

物理指向補間による弾性体アニメーションの制御手法*

近藤 亮[†] 金井 崇[‡] 安生 健一[§]

rk@sfc.keio.ac.jp kanait@acm.org anjyo@olm.co.jp

[†] 慶應義塾大学 政策・メディア研究科

[‡] 理化学研究所

[§] オー・エル・エム デジタル

自然な変形物のアニメーションを得るためには、物理シミュレーションなどの手法が用いられる。しかしながら、ユーザの意図する動きをシミュレーションを用いて得るためには、複雑な初期パラメータの調整が必要であり、実際の映像制作に活かされることは少なかった。そこで本論文では、ユーザが形状を制御することのできる弾性体のアニメーション手法を提案する。キーフレームで指定された形状情報を用い、物理法則に基づいた物体の変形を制御することができる。また、これを実現するために有限要素法を用いた新たな形状補間手法を開発した。これらの制御手法により、自由度の高い自然なアニメーションを生成することが可能である。

1 はじめに

アニメーションコンテンツにおいて、最も重要なのは作り手の意図に沿った動きが行われることである。キーフレームという手法は、時間に対する動きを定義する際に非常に直感的な方法であり、アニメーション制作の現場では一般的な考え方となっている。しかしながら、例えば現実感のある自然な物体の変形アニメーションを手作業で一つ一つ作るには、作り手に高い技術と時間のコストが要求される。

一方でシミュレーション技術の分野では、計算機を用いることで物理的に正確な動きを模倣するための様々な研究がなされてきた。しかしながら得られる結果は複雑な物理パラメータの初期値に依存し、直感的にその運動の結果を制御することは不可能である。このため、物理的な自然さだけでなく、作り手の意図を満たすことも重要となるアニメーション制作において、シミュレーション技術が応用される機会は少なかった。

近年、粒子や剛体運動を例として、このような物理法則とユーザによる制御の両方の観点からの研究の必要性が高まっている [3, 17, 18, 19]。そこで本研究では、変形物の中でも最も一般的な、弾性物体を対象とした物理法則に基づくアニメーションの制御手法を提案する。

ユーザの意図に沿った物体の変形を実現するため、ユーザは従来のキーフレームの考え方と同様に、時間軸に沿って物体の形状を定義することができる。アニメーションが再計算されると、これらの形状は弾性体の形状変形を誘導する制約となり、ユーザの意図に沿いながらも、物理法則に基づいた自然なアニメーションが可能となる。このことを実現するため、有限要素法に基づく新たな形状補間手法を開発した。これらの制御手法を用いることで、実際にユーザが自由度の高い自然なアニメーションが生成できることを示す。

2 関連研究

物理法則のモデル化は、これまでに数多くの研究がなされてきた。剛体や柔軟物体のみならず、現在では煙や水などの流体、またそれらの異なる物質間の相互作用なども再現することができるようになった。

物理法則を制御するための研究では、関節物体や剛体、煙を対象としたものがその殆どを占めている。特に関節物体を運動学に基づき最適な経路で動かすことで自然な動きを自動生成する試みは数多くあり [19, 9, 6]、人間などの動きを再現するために欠かせない技術の一つとなっている [8]。

単一の剛体の軌道制御では、衝突時の軌跡の変化に微量のばらつきを加える手法 [3] や、連鎖モンテカルロ法を用いて多体アニメーションの制御に対応したもの [5] がある。このように剛体シミュレーションでは、初期条件や衝突時のパラメータを変化させることで物体の軌道を制御するのが一般的である。またユーザの指定する軌道を満たすようなパラメータをインタラクティブに計算することで、直感的に剛体運動の軌道を制御する手法も提案されている [17, 16]。

流体の分野では、特に煙のアニメーションの制御の手法が近年提案されている [18, 7, 13]。ユーザの意図する密度分布などから誤差評価関数を定め、その値を最小化するように煙の流れを促進または抑制するのが一般的な方法となっている。

一方で、物理法則を用いて物体の相互作用により発生する受動的・副次的な動きを効率よく計算する研究も行われており、計算コストの膨大な増加を抑えつつも、異なる物体間の相互干渉を安定して再現する手法なども提案されてきた [2, 15]。

本研究が対象とする柔軟物体に関する研究では、高速化 [14] や安定性の向上 [14, 11] についての手法が多く提案されてきた。また弾性体同士よりも正確な衝突反応の計算 [10] や、ボーン主導での変形モデル [4] など、多くの状況に対応する計算モデルが考案されてきた。しかしながら、これらの手法では運動の結果は初期値のみに依存してお

* *Elastic body animation using physics-oriented interpolation*, by Ryo KONDO (Keio University, Graduate School of Media and Governance) and Takashi KANAI (RIKEN) and Ken-ichi ANJYO (OLM Digital, Inc.)

