

歪みベース力学

Matthias Müller Nuttapong Chentanez Tae-Yong Kim Miles Macklin

NVIDIA

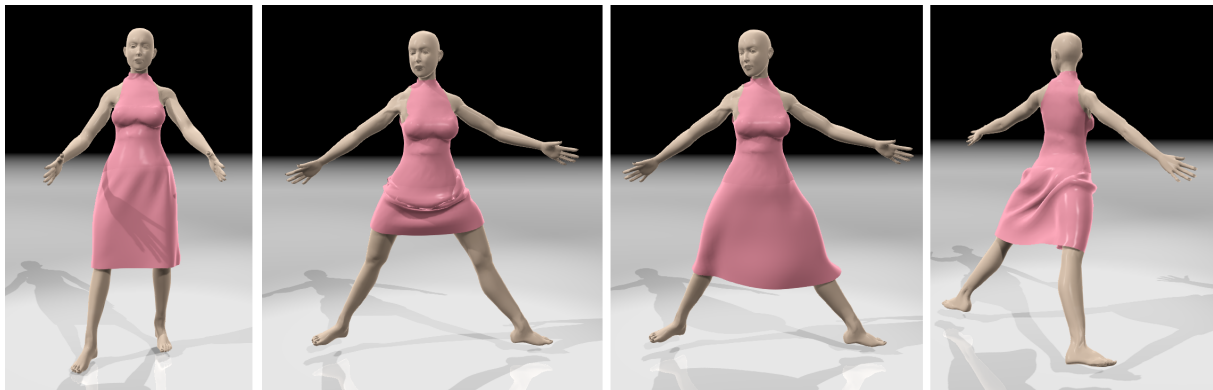


図1: 左から右: バインドポーズでのタイトなスカート。バインドポーズから極めて大きく異なる姿勢で生じる課題。全体の剛性値を下げる事でドレスのたるみを作っています。垂直方向に沿った異方性の剛性値を弱める事で課題を解決しています。

概要

我々は、位置ベース力学(PBD)フレームワークの新しい拘束手法を提案します。これは、シミュレーションするメッシュの辺の方向とは独立した方向のひずみのコントロールを可能とします。頂点の間の距離制約の代わりに、Green - St Venant(サン・ブナン)歪みテンソルの制約を用います。個別の歪み系数に対応する剛性値を変化させることで、異方性の振る舞いのシミュレーションが可能となります。

Greenの回転独立、非線形歪みテンソルを直接用いる事により、歪み制限手法を用いた変形勾配の極分解を行う必要がなくなります。加えて、歪みテンソルの対角成分に相当する制約条件を修正することを提案します。これは、シングルステップで解けますし、対応するせん断方向の抵抗から生じる引き伸ばしを分離するために非対角成分の要素の制約条件を修正します。PBDフレームワークの中で制約条件を定式化するため、それらの制約条件は、歪みの制限だけでなく、変形物体の実際のシミュレーションを行うことが出来ます。その一方、伝統的な歪み制限法は、分離したシミュレーション手法と合わせて用いる必要があります。

Categories and Subject Descriptors (according to ACM CCS): 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

1. 導入

Position Based Dynamics [MHR06] は、そのシンプルさや速度や堅牢さのためゲームや映画業界での布や柔軟体のシミュレーション手法として多く用いられるようになってきました。

従来、メッシュは辺に沿った距離や曲げ角度の制約でシミュレーションされてきました。本手法では、メッシュを構成する辺の方向に沿った歪みのみを制御します。

力ベースの手法の質点バネシステムも同様です。力ベースのフレームワークでは、有限要素法(FEM)定式化の過程で、この依存は取り除かれ、テセレーションの独立方向での堅さ値に特化されます。我々の目標は、位置ベース物理法を用い同じことを実現する制約条件を見つけることです。

辺の長さ制約の代わりに、パーティクル位置の変形から引き起こされる Green - St Venant 歪みテンソルを用います。歪みから応力を求め、変形エネルギーを積分して力を求める FEM と異なり、我々は、それぞれの歪みテンソルの制約条件から直接位置を求めます。

多くの場面で、予め決めておいた独立した三角形化の方向に沿って歪みを制御する事は有益です。例えば、布の堅さは、通常相対的に小さいせん断抵抗の場合、縦糸方向と横糸方向で異なります。多くの場合、アーティストは、バインドポーズよりも布が伸びたフレームを含むアニメーションを作りがちです。バインドポーズの範囲へとソルバで強制することは、図1.で示すような衝突処理内部で深刻な問題を引き起こします。水平方向の堅さを減らす一方で、布の伸びを引き起こす等方性の堅さを削減することでのみ、この問題をうまく解決することが出来ます。この振る舞いは、辺の制約や、縦糸と横糸の方向に沿った正規三角形化の制約を用いることで近似が可能です。しかしながら、実際のところ、衣服を着用したキャラクターは、クロスシミュレーションを考慮してデザインされることはあまりありません。その上、正規化されたメッシュの対角辺は、独立すべき歪みがまだ連結されたままなのです。

四面体メッシュを用いた皮膚層のシミュレーションは、独立した歪みのコントロールが有益であることを示すもう一つのサンプルです。ここで皮膚のスライドが、キャラクター表面に沿ったせん断抵抗を低下させることでシミュレーション可能となります。(参照図 5).

我々の手法で取り組み可能な3番目の課題を見てみましょう。これは、三角形メッシュの辺の長さを完全に制約することで、一般的な方向から曲げられることを防ぐことに言及しています。English らは [EB08] この問題を、規則に準拠する三角形を用いることで解きました。

規則に準拠しないメッシュは、衝突応答に適していないので2つのメッシュがシミュレーションのために必要になります。Goldenthal らは [GHF*07] ロックを防ぐために、正規の四角形メッシュの辺のみを制約しました。我々の手法では、せん断方向の剛性値のみを減らすことで任意の三角形メッシュで同じ効果を実現しました。

我々の主要な貢献は

- 変形オブジェクトをシミュレーションするための歪みテンソルの要素制約から PBD の制約を求めた。
- Cauchy 歪みテンソルと変形勾配の極分解を用いる代わりに Green 回転独立歪みテンソルを用いた。
- 2面角の制約を修正することでシングルステップで解くことを可能とした。

- 非2面角の制約を修正することでせん断方向の抵抗を分離しました。

2. 関連研究

Müller らが Position Based Dynamics を [MHR06]で発表してからさまざまな改善手法が提案されてきました。Kubiak ら [KPGF07] は、手術シミュレーションの糸をシュミレーションする方法として拡張しました。Müller [Mül08] は、階層化 PBD を高解像度のメッシュのシミュレーションコストを減らすための手法として提案しました。三角形メッシュのための新しい曲げモデルが Kelager ら [KNE10] によって提案され、新しい体積保存制約が Diziol ら [DBB11] によって考案されました。Kim ら [KCMF12] と Müller ら [MKC12] は、髪の毛の表現に用いる1次元のロッドの特別なケースについて研究し、伸縮しない高速な方法を提案しました。position based 手法のサーベイとして、[BMOT13] を参照しました。

