

布のリアルタイム・シミュレーションのためのアルゴリズム

小堀 誠[†]大川 善邦[‡]日本大学大学院工学研究科[†]日本大学工学部[‡]

1 はじめに

3D グラフィックスにおける布の挙動をリアルタイムで実現することを考える。布の動力学は線形バネモデル[1]を用い、布とオブジェクトの衝突は、布の頂点とオブジェクトのポリゴン間において行う[2]。

衝突検知の計算負荷が大きいので、これを低減するために、布を粗いメッシュに分割する。この場合、メッシュ内部の点とオブジェクトの衝突は検知できないので、オブジェクトが布を突き破って外へ現れることがある。

この現象を補正するために、グラフィックボードの仮レンダリング機能を利用して、布がオブジェクト内へ食い込んだ点を検出し、その点を強制的にオブジェクトの外部へ引き出すアルゴリズムを提案する。グラフィックボード内における布のメッシュは、きめ細かく分割するので、補正の精度は高くなる。CPU における粗いメッシュとグラフィックボードにおける細かいメッシュを2重に使用して衝突を検知し、処理時間を低減する。

2 布の動力学と衝突検知

布の動力学として Fig. 1 に示した線形バネモデルを用いる。オブジェクトは Fig. 2 に示したものをを用いる。机の上にアヒルのおもちゃを置いた状況を考える。布は上部から重力によって落下する。マウスによって布の一端を掴み、布を引き回すこともできる。

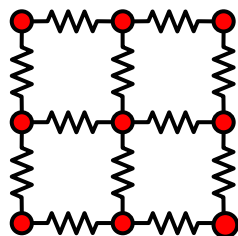


Fig. 1

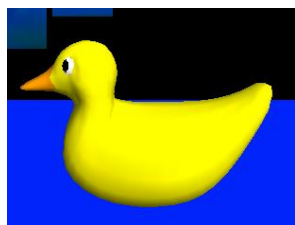


Fig. 2

予備実験として、布の頂点数に対して、画面表示のフレームレートを測定した。その結果を Fig. 3 に示す。

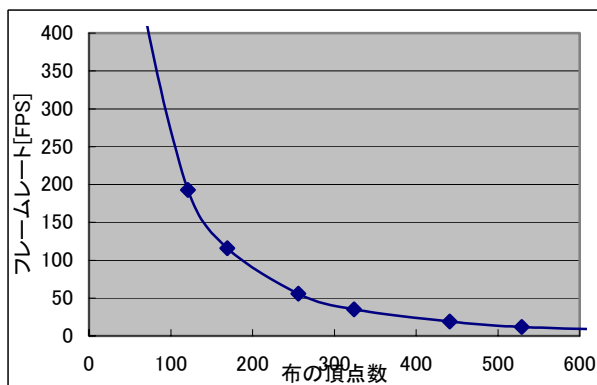


Fig. 3

数字そのものは、使用するコンピュータの能力によって変化するが、衝突検知の回数が増えるにしたがって、画面表示のフレームレートが減少する。

一般に、布は人の服装などに用いる。人を構成するポリゴンは、デザイン等の条件によって決まるので、布からの条件によって増減できない。布のリアルタイム動作を実現するためには、布の頂点数を小さく抑えることが必要である。

3 概略アルゴリズム

本研究において採用したアルゴリズムは、大別すると

1. CPU における処理
2. グラフィックボードにおける処理

CPU の処理は

1. 1. 粗い布に関して動力学計算を行う
1. 2. 粗い布とオブジェクトの衝突検知を行う
1. 3. 衝突検知したとき、衝突力学を適用して新しい頂点位置を計算する

グラフィックボードの処理は

2. 1. 布のメッシュを細分化し、それらの頂点の位置を計算する
2. 2. カメラの視点を決めて、オブジェクトを仮レンダリングする
2. 3. 細分化した布の頂点とオブジェクトの遠近関係を取得する
2. 4. もしカメラに対して、オブジェクトが近ければ布の頂点を移動する

4. グラフィックボード内におけるアルゴリズム

グラフィックボードは、CPU から送られてきたメッシュに対して、ボード内に実装されているテセレータを用いて、メッシュを細分化する。

CPU 内で用いるメッシュとグラフィックボードのメッシュの例を Fig. 4 に示した。

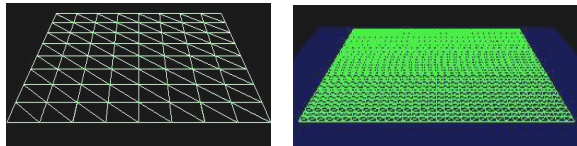


Fig. 4

次に、Fig. 5 に示した 6 個のカメラ位置からオブジェクトを見て、レンダリングを実行するのだが、このとき、レンダリングのターゲットをディスプレイではなくて、ボード内のテクスチャメモリとする。

