

触覚ディスプレイを用いた 3 次元形状変形システムの構築

田村博之^①, 伊藤渉^②, 徳山喜政^①, 今野晃市^③東京工芸大学工学部^①, コナミエンタテインメント^②, 岩手大学工学部^③

1. はじめに

従来のモデリングでは、マウスやペンタブレットなどの 2 次元座標入力装置を使用して形状変形を行っているが、その方法では直観性と対話性に欠け、操作になれるまでは非常に扱いにくい。

本研究では 3 次元座標入力装置による入力で、直観的かつ対話的に形状変形を実現するシステムの構築を行う。3 次元での形状変形に力覚のフィードバックを加えることで視覚と触覚の融合を図り、より直観的に操作できるシステムを目指す。本稿ではシステムの設計や、いくつかの機能について報告する。

2. PHANTOM について

本研究では SensAble 社の触覚ディスプレイ「PHANTOM Premium 1.5/6DOF-SM」を用いてシステムを構築する(図 1)。本システムは、コンピュータ上の仮想オブジェクト(CGモデル)に触れると、形状や質感などを計算し、それに応じた摩擦力や抵抗などの力をユーザ側にフィードバックするシステムである。PHANTOM の動作は OpenHaptics というライブラリを使用して制御する。

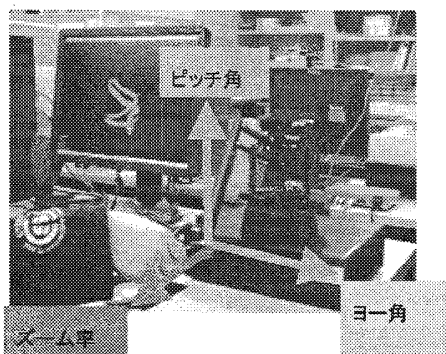


図 1 PHANTOM 力覚提示装置

3. システム設計図

本システムは、C++ 言語と OpenGL をベースに開発されている。図 2 はシステムの設計図を示す。

CAppClass クラスでは、メンバ関数であるウィンドウプロシージャが実質アプリケーションの本体であり、そこから渡されたメッセージに応じて各オブジェクトへ命令を出す。また、CModel クラスや各変形マネージャ・状態クラス・描画クラスなどの保持・管理も行う。

モデルのデータは CModel クラスに格納される。モデルは複数の立体(CBody)で構成され、立体は複数の頂点(CVertex)、稜線(CEdge)、面(CLoop)で構成される。頂点・稜線・面の移動に伴うモデルデータの更新はそれぞれ CPointManager, CEdgeManager, CLoopManager クラスが行う。

PHANTOM を制御するために必要な HapticsDevice の生成・削除と制御を CHapticManager クラスが行う。

描画に必要な情報(ビューポート、画角等)は CViewManager クラスが保持・管理する。視点の移動は CCamera クラスだけでできるようにする。ユーザーがカメラを移動するときは、CViewManager が持つ CCamera オブジェクトを取得し、カメラクラス内の位置、注視点、ヨー・ピッチ・ロール角等の値を操作する事で、視点の変更を行う。描画系を CViewManager と CCamera に分ける事により、プログラミングのし易さ・ソースの可読性の向上に繋がる。

アプリケーションは、「物体に触れる状態」、「オブジェクト選択状態」、「頂点移動中の状態」、などの状態をもつ。これらの状態は IStateBase インタ

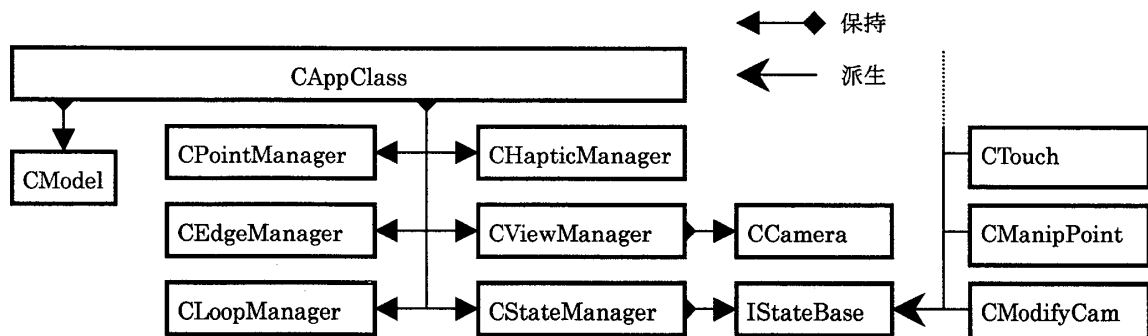


図 2 システム設計図

ーフェイスを持つ CTouch, CManipPoint, CModifyCam などのクラスで表現され, インスタンスの登録や状態の変更などを StateManager クラスが行う. そして状態を管理するために, スタックベースの管理と, クエリーシステムを統合した“スタックベースの状態マシン”を導入する[1][2].

