

NAIST-IS-MT9751100

修士論文

陰関数表現を用いた没入型曲面形状モデラ

松宮 雅俊

1999年2月12日

奈良先端科学技術大学院大学
情報科学研究科 情報システム学専攻

本論文は奈良先端科学技術大学院大学情報科学研究科に
修士(工学) 授与の要件として提出した修士論文である。

松宮 雅俊

指導教官： 横矢 直和 教授
千原 國宏 教授
竹村 治雄 助教授

陰関数表現を用いた没入型曲面形状モデラ*

松宮 雅俊

内容梗概

3次元曲面形状のモデリングに一般に用いられている従来の3次元CADや3次元CGソフトウェアには、操作性や直感性、形状の視認に問題がある。本論文では、このような問題の改善を目的として、仮想環境内で3次元曲面形状の対話的かつ直感的なモデリングが可能なモデラを実現するための手法を提案する。提案手法では、陰関数表現を用いることにより、滑らかな自由曲面形状を手や指を用いて直接的に操作し、切削や付加といった形状の変形を実現している。本研究で採用した陰関数表現には、形状変形に際して、形状の衝突判定や変形処理に複雑な演算を必要とせず、その定義から自然に変形を表現することが可能であるという利点がある。一方で、没入型モデラでは、3次元形状の変形やユーザの視点移動による画像を実時間で提示する必要があるが、陰関数表現による形状の描画に一般的に用いられるレイトレーシング法は、実時間提示には不向きである。そこで本研究では、陰関数表現を効率良くポリゴンに変換する手法を提案し、高速なレンダリングを実現する。最後に、提案手法に基づくモデラを実際に試作し、複雑で滑らかな曲面を持った形状を手を用いた直感的な操作によって短時間でモデリングできることを、実際のモデリング結果によって示す。

キーワード

仮想現実感, 対話的モデリング, 陰関数表現, ポリゴン変換

*奈良先端科学技術大学院大学 情報科学研究科 情報システム学専攻 修士論文, NAIST-IS-MT9751100, 1999年2月12日.

An Immersive Curved Surface Modeler Using Implicit Surface Representations*

Masatoshi Matsumiya

Abstract

Most of existing 3D-CAD and 3D-CG softwares have problems in intuitive manipulation and understanding of 3D curved surfaces. This paper proposes a methodology of building an immersive modeler which enables us to design 3D solid objects composed of curved surfaces intuitively and interactively. We employ implicit surfaces to represent smooth free-formed surfaces. Owing to the definition of implicit surfaces, shape deformation is naturally expressed without complex calculation. Although implicit surfaces are well suitable for ray tracing methods in rendering, ray tracing methods are hard to run in real-time. To reduce the time required for rendering curved surfaces in interactive modeling, we have developed an algorithm which generates polygon patches from implicit surface representations, since conventional graphics hardware is optimized for displaying polygons. Experiments have shown that a user of the prototype system can design 3D solid objects composed of curved surfaces in a short time by deforming objects intuitively by hand in real-time.

Keywords:

virtual reality, interactive modeling, implicit surfaces, polygonization

*Master's Thesis, Department of Information Systems, Graduate School of Information Science, Nara Institute of Science and Technology, NAIST-IS-MT9751100, February 12, 1999.

目次

| | |
|----------------------------------|----|
| 1. はじめに | 1 |
| 2. 陰関数による曲面の表現 | 5 |
| 2.1 ポリゴン近似 | 5 |
| 2.2 パラメトリック表現 | 6 |
| 2.3 陰関数表現 | 8 |
| 3. 没入型曲面形状モデラ | 14 |
| 3.1 試作モデラの概要 | 14 |
| 3.2 陰関数処理 | 19 |
| 3.3 ポリゴン変換によるレンダリングの高速化 | 24 |
| 3.3.1 Bloomenthalらによるポリゴン変換手法 | 24 |
| 3.3.2 曲面変形領域を考慮したポリゴン変換の高速化手法の提案 | 28 |
| 4. モデリング例 | 31 |
| 5. 考察 | 36 |
| 6. むすび | 38 |
| 謝辞 | 40 |
| 参考文献 | 41 |

目 次

| | | |
|----|---------------------------------------|----|
| 1 | ポリゴン近似 | 6 |
| 2 | ベジエ曲面 | 7 |
| 3 | 陰関数表現による球面 | 9 |
| 4 | 様々なスケルトンの例 | 10 |
| 5 | 重み w による形状の変化 | 11 |
| 6 | 距離場関数の例 | 12 |
| 7 | 制御点 CP における距離場関数の傾きの影響 | 12 |
| 8 | モデリングの様子 | 15 |
| 9 | 試作モデラのシステム構成 | 15 |
| 10 | 人差し指による切削の様子 | 16 |
| 11 | 人差し指による付加の様子 | 17 |
| 12 | 手の平による切削の様子 | 17 |
| 13 | 仮想環境の様子 | 18 |
| 14 | データ処理の流れ | 19 |
| 15 | エルミート曲線による距離場関数 | 20 |
| 16 | 人差し指による切削の表現 | 21 |
| 17 | 人差し指による付加の表現 | 22 |
| 18 | 手の平による切削の表現 | 23 |
| 19 | ボクセル空間 | 25 |
| 20 | ボクセルの探索 | 25 |
| 21 | ボクセルの分割 | 26 |
| 22 | 4 面体と形状との交点 | 27 |
| 23 | ポリゴン変換用ボクセル空間と陰関数データ用ボクセル空間 | 29 |
| 24 | 制作過程の例 (ティーカップ) | 32 |
| 25 | モデリング例 1: ティーカップ | 33 |
| 26 | モデリング例 2: 水槽の中の魚 | 34 |
| 27 | モデリング例 3: 立体手書き文字 | 35 |

1. はじめに

3次元コンピュータグラフィクス(CG)を効果的に使用した画像や映像は、プレゼンテーション、映画、コマーシャル、ゲームなどに積極的に活用され、3次元CGの必要性は高まる一方である。そして、3次元CGの描画対象としては、球面や円筒面といった単純な曲面だけでなく、生物や工業製品などのように、自由曲面と呼ばれる複雑な曲面を含む形状が描かれることが多い。そこで、このような自由曲面を持った形状をモデリングするための計算機システムである曲面形状モデラが必要になる。自由曲面のモデリングに一般に用いられている曲面形状モデラとしては、AutoCAD[1]などの3次元CAD(Computer Aided Design)ソフトウェアや、Ray Dream Studio[2]、LightWave 3D[3]などの3次元CGソフトウェアがある。これらのソフトウェアでは一般に、3次元形状の透視投影図や3面図、形状の組み合わせや親子関係を示した構成図(一般にツリー構造)、様々な機能を選択し実行するためのアイコンやメニューなどが2次元ディスプレイ上に表示され、それらに対してマウスやタブレットなどの入力装置を用いて曲面のモデリングを行う。

これらのソフトウェアを用いて自由曲面をモデリングする方法は様々であるが、AutoCADではメニューやコマンドによって直接座標値を入力したり、4本の曲線によって閉じた平面図形を描き、その間を曲面として内挿する方法などがある。また、Ray Dream Studioではポリゴンによって近似された曲面形状を構成し、ポリゴンの頂点をマウスで移動することによって曲面形状を変形させる。LightWave 3DではMetaNURBSと呼ばれるパラメトリック曲面の制御点をマウスで直接移動したり、メニューから選択した機能によって移動することで曲面を変形させてモデリングを行う。

このように、既存の3次元CGソフトウェアでは、様々な形式で記述される3次元曲面を、マウス等を用いて操作するインタフェースが提供されている。しかしながら、これらの3次元CADや3次元CGソフトウェアには、次のような問題点がある。

- 入力装置としてキーボードのような装置や、マウスのような2次元操作に

適した装置を用いて 3 次元の形状をモデリングするため、例えばマウスを用いて 3 次元空間の形状を表示画面の奥行き方向に移動させるような場合に特殊な操作が必要となる。このような操作は、我々が日常的に実環境において行っている動作と異なるために直感性に欠け、操作性の悪化の原因となる。

- 3 面図や透視投影図など 2 次元座標系に変換された画像を用いて、2 次元画面上で 3 次元形状をモデリングするため形状の視認が困難である。
- コマンドや制御点の移動などによって間接的に曲面を変形させるため、変形操作を実行することによって曲面がどのように変形するかを予想することが困難である。

以上に挙げた問題点を改善するためには、曲面モデラが次のような特徴を持てばよいと考えられる。

1. 形状の変形操作が簡単で、実環境における動作と同じであること。
2. モデリング時に形状の視認が容易で、日常的に実環境において行っている動作と同じように、形状を立体的に様々な角度から見ることができること。
3. 変形操作を実行することによって曲面がどのように変形するかを予想することが容易で、望んだ通りの形状が得られること。

さらに曲面モデラとしては、滑らかな自由曲面形状を表現するために、次の特徴を備えることが望ましい。

4. 扱える形状が連続性のある滑らかな自由曲面であること。

これらの要求の幾つかを満たすと考えられるモデラについては、従来から多くの研究が行われている。Galyean らは、ボクセルデータで表現された形状を、彫刻するような操作感覚で変形させるモデラ [4] を開発している。入力装置には位置、姿勢を計測することができる 3 次元マウスを用いるため、実環境における彫刻操作に近い操作でモデリングが行える。しかし、形状変形はボクセルの消去や

生成によって行われているため、生成される曲面は連続性のある滑らかな自由曲面ではない。また、2次元画面上でモデリングを行うため、形状の視認が困難であるという問題点もある。Wangらの開発しているモデラ [5] についても、彫刻刀で木を削るようにしてモデリングが行えるが、2次元画面上でマウスを用いて操作するため、実環境での彫刻操作に比べると違和感がある。Nobleらの開発しているモデラ [6] では、パラメトリック曲面の制御点を移動させるのではなく、曲面を直接操作することができる。しかし、マウスを用いて曲面上の1点を指定して変形させる際に、その周囲がどのように変形するのかが分かりにくい。これらのモデラと同様に彫刻作業を仮想環境内で実現しようとする手法や、曲面変形を粘土を扱うようにして実現しようとする手法などについても、これまでにいくつかの研究 [7, 8, 9, 10] が行われているが、前述した4つの要求を全て満たすモデラはこれまでに開発されていない。

そこで本研究では、1.～4.の全ての要求を満たしたモデラの開発を目標とする [11, 12]。まず、要求1.と要求2.を満たすために、没入型モデラにおいて手や指を用いて形状を直接変形させるという操作方法を採用する。没入型モデラは仮想現実感技術によって、計算機の生成する仮想環境に没入して3次元形状のモデリングを行うことのできるシステムであり、以下のような利点がある。

