

修士論文

局所的に細密なデータ構造を持つ陰関数表現を用いた  
没入型仮想彫刻モデラ

海部 陸

2002年2月8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。

海部 陸

審査委員： 横矢 直和 教授  
千原 國宏 教授

# 局所的に細密なデータ構造を持つ陰関数表現を用いた 没入型仮想彫刻モデラ\*

海部 陸

## 内容梗概

仮想彫刻と呼ばれるモデラでは、盛る・削るといった直感的な操作で3次元自由形状のモデリングを行うことができる。本論文では、形状の部分的な拡大表示を行い、そこにより細かな変形を加えることができる没入型仮想彫刻モデラを提案する。本モデラでは、3次元形状は陰関数で表され、格子の頂点に置かれたスケルトンと距離場関数によって、その形状が決定される。この格子の間隔と距離場関数を変えることにより、加える変形の細かさを制御することが可能であり、これらをツールサイズごとに設定する。細かな変形を加えるときにはスケルトンを局所的に細密な格子の頂点に置くというデータ構造を用いる。また、陰関数表現を効率よくポリゴンに変換する手法を用いて高速にレンダリングを行う。この時、拡大の度合いに応じて適切な数のポリゴンに変換することにより、拡大時に描画が粗くなることを防ぐ。最後に、試作したモデラによって、実際に3次元形状のモデリングを行い、細かな表現を含む形状を簡単かつ直感的な操作でモデリングできることを示す。

## キーワード

仮想彫刻，3次元形状モデリング，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-MT0051022, 2002年2月8日.

# Immersive Virtual Sculpting

## Using Implicit Surface Representations

### with a Locally Detailed Data Structure \*

Riku Kaibe

#### Abstract

A virtual sculpting system enables a user to design 3-D free-formed solid objects intuitively by carving or adding materials. This paper presents an immersive virtual sculpting scheme which enables a user to zoom in a part of 3-D solid object and make the part detailed by carving or adding materials. In the proposed modeler, objects are represented by implicit surfaces based on point skeletons arranged on sampling grids and field functions. The level of detail of shape deformation depends on the size of sampling grids and field functions, which are set up in every tool size. The detailed deformations have the data structure of locally detailed sampling grids. To reduce the time required for rendering the surfaces and keep the quality of rendering, implicit surfaces are converted to an appropriate number of polygon patches according to the degree of the zoom. Experiments have shown that a user of the prototype modeler can easily design 3-D solid objects of complex shapes in an intuitive manner.

#### Keywords:

virtual sculpting, 3-D modeling, implicit surface, polygonization

---

\* Master's Thesis, Department of Information Systems, Graduate School of Information Science, Nara Institute of Science and Technology, NAIST-IS-MT0051022, February 8, 2002.

# 目次

1. はじめに	1
1.1 従来の仮想彫刻モデラ	1
1.2 本研究の方針	11
2. 没入型仮想彫刻モデラのインタフェース	15
2.1 没入型仮想彫刻モデラの概要	15
2.2 没入型仮想彫刻モデラの操作	16
3. 没入型仮想彫刻モデラの実現手法	21
3.1 形状の表現手法	21
3.1.1 陰関数による形状の表現	21
3.1.2 局所的に細密なデータ構造とその生成	25
3.2 形状の拡大表示手法	30
3.2.1 実時間処理のための高速描画	30
3.2.2 描画精度の変更	33
3.2.3 拡大の度合いとツールサイズの組み合わせ	35
4. モデリング例と考察	36
4.1 モデリング例	36
4.2 考察	47
5. むすび	51
謝辞	53
参考文献	54

