

translated by Nobuo Nakagawa  
nobuo\_nakagawa@outlook.com

Matthias Müller

Nuttapong Chentanez  
NVIDIA

Tae-Yong Kim

Miles Macklin



図1:複数のレイヤーを持つスカートを履いたダンサー。我々の手法は、堅牢に複雑な相互作用をシミュレーションし、あらゆる絡みあった状態からスムーズに正常な状態へ復帰させることができます。

## 概要

我々は、近接接触する複雑な変形オブジェクトを扱うのに適合した衝突検出方法と衝突応答の両方のための新しい手法を提案します。我々の手法は、時間刻み間の衝突イベントを取り逃すことなく、もつれを自動的にほどき難しい問題を堅牢に解決します。この手法の概念は単純で、そのアルゴリズムの規則性により、率直に並列化が可能です。

主要なアイデアは、オブジェクト間の空気をシミュレーションの前に一度セレーションし、シミュレーションの間は、その反転を防ぐために要素ごとに單一方向の制約を考慮します。もし、相対的に大きな回転と並進移動がシミュレーション中に存在する場合、追加的な動的メッシュ最適化ステップがメッシュのロックを防ぐために必要になります。このステップは、2Dの場合高速で、任意のシーンでのシミュレーションを可能とします。しかしながら、3Dの場合は、メッシュの最適化コストが高いので、3Dのシナリオの部分集合、比較的モーションが限定的な場合に適した手法です。この部分集合には2つの重要な課題が含まれています。複数レイヤーのクロスシミュレーションと、アニメーションする人物の薄い織物の表現です。

**CR Categories:** I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Physically Based Modeling; I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation and Virtual Reality

**Keywords:** collision detection, mesh optimization

## 1 導入

衝突検出と衝突後処理は、どちらも挑戦的な課題であり、コンピュータ・グラフィックスで活発な研究領域です。[Teschner ら 2005], [Yoon ら 2010]衝突検出の既存手法は、一般に、重複したプリミティブの検出を行うために均一格子や境界ボリューム階層などの空間最適化構造を用います。薄かったり、高速に動くオブジェクトの場合、衝突イベントを一切取りこぼす事のない静的なオブジェクトボリュームよりもむしろ、時間刻みを超えたスイープボリューム

を考慮する必要があります。衣服のレイヤーのように、近接した複雑なオブジェクトのためにこのような処理が必要となりますが、この処理は高価でシミュレーションのボトルネックになります。

交差したプリミティブのペアが見つかれば、その交差を解決するために、適切な応答力を加える必要があります。衣服や、深く貫通したボリュームのあるオブジェクトのように薄い構造のために、正確な応答力の方向を探すのは挑戦的な課題です。絡みあった状態から適切な状態に戻すことは、全体問題となり、個別に衝突のペアを分析する代わりに全体のオブジェクトが考慮される必要があります。[Wicke ら 2006]この問題を解決する方法は、正確な全ての衝突応答を正確に行いねじれた状態を避けることです。[Brochu ら 2012]これらの手法は、計算コストが高くなりがちです。さらに、人物のように運動するオブジェクトは、[Baraff ら 2003]のようなケースでは、衣服が貫通する事を避けることが不可能です。

このプロジェクトでは、私たちは特に、薄い複数レイヤーの衣服を着た人物の衝突応答とそれら2つの相互作用(図1)について関心があります。このシナリオでは、ある時点での時間では、絡みあった状態が避けられない事がたびたびあります。例としては、人物が座ったり、横たわったり、脇の下に被さる衣服などが挙げられます。さらにもまた、衝突の解決にかけられる時間が限られている場合の絡みあった状態があります。典型的には、一時的な衝突が時間経過に応じて解決される手法が許容されます。その一方継続的に絡み合った状態は許容されません。

それゆえ、我々の手法で最も重要な機能は、あらゆる絡みあった状態からスムーズに正常な状態へと復帰する機能です。これは、ゲーム向けの薄く、複数のレイヤーの衣服シミュレーションでは鍵となる機能となります。我々の手法は、衝突検出と衝突応答を同時に扱うことも可能で、連続的な衝突が考慮されています。アルゴリズムは、概念的にシンプルに理解・実装が可能です。

我々の貢献は:

- 堅牢な衝突応答のためにオブジェクト間の空気を三角形化するアイデア

- ・単一方向の制約を用い三角形要素の体積を正になるように強制することで衝突を避けること。
- ・メッシュ最適化を用いることで、お互いのオブジェクトに対し自由に動く事を可能とする。

## 2 関連研究

このセクションでは、我々は、最も関連の強い衝突検出と衝突応答に関する先行研究をカバーします。そして読者には、[Teschner ら 2005], [Yoon ら 2010] をこのトピックに関するより完全なサービスとして紹介します。

### 2.1 変形オブジェクトのための衝突検出

オブジェクト間と衣服や組織などの変形オブジェクトの自己衝突の検出のために、全てのプリミティブ、例えば三角形や四面体のペアが交差しているかどうかをテストされなければなりません。典型的には、この仕事は潜在的なペアの数の多さや、オブジェクトが変形する全てのフレームでのプリミティブの変形の間の空間構造が変化する事実により、衝突応答のパイプラインの最も計算コストが高い部分となります。それゆえ、不要なテストを効果的に捨てるために様々な高速化手法が提案されてきました。例としては、空間ハッシング [Teschner ら 2003]、直行三角形集合の色分解 [Govindaraju ら 2005] 接続性ベースカーリング [Tang ら 2008]、分離軸テスト [Tang ら 2011]、放射ベースカーリング [Wong ら 2013]、[Wong と Cheng 2014]、[Wong ら 2014] シミュレーション空間ベースカーリング [Barbic と Jame 2010]、[Zheng と James 2012]、代表プリミティブ使用 [Curtis ら 2008]、GPU を用いたオブジェクトのラスタライズ [Govindaraju ら 2003]、[Heidelberger ら 2004]、[Faure ら 2008]、[Wang ら 2012]、様々な階層的な手法 [Mezger ら 2003]、[Schvartzman ら 2010]、[Schvartzman ら 2009]。