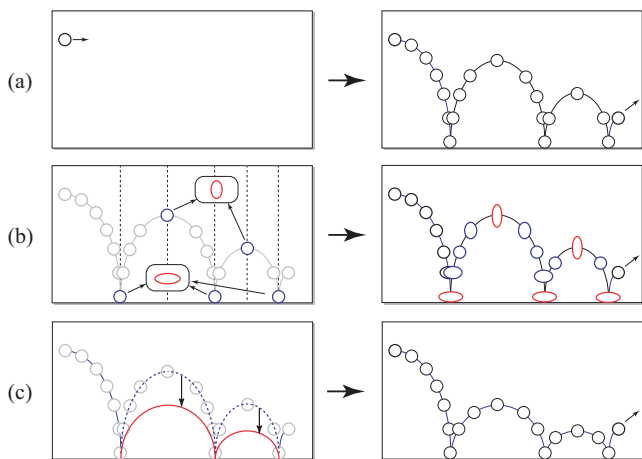


図 1: アニメーション制御のためのフレームワーク (a) 従来の物理法則アニメーション (b) 弾性物体の変形制御 (c) 弾性物体の軌道制御

り、弾性物体の柔軟な変形をユーザが制御するための手法はこれまで考えられてこなかった。

3 フレームワーク

ここでは物理法則に基づく弾性物体を対象とした、ユーザの意図するアニメーションの生成のためのフレームワークについて示す。弾性体変形に関しては、物理的に自然な変形を行うために、2章で挙げた物理法則に基づくアニメーション手法を用いることが望ましい。しかしながら、従来の物理法則アニメーション手法のみでは、ユーザの思い通りに形状変形を行うことは非常に困難である。先に述べたように従来のアプローチでは、与えられた外力に従って挙動が自動的に計算されるのみであり、その変形の結果を制御することはできない。そのためユーザの意図する結果に近い動きを得るためには、適切な初期状態のパラメータを手作業で探し当てる必要があり、これには多大な労力と時間が必要とされ、あまり現実的な解決策ではなかった。

一方で本手法では、従来の手法とは逆のアプローチをとる。物体の部分的な挙動をあらかじめユーザが指定し、それを元に全体のアニメーションを生成する。特に、ユーザの意図するアニメーションを実現するため、弾性物体の二種類の動きを入力情報とし、それらに対する制御機構を備える。

1. 物体の変形の制御。ある時点での物体の局所的な変形形状が、ユーザにとって思い通りのものになるように制御できるような機構。
2. 物体の軌道の制御。物体を巨視的に見たとき、その動く軌跡がユーザにとって思い通りのものになるように制御できるような機構。

ただし、これらの制約は物理法則に基づく動きの自然さとのトレードオフとなる。本手法では、物理的にそれらし

い動きをユーザが入力として与えてやることで、ユーザの意図する一連の自然なアニメーションを容易に生成することができることを示す。

図 1 は、アニメーション編集のためのフレームワークの概要を示しており、主に以下の三つのフェーズから構成されている。

物理法則アニメーション まず従来の物理法則アニメーションの手法を用いて弾性物体の挙動を計算する(図 1(a))。このとき、弾性体変形の計算モデルとして [14] を利用している。物体をあらかじめ四面体要素に分割しておき、四面体要素の各頂点の座標を x 、質量を M 、かかる外力を f_{ext} とすると、陰解法を用いて以下の運動方程式を解く。

$$M\ddot{x} + C\dot{x} + K(x - o) = f_{ext}, \quad (1)$$

このとき o 、 K は、それぞれ弾性体の安定状態における頂点座標とそのときの四面体の剛性行列を表す。弾性体の運動を計算する過程で、後の編集作業に利用できるようにその履歴を記録する。ここでは、各頂点の座標と速度を記録しておくことで、アニメーション中の任意の時刻での物体の位置や変形状態を復元することができる。

