

サーフェス型吹き付けモデリングシステム

水野 隆行† 田中 敏光‡ 大西 昇†

†名古屋大学工学研究科 ‡名古屋大学大型計算機センター

Abstract

吹き付けモデラとは、仮想空間に置いたスプレーから物体表面に粒子を吹き付けることで物体を削ったり盛り上げたりする立体造形システムである。この手法は、サンドブラストと呼ばれる現実の加工法をまねることで、仮想空間に手軽でリアルなモデリング環境を実現している。

これまでの吹き付けモデラは Z-List という構造を用いて物体の形状を記述していた。Z-List は、Z 方向を高さを選んだ正四角柱の集まりとして物体を表現するので、吹き付け方向を Z 軸に選ぶことで、高速な形状変形が可能である。しかし、吹き付け方向を変えるたびに形状をサンプリングしなおさなければならないため、量子化誤差により物体表面に不必要な凹凸が生じてしまう。

本研究では、Z-List 記述とサーフェス記述の長所を融合して、高速かつ回転による変形を生じない吹き付けモデリングを提案する。具体的には、物体表面をポリゴンで記述することで、再サンプリングを不要にする。ただし、吹き付けで移動する頂点や変形されるポリゴンを効率よく見つけるため、Front-List と Polygon-List を導入する。作業空間は Z-List と同様に z 軸方向に伸びた正四角柱に分割される。部分空間毎に、そこに含まれるポリゴンが Polygon-List に、粒子が到達し得る頂点が Front-List に記録される。これらのリストを用いることで、従来のシステムと同様に実時間モデリングが実現できた。

Spray modeling system for polygonal surfaces

Takayuki Mizuno †, Toshimitsu Tanaka ‡, Noboru Ohnishi †

† School of Engineering, Nagoya University, ‡ Computation Center, Nagoya University

Abstract

"Spray modeler" is an interactive 3D modeling system that shaves and swells objects by spraying virtual particles onto their surfaces. This method achieves the simple but realistic modeling environment by following the real method called "Sand blast".

Previous spray modeler described the object shape by using the "Z-List" structure. The Z-List approximates an object as a set of square pillars along the Z-axis. The shape description is better for fast modeling, however, roughness caused by re-sampling appears on object surfaces whenever the direction of spraying was changed.

This paper presents a fast and aliasing free spray modeling system. Object surfaces are described with polygons, so that the re-sampling is not needed. To accelerate modeling speed, object space is segmented on the x-y plane like the Z-list. In each subspace, non-occluded polygon vertices are primarily selected and saved in a Front-list. Thus, object surfaces are quickly modified with moving the vertices in the list. In addition, polygons intersecting to each subspace are saved in the Polygon-list. The list is used to reduce the cost for making holes. As a result, the processing speed equal with an old system could be achieved.

1. はじめに

近年、3次元CGの利用分野が広がるにつれ、直感的に立体形状を作りたいという要求が高まっている。しかし、3次元CADで一般的な三面図入力やCGモデラで使われているプリミティブの組み合わせによる形状定義では形状の変形に手間がかかるため、曖昧にしか形を捉えていない物体を立体化する目的には向いていない。B-SplineやBezier曲面を使うと制御点の移動による連続的な形状変形が可能だが[1]、曲面どうしの交線計算を必要とする穴あけ操作を実時間処理することは難しい。

実世界の彫刻や粘土細工では、単純な変形操作を繰り返すことで形を作っている。この点に着目して、彫刻刀やろくろを模倣したモデリングシステムが研究されている[2]~[6]。しかし、これらの方法では道具と物体との接触や、道具を操作する力加減を知るために必須である力感のフィードバックが再現されていない。このため、現実の造形とはほど遠い実操作感覚となっている。仮想的な力感を提示するための様々な対話デバイスも提案されてはいるが[7]~[12]、物体との点接触を再現する段階にとどまっており、土や粘土を扱う手のひら全面の感覚を再現するには至っていない。