### 2.2 連続的な衝突検出

変形するオブジェクトの場合、一つでも衝突を見逃すと後で回復が難しいねじれ状態を引き起こします。[Bridson ら 2002] は、衣服のメッシュの三角形間の連続的な衝突検出(CCD)を用いた堅牢な衝突応答を提案しました。CCD のコストを削減するために様々な手法が提案されてきました、例えば、保守的前進法 [Zhang と Kim 2012]、非貫通フィルタ [Tang ら 2010]、境界ボリューム階層 [Hutter と Fuhrman 2007]、[Wing ら 2010] などがあります。解析的に正確ですが、これらの手法は数値的な不正確さにより間違った値になることがあります。この問題は、[Brochu ら 2012] CCD テストの為の正確で区間計算法を用いて取り組まれています。[Wang ら 2014] は、数値閾値の選択を自動化する基準を提案しています。[Tang ら 2014] はバーンシュタイン符号分類に基づいたブーリアン衝突クエリを用いています。

### 2.3 体積ベース手法

我々の手法は、これら全ての手法とは少し異なり、オブジェクト間の空間が時間刻みの間に反転したかどうかをテストすることで衝突を検出します。[Sifakis ら 2008]もまた、オブジェクト間の空間について研究されていますが、永続的なメッシュの反転を検出する代わりに、それぞれの時間刻みの始まりで再初期化される一時要素の体積を保存しています。オブジェクト間の体積を考慮する代わりに、[Allard ら 2010] は、オブジェクトが交差している体積から接触力を導きました。[Levin ら 2011] は、個体間の衝突を扱うオイラー手法を用いました。この設定では、オブジェクト間の空気の体積は、よくモデル化されていますが、我々の手法とは対照的に、規則的な固定背景格子を用いています。ここでは、衝突は伝統的な方法で処理されます。なぜならセルはその形状を維持しますし、反転しないからです。

衝突応答のためにオブジェクト間の体積を考慮するというアイデアは、グラフィックスコミュニティ外から提案されてきました。[Agarwal ら 2000] は、彼らが擬似三角形化と呼ぶ運動データ構造を 2D オブジェクト間の衝突検出に用いる方法を主張しています。我々のシンプルな辺の反転手法と対象的に、これらの筆者は真の衝突だけを検出して解決しています。しかしながら、その複雑性と 2D の制限の為に、手法は複雑なオブジェクトの実時間シミュレーションには適していません。[Pagano ら 2008] もまた、衝突検出の反転を防ぐアイデアについて議論していますが、主に連続的なケースであり、衝突後の状態復帰の議論はなく多くはメッシュベースの側面での議論がなされています。これらの手法はまた 2D の制限があります。

### 2.4 もつれの解決と履歴ベース処理

時々、変形オブジェクトはもつれた状態を強いる事があります。衣服のシミュレーションにおいては、運動する人物の脇の下や座ったり、横たわったりした時にこの問題が起ります。この状況では、CCD アルゴリズムでは、避けることが出来ません。この問題に対処するために、[Baraff ら 2003] は、衣服のもつれを解く手法を提案しました。これは、衣服のジオメトリの交差を分析して、時間に応じてもつれをきちんと解く応答する力を計算する方法です。手法は、[Wicke ら 2006] によって拡張され、境界のあるメッシュも扱うことができ、[Volino ら 2006] 輪郭最小化、手法がもつれをほどくために提案されました。

もつれの問題を解決するための異なるアプローチとして、位置の修正をより早い段階で行い、衝突しないようにすることです。そのような履歴ベースの手法が、[Selle ら 2008] により提案され、引力と斥力のどちらかが衝突ペアの解決に必要とされます。我々の手法は、同じカテゴリーに分類されます。Selle らの研究は、もつれ状態のメモリを毎時間刻み毎に再初期化して時間刻み間の繰り返し手法でしか用いない一方、我々はシミュレーション開始に一度だけ初期化し全シミュレーション計算に用います。我々のケースでは、メモリーは、プリミティブの相互の空間的位置で構成されます。複数レイヤーの衣服のデザインに特化したより特殊なメモリ配置が [Perez ら 1999] によってデザインされました。これは、それぞれのレイヤーをアニメーションする骨構造の周りで定義された同一表面に向けて押す手法です。[Wong ら 2004] は、衣服の頂点を、重心補完に加え幾つかの前の層の中の三角形からの法線のオフセットを与えられたレイヤーに加えた衣服の頂点を押す手法を提案しています。

複雑な形状を堅牢にシミュレーションするよく知られる手法は、形状を周囲の四面体メッシュへと拡張することです。[Müller と Gross 2004] 我々のアイデアの 2 つの主要な貢献は、エアーメッシュの頂点と同一の空間を占めるシミュレーションされた可視形状の頂点と、エアーメッシュは、それらの体積が正でない場合は完全に無効となることです。付属の動画の中で、我々は、図 1 の中のスカートが、完全に体積保存を有効にした場合に、どのように意義深いのか示します。

### 2.5 反転の処理とメッシュの改善

我々の手法の基本となるアイデアは、オブジェクト間のエアーメッシュの追跡です。衝突処理では、2 つのタスクを行います。文字通り反転の処理とメッシュの改善です。反転した要素の扱いは、変形形状のシミュレーションに有限要素ベースの手法が関連しています。基本的なアイデアは、変形テンソルの最小の負の固有値を決定し、対応する固有ベクトルから求めた方向へ復元力を適用することです。我々は位置ベースの物理計算手法を用いるので、それぞれの要素の体積が正になるように単純に单一方向の制約を加えるだけです。[Muller ら 2007]

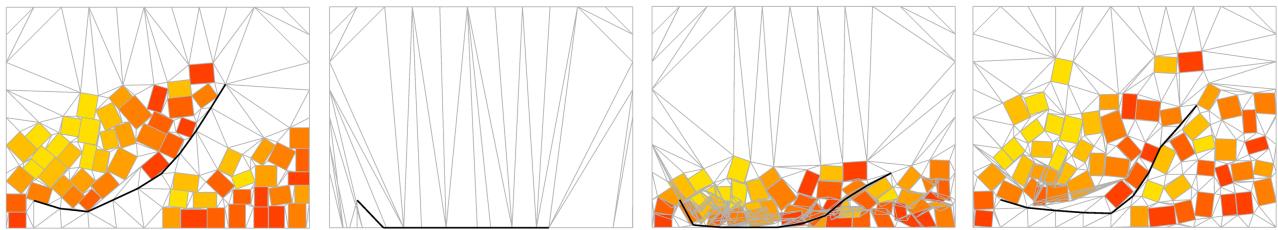


図2:60個の箱とロープで構成される2Dのシーン。エアーメッシュ(白)は、安定的に1次元のロープをすり抜けるのを防ぎます。全シーンをX軸へ圧縮した後でも、オブジェクトとその相対位置が復元されている事がわかります。

