)$$

$$\begin{pmatrix} \frac{\partial \dot{\mathbf{x}}}{\partial u_i} \\ 1 \end{pmatrix} = V M^{-1} \begin{pmatrix} \frac{\partial \mathbf{u}}{\partial u_i} \\ 1 \end{pmatrix} = \begin{pmatrix} \delta_i \\ 1 \end{pmatrix} \quad (10)$$

となる. これらの式を式 2, 式 3 に代入することで歪みテンソル  $\epsilon_{ij}$ , 歪み速度テンソル  $\nu_{ij}$  が計算できる. ただし  $\delta_i$  はクロネッカーのデルタを行列表示した時の  $i$  列目を縦に並べてつくったベクトルである.

構造定数  $C_{ijkl}$ ,  $D_{ijkl}$  が既知である場合は式 2 や式 3 に値を代入して応力テンソルが計算できる.

次にこの応力テンソル  $\sigma_{ij}$  を用いて四面体の各頂点に作用する力を計算をする. 四面体のマテリアル座標での体積  $\text{vol}$  は

$$\text{vol} = \frac{1}{6} (\mathbf{m}_2 - \mathbf{m}_0) \times (\mathbf{m}_1 - \mathbf{m}_0) \cdot (\mathbf{m}_3 - \mathbf{m}_0) \quad (11)$$

で与えられ, これと応力から四面体の各頂点に作用する力  $\mathbf{f}$  を並べて作った行列  $F$  は

$$F = -\frac{\text{vol}}{2} P M^{-1} \sigma M^{-1T} \quad (12)$$

となる. ただし  $\sigma$  は応力テンソル  $\sigma_{ij}$  の行列表現である.

各頂点は通常複数の四面体により共有されており, 各四面体の応力の総和が頂点にかかることでその頂点の速度が変化する. 変形の更新毎に最初に各頂点の加速度を  $\mathbf{0}$  に初期化しておき, 各頂点を共有する四面体からの応力  $\mathbf{f}_i$  を質量  $m_i$  で割って加速度を求め足すことでその時刻における頂点の加速度が求まる.

#### 4. 提案法の基本的な考え方

現実の物体は非均質であり  $C_{ijkl}$  や  $D_{ijkl}$  は空間変化する. また物体をリアルに表現するためには物体を細かく分割する必要があり, 全領域を同程度の大きさで分割すると四面体の数が増え過ぎてしまう.

そこで, 提案法では, 階層的な四面体分割に基づいたシミュレーション法を考える. 図 2

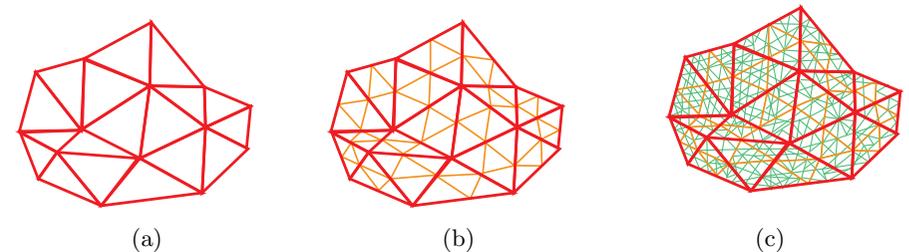


図 2 (a) ポリゴンで構成された物体をドロネー分割してできた四面体の集合. (b) レベル 0 の四面体を 1 回分割してできた四面体の集合. (c) レベル 0 の四面体を再帰的に  $n$  回分割してできた四面体の集合. 説明のため 2 次元で表している.