直感的な視認

HMD(Head Mounted Display)や液晶シャッター眼鏡などによって、左右の目にそれぞれの視点からの画像を提示して両眼立体視することにより、対象物を立体的に見ることができる。また、磁気トラッカなどの3次元位置センサによって、頭部の位置、姿勢を計測し、その視点からの画像を実時間提示する。これらにより、3次元形状やそれらの位置関係を容易に把握することが可能となる。

直感的な操作

3次元位置センサや、それを取り付けた手形状入力装置などの装置を用いることで、2次元の入力装置では煩雑になりがちな3次元形状の

操作を直接的に行うことができる。

対話的モデリング

仮想環境内では、3次元形状に対する操作は直ちに提示画像に反映され、その操作結果が即座に視認できるため、効率的なモデリング環境を構築することができる。

このような特徴を有する仮想環境で、手を用いた直接操作を可能とすることにより、実環境での作業に近いモデリング作業を行うことができるため、簡単で直感的な曲面形状の変形操作を実現できる。

さらに本研究では、要求3.と要求4.を満たすために、曲面形状の表現手法として陰関数表現 (Implicit Surfaces) を採用することにより、手や指を用いて直接的に形状の切削や付加が行える手法を提案する。これにより、曲面の制御点の移動による間接的な形状変形操作と比較して、変形操作後の形状変化を想像することが容易になり、曲面モデリングの効率を向上させることができる。また、陰関数表現は、その定義から自然に曲面形状の変形を表現することが可能であり、また曲面は常に連続で滑らかである。さらに陰関数表現には、閉曲面の内部と外部を簡単な不等式で判定できるという性質があり、この性質のために、陰関数表現で記述された曲面形状では、形状の衝突判定のために特別な処理が必要ではなく、処理の高速化が図れるという特徴もある。

先述の4つの要求に加えて、没入型モデラでは、3次元形状の変形やユーザの視点移動による画像を実時間で提示する必要があるため、陰関数表現による形状を高速にレンダリングする手法についても検討する必要がある。本論文では、陰関数表現を効率良く実時間でポリゴンに変換する手法を提案し、高速なレンダリングを実現する。

以下では、2章で曲面の陰関数表現について、その定義や特徴を述べる。3章で先述の設計方針に従って試作した没入型曲面形状モデラについての詳細を述べた後、4章で試作モデラによるモデリング例を示し、複雑で滑らかな曲面を持った形状を、手を用いた直感的な操作によって短時間でモデリングできることを実験的に確認する。最後に考察を加えてむすびとする。

2. 陰関数による曲面の表現

1章において述べたように，本研究で開発するモデラにおいては，曲面形状の変形を手や指を用いて直接的に行う．この変形操作においては，切削の跡や付加された形状が，曲面に接している手や指の形状に一致するようにする．これにより，変形操作を実行することによって曲面がどのように変形するかを容易に予想することが可能となる．モデラで用いる曲面形状の表現手法は，このような操作が可能であると同時に，変形の際に連続性のある滑らかな自由曲面を保つという性質が必要となる．

そこで本章では，このような性質を有する曲面の表現手法について検討する．計算機による曲面の表現手法としては，従来，ポリゴンによる近似やパラメトリック表現，陰関数表現が用いられている．以下では，これら曲面の表現手法について，その定義や特徴を述べた後，本研究で検討するモデラに用いる曲面の表現手法であるスケルタルサーフェス (Skeletal Implicit Surfaces)[13, 14] について述べる．

2.1 ポリゴン近似

ポリゴンによる曲面の近似とは，図 1 に示すように，曲面をポリゴン（多角形）の頂点や稜線，面といった情報のみによって表現するものである．現在の計算機では，ポリゴンのレンダリングに特化された専用のハードウェアが普及しているために，ポリゴン近似には，その形状を高速にレンダリングすることができるという利点がある．次節で説明するパラメトリック表現によって曲面形状を表現するモデラであっても，レンダリングの高速化のために最終的に形状データをポリゴン変換するモデラもある．しかし，ポリゴン近似を本研究で検討するモデラに適用する場合，以下のような問題点が考えられる．

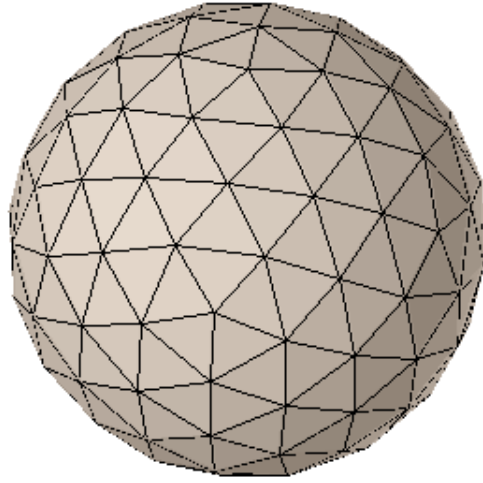


図 1 ポリゴン近似

- 曲面の連続性に関する保証がない.
- 手や指による形状の切削や付加の際に，切削の跡や付加された形状が，手や指の形状になるように変形させるには，ポリゴン同士の厳密な衝突判定や変形処理が必要になる．しかし，このような処理には多くの計算量が必要のため，実用的な処理速度と変形の精度を実現することが困難である．

このような問題点から，ポリゴン近似による曲面表現は，本研究で開発する曲面形状モデラでの曲面形状の変形操作を実現するためには適していないと考えられる．

2.2 パラメトリック表現

パラメトリック表現とは，パラメータの陽関数形式で表現された曲面である．パラメトリック曲面の1つであるベジエ (Bézier) 曲面 [15] は，図 2に示すように，格子状に並んだ制御点 p_{ij} によって定義される曲面である．この時，制御点は曲面の四隅の点と概形を定める．

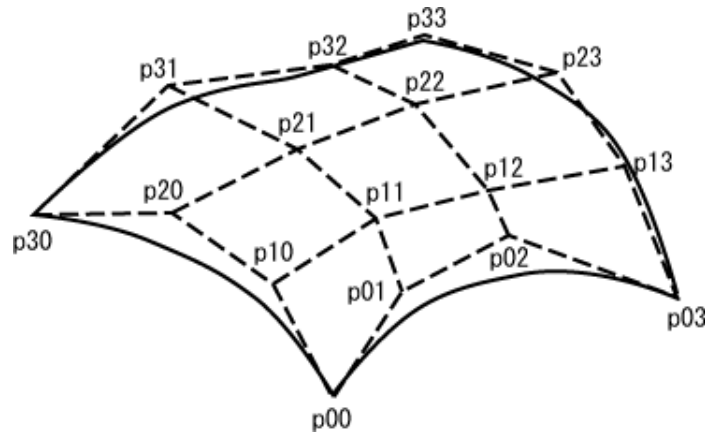


図 2 ベジエ曲面

ベジエ曲面は式 (1) で表される. 式 (1) に示すように, ベジエ曲面上の点 $S(u, v)$ は, 制御点 p_{ij} の加重平均になっている. パラメータ u, v を 0 から 1 まで変化させると, 各制御点に対する重みである $B_i^n(u)$ と $B_j^m(v)$ が連続的に変化して, 点 $S(u, v)$ が端点から端点へ連続的に移動する. この点の軌跡がベジエ曲面となる.

$$S(u, v) = \sum_{i=0}^n \sum_{j=0}^m p_{ij} B_i^n(u) B_j^m(v) \quad (1)$$

ここで, $B_i^n(u)$ と $B_j^m(v)$ は, n 次と m 次のバーンスタイン (Bernstein) 多項式関数であり, 式 (2) で定義される.

$$\begin{aligned} B_i^n(t) &= (1-t)B_i^{n-1}(t) + B_{i-1}^{n-1}(t) \\ B_0^0(t) &= 1 \\ B_{-1}^n(t) &= 0 \end{aligned} \quad (2)$$

パラメトリック曲面の変形は制御点を移動することで行われるが, パラメトリック表現を本研究で検討するモデルに適用する場合, 手や指による形状の切削や付加の際に, 切削の跡や付加された曲面の形状が, 手や指の形状になるように制御点を移動する必要がある. しかし, 曲面の形状から制御点の位置を逆算することは原理的に困難であるため, 例えば制御点の移動によって生成された曲面と, 手

や指の形状とを比較し、差が小さくなるように制御点を再び移動させるといった反復処理などの特別な処理が必要となる。このような処理は複雑になり、実用的な処理速度と変形の精度を実現することが困難であると考えられる。このような理由から、パラメトリック曲面は本研究で開発を目指すモデルの設計方針に従った変形操作を実現するためには適していないと考えられる。

2.3 陰関数表現

陰関数表現とは曲面上の点が満たすべき条件を陰関数の形式で表現することによって曲面を定義する表現手法であり、一般に式 (3) で定義される。

$$\{P \in R^3 | f(P) = 0\} \quad (3)$$

式 (3) において、陰関数 $f(P)$ に用いる関数によって様々な曲面表現があり、例えば、式 (4) のような多項式関数を用いた場合は、代数曲面 (Algebraic Surface) と呼ばれ、後述の式 (5) のような関数を用いた場合は、スケルタルサーフェスと呼ばれる。

陰関数表現に共通する性質として、 $f(P) = 0$ となる点 P は曲面上の点であり、この点の集合が曲面となる。また、曲面をはさむ両側の点で関数 $f(P)$ を求めると、一方は $f(P) > 0$ となり、他方は $f(P) < 0$ になる。例えば、原点を中心とする半径 r の球面は式 (4) で表される。

$$f(P) = x^2 + y^2 + z^2 - r^2 = 0 \quad (4)$$

式 (4) において、 $f(P) = 0$ となる点 P の集合が半径 r の球面となる。また、 $f(P) > 0$ となる点 P は球面の外部、 $f(P) < 0$ となる点 P は球面の内部の点であることを示す。これを簡単のため 2 次元で表した図を図 3 に示す。