一方、非接触の形状変形方法を模倣すれば、特殊なデバイスを用いることなく実際の作業に近いインターフェイスを提供できる。そこで、サンドブラストと呼ばれる石材加工方法をまねた、吹き付けモデリングシステム[13]が野田らによって提案されている。この手法では、吹き付ける粒子の量に比例して仮想空間の立体が実時間で変形するので、ボタンを押しながらマウスを動かすという簡単な操作で立体を造形できる。z-listと呼ばれる空間を量子化して記述する手法を用いることで、穴あけ処理も実時間で行える。

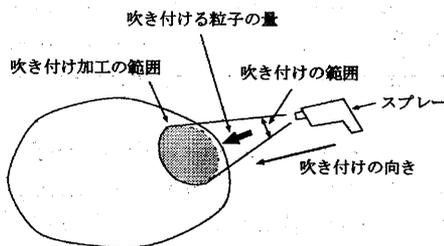


図1. 吹き付けモデリング

しかし、これまでの吹き付けモデラでは、吹き付けの方向を変えるたびにZ-listを再構築しなければならないので、量子化による形状のひずみが生じていた。本研究では、物体形状をサーフェスで記述することで、吹き付け方向の変更に伴う量子化誤差の問題を解決する。さらに、Polygon-ListとFront-Listと名付けた高速処理のためのサーチリストを用いることで、実時間の形状修正を実現する。

2. 従来の吹き付けモデラ

2.1. Z-Listによる形状記述

従来の吹き付けモデラはZ-Listで物体形状を記述している。Z-Listはvoxelと同様に3次元空間を量子化する手法である。ただし、x方向とy方向は等間隔に離散化されるが、z方向は連続量として扱われる。サンプル点(X, Y)のZ-Listはその点を通りz軸と平行な直線が物体と交差する区間を表している。このため、ので、xy平面上のすべての点に対してZ-Listを求めることで、物体形状を記述できる。物体の表面はMarching cubes法[14][15](voxelから境界面を生成するための手法)を拡張したアルゴリズムで効率よく生成できる。

吹き付けモデリングシステムでは、スプレーから放射される粒子はz軸と平行に飛ぶと近似している。このため、Z-Listで物体を記述しておけば、リストの最上位のz値を増減させるだけで形状を変形することができる。トポロジーの変更を伴う穴あけもz-listの1つの区間を取り除くだけで完了する。この際に必要な表面の張り替えは、先に述べたアルゴリズムを部分的に適用することで実現できる。これらの性質により、実時間で連続的に形状を加工することができる。

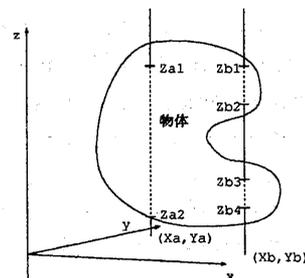


図2. Z-List

2.2. 従来手法の問題点と改良方法

吹き付けはz軸方向に行われるので、加工する向きを変えるには物体を回転させなければならない。しかし、形状の再サンプリングにより、図3に示すような輪郭部のエアアシングが生じる。サンプリング間隔を細かくすればエアアシングを小さくできるが、形状記述に必要なz-listの数が膨大になるため、実時間処理できなくなる。

そこで、本研究では、物体表面を記述するポリゴンの座標を直接操作する吹き付けモデリングシステムを提案する。この方法では形状をZ-listで記述しないので、エアアシングの問題は生じない。ただし、吹き付け範囲内にあるポリゴン効率よく検索するためにZ-Listと同様の空間分割を採用する。このシステムをサーフェス型吹き付けモデラと名付ける。



図3. Z-List型でのモデリング例

3. サーフェス型吹き付けモデラ

3.1. データ構造

本研究で提案するサーフェス型吹き付けモデラは、物体表面を“頂点”とそれを繋いで出来る“三角形”で定義する。さらに、z-listと同様の空間分割で作られる“Front-List”と“Polygon-List”を用いて形状の加工時間を短縮する。以下にそれぞれについて述べる。