のように, まずポリゴンで構成された物体をドロネー分割で四面体にラフに分割する. この

四面体をレベル 0 の四面体と呼ぶ。次に再帰的に細かい四面体に分割する。分割数に応じてレベルが増えていく。レベルが高いほど細かい。

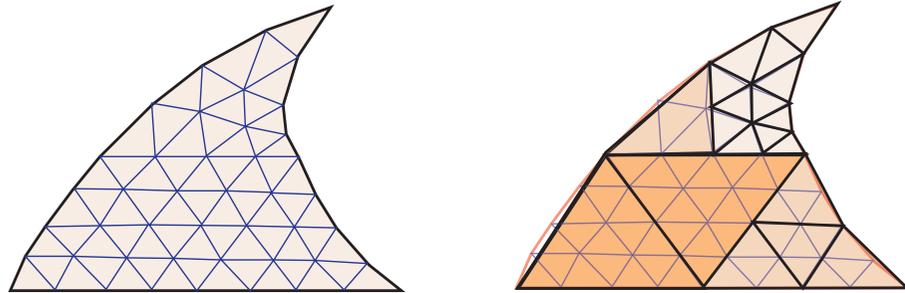


図 3 階層的な四面体分割の適応的な階層選択のイメージ図。説明を簡単にするため、ここでは二次元で表す。左側が全体が最も細かい解像度で表された場合。右側が最適な解像度で表された場合。

物体が変形するとき、変化が大きい部分と小さい部分が存在する。例えば図 3 のような例を考える。

変化が小さい部分では隣接する四面体の歪みテンソルの差は少ない。逆に変化が大きい部分では隣接する四面体の歪みテンソルの差が大きい。このことから変化の小さい部分は粗く分割し内部は歪みテンソルの値が同じであると近似し、変化が大きい部分はより細かく分割するというモデルを考える。このアダプティブなシミュレーションモデルを構成するためには以下の二つの問題点を解決する必要がある。まず第一点は、非均質で異方性がある物体を扱うので、 $C_{ijkl}$  や  $D_{ijkl}$  は物体を離散化する解像度に応じて値が異なる。このため、各解像度における  $C_{ijkl}$  や  $D_{ijkl}$  を導出する必要がある。二点目は、適切な解像度をどのように選択するかということである。

以下ではまず一点目の問題を解決するために、粗い解像度における  $C_{ijkl}$  や  $D_{ijkl}$  を計算する方法を導出する。次に、二点目の問題を解決するために、粗い解像度でシミュレーションした場合に生ずる近似誤差に注目し、この近似誤差の大きさに基づいて、適切に解像度を選択するための方法を提案する。

#### 4.1 異なる解像度における弾性定数 $C_{ijkl}$ の計算法

物体の表現は階層構造を持った四面体構造で与えられており、最も細かい解像度における各四面体の弾性定数  $C_{ijkl}$  や粘性定数  $D_{ijkl}$  が既知であることを仮定する。以下では細かい

解像度の弾性定数から一階層粗い解像度の弾性定数  $C_{ijkl}$  を計算する方法を述べる。これを繰り返すことで全解像度の弾性定数  $C_{ijkl}$  を求めることができ、さらにその計算は事前に計算可能である。

まずある解像度に属する四面体を  $T^{\text{coarse}}$  とし、これに対応する 1 段階細かい四面体の集合を  $\{T_m^{\text{fine}}\}$  とする。粗い解像度の四面体の変形で細かい解像度の四面体の変形を近似できる場合（近似可能な条件は後述する）を考える。さらに粗い解像度でのシミュレーションでの便宜性を考え

$$\sigma_{ij}^{\text{coarse}} = \sum_{k=0}^2 \sum_{l=0}^2 C_{ijkl}^{\text{coarse}} \epsilon_{kl} \quad (13)$$

を充たすとする。

$T_i^{\text{fine}}$  から  $T_i^{\text{coarse}}$  の弾性定数  $C_{ijkl}$  を求める方法として仮想的に変形をして近似した値を求める方法提案する。計算の都合上、最初に弾性定数を求め、次に粘性定数を求める。静止している場合は、式 3 より  $\nu_{ij} = 0$  となり、変形して釣り合っている状態では粘性定数がかかる項が 0 になるので、応力と変形に寄与する係数は弾性定数のみとなるからである。弾性

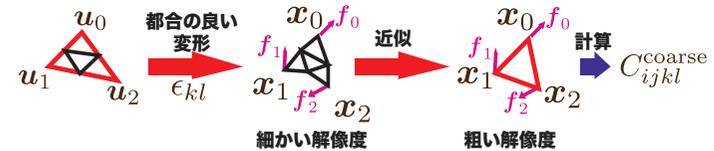


図 4  $T_i^{\text{fine}}$  から  $T_i^{\text{coarse}}$  の弾性定数を、パラメータを逐次変えて計算して求める方法。ある変形  $\epsilon_{kl}$  の状態で静止させるために各頂点に外力  $f_i$  を加える。ここでは簡単のため二次元で表している。

