

AR空間における立体モデルの集合演算と局所変形 Set Operation and Local Deformation of Solid Model in AR Space

阿部 直樹[†] 向井伸治[†] 福崎健志[†]

Naoki Abe Shinji Mukai Takeshi Fukuzaki

1. はじめに

近年、現実世界の映像に計算機によって生成された情報や、3次元オブジェクトをオーバーレイ表示するAR(Augmented Reality: 拡張現実感)技術が注目を集めている。この技術は新しいユーザインタフェースの構築や、作業支援、情報提示に役立てることを目的にさまざまな分野での活用が期待されている。また、AR技術を活用した研究として、AR空間上で立体モデルを生成するシステムが報告されている。これらの研究の目的に共通して言えることは、立体モデリングを簡単に行えるようなインタフェースやモデリング手法を提案している点である。しかし、生成した立体モデルに対し、AR空間上で形状編集操作を行うことが可能なシステムはあまり多くない。

本研究では、立体モデリングの簡便化を図るために、あらかじめ作成しておいた基本形状をAR空間上に配置していき、これに対し形状編集操作として集合演算と局所変形を繰り返し行うことで任意の形状を生成する立体モデリングシステムの開発を試みる。本システムにおける立体モデルのデータ構造は、クリスタル構造[1][2]を採用している。これにより、形状の編集履歴を少ない情報量で保持することが可能である。

2. 関連研究

AR空間で立体モデルを生成するシステムがいくつか報告されている。

Manfredらは、AR空間上に基本形状を配置していき、家具のモデルを生成するシステム[3]を提案している。この基本形状には円柱や板などがあり、これらと同じ大きさの実際の木材にARマーカを取り付け、この上に立体モデルが表示される。これにより、基本形状を配置する際に立体モデルに直接触れているかのような力覚的フィードバックを得ている。しかし、あらかじめ用意された基本形状では生成不可能なモデルが存在するという問題点がある。

青木らは、エアブラシ型入力インタフェースを用いてボクセルを塗料のように吹き付けることでモデリングを行うシステム[4]を提案している。ボクセルの吹き付け、消去、彩色によって簡易なモデリングを行うことができるシステムとなっている。しかし、オブジェクトの一括切断や、やり直し機能など、モデリングを効率的に行うための諸機能が充実していないという問題点がある。

これらのシステムに共通しているのは、一般的なCADシステムのように、2次元図形をスケッチしてから3次元立体モデルを生成するような方法を取っていない点である。本システムはこれを参考に、モデリングの簡易化のために3次元立体モデルをAR空間上に配置していくことで、モ

デルを生成していくシステムを開発した。

3. システム構築のための要素技術

3.1 ARToolKit

本システムにはC/C++用プログラミングライブラリであるARToolKit[5]を使用した。これにより、カメラから画像を取得し、その画像からマーカを検出しそのパターンを認識することができる。また検出したマーカの3次元位置・姿勢をリアルタイムで計測し、この情報を元に現実世界の映像に立体モデルをオーバーレイ表示することができる。これにより、マーカ上もしくはその周囲に立体モデルが存在しているように感じることができる。

複数マーカ間の座標変換行列を取得する方法について考える。ARToolKitにおいて、マーカのカメラに対する3次元位置・姿勢は、式(1)のような回転成分 r と並進成分 t を含んだ変換行列 T として得られる。マーカ1とマーカ2の変換行列をそれぞれ T_1, T_2 とし、マーカ1とマーカ2の座標系をそれぞれ $[X_1 Y_1 Z_1 1]^T, [X_2 Y_2 Z_2 1]^T$ とするとマーカ1におけるマーカ2の座標系は式(2)のようになる。

$$T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

$$\begin{bmatrix} X_2 \\ Y_2 \\ Z_2 \\ 1 \end{bmatrix} = T_1^{-1} T_2 \begin{bmatrix} X_1 \\ Y_1 \\ Z_1 \\ 1 \end{bmatrix} \quad (2)$$

これにより、あるマーカを基準の座標系とした場合に、その座標系における他のマーカの3次元位置・姿勢を取得できる。

3.2 クリスタル構造

一般的に立体モデルはハーフエッジ構造で記述されることが多いが、本システムにおける立体モデルのデータ構造にはクリスタル構造を採用している。クリスタル構造とは、各頂点に対して隣接する点の情報を与えるだけのものであり、単純で冗長度が少ないデータ構造となっている。平面と曲面が混在する立体であっても同一のデータ構造で記述することができる。

