

例示ベースの弾性変形の実時間計算手法

Real-Time Example-Based Elastic Deformation

小山 裕己[†] 高山 健志^{†,*} 梅谷 信行[†] 五十嵐 健夫^{†,‡}

Yuki KOYAMA[†] Kenshi TAKAYAMA^{†,*} Nobuyuki UMETANI[†] Takeo IGARASHI^{†,‡}

[†]東京大学 (The University of Tokyo) [‡]JST ERATO

E-mail: koyama@is.s.u-tokyo.ac.jp

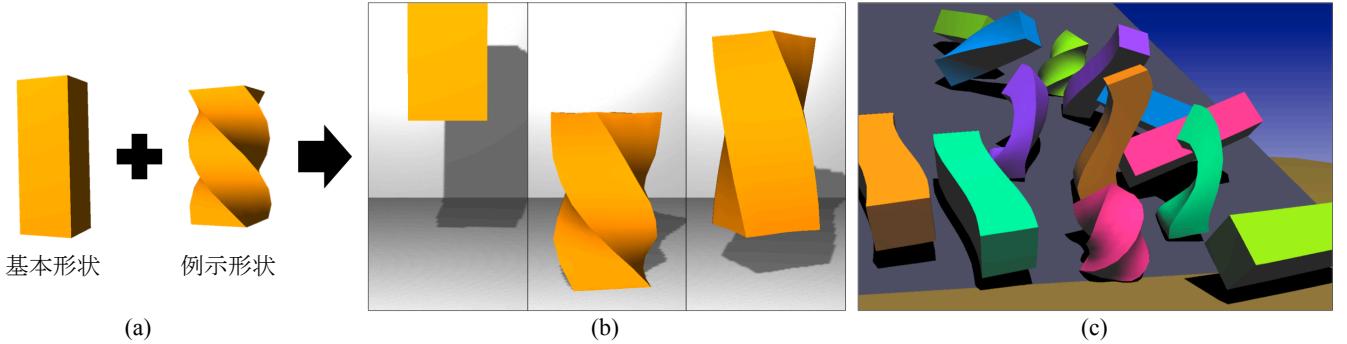


図1: (a) ユーザによる入力形状. (b) 生成された例示ベースの弾性体の落下のアニメーションの様子. (c) 2種類の例示ベースの弾性体 ((a)で生成されたものと, (a)と同じ基本形状に対してS字形の例示形状を与えたもの) を用いて生成したシーンにおけるアニメーションの様子. このシーンはラップトップ (CPU: Core i7 M620 2.66GHz, GPU: GeForce GT330M) 上で実時間 (約14 fps) で計算されたものである.

概要

例示ベースの弾性変形は弾性体のアニメーションの作成を支援する手法であるが、既存手法では計算コストが高く、実時間では計算することができない。そこで本稿では Shape Matching 法による弾性体の表現を拡張することによって、例示ベースの弾性変形を実時間で計算できる高速な手法を提案する。

1. はじめに

リアルな CG アニメーションを製作するには物理シミュレーションを用いることが効果的である。そこで、アーティストが演出意図を実現しやすいような物理シミュレーションの手法が重要となる。そのような手法としては、流体[12]、剛体[13]、布[1]、弾性体[5]などを対象とした手法が提案してきた。

そのような中、Martin ら[6]は例示ベースの弾性体という概念を提案した。これは弾性体のアニメーションの作成を支援するもので、弾性体の変形形状の例（例示形状）をアーティストが指定することで、その形状

に変形しやすい弾性体を実現できるというものである。例えば図1 ((a), (b)) は四角柱の形状をした弾性体に対してねじれた形状を例として与えた場合の挙動を示したものであり、このようにねじれやすい性質を持つ四角柱を簡単に表現できる。例示ベースの弾性体を用いることで、このようなアニメーションを作成する際に非直感的な物理パラメタの調整をする必要がなく、また変形形状を直接デザインできるという利点がある。また、キーフレームを用いたアプローチ[5]に比べ、アニメーションのシナリオが固定である必要がないという点も、例示ベースの弾性体の特徴である。すなわち、どのようなシーンでどのような時系列で変形するかなどといった情報が、変形の性質のデザインの段階では必要ない。これにより、様々なシナリオで同じ弾性体を使い回すことができたり、シナリオを事前に決定できないインタラクティブなアプリケーションで使用することができますといった利点が期待される。

しかしながら、Martin らの手法は計算コストが高く、例示ベースの弾性変形を実時間で計算することができないという問題点がある。Martin らの手法は有限要素法による弾性体の表現を用いており、例示の効果を実

*現在の所属は ETH Zurich.