定数を求めるには、あらかじめ四面体のある歪みテンソルにしたがって変形しておき、この物体を静止させるために加えるべき外力を求める。外力と応力が釣り合っていると仮定すると、四面体の応力テンソルを推定することができるので、歪みテンソルと応力テンソルの関係により、弾性定数  $C_{ijkl}$  を求めることができる。ここで式 2 を見ると添字  $i, j, k, l$  の可換性、すなわち

$$C_{ijkl} = C_{ijlk}, C_{ijkl} = C_{jikl}, C_{ijkl} = C_{klij} \quad (14)$$

から応力  $\sigma$  及び歪みテンソル  $\epsilon$  は対称行列で表され独立な変数は 6 個である。また弾性定数  $C_{ijkl}$  も同様の可換性から 21 個の独立変数からなることが求まる。

この性質を利用して四面体を以下の歪みテンソルにしたがって変形する。この変形に関してはお互いに一次独立な歪みテンソルを取る限りは任意の変形が許されるが計算の簡単のため、以下の基底をとる。

$$\epsilon^b = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \\ \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \quad (15)$$

この基底をとると、式13の総和部分が $\epsilon$ の対角成分1つ、もしくは $\epsilon$ の非対角成分の対称な2つの成分の合計となり $C_{ijkl}$ を求めるのに簡潔な形となる。

基底 $\epsilon_{kl}^b$ に対して変数 $\alpha$  ( $\alpha \in \mathbf{R}$ ) を用いて $\alpha\epsilon_{kl}^b$ という歪みテンソルを考える。この歪みを適用したときに推定された応力テンソルを $\sigma_{ij}(\alpha)$ とすれば、粗い解像度での変形で近似が成り立つ限りは $\sigma_{ij}(\alpha)$ と $\alpha\epsilon_{kl}^b$ の間に線形の関係がなりたち、 $\sigma_{ij}(\alpha) = \alpha C_{ijkl}\epsilon_{kl}^b$ と表されると仮定する。そこで $C_{ijkl}$ を決めるためには複数の $\alpha$ に対して $\sigma_{ij}(\alpha)$ を求めて、その関係をフィッティングすればよい。

ここで $\epsilon$ から実際の変形 $D\mathbf{x}/D\mathbf{u}$ を求める必要がある。式2から

$$\frac{\partial \mathbf{x}}{\partial u_k} \cdot \frac{\partial \mathbf{x}}{\partial u_l} = \alpha \epsilon_{kl}^b + \delta_{kl} \quad (16)$$

であり、左辺はJacobian  $J = D\mathbf{x}/D\mathbf{u}$ と $J$ の転置 $J^T$ の積としてかける。この $J$ が求めたい式である。

$J$ は対称行列であり $JJ^T = JJ = A = \alpha\epsilon_{kl}^b + \delta_{kl}$ を満たす。 $A$ を対角化した行列を $B$ とすると、 $A = U^{-1}BU = U^{-1}\sqrt{B}\sqrt{BU} = U^{-1}\sqrt{BUU^{-1}}\sqrt{BU}$ となるので、 $J = U^{-1}\sqrt{BU}$ とすれば、 $A = J^TJ = JJ$ を満たす。

次に仮想的な変形から計算した各頂点に作用する力 $\mathbf{f}$ から応力テンソル $\sigma$ を求める。これは式12から

$$\sigma = -\frac{2}{\text{vol}}MP^{-1}FM^T \quad (17)$$

として求める。

次に $C_{ijkl}$ を近似的に求める方法について説明する。

まず式15の1つ目の基底 $\epsilon^1$ について求めた $\sigma_{ij}$ からどのように $C_{ijkl}$ を求めるかについて説明する。式13に代入すると、 $(k, l) = (0, 0)$ 以外の成分は0であることを考えると

$$\sigma_{ij} = \sum_k \sum_l C_{ijkl}\alpha\epsilon^1 = C_{ij00}\alpha \quad (18)$$

となる。よって $\sigma$ は $\alpha$ に線形になり線形フィッティングすることでその傾きから $C_{ij00}$ をきめることができる。2つ目の基底 $\epsilon^2$ 、3つ目の基底 $\epsilon^3$ についても同様のアプローチで $C_{ij11}$ 、 $C_{ij22}$ を決めることができる。