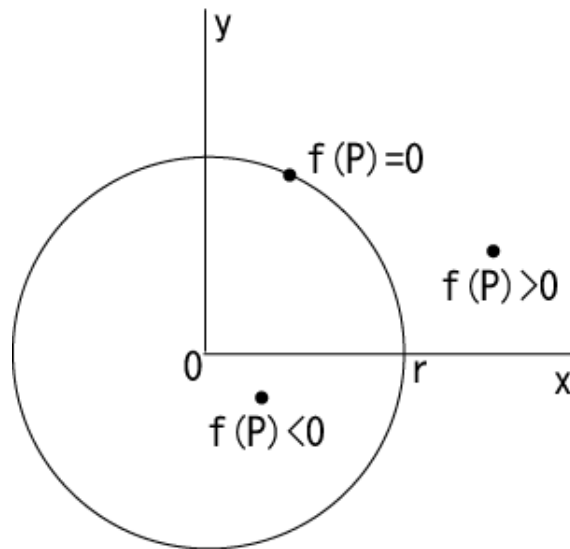


図 3 陰関数表現による球面

このように、陰関数表現には関数値の符号から閉曲面の内部と外部を簡単に識別できるという特徴があり、この性質は、曲面に指や手を接触させて変形を行う本研究で開発するモデラに適していると考えられる。次に、様々な陰関数表現の中でも、後述の理由から本研究での利用に適していると思われるスケルタルサーフェスについて説明する。

陰関数表現の一種であるスケルタルサーフェスでは、曲面形状は式 (5) の陰関数多項式で表現され、このとき曲面形状はスケルトン S と重み w 、距離場関数 (Field Function) F によって決定される。

$$\{P \in R^3 | f(P) = c\}$$

ただし

$$f(P) = \sum_{i=1}^n w_i F_i(d(P, S_i)) \quad (5)$$

スケルトン S は形状の骨格となるもので、点や線分、平面などを用いることができる。様々なスケルトンの例として簡単のため 2 次元で表したものを図 4 に示す。また、 $d(P, S_i)$ は点 P とスケルトン S 上の最も近い点との距離を表す。

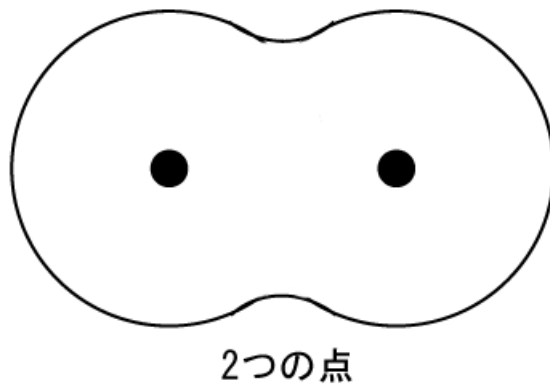
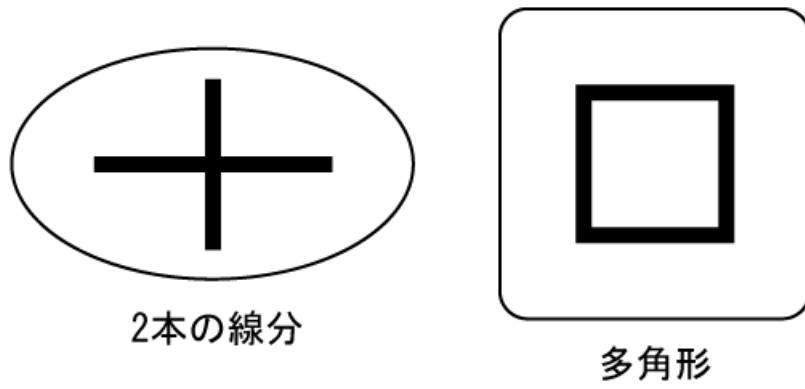
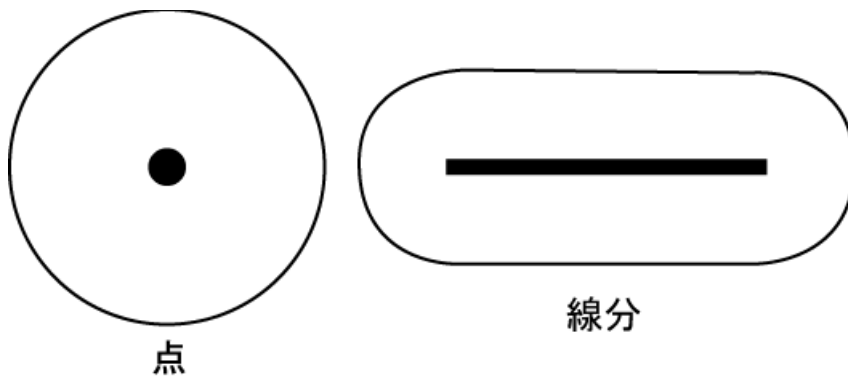


図4 様々なスケルトンの例

重み w はスカラー値であり，正負の値を取る．2つの点スケルトンから成る形状の，重み w による変化の例を図 5に示す．図 5(a)，(b)に示すように，重み w が正の値同士の場合，一方の形状が他方の形状に付加される．また，図 5(c)に示すように，重み w が正の値と負の値の形状を組み合わせた場合，正の値の形状が負の値の形状に切削される．

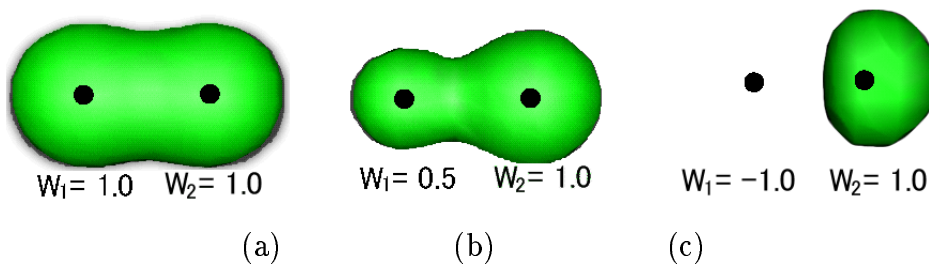


図 5 重み w による形状の変化

距離場関数 F は典型的には図 6に示されるような形状となる関数が使われる．制御点 CP における傾きと影響範囲となる幅を制御することで形状の変形具合を制御することができる．制御点 CP の傾きによる形状の変化を図 7に示す．傾きが小さい場合つなぎ目が緩やかになり，傾きが大きい場合つなぎ目が急になる．

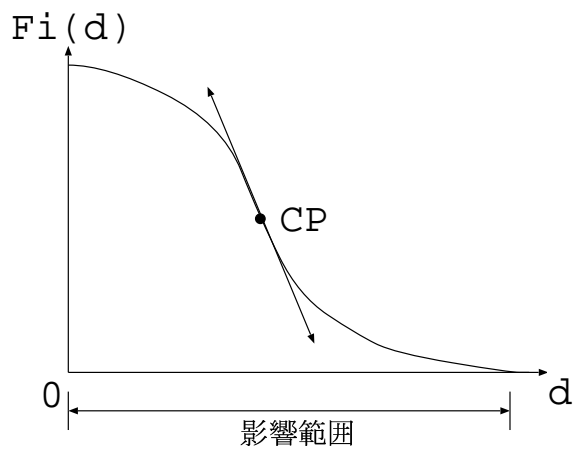


図 6 距離場関数の例

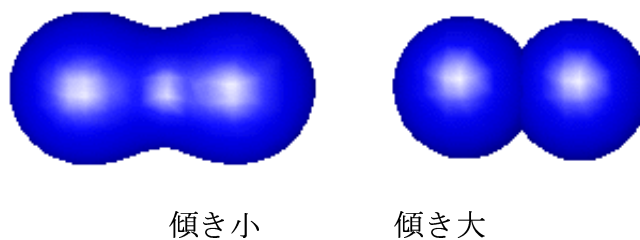


図 7 制御点 CP における距離場関数の傾きの影響

以上述べた陰関数表現の 1 つであるスケルタルサーフェスを、次のような理由から本研究で検討するモデラに用いる。

- スケルトンの形や位置は容易に制御できるため、手や指、プリミティブの形状が表現しやすい。
- 手や指、プリミティブ形状にスケルタルサーフェスを用いれば、スケルタルサーフェスの定義から自然に、手や指による形状の切削や付加といった変形を表現することができる。

- スケルタルサーフェスによる形状は常に C^1 連続な曲面となるため，変形の際に連続性のある滑らかな曲面を保つことができる。

しかし，手や指による形状の切削や付加といった変形の際に，切削の跡や付加された形状が手や指の形状になるようにするには，そのような変形に適した距離場関数を考える必要がある。また，没入型モデラでは，3次元形状の変形やユーザの視点移動による画像を実時間で提示する必要がある。陰関数表現による形状のレンダリングには一般的にレイトレーシング法が用いられるが，実時間提示には向かない。そこで，陰関数表現を実時間でレンダリングする手法を考える必要がある。これらの問題については3章で述べる。

3. 没入型曲面形状モデラ

本章では，仮想環境に没入しながら，手と指を用いて対話的に曲面形状を変形することが可能な没入型曲面形状モデラについて述べる．まず 3.1において，試作した没入型曲面形状モデラ（以下，試作モデラ）の概要を説明し，手と指を用いて行う 3 つの曲面形状変形操作について述べる．

3.1 試作モデラの概要

試作モデラでは，仮想環境にあらかじめ用意された球や円筒などのプリミティブに対し，HMD(Head Mounted Display)を装着し仮想環境に没入したユーザが，手形状入力装置を装着した手で変形を加えることによってモデリングを行う．HMDと手形状入力装置には 3次元位置センサが取り付けられており，頭部と手の位置および姿勢を計測する．試作モデラを使用している様子を図 8に，システム構成を図 9に示す．なお，試作モデラでは，計算機として SGI 社 ONYX2(MIPS R10000, 195MHz, 16CPU)を用い，HMDとしてオリンパス社 Mediamask を，3次元位置センサとして Polhemus 社 3SPACE Fastrak を，手形状入力装置として日商エレクトロニクス社 Super Glove をそれぞれ接続して使用した．