物体軌道の制御 物体の軌道を編集し、修正後の軌道に従ってアニメーションを修正する(図 1(c))。物体全体の軌跡が編集された軌道と一致するよう、運動計算の際に補正が施される。

物体変形の制御 物体のある時刻における変形形状を編集し、その編集後の形状に従ってアニメーションを修正する(図 1(b))。この形状の編集には、キーフレームアニメーションの考え方を利用する: ある時刻 t_i における形状をキーフレームとするとき、そこでの形状をユーザが修正する。するとキーフレームにおける修正形状を満たすよう、キーフレーム間の中間形状が再計算される。このとき、視覚的なもっともらしさを満たすために、新たに開発した有限要素法を用いた形状補間手法を施す(4章にて後述)。

上記のフレームワークの中で、まず物理法則アニメーションの手法を用いて元となる弾性体の動きの計算が行われ、その結果を元にユーザが意図に従い編集制御を進める。変形と軌道の制御は、それぞれが独立した作業となるが、その両方を組み合わせることも可能である。また編集後に行われる再計算により、物理法則アニメーションのもつ視覚的なもっともらしさを損なうことなく、自然なアニメーションを維持することができる。

図 2 に、本フレームワークにおけるユーザの操作と弾性体アニメーションの関係を表すダイアグラムを示す。まず式 (1) を解くことで、元となる弾性体の動きを計算する。こうして得られた動きに対し、ユーザは、キーフレームに基づく変形の制御や、軌道の制御を行う。これらの修正結

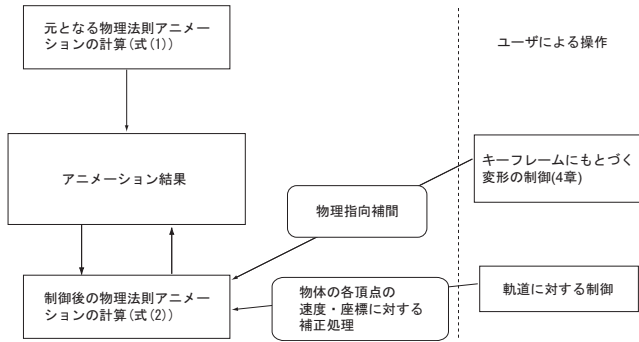


図 2: アニメーションの編集作業の流れ

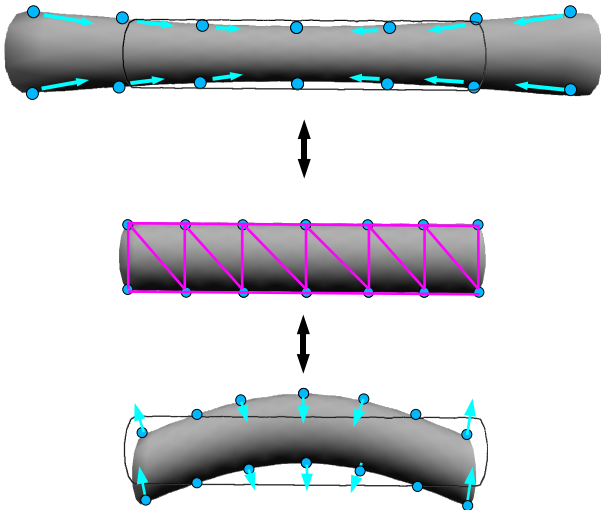


図 3: 弾性物体の安定形状と変形時の復元力

果を反映させて式 (2) を解くことで、弾性体アニメーションの再計算を行う (4章に後述)。このとき、ユーザによる制御の結果は、アニメーション計算に組み込む形で反映させる。ユーザによる修正と弾性体アニメーションの再計算の繰り返しにより、ユーザの意図が反映されたアニメーションが作成される。

