

推薦論文

例示ベース弾性変形の実時間計算手法

小山 裕己^{1,a)} 高山 健志^{1,†1,b)} 梅谷 信行^{1,†2,c)} 五十嵐 健夫^{1,d)}

受付日 2013年4月6日, 採録日 2013年6月14日

概要: 本論文では, 実時間で計算可能な例示ベース弾性変形の手法を提案する. 例示ベース弾性変形とは Martin らによって提案された手法であり, 弾性体の変形形状を例示することによって直感的に変形の様子を制御することができるものである. しかしながら, Martin らの有限要素法に基づいたアプローチでは計算コストの高い非線形最適化計算を解く必要があり, ゲームなどの実時間アプリケーションでの使用は困難である. そこで我々は, 高速かつ頑健で, 実装が容易な手法として知られている Shape Matching 法に基づいて, 非線形最適化計算を用いずに例示ベース弾性変形を実現する手法を開発した. 我々の手法は Martin らの手法に比べ, 同等の例示ベース弾性変形の効果を表現でき, かつ数百から数千倍程度高速に計算可能である.

キーワード: 弾性体シミュレーション, Shape Matching 法, アーティスト制御

Real-Time Example-Based Elastic Deformation

YUKI KOYAMA^{1,a)} KENSHI TAKAYAMA^{1,†1,b)} NOBUYUKI UMETANI^{1,†2,c)} TAKEO IGARASHI^{1,d)}

Received: April 6, 2013, Accepted: June 14, 2013

Abstract: We present an example-based elastic deformation method that runs in real time. Example-based elastic deformation was originally presented by Martin et al., where an artist can intuitively control elastic material behaviors by simply giving example deformed poses. Their FEM-based approach is, however, computationally expensive requiring nonlinear optimization, which hinders its use in real-time applications such as games. Our contribution is to formulate an analogous concept using the shape matching framework, which is fast, robust, and easy to implement, without using nonlinear optimization. Our approach is two or three orders of magnitude faster than Martin et al.'s approach while producing comparable example-based elastic deformations.

Keywords: elastic simulation, shape matching dynamics, artist control

1. はじめに

リアリスティックなコンピュータ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, アーティストの演出意図に沿ってシミュレ-

ーション結果を制御できることが重要となる. そのようなアーティスト制御を目的とした物理シミュレーション手法としては, 流体 [16], 剛体 [17], 布 [1], 弾性体 [6] などを対象とした様々な手法が提案されてきた.

そのような中, Martin ら [7] は例示ベース弾性体という概念を提案した. これは弾性体の変形アニメーションの製作を支援するもので, 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. 例示ベース弾性

¹ 東京大学
The University of Tokyo, Bunkyo, Tokyo 113-0033, Japan

^{†1} 現在, ETH Zürich
Presently with ETH Zürich

^{†2} 現在, Autodesk Research
Presently with Autodesk Research

a) koyama@is.s.u-tokyo.ac.jp

b) kenshi84@gmail.com

c) n.umetani@gmail.com

d) takeo@acm.org

本論文の内容は 2012 年 6 月の Visual Computing/グラフィクスと CAD 合同シンポジウムで報告され, グラフィクスと CAD 研究会主査により情報処理学会論文誌ジャーナルへの掲載が推薦された論文である.

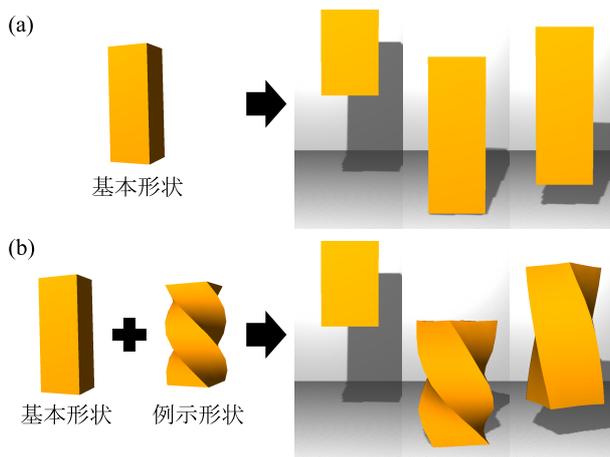


図 1 (a) 通常の弾性体オブジェクト（ここでは四角柱）の床との衝突のアニメーションの例. (b) 例示ベース弾性体のためのユーザによる入力形状と、それによって生成された例示ベース弾性体オブジェクトの床との衝突のアニメーションの例

Fig. 1 (a) An example of an animation where a conventional elastic cuboid bounces off the floor. (b) An example of user-specified input data for example-based elastic materials and a result animation of the example-based elastic cuboid generated from the input data.