現するために特別な保存力を計算している。その際に非線形最適化が必要となり、これが計算のボトルネックとなってしまっているのである。これによって、インタラクティブなアプリケーションでの使用が制限されてしまっている。

このような制限を克服するため、本稿では特に実時間でも計算できる高速な例示ベースの弾性変形の手法を提案する。本提案手法は Shape Matching 法による弾性体の表現[8]に基づいており、保存力を計算する代わりに弾性体の目標の形状そのものを変化させながらシミュレーションを行うことによって、非線形最適化を用いずに例示による効果を実現している。本提案手法は Martin らの手法と似たような変形を再現しつつ、数百倍程度高速に計算でき、特に実時間アプリケーションでも使用することができる。

Shape Matching 法は Martin らの用いた有限要素法に比べると物理的な正確さという点では劣るが、見た目には十分自然な挙動を表現できており、また特にアーティストの演出意図が重視される文脈においては見た目以上の物理的な正確さはあまり重要でないと考えられるため、本提案手法は十分実用性があるものと考えられる。また Martin らと同様に、本提案手法においても例示の効果を局所化させる拡張を行うことが可能である。

2. 関連研究

Shape Matching 法を用いた弾性体の表現は高速で安定なものであり、Müller ら[8]によって初めて提案されて以来、様々な拡張が提案してきた。FastLSM [9]は格子状にパーティクルを配置した構造に対して効率的に計算を行う手法である。Steinemann ら[11]はこれに対し八分木のデータ構造を導入することで更に効率的に計算する手法を提案した。Chain Shape Matching [10]は鎖状に並んだパーティクルに対して Shape Matching 法を適用する手法で、複雑なヘアスタイルを表現することができる。Oriented Particles [7]は各パーティクルに対して位置だけでなく回転の情報も持たせることによって、非常に少ないパーティクルでも安定したシミュレーションを行えるようにした拡張である。

このように Shape Matching 法は様々な拡張が提案されているが、我々の知る限り、例示ベースの弾性変形を実現するのは本提案手法が最初である。また、本

提案手法は上記のような拡張と組み合わせて用いることも可能である。

また ProcDef [4]は Shape Matching 法を拡張することで能動的な動作を生成する手法である。この手法では各局所領域の目標の形状を能動的に変えていくことで全体の動きを表現している。これに対して本提案手法では各局所領域の目標の形状がそのときの変形状態に応じて受動的に変わっていく点が異なっている。

3. 提案手法の概観

本提案手法では、次の 2 つの操作を各タイムステップで行うことで例示ベースの弾性変形を実現する。

1. 現在の形状を例示集合 E (基本形状と例示形状の線形補間, §4.2.) に投影する。(§4.3.)
2. 投影先の点を目標の形状とし、現在の形状を目標の形状に向かって引っ張る。(§4.1.)

これらの操作を模式的に表したのが図 2 である。ただし、ここでは 1 つの基本形状に対して 2 つの例示形状を与えた場合を表しており、それぞれの形状はベクトルによって記述 (§4.2.) されている。