4 キーフレームを用いた変形制御

はじめに弾性体アニメーションにおける運動方程式について述べる。弾性体変形の計算モデルとして Müller らの手法 [14] を利用しており、陰解法を用いて運動方程式 (1) を解く。ここで特に剛性行列 K と安定状態における頂点座標 o について着目する。これらは弾性体の元の形状に寄与し、通常はアニメーション全体を通じて変化せず一定の値をとる。この形状のことを安定形状 (Resting Shape) と呼ぶ (図 3中)。この安定形状を初期状態として、外から力を受けることで変形や伸縮が起るとき、物体の内部に安定形状に戻ろうとする弾性力が働く (図 3上・下)。

本手法では、ユーザによって指定されたキーフレームの形状を弾性体の安定形状とすることで変形の制御を行う。このとき単純にキーフレームの時刻において安定形状を入れ替えてしまうと、安定形状が急激に変化するため、弾性

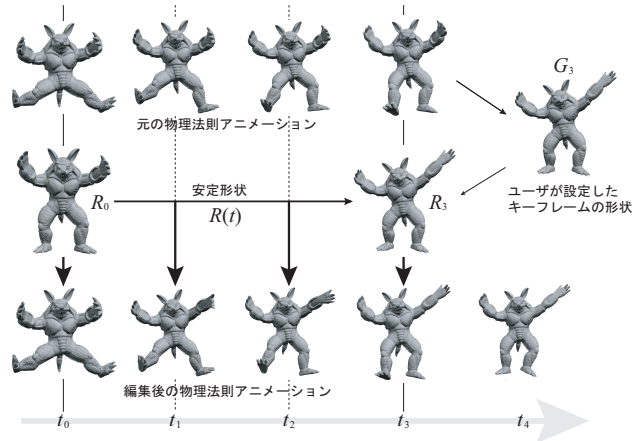


図 4: 変形のキーフレーム制御

体に瞬間的に大きな弾性力が生じ不自然な振動が発生してしまう。そこで時刻に対する補間関数 $R(t)$ を定義することで、前後の隣接するキーフレームの形状の補間を行う。そうして得られた毎時刻における $K(t)$, $o(t)$ を式 (1) に代入して解く。

$$M\ddot{x} + C\dot{x} + K(t)(x - o(t)) = f_{ext} \quad (2)$$

変形制御の概要を図 4に示す。元のアニメーションにおいて、安定形状 R は一連のアニメーションを通して一定である。一方、ユーザがキーフレームの形状 (t_i, G_i) を定義したとき、 R_i を G_i で置き換え、その間の補間関数 $R(t)$ の計算を行い $K(t)$, $o(t)$ を得る。これを用いて運動方程式 (2) を再計算することで、ユーザによる形状制御と弾性物体の物理法則の両方を考慮した動きが得られる。なお、ユーザは形状変形の制御を行うためにキーフレームを用いて新たな形状 (t_i, G_i) を挿入すればよく、挿入された際、安定形状の再計算や補間関数 $R(t)$ の計算は自動的に行われる。

5 物理指向補間

上述したキーフレーム群の安定形状の補間関数 $R(t)$ の計算を行う際、得られる中間形状 R_i は、弾性体として自然な変形形状であることが望ましい。一方で R_i は、顕著な体積変化や歪みを回避する必要がある。このことを実現するために、ここでは FEM を用いた新たな補間手法を提案する。類似する補間手法としては Alexa らによる近似剛体補間 [1] があるが、本手法は弾性体の計算モデルをそのまま利用することができ、より弾性体の動きに適した補間手法であると言える。

二つの隣接した安定形状 R_p, R_q 間の補間関数 $R_{pq}(t)$ について考える。ここでは簡単のため、形状の剛体変換 (移動と回転) を無視し、両形状の変形に対する補間のみを考える。また両形状間の自然な変形過程を得るため、この変形は局所的な外力に因るものではないと仮定し、 $f_{ext} = 0$ とする。