# 目 次

1	3面図	2
2	形状の組み合わせを示した構成図	2
3	メニュー	3
4	ベジエ曲線	4
5	Studio Max における形状プリミティブ	4
6	バンプマッピング	5
7	全体表示用の形状と拡大用表示用の形状	6
8	ボクセル表現	8
9	パラメトリック表現	9
10	細かなパーツのモデリング	11
11	微妙に曲がった線状の削り取り	11
12	尖った形状のモデリング	12
13	ボクセルの細分化	13
14	制御点の増加	13
15	モデリング作業の様子	15
16	初期形状とツール	16
17	没入型仮想彫刻モデラのシステム構成図	17
18	サイズ可変の盛り付けツール	18
19	サイズ可変の削り取りツール	19
20	拡大表示機能	20
21	陰関数表現における形状の内外判定	22
22	距離場関数の例	23
23	スケルトン	23
24	距離場関数の制御点 CP の傾きによる影響	24
25	距離場関数の横軸のスケールによる影響	25
26	盛り付けの表現	26
27	削り取りの表現	26
28	局所的に細密なデータ構造	27

29	スケルトンの配置 . . . . .	27
30	ツールサイズとスケルトン配置用格子 . . . . .	29
31	エルミート曲線による距離場関数 . . . . .	29
32	ポリゴン変換用格子 . . . . .	30
33	ポリゴン変換用格子の分割 . . . . .	31
34	4面体と形状との交点 . . . . .	32
35	描画精度の変更 . . . . .	33
36	拡大の度合いとポリゴン変換用格子 . . . . .	34
37	ツールサイズと拡大の度合い . . . . .	35
38	モデリング例：雪だるま . . . . .	36
39	顔と帽子のモデリング . . . . .	37
40	拡大部分 (顔) の指定 . . . . .	37
41	顔の拡大表示 (拡大の度合い：2段階) . . . . .	38
42	口，鼻，目のモデリング . . . . .	38
43	雪だるまの顔 . . . . .	39
44	図 43 の四角で囲んだ部分の拡大表示 (拡大の度合い：5段階) . . . . .	39
45	モデリング例：鳥 . . . . .	40
46	翼の拡大表示 (拡大の度合い：2段階) . . . . .	41
47	図 46 の四角で囲んだ部分の拡大表示 (拡大の度合い：3段階) . . . . .	41
48	図 47 の四角で囲んだ部分の拡大表示 (拡大の度合い：5段階) . . . . .	42
49	モデリング例：カニ . . . . .	43
50	目の拡大表示 (拡大の度合い：2段階) . . . . .	44
51	一番下の右足の拡大表示 (拡大の度合い：4段階) . . . . .	44
52	モデリング例：イルカ . . . . .	45
53	口の拡大表示 (拡大の度合い：2段階) . . . . .	46
54	スケルトンを置いた場合 . . . . .	48
55	スケルトンを消去した場合 . . . . .	49
56	異なるツールサイズの削り取り . . . . .	50
57	異なるツールサイズの盛り付け . . . . .	50

## 表 目 次

1	提案モデラの特徴と主な機能 . . . . .	15
2	共通操作 . . . . .	17

# 1. はじめに

3次元コンピュータグラフィックス(CG)を用いた映画やテレビ番組の増加, ゲーム機の高性能化に伴う3次元CGを用いたゲームの増加により, 3次元CGは身近になった。その結果, 3次元CGの需要は年々高まっている。それに伴い, 3次元形状をモデリングするための技術はますます重要になっている。しかし, 3次元形状は依然としてCGデザイナーなどの熟練者によってモデリングされたものであり, 誰でも簡単にモデリングできるようにはなっていない。そこで, 近年, 仮想彫刻モデラと呼ばれる熟練者以外の人々でも簡単にモデリングが行えるモデラが開発されている。仮想彫刻モデラでは, 盛る・削るといった直感的な操作で容易に3次元自由形状のモデリングを行うことができる。以上については1.1節で詳しく述べる。仮想彫刻モデラにおいて, 形状の一部を拡大して細かい変形を加えることができれば表現力を挙げることができると考えられる。しかし, 従来 of 仮想彫刻モデラにおいて, これを実現しようとする問題が生じる。1.2節でその問題について述べた後, その問題を起こすことなく形状の一部を拡大して細かい変形を加えることが可能な仮想彫刻モデラの開発方針について述べる。

