

距離マップおよび法線マップを用いた ハイポリゴンモデルへの力覚提示

脇田 航^{†1} 井門 俊^{†1}

本論文では、距離マップおよび法線マップを用いたハイポリゴンモデルへの力覚提示について述べる。ハイポリゴンモデルへの力覚提示では、LOD 手法や境界モデルを用いたジオメトリベースの手法が用いられているが、低コストかつ高精細な形状知覚を行うことは困難である。一方で、2次元画像に応じて低コストでポリゴン内部の凹凸表現を行うテクスチャベースの手法がいくつか提案されている。我々は、テクスチャベースの手法を用いてハイポリゴンモデルへの力覚提示を試みる。具体的には、ハイポリゴンモデルおよびハイポリゴンモデルを粗く近似したローポリゴンモデルを用意し、ローポリゴンモデルのUVマップへ、ハイポリゴンモデルとローポリゴンモデル表面の距離、およびハイポリゴンモデルの頂点法線ベクトルを色付けることにより、距離マップおよび法線マップを生成する。交差判定はローポリゴンモデルに対して行い、交点における距離マップの値に応じてローポリゴンモデルの表面高さを動的に変化させ、反力を法線マップの値に応じて振動させる。これにより、低コストでハイポリゴンモデルへの力覚提示が可能となった。

A Haptic Rendering for High Polygon Model Using Distance Map and Normal Map

WATARU WAKITA^{†1} and SHUN IDO^{†1}

This paper describes a haptic rendering for high polygon model using distance map and normal map. In haptic rendering for high polygon model, it is difficult to have a high definition haptic perception of the shape at low cost, even though a geometry-based technique using the LOD technique and the bounding model has been used. On the other hand, several texture-based techniques have proposed to represent the convexo concave of polygons depending on the 2D image at low cost. We attempt a haptic rendering for the high polygon model using a texture-based technique. Specifically, user prepares a high polygon model and a low polygon model derived from coarsely-approximated the high polygon model. A distance map can be obtained by coloring the distance between object surfaces onto the UV map of a low polygon model, and a normal map can be

obtained by coloring the vertex normal of a high polygon model respectively. The intersection detection is implemented for a low polygon model, the surface height is dynamically changed according to the distance map and the direction of the reaction force is perturbed according to the normal map. As a result, our system enabled haptic rendering for high polygon model at low cost.

1. はじめに

近年、3DCG 技術の発達にともない、映画、テレビ、ゲーム、Web などの様々な分野で 3DCG に触れる機会が増えてきた。また、プログラミング可能な GPU (Graphics Processing Unit)^{1),2)} の発達にともない、高精細な 3D シーンを柔軟かつリアルタイムにレンダリングを行うことが可能となりつつある。通常、高精細な 3D オブジェクトを構築するには、細部まで作り込んだ形状モデルやテクスチャなどが必要となる。しかしながら、携帯機器やコンシューマ機などのようなハードウェアの性能の制限を受ける環境において、高精細な 3D オブジェクトを厳密かつリアルタイムにレンダリングを行うことは困難である。このような環境では、レンダリングコストを抑えるため、ハードウェアの性能に応じてポリゴン数やテクスチャの解像度を制限するのが一般的である。また、レンダリング品質向上のため、視点距離などに応じてポリゴン数やテクスチャの解像度を变化させる LOD (Level of Detail) 手法³⁾ や、GPU が使える環境では法線マッピング^{2),4)} などを用いて、コストを抑えつつリアルタイムに擬似高精細なレンダリングを行う工夫がなされている。

一方、力覚の分野において、これら 3D オブジェクトの形状知覚は重要な課題である。触・力覚提示の需要は、医療、福祉、教育、CAD、エンタテインメントなどの様々な分野で高まりつつあるが、力覚提示を行うための装置には、視覚や聴覚分野のモニターやスピーカのような標準的なものがなく、また、装置そのものが高価なため、導入が困難といった問題がある。しかしながら、近年、Novint 社の Falcon のような安価な 3 自由度の設置型力覚提示装置が登場するなど、将来、市場拡大やさらなる低価格化などによる普及が進めば、導入が困難であった様々な分野への応用が期待される。通常、力覚提示において十分な形状知覚を行うためには、物理演算や視覚提示などにかかるコストも考慮しつつ、1 kHz 以上の更新レートの確保が必要とされている。近年、AGEIA Technologies 社の PhysX のような