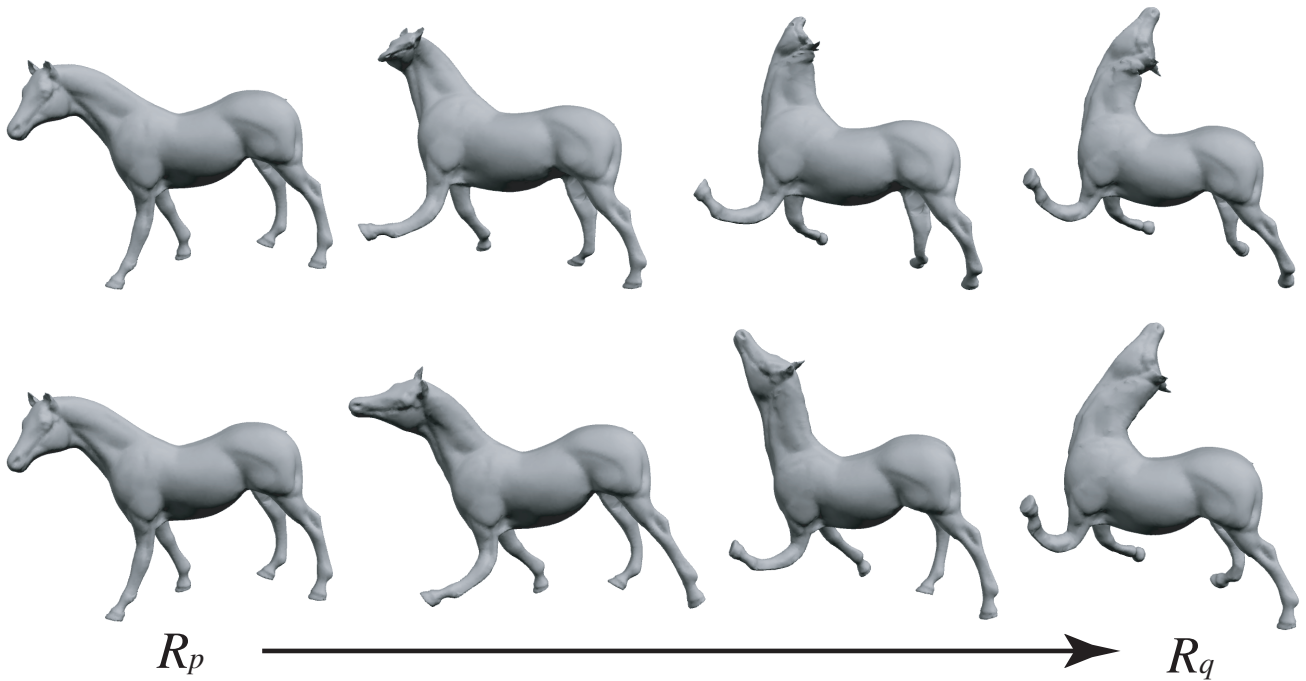


図 5: 物理指向補間と線形補間による変形の比較

$R_{pq}(t)$ は, R_p を出発点として以下の式を解くことで得る:

$$M\ddot{x} + C\dot{x} + K_q(x - o_q) = 0 \quad (3)$$

ここで K_q, o_q はそれぞれ R_q の剛性行列と各頂点座標を示す。式 (3) は形状 R_p の状態から, 安定形状 R_q に向かって弾性力によって復元する弾性体の運動を表している。ここでは Müller らの手法と同様に, 陰解法を用いて解く。

しかしながら形状 R_p, R_q 間の変位が大きいと, 計算の最初の段階で物体内部に生じる弾性力が非常に大きくなるため, 計算が不安定になったり過剰な振動が現れる可能性が高い。これを解決するため, 以下の二つの手法のどちらかを用いることで弾性力を抑制する。一つは速度をステップ毎にリセットして速度の蓄積をなくすことで, 過剰な振動を抑える。もう一つは減衰係数を表す対角行列 C の値を従来よりも大きく設定し, 減衰力を強くすることで振動を抑える。我々の実験では, およそ 5 ~ 10 倍の値に設定することが望ましい。

図 5 に物理指向補間の結果を示す。図上の線形補間した場合と比べ, 変位の大きな部位であっても, 体積変化や歪みの少ない自然な補間形状が得られることが確認できる。また, 従来の弾性体の計算モデルの範囲で計算が行えるため, 余分な実装を必要としない。

6 結果

本手法による弾性体アニメーションの変形制御の結果を図 6 に示す。約 900 個の四面体要素からなる体積メッシュを用いて運動と形状制御および衝突処理の計算を

行い, その一方で描画にはおよそ 2 万ポリゴンからなる密な表面メッシュを用いた [12]。計算機には Pentium4 2.5GHz(RAM 1GB), GeForceFX Go 5600 を使用した。