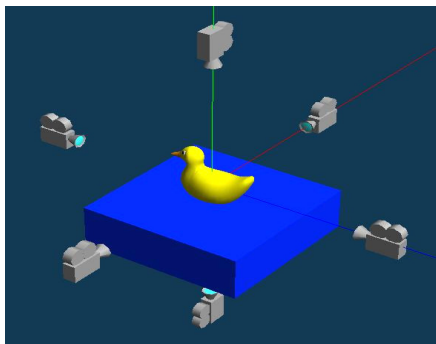


Fig. 5

この処理を、ここでは、仮レンダリングと呼ぶ。仮レンダリングによって、カメラから見たオブジェクト(厳密に言えば、カメラに最も近いオブジェクト上の点)のカラー、座標、などの情報がピクセル単位でテクスチャメモリに記録される。もし、オブジェクトが静止物体であれば、この操作は最初に 1 回行えばよいことになるが、オブジェクトの位置が動的に変化する場合、この操作はフレームごとに行う。

次に、細分化されたメッシュの頂点の座標をカメラ座標に変換する。この操作によって、メッシュ頂点のピクセル座標が取得できる。この座標を用いて仮レンダリングされたテクスチャメモリからオブジェクトの位置を取得する。ここで、メッシュの座標とオブジェクトの座標を比較することによって、カメラに対するオブジェクトとメッシュの遠近関係を決定することができる。

布のメッシュがカメラに近い場合はそのままとし、逆に、オブジェクトが布よりカメラに近い場合は、布の頂点を移動する。

頂点の移動は仮レンダリングされた物体の該

当位置に関連付けられた法線データの方角に移動することとした。

5. 実験結果

Fig. 7 に、CPU 処理だけの画面とグラフィックボードによる補正を行った画面を示した。

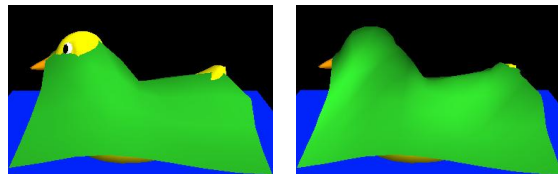


Fig. 7

グラフィックボードによる補正の効果は明らかである。 Fig. 8 に、CPU 処理だけの場合のフレームレートとグラフィックボードによる補正を行った場合のフレームレートを示した。

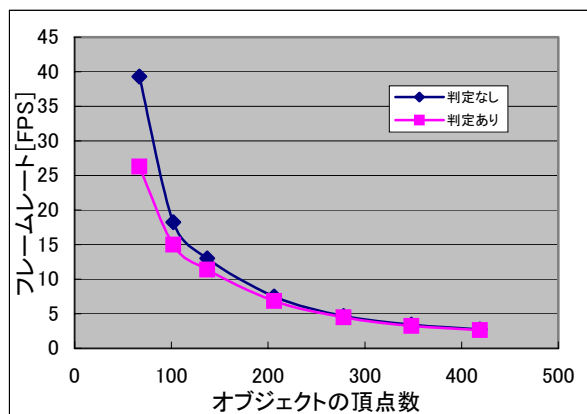


Fig. 8

グラフィックボードの補正を行っても、処理時間はほとんど変わらないことが示されている。これは、CPU 処理とグラフィックボードの処理が並行的に実行されるからである。

6. おわりに

布のシミュレーションを行う際に、布とオブジェクトの衝突検知に時間がかかる。布をリアルタイムでシミュレーションするためには、このボトルネックを解消しなければならない。

本研究では、衝突の検知を 2 段階に分け、第 1 段階で粗いメッシュで処理を行い、第 2 段階では細かいメッシュを用いて処理を行うアルゴリズムを提案した。しかも、これらの 2 つの処理は、CPU とグラフィックボードにおいて、並行的に実行できる。

今後の課題として、たとえば人に服を着せて運動させる状況を実現する、などがある。

7. 参考文献

- [1] D. H. House & D. E. Breen (ed.), Cloth Modeling and Animation, A.K. Peters, Ltd, 2000.
- [2] G. Bergen, Collision Detection in Interactive 3D Environments, Elsevier, 2004.