^{†1} 愛媛大学大学院理工学研究科

Graduate School of Science and Engineering, Ehime University

PPU (Physics Processing Unit) の登場により、将来的に物理演算を柔軟かつリアルタイムに行えるようになることが期待できるが、ハードウェアの性能の制限を受ける環境においては、3D オブジェクトのポリゴン数が増えるにつれ、計算コストも増大するため、十分な形状知覚を行うための更新レートを確保することが困難となる。このため、いかにコストを抑えつつ、高精細な形状知覚を行えるかが問題となる。低コストで高精細な形状知覚を行う試みとして、境界モデルを用いたもの⁵⁾ や LOD 手法を用いたもの^{6),7)}、3DCG におけるグーローシェーディングのように、ポリゴン内部の法線ベクトルを補間し、法線方向に反力を振動させることで、ローポリゴンで滑らかな勾配を表現可能なフォースシェーディング手法⁸⁾ などが提案されている。フォースシェーディング手法のようなポリゴン内部への精細な力覚提示手法として、材質に合った力覚提示用のシェーダプログラムを用意しておき、ポリゴンに割り当てられた材質に応じて力覚提示を行う手法⁹⁾、フォスマップ (法線マップ) を 3D オブジェクトにマッピングし、交点座標におけるフォスマップの画素値に応じて反力方向を振動させることで、ポリゴン内部のテクスチャに応じた勾配表現を行うフォスマッピング手法^{10),11)}、高さマップを 3D オブジェクトにマッピングし、高さマップの画素値に応じて動的に 3D オブジェクトの表面高さを変化させることで、ポリゴン内部の高さ表現を行う手法^{11),12)} などが提案されている。しかしながら、フォスマッピング手法単体では、ポリゴン内部の高さの違いを明確に感じることはできず、また、高さマップを用いた手法単体では、勾配の違いを感じることはできない。このため、我々はテクスチャに合わせた法線マップと高さマップを用いて、法線マッピングによる擬似高精細モデルのポリゴン内部の擬似凹凸への勾配および高さ表現可能な力覚提示手法¹³⁾ の提案を行った。

これらテクスチャベースの力覚提示では、低コストでテクスチャに応じた 3D オブジェクト表面の凹凸表現が可能であるが、3D オブジェクトの形状が肌理細くなるにつれ、3D ポインタ (デバイスの先端位置を仮想空間内の座標に変換したもの) と 3D オブジェクト表面との干渉チェックの際、ポインタがオブジェクト表面に触れた瞬間、ポインタが弾き飛ばされたり、裏面へ突き抜けたり、また、エッジ (ポリゴンとポリゴンの境目) 部分において、小刻みに振動したりするといった不具合が生じやすくなり、滑らかな力覚提示を行うことが困難となる。

そこで我々は、肌理細かでないローポリゴンモデルかつテクスチャベースの力覚提示手法が、低コストかつポリゴン内部の凹凸感を滑らかに表現可能なことに注目し、距離マップおよび法線マップを用いたハイポリゴンモデルへの力覚提示手法の提案を行う。具体的には、あらかじめハイポリゴンモデルおよびハイポリゴンモデルを粗く近似したローポリゴンモ

デルを用意しておき、ローポリゴンモデルに無地のテクスチャをラッピングし、両モデル表面間の距離およびハイポリゴンモデルの頂点法線ベクトルを、ローポリゴンモデル表面のテクスチャにマッピングする。交差判定はローポリゴンモデルに対して行い、3D ポインタが 3D オブジェクト表面に一定距離以内に近づいた場合をヒットしたと判定し、そのときの交点における距離マップの値に応じてローポリゴンモデルの表面高さを動的に変化させ、反力を法線マップの値に応じて振動させる。これにより、低コストでハイポリゴンモデルへの力覚提示が可能となるかどうか評価実験を行い、提案手法の有効性の検証を行う。

2. システム設計および構築

2.1 システム概要

本研究では、ローポリゴンかつテクスチャベースの力覚提示を行うことにより、低コストで高精細モデルへの力覚提示を可能にすることを目的とし、ハイポリゴンモデルとローポリゴンモデルから距離マップおよび法線マップを生成するソフトウェアの開発およびローポリゴンモデルからハイポリゴンモデルへの力覚提示システムの開発を行う。

2.2 ポリゴンベースの力覚提示手法

力覚提示によりポリゴンの平面を表現する方法として、ペナルティ法に基づき、ポリゴンに 3D ポインタが侵入したときの侵入量などに応じて反力を計算し、面法線方向に反力を発生させる手法^{14),15)} が用いられているが、3D オブジェクトのポリゴン数が増え、形状が複雑になるにつれ、反力提示に不具合が生じやすくなる。たとえば、ペナルティ法に基づいた力覚提示^{14),15)} において、オブジェクト内で定義されたポリゴン順に 3D ポインタと 3D オブジェクトのポリゴンとの干渉チェックを行う場合、薄い板や細い円柱といった表面と裏面との距離が小さい部分では、ポインタがオブジェクト表面に触れた瞬間、ポインタが裏面に突き抜けたり、また、干渉が起こったポリゴンとは別のポリゴンの面法線方向に反力が発生し、ポインタが弾き飛ばされたりするなどの不具合が生じる^{5),16)}。このため、干渉が起こったポリゴンおよび隣接するポリゴンを優先的に探索し、再度干渉チェックを行う Constraints-based God-object Method¹⁶⁾ が提案されているが、ポリゴン数が増え、オブジェクト表面が肌理細くなるにつれ、干渉チェック中のポリゴンが隣接ポリゴンへ移行する際、ポインタが隣接ポリゴンを大きく越えて移動し、ペナルティ法に基づく力覚提示同様、突き抜けや弾き飛ばされるといった現象が連続的に発生し、3D オブジェクトの表面をなぞる際、細かな凹凸を表現することが困難となる (図 1 参照)。

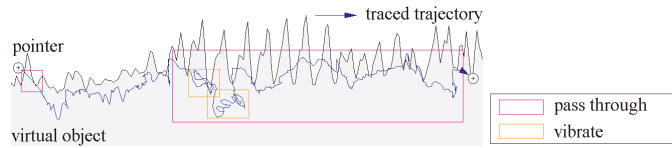


図 1 ポリゴンベースの力覚提示手法の問題
Fig.1 Problem of polygon-based haptic rendering technique.