## 1.1 従来の仮想彫刻モデラ

3次元形状のモデリングには一般的に, AutoCAD[1]などの3次元CADやShade[2], LightWave[3], Form・Z[4], Ray Dream Studio[5], 3D Studio Max[6]などの3次元CGソフトウェアにおけるモデラを用いる。

これらの3次元CADや3次元CGソフトウェアにおけるモデラでは, 一般的に図1に示すような上面図, 正面図, 側面図からなる3面図により, 多方向から3次元形状を見てその形状を把握する。また, 形状の組み合わせを示した構成図により, 作成した形状を個別のパーツとして階層構造で管理する。図2の四角で囲んだ構成図は, 同図のカップとスプーンの組み合わせを示したものである。

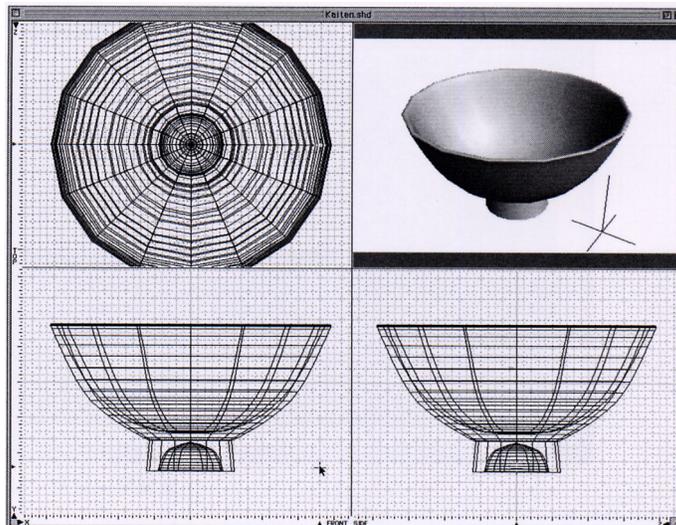


図 1 3面図

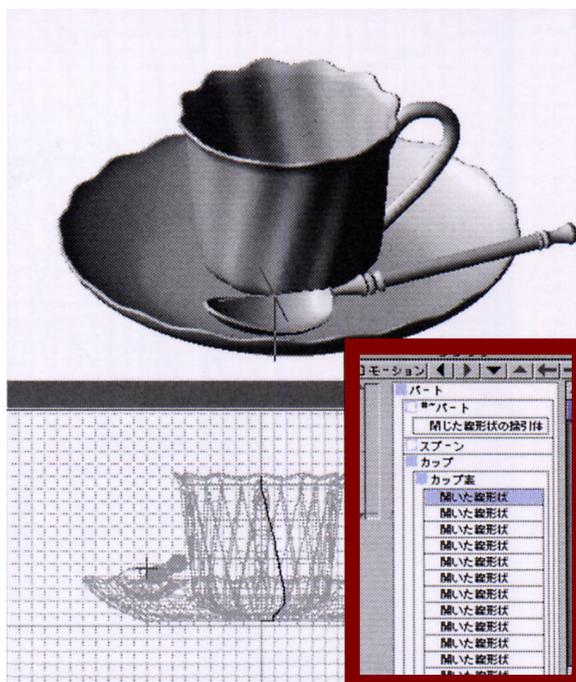


図 2 形状の組み合わせを示した構成図

3次元形状の移動や回転などのモデラが持つ機能を実行するには、マウスなどの入力装置を用いて、図3に示すようなメニューからその機能を実行するための項目を選択する。そして、竹ひご細工のように3次元形状の骨格を曲線の組み合わせ