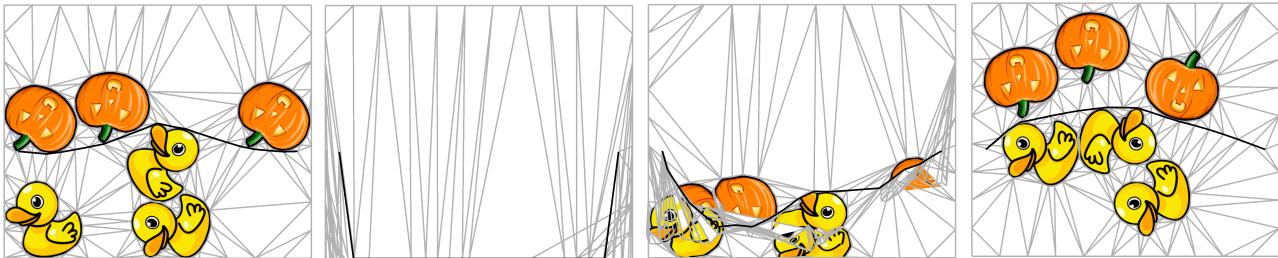


図3:2Dでの一般的なオブジェクト。もう一度。シーンが一直線に圧縮された後でさえエアーメッシュが、ロープの正確な側のオブジェクトを保持していることがわかります

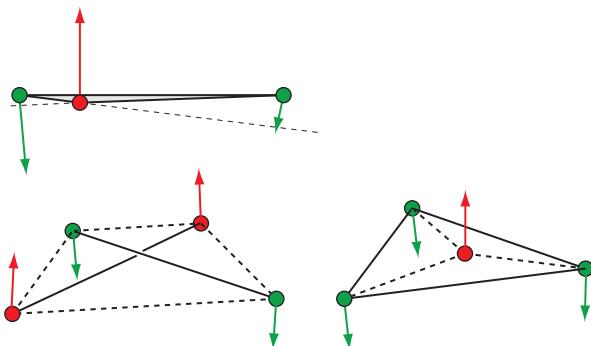


図4:完全にロックしない2D(三角形、上段)と3D(四面体、下段)の2体の衝突のためのエアーエлементの体積制約勾配ベクトル表示。

四面体メッシュの改善に関して多くの研究があります。[Fritag と Ollivier-Gooch 1997] は、平滑化と様々な幾何的な操作(例えば、一般的なエッジ消去を行いメッシュの品質を改善させるのと同様に、2→3 と 3→2 のフリップなどの操作など)を活用しました。[Klingner と Shewchuk 2007] は、頂点挿入操作と、メッシュ品質を改善させる様々な発見的手法を活用しました。この手法は、さらに [Wicke ら 2010] によって改善され大きな弾塑性変形物体のシミュレーションが行われています。我々は、一般的なエッジ消去法を用い、[Shewchuck 2002] の手法で複数面の消去を行いました。

### 3 手法

我々の手法は、3つのステップで構成されています。最初に、シミュレーション前のプリプロセスステップと同様にオブジェクト間のエアーメッシュをテセレーションします。シミュレーションの間、我々は、エアーエлементが反転するのを防ぐために単一方向の(不等式)制約を考慮します。三番目に、大きな回転時の不自然な結果を避けるために、動的メッシュ最適化ステップをシミュレーションの間に実行します。(図 6を参照ください)それでは、この3つのステップのより詳細を見ていきましょう。

#### 3.1 オブジェクト間の空間のテセレーション

事前計算のステップとして、オブジェクトの間のエアーエлементがテセレーションされます。我々のシンプルで包括的なサンプル(図 2, 5, 6)では、エアーエлементを手続き的に生成しています。一般的なケースでは、オブジェクト間の空間に対してシミュレーションされたメッシュ表面と合致するだけのテセレーションが必要となります。このステップはシミュレーションが開始する前に一度だけ実行すればよいので、計算時間は深刻なものにはなりません。この仕事の為に用いることができる既存のパッケージがあります。たとえば、[Si2015]のTetGenです。

#### 3.2 要素毎の制約

シミュレーションの間中、エアーエлементは、その体積が正である限り無抵抗です。要素が反転した時のみ、その体積を0にするように復元力を発揮します。そのような、單一方向の制約が、力あるいは力積ベースのシステムの構築を可能とします。我々は、我々のシミュレーションに位置ベース物理手法(PBD) [Muller ら 2007] を用いるのでエアーエлементの單一方向の体積制約をシンプルに考慮することができます。

$$C_{\text{air}} = |(\mathbf{p}_2 - \mathbf{p}_1) \times (\mathbf{p}_3 - \mathbf{p}_1)| \geq 0 \quad (1)$$

$$C_{\text{air}} = \det [\mathbf{p}_2 - \mathbf{p}_1, \mathbf{p}_3 - \mathbf{p}_1, \mathbf{p}_4 - \mathbf{p}_1] \geq 0 \quad (2)$$

2D および 3D の場合、それぞれ、 $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4$  は、要素に隣接した頂点となります。2つの隣接物体の、反転せず、ロックしないエアーエлемент集合が図4で示されます。一般的な、プリミティブの衝突するケース、言い換えると、2Dでの頂点-辺の衝突と、3Dでの辺と辺、面と頂点の衝突を扱っています。運動量は、PBD の更新の際に保存されます。例えば、2Dの赤いパーティクルに近い緑のパーティクルは、より大きな位置の更新を受けます。体積制約の勾配を用いる代わりに、もし、それらの参照がパーティクルに保存されていれば、衝突オブジェクトから法線方向を直接計算することも可能です。

我々の、並列ヤコビ統一ソルバは、きめの細かい処理を可能とし、様々な制約の処理を交互に行います。理論的に、衝突処理カーネルは、ソルバのループの中のあらゆる位置に配置が可能です。しかしながら、弾性性よりも衝突応答処理をより優先するために、我々は、その計算処理をそれぞれのイテレーションの中で最後に行うことで、複数の時間刻みのトータルでよりよい衝突応答を可能とします。