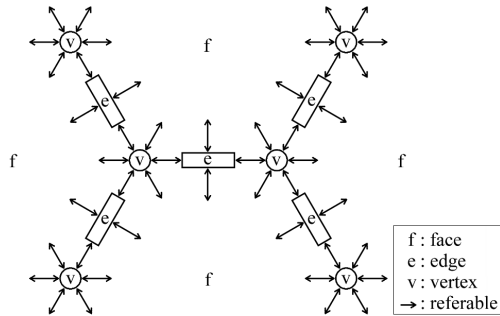


図 2 3D オブジェクトのデータ構造
Fig.2 Data structure of 3D object.

2.3 3D オブジェクトのデータ構造

本システムでは、反力計算に Constraints-based God-object Method¹⁶⁾ の概念を導入する。この手法では、凹形状のエッジ部分において、3D ポインタとポリゴンとの間で干渉が起こらず、ポインタがすり抜ける不具合が稀に生じる¹⁶⁾。このような不具合を防ぐには、3D ポインタとポリゴンとの干渉チェックだけでなく、エッジに法線情報（たとえば、エッジを共有するポリゴンの面法線ベクトルの平均）を持たせ、3D ポインタがエッジ付近に近づいた際、エッジの法線方向に干渉チェック用のポリゴンを動的に生成するなど、何らかの対策が必要となる。このため、本システムでは、3D オブジェクトのデータ構造を、3D オブジェクト内のどの要素（頂点、エッジ、ポリゴン）からでも隣接する各要素の情報を参照可能なものとした。

本システムで用いている 3D オブジェクトのデータ構造を図 2 に示す。ここで、f は 3D オブジェクトの面（ポリゴン）、e はエッジ、v は頂点、矢印は先の要素の情報を参照可能であることを表す。各頂点は、

- 頂点が属するポリゴン番号
- 頂点が属するエッジ番号
- 頂点の位置ベクトル
- 頂点法線ベクトル

各エッジは、

- エッジが属するポリゴン番号
- エッジに属する始点頂点番号
- エッジに属する終点頂点番号

各ポリゴンは、

- ポリゴンに属するエッジ番号
- ポリゴンに属する頂点番号

などの情報を保持する。各要素間で互いの情報を保持しているため、各要素を介することで、

- 各頂点を共有する全エッジ
- 各エッジを共有する全ポリゴン
- 各ポリゴンに隣接する全ポリゴン

といった探索が可能となる。

2.4 テクスチャベースの力覚提示および視覚提示

ハイポリゴンモデルへの力覚提示において、ポリゴンベースの手法を用いると、前述の力覚提示の不具合が生じやすくなる。また、視覚提示にかかるコストに加え、力覚提示にかかる計算コストも増大するため、本システムでは、力覚提示にローポリゴンモデルを用い、ハイポリゴンモデルとローポリゴンモデルから作成した距離マップに応じてローポリゴンモデルの表面高さを動的に変化させ、法線マップに応じて反力方向を摂動させることにより、力覚提示の不具合や計算コストを抑えつつ、ローポリゴンモデルからハイポリゴンモデルへの力覚提示を行う。また、本システムでは、視覚提示にかかるコストを抑えるため、ディスプレイリスト¹⁷⁾ による描画を行っているが、我々が提案した擬似凹凸への力覚提示¹³⁾ のように、プログラミング可能な GPU^{1),2)} を転用し、視覚提示を GPU 側で処理すれば、より柔軟かつ高速な力覚提示の実現が期待できる。

2.5 システム構成

本システムの構成を図 3 に示す。本システムは、3D アプリケーション、ディスプレイ、および力覚提示装置で構成され、3D アプリケーションは主に、

- ファイル入出力部

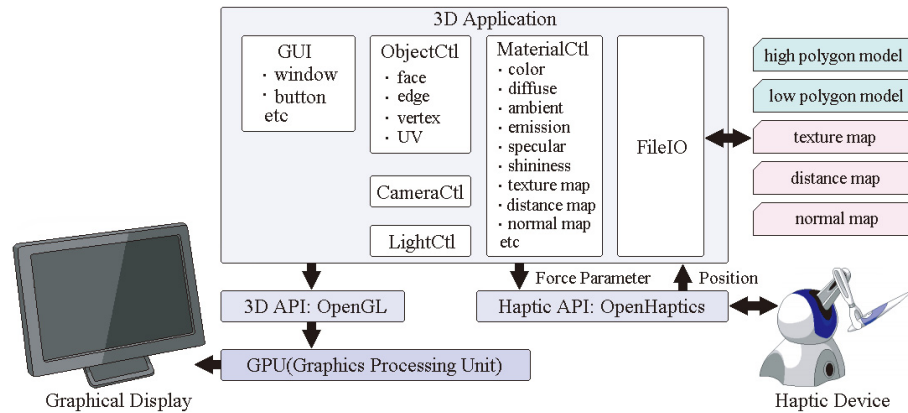


図3 システム構成
Fig. 3 System architecture.