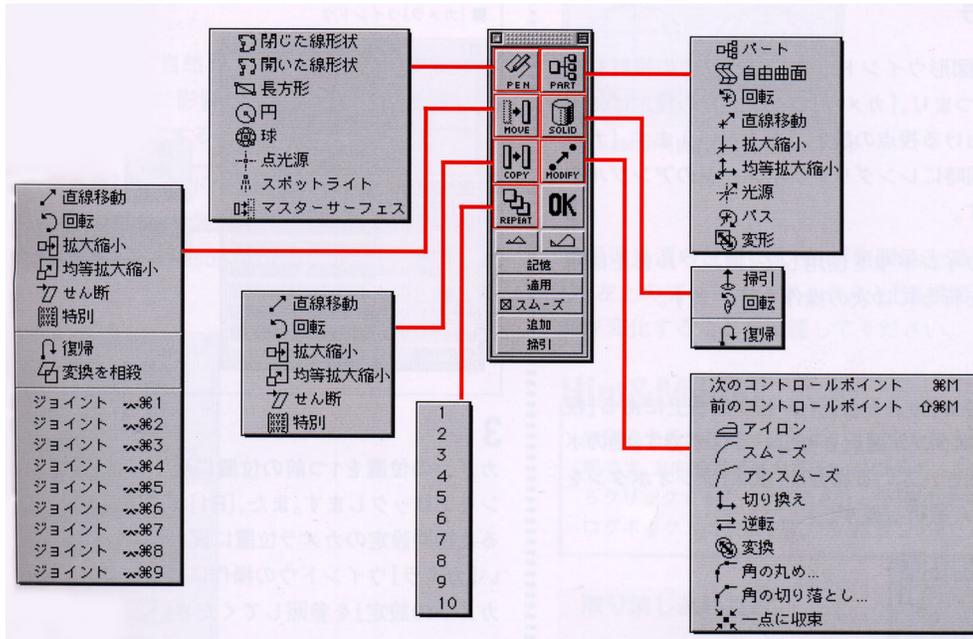


図3 メニュー

わせで表す方法、あらかじめ用意された形状プリミティブを組み合わせる方法、平面形状を作成しその形状を垂直方向に平行移動させ、その軌跡に3次元形状を生成する方法などでモデリングを行う。例えば、Shade[2]では3次元形状の骨格はベジエ曲線と呼ばれる曲線を使用する。このベジエ曲線は図4に示すように、制御点と呼ばれる点とハンドルと呼ばれる線をマウスで調整することにより描く。3D Studio Max[6]では、図5に示すようなボックス、円錐、球、天球体などの形状プリミティブが10種類用意されており、マウスのドラッグなどによって形状プリミティブのサイズの変更や複数の形状プリミティブの組み合わせを行う。