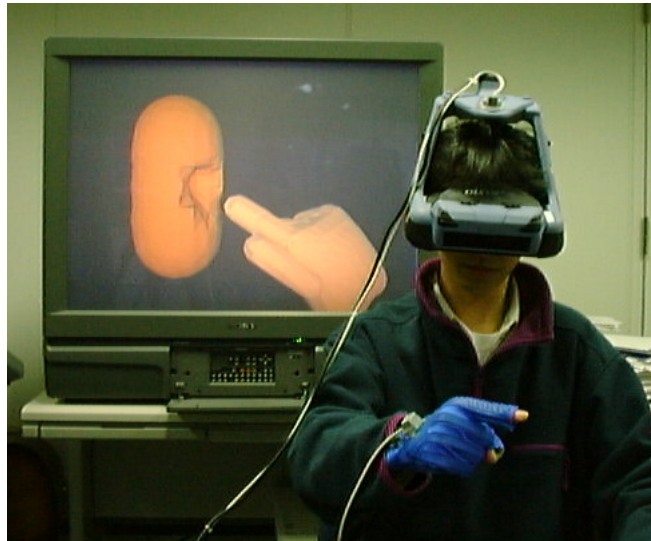


図 8 モデリングの様子

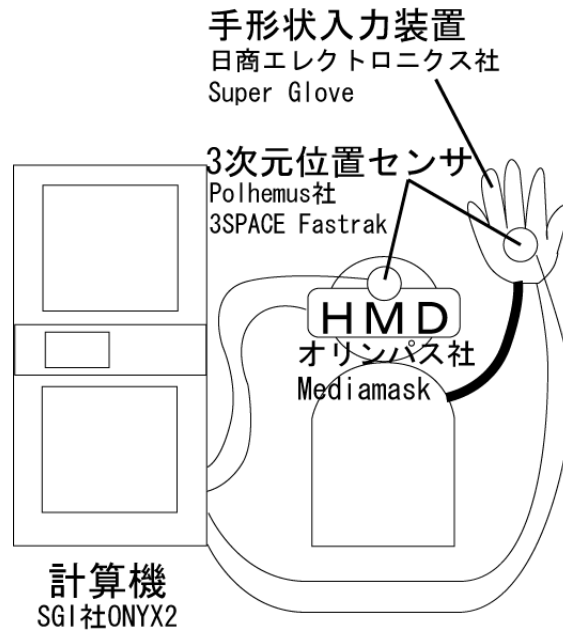


図 9 試作モデルのシステム構成

試作モデラでは、プリミティブ形状に対して、次の3つの形状変形操作を行うことができる。

人差し指による切削

曲面の切削はプリミティブ形状に対して人差し指を接触させ、接触部分のプリミティブ形状を指形状にあわせて変形させることにより行う。この時、人差し指を物体内部へ移動させると触れた部分が次々に切削されていくため、図10に示すように、ちょうど指でプリミティブ形状をへこませたように変形させることができる。切削されたプリミティブ形状は連続性のある滑らかな曲面を維持するため、切削跡は滑らかになる。

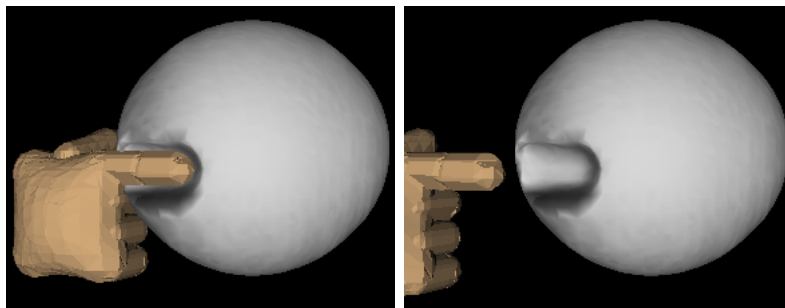


図10 人差し指による切削の様子

人差し指による付加

プリミティブに対する形状の付加は、形状付近に人差し指を近付けた時に、その指に向かって物体表面が飛び出すように形状を変形させることによって実現する。図11に示すように、指を動かすとその軌跡が曲面形状として残る。形状を付加したプリミティブ形状は連続性のある滑らかな曲面を維持するため、形状同士の接続部分は滑らかになる。

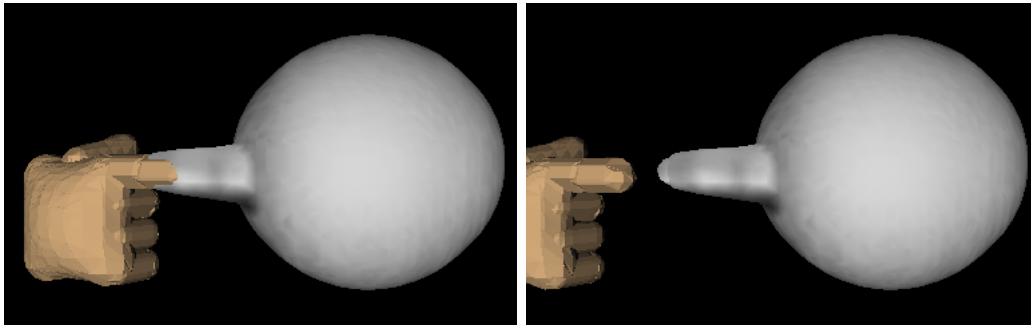


図 11 人差し指による付加の様子

手の平による切削

手の平による切削はプリミティブ形状に対して手の平を接触させ、接触部分のプリミティブ形状を平面に見立てた手の平にあわせて変形させることにより行う。このとき、図 12に示すように、手の平の法線方向の形状が残り、手の甲の法線方向の形状が切削される。ちょうど手の平でプリミティブ形状を押しつぶしたような変形が起こる。切削されたプリミティブ形状は連続性のある滑らかな曲面を維持するため、切削跡の輪郭部分は滑らかになる。

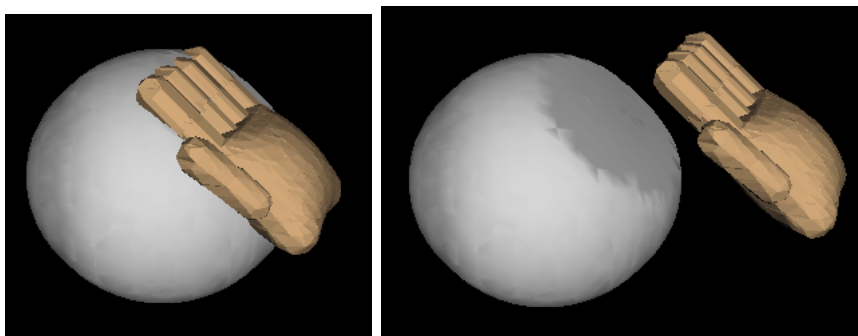


図 12 手の平による切削の様子

モデラの実行時には，図 13に示す仮想環境中の右下に設置されたスイッチを触ることで，切削モード，付加モード，平面切削モードを選択することができる．スイッチには現在のモードが，切削モード時には「S Subtractive」，付加モード時には図 13に示すように「A Additive」，平面切削モード時には「P Plane」のように表示される．

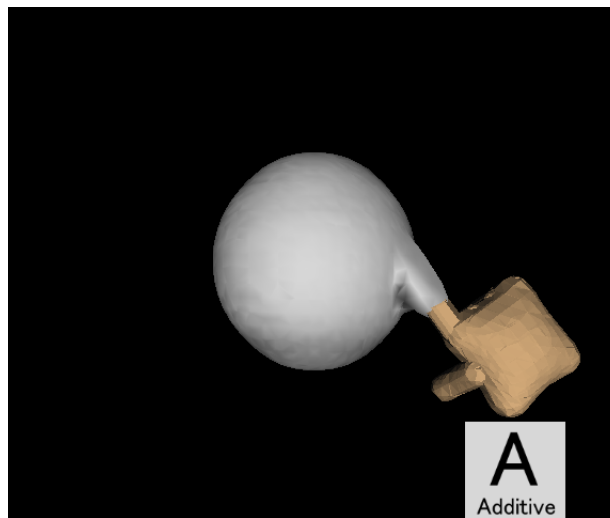


図 13 仮想環境の様子

次に，これらの 3 つの変形操作を実現するための試作モデラでのデータ処理の流れを図 14に示す．まず，3次元位置センサと手形状入力装置からの計測データをもとに，陰関数処理部で陰関数データを生成する．なお，3.2節で述べるが，陰関数データとは試作モデラで用いる陰関数表現における形状データである．次に，この陰関数データをポリゴン変換部で頂点や稜線といったポリゴンデータに変換する．そして，ポリゴンデータをグラフィクスライブラリの一つである OpenGL を用いてレンダリングし，HMD に出力するという流れで処理が行われる．3.2, 3.3節では，この陰関数処理とポリゴン変換について詳述する．

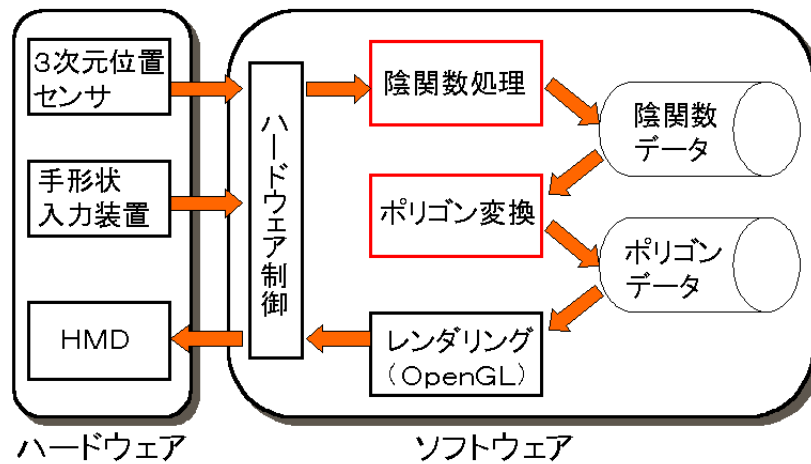


図 14 データ処理の流れ

3.2 陰関数処理

2章で述べた陰関数表現の特徴を利用して，形状のモデリングを行う．本節では 3.1節で述べた 3 つの変形操作について，それぞれで行われる変形処理について述べる．

まず試作モデラでは，2章で述べたスケルタルサーフェスの表現式 (5) を次の式 (6) のように改良して用いる．

$$f(P) = \sum_{i=1}^n w_i F_i(\sqrt{d(P, S_i)}) \quad (6)$$

このように式変形するのは，経験的に，表現式 (5) のままでは指形状に沿ってプリミティブを切削することができず，指形状の周囲が広範囲に渡って切削されてしまうためである．また，同様の理由から距離場関数 F として式 (7) に示すエルミート (Hermite) 曲線 [15] による関数を一貫して用いる．式 (7) において， $H_i^3(d)$ は式 (8) に示す 3 次エルミート関数であり， p_i ， v_i は図 15 に，距離 d に対する距

離場関数の変化の様子とともに示したベクトルである.