- マテリアル管理部
- オブジェクト管理部
- カメラ管理部
- ライト管理部
- GUI部

で構成される。ファイル入出力部では、各種 3D オブジェクトファイルやテクスチャの入出力を行う。マテリアル部では、ライトを当てたときの各種反射係数やテクスチャなどの管理を行う。オブジェクト部では、面、エッジ、頂点、および UV 座標（テクスチャ座標）の管理を行う。GUI 部では、ウィンドウ、ボタンなどの各種 GUI コンポーネントの管理を行う。本システムでは、力覚提示装置に SensAble Technologies 社の PHANTOM Omni を用いた。このため、PHANTOM 制御用 API である OpenHaptics を介してデバイス位置の取得や反力の出力を行う。

3. 距離マップおよび法線マップ生成

本章では、ハイポリゴンモデルおよびローポリゴンモデルを用いた距離マップおよび法線マップ生成手法について述べる。

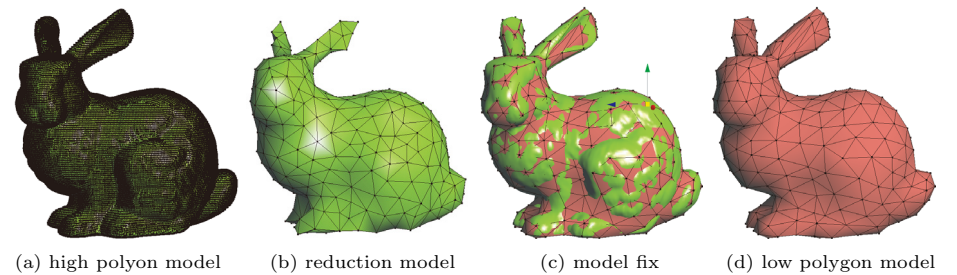


図4 ローポリゴンモデルの作成
Fig. 4 Low polygon model creation.

3.1 ローポリゴンモデルの作成

まず、図 4 (a) のようなハイポリゴンモデルをあらかじめ用意しておき、3D モデリングソフトを用いて、ハイポリゴンモデルを粗く近似したローポリゴンモデルを作成する (図 4 (c), (d) 参照)。この際、ポリゴンを 1 から生成していくと時間がかかるため、モデリングソフトのポリゴンリダクション機能を用いてハイポリゴンモデルのポリゴン数を減らし (図 4 (b) 参照)、ポリゴンを生成し直したり、頂点を移動させてポリゴンの大きさを整えたりするなど、手作業で調整を行っていくと、ポリゴンを 1 から生成するより短時間でモデリングを行うことができる。

次に、作成したローポリゴンモデルの各ポリゴンを、図 5 のように、ポリゴンどうしが重ならないように、2 次元のテクスチャ空間 (u, v) に展開 (UV 展開) する。

3.2 距離マップおよび法線マップの生成

前節で用意したハイポリゴンモデルおよびローポリゴンモデルを用いて、距離マップおよび法線マップの生成を行う。まず、ハイポリゴンモデルの各頂点の位置ベクトル O からローポリゴンモデルの各ポリゴンの面法線方向 (正および負方向) への線分とローポリゴンモデルの各ポリゴンとの交差判定を行う。ここで、交差判定には Möller らの手法¹⁸⁾を用いた。線分とポリゴンが交差した場合、 O と最も近いポリゴンの各頂点 V_0, V_1, V_2 、交点座標 P 、 O と P との距離 D 、および O の頂点法線ベクトル N を取得する。ここで、 D の値は、表面高さを考慮した最大距離 D_{max} (値は任意) で割ることにより $[-1.0, 1.0]$ の範囲に変換したあと、 $[0, 255]$ の範囲に変換しておく。同様に N の 3 成分をそれぞれ $[0, 255]$ の範囲に変換しておく。

次に、 $P - V_1$ および $V_2 - V_0$ から、エッジ V_0V_2 の内分比 $a : b$ を求め、同様に $c : d$

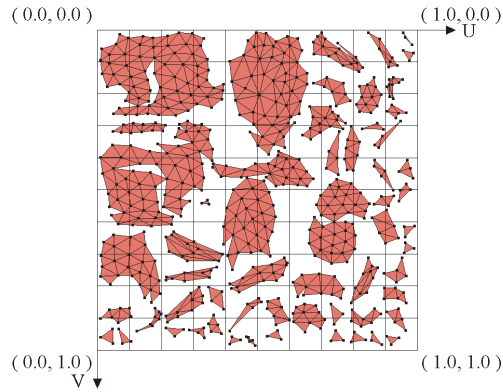


図 5 UV 展開図
Fig. 5 UV development.

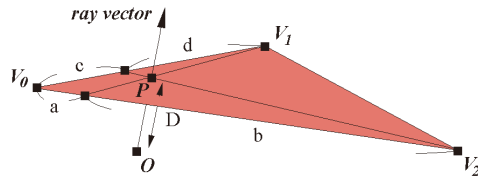


図 6 内分比計算
Fig. 6 Interior division ratio calculation.