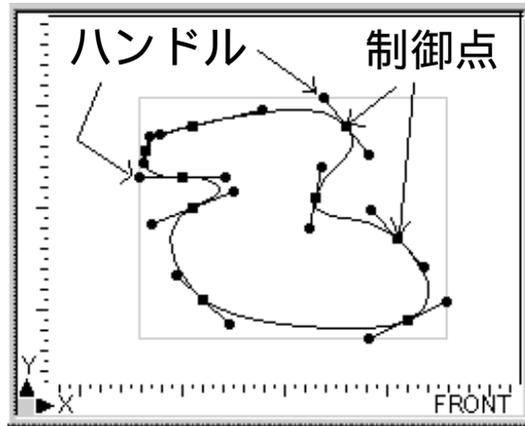


図 4 ベジエ曲線

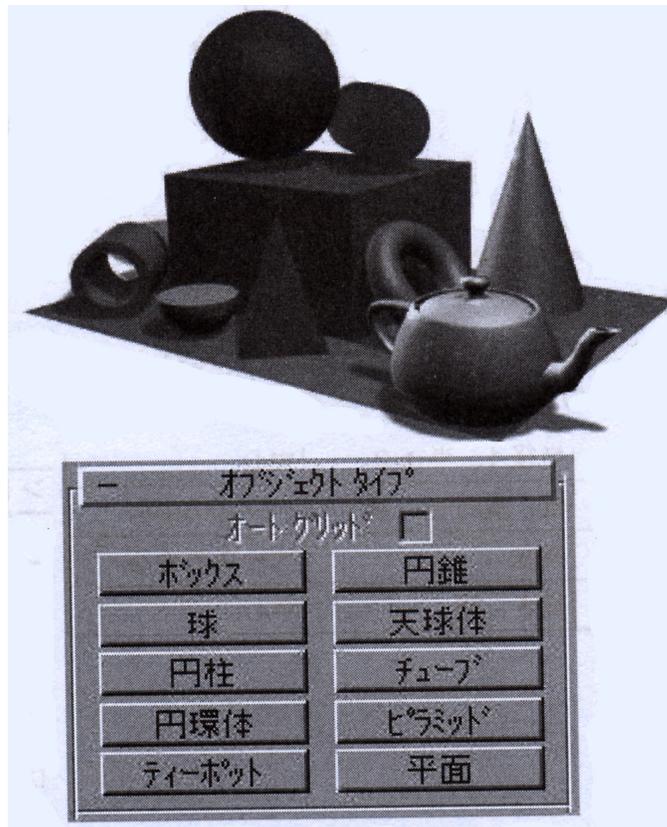


図 5 Studio Max における形状プリミティブ

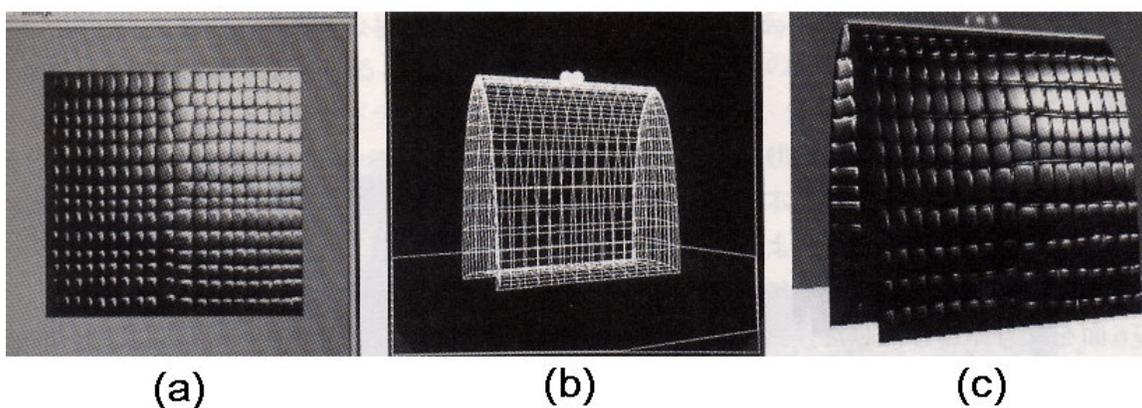


図 6 バンプマッピング

以上のモデリング方法で全ての3次元形状を作成するのではなく、以下のような様々な方法を組み合わせて3次元形状を作成する。

ハンドバッグの皮革などの細かい凹凸のある形状を上述のモデリング方法で作成するのは困難である。そのような細かい凹凸のある形状は、バンプマッピング [11] という方法で作成することができる。バンプマッピングは物体表面にテクスチャの凹凸をマッピングし、これにより物体表面の法線ベクトルを傾かせることで、シェーディングにおいて擬似的に細かな凹凸を表現する手法である。バンプマッピングの例として、皮革テクスチャ図 6(a) をハンドバッグ図 6(b) にマッピングした例を図 6(c) に示す。

また、あらかじめ決まった視点の移動を行うアニメーションやゲームで用いる3次元CGを効率よく作成する方法がある。これは、それぞれの視点に合わせて、別々の形状をモデリングする方法である。例えば、視点が図 7(a) の鳥瞰図を見る位置から図 7(a) の四角で囲んだ部分を見る位置に移動する場合は、図 7(b) のように拡大表示用の形状を用意しておき、視点が移動したときに拡大表示用の形状を描画する。図 7(b) は、図 7(a) の四角で囲んだ部分を実際に拡大して細かいモデリングをしたものではなく、それに似せた大きな形状に対してモデリングを行った形状である。この視点の移動に対して、必要になる形状は図 7(a) の形状と