体の例を図 1 に示す。この例は四角柱の形状を持つ弾性体に対し、ねじれた形状を例示形状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。例示ベース弾性体を用いることで、このようなアニメーションを製作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、変形の性質を直接、直感的に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 [6] 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外力を受けて変形するかなどといった情報は、弾性体の変形の性質をデザインする段階では必要ない。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も使用することができたりする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、例示の効果を実現するための特別な保存力を計算する。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いる。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いる。

このような制限を克服するため、本論文では特に実時間でも計算できる高速な例示ベース弾性変形の計算手法を提案する。本提案手法は Shape Matching 法 [9] による弾性

体の表現に基づいており、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動的に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、数百から数千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の保証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

2.1 Shape Matching 法

Shape Matching 法は Müller ら [9] によって提案された幾何学的方法に基づく柔軟物体の表現手法である。過重を取り除くと基本形状に戻るといった弾性体の性質をよくモデル化できている手法であり、有限要素法などに基づく他の弾性体シミュレーション手法に比べて高速で頑健であるといった特徴があるため、特にインタラクティブな用途での活躍が期待されている弾性体シミュレーション手法である。この手法は Müller らによって初めて提案されて以来、様々な拡張が提案されてきた。FastLSM [10] 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 [14] はこれに対し八分木のデータ構造を導入することで、さらに効率的に計算する手法を提案した。Chain Shape Matching 法 [11], [12] は鎖状に並んだパーティクル群に対して Shape Matching 法を適用する手法で、髪の毛や紐状の弾性体を表現することができる手法である。Oriented Particles [8] は各パーティクルに対して位置だけでなく向き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り、鎖状や平面状に配置されたパーティクル群に対しても適用することができる手法となっている。また、体積保存を考慮して変形させる拡張 [3], [18] や、粘弾性体を表現するための拡張 [15] も存在する。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、例示ベース弾性変形を実現するものは本提案手法が最初である。

また ProcDef [5] や実時間心臓拍動シミュレータ [4] は Shape Matching 法を拡張することで柔軟物体の能動的な動作を生成する手法であり、これら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。本提案手法でも各局所領域の目標の形状を変

えるというアプローチをとっているが、目標の形状がそのときの変形の状態に応じて受動的に変わっていくという点で大きく異なっている。

2.2 例示ベース弾性体

例示ベース弾性体はアーティストフレンドリな弾性体シミュレーション手法として Martin ら [7] が提案したものであるが、先に述べたとおり、Martin らの手法は計算コストが非常に高いという問題点がある。我々の手法は、Martin らの手法と同様の効果を実時間で計算するというものである。また、Schumacher ら [13] も例示ベース弾性体を高速に計算するための手法を提案しているが、実時間で計算するには至っていない。Schumacher らの手法と我々の手法との間の共通点や相違点については 6.2 節で説明する。

3. 実行時の計算の概要

本提案手法の詳細を記述する前に、まず本提案手法の実行時の計算の概要を示す。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、例示ベース弾性変形を実現する。

- (1) 現在の形状を例示集合 \mathcal{E} (4.2 節) に投影 (4.3 節) する。
- (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 節)。

以上の操作を模式的に表したのが図 2 である。この例では、直方体の1つの基本形状に対し、2つの例示形状を与えた場合を表しており、それぞれの形状は高次元ベクトルとして記述 (4.2 節) されている。Martin らの手法との最も大きな違いは、Martin らの手法ではこれらの高次元ベクトルの非線形な補間として例示集合 \mathcal{E} を定義していたのに対し、我々の手法では単純な線形補間によって例示集合 \mathcal{E} を表現している点である。これによって、我々の手法では投影の操作を行う際に非線形最適化を用いることなく、効率的に計算を行うことができる。

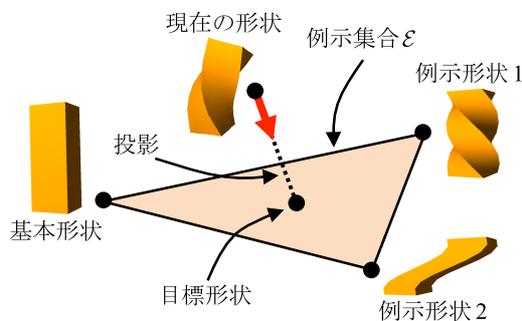


図 2 実行時の計算の概要。ここでは直方体の基本形状に対して2つの例示形状を与えた場合を表しており、現在の形状はその時点における変形形状を表している

Fig. 2 Overview of run-time operations. Here, the elastic cuboid model has its rest pose, current deformed pose, and two example poses.