$$F(d) = \begin{cases} p_0 H_0^3(d) + v_0 H_1^3(d) \\ \quad + v_1 H_2^3(d) + p_1 H_3^3(d) & (0 \leq d < 1.0) \\ p_1 H_0^3(d - 1.0) + v_1 H_1^3(d - 1.0) \\ \quad + v_2 H_2^3(d - 1.0) + p_2 H_3^3(d - 1.0) & (1.0 \leq d \leq 2.0) \end{cases} \quad (7)$$

$$\begin{aligned} H_0^3(t) &= (2t + 1)(1 - t)^2 \\ H_1^3(t) &= t(1 - t)^2 \\ H_2^3(t) &= t^2(t - 1) \\ H_3^3(t) &= t^2(3 - 2t) \end{aligned} \quad (8)$$

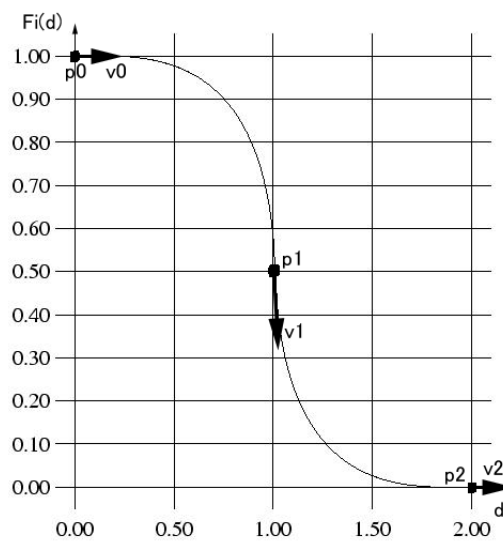


図 15 エルミート曲線による距離場関数

次に、プリミティブ形状が 3 つの点スケルトンから成る形状である場合を例に挙げて、前節で述べた 3 つの曲面変形処理についてそれぞれ説明する。プリミティブ形状 $f_1(P)$ は式 (6) から、点スケルトンを $PointS_i (i = 1 \sim 3)$ 、重みを

$w_1 = w_2 = w_3 = 1.0$ として次の式 (9) で表される.

$$f_1(P) = F(\sqrt{d(P, PointS_1)}) + F(\sqrt{d(P, PointS_2)}) + F(\sqrt{d(P, PointS_3)}) \quad (9)$$

人差し指による切削

人差し指形状 $f_2(P)$ は線スケルトンで表現し, 線スケルトンを $LineS$ として, 式 (10) で表される.

$$f_2(P) = w_1 F(\sqrt{d(P, LineS)}) \quad (10)$$

指形状 $f_2(P)$ によるプリミティブ形状 $f_1(P)$ の切削は, 図 16 に示すように, $f_1(P) - f_2(P)$ という減算によって表現することができ, 変形後の形状 $f(P)$ は, 指形状の重みを $w_1 = -1.0$ として式 (11) のようになる.

$$\begin{aligned} f(P) &= f_1(P) - f_2(P) \quad (11) \\ &= F(\sqrt{d(P, PointS_1)}) + F(\sqrt{d(P, PointS_2)}) \\ &\quad + F(\sqrt{d(P, PointS_3)}) - F(\sqrt{d(P, LineS)}) \end{aligned}$$

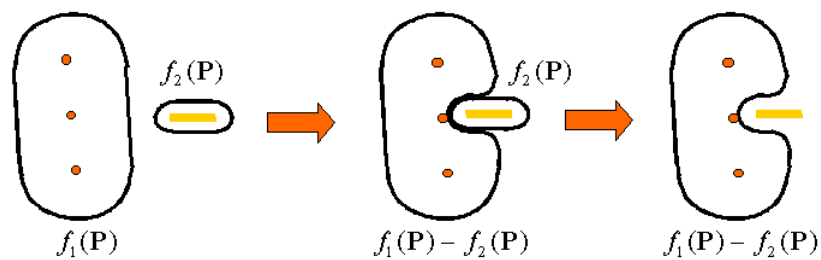


図 16 人差し指による切削の表現

人差し指による付加

人差し指形状 $f_2(P)$ は切削時と同様に線スケルトンで表現し、式 (10) で表される。指形状 $f_2(P)$ によるプリミティブ形状 $f_1(P)$ への付加は、図 17 に示すように、 $f_1(P) + f_2(P)$ という加算によって表現することができ、変形後の形状 $f(P)$ は、指形状の重みを $w_1 = 1.0$ として式 (12) のようになる。

$$\begin{aligned} f(P) &= f_1(P) + f_2(P) & (12) \\ &= F(\sqrt{d(P, PointS_1)}) + F(\sqrt{d(P, PointS_2)}) \\ &\quad + F(\sqrt{d(P, PointS_3)}) + F(\sqrt{d(P, LineS)}) \end{aligned}$$

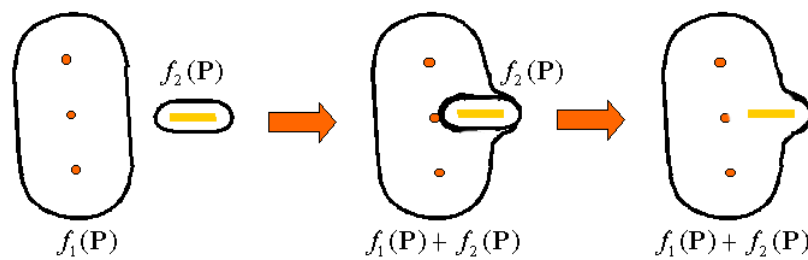


図 17 人差し指による付加の表現

手の平による切削

手の平形状 $f_2(P)$ は無限平面スケルトンで表現し、無限平面スケルトンを $PlaneS$ として、式 (13) で表される。

$$f_2(P) = w_1 F(\sqrt{d(P, PlaneS)}) \quad (13)$$

手の平形状 $f_2(P)$ によるプリミティブ形状 $f_1(P)$ の切削は、図 18 に示すように、 $f_1(P) - f_2(P)$ という減算によって表現することができ、

変形後の形状 $f(P)$ は、指形状の重みを $w_1 = -1.0$ として式 (14) のようになる。

$$\begin{aligned}
 f(P) &= f_1(P) - f_2(P) & (14) \\
 &= F(\sqrt{d(P, PointS_1)}) + F(\sqrt{d(P, PointS_2)}) \\
 &\quad + F(\sqrt{d(P, PointS_3)}) - F(\sqrt{d(P, PlaneS)})
 \end{aligned}$$

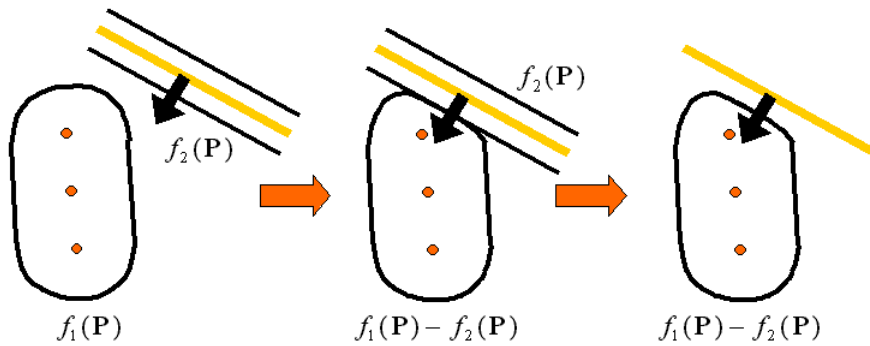


図 18 手の平による切削の表現

以上の手法では、 $f_1(P)$ と $f_2(P)$ を構成するスケルトンの形や位置といった幾何情報と距離場関数、減算または加算という属性を陰関数データとして保持することで、変形した形状を維持することができる。減算または加算においては、前述のように重み w をそれぞれ $w_1 = -1.0$ 、 $w_2 = 1.0$ として計算する。そこで、指形状のスケルトンの幾何情報、減算または加算という属性を、指形状が一定距離動くごとに取得し更新することによって、陰関数データを生成していく。

次節では、このようにして生成される陰関数データを実時間でレンダリング処理して表示する手法について述べる。

3.3 ポリゴン変換によるレンダリングの高速化

モデラの実行時には、陰関数表現を用いた形状を実時間で描画してユーザに提示する必要がある。陰関数表現では曲面上の任意の点での法線が容易に求まることから、その描画手法としてはレイトレーシング法を用いるのが一般的であるが [16, 17]、レイトレーシング法は実時間処理には向かない。そこで、本研究においては陰関数表現を用いた形状をポリゴンで近似し、ハードウェアによる高速なポリゴン描画手法を利用する。陰関数表現を用いた形状をポリゴンに変換する手法としては、スケルトンから多方向に伸ばした直線と曲面との交点を頂点としてポリゴンを生成する Desbrun らの手法 [18] や、ボクセルと曲面との交点を頂点としてポリゴンを生成する Bloomenthal らの手法 [19] などがある。Desbrun らの手法では陰関数多項式の減算によって生成される形状に対処することができないため、試作モデラに適用することはできない。Bloomenthal らの手法では減算にも対応できるため、試作モデラでは Bloomenthal らの手法を採用する。しかし、Bloomenthal らの手法を試作モデラに適用する際、実時間処理の妨げとなる問題点が生じる。そこで、この問題点を改善し実時間で処理を行うポリゴン変換手法を提案する。以下では、Bloomenthal らの手法とその問題点について述べた後、試作モデラにおけるポリゴン変換の高速化手法について述べる。

3.3.1 Bloomenthal らによるポリゴン変換手法

Bloomenthal らの手法では、変換対象形状とボクセルとの交点を求め、それを頂点としてポリゴンを生成する。以下に具体的なアルゴリズムを述べる。

1. 変換対象形状の周囲にボクセル空間を設定する。ここで、ボクセルとは単位立方体であり、ボクセルを格子状に配置したものをボクセル空間と呼ぶ。これを簡単のため 2 次元で表したものを図 19 に示す。