物体全体の時刻に対する位置と方向を記録した軌道情報を用意しておき (図中曲線), アニメーションの運動計算の際, その情報を元に各頂点の速度・座標に対し補正を行っている。これにより, ユーザが形状に編集を加えることで物体の軌道が大きく変化してしまうのを防ぎ, ユーザは弾性体の形状制御のみに集中することができる。

まず最初に安定形状が一定である元のアニメーションを計算する (図 6(a))。次に得られたアニメーション結果をもとに, 任意の時刻での形状を修正し, 新たな安定形状のキーフレームとして定義する (図 6(b))。アニメーションの再計算を行うと, キーフレームの形状に従って安定形状が変化し, それに誘導されるようにしてキーフレームを反映した弾性体の変形が行われる (図 6(c))。ユーザがキーフレームの形状を挿入した際, 物理指向補間の処理に約 1 秒を要した。その後のアニメーションの再計算はインタラクティブに処理することができた。

元のアニメーションは安定形状が変化しないため, 言えばゴム人形のような動きが得られる。一方で本手法では, ユーザの定義したキーフレームの形状を考慮しながらも物理法則に即した変形を行うため, 躍動感と自然さを併せ持ったアニメーションが得られた。

一方で, より複雑なアニメーションを生成するための課題となる点も見受けられた。最も大きな点は, ユーザが定義したキーフレームの形状に対し, 実際の再計算されたアニメーション結果ではそれより僅かに遅れて変形が行われることである。本手法では, 安定形状 $o(t)$ とその剛性行