歪み項をクロスシミュレーションに用いるアイデアは、力ベース手法では Baraff と Witkin の [BW98] で研究されていました。我々は、このアイデアを position based のフレームワーク内でクロスと四面体オブジェクトへと一般化して適用しました。

力ベースの歪み制限手法は PBD と関係が深いです。ここで、頂点の位置や速度は、それぞれの力ベースのシミュレーションステップの後に直接操作されます。

我々の手法と対照的に、歪み制限法は、実際のマテリアルのシミュレーションに歪み項を用いる一方、マテリアルが伸びすぎたときにのみ作動します。その一方、我々の手法は、力積ベースの設定を歪み制限と同様に用いることが可能です。

Provot [Pro95] と Bridson ら [BMF03] は、質点バネシミュレーションの文脈で、バネの長さが指定した制限値より伸びたり縮んだりしないように制限する方法を提案しました。Hong ら [HJC*05] は、大きなタイムステップの使用が可能となる陰解法の定式化を行いました。Gold-enthal ら [GHF*07] は、収束率を上げるために、標準四角形の辺の長さの上限制約を制限するグローバルソルバを用いました。

有限要素法(FEM)の文脈では、Picin-bono ら [PDA03] は、ある方向への歪みを制限するためのエネルギー項を加えることで異方性の歪みの制限を実現しました。Perez ら [PCH*13] は、ラグランジュの未定乗数を用い、歪み項の等方性の制約を実現しました。Hernandez ら [HCPO13] は、異方性マテリアルのサポートを行い、この手法を改善しました。

最も関連の深い2つの研究は Thomaszewski ら [TPS09] と Wang ら [WOR10] によるものです。したがって、これらの手法との違いを少し詳細に議論します。両研究は、単一要素を解き、我々と同様にガウス・サイデル法とヤコビ繰り返し法を用いてグローバルに解きます。

Wang ら [WOR10] は、歪みを等方性に制限しました。彼らの手法は、Tsiknis ら [Tsi06] の歪み制限手法の拡張です。Wang らは、極分解を用いて要素の変形勾配から主軸歪みの拡張を行いました。次に、彼らは主軸歪みをクランプして新しいクランプされた変形勾配を計算しました。

この新しい勾配は、要素の初期形状を目的の形状に再び変形させるのに用いられます。その一方、我々は、個々のせん断および伸び方向のために異なった位置修正ベクトルを計算します。これは、異方性の振る舞いのシミュレーションのための異なったスケーリング要素を組み合わせます。

Thomaszewski ら [TPS09] は、Cauchy の線形歪みテンソルを用い、極分解を用い変形勾配の回転要素を展開しました。線形歪みを用いると、歪み成分は、隣接頂点の速度に線形に依存します。歪み成分を特殊化することで、三角形の3x3速度成分を6x6線形系にすることができ、加えて極分解を実行することでそれぞれの要素を解くことができます。この計算は、5x5サブブロックの逆行列を事前計算することで高速化が可能です。しかしながら三角形毎に float 25 個の記憶領域が必要となります。四面体の場合、線形システムは、12x12次元となります。我々の場合では、結果が示すとおり、三角形を扱うときより酷く遅くなるわけではありません。

両方の手法においても、連結されるものは別々に扱わなくてはなりません。Thomaszewski らの手法では、線形システムは制約されすぎ、最小二乗問題を解かなくてはなりません。その一方、PBD のフレームワークでは、頂点は単純に連結さ質量の逆数を 0 に設定することで頂点は単純に連結されます。

3. 手法

我々は、既存の距離拘束を置き換える新しい歪みベースの位置拘束を導入しました。完全さのために、PBD の基本概念を完結に復習したいと思います。

3.1. PBDの基礎

N 個のパーティクルがあり、位置 \mathbf{x}_i 速度 \mathbf{v}_i 質量の逆数 w_i を持つと仮定します。主要な PBD のシミュレーションステップは下記ようになります。

```
initialize  $\mathbf{x}_i$  and  $\mathbf{v}_i$ 
while simulating do
     $\mathbf{v}_i \leftarrow \mathbf{v}_i + \Delta t \mathbf{f}_i$ 
     $\mathbf{p}_i \leftarrow \mathbf{x}_i + \Delta t \mathbf{v}_i$ 
     $\mathbf{p}_i \leftarrow \text{solve}(\mathbf{p}_i)$ 
     $\mathbf{v}_i \leftarrow (\mathbf{p}_i - \mathbf{x}_i) / \Delta t$ 
     $\mathbf{x}_i \leftarrow \mathbf{p}_i$ 
end
```

シミュレーションループの中で、速度を更新した後に、予想位置 \mathbf{p}_i は陽的オイラーステップを用いて計算されます。これらの位置の制約 C_j に適合するソルバでこれらの位置は修正されます。位置の拘束条件は、スカラー関数 $C(\mathbf{p}_1, \dots, \mathbf{p}_N)$ によって定義されます。このスカラー関数は、制約が満たされた場合に0になります。非線形方程式を解く手法の一つであるガウスサイデル(Gauss-Seidel)法で全ての拘束条件を複数回繰り返して解きます。

単一の拘束条件 C_i 位置 \mathbf{p}_i の修正値 $\Delta \mathbf{p}_i$ は、下記のように計算されます。

$$\Delta \mathbf{p}_i = -s k w_i \frac{\partial}{\partial \mathbf{p}_i} C(\mathbf{p}_1, \dots, \mathbf{p}_N), \quad (1)$$

ここで

$$s = \frac{C(\mathbf{p}_1, \dots, \mathbf{p}_N)}{\sum_j w_j \left| \frac{\partial}{\partial \mathbf{p}_i} C(\mathbf{p}_1, \dots, \mathbf{p}_N) \right|^2} \quad (2)$$

剛性値パラメータ $k \in [0, 1]$ とします。係数 k は、物理的に正確な値ではありません。なぜなら、この効果がタイムステップの大きさと繰り返し回数に依存していたからです。しかしながら、ゲームでは、どちらも定数であることが多く、 k は、直感的に編集することが可能です。

上記方程式は、現在のパーティクル状態に基づく拘束関数のローカル線形化から求められます。PBD を堅牢にしているのは、線形化が、それぞれの制約のプロジェクトの前に更新されることで、グローバルなソルバで固定されないことにあります。この計算の後に、速度と位置は、修正された予測位置に基づいて更新されます。