また、点に何らかの変更が加えられた箇所限定して履歴の更新が可能のため、全ての操作履歴を記憶させておいても、メモリ量の深刻な増加にはつながらない。このことから、形状演算や局所変形を頻繁に行う本システムにとってクリスタル構造は最適であると考えられる。

[†] 前橋工科大学 Maebashi Institute of Technology

4. 提案システム

4.1 システムの構成

本システムは、ビデオシースルー型ヘッドマウントディスプレイ、PC、キーボード、USB マウスをベースに作成した入力デバイス、基準マーカ、操作マーカで構成される。図 1 は実際にシステムを使用している様子を示している。

本システムで使用した PC は、Dell 製 Optiplex 9010(CPU : Intel Core i7-3770 3.40GHz, Memory : 8GB, OS : Windows 7 Professional 64bit), ビデオシースルー型ヘッドマウントディスプレイは、VUZIX 製 Wrap1200AR を使用した。開発環境は、Microsoft Visual Studio Professional 2012, 立体モデルの表示には、OpenGL を使用した。

4.2 モデル生成の流れ

本システムにおける立体モデル生成のイメージを図 2 に示す。まず初めに、ユーザは AR 空間上に基本形状を 1 つ配置する。この配置された基本形状に対し、ユーザは局所変形を行うかどうかを選択する。もし局所変形を行う場合は、後述する本システムにおいて実行可能な局所変形操作を加えることで、新たな形状へと更新することができる。局所変形を行わない場合は、すでに配置されている基本形状と重なり合うように新たな基本形状を配置することで、集合演算を行うことができる。これにより新たな形状へと更新される。そして、新たに生成された形状に対してさらに集合演算と局所変形を行うことで任意形状を生成することができる。



図 1 実際にシステムを使用している様子

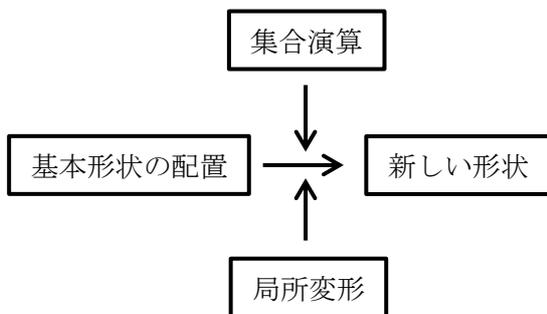


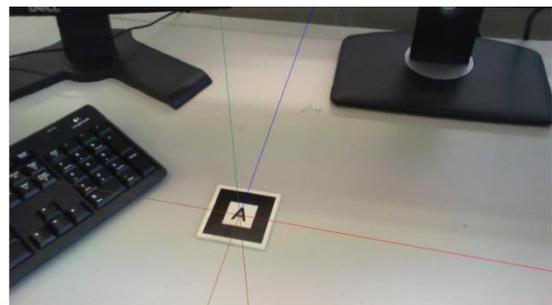
図 2 立体モデル生成のイメージ

4.3 基本形状の配置

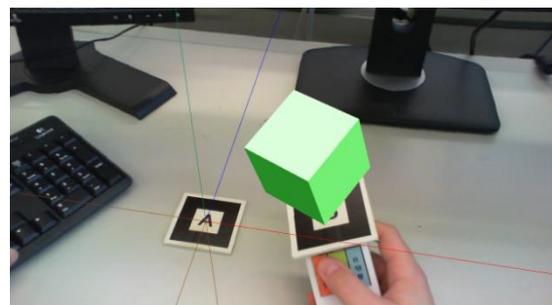
本システムにおける AR 空間上での立体モデリングは、基準マーカの中心を原点とした 3 次元座標で行われる。図 3 は基本形状の配置の流れを示している。まずユーザは入力デバイス进行操作し、配置する基本形状を選択する。その後、入力デバイスに取り付けられた操作マーカを任意の位置・姿勢へ移動し、入力デバイスの決定ボタンを押すことで基準マーカ座標系へ配置される。このとき操作マーカ座標系に存在している基本形状は、式(2)より基準マーカ座標系へと変換される。これにより、ユーザは視覚的な情報を頼りに、直感的に基本形状を配置できる。