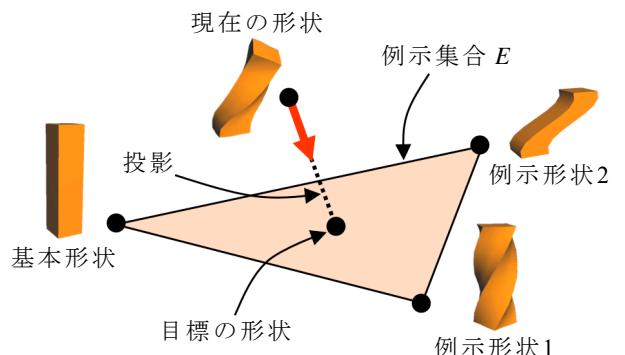


図 2: 提案手法の概観

4. 提案手法

4.1. Shape Matching 法の拡張

本提案手法は Müller ら[8]の提案した Shape Matching 法による弾性体の表現を拡張することで例示ベースの弾性体を表現する。特に、互いにオーバラップした局所領域を持つ構造の Shape Matching 法(これは[9], [4]などでも使用されている)を扱う。すなわち、局所領域毎に Shape Matching 法を適用し、それらの影響を合併して全体の変形を実現する。

本提案手法におけるユーザの入力は、モデルの基本形状(初期形状)を表す四面体メッシュと、 n 種類の例示形状である。例示形状とは基本形状を変形させたもので、基本形状の四面体メッシュと同じ構造を保つ

たまま、頂点の位置を自由に移動させたものである。また、四面体メッシュに対してレンダリング用の詳細な三角形メッシュを埋め込むことも可能である。

四面体メッシュの頂点はそれぞれ1つのパーティクルとして扱われ、パーティクル*i*に対して1つの局所領域*N_i*が定義される。局所領域*N_i*の要素はパーティクル*i*とその一近傍頂点(1-ring neighborhood [4])である。なお、本提案手法では四面体メッシュを基にした局所領域の定義を採用したが、[9]のように格子状の構造によって局所領域を定義するなどしても同様に例示による効果は得られると考えられる。

パーティクル*i*に対応する初期位置、現在位置、質量をそれぞれ $\mathbf{x}_i^0, \mathbf{x}_i, m_i$ とする。各タイムステップについて、まず局所領域*r*における初期形状からその時刻での形状への変換を近似した線形変換行列

$$\mathbf{A}_r = \left(\sum_{i \in N_r} \tilde{m}_i \mathbf{p}_i \mathbf{q}_i^T \right) \left(\sum_{i \in N_r} \tilde{m}_i \mathbf{q}_i \mathbf{p}_i^T \right)^{-1} \in \mathbb{R}^{3 \times 3} \quad (1)$$

を計算する。ここで、 $\tilde{m}_i = \frac{m_i}{|N_r|}$ は有効質量[9]、 $\mathbf{c}_r^0 = \frac{\sum_{i \in N_r} \tilde{m}_i \mathbf{x}_i^0}{\sum_{i \in N_r} \tilde{m}_i}, \mathbf{c}_r = \frac{\sum_{i \in N_r} \tilde{m}_i \mathbf{x}_i}{\sum_{i \in N_r} \tilde{m}_i}$ はそれぞれ初期状態とその時刻での局所領域*r*における重心、 $\mathbf{p}_i = \mathbf{x}_i - \mathbf{c}_r, \mathbf{q}_i = \mathbf{x}_i^0 - \mathbf{c}_r^0$ は重心からの相対位置を表している。続いて線形変換行列 \mathbf{A}_r をその歪み成分である $\mathbf{S}_r = \sqrt{\mathbf{A}_r^T \mathbf{A}_r} \in \mathbb{R}^{3 \times 3}$ と回転成分である $\mathbf{R}_r = \mathbf{A}_r \mathbf{S}_r^{-1} \in \mathbb{R}^{3 \times 3}$ に分解する。次に、このタイムステップにおけるパーティクル*i*の局所領域*r*に関する目標位置 $\mathbf{g}_{r,i}$ を計算するのだが、従来のShape Matching法では初期形状の形状保存変換、すなわち

$$\mathbf{g}_{r,i} = \mathbf{R}_r (\mathbf{x}_i^0 - \mathbf{c}_r^0) + \mathbf{c}_r \quad (2)$$

によって計算していたのに対し、本提案手法では

$$\mathbf{g}_{r,i} = \mathbf{R}_r \tilde{\mathbf{S}}_r (\mathbf{x}_i^0 - \mathbf{c}_r^0) + \mathbf{c}_r \quad (3)$$