例を挙げると、 $C(\mathbf{x}_1, \mathbf{x}_2) = |\mathbf{x}_1 - \mathbf{x}_2| - d$ は、パーティクル1と2の距離を d に拘束します。この拘束条件関数を式(1)と式(2)に代入することで修正値は下記ようになります。

$$\Delta \mathbf{p}_1 = -\frac{w_1}{w_1 + w_2} (|\mathbf{p}_1 - \mathbf{p}_2| - d) \frac{\mathbf{p}_1 - \mathbf{p}_2}{|\mathbf{p}_1 - \mathbf{p}_2|} \quad (3)$$

$$\Delta \mathbf{p}_2 = +\frac{w_2}{w_1 + w_2} (|\mathbf{p}_1 - \mathbf{p}_2| - d) \frac{\mathbf{p}_1 - \mathbf{p}_2}{|\mathbf{p}_1 - \mathbf{p}_2|} \quad (4)$$

3.2. 四面体拘束

ここから新しい拘束条件に取り掛かります。我々の基本アイデアは、三角形と、四面体のメッシュの両方に適用可能です。最初に、単一の四面体を考えましょう。パーティクルのペアごとの距離を辺で制御する拘束条件の代わりに、4頂点全てを含む制約を導入しました。そして、与えられた値を想定するグリーン(Green)の歪テンソル項を用いるようにしました。これらの制約条件を定式化するために、四面体に隣接する4つのパーティクルの位置に基づく歪み項の表現が必要となりました。

$\mathbf{q}_0, \mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3$ をマテリアル座標とし、 $\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3$ を対応するワールド座標とします。平行移動は歪みに影響しないため、 \mathbf{q}_0 と \mathbf{p}_0 を 0 とします。そうすると下記を用いることで

$$\mathbf{P} = [\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3] \text{ and} \quad (5)$$

$$\mathbf{Q} = [\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3] \quad (6)$$

我々は、変形勾配を下記のように表現できます。

$$\mathbf{F} = \mathbf{P} \mathbf{Q}^{-1} \quad (7)$$

そして、Green - St Venant 歪みテンソルは

$$\mathbf{G} = \mathbf{F}^T \mathbf{F} - \mathbf{I} \quad (8)$$

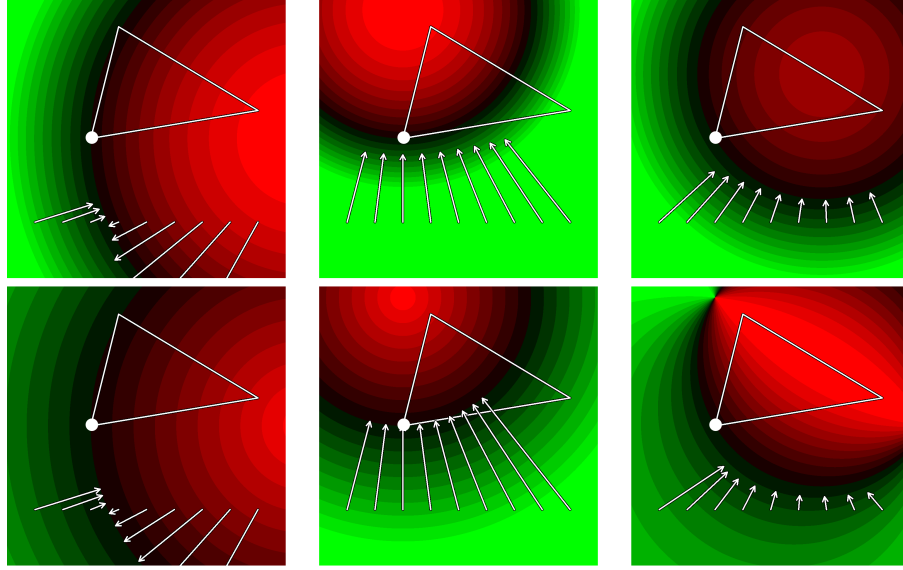


図 2 拘束ベクトルと他の接続されている δ つの頂点がある平面へと向かう白い頂点への投影ベクトル。3列は、 x, y , 剪断方向の歪みです。(緑は正の値、赤が負の値でクランプされています。)上段では、スタンダードな歪み関数 $S_{11-1}, S_{22-1}, S_{12}$ を用いました。下段は、修正された拘束関数 $\sqrt{S-1}, \sqrt{S-1}, S/|f||f|$ を用いました。ここで留意すべきは、零点集合は変化しないことです。上段に比較して、 x -と y -方向の拘束は、1ステップで解かれています。なぜならプロジェクション方向に沿って勾配は定数だからです。修正されたせん断関数は、プロジェクションベクトルをせん断をストレッチ抗力から分解します。

ここで \mathbf{I} は、恒等行列とする。Green の歪みテンソルのオリジナルの定義から係数を $\frac{1}{2}$ に落とします。なぜなら我々の拘束の定式化でキャンセルされるからです。行列 \mathbf{Q}^{-1} は、定数ですので事前計算されます。

\mathbf{G}_{ii} の対角要素は、伸びを表していて、 $\mathbf{G}_{ij} = \mathbf{G}_{ji}$ の非対角要素は、マテリアルフレームの主方向に関するせん断を表しています。

ここで、3つの伸び、せん断の拘束関数を示します。

$$C(\mathbf{p0}, \mathbf{p1}, \mathbf{p2}, \mathbf{p3}) = S_{ii} - s_i^2 \quad (9)$$

$$C(\mathbf{p0}, \mathbf{p1}, \mathbf{p2}, \mathbf{p3}) = S_{ij} \quad i < j, \quad (10)$$

ここで、 $\mathbf{S} = \mathbf{F}^T \mathbf{F}$ で S_i は、定常の長さで通常は1になります。これらの制約は、パーティクルをストレッチがなく、せん断もない状態へと導きます。それぞれの制約で分離された、堅さ係数 k_{ij} を用いることで、異方性の素材のシミュレーションが可能となります。ここで留意すべきなのは、多くの歪み制限手法のように極分解で推定する必要がないことです。なぜなら、回転独立は、グリーンの歪みテンソルの定義から言えるからです。

3.3. 三角形の拘束条件

三角形に該当する行列 \mathbf{P} と \mathbf{Q} は正方行列ではありません。なぜならパーティクルの数は1つに削減されていますが、パーティクル位置の次元は同じです。したがって \mathbf{Q}^{-1} は定義できません。この問題は、三角形メッシュの2次元テクスチャー座標で三角形の定常状態を定義し自然な状態を選ぶことで解く事が出来ます。