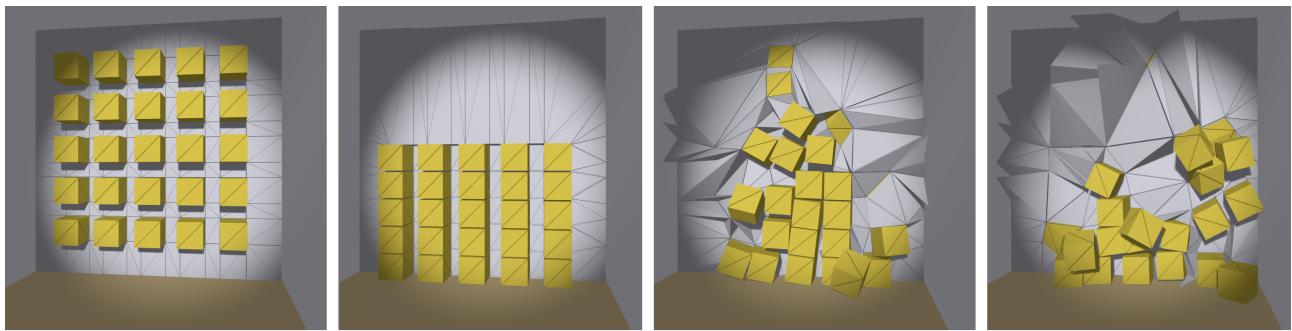


図5: 3Dシーンの抜粋、黄色い部分は四面体オブジェクトでグレイ部分はエアーメッシュとなります。左1:初期状態 左2:重力を有効にした時、箱が床に落ち、安定した積み重なった状態を形成します。右1,2:メッシュ最適化により、オブジェクトは、それぞれに対して自由に動きます。

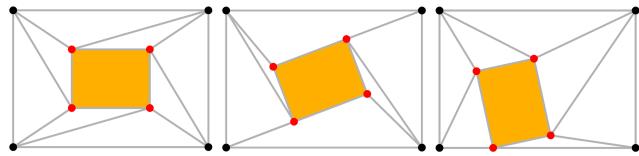


図6:左:動的なオレンジのボックスが、長方形の境界に囲まれているシンプルな2Dのケース。白い三角形が反転するのを防ぐために、箱は、壁に対して正確に衝突します。中:境界に対して相対的に大きく回転した時、エアーテンソルは、箱と衝突しない場合でも三角形はロックします。右:フリップによるエアーメッシュの最適化によりこの問題を解決し箱が自由に動く事を可能とします。

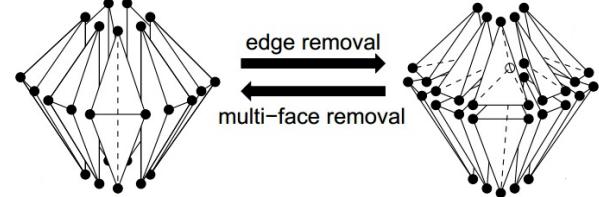


図7:四面体メッシュにおける、辺の除去と複数面の除去操作。イメージは、Jonathan Shewchuk の提供によるものです。

とで、複数の時間刻みのトータルでよりよい衝突応答を可能とします。

### 3.3 メッシュ最適化

図6の実験が示すように、エアーテンソルは、オブジェクト間に衝突がない場合でさえ反転することがあります。この問題は、オブジェクトが比較的大きな回転あるいは移動を行った際に生じます。そのようなケースでは、エアーメッシュはロックしオブジェクトが自由に動けなくなってしまいます。[Agarwal ら 2000] によって提案された、動的な擬似三角形化手法は、この問題を解決し正確な衝突のみを報告します。理論は形式的に説明されました。しかしながら、先に述べたとおり、このデータ構造とその更新アルゴリズムは、極めて計算コストが高いものです。我々は、辺の単独フリップがロックをとても効果的に防ぐことを発見しました。図2と図3で示される、2Dのケースで、我々は、隣接三角形の品質を改善するときはいつでも辺のフリップを実行します。我々は、下記を

$$q_{triangle} = \frac{4}{\sqrt{3}} \frac{A}{l_1^2 + l_2^2 + l_3^2} \quad (3)$$

品質測定に用います。Aは、三角系の面積で  $l_1, l_2, l_3$  はそれぞれ三角系の辺の長さを示しています。この測定では、1を等辺の三角形と考えます。辺の単独フリップでロックが起こらない理論的な保証は無いけれども、添付のビデオが示す通り、大量のオブジェクトと複雑な交差があるシーンでもいかなるロック不具合が生じにくい事がわかります。

辺の単独フリップが、図6の回転を自由に行えるようにしたことは明らかでした。オブジェクトの辺の距離拘束と辺のフリップつきのシンプルなエアーダミングが、沢山の自由に動くオブジェクトとロープの完全な2D柔軟体のシミュレーションを可能としています。継続的な衝突検出と安定したスタッキングも図2と図3で示されています。

全てのフリップが許容されているわけではないことに言及しておくことが重要です。ある辺のフリップは、両方の隣接三角形がエアーメッシュに属していて、辺自身がロープの一部では無いときのみ許容されます。その一方、オブジェクトは自由に貫通することが出来ます。

図5で示す実験の通り、四面体メッシュは、三角形メッシュの最適化よりもより複雑な問題となります。同じ概念が3Dのシーンでもうまく働きます。単純な辺のフリップの代わりに、様々な最適化処理がこれまで提案されてきました。我々は、[Shewchuk 2002]に基づいた、辺の除去と複数面除去手法を用います。これは、図7で示され、品質は下記で図られます。

$$q_{tetrahedron} = \frac{12}{\sqrt{2}} \frac{V}{l_{rms}^3}, \quad l_{rms} = \sqrt{\frac{l_1^2 + l_2^2 + l_3^2 + l_4^2 + l_5^2 + l_6^2}{6}}, \quad (4)$$

ここで、 $l_{rms}$ は、平方平均のルートで、 $l_1, l_2, l_3, l_4, l_5, l_6$ は辺の長さで、Vは、四面体の体積です。この値が1の場合正四面体となります。2Dのケースと同様に、最適化操作は、全ての除去四面体の面が、エアーセルに隣接していて、布の一部では無い時のみ許されます。

## 4 議論

この節では、我々の手法の特性について議論していきます。

### 4.1 メモリ

我々の手法の重要な側面は、エアーメッシュが、オブジェクトの相対位置のメモリを持つということです。この情報は、エアーテンソルの回転を保持し、要素が完全に平らになってしまってもなお失われないということです。