によって計算する。ここで、 $\tilde{\mathbf{S}}_r \in \mathbb{R}^{3 \times 3}$ は目標の形状を歪ませる(図3)要素であり、§4.2.で説明する「例示集合」に対して、§4.3.で説明する「投影」という操作を行うことで得ることができる。

最後に従来のShape Matching法と同様、パーティクル*i*の目標位置 \mathbf{g}_i を、パーティクル*i*が属する全ての局所領域での目標位置の平均 $\mathbf{g}_i = \langle \mathbf{g}_{r,i} \rangle_{i \in r}$ とし、これを用いてパーティクル*i*の位置 \mathbf{x}_i と速度 \mathbf{v}_i を

$$\mathbf{v}_i(t+h) = \mathbf{v}_i(t) + \alpha \frac{\mathbf{g}_i(t) - \mathbf{x}_i(t)}{h} + h \frac{\mathbf{f}_{\text{ext}}(t)}{m_i} \quad (4)$$

$$\mathbf{x}_i(t+h) = \mathbf{x}_i(t) + h \mathbf{v}_i(t+h) \quad (5)$$

のように更新する。ここで h は1タイムステップあたりの時間幅、 $\alpha \in (0, 1]$ は硬さに関するパラメタ、 \mathbf{f}_{ext} は外力を表す。

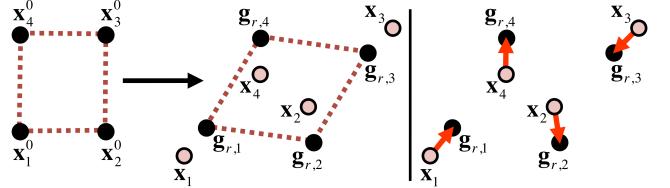


図3: 拡張した Shape Matching 法

4.2. 変形記述子と例示集合

Martinらの手法は、例示の効果を実現するために好みの変形の集合(例示集合)のうち、その時刻における変形に最も近い変形を探し、その変形に近づくように特別な保存力を計算するというものである。そこで彼らはまず、グリーンの歪みテンソルを用いて変形記述子を定義した。具体的には、四面体の数が*m*、頂点の数が*m'*の四面体メッシュの変形時の頂点の位置 $\mathbf{x} \in \mathbb{R}^{3m'}$ に対し、四面体*i*のグリーンの歪みテンソルを $\mathbf{E}_i \in \mathbb{R}^6$ として、 $\mathbf{E}(\mathbf{x}) = (\mathbf{E}_1^T \dots \mathbf{E}_m^T)^T \in \mathbb{R}^{6m}$ を変形 \mathbf{x} に体する変形記述子と定義した。なお、グリーンの歪みテンソルは 3×3 の対称行列として得られるものであるが、ここでは6次元のベクトルとして表現している。ここで、写像 $\mathbf{x} \mapsto \mathbf{E}(\mathbf{x})$ の値域 $F \subset \mathbb{R}^{6m}$ (「実現可能集合」と呼ぶ)は非線形な集合である。すなわち、線形結合 $w_1 \mathbf{E}(\mathbf{x}^1) + w_2 \mathbf{E}(\mathbf{x}^2)$ は一般には*F*の要素ではなく、これに對応する有効な変形形状は存在しない。そこで彼らは、 $\left\{ \sum_{k=0}^n w_k \mathbf{E}(\mathbf{x}^k) \right\}$ (\mathbf{x}^0 は初期形状、 \mathbf{x}^k は*k*番目の例示形状を表す)によって得られる凸包を実現可能集合*F*に投影して得られる集合*E*を例示集合と定義した。このように定義された例示集合*E*に対し、各タイムステップにおける変形形状を非線形最適化によって投影することで、特別な保存力の計算に必要な目標となる変形形状 $\tilde{\mathbf{x}}$ を得る。また、この非線形最適化が計算のボトルネックとなっている。(図4 左)