次に残りの基底について議論する。まず4つ目の基底 $\epsilon^4$ について説明する。式13に代入すると、 $(k, l) = (0, 1), (1, 0)$ 以外の成分は0であることを考えると

$$\sigma_{ij} = \sum_{k=0}^2 \sum_{l=0}^2 C_{ijkl}\alpha\epsilon^4 = (C_{ij12} + C_{ij21})\alpha \quad (19)$$

となる。

ここで $(k, l)$ の対称性から $C_{ijkl} = C_{ijlk}$ となり、 $\sigma$ と $\alpha$ について線形フィッティングして求められた傾きの半分が求める値、 $C_{ij01}$ 、 $C_{ij10}$ となる。5つ目の基底 $\epsilon^5$ 、6つ目の基底 $\epsilon^6$ についても同様。以上の手続きにより全 $C_{ijkl}$ を決めることができる。

線形フィッティングについては最小二乗法を用いて傾きを求め、その値（もしくは1/2をかけた値）を $C_{ijkl}$ とする。

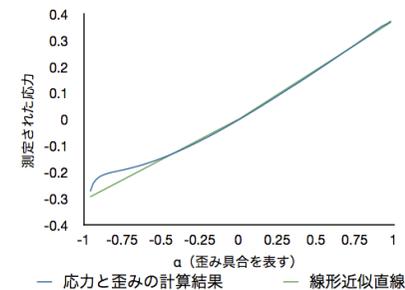


図5 計算に都合のよい変形に対する応力を計算し最小二乗法でフィッティングした図。

大きく変形をすればするほど線形からそれる可能性が高くなる。よって $\alpha$ の絶対値が小さい方に重みを大きくしてフィッティングした。例えば図5は四面体の弾性定数を求める際に、

$\epsilon_{00}$  の基底の応力の  $\sigma_{00}$  成分を計算し見やすくするためにプロットしたものである。(プログラム内部ではフィッティングして自動的に傾きを求める。) この例では  $\alpha$  が -1.0 から 1.0 について計算している。-0.5 以上の領域ではよく線形に近似されているがこの領域外では近似精度が悪いことがわかる。近似精度の悪い領域では細かい解像度で計算し、領域内では粗い解像度で近似された値で計算すればよい。フィッティングした直線からのずれから判定条件を決める。次の節ではこの判定条件について示す。

#### 4.2 階層選択条件

この節では粗い解像度とそれに対して 1 段階細かい四面体の集合についてどのような条件で粗い解像度による近似を採用するかを階層選択条件について述べる。

本手法では近似が成り立つ条件を、近似した四面体において式 13 が充たされる、としている。換言すれば、近似が成り立たない条件は、応力の計算結果が上記のような線形フィッティングした傾きから大きくずれた場合を表す。

計算した値を  $\sigma(\alpha)$ 、線形フィッティングして得られた傾きを  $C$  としたときに

$$\frac{|C\alpha - \sigma(\alpha)|}{(C\alpha + \Delta)} < \tau \quad (20)$$

を近似成立条件とした。ここで  $\tau$  は近似の精度、 $\Delta$  は  $\alpha$  が 0 に近い時の誤差を吸収するための値である。本稿では  $\Delta = 0.01, \tau = 0.01$  としている。

基底  $\epsilon_{kl}$  毎に 9 種類の応力  $\sigma_{ij}^{(kl)}$  を求めフィッティングするがそれぞれについて近似条件を満たさなくなる値  $\alpha_{ij}^{(kl)}$  のうち最小の値もの  $\min_{ij}(\alpha_{ij}^{(kl)})$  を基底  $\epsilon_{kl}$  に対する許容変形値として採用する。

シミュレーション時に毎フレームアクティブな近似四面体の歪みテンソル  $\epsilon_{kl}$  を求め、各  $kl$  についてその四面体の許容変形値  $\alpha_{kl}$  と比較しその一つでも満たしていなければ一段階細かくして再計算する。

### 5. 実験結果

提案法を C++ と OpenGL を用いて実装し、8.00GB RAM, Intel(R) Core(TM) i7 CPU X980 3.33GHz, NVIDIA GeForce GTX480 を搭載した PC で実験を行った。