[Selle ら 2008] のそれぞれの時間刻みの最初で再初期化されるアプローチとは対照的に、シミュレーションの開始時に一度だけ初期化されます。そして、時間刻みをまたがってメッシュ最適化処理に一貫して用いられます。[Selle ら 2008] と異なり、この情報は絡みあつた状態から正確に復帰することを可能とします。

[Muller ら 2005] の完全にフラットにつぶれた状態からアヒルが完全に元の形状に戻る実験に影響を受けて、我々は、グリム童話のブレーメンの音楽隊 [Grimm と Grimm 1812] のシーンから開始しました。このシーンでは、4匹の動物がそれぞれ積み重なっています。(図8 参照) それらが、全てのY座標が0になるように圧縮された後、-物語の一部ではありません - それらが個々に復元されるだけではなく、それらの頂点順も完全に保存されています。

図2と図3は、2Dでの同等な実験を示しています。2番目のイメージは、全てのY座標を0に設定して圧縮しています。続くスクリーンショットは、オブジェクトが正確な順で跳ね上がりロープの正確な側にとどまつたまま、完全なシーンが復元されるかを示しています。

さらに実践的で、より重要なケースが図10の衣服のもつれを解くケースです。我々の手法は、もつれ合った状態を検出するだけではなく、任意のもつれ合った状態から複数の時間刻みに渡ってなめらかで安定した復元を可能とします。我々はこのような能力は既存の手法では実現できないと考えています。

#### 4.2 最適化を無効にする

3D メッシュの最適化の複雑性により、エアーメッシュは、典型的な、自由な移動や、バケツにオブジェクトを落とすようなベンチマーク用のシーンでは最も良い選択肢とはなりません。しかしながら、そのようなシーンはゲームや映画でよくあるケースではありません。

幸運なことに、最適化ステップがない場合でも一部の問題では、我々の手法はとても良く働きます。特に図1や図9で示されるゲームや組織の衣服シミュレーションで重要です。これらは、従来の衝突処理手法では最も難しいケースでもあります。メッシュ最適化ステップを取り除いた理由は、衣服のレイヤーや皮膚組織では、大きな回転が表面で起こらないので深刻な問題を引き起こさないからです。

制限されたアプリケーションあるいは、最適化ステップの省略を行う動機は、方向の制約のみで保守的な衝突応答が可能となるからです。その意味において、衣服と皮膚組織のシミュレーションで衝突のない状態を保証し、我々の事例で示す通り視覚的な不具合も最小限となります。このことは、既存手法で限定された繰り返し回数が交差状態を引き起こすとの対照的です。

それぞれの時間刻みでの最適化に加えて、完全に最適化を取り除くことが、メッシュの最適化をそれぞれの時間刻みのみで行うことを利用にし、あるいは、複数の時間刻みにまたがって負荷を分散するために要素を分割することも可能となります。計算速度と正確さを調整する別の手法もあります。エアーフレームの間の繰り返し回数を調整する方法です。高速に動く場合で、少ない繰り返し回数の場合は、交差が未解決のまま残ってしまうかもしれません、動きが遅くなつた際に常に正確に修正が行われます。

#### 4.3 ハイブリッド衝突処理

オブジェクト間の距離イプシロンを維持するためと、静的摩擦、動的摩擦の処理を行うために、エアーメッシュのプロジェクトの後に、第二の衝突処理パスを行います。ここで、プリミティブの内部のペアとエアーメッシュの内部を探索することでイプシロン距離を検出します。その後に、[Macklin ら 2014]に記載のある分離と摩擦を適用するために通常の衝突応答制約を適用します。

そのような、ハイブリッドの手法で、エラー要素の数を減らす事も可能となります。詳細な衝突メッシュの為に、エラー要素の数は極めて大きくなることがあります。この場合、我々の手法は、オブジェクトが荒い衝突処理で使われるプロジェクトの後に、正しい側にあるかどうかの大域衝突段階で用いられます。従来の衝突応答の第二のパスでは、小さな要素が貫通していないか確認します。エラー要素の数を削減するために、この手法は、任意の3Dシーンで現実的にできるかもしれません。我々は、ハイブリッド手法のテストを行っていないので、これは今後の課題の一つです。

## 5 結果

全ての例で、我々は、Intel Core i7 の 3.1 GHz の CPU のシングルコアを用いています。全ての計測結果は、表1にまとめられています。コンセプトの違いにより、既存手法と我々の新しいアプローチとのパフォーマンスの比較は難しいです。例えば、純粋な衝突検出手法と対照的に、我々の手法では衝突検出と応答を同時に行います。エアーメッシュを用いることで、タイムステップの開始時点での連続的な衝突検出は、ほぼ無料です。全てのエラー要素を繰り返し、体積が反転していないか確認するだけだからです。我々の事例では、衝突の解決に多くの時間がかかります。我々の手法の利点は、このプロセスの間に新しい衝突を絶えず確認することです。また、我々の手法は、特殊なタイプの問題を目的としており、与えられたシナリオに強く依存してパフォーマンスが異なります。

コンセプトの証明の為に、2D 版の実装を行いました。図6は、最も基本的なシーンです。長方形の領域の中に箱があります。このシーンは4つの粒子を含み、長方形の5つの辺の制約と8つのエアーフレームがあります。与えられた制約で粒子のシミュレーションを行うことで、箱と境界の衝突の期待された振る舞いをさせることができます。箱が回転し過ぎるとエラー要素はロックします。この問題は、エアーフレームを有効にすることで解決できます。