適切に異方性クロスシミュレーションを行うためテクスチャー座標はクロスの縦糸と横糸に沿っている必要があります。 \mathbf{Q} は、 2×2 行列になり、 \mathbf{Q}^{-1} も上手く定義できました。定義は以下のようになります。

$$\mathbf{S} = \mathbf{Q}^{-1T} \mathbf{P}^T \mathbf{P} \mathbf{Q}^{-1} \quad (11)$$

これは、 $\mathbf{S}, \mathbf{Q} \in \mathbb{R}^{2 \times 2}$ と $\mathbf{P} \in \mathbb{R}^{3 \times 2}$ でも有効です。ここで留意すべきは、四面体および三角形の歪み制約は 3.7. 節で述べられる考えでは値がなくなることです。

3.4. 歪みの平方根

拘束条件の自然な定義は、多くありますが、式 (9) で示すストレッチ拘束をより安定に定式化する手法があります。これを理解するために、2点の間の長さ d を持つシンプルな距離拘束を見てみましょう。この2つの拘束関数は、下記の通りです。

$$C(\mathbf{p1}, \mathbf{p2}) = |\mathbf{p1} - \mathbf{p2}| - d \quad (12)$$

$$C(\mathbf{p1}, \mathbf{p2}) = |\mathbf{p1} - \mathbf{p2}|^2 - d^2 \quad (13)$$

これらは、どちらも有効です。しかしながら前者は $\mathbf{p1} - \mathbf{p2}$ に沿った線形関数に対して後者は異なります。これは、PBD の拘束を線形でプロジェクションする事で、前者の拘束を1度で完全に解ける事を示しています。これは、該当する我々のストレッチ手法の後者の拘束関数では成り立ちません。

Matthias Müller Nuttapong Chentanez Tae-Yong Kim Miles Macklin NVIDIA / Strain Based Dynamics

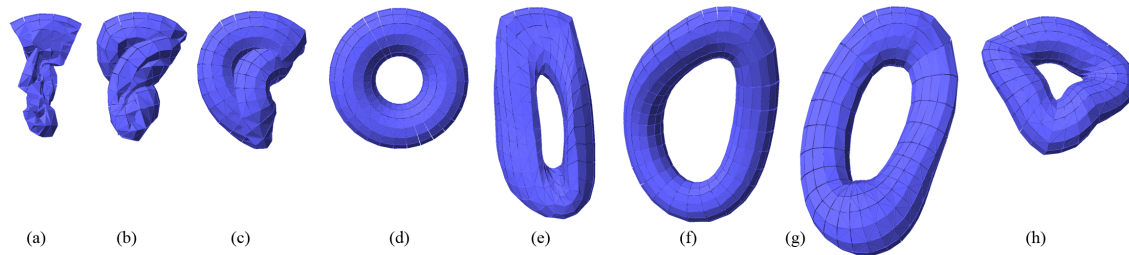


図4: 柔軟体の堅さパラメータの変化 図 (a) - (d) は、大きく絡み合った状態のトーラスを体積の剛性値を増加させることリカバリーさせる様子を示しています。(e) で、体積保存の剛性値を除いて全てを削減しました。その結果、トーラスは大きく変形しますが、体積は保存されています。図 (f) は、体積拘束の剛性値とトーラスの主軸に沿った剛性値のみ弱めたものです。強いせん断拘束と、低い歪み抵抗の結果が、図 (g) で示されます。ここで角度の歪みは小さい一方形状は伸びています。図 (h) は、逆の設定を示しています。ここで、伸びは小さい一方、トーラスは強く曲がっています。

この問題は、式 (9) を下記のように置換することで解く事が可能です。

$$C(\mathbf{p0}, \mathbf{p1}, \mathbf{p2}, \mathbf{p3}) = \sqrt{\mathbf{S}_{ii}} - s_i. \quad (14)$$

この変更で、要素のストレッチ拘束を1プロジェクトステップで正確に解くことが可能となります。なぜなら、拘束関数の勾配は、定数で、図2 で示される2つのカラムの最初のイメージの下のようにプロジェクト方向に沿って定数だからです。提案された拘束条件に基づいた位置修正値の導出は付録を確認下さい。この修正により、相対的に同じ残存ストレッチ値でのソルバの繰り返し回数は、平均で25%削減できました。より重要なことは、シミュレーションの安定性はオーバーシュootingの回避により向上したことです。

3.5. ストレッチからのせん断の分解

せん断拘束関数 \mathbf{S}_{ij} は、 $\mathbf{S}_{ij} = \mathbf{f}_i \cdot \mathbf{f}_j$ と書くことが出来ます。ここで \mathbf{f}_i と \mathbf{f}_j は、 i^{th} と j^{th} の \mathbf{F} の縦列ベクトルです。しかしながら、この関数は、変形した座標系の軸の間の角度だけ、(カラムベクトルの内積) をペナルティするだけでなく、主軸のストレッチもまたペナルティします。(例:カラムベクトルの大きさ) ストレッチから歪みを分解するために、下記修正せん断制約を提案します。

$$\tilde{\mathbf{S}}_{ij} = \frac{\mathbf{f}_i \cdot \mathbf{f}_j}{|\mathbf{f}_i| |\mathbf{f}_j|} = \frac{1}{|\mathbf{f}_i| |\mathbf{f}_j|} \mathbf{S}_{ij}. \quad (15)$$

修正制約関数と該当する PBD のプロジェクトは、図2の最後の縦列で示されます。せん断拘束は、プロジェクト方向に沿って非線形なのでオーバーシュootingがありえます。安全な側にするために、せん断拘束を図2で行ったように削減することが可能です。せん断拘束にもかかわらず、我々の実験では、不安定さは見られませんでした。

© The Eurographics Association 2014.