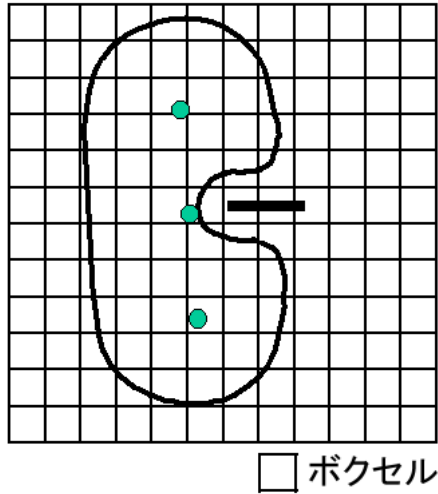


図 19 ボクセル空間

2. 図 20に示すように，変換対象形状の表面と交差するボクセルを探索しながら，探索ごとに以下の処理を繰り返す。

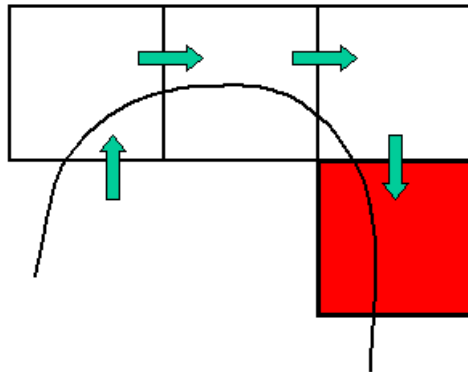


図 20 ボクセルの探索

- (a) ボクセルの頂点を4点ずつ組み合わせて6個の4面体に分割する。このとき図 21に示すように，6個の4面体それぞれが互いに隣接するように分割する。

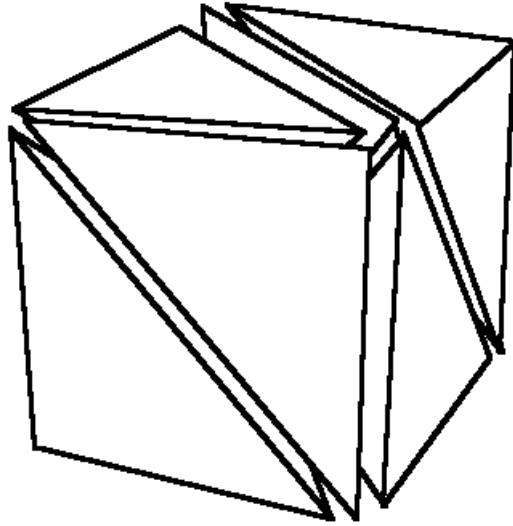


図 21 ボクセルの分割

- (b) 4 面体の各頂点座標における陰関数値を求める.
- (c) 図 22に示すように, 求められた陰関数値からその頂点が形状の内部であるか外部であるかが判別できるので, それを属性として頂点の情報に付加する.
- (d) 両端点の属性が異なる稜線は必ず形状と交差するため, その稜線について形状との交点を求める. 稜線と形状との交点を求めるということは, 稜線の端点間に含まれた陰関数式の根を求めることであり, これを非線形方程式の根を求める手法である二分法によって求める. 二分法とは, 関数のゼロ点を横切る区間の midpoint での関数の値を評価して符号を調べ, それと同符号の端点を midpoint と取り替えるという処理を繰り返す, 収束させて根を求めるという手法である [20].
- (e) 図 22に示すように, 求められた交点を基に, 三角形ポリゴンを生成する. 同時に, 求められた交点における法線を計算する.

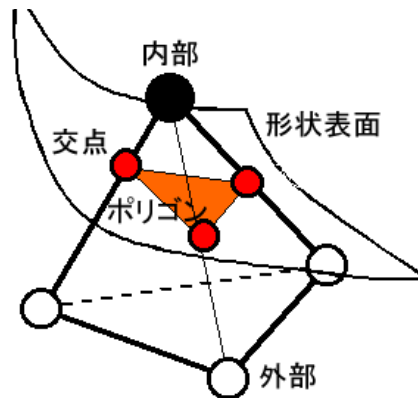


図 22 4 面体と形状との交点

以上で述べたポリゴン変換手法の利点を以下に整理する。

- ボクセル頂点の陰関数値を基に交点を推定するため、陰関数多項式の加算・減算によって生成される形状を変換することができる。
- 陰関数値が求められればポリゴン変換できるので、スケルタルサーフェス以外の陰関数表現にも使用できる汎用性の高い手法である。
- 形状表面探索後のボクセルごとの処理は独立しているため、並列に処理することができる。

しかし、以上述べた手法を試作モデラに適用する際に、以下のような問題点が生じる。

1. Bloomenthal らの手法では形状表面を探索しながらポリゴン変換を行うため、変換にかかる処理量が形状の表面積に比例する。そのため、試作モデラにおいて、形状の変形量が増加し表面積が増大するにつれて、ポリゴン変換処理にかかる時間も増加し、実時間処理ができなくなる。
2. 試作モデラでは陰関数値を計算するために陰関数データを参照する。ポリゴン変換処理の中でボクセルの頂点の陰関数値を求める処理があるが、モデリングが進むにつれて陰関数データは増加するため、ポリゴン変換の演算量も増加し、実時間処理ができなくなる。

次節では、これらの問題点を解決するために、スケルタルサーフェスの性質を利用したポリゴン変換の高速化手法を提案する。

3.3.2 曲面変形領域を考慮したポリゴン変換の高速化手法の提案

3.3.1節で述べた Bloomenthal らの手法を試作モデラに適用する際の問題点を改善するために、以下のような手法を提案する。

問題1に対して、試作モデラでは変形する領域は指形状の近傍のみであるので、モデラの初期化時に1度全ボクセルをポリゴン変換すれば、モデラの実行時には図23(a)に示す灰色のボクセル(変換領域)のみをポリゴン変換すれば良い。これによって、処理を軽減することができる。さらに、処理量が形状の表面積に依存しないため、形状の変形量が増加しても処理時間に影響しない。

問題2に対して、陰関数データの生成時にもボクセルを利用することを考える。2章で述べたように、スケルタルサーフェスでは、スケルトンごとに距離場関数の影響範囲があるため、図23(a)の変換領域が影響範囲内に入っている陰関数データのみを参照すれば良い。そのためにポリゴン変換用のボクセル空間とは別に、図23(b)に示すような陰関数データ用のボクセル空間を設定し、陰関数データ用ボクセルごとに陰関数データを記憶する。ポリゴン変換時には、変換領域の近傍となる、図23(b)に示す処理範囲内の陰関数データのみを参照することによって、処理を軽減するとともに、演算量の増加を抑えることができる。

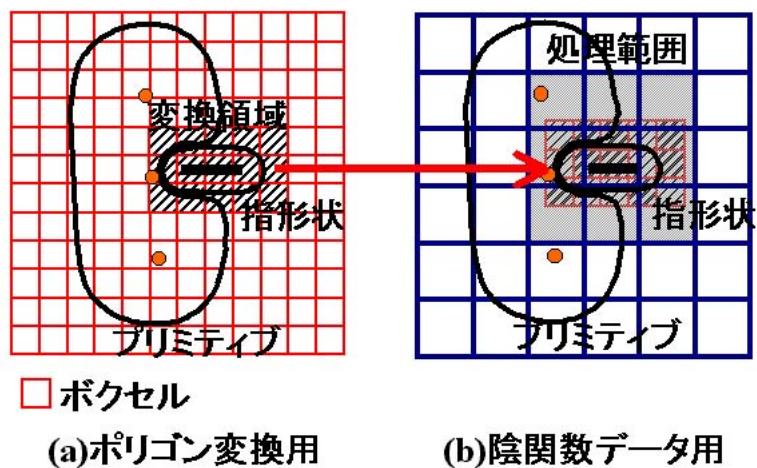


図 23 ポリゴン変換用ボクセル空間と陰関数データ用ボクセル空間

以上の手法を導入した，試作モデラにおけるポリゴン変換処理の手順を以下に示す．

1. 変換対象形状の周囲にポリゴン変換用ボクセル空間と陰関数データ用ボクセル空間を設定する．
2. 変換対象形状の表面と交差する全ポリゴン変換用ボクセルをポリゴンに変換する．
3. 変換領域にあるポリゴン変換用ボクセルに対して，以下の処理を繰り返す．
 - (a) ポリゴン変換用ボクセルを 6 個の四面体に分割する．
 - (b) 四面体の各頂点座標における陰関数値を求める．この時，変換領域と重なる陰関数データ用ボクセルに格納されている陰関数データを用いて計算する．
 - (c) 求められた陰関数値からその頂点の内外判定を行い，結果を属性として付加する．
 - (d) 両端点の属性が異なる稜線は必ず形状と交差するため，その稜線について形状との交点を二分法によって推定する．

- (e) 求められた交点を基に，三角形ポリゴンを生成する．同時に，求められた交点における法線を計算する．

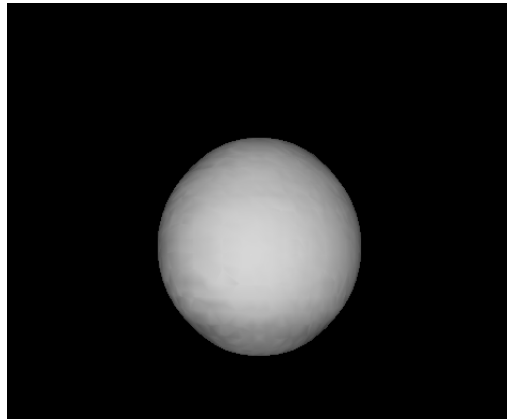
なお，3.3.1節においても述べたように，ボクセルごとのポリゴン変換を含めた以上の処理は，ボクセルごとに独立した処理であるため，これらを複数の CPU で並列に処理することが可能である．試作モデラにおいては，3.1節で述べた ONYX2 において 3 個の CPU を用いている．

以上の手法には，モデリングが進行するに従って陰関数データが増加し続けるという問題が残されている．そこで，無駄な陰関数データの増加を防ぐために，1 つの陰関数データ用ボクセル内に記憶できる陰関数データの最大数を設ける．最大数以上の陰関数データは生成できないため，最大数に達したボクセルの領域には，それ以上の変形を加えることができなくなる．このため，試作したモデラにおいては，1) ポリゴン変換用ボクセルの大きさ，2) 陰関数データ用ボクセルの大きさ，3) 1 ボクセルあたりの陰関数データの最大記憶数，をパラメータとして，これをユーザの希望するモデリングの複雑さと，計算機の性能に応じて設定することとした．

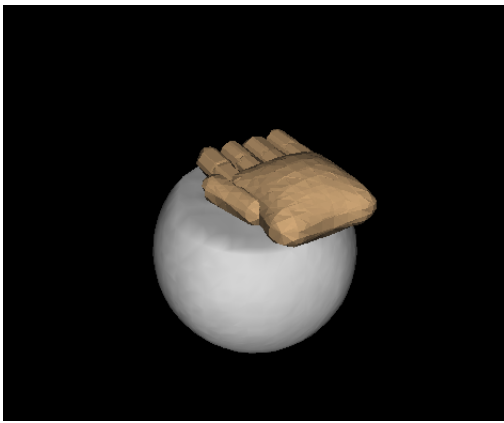
4. モデリング例

試作モデラによるモデリングの例として、ティーカップ、水槽の中の魚、立体手書き文字の3つを制作した。

はじめにティーカップを例に制作過程を順に説明する。まず図 24(1)のように球プリミティブを用意する。ティーカップの飲み口を作るために、まず図 24(2)のように球プリミティブの上部を手の平によって切削して平らにする。次に、図 24(3)のように指による切削で飲み口と中に入っている飲み物をモデリングしていく。指による切削を繰り返し、飲み口と中に入っている飲み物をモデリングした結果を図 24(4)に示す。そして、図 24(5)のように指による付加で取っ手の部分をモデリングしていく。