を求め (図 6 参照)。これらの内分比および交差したポリゴンの各頂点 V_0, V_1, V_2 に対応するテクスチャ空間上の UV 座標 T_0, T_1, T_2 から、 P に対応するテクスチャ空間上の UV 座標 T_p を以下の式で求める。

$$T_p = T_0 + \frac{ad(T_2 - T_0) + bc(T_1 - T_0)}{ad + bc + bd} \quad (1)$$

最後に、距離マップおよび法線マップのサイズおよび T_p から各種マップのピクセル座標を求め、スケール変換した距離値および法線ベクトルを色付ける。

図 7 に出力した距離マップおよび法線マップを示す。ハイポリゴンモデルにおいて、部分的にポリゴンが大きく、頂点間の距離が遠い部分では、生成したマップに穴が開くため、各種マップ生成時、平均化などで補間したり、各種マップ生成後に UV マップ (UV 展開図) を見ながら手作業で穴埋めしたりするなどの調整が必要となる。

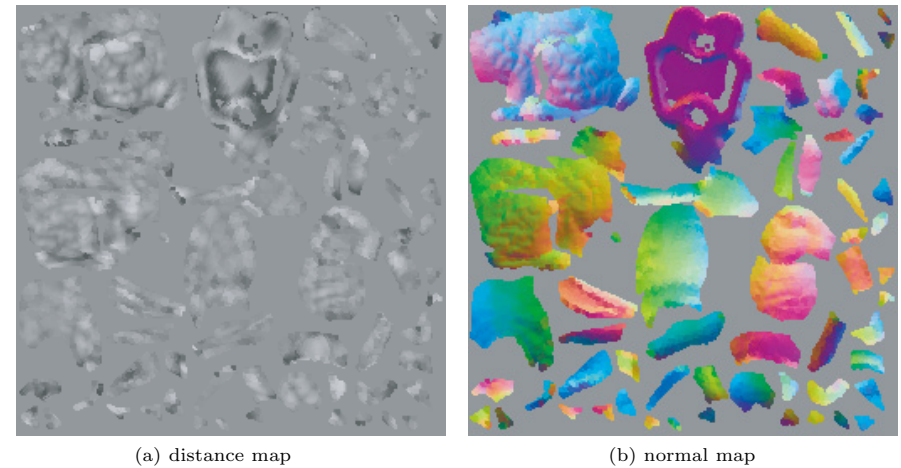


図 7 各種出力結果
Fig. 7 Output results.

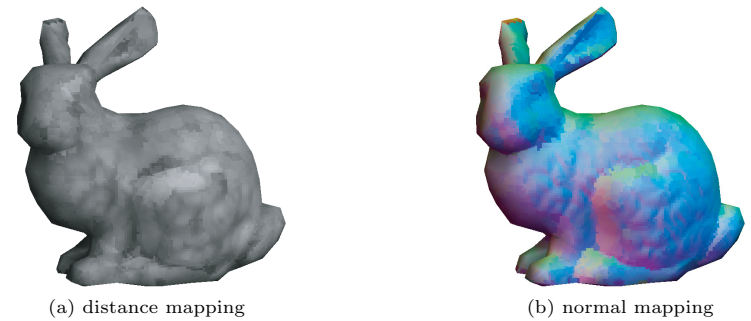


図 8 各種マッピング結果
Fig. 8 Mapping results.

図 8 (a) に、生成した距離マップを UV 展開されたローポリゴンモデルにマッピング (UV マッピング) したものを示す。この距離マップに応じてローポリゴンモデルの表面高さを動的に変化させることで高さの違いの表現を行う。白い部分ほど高く、黒い部分ほど低い部分となる。また、図 8 (b) に、生成した法線マップをローポリゴンモデルに UV マッピングしたものを示す。ハイポリゴンモデルの頂点法線ベクトルが色付けられているのが分かる。こ

の法線方向に反力を振動させることで、勾配の違いの表現を行う。

4. ハイポリゴンモデルへの力覚提示

本章では、前章で用意した距離マップおよび法線マップを UV マッピングしたローポリゴンモデルからハイポリゴンモデルへの力覚提示について述べる。

4.1 力覚提示アルゴリズム

まず、仮想空間（ワールド空間）内の 3D ポインタの位置を決定するため、デバイス空間のデバイスの先端位置をカメラ空間に合わせて、ワールド空間座標に変換する。交差判定はローポリゴンモデルに対して行い、前回の交差判定でヒットしたポリゴンが存在する場合は、そのポリゴン周りを優先的に探索する。また、距離マップの最大距離を考慮し、接触の可能性がある場合をヒットしたと判定し、ヒットした場合、ヒットしたポリゴン上の交点座標に対応する UV 座標を計算する。次に、UV 座標に対応する距離マップおよび法線マップの画素値を取得し、距離マップの値に応じてローポリゴンモデルの面法線ベクトル方向に複製したポリゴンを動的に生成する。ワールド空間内に複製ポリゴンが存在する場合は、複製ポリゴンと交差判定を行い、ポインタとオブジェクト表面が干渉した場合は法線マップの値をデバイス空間座標に変換し、デバイス空間に変換した法線ベクトルおよびポインタの侵入量に応じて反力を計算し、デバイスの先端位置を振動させる（図 9 参照）。

4.2 3D ポインタとローポリゴン表面との交差判定

デバイス空間のデバイスの先端位置をカメラ空間に合わせてワールド空間座標に変換するため、カメラ操作でカメラの向きや位置が変わるたびにカメラ空間の計算を行い、デバイ

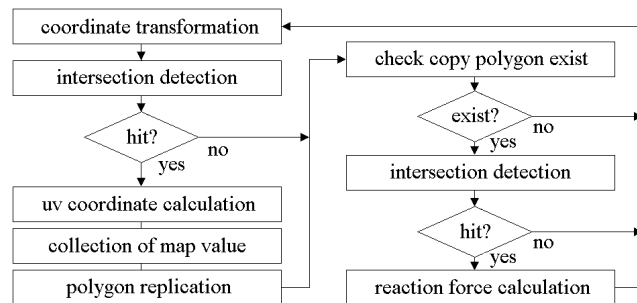


図 9 力覚提示アルゴリズム
Fig. 9 Haptic rendering algorithm.