3.6. 一般せん断方向

式 (9) と式 (10) は、グローバル座標に沿った歪みに制約します。あるケースでは、これは望ましくありません。スキンスミュレーションのためのキャラクターの表面上に四面体のレイヤーがあることを想定させてください。そして、スキンは、サーフェイスの接線方向には簡単にスライドするが法線方向にはスライドしないことを仮定させてください。この問題に関してシンプルでエレガントな解決方法があります。式を修正する必要さえありません。ただ、四面体の定常時の形状を修正するだけで良いのです。事前計算状態で、四面体毎にローカルなフレームが計算されます。上記例では、このフレームは四面体の位置のサーフェイス接線および法線ベクトルによって作られます。

このローカルフレームに関して、 $\mathbf{q}_0, \mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3$ はシンプルに格納されます。

3.7. 体積と面積保存

上記拘束のどれも、四面体や三角形自体の体積を制御しません。全ての拘束が同時に満たされる場合に限り、 $\mathbf{G} = \mathbf{I}$ となり、その結果 $\det(\mathbf{F}) = 1$ となります。すなわち、体積が保存されます。しかし体積保護を独立して制御することが重要な事がよくあります。インスタンスのためにきちんと体積保存される柔らかい素材のシミュレーションが可能となります。分離された、体積/面積保存制約を追加することはPBDフレームワーク内で率直な実装が可能です。シンプルに下記のように定義しました。

$$C_{\text{volume}}(\mathbf{p0}, \mathbf{p1}, \mathbf{p2}, \mathbf{p3}) = \det(\mathbf{F}) - 1 \text{ and} \quad (16)$$

$$C_{\text{area}}(\mathbf{p0}, \mathbf{p1}, \mathbf{p2}) = \det(\mathbf{F}) - 1 \quad (17)$$

3Dの物体と2Dのクロスのそれぞれで、最初の制約が[MHR06]で提案されている体積保存項に該当します。定常ストレッチで一つ以外のものと互換性を持たせるため、これらの式は下記の通り一般化されます。

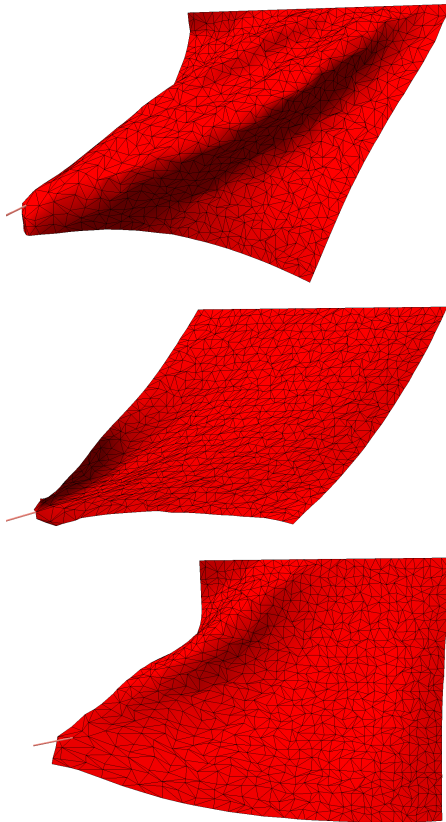


図3: 異なった歪み項でクロスの剛性値を変化させたもの。上から下へ、xストレッチ、yストレッチへ、せん断の抵抗値は(高、高、高)、(高、高、低)、(低、高、高)

我々の手法は、これらのモードを、非正規なテセレーションをされた三角形メッシュ上で独立して制御することを可能とします。

$$C_{\text{volume}}(\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3) = \det(\mathbf{F}) - s_1 s_2 s_3 \quad (18)$$

$$C_{\text{area}}(\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2) = \det(\mathbf{F}) - s_1 s_2. \quad (19)$$

この拘束の別の重要な機能は、下記の場合四面体の要素の反転を扱えることです。

$$\det(\mathbf{F}) = 1 \quad (20)$$

$$\det(\mathbf{P}) \det(\mathbf{Q}^{-1}) = 1 \quad (21)$$

$$\det(\mathbf{P}) = \det(\mathbf{Q}) \quad (22)$$

上記式は、四面体の符号付き体積は、その符号付き定常体積と適合しなければならない事を示しています。付録内で、この拘束の定式化の詳細まで述べています。体積が反転する場合や頂点が正しい向きの面へと射影される事などです。

要素の反転は、FEMベースの手法でも一般的な問題です。



図5: エイリアン牛の皮膚のスライド。上のイメージは、サーフェスを不規則に引っ張った線形ブレンドスキニングによって作られたスキンの変形です。我々の手法は、小さい接線せん断抵抗を用い正確な皮膚のスライドを実現できるようなサーフェス上の四面体レイヤーのシミュレーションを可能とします。(下図)

力は、一般に歪みテンソルからのみ導かれるからです。様々な論文例えば [TTF04] などは、この問題のみに集中しています。我々の体積保存項は、図 4 で示される通りシンプルで効果的な解決方法です。

しかしながら、体積保存のために追加の拘束を加えることで、ジッターリングを引き起こします。過剰にシステムを拘束してしまうからです。これは、歪みの堅さと体積拘束が 1 に近づいたときに過剰な変形を引き起こします。実験中にこの問題を見つけ、体積拘束を少しだけ緩めることでこの問題を解決しました。

3.8. 曲げ

クロスシミュレーションでは、歪みは内在する大きさなので歪みベースの制約は外部的には曲げに影響しません。したがって曲げは、分離して扱うことが出来ます。

曲げの抵抗をシミュレーションするために [MHR06]で提案されている隣接三角形のペアの 2 面角の拘束を用います。しかしながら、パーティクルの位置に関する、

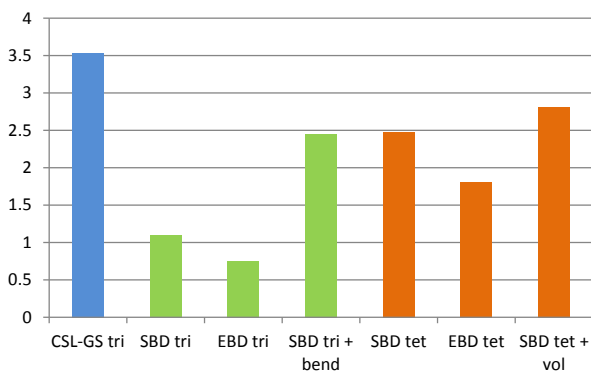


図6: 3600要素以上に対して1イテレーションにかかる時間(ミリ秒)、ここで CSL-GS は、ガウスサイデルイテレーションの連続体ベースの歪み制限を意味しています。SBD は、歪みベース運動で EBD はエッジベースの運動です。

Bridson ら [BMF03] で記述されている三角形ペアの曲げモードと呼ばれる2面角の導関数のために単純化された式を用います。このモードは、この論文では述べられていないですが、スケーリングを含む曲げ角度の導関数と一対一に対応しています。

符号だけ、付録で示すように三角形のペアの向きに応じて反転する必要があります。

3.9. ダンピング

ダンピングのために、一般的な PBD の式を用います。これは、位置修正量 $\Delta \mathbf{p}_1, \dots, \Delta \mathbf{p}_N$ を下記式のように拘束する特別な値で相対速度をダンピングするものです。

$$\mathbf{v}_i \leftarrow \mathbf{v}_i - k \left(\sum_{j=1}^N \mathbf{v}_j^T \mathbf{n}_j \right) \mathbf{n}_i \quad (23)$$