図 6 は最大がレベル 3 の四面体が地面に衝突するシーンのシミュレーションである。ここで最大レベル  $n$  の弾性体のシミュレーションとは、本手法を用いて  $n$  レベルまで分割した弾性体を適応的に階層選択する手法でシミュレーションすることをいい、以降このように表現する。最大がレベル 3 の四面体 (四面体数 512 個) の全てのレベルの構造定数の計算に要

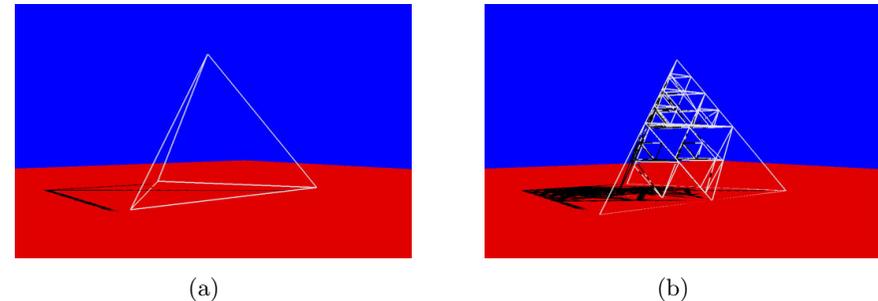


図 6 衝突前後の階層の変化を表した図。(a) は衝突直前、(b) は衝突後。歪みが伝わり先端が密となり細かい階層が選択されている。

した時間は 51 分 04 秒であった。図 6(a) は衝突前であり、変形が小さいためレベル 0 で近似されている。衝突直後、図 6(b) のように一部が大きく変形し細かい解像度 (レベル 1~レベル 3) になっている。

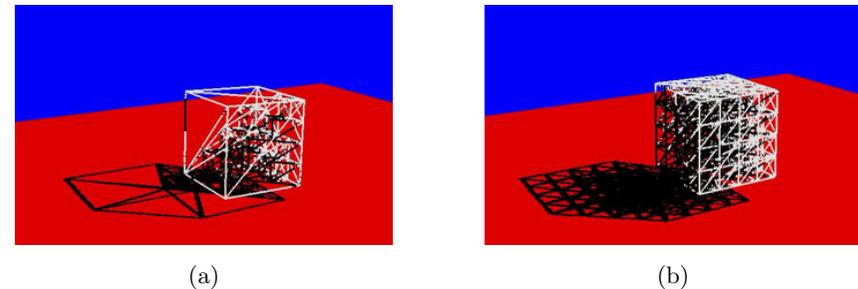


図 7 (a) 適応的な階層選択をした変形と (b) 最も細かい階層での変形の比較。

適応的な階層選択をした変形と最も細かい階層での変形とを比較した図が図 7 である。実験にはレベル 2 の四面体を用いた。変形の計算に要した時間は、本手法を用いた場合には 1ms 以下であったのに対し、最も細かい階層で全て計算した場合には常に 4ms から 5ms の時間を要した。ランタイムの計算速度は明らかに本手法のほうが優れていることがわかる。その一方で (a) と (b) は同時刻での変形の様子を捉えたものである。よく見ると四面体の傾きが異なっており最も細かい階層の変形とはずれてしまっている。これはレベルが変更する