スの先端位置をカメラ空間に合わせて、3D アプリケーションのワールド空間座標に変換する。これを 3D ポインタの位置 O とする。

次に、Möller らの手法¹⁸⁾ を用いて O からローポリゴンモデルの各ポリゴンの面法線方向（正および負方向）への線分とローポリゴンモデルの各ポリゴンとの交差判定を行う。交差した場合、 O と交点 P との距離 D の計算を行い、前章で用いた最大距離 D_{max} を考慮し、ポインタがオブジェクト表面に一定距離 T （ただし $T > D_{max}$ ）以内に近づいた場合のみヒットしたと判定する（図 10 参照）。

4.3 距離マップに応じた複製ポリゴンの生成

ヒットした場合、式 (1) 同様、交点 P に対応するテクスチャ空間上の UV 座標を計算し、距離マップおよび法線マップの画素値を取得する。距離マップの値は $[0, 255]$ となっているため、 $[-D_{max}, D_{max}]$ の範囲に変換する。また、法線マップの値は $[0, 255]$ となっているため、 $[-1.0, 1.0]$ の範囲に変換する。

次に、距離マップの値に応じてローポリゴンモデルの面法線ベクトルの正または負方向に移動させたポリゴンを複製する（図 11 参照）。

4.4 反力計算

ワールド空間内に複製ポリゴンが存在する場合、再度 Möller らの手法¹⁸⁾ を用いて 3D ポインタ位置 O から複製ポリゴンの面法線ベクトル方向（負方向）への線分と複製ポリゴンとの交差判定を行い、交差した場合、 O と交点との距離（侵入量） L の計算を行う。

次に、 $[-1.0, 1.0]$ の範囲にスケール変換した法線マップの値をカメラ空間に合わせてデバイス空間に変換し、デバイス空間の法線方向 N_d を求める。

$(L > 0)$ の場合、 N_d および L から反力計算を行い、デバイス側で反力として出力する。力覚提示プロセスの更新カウンタを j としたとき、反力 f_j は以下の式で計算を行う。

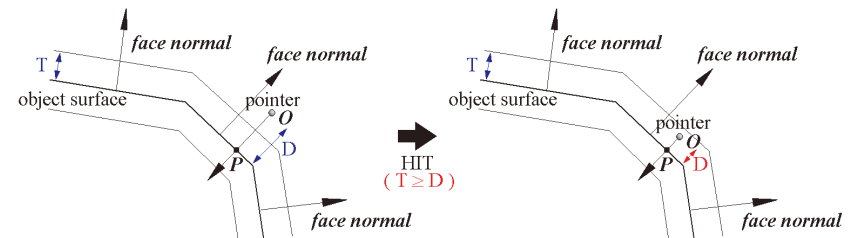


図 10 交差判定
Fig. 10 Intersection detection.

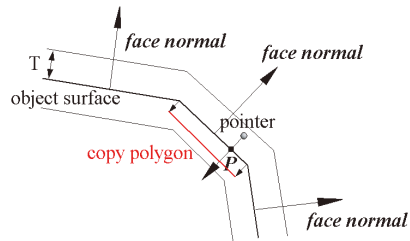


図 11 複製ポリゴンの作成
Fig. 11 Copy polygon creation.

$$f_j = N_d k_p L_j + N_d k_d \frac{L_j - L_{j-1}}{\Delta t} \quad (2)$$

ここで、 k_p はバネ定数、 k_d はダンパ定数、 Δt は力覚提示プロセスの更新周期とする。

5. 評価実験

本章では、提案システムの有効性を示すための実験について述べる。

5.1 実験条件

表 1 に評価実験に用いたモデルを示す。提案手法が、ローポリゴンモデルと距離マップおよび法線マップを併用することで、力覚提示にどのような効果が現れるかを調べるため、ポリゴンベースの手法のモデル 1 に加え、モデル 2~5 において、それぞれのマップを組み合わせたローポリゴンモデルを用いて比較を行った。提案手法はモデル 5 となる。ここで、ハイポリゴンモデルには、図 4(a) の 69,670 ポリゴンのモデル、ローポリゴンモデルには、図 4(d) の 635 ポリゴンのモデル、距離マップには、図 7(a) の 256×256 のテクスチャ、法線マップには、図 7(b) の 256×256 のテクスチャを用いた。また、モデル 1 の力覚提示では、立方体を $7 \times 7 \times 7$ で等間隔に区切ったボクセルモデルでハイポリゴンモデルを囲み、3D ポインタが侵入したボクセル内のポリゴン群に対して Constraints-based God-object Method¹⁶⁾ に類する反力計算を行うことで、交差判定にかかる計算コストを抑えるようにした。また、モデル 2~5 の力覚提示では、前章の反力計算法に基づき、距離マップをオンにする場合、3D オブジェクトの表面高さを変化させた複製ポリゴンを動的に生成し、オフにする場合、表面高さを変化させずに複製ポリゴンを動的に生成するようにした。また、法線マップをオンにする場合、法線マップの法線値に応じて反力方向を動的に変化させ、オフにする場合、複製ポリゴンの面法線ベクトル方向に反力を発生させるようにした。

表 1 比較条件

Table 1 Comparison conditions.