ここで $k \in [0, 1]$ で、該当する剛性値は $n_i = \Delta p_i / |\Delta p_i|$ です

のみに適用されることです。いかなる人工的なダンピングが適用されるわけではありません。モード自体が変わらない場合は、剛体の運動に合計としてとり消される形となるからです。

4. 結果

図3で示される実験では、クロスを構成する長方形の中に非正規の三角形化が施されたものを用いています。x-ストレッチ、y-ストレッチ、せん断方向それぞれに剛さのパラメータを修正しました。まったく整っていないメッシュであろうと、クロスは期待される見た目になります。

図4は、体積のある、ローカル軸に沿った要素を持つ体積のあるトーラスの四面体のケースを示しています。反転した要素の修正のためのボリューム項の有用性を示すため、図4(a)で示される酷く裏返った状態から始めました。最初の4つのイメージが示すように、形状は2,3回のイテレーションで修正されます。

異なったモードの剛さパラメータが、興味深いボリューム効果を生み出します。図4(c) では、体積保存の剛性値以外を削減しました。

体積の剛性値と、主軸に沿った剛性値のみを削減することで図4(f)で示される挙動を作ることが出来ました。図4(g)で示される高せん断と低ストレッチ抵抗設定の場合、辺の間の角度よりもより辺の長さが変形することがわかります。逆の設定の図4(h)では、トーラスは大きく曲がっていますが、伸びは少ないです。

図6は、3GHz の Core i7 CPU を用いた計測情報です。Wang ら [WOR10] の論文から歪み制限だけのパフォーマンス値を抽出することは出来ません。なぜなら筆者らは、歪み制限とシミュレーションの両方の時間計測のみを行っているからです。Thomaszewski ら [TPS09] によって出されたパフォーマンス値から、3600トライアングルで歪み制限の1イテレーションの5.3ミリ秒の時間を抽出することが出来ます。2GHz CPU が用いられたということなので 1.5 で除算したものです。本提案手法は、連続体ベースの歪み制限(CSL)よりも3倍以上高速です。そして Edge Based Dynamics (EBD)よりも30%ほど遅いです。

四面体に関しては、EBD に関して我々の手法の速度低下は少し大きくなります。

両手法とも、拘束の数は3から6に増加しますが、拘束毎の頂点数の増加は SBD の場合3から4で、その一方 EBD の場合は2のままです。Thomaszewski et らによって分析されていないけれども、四面体に関する CSL は、要素ごとの線形システムが 12x12 に増加する場合で少し遅くなると考えています

クロスシミュレーションの場合の共通の問題は、バインドポーズにおいてクロスがよくフィットしますが、そこからポーズが大きく異なる場合に問題が生じます。

一般に剛さを下げると、伸びたクロスになります。図1で示す結果のように、垂直方向にだけ剛さを下げる事のみこの問題を解決することが出来ます。

図5で示すキャラクターの皮膚層のシミュレーションでもこの手法を用いることが出来ます。与えられた例では、スキンのスライドは、線形ブレンドスキニングの非均一な伸びを劇的に削減します。

5. 結論

PBD フレームワークでの、変形オブジェクトのシミュレーションのための新しい方法を提案しました。辺の距離拘束の代わりに、Green - St Venant 歪みテンソルのエントリに該当する変形モード毎の位置のプロジェクト集合を導きました。プロジェクトをより堅牢に、ストレッチからせん断を分離する拘束条件の修正方法を提案しました。また、体積保存についても議論し、曲げとダンピングについても議論しました。

我々の定式化は、ゲームでよく使われる PBD と、リアルタイムアプリケーションでは、計算コストが高すぎる連続体ベースの FEM ベースの手法のギャップを埋めるステップです。

我々のフレームワークと、付録で示される位置修正の陽解法は、比較的シンプルですが、この仕事は、ゲーム業界や映画業界で広く採用されるに十分実用的です。

References

- [BMF03] BRIDSON R., MARINO S., FEDKIW R.: Simulation of clothing with folds and wrinkles. In *ACM SIGGRAPH Symposium on Computer Animation* (2003), pp. 28–36. 2, 7, 9
- [BMOT13] BENDER J., MÜLLER M., OTADUY M. A., TESCHNER M.: Position-based methods for the simulation of solid objects in computer graphics. In *EUROGRAPHICS 2013 State of the Art Reports* (2013), Eurographics Association. 2
- [BW98] BARAFF D., WITKIN A.: Large steps in cloth simulation. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1998), SIGGRAPH '98, ACM, pp. 43–54. 2
- [DBB11] DIZIOL R., BENDER J., BAYER D.: Robust real-time deformation of incompressible surface meshes. In *Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (New York, NY, USA, 2011), SCA '11, ACM, pp. 237–246. 2
- [EB08] ENGLISH E., BRIDSON R.: Animating developable surfaces using nonconforming elements. *ACM Trans. Graph.* 27, 3 (Aug. 2008), 66:1–66:5. 2
- [GHF*07] GOLDENTHAL R., HARMON D., FATTAL R., BERCOVIER M., GRINSFUND E.: Efficient simulation of inextensible cloth. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2007)* 26, 3 (2007), to appear. 2
- [HCJ*05] HONG M., CHOI M.-H., JUNG S., WELCH S., TRAPP J.: Effective constrained dynamic simulation using implicit constraint enforcement. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on* (2005), pp. 4520–4525. 2
- [HCP013] HERNANDEZ F., CIRIO G., PEREZ A. G., OTADUY M. A.: Anisotropic strain limiting. In *Proc. of Congreso Español de Informática Gráfica* (2013). 2
- [ITF04] IRVING G., TERAN J., FEDKIW R.: Invertible finite elements for robust simulation of large deformation. In *Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation* (Aire-la-Ville, Switzerland, Switzerland, 2004), SCA 04, Eurographics Association, pp. 131–140. 6
- [KCMF12] KIM T.-Y., CHENTANEZ N., MÜLLER-FISCHER M.: Long range attachments - a method to simulate inextensible clothing in computer games. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2012), Eurographics Association, pp. 305–310. 2
- [KNE10] KELAGER M., NIEBE S., ERLEBEN K.: A triangle bending constraint model for position-based dynamics. In *VRIPHYS* (2010), Erleben K., Bender J., Teschner M., (Eds.), Eurographics Association, pp. 31–37. 2
- [KPGF07] KUBIAK B., PIETRONI N., GANOVELLI F., FRATARCANGELI M.: A robust method for real-time thread simulation. In *Proceedings of the 2007 ACM Symposium on Virtual Reality Software and Technology* (New York, NY, USA, 2007), VRST '07, ACM, pp. 85–88. 2
- [MHR06] MÜLLER M., HENNIX B. H. M., RATCLIFF J.: Position based dynamics. *Proceedings of Virtual Reality Interactions and Physical Simulations* (2006), 71–80. 1, 2, 5, 6, 9