4.4 集合演算

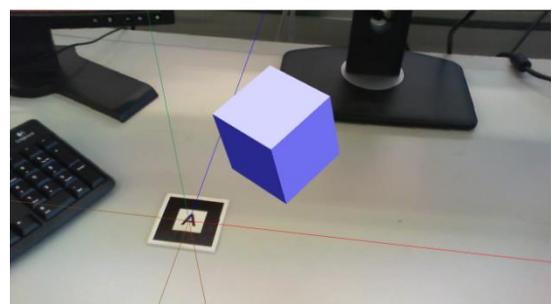
基本形状の単純な配置だけでは生成可能なモデルに限られるため、集合演算を行うことでこれを解決する。基準マーカ座標系にすでに存在している形状と、新たに配置する基本形状が重なり合うように配置することで、形状同士の和・差・積を取ることができる。図 4 は集合演算の例を示している。



(a)基本形状が未配置の状態

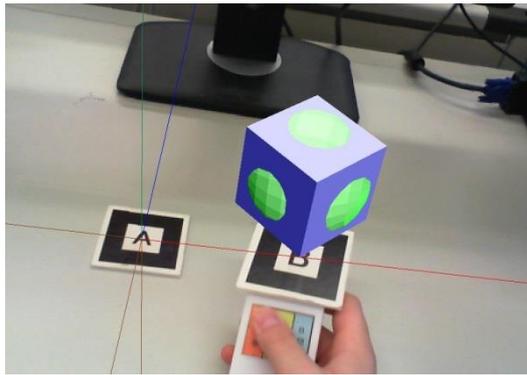


(b)任意の位置へ操作マーカの移動

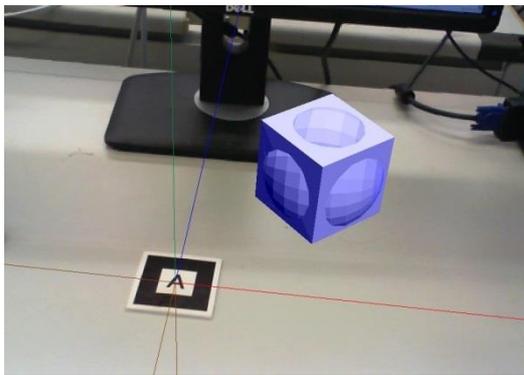


(c)基本形状配置が完了

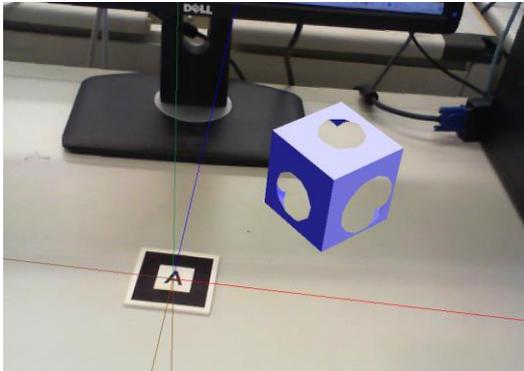
図 3 基本形状の配置の流れ



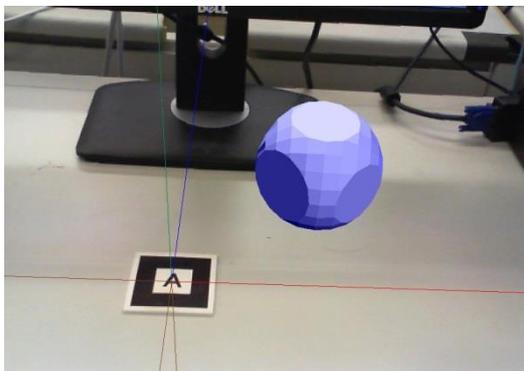
(a)基本形状の配置



(b)和



(c)差

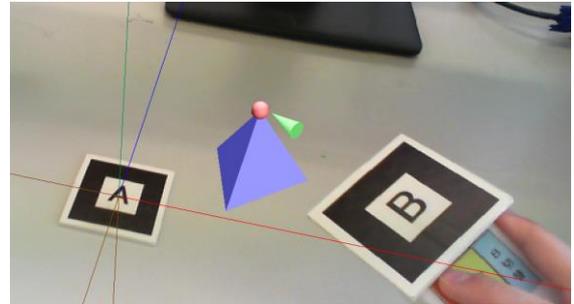


(d)積

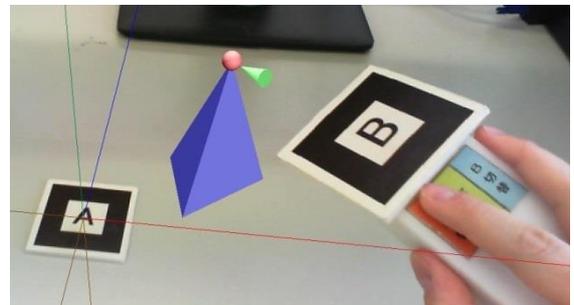
図4 集合演算の例

4.5 局所変形

本システムではより直感的な形状の編集を行うために、局所変形が実行可能である。局所変形操作として、頂点・稜線の削除、頂点・稜線・面の移動、面取り、フィレット、面の細分化、形状の切断が実行可能である。図5、6は局所変形の例を示している。局所変形を実行する部位の決定