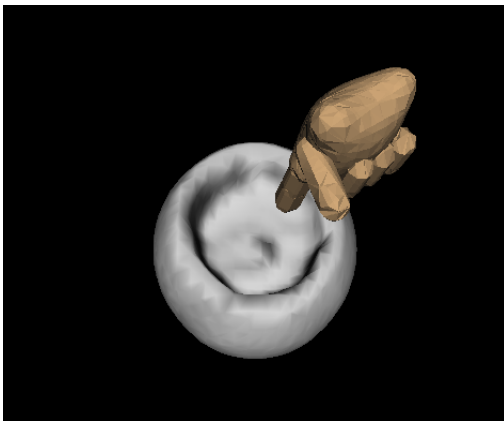
(1)



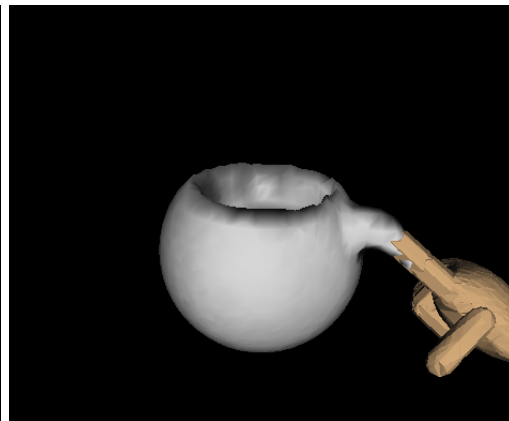
(2)



(3)



(4)



(5)

図 24 制作過程の例 (ティーカップ)

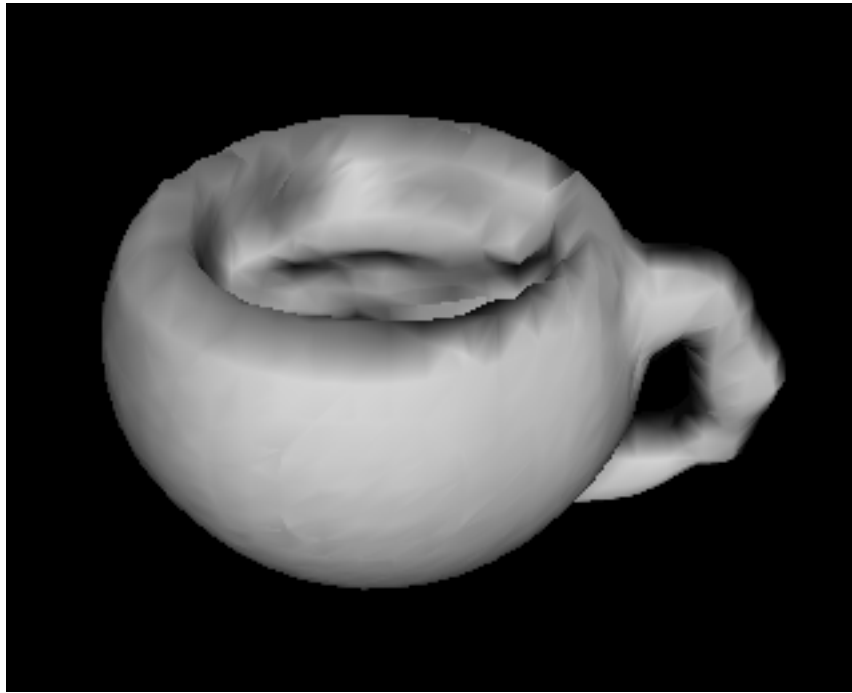


図 25 モデリング例 1：ティーカップ

最後に仕上げとして微調整を行い，図 25に示すティーカップが完成した．
飲み口をモデリングする際，飲み口の厚さを一定にするために慎重にモデリングをする必要があり手間取ったが，制作時間は約 20 分であった．

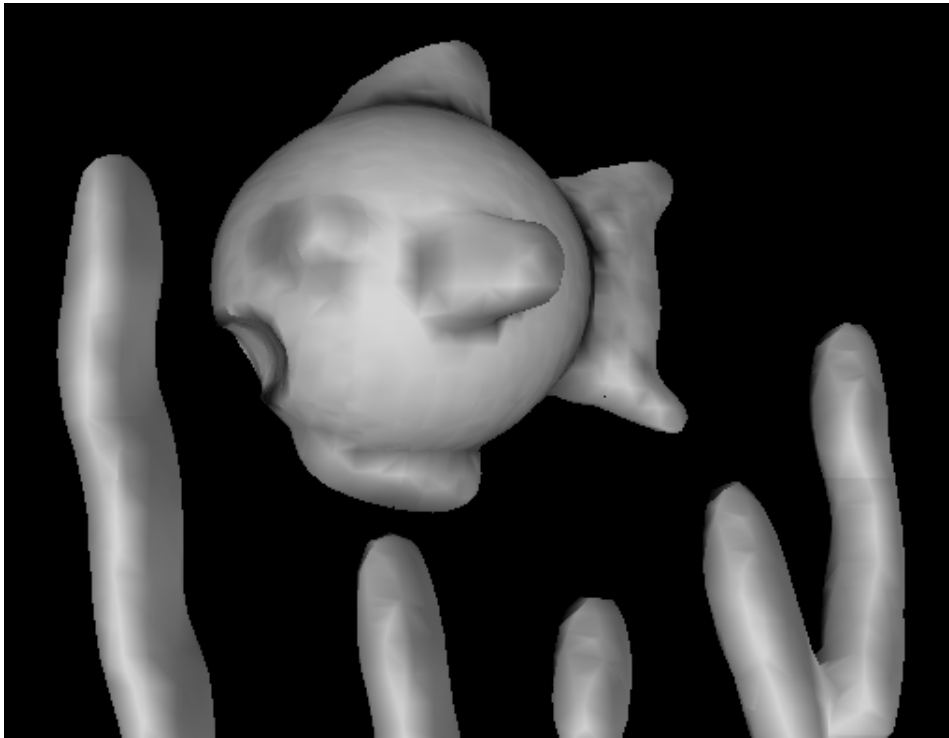


図 26 モデリング例 2：水槽の中の魚

図 26は水槽の中の魚をモデリングした例である。球プリミティブに指による付加で目やひれを付け、指による切削で口を開けた。そして、指による付加で海草をモデリングした。モデリング例 1 に比べると慎重な作業が少なかったため、形状が複雑な割にモデリング例 1 と同様に約 20 分で作成した。

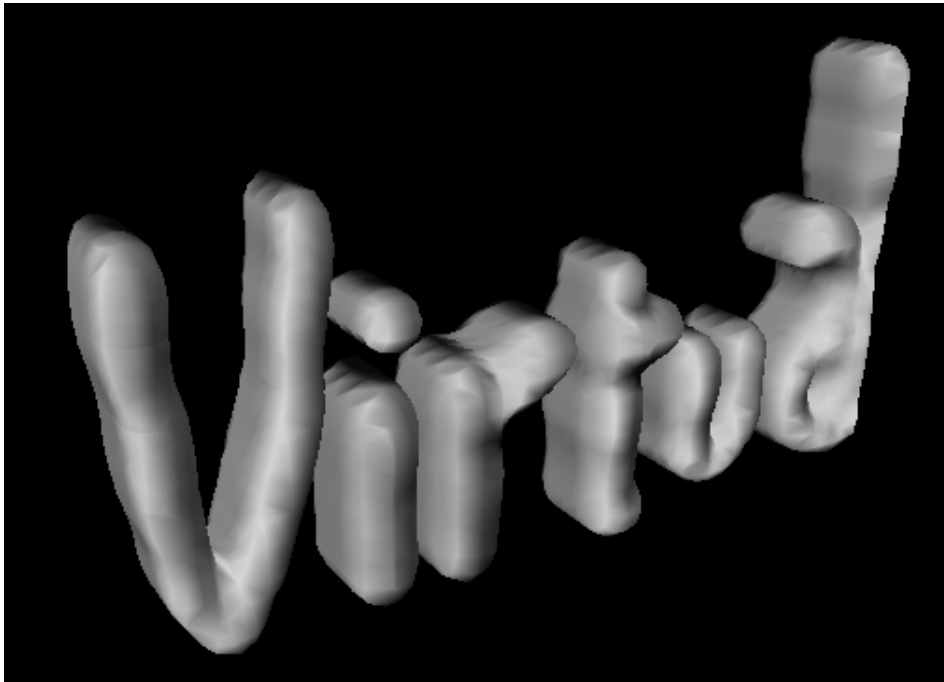


図 27 モデリング例 3：立体手書き文字

図 27は立体手書き文字のモデリング例である。付加モードにして指で文字を書くことによって、このような立体的な手書き文字を簡単にモデリングすることができる。制作時間は約 5 分であった。このように、試作モデラによって複雑で滑らかな曲面を持った形状を短時間でモデリングすることができる。

なお、モデリング作業時の画像更新レートは、CPUとしてMIPS R10000(195MHz)をポリゴン変換処理に 3 つ、その他の処理に 1 つ用いて、約 10 fps である。この時、3.3.2節で述べたモデリングの精度を制御するパラメータの値は、ポリゴン変換用ボクセルの大きさを $120 \times 120 \times 120$ 、陰関数データ用ボクセルの大きさを $48 \times 48 \times 48$ 、1 ボクセルあたりの陰関数データの最大記憶数を 10 個とした。

5. 考察

本章では、1章で挙げた曲面モデラの満たすべき4つの要件について、試作した没入型曲面形状モデラがこの条件を満たしているかを検証する。

1) 形状の変形操作が簡単で、実環境における動作と同じであること

試作モデラでは形状の変形操作として、仮想環境に没入して手の平で形状を切削したり、人差し指で形状を切削や付加することができ、実環境での動作によって曲面変形操作を実現していると言える。しかし、例えば実環境の粘土の変形において起こる、片側をへこませれば反対側が膨らむなどのような状況までは再現できておらず、実環境での動作に比べると違和感がある。また、形状に触れた時に反力がないため、実際のモデリング時には慎重に手を動かさなければ切削しすぎるような場合もあった。そこで、形状の材質感や反力を与えることによって、より実環境での動作に近づけ、それによるモデリングへの影響を調べる必要がある。