際に生じる誤差に依るところが大きく、レベル変更方法について改良する必要があることがわかる。

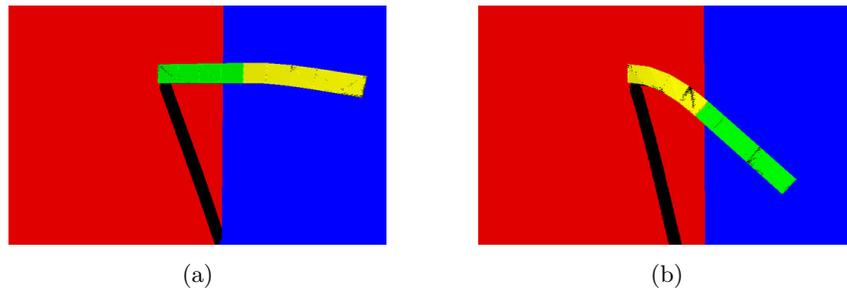


図8 硬さの異なる部分が連続している時の変形。(a) 壁側が硬い場合 (b) 壁側が柔らかい場合。

非均質な物体の変形の例として柔らかい素材でできた部分と硬い素材でできた部分を逆にしたときの形状の変化の実験を行った。結果は図8の通りである。緑の部分が黄色の部分に比べて100倍硬く設定されている。図8(b)では壁側が硬い素材でできており、まっすぐのび、柔らかい部分は接合部から重力により下に垂れ下がっている。一方で図8(a)では壁側が柔らかい素材でできており、壁からすぐに下へと垂れ下がっている。壁との接合部にかかる質量は図8(a)の硬い素材との接合部に比べて大きいため、より角度がついて垂れ下がっていることがわかる。

最後にレベル3の四面体の落下アニメーションの結果を図9に示す。パラメーターの値はそれぞれ  $\lambda = 0.419 \times 10^{-2}$ ,  $\phi = 0.419 \times 10^{-7}$ ,  $\mu = 0.516 \times 10^{-2}$ ,  $\psi = 1.44 \times 10^{-7}$ ,  $\rho = 0.5e \times 10^{-5}$  として計算した。落下直前まではマテリアル座標系と一致しているのでレベル0のままで計算されており、また落下後も一部は粗い階層が選択されている。

## 6. まとめと今後の課題

本稿では等方的、非等方的、均質、非均質な弾性体を包括的に取り扱うことが可能なアダプティブな変形シミュレーションモデルを提案した。細かな解像度で物体を離散化した場合の性質から粗い細かな解像度で物体を離散化した場合の性質を導出し、粗い解像度で物体を離散化した場合で近似が許される条件とその評価法を提案した。