4. 提案手法の詳細

4.1 Shape Matching 法の拡張

本提案手法は、Müller ら [9] 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、本提案手法では互いにオーバーラップした局所領域を持つ構造の Shape Matching 法 (これは [5], [10] などでも使用されている。) を扱う。この方法は、局所領域ごとに 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方法である。

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ったまま、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三角形メッシュを埋め込むことも可能であり、本研究では結果アニメーションの生成の際にシミュレーション用に粗い四面体メッシュ、レンダリング用に詳細な三角形メッシュを用いることで、計算コストを抑えつつ見た目の質を上げる工夫を行った。

四面体メッシュの頂点はそれぞれ Shape Matching 法における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 \mathcal{N}_i が定義される。局所領域 \mathcal{N}_i の要素は、パーティクル i とその1近傍頂点 (1-ring neighborhood [5]) とする。なお、本研究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、文献 [10] のように格子状の構造によって局所領域を定義する場合にも、本提案手法はそのまま適用可能である。

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点における形状への変換を近似した線形変換行列

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

を計算する。ここで、 \tilde{m}_i は有効質量 [10]、 $\mathbf{c}_r^0 = \frac{\sum_{i \in r} \tilde{m}_i \mathbf{x}_i^0}{\sum_{i \in r} \tilde{m}_i}$ 、 $\mathbf{c}_r = \frac{\sum_{i \in r} \tilde{m}_i \mathbf{x}_i}{\sum_{i \in r} \tilde{m}_i}$ はそれぞれ初期状態とその時点での局所領域 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 $\mathbf{R}_r = \mathbf{A}_r \mathbf{S}_r^{-1}$ 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

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

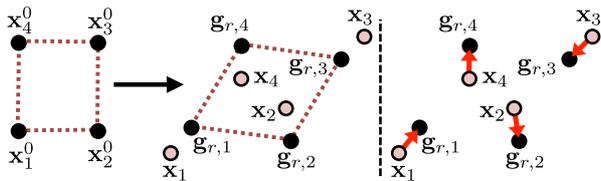


図 3 本提案手法における目標位置の計算方法. パーティクル i の局所領域 r における目標位置 $\mathbf{g}_{r,i}$ を計算するために, 従来の Shape Matching 法では基本形状を変形させずに目標形状としていたのに対し, 本提案手法では投影 (4.3 節) によって得られる $\tilde{\mathbf{S}}_r$ を用いて適切に基本形状を変形させたものを目標形状として用いている

Fig. 3 Our method for calculating the goal positions. In order to obtain the goal position $\mathbf{g}_{r,i}$ of particle i in local region r , the rest shape of the region is appropriately deformed by the strain $\tilde{\mathbf{S}}_r$ that is obtained by projection (Section 4.3) in our method, while the standard shape matching uses the undeformed rest shape as the goal shape.

によって計算する (図 3). ここで, $\tilde{\mathbf{S}}_r \in \mathbb{R}^{3 \times 3}$ は目標の形状を歪ませる要素であり, 4.2 節で説明する例示集合 \mathcal{E} に対して, 4.3 節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 が属するすべての局所領域での目標位置の平均 $\mathbf{g}_i = \langle \mathbf{g}_{r,i} \rangle_{r \ni i}$ 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.

4.2 変形記述子と例示集合

4.2.1 Martin らの手法

Martin らの手法は, 例示の効果を実現するために好ましい変形形状の集合 (例示集合) のうち, その時点における変形形状に最も近い変形形状を探し, その変形形状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 として, $\mathbf{E}(\mathbf{x}) = (\mathbf{E}_1^T, \mathbf{E}_2^T, \dots, \mathbf{E}_{M'}^T)^T \in \mathbb{R}^{6M'}$ を変形 \mathbf{x} に対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, その自由度は 6 であるので, 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 $\mathbf{x} \rightarrow \mathbf{E}(\mathbf{x})$ の値域 $\mathcal{F} \subset \mathbb{R}^{6M'}$ (実現可能集合と呼ぶ) は非線形な集合である. すなわち,

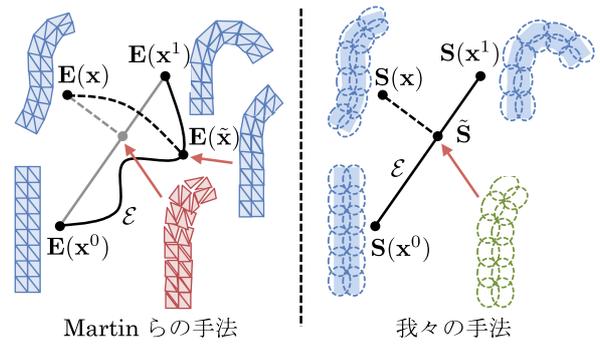


図 4 Martin らの手法と我々の手法の比較. Martin らの手法では例示集合 \mathcal{E} と投影が非線形だったのに対し, 我々の手法では, 例示集合 \mathcal{E} は単純な線形補間によって定義されており, したがって投影を簡単に扱うことができる

Fig. 4 Comparison between Martin et al.'s and our approach. In Martin et al.'s approach, example manifold \mathcal{E} and the projection are nonlinearly defined. In contrast, in our approach, example manifold \mathcal{E} can be defined as a simple linear interpolation, thus the projection can be handled easily.