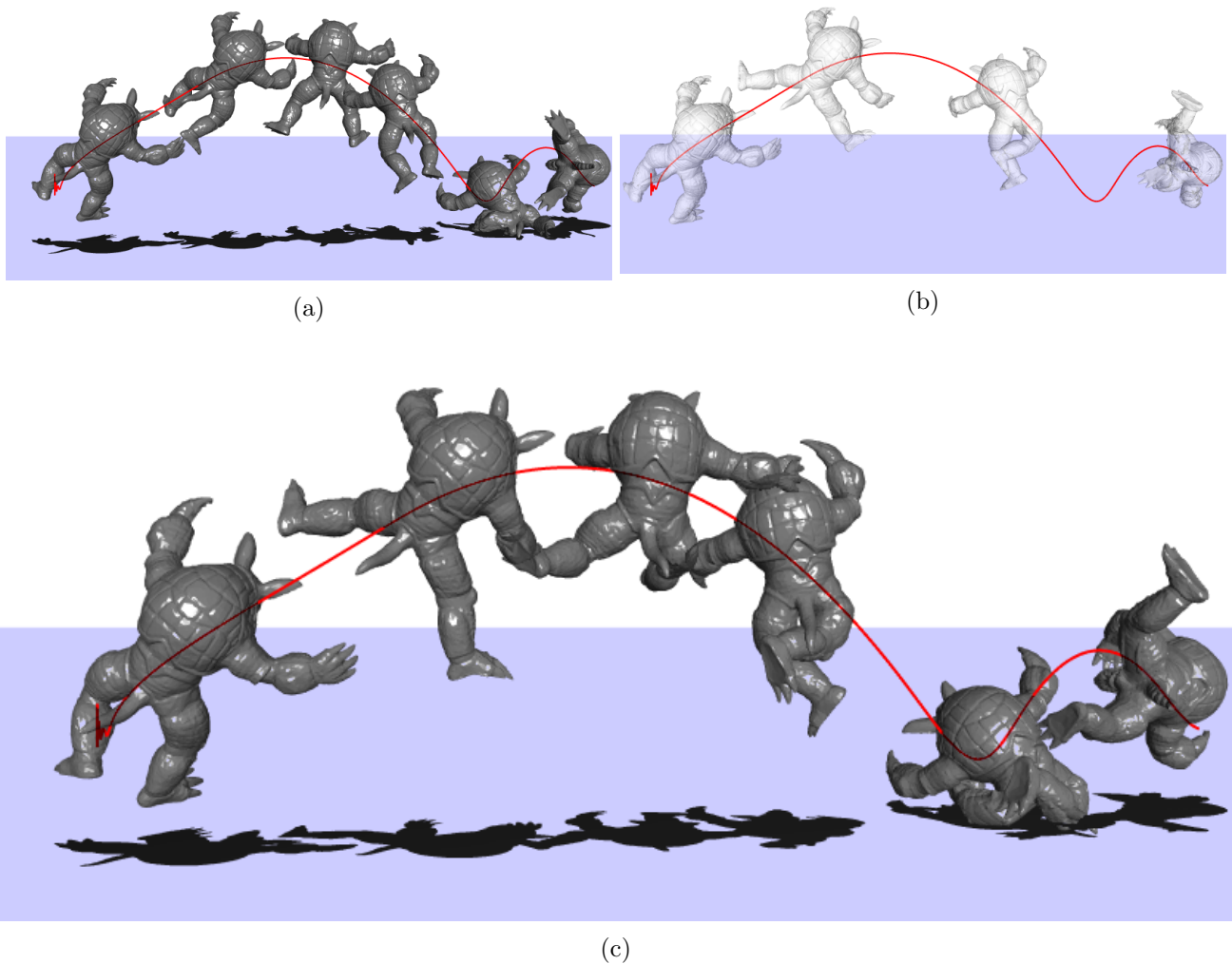


図 6: 本手法による変形制御の結果 (a) 元の弾性体アニメーション (b) ユーザの与えた形状キーフレーム (c) 形状制御の結果

列 $K(t)$ を変化させながら運動方程式 (2) を解くことで、物体の形状変形の誘導を行う。弾性体では、物体の形状 x と安定形状 $o(t)$ に差が生じて、初めて復元力が発生しているため、実際の変形移動がなされるまで数フレームの遅れが生じる。より柔らかい物体であれば剛性が低いいため十分な復元力が得られず、この傾向が強くなる。同様の理由により、ユーザが定義した安定形状のキーフレームの間隔が狭すぎる場合、物体の変形制御が追従しきれない場合もある。

この問題の解決策としては、ユーザのキーフレームへの追従を満たす幾何学的な変形と、物理法則に基づく弾性体変形の度合いを、ユーザ自身が調整することで、意図するアニメーションに近づけるという方法が考えられる。物体の形状を、キーフレームの形状に厳密に追従させた場合、つまり $x = o(t)$ であった場合、衝突変形や弾性体特有の振動など、物理的に自然な動きは得られないもの、ユーザの意図するキーフレームの形状に一致する。一方、追従を弱めるほど物理法則に基づいた挙動となり、先のキーフレームへの遅れの問題が生じる。そこで、作り手が意図するアニメーション結果が得られるよう、追従の度合いを調

整する機構が必要になると考えられる。

また本手法は、弾性体のアニメーションの中で、特に物体の変形の制御に着目している。現在の実装では、物体全体の移動や回転など軌跡に対する制御については簡単のため、編集された軌道情報と一致するように、各頂点の座標と速度に対し均一に補正を行っている。今後、より自由度の高い自然な動きを効率よく得るためには、[17, 16] などの手法を応用し軌道情報の生成をユーザの意図を反映しながらも自動化することや、変形制御に応じた軌跡の自動補正などの機能が必要であると考えられ、今後の課題である。

7 結論

本論文では、弾性体アニメーションを物理法則に基づいてより直感的に制御するための手法を提案した。ユーザが満たすべき形状をキーフレームを用いて指定することで、物理法則に基づいた自然な変形を満たす動きを得ることができた。

本研究では、弾性物体の安定形状を変化させることで変形の制御を行っており、弾性体運動を大幅に逸脱すること

なく任意の形状に近づけることが可能である。また、二つのキーフレームの形状を補間するため、有限要素法を用いた形状補間手法を提案した。これにより物体内部の体積変化が抑えられ、物理的に自然な中間形状が得られた。

今後の課題としては、前章で述べたように、キーフレームと物理法則による運動の追従の度合いの制御機構を備えることが挙げられる。また、より洗練されたユーザーインターフェースを開発し、統合されたアニメーション編集フレームワークに組み込むことで、ユーザの思い通りの変形物のアニメーションを、効率良く生成・編集することが可能になると考えられる。