頂点

それぞれの頂点は、その3次元座標値と、その頂点を含む三角形へのポインタを保持する。

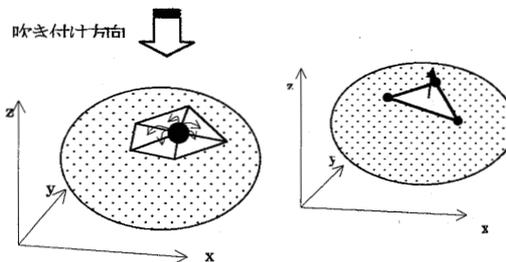


図4. 頂点

図5. 三角形

三角形

それぞれの三角形は、法線ベクトルと3つの頂点へのポインタを記録する。

Polygon-List

図6に示すように、作業空間をxy平面で等間隔に分割して柱状の部分空間を作る。各部分空間に含まれるか交差する三角形(裏面も含む)をあらかじめ調べて、(x, y)をインデックスとするリストに登録する。このPolygon-Listは穴あけ処理の際に、裏側の三角形を効率よく求めるために使われる。また、次に述べるFront-Listを作るためにも使われる。

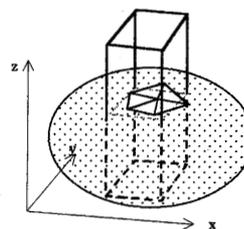


図6. Polygon-List

Front-List

各部分空間で、Polygon-Listに登録された三角形に属する頂点の中から、他の三角形に隠されていないものを選び出し、Front-Listに登録する(図7)。この頂点が吹き付け加工の対象となる。同時に、それぞれの頂点を通りz軸と平行な直線が裏面と交差する点を求め(図8)、そのz座標を保存する。また、図9に示すような、加工方向に向いた三角形の頂点で他の三角形に隠されているものは、穴あけにより最前面に出てくることがあるので、Front-Listの補助リストに保存しておく。

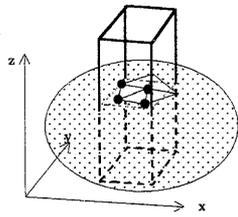


図7. Front-List

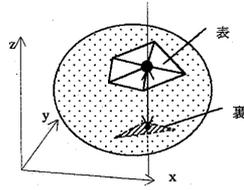


図8. 裏側の面のz座標

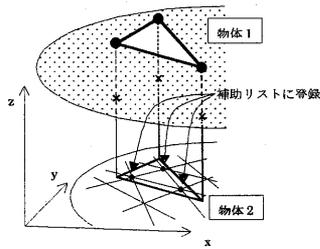


図9. 重なりのある物体

3.2. 吹き付けによる加工

サーフェス型モデラも z -list 型モデラと同じように吹き付け加工を z 軸方向に限定しているため、形状変形は頂点の z 値を増減することで実現される。マウスの位置を通り z 軸と平行な直線を軸とする円筒に含まれる頂点のうち、物体に隠されていない点が加工の対象となる。この頂点は Front-List を使い容易に決定できる。以下に、穴が空かない場合の形状加工の手順を示す。

1. マウス位置と指定された吹き付け半径から決まる円に含まれるサンプル点の Front-List を取り出す(図 10)。
2. そこに登録されている頂点に対して吹き付け粒子の分布パターンに応じて z 値を増減させる。

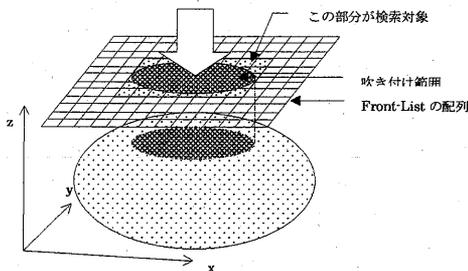


図10. Front-List をピックアップ

3.3. 穴あけ処理

切削を続けて表面が裏面に達すると物体に穴が開く。このときには物体表面を記述する三角形の接続を変更する必要がある。一般に、穴あけは手間のかかる処理だが、提案手法では次の手順で処理時間を短縮している。