	model 1	model 2	model 3	model 4	model 5
graphic model	high	high	high	high	high
haptic model	high	low	low	low	low
distance mapping	off	off	on	off	on
normal mapping	off	off	off	on	on

表 2 計算時間結果

Table 2 Computation time results.

	total update count	average time [μ sec]
model 1	9,089	119.78
model 2	9,286	12.35
model 3	9,285	12.59
model 4	9,295	13.25
model 5	9,325	10.95

実験に使用した CPU は Xeon 2.4GHz \times 2, GPU は ATI FireGL T2-128M である。本実験では、1kHz の更新周期で図 9 の処理を行い、画面上的 3D ポインタを再描画するため、60Hz の更新周期で描画処理を行った。また、反力計算の際、式 (2) のバネ定数 k_p を 0.5、ダンパ定数 k_d を 0.5 として実験を行った。

5.2 絶対評価

絶対評価として、前節の実験条件でそれぞれ力覚提示を行い、反力発生時の図 9 のデバイスの先端位置の座標変換処理から反力計算までの計算時間の測定を行った。

表 2 に、計算時間を 10 秒間測定したときの力覚提示プロセスの総更新数および平均計算時間を示す。モデル 1~5 ともに十分な計算速度が得られていることが分かる。また、モデル 1 に比べ、モデル 2~5 では少ない計算時間で反力計算を行っていることが分かる。

5.3 主観評価

主観評価として、被験者 12 名に対し、前述の実験条件でそれぞれ力覚提示を行い、5 段階評価 (E1: できない, E2: ややできない, E3: どちらともいえない, E4: ややできた, E5: できた) でアンケートを行った。評価項目は次のとおりである。

- A) 実験用モデルへの力覚提示を滑らかに行うことができたか
- B) 実験用モデルの表面の高さの違いを感じる事ができたか
- C) 実験用モデルの表面の勾配の違いを感じる事ができたか

表 3 統計および検定結果
Table 3 Statistical and test results.
A) smooth rendering

	E1	E2	E3	E4	E5	significance probability	determination
model 1	3	7	1	0	1	0.0005563	impossible
model 2	0	2	2	5	3	0.0194640	possible
model 3	0	3	2	5	2	0.1193086	—
model 4	0	0	1	8	3	0.0000213	possible
model 5	0	0	3	4	5	0.0002995	possible

B) height

	E1	E2	E3	E4	E5	significance probability	determination
model 1	2	5	2	2	1	0.1196130	—
model 2	0	6	4	2	0	0.0949799	—
model 3	0	1	5	6	0	0.0290963	possible
model 4	0	1	2	8	1	0.0015396	possible
model 5	0	0	2	6	4	0.0000901	possible

C) gradient

	E1	E2	E3	E4	E5	significance probability	determination
model 1	1	4	4	3	0	0.4070426	—
model 2	0	6	5	1	0	0.0290963	impossible
model 3	0	3	3	5	1	0.2289211	—
model 4	0	0	4	3	5	0.0008785	possible
model 5	0	0	1	7	4	0.0000238	possible

また、アンケート結果の有意性について、「E3：どちらともいえない」から有意差があるかどうか Mann-Whitney の U 検定で検定（両側検定，有意水準 5%）を行った。

表 3 にアンケートの統計および検定結果を示す。滑らかさについてはモデル 3 を除いて有意差が見られた。モデル 1 の結果については、エッジ部分で突き抜けが多発したためだと思われる。高さの違いについてはモデル 1, 2 を除いて有意差が見られ、モデル 4 においては、実際に表面の高さは変化していないにもかかわらず、被験者はモデル 3 よりも高さの違いを感じていることが分かる。勾配の違いについてはモデル 1, 3 を除いて有意差が見られた。総合的な結果として、モデル 4 およびモデル 5 の評価が高く、両者を比べると、提案手法であるモデル 5 の方が、オブジェクト表面の高さを動的に変化させつつ反力を摂動させるためか、モデル 4 に比べ、高さの違いおよび勾配の違いともに有意確率が低い結果となった。

以上の結果より、従来手法に比べ、提案手法を用いることにより、ローポリゴンモデルが

らハイポリゴンモデルへの力覚提示を少ない計算時間かつ高精細に実現可能であることを確認することができた。

6. おわりに

本研究では、ローポリゴンかつテクスチャベースの力覚提示手法が、低コストかつポリゴン内部の凹凸感を滑らかに表現可能なことに注目し、距離マップおよび法線マップを用いたハイポリゴンモデルへの力覚提示手法の提案として、ハイポリゴンモデルおよびローポリゴンモデルから距離マップおよび法線マップを生成するソフトウェアの開発および各種マップを用いたローポリゴンモデルからハイポリゴンモデルへの力覚提示システムの開発を行った。本手法では、あらかじめハイポリゴンモデルおよびハイポリゴンモデルを粗く近似したローポリゴンモデルを用意し、ローポリゴンモデルの UV マップへ、ハイポリゴンモデルとローポリゴンモデル表面の距離、およびハイポリゴンモデルの頂点法線ベクトルを色付けることにより、距離マップおよび法線マップを生成する。生成した各種マップは、ローポリゴンモデルに対して UV マッピングを行い、交差判定はローポリゴンモデルに対して行う。3D ポインタが 3D オブジェクト表面に一定距離以内に近づいた場合をヒットしたと判定し、そのときの交点における距離マップの値に応じてローポリゴンモデルの表面高さを動的に変化させ、反力を法線マップの値に応じて摂動させることにより、低コストでハイポリゴンモデルへの力覚提示が可能となるかどうか評価実験を行い、提案手法の有効性を示すことができた。