提案法を用いれば、変形時に変化の少ない部分は粗い解像度で変形をシミュレーションし、

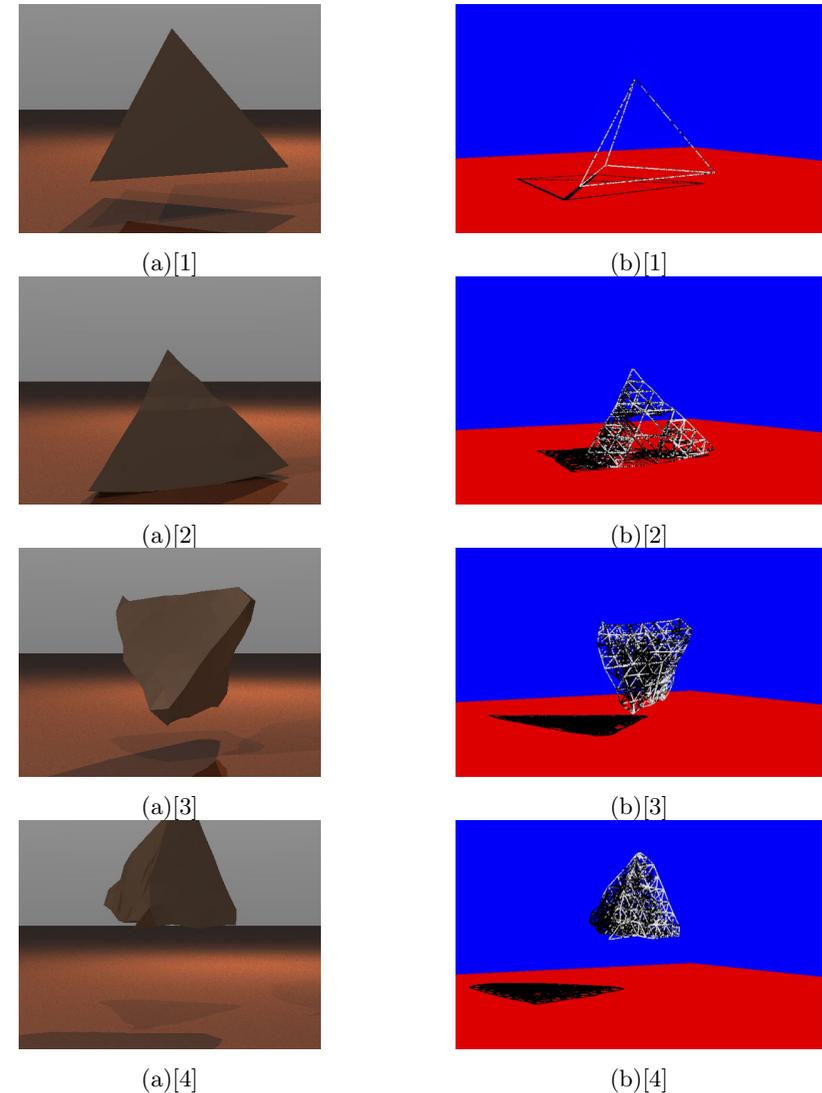


図9 Level3の四面体の変形アニメーション。(a) レンダリング結果 (b) ワイヤフレーム表示。粗い四面体の部分は粗い階層が選択されている。[1] 落下開始 55 フレーム (衝突前)。[2] 落下開始 66 フレーム (衝突直後)。[3] 落下開始 99 フレーム。[4] 落下開始 120 フレーム

変化の大きな部分は細かい解像度でシミュレーションすることで全領域を細かい解像度で計算する場合よりも少ない計算量でシミュレーション可能になると考えられる。

構造定数のうち弾性定数  $C_{ijkl}$  の近似法についてのシミュレーションを提案したが粘性定数  $D_{ijkl}$  の近似法についてはまだ議論できていない。将来的には粘性定数  $D_{ijkl}$  も同様にして求められると考える。

現在は最初の近似計算も変形時の計算もシングルスレッドで計算をしているが、本手法は各四面体毎についての計算が多く、並列化に向いている手法であると考えられる。将来的には GPU 上で動かすことで高速化する予定である。

弾性体のみをシミュレーションの対象としているが塑性をもつ弾性体も扱えると考えられる。また変形だけではなく破壊に対しても適用することが今後の課題である。

### 参 考 文 献

- 1) A. Norton, G. Turk, B. Bacon, J. Gerth, and P. Sweeney. Animation of fracture by physical modeling. *The Visual Computer*, 7(4):210-217. (1991).
- 2) A. Nealon, M. Muller, R. Keiser, E. Boxerman, and M. Carlson. Physically based deformable models in computer graphics. *Comput. Graph. Forum*. Vol. 25 (4). 809-836. (2005).
- 3) S.Platt, N. Balder Animating facial expressions. *SIGGRAPH Comput.Graph.*, 245-252. (1981)
- 4) D. Terzopoulos, J. Platt, A Barr, and K. Fleischer. Elastically deformable models. *SIGGRAPH Comput. Graph*. Vol. 21. 205-214. (1989).
- 5) D. Terzopoulos and H. Qin . Dynamic nurbs with geometric constraints for interactive sculpting. *ACM Trans. Graph*. 13, 2, 103-136. (1994).
- 6) D. K. Pai. STRANDS: Interactive simulation of thin solids using Cosserat models. *Comput. Graph. Forum*. 32(3). 347-352. (2002).
- 7) B. Nadler and M. Rubin. A new 3-d finite element for nonlinear elasticity using the theory of a cosserat point. *Int. J. of Solids and Struct*. 40, 4585-4614. (2003).
- 8) K. Bathe. *Finite Element Procedures*. Prentice Hall.(1996).
- 9) J. P. Gouret, N. M. Thalmann, and D. Thalmann. Simulation of object and human skin formations in a grasping task. *SIGGRAPH Comput. Graph*. 23, 3, 21-30. (1989).
- 10) J. O' Brien, and J. Hodgins. Graphical models and animation of brittle fracture. *SIGGRAPH Comput. Graph.*, 137-146. (1999).
- 11) S. Cotin, H. Delingette, and N. Ayache. Real-time elastic deformations of soft tissues for surgery simulation. *IEEE TVCG* 5, 62-73. (1999).
- 12) J. Weiss, B. maker, and S. govindjee. Finite element implementation of incompressible, transversely isotropic hyperelasticity. *Comp. Meth. in App. Mech. and Eng*. 135, 107-128. (1996).
- 13) M. Muller, and M. Gross. Interactive virtual materials. In *Graphics Interface* 239-246. (2004).
- 14) F. Francois, G. Benjamin, B. Guillaume and D. K. Pai. Sparse Meshless Models of Complex Deformable Solids. *ACM Trans. Graph*. 30 (4), 73:1-73:10. (2011).