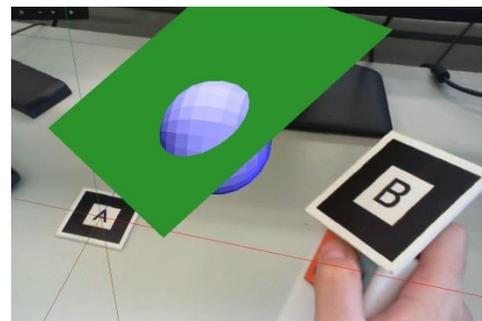


(a)頂点の選択

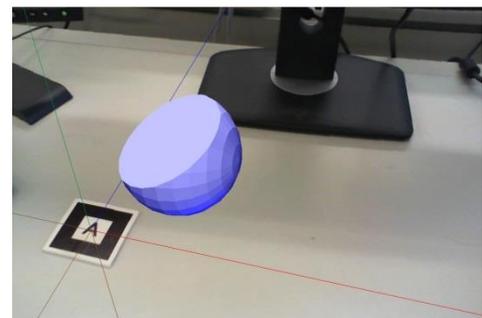


(b)頂点の移動

図5 局所変形の例1(頂点の移動)



(a)切断面の配置



(b)切断された形状

図6 局所変形の例2(形状の切断)

には仮想カーソルを使用する。仮想カーソルは操作マーカに表示される。仮想カーソルの先端の座標値に最も近い頂点・稜線・面を選択することができる。頂点・稜線・面を移動させる場合、移動箇所を選択した状態で入力デバイスの決定ボタンを押した座標から、決定ボタンを離した座標への移動量分だけ、移動させることができる。また形状を切断する場合は、操作マーカに切断面を表示し、これと切断対象となる形状が重なり合うように配置しなければならない。切断面を配置後に入力デバイスの決定ボタンを押すことで切断が実行される。本システムにおける切断とは、切断面の正領域側か負領域側に存在する、形状の一部を削除することである。どちら側を削除するかは任意に選択することが可能である。

4.6 履歴操作

本システムは、集合演算と局所変形を頻繁に行うことから、undo/redo操作が可能である。ユーザは形状に不満がある場合や、操作を間違えた場合に形状の履歴をさかのぼりやり直すことが可能である。

5. 作成例とシステムの評価

本システムを用いて生成したモデルの例を図7, 8に示す。図7のモデルは和、差の集合演算、頂点の移動を行うことで生成した。図8のモデルは和の集合演算、頂点の移動、形状の切断、面の細分化を行うことで生成した。