Martinらが特別な保存力を計算することによって例示の効果を実現していたのに対し、本提案手法ではShape Matching法における各局所領域の目標の形状を適切に歪ませる(図3)ことによって例示の効果を実現する。まず変形記述子の定義についてであるが、付録で詳しく説明するように、式(1)で計算される線形変換行列 \mathbf{A}_r は局所領域*r*における変形勾配テンソルの近似と考えることができ、したがってその歪み成分である対称行列 \mathbf{S}_r は右ストレッチテンソルの近似と考えることができる。そこで本提案手法では、グリーンの歪みテンソルの代わりに右ストレッチテンソル

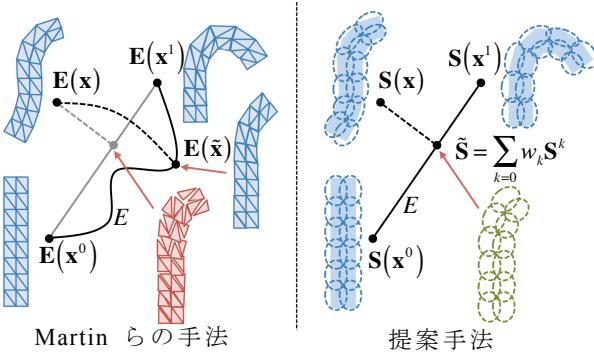


図4: Martin らの手法との比較

と用いて、変形記述子を $\mathbf{S}(\mathbf{x}) = (\mathbf{S}_1^T \cdots \mathbf{S}_m^T)^T \in \mathbb{R}^{6m}$ と定義する。なお、ここでは \mathbf{S}_r を 6 次元のベクトルとして表現している。ここで、Shape Matching 法では局所領域毎に独立にパーティクルの目標位置を計算するため、記述子に対応する変形形状が実際に存在しない場合でも有効な変形を実現できる点が、有限要素法を用いた Martin らの手法と大きく異なっている。これにより、本提案手法では記述子の単純な線形結合 $\left\{ \sum_{k=0}^n w_k \mathbf{S}(\mathbf{x}^k) \right\}$ によって得られる凸包を例示集合 E と定義することができる。このように定義した例示集合 E に対して、その時刻における変形形状を線形投影（詳しくは §4.3.で説明する）することによって、目標となる変形の記述子 $\tilde{\mathbf{S}}$ 、更に局所領域 r における目標となる形状を変形させる成分 $\tilde{\mathbf{S}}_r$ を得ることができる。（図4 右）

4.3. 例示集合への投影

基本形状と k 番目の例示形状を表した変形記述子をそれぞれ $\mathbf{S}^0 = \mathbf{S}(\mathbf{x}^0), \mathbf{S}^k = \mathbf{S}(\mathbf{x}^k)$ と表すと、例示集合 E に対してその時刻における形状 $\mathbf{S} = \mathbf{S}(\mathbf{x})$ を投影するとは、 \mathbf{S} に最も近い例示集合 E の要素 $\tilde{\mathbf{S}} = \sum_{k=0}^n w_k \mathbf{S}^k$ を得るということである。そこでまず

$$\left\| \sum_{k=1}^n w_k (\mathbf{S}^k - \mathbf{S}^0) - (\mathbf{S} - \mathbf{S}^0) \right\|^2 \quad (6)$$

を最小にする重み w_1, \dots, w_n を計算する。具体的には、 $\mathbf{w} = (w_1 \cdots w_n)^T \in \mathbb{R}^n, \mathbf{L} = (\mathbf{S}^1 - \mathbf{S}^0 \cdots \mathbf{S}^n - \mathbf{S}^0) \in \mathbb{R}^{6m \times n}$ として

$$\mathbf{w} = (\mathbf{L}^T \mathbf{L})^{-1} \mathbf{L}^T (\mathbf{S} - \mathbf{S}^0) \quad (7)$$

を計算すれば良い。ここで、各タイムステップにおいて $(\mathbf{L}^T \mathbf{L})^{-1} \mathbf{L}^T$ は不变であるので、シミュレーションの前に一度だけ計算すれば良い。次に、 $w_0 = 1 - \sum_{k=1}^n w_k$ を計算する。