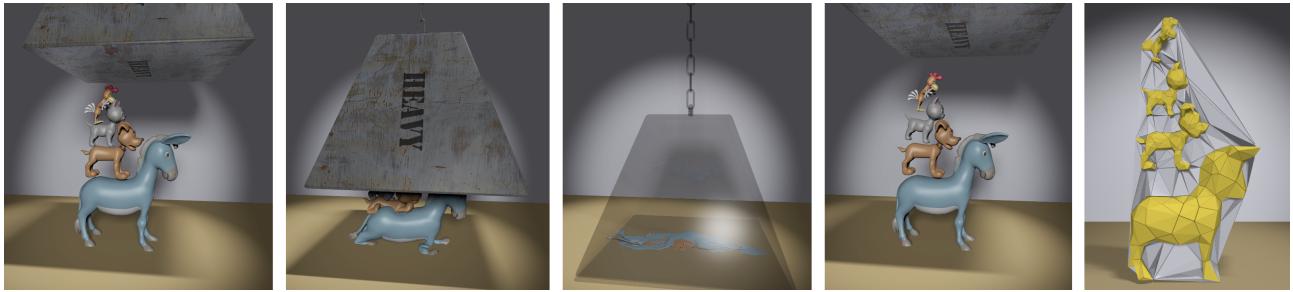
同じコンセプトは、より大きなシーンでも同様です。図2は、60個の箱とロープがあり、エラー要素の体積を0に維持しながら全てが対話的に適切に動きます。エアーメッシュがオブジェクトの相対位置のメモリを持つ事を示すために、全てのY座標の位置を0に設定します。この拘束を解くことで、オブジェクトは形状とその順番を復元します。このシーンは150fps以上で動きます。

図3の3番目のシーンは、より複雑な形状でもこのアプローチが上手く働くことを示しています。再び、全シーンを圧縮した後にオブジェクトが跳ね上がりロープの正しい側にあることがわかります。2D メッシュの最適化は単純なので、全てのシーンは 5ms 以下の計算時間しかかかりません。

2D 手法がうまくいくことを確かめた後に、3D のソルバも同様に書きはじめました。図5で、複数の箱のシーンを 3D で同様に表現したものを示します。再び、全てのエアーフレームの体積を正に保つことで正確な振る舞いを得ることが出来ました。重力を適用した後に、箱は床に落ちて振動することなく安定して自立します。ランダムな力を加えた時、箱は3Dのメッシュ最適化によって自由に動きます。予想通り、3Dの場合にはメッシュ最適化により時間がかかり 80 ms ほどかかります。これは、シミュレーションのボトルネックになります。このような特定なシナリオでは、従来通りの剛体ソルバの方が良い選択肢です。

図8 のブレーメンの音楽隊のシーンでは、4フレームおきのメッシュ最適化のみで目立った不具合もなく処理が行えています。持ちられているシミュレーションメッシュは図8の右端の図で示されています。

最終的に我々は手法のテストをさらに 2 つの現実的なモデルで行っています。図9で示される相撲取りと図1で示されるダンサーです。



**図8:** ドイツのお伽話、「ブレーメンの音楽隊」のシーン。4匹の動物は完全にペちゃんこに潰れます。全ての動物の形状が復元されるのに加えて、空間的な順序もおもりが上がった後に保存されています。背後にある、シミュレーション用メッシュが右端で示されています。オブジェクトの衝突用メッシュは黄色で、エアーメッシュはグレーで示されています。



**Figure 9:** 図9:白で示されるローカルエアーメッシュで衝突を解決しています。2番目と4番目のイメージは、エアーメッシュを無効にした状態を示しています。未解決の衝突判定が、脇の下と腿と腹の間に明らかにみられます。エアーメッシュを有効にすることで、衝突がズムーズかつ安定的に解決されます。

Scene	object elements	air elements	object sim	iters	air sim	iters	air optimization	total
2D boxes	120	416	3.3 ms	20	1.6 ms	20	1.6 ms	6.5 ms
2D objects	269	165	3.4 ms	20	0.5 ms	20	0.9 ms	4.8 ms
3D boxes	250	2775	3.5 ms	10	1.7 ms	10	80 ms	85.2 ms
3D characters	1919	2739	6.5 ms	10	2.1 ms	10	14 ms	22.6 ms
sumo fighter	32,000	1700	35 ms	1	15 ms	10	-	50 ms
dancer	47,000	165,000	119 ms	10	520 ms	20	-	639 ms

**表1:**表1:我々のシーン例のサイズと時間計測結果です。最初の2つの列は、それぞれオブジェクト要素の数(三角形あるいは四面体)とエアーメッシュの数を意味しています。次の4つの列は、オブジェクトのシミュレーションによってかかった時間です。オブジェクトのイテレーション回数とエアーメッシュのシミュレーションにかかった時間とエアーメッシュのイテレーション回数です。最終的に、エアーメッシュの最適化にかかった時間を得ました。最後の2つのシーンは、最適化をオフにした値です。



図10: 我々の手法は、複雑に絡み合った状態をスムーズに解決可能です。左:エアーメッシュを無効に設定した後の配置。中と右:スカートが滑らかに絡まりを解いた状態



図11: 図1のダンサーのシーンで用いられたエアーメッシュ

どちらのケースでも、我々の手法はメッシュ最適化を無効にしても、もっともらしい結果を得られています。

腋の下の皮膚組織の衝突を扱うのは、挑戦的な課題です。図9の左端で示されるように小さなエアーメッシュを加える事でこの問題を解決しました。3番目との4番目のイメージはエアーメッシュを無効にした状態のものです。これらを有効にすることで衝突は滑らかに解決できます。数値と制約の順序は固定されたままでし衝突セットのおかげでジッターはありません。腿と腰の間に2つのエアーメッシュを追加することで衝突を解決します。エアーメッシュはとても小さいので衝突の処理にかかる時間はほとんど無視できるものです。

図1では、複数レイヤーのスカートを身につけたダンサーをシミュレーションするために、47000の衣服のメッシュと人物の足の頂点を含む165000のエアーメッシュ四面体を用いています。図11は、結果のメッシュを示しています。シーン全体は、毎秒2フレームでまだ動いています。図10と付属のビデオでは、イテレーション回数を0にした時に何が起こるのかを示しています。人物の衝突もエアーメッシュで処理されているので、スカートは足から抜け落ちてしまい、レイヤーはそれぞれ揺れ落ちてしまいます。エアーメッシュを有効にした後に、スカートは正しい状態へスムーズに修正されます。

## 6 結論と今後の作業

我々は、エアーメッシュの概念を紹介してきました。エラー要素の反転を防ぐことで、堅牢な衝突検出と衝突応答が可能となります。我々の手法の重要な機能は、エアーメッシュがオブジェクトの相対位置の情報を保持しており、そのことが絡みあった状態から正しい状態への復元を可能とすることです。それゆえ、この手法は、皮膚組織のレイヤーや複数レイヤーの衣服などのような近接した複雑なオブジェクトに適しています。