線形結合 $w_1 \mathbf{E}(\mathbf{x}^1) + w_2 \mathbf{E}(\mathbf{x}^2)$ は一般には \mathcal{F} の要素ではなく, これに対応する有効な変形形状は存在しない. そこで彼らは, $\{\sum_{k=0}^N w_k \mathbf{E}(\mathbf{x}^k)\}$ (ただし \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 \mathcal{F} に投影して得られる集合 \mathcal{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 \mathcal{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).

4.2.2 我々の手法

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 A.1 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を用いて, 変形記述子を $\mathbf{S}(\mathbf{x}) = (\mathbf{S}_1^T, \mathbf{S}_2^T, \dots, \mathbf{S}_M^T)^T \in \mathbb{R}^{6M}$ と定義する. なお, ここでは 3×3 の対称行列である \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 法では局所領域ごと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という点で, 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 $\{\sum_{k=0}^N w_k \mathbf{S}(x^k)\}$ によって得られる凸包を例示集合 \mathcal{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 \mathcal{E} に対して, その時点における変形形状を線形投影 (詳しくは 4.3 節で説明する) することによって, 目標となる変形記述子 $\tilde{\mathbf{S}}$, さらに局所領域 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 $\mathbf{S}^0 = \mathbf{S}(x^0), \mathbf{S}^k = \mathbf{S}(x^k)$ と表すと, 例示集合 \mathcal{E} に対してその時点における変形形状 $\mathbf{S} = \mathbf{S}(x)$ を投影するとは, \mathbf{S} に最も近い例示集合 \mathcal{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, w_2, \dots, w_N を計算する. 具体的には, $\mathbf{w} = (w_1, w_2, \dots, w_N)^T \in \mathbb{R}^N, \mathbf{L} = (\mathbf{S}^1 - \mathbf{S}^0, \mathbf{S}^2 - \mathbf{S}^0, \dots, \mathbf{S}^N - \mathbf{S}^0) \in \mathbb{R}^{6M \times N}$ として

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

を計算すればよい. ここで, 各タイムステップにおいて $(\mathbf{L}^T \mathbf{L})^{-1} \mathbf{L}^T$ は不変であるので, シミュレーションの前に一度だけ計算すれば良い. 続いて, $w_0 = 1 - \sum_{k=1}^N w_k$ 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, 例示された変形形状の外挿を計算してしまう可能性がある. 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すべての重みから引く」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, 弾性体が例示集合 \mathcal{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, その時点での変形形状と目標の形状が完全に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 β を用いて $w'_0 = w_0 + (1 - \beta) \sum_{k=1}^N w_k, w'_k = \beta w_k (k = 1, 2, \dots, N)$ によって得られる重みを用いる. ここで β は 1 よりもわずかに小さい ($\beta = 0.995$ 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 と同様の弾性体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.

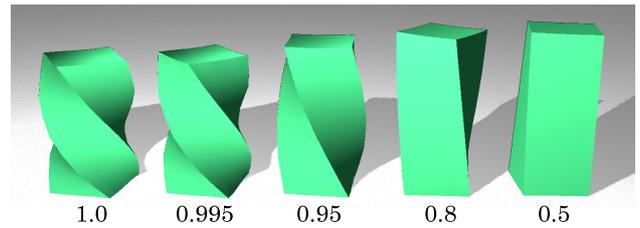


図 5 β の値にともなって変化する例示ベース弾性変形の効果
Fig. 5 Varying the magnitude of the example-based elastic deformation effect by changing β .

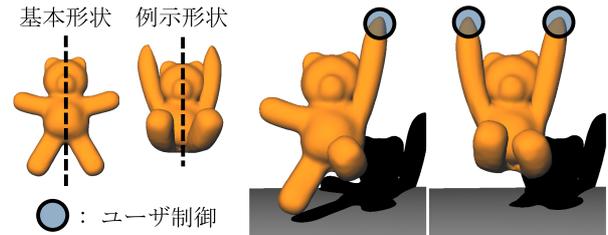


図 6 2つの独立したグループに分けられたテディ
Fig. 6 A teddy bear whose local regions are divided into two separate groups at its middle.



図 7 全体で 1つの例示集合を形成した場合 (左) と, 各局所領域でそれぞれ独立な例示集合を形成した場合 (右) 比較

Fig. 7 Comparison between the case of using globally defined example manifold (left) and the case where each local region has its own example manifold independently (right).

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 (ここでは重力はなく, 弾性体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振舞いによってのみ決まるという点で, 物理的により自然な, 異方性弾性体のような変形が表現できる. 図 7 は図 1 と同様の弾性体を重力下で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