このような操作だけでは負の重みも許してしまっているため、例示された変形の外挿を行っていることになる。これは、アーティストが変形の様子をデザイ

ンするという目的を考えると望ましくないことである。そこで負の重みを除去するために「最も小さい負の重みを選び、それを 0 にし、そのときの絶対値を n で割った値を残りの全ての重みから引く」という操作を、負の重みがなくなるまで繰り返し行う。

別の問題としては、弾性体モデルが例示集合 E のある要素が表す変形形状になったまま静止し、基本形状に戻らない可能性があるという問題が考えられる。これは、本提案手法では各局所領域の目標の形状そのものを変形させる（図3）ため、その時刻での形状と目標の形状が一致してしまうことがあり得るためである。このような問題を避けるため、上記のような方法で求めた重みに対し、新たに導入するパラメタ $\beta \in [0, 1]$ を用いて $w'_0 = w_0 + (1 - \beta) \sum_{k=1}^n w_k, w'_k = \beta w_k (k = 1, \dots, n)$ によって得られる重みを用いる。ここで β は 1 よりも僅かに小さい ($\beta = 0.995$ など) だけでも十分に効果があり、 β が小さくなるほど例示による効果は小さくなる。図5は図1 ((a), (b)) と同様の弾性体モデルを重力下で床に静止させたときに、 β の値に応じてどのように例示の効果が変わるかを示したものである。

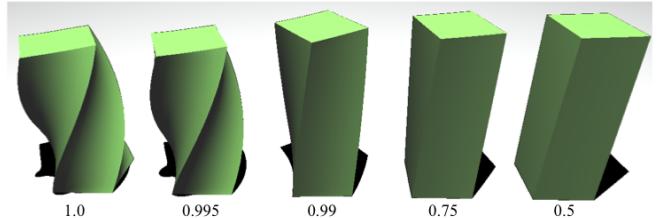


図5: β の値に伴って変化する例示の効果

4.4. 例示による効果の局所化に関する拡張

Martin らの手法と同様、本提案手法においても例示による効果を局所化させる拡張を行うことが可能である。そのためには、局所領域全体を複数の独立したグループに分け、それぞれのグループで例示集合を考え、投影を行えば良い。図6は弾性体モデルを縦に 2 つのグループに分けた例であり、図中の丸はユーザによる制御点を表している。（ここでは重力ではなく、弾性体モデルは腹部に当たる部分で空間に固定されている。）

また、各局所領域をそれぞれ独立なグループとする（このとき、それぞれのグループにおける例示集合は 6 次元空間上に定義される）ことによって、各局所領域における変形が局所的なふるまいによってのみ決まるという点で、物理的により自然な変形が表現できる。図7は図1 ((a), (b)) と同様の弾性体モデルを重力下で

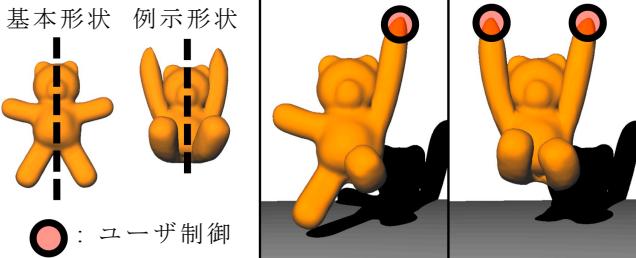


図6: 2つの独立したグループに分けられたテディ



図7: 全体で1つのグループとした場合(左)と各局所領域をそれぞれ独立なグループとした場合(右)

床に落下させた際の変形を表しており、左は全体で1つのグループ、右は各局所領域をそれぞれ独立なグループとした場合を表しており、前者は全体で一様に例示の効果が表れているのに対し、後者は部位によって例示の効果の表れ方が異なっている。

5. 結果

図1、図6-9は本提案手法を用いた例示ベースの弾性体のアニメーションの様子であり、いずれの例も実時間で計算している。表1はそれぞれのシミュレーションの各タイムステップにおける計算時間(単位はms)を表しており、 t_{sm} は従来のShape Matching法の計算にかかった時間、 t_{proj} は例示集合 E への投影にかかった時間、 t_{tot} はタイムステップ全体の計算にかかった時間を表している。なお、レンダリングに要した時間は含んでいない。計測はIntel Core i7 M620 2.66 GHz CPUで行った。表1から分かる通り、従来のShape Matching法に比べ、本提案手法を用いた場合の計算時間の増加は僅かである。また、Martinらの手法では類似の条件で計算するのに1タイムステップあたり数秒程度かかるため、本提案手法はこれに比べ数百倍程度高速であると言える。更に、生成されたアニメーションから、本提案手法によってMartinらの手法とほとんど同様の例示の効果が実現できていると考えられる。