我々は、典型的な自由に動く剛体のシーンにも応用できる事を示しました。2Dでは、この手法は堅牢で高速で実装も簡単です。したがってモバイル2Dゲームの物理効果として魅力的な手法となります。3Dでは、比較的高価なメッシュ最適化操作のため、伝統的な衝突検出手法は、従来の剛体シーンではより適した手法になります。幸運なことに、顕著な視覚的不具合無しでメッシュ最適化処理は無効にすることも可能です。

今後の作業として様々な可能性があります。例えば、[Klingner と Shewchuk 2007] によって提案された頂点挿入を実装したエアーパーティクルがあげられます。別のアイデアとしては、ハイブリッド手法の大域段階で必要に応じて動的にエラー要素を生成した後削除したりすることもできるでしょう。

我々の手法を並列化することも可能です。実のところ、すでに GPU で高速化された様々な素材のシミュレーションのためのフレームワークをすでに持っていて、それは位置ベース物理にもとづいています。我々がやるべきことは、単一方向体積制約という制約を加えるだけです。すでにそのような統合が作業中です。

## 7 謝辞

NVIDIA と PhysX のチームのサポートに感謝します。そして、匿名の査読者と彼らの価値のあるフィードバックにも感謝します。

## 参考文献

- AGARWAL, P. K., BASCH, J., GUIBAS, L. J., HERSHBERGER, J., AND ZHANG, L. 2000. Deformable free space tilings for kinetic collision detection. In *International Journal of Robotics Research*, 83–96.
- ALLARD, J., FAURE, F., COURTECUISSE, H., FALIPOU, F., DURIEZ, C., AND KRY, P. G. 2010. Volume contact constraints at arbitrary resolution. *ACM Trans. Graph.* 29, 4 (July), 82:1–82:10.
- BARAFF, D., WITKIN, A., AND KASS, M. 2003. Untangling cloth. *ACM Trans. Graph.* 22, 3 (July), 862–870.
- BARBIĆ, J., AND JAMES, D. L. 2010. Subspace self-collision culling. *ACM Trans. Graph.* 29, 4 (July), 81:1–81:9.
- BRIDSON, R., FEDIKIW, R., AND ANDERSON, J. 2002. Robust treatment of collisions, contact and friction for cloth animation. *ACM Trans. Graph.* 21, 3 (July), 594–603.
- BROCHU, T., EDWARDS, E., AND BRIDSON, R. 2012. Efficient geometrically exact continuous collision detection. *ACM Trans. Graph.* 31, 4 (July), 96:1–96:7.
- CIVIT-FLORES, O., AND SUSN, A. 2014. Robust treatment of degenerate elements in interactive corotational fem simulations. *Computer Graphics Forum* 33, 6, 298–309.
- CURTIS, S., TAMSTORF, R., AND MANOCHA, D. 2008. Fast collision detection for deformable models using representative-triangles. In *Proceedings of the 2008 symposium on Interactive 3D graphics and games*, ACM, New York, NY, USA, I3D ’08, 61–69.
- FAURE, F., BARBIER, S., ALLARD, J., AND FALIPOU, F. 2008. Image-based collision detection and response between arbitrary volume objects. In *Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, SCA ’08*, 155–162.