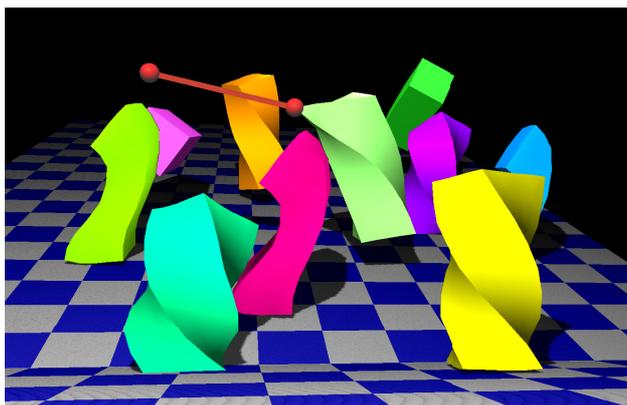


図 8 2種類の例示ベース弾性体（直方体に対してねじれた形状とS字形の例示形状をそれぞれ与えたもの）を用いて生成した坂道のシーンにおけるアニメーションの様子。シーン全体で2,250個のパーティクルを含んでおり、49FPSで実行されている

Fig. 8 An animation of a slope scene containing two types of example-based elastic cuboids (generated from twist-shaped and S-shaped example poses) represented by 2,250 particles in total; this scene is simulated at 49 frames per second.

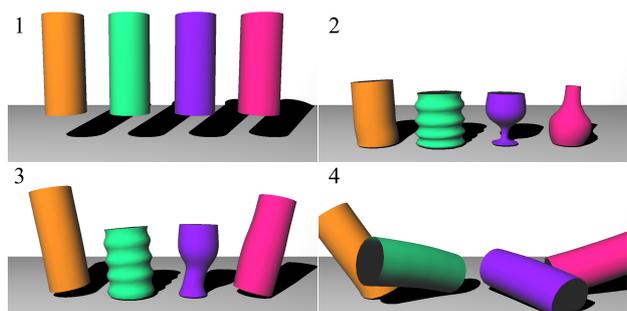


図 9 円柱型の四面体メッシュに円筒型の三角形メッシュを埋め込んだ例示ベース弾性体のアニメーションの様子

Fig. 9 Four cylinders undergoing very different elastic deformations.

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 8, 図 9, 図 10 に、本提案手法を用いて生成した例示ベース弾性体のアニメーションの例を示す。これらは、いずれ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1タイムステップあたりの計算時間（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、ここでは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 を搭載したデスクトップ PC で行った。この表から分かりますと、従来の Shape Matching 法に比べ、本提案手法を用

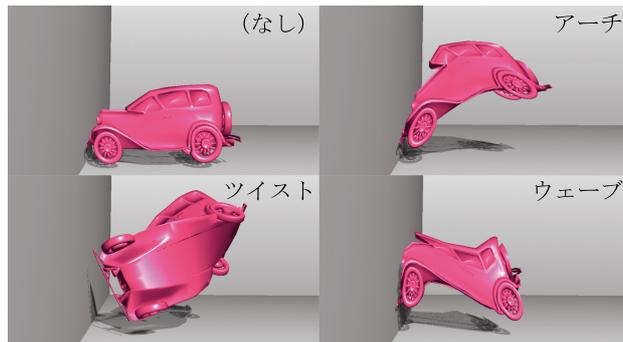


図 10 壁に激突する車のアニメーションの様子。1つは例示形状を与えた場合、残りの3つはそれぞれ異なる例示形状を与えた場合である

Fig. 10 Cars colliding against the wall. One (upper left) is simulated without any example pose, and the others are simulated with different kinds of example poses.

表 1 1タイムステップあたりの計算時間（単位はms）
Table 1 Timings per time step in milliseconds.

モデル	#パーティクル	t_{sm}	t_{proj}	t_{tot}
直方体	225	0.21	0.091	0.33
直方体（局所化）	225	0.21	0.17	0.40
円筒	2,025	2.0	0.86	3.1
テディ	1,280	1.3	0.72	2.2
車	192	0.17	0.078	0.27

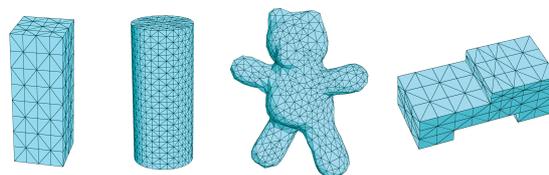


図 11 本論文でアニメーション生成のために用いた四面体メッシュ
Fig. 11 Tetrahedral meshes used in this paper for generating the animations.

いた場合の計算時間の増加はわずかである。また、図 11 に結果生成のために用いた四面体メッシュを示す。

6. 考察

6.1 Martin らの手法との結果の比較

図 12 に、ねじれた形状を例示として与えた直方体を重力下においた場合に関する、Martin らの手法と我々の手法の結果の簡単な比較を示す。ただし、ここではシミュレーションやレンダリングの条件は厳密には同じでなく、また有限要素法と Shape Matching 法ではシミュレーションの枠組みが異なるために例示ベース弾性変形の効果だけを厳密に比較することが本質的に困難であるという点に注意されたい。しかしながら、この比較画像からも観察できるように、我々の手法は Martin らの手法と実質的に同様な効果を生み出していると我々は考えている。また、計算コストに関しても厳密な比較が難しいものの、Martin らの手法

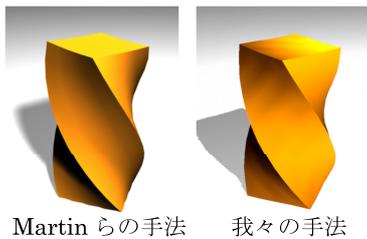


図 12 Martin らの手法と我々の手法の結果の比較. 左の画像は文献 [7] のものである (©2011 Association for Computing Machinery, Inc. Reprinted by permission.)