1. 加工により表面の z 値が減少し、その値が Front-List に保存された背面の z 値よりも小さくなったとき、裏面に到達したと判定する。裏面に到達したら、頂点の z 値を裏面の z 値に固定し、以後の変形を禁止する。また、図 12 に示すように、表面の頂点が接する位置で裏面の三角形を分割する。
2. 同一三角形に含まれる 2 つの頂点が裏面に接したときには、図 13 のように 2 つの頂点を結ぶ線分で裏面を分割し、三角形を作り直す。この時点で、裏面の形状が変更されるが、表面と裏面は重なっているだけで、それぞれの接続は保持されている。
3. 3 つ目の頂点が裏面に接した時点で穴をあける(図 14)。始めに、2 と同様に残り 2 辺で裏面を分割する。次に、表の三角形とそれに隠された裏の三角形を取り除く。最後に、表面の三角形の頂点を消去する。このとき、表面三角形と頂点を共有する全ての三角形に対して、削除される頂点へのポインタを対応する裏面の頂点に付け替える。この時点で、表面と裏面が繋がり、面のトポロジーが変更される。
4. 削除された三角形の陰になっている頂点が補助リストにある場合は、その頂点を補助リストから Front-List に移動する。

なお、三角形の追加・削除に応じて、その都度、Polygon-List を更新する。

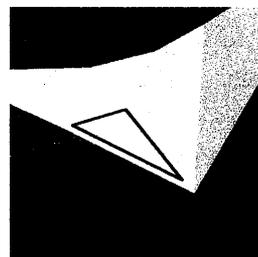


図11. 穴あけ前

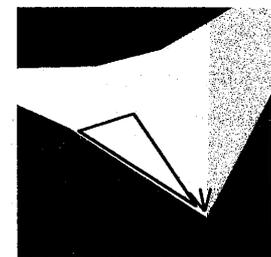


図12. 表の頂点を裏の三角形に接続

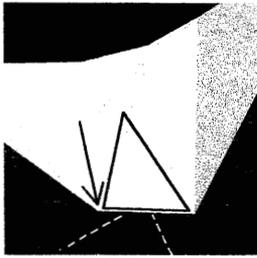


図13. 2頂点を結ぶ線で分割

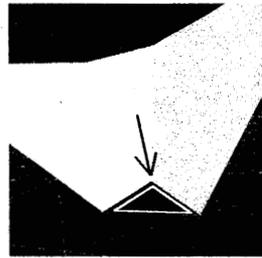


図14. 表の三角形とその裏面を削除

3.4. 吹き付けの方向の変更

変形処理やツールの操作を簡単にするため、吹き付けの向きを世界座標のz軸方向に固定している。したがって、別の方向から吹き付けるには物体を世界座標系で回転すればよい。ポリゴンサーフェスで定義された物体は回転しても形状が変わらないので、Z-List型のような不必要な凹凸が生じることはない。ただし、頂点と三角形の前後関係は変化するので Polygon-List と Front-List は回転の度に再構築している。

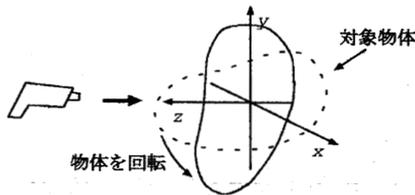


図15. 吹き付けの向きの変更

4. システムの概要

実験システムは、SGI社のOCTANE (MIPS R10000, 195MHz, メモリ 256Mbytes)上にC++言語とOpenGLを用いて構築した。ユーザインターフェイスはMotifを用いて作成した。