- FREITAG, L. A., AND OLLIVIER-GOOCH, C. 1997. Tetrahedral mesh improvement using swapping and smoothing. *International Journal for Numerical Methods in Engineering* 40, 21, 3979–4002.
- GOVINDARAJU, N. K., REDON, S., LIN, M. C., AND MANOCHA, D. 2003. Cullide: Interactive collision detection between complex models in large environments using graphics hardware. In *Proceedings of the ACM SIGGRAPH/EUROGRAPHICS Conference on Graphics Hardware*, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, HWWS '03, 25–32.
- GOVINDARAJU, N. K., KNOTT, D., JAIN, N., KABUL, I., TAM-STORF, R., GAYLE, R., LIN, M. C., AND MANOCHA, D. 2005. Interactive collision detection between deformable models using chromatic decomposition. *ACM Trans. Graph.* 24, 3 (July), 991–999.
- GRIMM, J., AND GRIMM, W. 1812. Town musicians of bremen. In *Grimm's Fairy Tales*, vol. KHM 27.
- HEIDELBERGER, B., TESCHNER, M., AND GROSS, M. 2004. Detection of collisions and self-collisions using image-space techniques. In *Proceedings of Computer Graphics, Visualization and Computer Vision*, 145–152.
- HUTTER, M., AND FUHRMANN, A. 2007. Optimized continuous collision detection for deformable triangle meshes. *Journal of WSCG* 15, 1-3, 25–32.
- IRVING, G., TERAN, J., AND FEDKIW, R. 2004. Invertible finite elements for robust simulation of large deformation. *Proc. of the ACM SIGGRAPH/Eurographics Symp. on Comput. Anim.*, 131–140.
- IRVING, G., SCHROEDER, C., AND FEDKIW, R. 2007. Volume conserving finite element simulations of deformable models. *ACM Trans. Graph.* 26, 3.
- KLINGNER, B. M., AND SHEWCHUK, J. R. 2007. Aggressive tetrahedral mesh improvement. In *Proceedings of the 16th International Meshing Roundtable*, 3–23.
- LEVIN, D. I. W., LITVEN, J., JONES, G. L., SUEDA, S., AND PAI, D. K. 2011. Eulerian solid simulation with contact. In *ACM SIGGRAPH 2011 Papers*, ACM, New York, NY, USA, SIGGRAPH '11, 36:1–36:10.
- MACKLIN, M., MÜLLER, M., CHENTANEZ, N., AND KIM, T.-Y. 2014. Unified particle physics for real-time applications. *ACM Trans. Graph.* 33, 4 (July), 153:1–153:12.
- MEZGER, J., KIMMERLE, S., AND ETZMUS, O. 2003. Hierarchical techniques in collision detection for cloth animation. *Journal of WSCG* 11, 1-3.
- MÜLLER, M., AND GROSS, M. 2004. Interactive virtual materials. In *Proceedings of Graphics Interface 2004*, Canadian Human-Computer Communications Society, School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, GI '04, 239–246.
- MÜLLER, M., HEIDELBERGER, B., TESCHNER, M., AND GROSS, M. 2005. Meshless deformations based on shape matching. In *ACM SIGGRAPH 2005 Papers*, ACM, New York, NY, USA, SIGGRAPH '05, 471–478.
- MÜLLER, M., HEIDELBERGER, B., HENNIX, M., AND RATCLIFF, J. 2007. Position based dynamics. *J. Vis. Comun. Image Represent.* 18, 2 (Apr.), 109–118.
- PAGANO, S., AND ALART, P. 2008. Self-contact and fictitious domain using a difference convex approach. *International Journal for Numerical Methods in Engineering* 75, 1 (July), 29–42.
- PREZ-URBIOLA, R. E., AND RUDOMIN, G. I. 1999. Multi-layer implicit garment models. In *Shape Modeling International Proceedings*, 66–71.
- SCHVARTZMAN, S. C., GASCÓN, J., AND OTADUY, M. A. 2009. Bounded normal trees for reduced deformations of triangulated surfaces. In *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '09, 75–82.
- SCHVARTZMAN, S. C., PÉREZ, L. G., AND OTADUY, M. A. 2010. Star-contours for efficient hierarchical self-collision detection. *ACM Trans. on Graphics (Proc. of ACM SIGGRAPH)* 29, 3.
- SELLE, A., SU, J., IRVING, G., AND FEDKIW, R. 2008. Robust high-resolution cloth using parallelism, history-based collisions and accurate friction. *IEEE Transactions on Visualization and Computer Graphics*, (TVCG) 15, 2, 339 – 350.
- SHEWCHUK, J. R. 2002. Two discrete optimization algorithms for the topological improvement of tetrahedral meshes. *Manuscript*.
- SI, H. 2015. Tetgen, a delaunay-based quality tetrahedral mesh generator. *ACM Trans. Math. Softw.* 41, 2 (Feb.), 11:1–11:36.
- SIFAKIS, E., MARINO, S., AND TERAN, J. 2008. Globally coupled collision handling using volume preserving impulses. In *Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, SCA '08, 147–153.
- STOMAKHIN, A., HOWES, R., SCHROEDER, C., AND TERAN, J. M. 2012. Energetically consistent invertible elasticity. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, SCA '12, 25–32.
- TANG, M., CURTIS, S., YOON, S.-E., AND MANOCHA, D. 2008. Interactive continuous collision detection between deformable models using connectivity-based culling. In *SPM '08: Proceedings of the 2008 ACM symposium on Solid and physical modeling*, ACM, New York, NY, USA, 25–36.
- TANG, M., MANOCHA, D., AND TONG, R. 2010. Fast continuous collision detection using deforming non-penetration filters. In *Proceedings of the 2010 ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, ACM, New York, NY, USA, I3D '10, 7–13.
- TANG, M., MANOCHA, D., YOON, S.-E., DU, P., HEO, J.-P., AND TONG, R.-F. 2011. Volecd: Fast continuous collision culling between deforming volume meshes. *ACM Trans. Graph.* 30, 5 (Oct.), 111:1–111:15.
- TANG, M., TONG, R., WANG, Z., AND MANOCHA, D. 2014. Fast and exact continuous collision detection with bernstein sign classification. *ACM Trans. Graph.* 33 (November), 186:1–186:8.
- TESCHNER, M., HEIDELBERGER, B., MUELLER, M., POMERANETS, D., AND GROSS, M. 2003. Optimized spatial hashing for collision detection of deformable objects. 47–54.
- TESCHNER, M., KIMMERLE, S., HEIDELBERGER, B., ZACHMANN, G., AND RAGHUPATHI, R. 2005. Collision detection for deformable objects. *Computer Graphics Forum* 24, 1, 61–81.

- VOLINO, P., AND MAGNENAT-THALMANN, N. 2006. Resolving surface collisions through intersection contour minimization. In *ACM SIGGRAPH 2006 Papers*, SIGGRAPH '06, 1154–1159.
- WANG, B., FAURE, F., AND PAI, D. K. 2012. Adaptive image-based intersection volume. *ACM Trans. Graph.* 31, 4 (July), 97:1–97:9.
- WANG, H. 2014. Defending continuous collision detection against errors. *ACM Trans. Graph.* 33, 4 (July), 122:1–122:10.
- WICKE, M., LANKER, H., AND GROSS, M. 2006. Untangling cloth with boundaries. In *Proceedings of VMV*, 349–356.
- WICKE, M., RITCHIE, D., KLINGNER, B. M., BURKE, S., SHEWCHUK, J. R., AND O'BRIEN, J. F. 2010. Dynamic local remeshing for elastoplastic simulation. *ACM Trans. Graph.* 29, 4 (July), 49:1–49:11.
- WONG, S.-K., AND CHENG, Y.-C. 2014. Gpu-based radial view-based culling for continuous self-collision detection of deformable surfaces. *The Visual Computer*, 1–15.
- WONG, W. S.-K., BACIU, G., AND HU, J. 2004. Multi-layered deformable surfaces for virtual clothing. In *Proceedings of the ACM symposium on Virtual reality software and technology*, ACM, New York, NY, USA, VRST '04, 24–31.
- WONG, S.-K., LIU, C.-M., BACIU, G., AND YEH, C.-C. 2010. Robust continuous collision detection for deformable objects. In *Proceedings of the 17th ACM Symposium on Virtual Reality Software and Technology*, ACM, New York, NY, USA, VRST '10, 55–62.
- WONG, S.-K., LIN, W.-C., HUNG, C.-H., HUANG, Y.-J., AND LII, S.-Y. 2013. Radial view based culling for continuous self-collision detection of skeletal models. *ACM Trans. Graph.* 32, 4 (July), 114:1–114:10.
- WONG, S.-K., LIN, W.-C., WANG, Y.-S., HUNG, C.-H., AND HUANG, Y.-J. 2014. Dynamic radial view based culling for continuous self-collision detection. In *Proceedings of the 18th Meeting of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, I3D '14, 39–46.
- YOON, S.-E., KIM, Y. J., HARADA, T., KIM, Y. J., AND YOON, S.-E. 2010. Recent advances in real-time collision and proximity computations for games and simulations. In *ACM SIGGRAPH ASIA 2010 Courses*, ACM, New York, NY, USA, SA '10, 22:1–22:110.
- ZHANG, X., AND KIM, Y. 2012. Simple culling methods for continuous collision detection of deforming triangles. *IEEE Transactions on Visualization and Computer Graphics* 18, 7, 1146–1155.
- ZHENG, C., AND JAMES, D. L. 2012. Energy-based self-collision culling for arbitrary mesh deformations. *ACM Trans. Graph.* 31, 4 (July), 98:1–98:12.