Fig. 12 Comparison between Martin et al.'s result and ours. The left image is from Ref. [7], ©2011 Association for Computing Machinery, Inc. Reprinted by permission.

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のに対し、我々の手法ではこれに比べ数百から数千倍程度高速に計算できているといえる。

6.2 Schumacher らの手法との共通点と相違点

例示ベース弾性変形を高速に計算するための手法として、我々の手法の他に、Schumacher ら [13] の手法が知られている。Schumacher らの手法と我々の手法の共通点としては、例示集合 \mathcal{E} の定義として、入力形状の変形記述子の単純な線形補間を用いているという点があげられる。これによって、投影によって得られる変形記述子は有効な形状を表さなくなるが、我々の手法と同様、Schumacher らの手法でもそのような有効でない形状の変形記述子をそのまま用いて定式化することで、計算コストの削減を行っている。我々の手法との相違点としては、我々の手法では Shape Matching 法と組み合わせて定式化したために非線形最適化計算を行う必要がなくなったが、Schumacher らの手法は Martin らの手法と同様に有限要素法に基づいているため、依然として非線形最適化計算を行う必要がある点があげられる。Martin らの手法に比べ、Schumacher らの手法は非線形最適化計算の際の変数の数が少なく、計算コストは小さくなっているものの、数十倍程度の高速化にとどまっており、実時間アプリケーションでの使用には不向きである。

6.3 物理的な正確さ

アーティストによる映像表現という文脈においては物理的な正確さの重要度は低いと考えられるものの、例示ベース弾性変形による効果以外については、より物理的に正確であることが要求される状況も考えられる。Martin らの手法が連続体力学に基づいた有限要素法を用いたアプローチであるのに対し、本提案手法は幾何学に基づいた Shape Matching 法を用いたアプローチであり、また Shape Matching 法による弾性体表現の物理的な正確さの評価はまだ十分に研究されていないため、本提案手法は物理的な

正確さの保証ができない。この点は本提案手法の欠点であるといえる。

6.4 入力形状について

本提案手法では、入力形状の補間の際に四面体要素間の衝突などを考慮していないため、例示形状によっては途中の補間形状で自己干渉を起こしてしまう可能性がある。このような問題は、Martin らの手法や Schumacher らの手法でも発生するものと考えられる。

また本提案手法では、Martin らの手法や Schumacher らの手法と同様、基本形状と例示形状はすべて同じメッシュ構造を持っている必要がある。このような制限はアーティストにとって大きな制約となってしまうと考えられるため、これを解決することは重要である。これには、メッシュ構造に依存しない変形記述子や例示集合を用いる必要があると考えられる。

入力形状の四面体メッシュについては、本提案手法においては各四面体要素の大きさが均一であることが望ましいと考えられる。その理由は、本提案手法では四面体要素の大きさに依存して局所領域の大きさが決定し、また互いにオーバーラップした局所領域を持つ構造の Shape Matching 法では局所領域の大きさがその局所領域の硬さに影響するため、各四面体要素の大きさが不均一だとオブジェクトの硬さも部位によって不均一になってしまうからである。

7. まとめ

本論文では例示ベース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弾性体を実現している。

本提案手法では 3 次元の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 [11], [12] や Oriented Particles [8] などの手法と組み合わせることで、布などの 2 次元の構造や、髪の毛などの 1 次元の構造に対しても本提案手法を応用していくことが将来研究として考えられる。また、Schumacher ら [13] が Explicit Weight Control としてその応用例を提示しているように、その時点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様々な情報を考慮して例示集合に投影する仕組みを考えることで、興味深い効果が得られることが期待できると考えている。

謝辞 本研究で使わせていただいた車の形状データは Princeton Shape Benchmark に含まれるものである。

参考文献

[1] Bergou, M., Mathur, S., Wardetzky, M. and Grinspun, E.: TRACKS: Toward directable thin shells, *ACM Trans. Graph.*, Vol.26 (2007).

[2] Bonet, J. and Wood, R.D.: *Nonlinear Continuum Mechanics for Finite Element Analysis*, Cambridge University Press, 2 edition (2008).

[3] Diziol, R., Bender, J. and Bayer, D.: Robust real-time deformation of incompressible surface meshes, *Proc. 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp.237–246, ACM (2011).

[4] Ijiri, T., Ashihara, T., Umetani, N., Igarashi, T., Haraguchi, R., Yokota, H. and Nakazawa, K.: A Kinematic Approach for Efficient and Robust Simulation of the Cardiac Beating Motion, *PLoS ONE*, Vol.7, No.5, p.e36706 (2012).

