I-037

点群データを用いた三次元 CAD に関する研究
A Study on Point Clouds modeler

山口 輝† 西尾 孝治† 小堀 研一†
Hikaru Yamaguchi Koji Nishio Ken-ichi Kobori

1. はじめに

近年，CG 技術の発展に伴い，さまざまな分野で三次元形状が用いられている．特に工業分野では三次元 CAD と呼ばれるコンピュータ設計支援システムが多用されており，これらのシステムには曲線や曲面で表現された三次元形状を用いる場合が多い．しかし，従来の三次元 CAD に用いられる曲線や曲面はデータ構造が複雑であり，編集に制約があるため，直観的な操作が難しい問題がある．

そこで，直観的な操作が容易な形状表現として離散データが注目されている．離散データで表現される三次元形状とは，点やボクセルといった単一の要素の集まりで形状を表現したもので，データ構造が単純であるため編集に制約がなく，直感的な編集が容易となる．

ところで，点群データは点の集まりで形状を表現した離散データで，局所的に自由に解像度を変更できるため，他の離散データに比べ高い精度で形状を表現できる．また点群データには，レンジファインダなどの三次元計測機器を用いて容易に実物体を再現したデータを得ることができ，さらに，隣接関係を持たないため GPU などの並列演算装置に処理を分担させやすいといった利点もある．点群データを用いたモデリング手法はいくつか提案されている[1][2]が，大半が自由変形の手法を提案しており，従来の三次元 CAD の機能を意識しているものは少ない．

そこで，本研究では従来の三次元 CAD と同様のモデリング機能を用いて，点群データの定義や編集を行うシステムを提案する．その際，対話的なモデリングを目標し，100 万点の形状に対し，すべての編集機能において 0.5 秒以内に処理ができるよう，システムの構築を行う．

2. システムの概要

本システムで提案する機能を図 1 に示す．同図に示すように本システムの機能は大きく作成，選択，編集の 3 種類に分けられ，従来の三次元 CAD の基本的なモデリング機能を参考にそれぞれの機能を実現する．

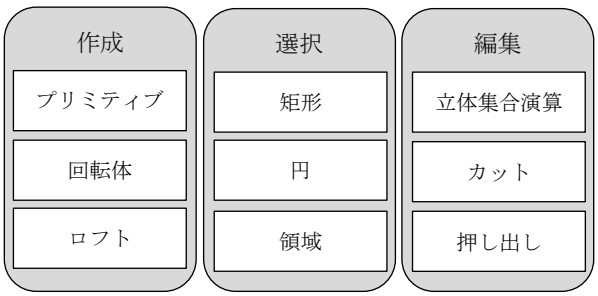


図 1 本システムの機能概要

†大阪工業大学

2.1 作成

作成機能では，編集を行う基となる形状の作成を行う．

1) プリミティブ
直方体や円柱，球，などの基本立体を作成する機能．
2) 回転体
曲線を任意の軸で回転させ，立体を作成する機能．
3) ロフト
複数の平面図形を連続させ，立体を作成する機能．

2.2 選択

選択機能では，形状表面の局所的な選択を行う．

1) 矩形
矩形内の形状表面を選択する機能．
2) 円
円内の形状表面を選択する機能．
3) 領域
形状の面領域を選択する機能．

2.3 編集

編集機能では，形状をユーザの意図した形に編集を行う．本システムでは，三次元 CAD の基本的な処理の中から 3 種類の機能を提案する．

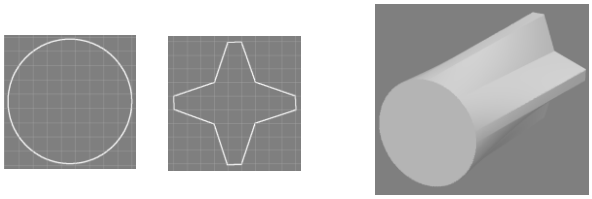
1) 立体集合演算
二つの形状に対して立体集合演算の和，差，積いずれかを行い，一つの形状にまとめる機能．
2) カット
指定された切断面から形状を二つに分ける機能．
3) 押し出し
曲線に沿って形状の表面を押し出す機能．