4.1. インターフェイス

システムのインターフェイスは、形状操作を行うメインウィンドウ(図16)と、吹き付けのパラメータを設定するコントロールパネル(図17)で構成されている。メインウィンドウはさらに、吹き付け加工を行

う作業画面と様々な視点から物体を表示する確認画面に分けられる。

作業画面には物体が正射影で表示される。物体表面にはマウスカーソルの代わりに吹き付け範囲を示す円が投影される。マウスの左ボタンを押す間、その円に含まれる物体表面が連続的に盛り上げられる。また、中ボタンを押すと物体が削られる。右ボタンでは後述する平滑化ツールを使って面を滑らかにすることができる。物体の投影図や加工の範囲は、ユーザの操作に対してリアルタイムに更新される。

確認画面にはモデリングしている物体の透視投影図が表示される。画面上をドラッグすることによって、物体の回転、平行移動、拡大・縮小が可能である。操作の種類は、確認画面下部のチェックボックスで選択する。吹き付け方向を変更するボタンもここにある。このボタンを押すと、確認画面に表示された物体の向きや大きさと一致するように、作業画面の物体が回転、平行移動、拡大・縮小される。これにより、物体を任意の方向から任意の拡大率で加工することができる。また、作業画面も確認画面もユーザの操作に対してリアルタイムに更新されるので、現実の形状加工と同様に結果を様々な角度から確認しながら作業を進めることができる。

吹き付けのパラメータはコントロールパネルで設定する。具体的には、吹き付け加工の半径、加工の速度、スプレー粒子の分布関数、の3種類のパラメータを設定できる。詳細は4.2節で説明する。データファイルのセーブ・ロード、データの初期化、プログラムの終了はメニューから実行できる。

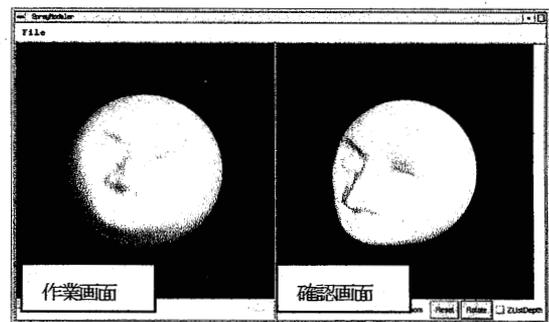


図16. メインウィンドウ

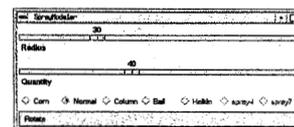


図17. コントロールパネル

4.2. 形状操作ツール

先に述べたように、吹き付け加工は作業画面内で行う。作業画面と確認画面の両方に加工される範囲が表示されるので、マウスをドラッグして移動し、ボタンを押してそこに含まれる物体表面を変形する。変形の程度はボタンを押し続ける時間で調節する。ボタンを押しながらマウスを移動すれば、広い範囲を連続して加工できる。これにより、微妙な変化を持つ曲面などの複雑な形状もモデリングできる。

加工の範囲、単位時間に吹き付けられる粒子の量はコントロールパネルで指定する。粒子の分布もここに列んだノズルを選び直すことで変更できる。現在のインプリメントでは次にあげるノズルが用意されている。

- 円錐状に盛り上げる/削るノズル
- エッジが目立たないようになめらかに盛り上げる/削るノズル
- 円柱状に盛り上げる/削るノズル
- 半球状に盛り上げる/削るノズル

でこぼこの表面を滑らかにするために、サンドペーパーを模倣したツールを用意している。このツールは、ボタンを押すたびに、吹き付け範囲に含まれる物体頂点のz座標をその頂点が接続している頂点のz座標の平均値で置き換える。したがって、ボタンを押す回数に比例して表面が滑らかになる。