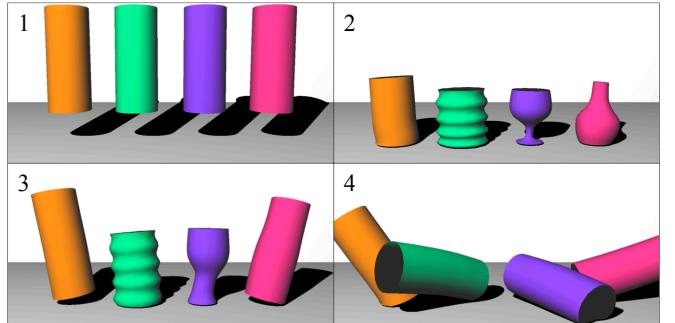


図8: 円柱型の四面体メッシュに円筒型の三角形メッシュを埋め込んだ例示ベースの弾性体のアニメーションの様子

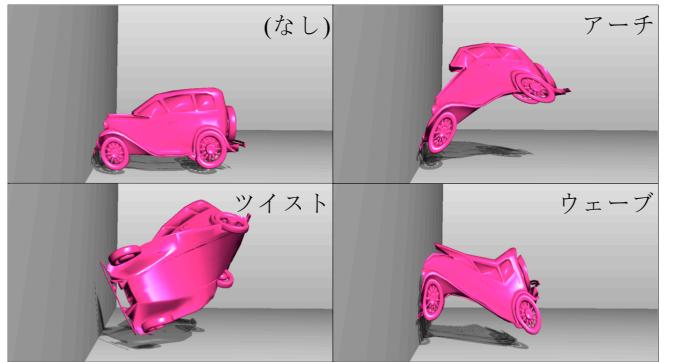


図9: 車型の弾性体モデルにそれぞれ異なる例示形状を与え、同じ初速で壁に衝突させた様子

モデル	頂点数	t_{sm}	t_{proj}	t_{tot}
四角柱	1225	1.5	0.17	2.2
円筒	2025	6.7	0.89	8.6
テディ	1280	4.6	0.57	5.5
車	1197	1.4	0.14	1.8

表1: パフォーマンス測定結果

6.まとめ

本稿では例示ベースの弾性変形を実時間で計算することができる高速な手法を提案した。既存手法が有限要素法に基づいたもので、非線形最適化が必要であったのに対し、本提案手法ではShape Matching法による弾性体の表現を拡張することで、非線形最適化を用いることなく例示ベースの弾性体を実現している。

本提案手法では三次元的な構造を持つ弾性体を扱ったが、Shape Matching法の拡張であるChain Shape Matching [10]やOriented Particles [7]などを組み合わせることで、布などの二次元的な構造や、髪の毛などの一次元的な構造に対しても本提案手法を応用していくことが将来課題として考えられる。また、その時刻

における変形形状だけでなく、速度や外力、周囲のオブジェクトとの相対位置などの様々な情報を考慮して例示集合に投影することで、興味深い効果が得られる可能性があると考えている。