2) モデリング時に形状の視認が容易で、日常的に実環境において行っている動作と同じように、形状を立体的に様々な角度から見ることができると

HMDとそれに付けられた3次元位置センサを用いることで、仮想環境に没入してモデリングを行う試作モデラでは、日常的に実環境において行っている動作と同じように、形状を立体的に様々な角度から見ることができる(両眼立体視と運動立体視)。しかし、提示画像の更新レートが低いため、頭部の動きと提示画像に時間的なズレが生じ、視認が困難になるという問題点が明らかになった。ポリゴン変換手法において、画像更新レートに応じて動的にボクセルの大きさを変更し、変換精度を調整することによって、画像更新レートを高めるなど、さらなる改善の必要がある。

3) 変形操作を実行することによって曲面がどのように変形するかを予想することが容易で、望んだ通りの形状が得られること

試作モデラでは、スケルタルサーフェスにおける定義式に改良を加えることにより、指による形状の切削や付加の際に、切削した跡や付加した形状が指の形状になるようにした。これによって、切削や付加による形状の変形を予想することが容易になり、望んだ通りの形状を得ることができた。しかし、陰関数データの生成は指が一定距離動くたびに行われるため、この距離を不適切に設定すると指を動かしても形状の変形が起こらなくなり、操作に違和感が生じてしまう。現在は実験的に違和感のない距離を設定しているが、評価実験などによって定量的な評価を行う必要がある。

4) 扱える形状が連続性のある滑らかな自由曲面であること

常に連続性を保つという特徴を持ったスケルタルサーフェスを用いたことで、連続性のある滑らかな自由曲面を扱うことのできる没入型曲面形状モデラを構築することができた。今後さらに、連続性のある滑らかな自由曲面とともに、エッジのある形状を扱えるように拡張すれば、モデラの表現力を向上させることができると考えられる。

以上のように改善すべき点は残されているものの、試作モデラでは先述の4つの要件をほぼ満たすことができたと考えられる。

6. むすび

本論文では、曲面モデラの満たすべき要件として、1) 形状の変形操作が簡単で、実環境における動作と同じである、2) モデリング時に形状の視認が容易で、日常的に実環境において行っている動作と同じように、形状を立体的に様々な角度から見るができる、3) 変形操作を実行することによって曲面がどのように変形するかを予想することが容易で、望んだ通りの形状が得られる、4) 扱える形状が連続性のある滑らかな自由曲面である、という4点を挙げ、これらの要求を満たす没入型曲面形状モデラを試作し、形状制作実験を通して有効性を検証した。

上述の条件を満たすために、本研究では対象形状を立体的に視認することが可能な没入型仮想環境において、手や指を用いて直接的に変形作業を行うというアプローチを採用した。この実現のために、本研究では曲面形状の表現方法として形状間の衝突判定を高速に行うことが可能な陰関数表現の中から、指や手による変形操作に適しているスケルタルサーフェスを採用した。そして、スケルタルサーフェスの表現を用いて、形状の加減算によって、指による形状の切削と付加、手の平による形状の切削という操作を定義した。

モデリング作業を円滑に行うためには、変形操作の結果が速やかに画面に表示される必要があるが、そのためには陰関数表現で表現された曲面形状を高速にレンダリングする必要がある。この実現のために、本研究では従来から提案されている陰関数表現をポリゴンに変換する手法を、指や手の平によって変形されている領域を考慮することによって高速化する手法を提案した。

これらの手法を組み合わせることにより作成した試作モデラは、没入型仮想環境において、連続性のある滑らかな曲面形状を対話的にモデリングすることが可能であり、先に挙げた4つの条件を満足するものとなっている。本論文においては4章で試作モデラを用いて実際にモデリングを行った結果を示し、複雑で滑らかな曲面を持った形状を短時間でモデリングすることができることを確認した。試作モデラでは精密さや正確さに欠けるため、精密な工業製品などを正確にモデリングするような用途には向かないが、設計の初期モデルのモデリングや、図25から図27に示したような正確さが求められないデザイン作成などの用途に有用であり、仮想環境を利用した新たな造形手法を提供できたと考える。

今後の課題として、1) より精度の高いデータ計測の行える入力装置の検討、2) パラメータの自動決定手法の検討、3) 様々なユーザによる評価、が挙げられる。まず、1) に関して、試作モデルで用いた磁気式の3次元位置センサでは、計測データの精度が満足なものではなく、微妙な操作が困難であるという問題点がある。そこで、より精度の高いデータ計測の行える入力装置の検討が必要である。その際、手のジェスチャ認識やハプティックインタフェース [21] の導入についても検討する必要がある。また、2) に関して、実装において3.3.2節で述べたパラメータは、現在のところ経験的に決定しているという問題点がある。そこで、画像更新レートを維持しつつ、その画像更新レートにおいて最高の精度となるような、パラメータの自動決定手法についても検討する必要がある。この際、これらのパラメータは現在の陰関数データの生成、管理手法に起因しているため、陰関数データの生成、管理手法の改善や、生成された陰関数データの最適化などについても合わせて検討する必要がある。さらに、3) に関して、本論文においてはユーザによる評価を行っていない。初心者やCGクリエイターなど様々なユーザによる、直感性や対話性、モデリングの容易さなどについての評価を行う必要がある。

謝辞

本研究の全過程を通して，懇切なる御指導，御鞭撻を賜った，ソフトウェア基礎講座 横矢 直和 教授に心より感謝の意を表します。

本研究の遂行にあたり，有益な御助言を賜った，像情報処理学講座 千原 國宏 教授に厚く御礼申し上げます。

本研究の全過程を通じて，懇切丁寧なる御指導，御鞭撻を賜った，ソフトウェア基礎講座 竹村 治雄 助教授に心より厚く御礼申し上げます。

本研究を通して，日頃より有益な御助言，御指導を頂いたソフトウェア基礎講座 岩佐 英彦 助手，ならびに 山澤 一誠 助手に深く感謝致します。

物心両面において常に温かい御支援を頂いたソフトウェア基礎講座 大隈 隆史 氏，清川 清 氏，ならびに，ソフトウェア基礎講座の諸氏に深く御礼申し上げます。

最後に，日頃より温かく支えて下さった ソフトウェア基礎講座 事務補佐員 福永 博美 女史に心より感謝致します。

参考文献

- [1] AutoCAD Release 14 ユーザガイド, オートデスク株式会社, 1997.
- [2] Ray Dream Studio 5 ユーザガイド, MetaCreations Corporation, 1997.
- [3] LightWave 3D version 5.5 日本語ユーザーガイド, NewTek Incorporated, 1997.
- [4] T.A. Galyean and J.F. Hughes: “Sculpting: An Interactive Volumetric Modeling Technique,” *Proc. ACM SIGGRAPH '91*, pp.267-274, 1991.
- [5] S.W. Wang and A.E. Kaufman: “Volume Sculpting,” *Proc. ACM Interactive 3D Graphics*, pp.151-156, 1995.
- [6] R.A. Noble and G.J. Clapworthy: “Sculpting and Animating in a Desktop VR Environment,” *Proc. IEEE Computer Graphics International*, pp.187-195, 1998.
- [7] S. Coquillart: “Extended Free-Form Deformation: A Sculpturing Tool for 3D Geometric Modeling,” *Proc. ACM SIGGRAPH '90*, pp.187-196, 1990.
- [8] 水野, 岡田, 鳥脇, 横井: “仮想彫刻—仮想空間における対話型形状生成の一手法”, 情処学論, Vol.38, No.12, pp.2509-2516, 1997.
- [9] J.P.Y. Wong, R.W.H. Lau and L. Ma: “Virtual 3D Sculpturing with a Parametric Hand Surface,” *Proc. IEEE Computer Graphics International*, pp.178-186, 1998.
- [10] Y.H. Chai, G.R. Luecke and J.C. Edwards: “Virtual Clay Modeling Using the ISU Exoskeleton,” *Proc. IEEE VRAIS '98*, pp.76-80
- [11] 松宮, 清川, 竹村, 横矢: “陰関数表現を用いた仮想空間内曲面モデリングシステム”, 信学総大, A-16-35, 1997.

- [12] 松宮, 清川, 竹村, 横矢: “陰関数表現を用いた仮想空間没入型曲面形状モデラ”, 日本バーチャルリアリティ学会第3回大会論文集, pp.53-56, 1998.
- [13] B. Wyvill: “Animation and Special Effects,” in J. Bloomenthal(ed.), *Introduction to Implicit Surfaces*, Morgan Kaufmann Publishers, pp.242-269, 1997.
- [14] M.P. Cani-Gascuel and M. Desbrun: “Animation of Deformable Models Using Implicit Surfaces,” *IEEE Trans. on Visualization and Computer Graphics*, Vol. 3, No. 1, pp.39-50, 1997.
- [15] コンピュータグラフィックス 技術系 CG 標準テキストブック, 財団法人画像情報教育振興協会, 1995.
- [16] J.C. Hart: “Sphere Tracing: A Geometric Method for the Antialiased Ray Tracing of Implicit Surfaces,” *The Visual Computer*, Vol. 12, No. 10, pp.527-545, 1996.
- [17] G. Wyvill: “Ray Tracing Implicit Surfaces,” in J. Bloomenthal(ed.), *Introduction to Implicit Surfaces*, Morgan Kaufmann Publishers, pp.166-195, 1997.
- [18] M. Desbrun, N. Tsingos and M.P. Gascuel: “Adaptive Sampling of Implicit Surfaces for Interactive Modeling and Animation,” *Proc. Implicit Surfaces '95*, pp.171-185, 1995.
- [19] J. Bloomenthal: “An Implicit Surface Polygonizer,” in P. Heckbert(ed.), *Graphics Gems IV*, Academic Press, pp.324-349, 1994.
- [20] W.H. Press, S.A. Teukolsky, W.T. Vetterling and B.P. Flannery, 丹慶, 奥村, 佐藤, 小林 (訳): NUMERICAL RECIPES in C 日本語版, 技術評論社, 1993.
- [21] T.V. Thompson II, D.E. Johnson and E. Cohen: “Direct Haptic Rendering of Sculptured Models,” *Proc. ACM Interactive 3D Graphics*, pp.167-176, 1997.