謝辞

本研究の一部は、独立行政法人科学技術振興機構、CREST の支援の下で推進されました。

参考文献

- [1] M. Alexa, D. Cohen-Or, and D. Levin. As-rigid-as-possible shape interpolation. In *Proc. SIGGRAPH '00*, pp. 157–164. ACM Press, New York, 2000.
- [2] D. Baraff and A. Witkin. Partitioned dynamics. Technical Reports CMU-RI-TR-97-33, Robotics Institute, Carnegie Mellon University, Pittsburgh, 1997.
- [3] R. Barzel, J. F. Hughes, and D. N. Wood. Plausible motion simulation for computer graphics animation. In *Computer Animation and Simulation '96 (Proc. Eurographics Workshop)*, pp. 183–197. Springer Wien, New York, 1996.
- [4] S. Capell, S. Green, B. Curless, T. Duchamp, and Z. Popović. Interactive skeleton-driven dynamic deformations. In *Proc. SIGGRAPH '02*, pp. 586–593. ACM Press, New York, 2002.
- [5] S. Chenney and D. A. Forsyth. Sampling plausible solutions to multi-body constraint problems. In *Proc. SIGGRAPH '00*, pp. 219–228. ACM Press, New York, 2000.
- [6] P. Faloutsos, M. van de Panne, and D. Terzopoulos. Composable controllers for physics-based character animation. In *Proc. SIGGRAPH '01*, pp. 251–260. ACM Press, New York, 2001.
- [7] R. Fattal and D. Lischinski. Target-driven smoke animation. In *Proc. SIGGRAPH '04*, pp. 441–448. ACM Press, New York, 2004.
- [8] M. Gleicher. Comparative analysis of constraint-based motion editing methods. In *Graphical Models*, Vol. 63, pp. 107–134. Academic Press, 2001.
- [9] R. Grzeszczuk, D. Terzopoulos, and G. Hinton. Neuroanimator: Fast neural network emulation and control of physics-based models. In *Proc. SIGGRAPH '98*, pp. 9–20. ACM Press, New York, 1998.
- [10] G. Hirota, S. Fisher, and A. State. An improved finite element contact model for anatomical simulations. *The Visual Computer*, 19(5):291–309, 2003.
- [11] G. Irving, J. Teran, and R. Fedkiw. Invertible finite elements for robust simulation of large deformation. In *Proc. ACM SIGGRAPH Symposium on Computer Animation (SCA 2004)*, pp. 131–140. ACM Press, New York, 2004.
- [12] R. Kondo and T. Kanai. Interactive physically-based animation system for dense meshes. In *Proc. Eurographics 2004 short paper*, pp. 93–96, 2004.
- [13] A. McNamara, A. Treuille, Z. Popović, and J. Stam. Fluid control using the adjoint method. In *Proc. SIGGRAPH '04*, pp. 449–456. ACM Press, New York, 2004.
- [14] M. Müller, J. Dorsey, L. McMillan, R. Jagnow, and B. Cutler. Stable real-time deformations. In *Proc. ACM SIGGRAPH Symposium on Computer Animation (SCA 2002)*, pp. 49–54. ACM Press, New York, 2002.
- [15] J. F. O'Brien, V. B. Zordan, and J. K. Hodgins. Combining active and passive simulations for secondary motion. *IEEE Computer Graphics & Applications*, 20(4):86–96, Aug. 2000.
- [16] J. Popović, S. M. Seitz, and M. Erdmann. Motion sketching for control of rigid-body simulations. *ACM Transactions on Graphics*, 22(4):1034–1054, 2003.
- [17] J. Popović, S. M. Seitz, M. Erdmann, Z. Popović, and A. Witkin. Interactive manipulation of rigid body simulations. In *Proc. SIGGRAPH '00*, pp. 209–217. ACM Press, New York, 2000.
- [18] A. Treuille, A. McNamara, Z. Popović, and J. Stam. Keyframe control of smoke simulations. In *Proc. SIGGRAPH '03*, pp. 716–723. ACM Press, New York, 2003.
- [19] A. Witkin and M. Kass. Spacetime constraints. In *Proc. SIGGRAPH '88*, pp. 159–168. ACM Press, New York, 1988.