- [MKC12] MÜLLER M., KIM T.-Y., CHENTANEZ N.: Fast simulation of inextensible hair and fur. In *VRIPHYS* (2012), Bender J., Kuijper A., Fellner D. W., Guérin E., (Eds.), Eurographics Association, pp. 39–44. 2
- [Mül08] MÜLLER M.: Hierarchical position based dynamics. *Proceedings of Virtual Reality Interactions and Physical Simulations* (2008). 2
- [PCH*13] PEREZ A. G., CIRIO G., HERNANDEZ F., GARRE C., OTADUY M. A.: Strain limiting for soft finger contact simulation. In *Proc. of World Haptics Conference* (April 2013), IEEE. 2
- [PDA03] PICINBONO G., DELINGETTE H., AYACHE N.: Non-linear anisotropic elasticity for real-time surgery simulation. *Graph. Models* 65, 5 (Sept. 2003), 305–321. 2
- [Pro95] PROVOT X.: Deformation constraints in a mass-spring model to describe rigid cloth behavior. *Proceedings of Graphics Interface* (1995), 147–154. 2
- [TPS09] THOMASZEWSKI B., PABST S., STRASSER W.: Continuum-based strain limiting. *Computer Graphics Forum* 28, 2 (2009), 569–576. 2, 3, 7
- [Tsi06] TSIKINIS K. D.: Better cloth through unbiased strain limiting and physics-aware subdivision. 2
- [WOR10] WANG H., O'BRIEN J., RAMAMOORTHY R.: Multi-resolution isotropic strain limiting. *ACM Trans. Graph.* 29, 6 (Dec. 2010), 156:1–156:10. 2, 7

付録 A: 拘束のプロジェクトののための陽解法

ここで、手法の実装を容易にするために拘束のための陽解法を示します。 $\mathbf{p}_0 = \mathbf{q}_0 = \mathbf{0}$ と仮定します。

歪みベース拘束

\mathbf{c}_i は、 \mathbf{Q}^{-1} の列で、 \mathbf{f}_i は、 \mathbf{F} の列とします。すなわち

$$[\mathbf{c}_1, \mathbf{c}_2] = \mathbf{Q}^{-1} \quad (24)$$

$$[\mathbf{f}_1, \mathbf{f}_2] = \mathbf{F} \quad (25)$$

そして

$$[\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3] = \mathbf{Q}^{-1} \quad (26)$$

$$[\mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_3] = \mathbf{F} \quad (27)$$

が、三角形と四面体のそれぞれに対応します。すると、 \mathbf{S} の各要素は下記のように計算されます。

$$\mathbf{S}_{ij} = \mathbf{f}_i \cdot \mathbf{f}_j = (\mathbf{Pc}_i) \cdot (\mathbf{Pc}_j), \quad (28)$$

ここで、三角形の場合 $i, j \in \{1, 2\}$ で四面体の場合 $i, j \in \{1, 2, 3\}$ となります。PBD 手法で必要になるパーティクルの位置に関する \mathbf{s} の要素の導関数は、

$$\nabla \mathbf{S}_{ij} = [\nabla_{\mathbf{p}_1}, \nabla_{\mathbf{p}_2}] \mathbf{S}_{ij} = \mathbf{f}_j \mathbf{c}_i^T + \mathbf{f}_i \mathbf{c}_j^T \quad (29)$$

$$\nabla \mathbf{S}_{ij} = [\nabla_{\mathbf{p}_1}, \nabla_{\mathbf{p}_2}, \nabla_{\mathbf{p}_3}] \mathbf{S}_{ij} = \mathbf{f}_j \mathbf{c}_i^T + \mathbf{f}_i \mathbf{c}_j^T \quad (30)$$

が三角形と四面体それぞれが対応します。そして

$$\nabla_{\mathbf{p}_0} \mathbf{S}_{ij} = - \sum_{k=1}^d \nabla_{\mathbf{p}_k} \mathbf{S}_{ij}, \quad (31)$$

ここで、 $d=2$ の場合が三角形で、 $d=3$ の場合が四面体となります。[MHR06] に従い、 \mathbf{S}_{ij} に関するパーティクルプロジェクトンベクトルを下記のように求めることが出来ます。

$$\Delta \mathbf{p}_k = -\lambda w_k \nabla_{\mathbf{p}_k} \mathbf{S}_{ij}, \quad (32)$$

ここで、 w_k は、パーティクル k の質量の逆数で

$$\lambda = \frac{\mathbf{S}_{ii} - s_i^2}{\sum_k w_k |\nabla_{\mathbf{p}_k} \mathbf{S}_{ii}|^2}, \quad (33)$$

$$\lambda = \frac{\mathbf{S}_{ij}}{\sum_k w_k |\nabla_{\mathbf{p}_k} \mathbf{S}_{ij}|^2}, \quad (34)$$

$$\lambda = 2 \frac{\sqrt{\mathbf{S}_{ii}} - s_i}{\sum_k w_k |\nabla_{\mathbf{p}_k} \mathbf{S}_{ii}|^2} \sqrt{\mathbf{S}_{ii}} \quad (35)$$

は、式 (9), (10), (14) のそれぞれに対応します。修正せん断拘束関数は、式 (15) で与えられ、勾配は

$$\nabla \tilde{\mathbf{S}}_{ij} = \frac{1}{|\mathbf{f}_i| |\mathbf{f}_j|} \nabla \mathbf{S}_{ij} - \frac{|\mathbf{f}_j|^2 \mathbf{f}_i \mathbf{c}_i^T + |\mathbf{f}_i|^2 \mathbf{f}_j \mathbf{c}_j^T}{|\mathbf{f}_i|^3 |\mathbf{f}_j|^3} \mathbf{S}_{ij}. \quad (36)$$

三角形のマテリアル座標系

三角形の頂点のテクスチャ座標、 $\mathbf{u}_i = (u_i, v_i)$ は、元の形状の伸びが考慮されていないのでマテリアルの座標としては使う事ができません。マテリアル座標 $\mathbf{q}_0, \mathbf{q}_1, \mathbf{q}_2 \in \mathbb{R}^2$ を計算するために、正規直行のローカルフレームが必要です。ワールド空間での三角形の頂点の位置を $\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^3$ とします。2つのワールド空間の接線ベクトル $\mathbf{t}_u, \mathbf{t}_v \in \mathbb{R}^3$ は、 u と v 軸に沿って下記となります。

$$(\mathbf{t}_u, \mathbf{t}_v) = (\mathbf{x}_1 - \mathbf{x}_0, \mathbf{x}_2 - \mathbf{x}_0) (\mathbf{u}_1 - \mathbf{u}_0, \mathbf{u}_2 - \mathbf{u}_0)^{-1} \quad (37)$$

これらの接線が、グローバルな位置からマテリアル座標へと変換するためのローカル座標を下記で与えます。

$$(\mathbf{c}_1, \mathbf{c}_2) = [\mathbf{n}_1, \mathbf{n}_2]^T (\mathbf{x}_1 - \mathbf{x}_0, \mathbf{x}_2 - \mathbf{x}_0), \quad (38)$$

ここで $\mathbf{n}_1 = \frac{\mathbf{t}_u}{|\mathbf{t}_u|}$ and $\mathbf{n}_2 = \frac{\mathbf{t}_v}{|\mathbf{t}_v|}$.