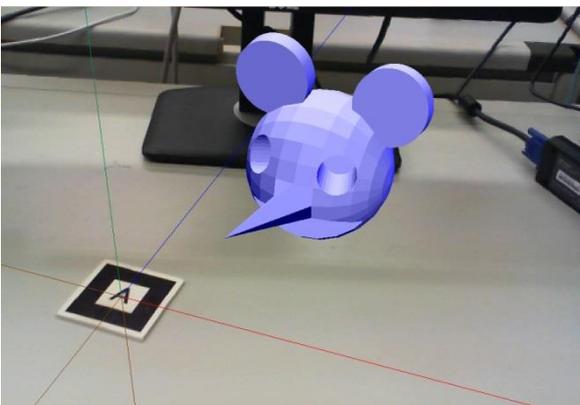


図7 モデル生成例1

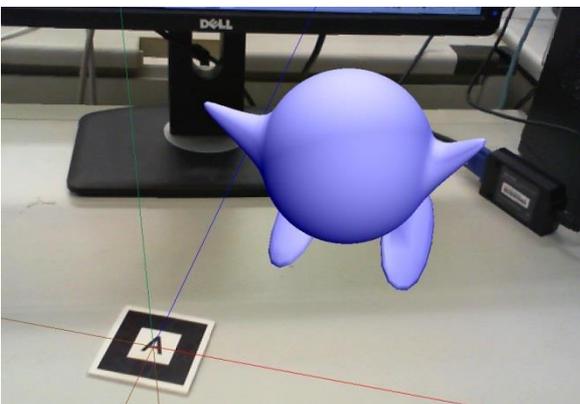


図8 モデル生成例2

本システムの使い易さについて評価するために、立体モデリング未経験者を含む20名の学生を対象に簡易ユーザテストとアンケート調査を行ったところ、立体モデリング未経験者でも本システムの特徴をすぐに理解し、操作方法を習得できていたことがわかった。

6. まとめと今後の課題

本研究では、基本形状を直感的にAR空間上に配置していくことで立体モデルを生成するシステムを開発した。また、形状編集操作として集合演算と局所変形が実行可能である。これにより、基本形状の組み合わせだけでも、多様な立体モデルを生成可能である。基本形状の配置や局所変形には、基準マーカに対する操作マーカの位置・姿勢を利用しているため、ユーザは操作マーカを任意の位置に動かすことで形状編集操作が可能となっている。履歴操作も可能となっているため、ユーザは試行錯誤を繰り返しながら立体モデリングを行うことができる。

本システムの評価のために、簡易ユーザテストとアンケート調査を行ったところ、立体モデリング未経験者でも本システムを扱えていたことがわかった。このことから本システムは初心者でも扱え、簡易モデルの生成や、実際の設計作業の前段階である、大まかな形状のイメージの検討などに役立つと考えられる。

本システムはキーボードからの入力によって一部の機能の操作が行われている。より直感的な操作を可能とするために、ARマーカの動きやジェスチャなどを認識し操作できるようにしていきたい。また、基本形状を配置する位置は、ユーザが目分量で決定していることから、整った寸法を持ったモデルの生成ができず、以前に生成したモデルと完全に同じものを生成する事は困難である。ゆえに今後はこれを解決する方法を検討していきたい。従来のシステムとの比較による、本システムの定量的評価も行う予定である。

参考文献

- [1] 清水誠司, 向井伸治, 古川進, “多様体ソリッドモデル記述用の新しいデータ構造について”, 日本設計工学会誌, Vol.42, No.1, pp.53-60(2007).
- [2] 清水誠司, 向井伸治, 古川進, “二次元・三次元図形の統一的データ表現と形状変形履歴の操作”, 電子情報通信学会論文誌, Vol. J90-A, No.10, pp.750-757(2007).
- [3] M.Lau, M.Hirose, A.Ohgawara, J.Mitani, T.Igarashi, “Situating Modeling: A Shape-Stamping Interface with Tangible Primitives”, Proc. of 6th Int. Conf. on Tangible, Embedded and Embodied Interaction, ACM, pp.275-282(2007).
- [4] 青木紘史, 三谷純, 金森由博, 福井幸男, “AR空間におけるエンプラシ型インタフェースによる形状モデリングシステム”, 映像情報メディア学会技術報告, Vol.36, No.16, pp.1-4 (2012).
- [5] 加藤博一, “拡張現実感システム構築ツール ARToolKitの開発”, 電子情報通信学会技術報告, PRMU, Vol.101, No.652, pp.79-86 (2002).
- [6] 橋本直, “ARToolKit 拡張現実感プログラミング入門”, アスキー・メディアワークス (2008).
- [7] 谷尻豊寿, “拡張現実感を実現する ARToolkit プログラミングテクニック”, カットシステム (2008).