今後の課題として、凹形状への対応、プログラミング可能な GPU^{1),2)} を転用したより柔軟かつ高速な力覚提示の実現、オブジェクト表面の質感表現、UV およびテクスチャ編集機能の実装などがあげられる。

参考文献

- 1) 宮田一乗, 高橋誠史, 黒田 篤: GPU コンピューティングの動向と将来像, 芸術科学会論文誌, Vol.4, No.1, pp.13-19 (2005).
- 2) Watt, A.H. and Policarpo, F.: *Advanced Game Development With Programmable Graphics Hardware*, p.384, A.K. Peters Ltd. (2005).
- 3) Sander, P.V. and Mitchell, J.L.: Progressive buffers: view-dependent geometry and texture LOD rendering, *ACM SIGGRAPH 2006: Proc. Conference on SIGGRAPH 2006 Course Notes, Course 26, Advanced Real-Time Rendering in 3D Graphics and Games*, pp.1-18 (2006).

- 4) Wrotek, P., Rice, A. and McGuire, M.: Real-Time Collision Deformations Using Graphics Hardware, *Journal of Graphics Tools*, Vol.10, No.4, pp.1–22 (2005).
- 5) Ruspini, D.C., Kolarov, K. and Khatib, O.: The Haptic Display of Complex Graphical Environments, *Proc. 24th Annual Conference on Computer Graphics and Interactive Techniques*, pp.345–352 (1997).
- 6) Otaduy, M.A. and Lin, M.C.: Sensation Preserving Simplification for Haptic Rendering, *Proc. ACM SIGGRAPH 2003/ACM Trans. Graphics*, Vol.22, No.3, pp.543–553 (2003).
- 7) 浅野寿朗, 石橋 豊, 峯澤聡司: 触る分散仮想博物館における展示オブジェクト表示と解説情報ストリーミング開始の適応型制御, *日本バーチャルリアリティ学会論文誌*, Vol.11, No.1, pp.27–38 (2006).
- 8) Morgenbesser, H.B. and Srinivasan, M.A.: Force Shading for Haptic Shape Perception, *Proc. ASME Dynamics Systems and Control Division*, Vol.58, pp.407–412 (1996).
- 9) Shopf, J. and Olano, M.: Procedural haptic texture, *Proc. 19th Annual ACM Symposium on User Interface Software and Technology*, pp.179–186, ACM Press (2006).
- 10) Stanney, K.: Haptic Rendering In Virtual Environments, *Handbook of Virtual Environments*, Basdogan, C. and Srinivasan, M.A. (Eds.) pp.117–134, Lawrence Erlbaum Associates, Inc. (2001).
- 11) Theoktisto, V., Fairén, M., Navazo, I. and Monclús, E.: Rendering detailed haptic textures, *2nd Workshop in Virtual Reality Interactions and Physical Simulations (VRIPHYS 05)*, pp.16–23 (2005).
- 12) 下条 誠, 仁木 亨: 高速応答が可能な可搬型ハプティクスディスプレイの開発, *電子情報通信学会論文誌 D-II*, Vol.J87-D-II, No.5, pp.1133–1141 (2004).
- 13) 脇田 航, 三谷広志, 井門 俊: ハイトマップ及びノーマルマップを用いた擬似凹凸への力覚提示, *FIT2007 第 6 回情報科学技術フォーラム情報科学技術レターズ*, Vol.6, No.LK-004, pp.347–350 (2007).
- 14) Mark, W.R., Randolph, S.C., Finch, M., Verth, J.M.V. and Taylor II, R.M.: Adding Force Feedback to Graphics Systems: Issues and Solutions, *Proc. SIGGRAPH '96*, pp.447–452 (1996).
- 15) Massie, T.H. and Salisbury, J.K.: The PHANTOM Haptic Interface: A Device

for Probing Virtual Objects, *Proc. ASME Winter Annual Meeting, Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, DSC-Vol.55-1, pp.295–301 (1994).

- 16) Zilles, C. and Salisbury, K.: A constraint-based God-object method for haptic display, *Proc. IEEE/RSJ International Conference Intelligent Robots and Systems*, pp.145–151 (1995).
- 17) Woo, M., Neider, J., Davis, T. and Shreiner, D.: *OpenGL Programming Guide: The Official Guide To Learning OpenGL, Version 2*, p.896, Addison-Wesley (2005).
- 18) Möller, T. and Trumbore, B.: Fast, Minimum Storage Ray-Triangle Intersection, *Journal of Graphics Tools*, Vol.2, No.1, pp.21–28 (1997).

(平成 19 年 10 月 9 日受付)

(平成 20 年 4 月 8 日採録)



脇田 航

平成 16 年愛媛大学工学部情報工学科卒業。平成 18 年同大学大学院理工学研究科情報工学専攻博士前期課程修了。現在、同大学院理工学研究科電子情報工学専攻博士後期課程在学中。3DCG, バーチャルリアリティ等の研究に従事。



井門 俊 (正会員)

平成 8 年東京工業大学大学院総合理工学研究科博士後期課程単位取得退学。現在、愛媛大学大学院理工学研究科講師。画像符号化, バーチャルリアリティ等の研究に従事。博士(工学)。