3. システムの構築

実際に点群データによってシステムの構築を行った．いくつかの機能について，実行結果と実現方法を示す．

3.1 ロフト

図 2 に図形 A と B を用いて作成機能のロフトを行った結果を示す．



(a) 図形 A (b) 図形 B (c) 結果
図 2 ロフトの結果

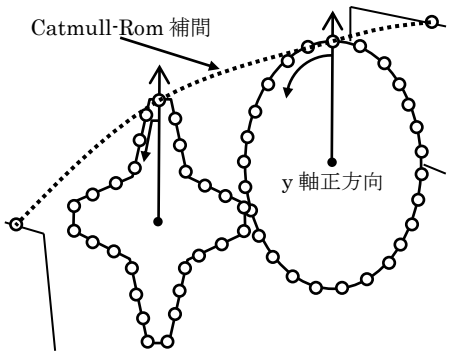


図 3 図形間の補間

ロフトでは，まず図 3 に示すように各図形で y 軸正方向に交わる交点から開始し，図形上に一定間隔で点を生成し前後の図形の同様の点と結びつける．そして，結びつけた点間を Catmull-Rom 補間することで，滑らかに図形を連続させている．

3.2 領域選択

図 4 に選択機能の領域選択を行った結果を示す．同図中の色が濃い部分が選択された領域である．

(a) 対象形状　　(b) 選択結果

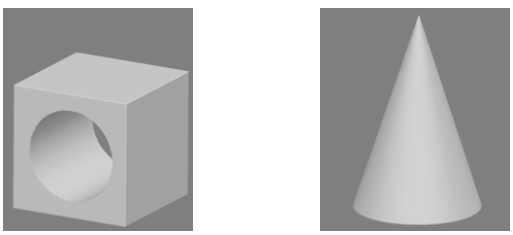
図 4　領域選択の結果

領域選択では，ユーザが指定した同図(a)に黒く示す形状表面上の一点から，連続した領域を選択する．連続しているとは，隣接する点の法線ベクトルの角度が閾値以下であることを示し，指定した一点から角度が閾値以下になるまで，徐々に選択領域を広げていくことで同図(b)のように連続した領域を選択する．

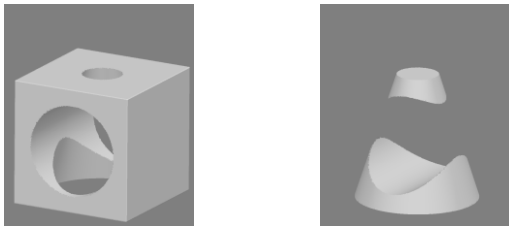
本システムでは，隣接する点の探索を高速化するためにオクトツリーを用いている．そのため，形状の定義や編集の度にオクトツリーを再構築している．

3.3 立体集合演算

図 5 に編集機能の立体集合演算を行った例を示す．

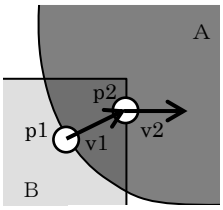


(a)　形状A　　(b)　形状B



(c)　A－B　　(d)　A∩B

図5 立体集合演算の結果



p1：形状 A のある点
p2：形状 B の p1 の最近隣点
v1：(p2-p1)のベクトル
v2：p2 の法線ベクトル

v1・v2 =>0：外　v1・v2 <0：内

図 6 最近隣点探索による内外判定

立体集合演算では，図 6 のように形状 A のある点 p1 が形状 B の内外どちらにあるかを，形状 B の最近隣点 p2 の法線ベクトル v2 と p1 と p2 を結ぶベクトル v1 の内積から判定している．同様の処理を形状 B にも行い，その情報からユーザの指定した演算にしたがって一つの形状にまとめている．

4. 実行結果

今回，処理を高速化するにあたって，一部の処理を NVIDIA 社の CUDA[3]を用いて GPU で行った．

本システムを用いてモデリングを行った一例を図 7 に示す．なお，モデリングを行った環境は CPU：Intel Core i5 2.67GHz，GPU：NVIDIA GeForce GTX 260 2.0GB，メモリ：4.00GB である．

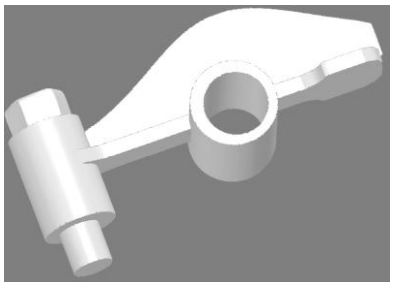


図 7 モデリング結果

すべての機能を点群で実現し，約 100 万点の形状に対して，0.5 秒以内に処理が終了していることから，本システムを用いることで直観的かつ対話的にモデリングが行えると考えられる．

今後の課題として，モデリング機能の拡充が挙げられる．

参考文献

[1] Mark Pauly, Richard Keiser, Leif P. Kobbelt, Markus Gross, "Shape modeling with point-sampled geometry", Proceedings of ACM SIGGRAPH 2003, pp.641-650(2003)
[2] Mario Botsch, Leif Kobbelt,"Real-Time Shape Editing using Radial Basis Functions", Computer Graphics Forum, Vol.24, pp.611-621 (2005)
[3] NVIDIA, "NVIDIA CUDA programming guide version 3.0", 2010