[5] Ijiri, T., Takayama, K., Yokota, H. and Igarashi, T.: ProcDef: Local-to-global Deformation for Skeleton-free Character Animation, *Computer Graphics Forum*, Vol.28, No.7, pp.1821–1828 (2009).

[6] Kondo, R., Kanai, T. and Anjyo, K.-I.: Directable animation of elastic objects, *Proc. 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp.127–134, ACM (2005).

[7] Martin, S., Thomaszewski, B., Grinspun, E. and Gross, M.: Example-based elastic materials, *ACM Trans. Graph.*, Vol.30, pp.72:1–72:8 (2011).

[8] Müller, M. and Chentanez, N.: Solid simulation with oriented particles, *ACM Trans. Graph.*, Vol.30, pp.92:1–92:10 (2011).

[9] Müller, M., Heidelberger, B., Teschner, M. and Gross, M.: Meshless deformations based on shape matching, *ACM Trans. Graph.*, Vol.24, pp.471–478 (2005).

[10] Rivers, A.R. and James, D.L.: FastLSM: Fast lattice shape matching for robust real-time deformation, *ACM Trans. Graph.*, Vol.26 (2007).

[11] Rungjiratananon, W., Kanamori, Y. and Nishita, T.: Chain Shape Matching for Simulating Complex Hairstyles, *Computer Graphics Forum*, Vol.29, No.8, pp.2438–2446 (2010).

[12] Rungjiratananon, W., Kanamori, Y., Metaaphanon, N., Bando, Y., Chen, B.-Y. and Nishita, T.: Twisting, tearing and flicking effects in string animations, *Proc. 4th International Conference on Motion in Games*, pp.192–203, Springer-Verlag (2011).

[13] Schumacher, C., Thomaszewski, B., Coros, S., Martin, S., Sumner, R. and Gross, M.: Efficient simulation of example-based materials, *Proc. ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp.1–8, Eurographics Association (2012).

[14] Steinemann, D., Otaduy, M.A. and Gross, M.: Fast adaptive shape matching deformations, *Proc. 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp.87–94, Eurographics Association (2008).

[15] Takamatsu, K. and Kanai, T.: A fast and practical method for animating particle-based viscoelastic fluids, *The International Journal of Virtual Reality*, Vol.10, No.1, pp.29–35 (2011).

[16] Thürey, N., Keiser, R., Pauly, M. and Rüdte, U.: Detail-preserving fluid control, *Proc. 2006 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp.7–12, Eurographics Association (2006).

[17] Twigg, C.D. and James, D.L.: Backward steps in rigid body simulation, *ACM Trans. Graph.*, Vol.27, pp.25:1–25:10 (2008).

[18] 高松賢二, 金井 崇: シェイプマッチング法による体積保存を考慮した形状変形手法, 画像電子学会誌, Vol.40, No.4, pp.549–557 (2011).

付 録

A.1 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 の重心点における変形勾配テンソル \mathbf{F}_r を近似していると考えられることを示す。以下では1つの領域について考えるので, 添字の r は適宜省略する。

後の計算を容易にするため, この領域の重心点の座標が原点 \mathbf{O} であるとする, 領域内の任意の点 $\mathbf{x} = (x_1, x_2, x_3)^T$ における変形時の変位 $\mathbf{u}(\mathbf{x})$, すなわち変形による相対的な移動量は, 原点 \mathbf{O} におけるテイラー展開によって

$$u_i(\mathbf{x}) = \sum_{k+l+m=n}^{n=0, \dots, \infty} a_i^{k,l,m} x_1^k x_2^l x_3^m \quad (i = 1, 2, 3) \quad (\text{A.1})$$

と表すことができる。連続体力学 [2] では, 変位 $\mathbf{u}(\mathbf{x})$ のヤコビ行列を, 変位勾配テンソル

$$\mathbf{Z} = \frac{\partial}{\partial \mathbf{x}} \mathbf{u}(\mathbf{x}) = \begin{pmatrix} a_1^{1,0,0} & a_1^{0,1,0} & a_1^{0,0,1} \\ a_2^{1,0,0} & a_2^{0,1,0} & a_2^{0,0,1} \\ a_3^{1,0,0} & a_3^{0,1,0} & a_3^{0,0,1} \end{pmatrix} \quad (\text{A.2})$$

と定義している。この変位勾配テンソル \mathbf{Z} を用いて, 変形勾配テンソル \mathbf{F} は $\mathbf{F} = \mathbf{I} + \mathbf{Z}$ と計算される。

ここで, この領域が質量を 1 とする 6 つのパーティクル

$$\begin{aligned} \mathbf{x}_1 &= (-\epsilon, 0, 0)^T, & \mathbf{x}_2 &= (\epsilon, 0, 0)^T, \\ \mathbf{x}_3 &= (0, -\epsilon, 0)^T, & \mathbf{x}_4 &= (0, \epsilon, 0)^T, \\ \mathbf{x}_5 &= (0, 0, -\epsilon)^T, & \mathbf{x}_6 &= (0, 0, \epsilon)^T \end{aligned} \quad (\text{A.3})$$