図18の四隅の盛り上がりは、4種類のノズルのそれぞれで吹き付けたときにできる形状を示している。また中央には、円柱状ノズルを動かしながら作ったでこぼこをサンドペーパーツールで半分ほどならした結果を示す。左側はでこぼこのままであるが、右側の表面は滑らかになっている。

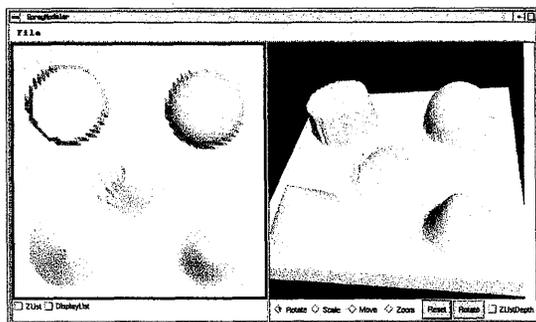


図18. ノズルの種類

4.3. 実時間表示のための工夫

対話的なモデリングを実現するには、ユーザの操作に対してリアルタイムに画面を書き換える必要がある。一度の吹き付け加工では物体のごく一部しか変形されないため、形状が更新された部分だけ画像を再描画すれば画面更新時間を短縮できる。実験システムで用いる OpenGL には登録しておいた描画オペレーションを高速に実行する Display List と呼ばれる機構があるので、形状が更新されない部分はこの機能を利用することで再描画時間を短縮する。

画面更新時間は、Display List を用いない場合は約 0.15 秒であったが、分割された Display List を用いることで約 0.05 秒へと短縮された。フレームレートに換算すると、約 6.67 frame/s から約 20.0 frame/s へ改善されたことになる。ただし、全体の頂点数 10243 個、ポリゴン数 20481 個で、加工対象の頂点数約 160 個としたときの数値である。

一般に、画面更新が 10 frame/s 以下になると、ユーザはストレスを感じると言われているので、Display List の導入により、ユーザの使用感を大幅に向上でき、実時間モデリングが達成できたと言える。

5. 実験

5.1. Z-List 型吹き付けモデラとの比較

回転の影響について比較するため、球面の一部を少し盛り上げた後、吹き付け方向を変更して、結果を比較する。図19の上段は左上方向から吹き付け加工した結果で、下段は吹き付け方向を左下に変更した後の形状である。

Z-List 型モデラでは回転前から吹き付け方向と平行な面にエアリングが発生している。吹き付け方向を変更すると新しい吹き付け方向と平行な面にエアリングが追加されるので、図19左下のように二つの帯ができてしまう。また、物体の縁に行くほどz軸方向に伸びたポリゴンができてしまう。

サーフェス型では全頂点を同じ回転行列で移動するだけなので形状は変化しない。このように、形状の品質ではサーフェス型が大きく勝っている。吹き付け向きを少しずつ変えながら加工を行う必要がある場合には、この回転による形状不変性がきわめて大きな利点となる。

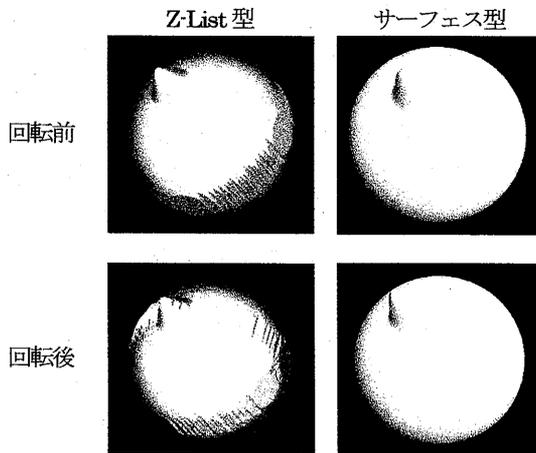


図19. 回転について比較

5.2. サーフェス型吹き付けモデラを使った 形状入力実験

図 20に筆者がサーフェス型吹き付けモデラで作成した人の頭部を示す。モデリング作業には約2時間かかった。あごや首などは吹き付けの向きを少しずつ変えながら加工したが、サーフェス型モデラでは、吹き付け方向を変更しても形状が不用意に変形することはないので、加工順序や加工方向の違いによる結果の違いを心配しなくてもよい。しかし、実験システムでは表面を記述するポリゴンの数が十分でないため、目や耳などの細かい部分が加工しにくかった。この解決法については次章で今後の課題として取り上げる。