\mathbf{t}_u と \mathbf{t}_v がお互いに垂直になるのを確認するために、後者は代替として三角形の法線と \mathbf{t}_u の外積として計算されます。

曲げ拘束

$\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4$ を2つの三角形 $(\mathbf{p}_1, \mathbf{p}_3, \mathbf{p}_4)$ と $(\mathbf{p}_2, \mathbf{p}_4, \mathbf{p}_3)$ が構成する曲げ要素の頂点とします。曲げ角 ϕ は、2つの三角形の法線で下記のように求められます。

$$\phi = \arccos \left(\frac{(\mathbf{p}_3 - \mathbf{p}_1) \times (\mathbf{p}_4 - \mathbf{p}_1)}{|\mathbf{p}_3 - \mathbf{p}_1| |\mathbf{p}_4 - \mathbf{p}_1|} \cdot \frac{(\mathbf{p}_4 - \mathbf{p}_2) \times (\mathbf{p}_3 - \mathbf{p}_2)}{|\mathbf{p}_4 - \mathbf{p}_2| |\mathbf{p}_3 - \mathbf{p}_2|} \right) \quad (39)$$

[BMF03] の曲げモードに対応する空間導関数は、下記の通り

$$\nabla_{\mathbf{p}_1} \phi = |\mathbf{e}| \mathbf{n}_1 \quad (40)$$

$$\nabla_{\mathbf{p}_2} \phi = |\mathbf{e}| \mathbf{n}_2 \quad (41)$$

$$\nabla_{\mathbf{p}_3} \phi = \frac{(\mathbf{p}_1 - \mathbf{p}_4) \cdot \mathbf{e}}{|\mathbf{e}|} \mathbf{n}_1 + \frac{(\mathbf{p}_2 - \mathbf{p}_4) \cdot \mathbf{e}}{|\mathbf{e}|} \mathbf{n}_2 \quad (42)$$

$$\nabla_{\mathbf{p}_4} \phi = \frac{(\mathbf{p}_3 - \mathbf{p}_1) \cdot \mathbf{e}}{|\mathbf{e}|} \mathbf{n}_1 + \frac{(\mathbf{p}_3 - \mathbf{p}_2) \cdot \mathbf{e}}{|\mathbf{e}|} \mathbf{n}_2, \quad (43)$$

ここで

$$\mathbf{e} = \mathbf{p}_4 - \mathbf{p}_3 \quad (44)$$

$$\mathbf{n}_1 = \frac{(\mathbf{p}_3 - \mathbf{p}_1) \times (\mathbf{p}_4 - \mathbf{p}_1)}{|\mathbf{p}_3 - \mathbf{p}_1| |\mathbf{p}_4 - \mathbf{p}_1|} \quad (45)$$

$$\mathbf{n}_2 = \frac{(\mathbf{p}_4 - \mathbf{p}_2) \times (\mathbf{p}_3 - \mathbf{p}_2)}{|\mathbf{p}_4 - \mathbf{p}_2| |\mathbf{p}_3 - \mathbf{p}_2|}. \quad (46)$$

全ての導関数の符号は、 $(\mathbf{n}_1 \times \mathbf{n}_2) \cdot \mathbf{e} > 0$ の場合反転する必要があります。

体積 / 面積 保存拘束

式 (22) から、下記が得られる。

$$C_{\text{volume}}(\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3) = \det(\mathbf{P}) - \det(\mathbf{Q}) \quad (47)$$

$$= \mathbf{p}_1^T (\mathbf{p}_2 \times \mathbf{p}_3) - \mathbf{q}_1^T (\mathbf{q}_2 \times \mathbf{q}_3) \quad (48)$$

そして勾配は

$$\nabla_{\mathbf{p}_1} C_{\text{volume}} = \mathbf{p}_2 \times \mathbf{p}_3 \quad (49)$$

$$\nabla_{\mathbf{p}_2} C_{\text{volume}} = \mathbf{p}_3 \times \mathbf{p}_1 \quad (50)$$

$$\nabla_{\mathbf{p}_3} C_{\text{volume}} = \mathbf{p}_1 \times \mathbf{p}_2 \quad (51)$$

$$\nabla_{\mathbf{p}_0} C_{\text{volume}} = -\mathbf{p}_2 \times \mathbf{p}_3 - \mathbf{p}_3 \times \mathbf{p}_1 - \mathbf{p}_1 \times \mathbf{p}_2 \quad (52)$$

$$(53)$$

同様に、三角形の面積保存拘束を導くことができる。

$$C_{\text{area}}(\mathbf{p}_1, \mathbf{p}_2) = |\mathbf{p}_1 \times \mathbf{p}_2|^2 - |\mathbf{q}_1 \times \mathbf{q}_2|^2. \quad (54)$$

その導関数は、

$$\nabla_{\mathbf{p}_1} C_{\text{area}} = 2 \mathbf{p}_2 \times (\mathbf{p}_1 \times \mathbf{p}_2) \quad (55)$$

$$\nabla_{\mathbf{p}_2} C_{\text{area}} = 2 \mathbf{p}_1 \times (\mathbf{p}_2 \times \mathbf{p}_1) \quad (56)$$

$$\nabla_{\mathbf{p}_0} C_{\text{area}} = -2 \mathbf{p}_2 \times (\mathbf{p}_1 \times \mathbf{p}_2) - 2 \mathbf{p}_1 \times (\mathbf{p}_2 \times \mathbf{p}_1) \quad (57)$$

$$(58)$$

該当するパーティクルプロジェクトンベクトルは

$$\Delta \mathbf{p}_k = -\lambda w_k \nabla_{\mathbf{p}_k} C_{\text{volume(area)}}, \quad (59)$$

ここで

$$\lambda = C_{\text{volume(area)}} / \sum_k w_k |\nabla_{\mathbf{p}_k} C_{\text{volume(area)}}|^2. \quad (60)$$