によって構成されているとする。このとき, 式 (1) の $\mathbf{p}_i, \mathbf{q}_i$ は, \mathbf{c} を領域の変位後の重心位置として

$$\mathbf{p}_i = \mathbf{x}_i + \mathbf{u}(\mathbf{x}_i) - \mathbf{c}, \quad \mathbf{q}_i = \mathbf{x}_i \quad (\text{A.4})$$

と表され, この領域における線形変換行列 \mathbf{A} は式 (1) より

$$\mathbf{A} = \left(\sum_{i=1}^6 \mathbf{p}_i \mathbf{q}_i^T \right) \left(\sum_{i=1}^6 \mathbf{q}_i \mathbf{q}_i^T \right)^{-1} \quad (\text{A.5})$$

$$\begin{aligned} &= \mathbf{I} + \left(\sum_{i=1}^6 \mathbf{u}(\mathbf{x}_i) \mathbf{x}_i^T \right) \left(\sum_{i=1}^6 \mathbf{x}_i \mathbf{x}_i^T \right)^{-1} \\ &\quad - \left(\sum_{i=1}^6 \mathbf{c} \mathbf{x}_i^T \right) \left(\sum_{i=1}^6 \mathbf{x}_i \mathbf{x}_i^T \right)^{-1} \end{aligned} \quad (\text{A.6})$$

$$= \mathbf{I} + \mathbf{B} - \mathbf{C} \quad (\text{A.7})$$

によって計算できる。式 (A.1) と式 (A.3) を代入して \mathbf{B} を具体的に計算すると,

$$B_{ij} = \begin{cases} a_i^{1,0,0} + \sum_{k=1}^{\infty} a_i^{2k+1,0,0} \epsilon^{2k} & (j=1) \\ a_i^{0,1,0} + \sum_{k=1}^{\infty} a_i^{0,2k+1,0} \epsilon^{2k} & (j=2) \\ a_i^{0,0,1} + \sum_{k=1}^{\infty} a_i^{0,0,2k+1} \epsilon^{2k} & (j=3) \end{cases} \quad (\text{A.8})$$

となる。また

$$\sum_{i=1}^6 \mathbf{c} \mathbf{x}_i^T = \mathbf{c} \sum_{i=1}^6 \mathbf{x}_i^T = \mathbf{c} \mathbf{0}^T = \mathbf{0} \quad (\text{A.9})$$

より、 $\mathbf{C} = \mathbf{0}$ となる。ここで $\epsilon \rightarrow 0$ とすると $\mathbf{B} - \mathbf{C} \rightarrow \mathbf{Z}$ となり、したがってこのとき $\mathbf{A} \rightarrow \mathbf{F}$ となることが分かる。以上より、線形変換行列 \mathbf{A} を変形勾配テンソル \mathbf{F} の近似と考えることができることが確認できた。また、このことは一般的なパーティクルの配置においても同様の手順で容易に示すことができる。

推薦文

本論文では3次元弾性体モデルの高速変形手法を提案している。もっともらしい変形の動きと制作者の演出意図を両立させる例示ベースの手法の1種であるが、非線形最適化計算を避けて線形補間をうまく使うことなどにより、従来手法に比べて数百倍の高速化を達成した。リアルタイムで最終的な作品の品質での変形アニメーション実行が可能となったことにより、CG映像制作プロセスに大きな変革を与えるだけでなく、その場で様々な変形を行うゲームなども実現できる。さらに、本論文のアイデアは、布のような2次元モデルや髪の毛のような1次元モデルにも適用できる可能性があり、将来の研究の拡がりも期待できる。よって、論文誌での掲載価値を十分に有していると判断し、本研究会は本論文を推薦論文として推薦いたします。

(グラフィクスとCAD研究会主査 柿本正憲)



梅谷 信行

1983年生。2012年東京大学大学院情報理工学系研究科コンピュータ科学専攻博士課程修了。現在、Autodesk Researchにて、シミュレーションとユーザインタフェース研究開発に従事。



五十嵐 健夫 (正会員)

2000年東京大学大学院工学系研究科博士(工学)。2002年同大学院情報理工学系研究科講師、2005年助教授、2011年教授。2007~2013年JST ERATO研究総括。学術振興会賞、SIGGRAPH若手科学者賞等受賞。専門はユーザインタフェースおよびグラフィクス。



小山 裕己 (学生会員)

1989年生。2012年東京大学理学部情報科学科卒業。同年同大学院修士課程進学。コンピュータグラフィクスの研究に従事。



高山 健志

1984年生。2012年東京大学大学院情報理工学系研究科コンピュータ科学専攻博士課程修了。現在、ETH Zurichにて日本学術振興会海外特別研究員としてコンピュータグラフィクスの研究に従事。