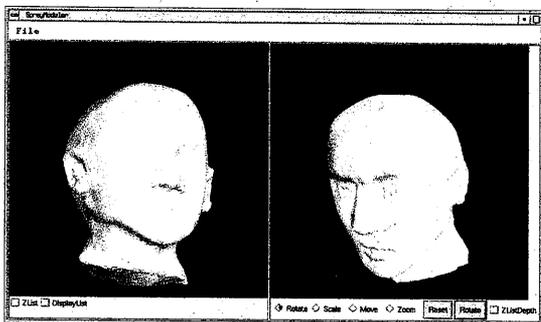


図20. 頭

図21のりんごは初心者が作成した。被験者は、提案システムはもとより、どの三次元モデリングシステムも使った経験がない。このため、操作方法を説明しながら30分ほど練習した後、モデリング作業

に取りかかってもらった。3次元を操作することが初めてだったため、はじめは多少とまどい気味であったが、すぐに慣れ、りんごの不規則な曲面を思い通りに表現することができた。吹き付けによるモデリングは操作が単純で、連続的に変形でき、結果がリアルタイムに表示されるため、初心者でも簡単に使いこなすことができる。

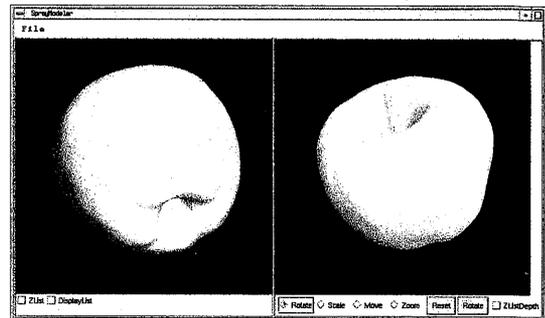


図21. りんご

6. まとめ

従来のZ-List型吹き付けモデラでは、吹き付け方向を変更するたびに、形状の再サンプリングによる表面の細かいでこぼこが生じていた。今回提案したサーフェス型モデラは、頂点を回転するだけで吹き付け方向を変更できるので、形状の劣化は起こらない。この点は従来のシステムに対する大きな利点である。また、物体の表面を覆うポリゴンの大きさがほぼ均一なので、ポリゴン数のわりにはなめらかな面が表現できる。ポリゴンを使うと穴あけ処理が複雑になるが、アルゴリズムの工夫により、Z-List型と同じ程度の処理時間で収まっている。このため、十分に実時間処理できている。

サーフェス型吹き付けモデリングシステムでは、マウスだけを使って3次元物体を造形することができる。形状はリアルタイムに修正され、表示される。スプレーの散布範囲を調節すれば、広範囲の大まかな加工から部分的な細かい加工まで、同じ感覚で作業できる。このように、本システムのインターフェイスは単純で結果も容易に予測できるので、操作に特別な知識を必要としない。2次元のペイントシステムと同様な手軽さで扱うことができる。このため、実験が示すように、全くの初心者でも直感的にモデリングすることができた。

しかし、形状操作を繰り返すうちに一部のポリゴンが大きくなりすぎることもある。また、微細加工するときにもポリゴンが相対的に大きくなるので、滑らかな加工が妨げられる。初期形状の分割数を増やせばこれらの障害を小さくできるが、根本的な解決にはならない。また、むやみにポリゴン数を増やすと実時間処理が難しくなる。