なお、今回使わせて頂いた車の形状データは Princeton Shape Benchmark に含まれるものである。

文 献

- [1] Bergou, M., Mathur, S., Wardetzky, M., and Grinspun, E. 2007. Tracks: toward directable thin shells. *ACM Trans. Graph.* 26 (July).
- [2] Bonet, J., and Wood, R. D. 2008. *Nonlinear Continuum Mechanics for Finite Element Analysis*, 2 ed. Cambridge University Press.
- [3] Gerszewski, D., Bhattacharya, H., and Bargteil, A. W. 2009. A point-based method for animating elastoplastic solids. In *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, ACM, New York, NY, USA, SCA '09, 133–138.
- [4] Ijiri, T., Takayama, K., Yokota, H., and Igarashi, T. 2009. Procdef: Local-to-global deformation for skeleton-free character animation. *Computer Graphics Forum* 28, 7, 1821–1828.
- [5] Kondo, R., Kanai, T., and Anjyo, K. 2005. Directable animation of elastic objects. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, ACM, New York, NY, USA, SCA '05, 127–134.
- [6] Martin, S., Thomaszewski, B., Grinspun, E., and Gross, M. 2011. Example-based elastic materials. *ACM Trans. Graph.* 30 (Aug.), 72:1–72:8.
- [7] Müller, M., and Chentanez, N. 2011. Solid simulation with oriented particles. *ACM Trans. Graph.* 30 (Aug.), 92:1–92:10.
- [8] Müller, M., Heidelberger, B., Teschner, M., and Gross, M. 2005. Meshless deformations based on shape matching. *ACM Trans. Graph.* 24 (July), 471–478.
- [9] Rivers, A. R., and James, D. L. 2007. Fastlsm: fast lattice shape matching for robust real-time deformation. *ACM Trans. Graph.* 26 (July).
- [10] Rungjirathanon, W., Kanamori, Y., and Nishita, T. 2010. Chain shape matching for simulating complex hairstyles. *Computer Graphics Forum* 29, 8, 2438–2446.
- [11] Steinemann, D., Otraduy, M. A., and Gross, M. 2008. Fast adaptive shape matching deformations. In *Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, SCA '08, 87–94.
- [12] Thürey, N., Keiser, R., Pauly, M., and Rüde, U. 2006. Detail-preserving fluid control. In *Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation*, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, SCA '06, 7–12.
- [13] Twigg, C. D., and James, D. L. 2008. Backward steps in rigid body simulation. *ACM Trans. Graph.* 27 (August), 25:1–25:10.

付録：変形勾配テンソルの近似としての線形変換行列

ここでは式(1)によって計算される線形変換行列 \mathbf{A}_r が局所領域 r における変形勾配テンソルの近似になっていることを示す。以下、添字の r は適宜省略する。

Müller ら[8]の定義によると、式(1)によって計算される線形変換行列 \mathbf{A} は、二次のエネルギー

$$\sum_i \tilde{m}_i \|\mathbf{A}\mathbf{q}_i - \mathbf{p}_i\|^2 \quad (8)$$

を最小化するものである。ただし $\mathbf{q}_i = \mathbf{x}_i^0 - \mathbf{c}^0$, $\mathbf{p}_i = \mathbf{x}_i - \mathbf{c}$ は初期形状と変形形状における重心から見たパーティクル i の相対位置を表している。一方、連続体力学[2]では変形勾配テンソルは次のように定義される。まず物質中のある一点を考え、初期位置を \mathbf{X} 、現在位置を \mathbf{x} と表すこととする。次に、この物質点から無限に近い位置にある任意の点を考え、初期位置を \mathbf{X}' 、現在位置を \mathbf{x}' と表したとき、

$$\mathbf{x}' - \mathbf{x} = \mathbf{F}(\mathbf{X}' - \mathbf{X}) \quad (9)$$

という関係を満たすような \mathbf{F} を変形勾配テンソルと呼び、これはこの物質点の近傍の変形を表している。ここで、局所領域の重心点における変形勾配テンソル \mathbf{F} が、この局所領域に含まれるパーティクルの変形を近似していると考えると、 \mathbf{F} は二次のエネルギー

$$\sum_i w_i \|\mathbf{F}\mathbf{q}_i - \mathbf{p}_i\|^2 \quad (10)$$

を最小化させていると考えることが可能である。ここで w_i はパーティクル i に関する重みである。ここで、重み w_i として質量 \tilde{m}_i を用いると、式(10)は式(8)と一致する。したがって、線形変換行列 \mathbf{A} は変形勾配テンソル \mathbf{F} の近似になっていると考えることができる。

また Gerszewski ら[3]は点群ベースの弾塑性体のシミュレーションに関して上記と類似の議論を行っている。