4. PHANTOM によるカメラ操作

PHANTOM の Proxy の位置と姿勢をキャプチャすることによって, 3次元座標入力が可能となる(図1). 本システムでは, x 軸方向でカメラのヨー角, y 軸方向でカメラのピッチ角, z 軸方向でズーム率を同時に変更できるように実装する. これらのカメラ操作を同時に行うことで, 従来のマウス等による2次元座標入力デバイスでの操作法では得られなかった, 高い直観性を得ることができる.

5. PHANTOM による頂点・稜線・面の移動

移動オブジェクト選択時, 頂点・稜線・面, それぞれに吸着する様な力がかかることにより, 直観的にオブジェクトを選択できるようになる.

しかし3軸方向に位置と姿勢を自由に変更できる操作法では, 2つの軸方向を固定して, 1つの軸方向だけを変更するような操作は, 苦手である. この問題を解決するために, カーソルを吸着する力が働く x y z 軸を, 編集を開始したときの座標を原点として描画する. これにより, ある軸方向だけの移動と自由な移動の両立が可能になる(図3).

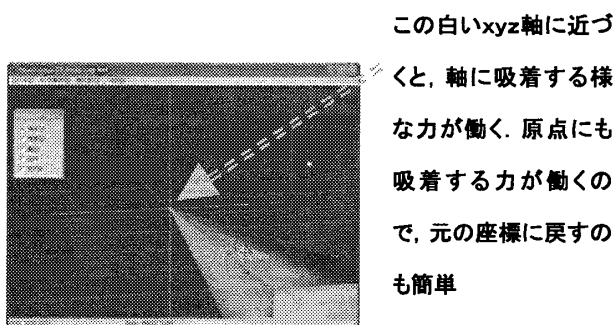


図3 オブジェクト選択時の工夫

オブジェクトの移動時, カメラに対する奥行き方向の移動量の判別が難しい(図4左). ライティングによる立体表面の光の加減や, 各軸上でのカーソルの位置表示など, 視覚的に分かりやすくする処理は行うことができるが, 感覚的に分かりづらいため. そこで, オブジェクトの元の座標からバネで引かれるような力覚をフィードバックすることで視覚と力覚の融合を図り, より直観性を向上させる(図4右). 図5は, 犬のキャラクターの胸部

近傍の頂点を移動した後の細分割曲面形状である.

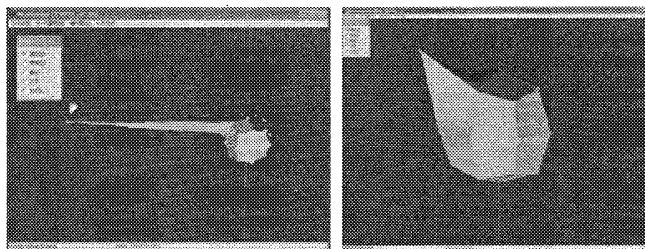


図4 前後方向の移動

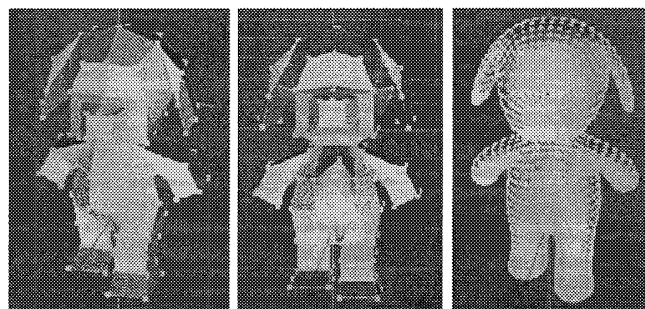


図5 頂点の移動例

6. PHANTOM とマウスクーソルの同期とインターフェースの実装

本システムは, アプリケーション中の PHANTOM デバイスの位置と, ウィンドウズカーソルの位置を同期させ, PHANTOM でマウスクリックが行えるように実装している. これにより PHANTOM デバイスでメニューなどのインターフェース操作が可能になる.

7. まとめと今後の課題

触覚ディスプレイを用いた3次元形状変形システムの内部構造や機能について述べた. そして, 数人の被験者に本システムを用いて形状変形の実験を行なってもらった結果, 全員が力覚のフィードバックがある場合の方が, 操作性・直観性において優れていると回答した.

今後の課題としては, インタフェースの充実, より高度な力覚の制御, カーソルの視認性の向上, 変形やモデリング機能の充実などが挙げられる.

参考文献

[1] Kim Palliser, Game Programming GEMS 日本語版, BornDigital, 2001.
 [2] Mark DeLoura, Game Programming GEMS 5 日本語版, BornDigital, 2006.