そこで、今後の課題として、ポリゴンを動的に分割する機能をインプリメントする。同時に、小さくなりすぎたポリゴンや滑らかな面を過剰に分割するポリゴンは不要な処理時間を増加させるので、ポリゴンの統合処理も導入する。ポリゴンの分割・統合に関する研究はすでに多数行われているので [16][17]、これらを参考に検討を進める。分割・統合処理はそれほど頻りに要求されないので、吹き付け方向を変更するときなど形状操作が中断されたときに限って行えば、実時間処理の障害とはならない。

参考文献

- [1] David R. Forsey and Richard H. Bartels, "Hierarchical B-Spline Refinement", *Computer Graphics*, Vol.22, No.4, 205-212 (1988)
- [2] 水野慎士, 岡田稔, 横井茂樹, 鳥脇純一郎: "CSG を用いた仮想彫刻", 情処研報, グラフィクスと CAD73-7(1995)
- [3] Tinsley A. Galyean, John F. Hughes: "Sculpting: An Interactive Volumetric Modeling Technique", *Computer Graphics*, Vol.25, No.4, p.267-274 (1991)
- [4] 吉田英正, 大野義夫: "対話的なレリーフ形状作成システムに関する研究", 情処研報, グラフィクスと CAD58-13 (1992)
- [5] 生田大介 他: "レリーフを対象にしたモデリング技法の提案", 東海支部連合大会予稿, p.398 (1999)
- [6] 亀井克之, 中村泰明, 阿部茂: "エネルギー最小化による変形可能仮想ろくろモデラ", 信学論, Vol.J76-D-II, No.8, p.1772-1779 (1993)
- [7] 中村康浩, 吉村哲也, 杉浦雅貴: "3次元直接操作インタフェースの実現", 情処研報, グラフィクスと CAD55-1 (1992)
- [8] William R. Mark, Scott C. Randolph, Mark Finch, James M. Van Verth, Russell M. Taylor II: "Adding Force Feedback to Graphics Systems: Issues and Solutions", *Computer Graphics Proceedings, Annual Conference Series*, p.447-452 (1996)
- [9] 清川清, 竹村治雄, 片山善章, 岩佐英彦, 横矢直和: "両手操作を用いた仮想物体モデラ VLEGO", 信学論 (A), Vol.J80-A, No.9, p.1517-1526 (1997)
- [10] Michael F. Deering: "The HoloSketch VR Sketching System", *Communication of the ACM*, Vol.39, No.5, p.54-61 (1996)
- [11] 山下樹里, 福井幸男, Cai Yi: "ViSurf: 3次元触・力覚呈示デバイスによる面の直接モデリング環境", *Visual Computing グラフィクスと CAD 合同シンポジウム* (1997)
- [12] 山下樹里, 福井幸男, 横井浩史, 下条誠: "3D-DDM: 三次元B-Spline自由曲面の直接変形手法", 電学論 C, Vol.115, No.2, p.253-260 (1995)
- [13] 野田靖之, 田中敏光, 大西昇: "吹き付けによる3次元形状モデリング", 信学技報 HIP97-54, p.23-30 (1998)
- [14] William E. Lorensen, Harvey E. Cline: "MARCHING CUBES: A HIGH RESOLUTION 3D SURFACE CONSTRUCTION ALGORITHM", *Computer Graphics*, Vol.21, No.4, p.163-169 (1987)
- [15] 石井智海, 安田孝美, 横井茂樹, 鳥脇純一郎: "マーチングキューブ法の改良アルゴリズムについて", 情処研報, グラフィクスと CAD60-9 (1992)
- [16] 小林順一, 大野義夫: "ポリゴンモデルの形状変形におけるメッシュの再構成", 情処研報, グラフィクスと CAD 89-6 (1998)
- [17] 早川勝之, 松岡司, 植田健次: "特徴抽出と稜線操作によるポリゴンメッシュの簡単化", 情処研報, グラフィクスと CAD 89-7 (1998)