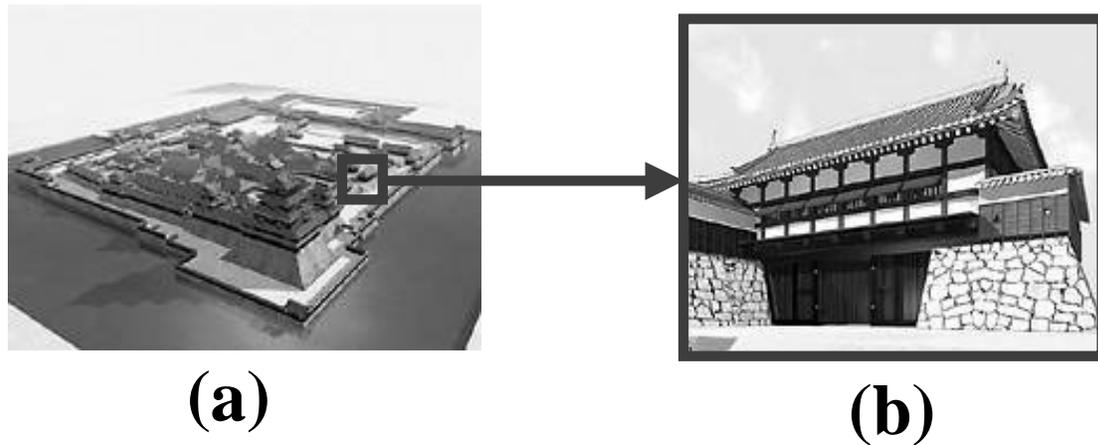


図 7 全体表示用の形状と拡大用表示用の形状

図 7(b) の形状だけである。つまり、他の部分の拡大表示用の形状を用意したり、図 7(a) の形状を実際に細かくモデリングする必要がない。

従来の 3 次元 CAD や 3 次元 CG ソフトウェアにおけるモデラでは、このような方法で 3 次元形状のモデリングを行う。しかしながら、これらの 3 次元 CAD や 3 次元 CG ソフトウェアにおけるモデラには、次のような問題点がある。

- 使用するには幾何や位相に関する数学的な知識や CG に関する専門的な知識が必要である。
- メニューにある多くの機能の把握、複雑なモデリング方法の習得、手法の組み合わせ方の検討などが必要で、モデリングを開始するまでにしなければならないことが多い。
- 粘土細工や彫刻などの実環境における形状作成の動作と異なるために直感性に欠ける。
- マウスやキーボードなどの 2 次元の入力装置を用いて 3 次元形状のモデリ

ングを行うために，それらの入力装置では対応しきれない作業を行う場合に特殊操作が必要となる．例えば，3次元形状の奥行き方向への移動は，正面から3次元形状を見ている場合，2次元のマウスによるドラッグでは表現できないので，他の操作方法で表現しなければならない．

- 2次元座標系に変換された画像である3面図や透視投影図などを用いて，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容易ではない．
- 制御点の移動などによって，3次元形状を変形させるため，変形後の3次元形状がどのような形状になるかを予想することが容易ではない．

これらがCGデザイナーなどの熟練者以外の人々にとって3次元形状のモデリングが困難である理由である．よって，熟練者以外の人々でも簡単にモデリングが行えるようにするには，以上に挙げた問題点を改善するモデラが必要である．問題点を改善するモデラが持てばよいと考えられる特徴として，以下の4つが挙げられる．

1. 幾何や位相に関する数学的な知識やCGに関する専門的な知識がなくても使用できる．
2. 実環境における動作と同じ動作で，3次元形状の変形操作を容易に行える．
3. 3次元形状がどのように変形するかを容易に予想することができる．
4. 3次元形状を立体的に見ることにより，3次元形状の視認を容易に行える．

以上の特徴の全て，あるいはいくつかを持ち，盛る・削るといった直感的な操作で容易に3次元自由形状のモデリングを行えるモデラが開発されている [7, 8, 9, 10]．これらのモデラは仮想彫刻モデラと呼ばれ，これらを用いればCGデザイナーなどの熟練者以外の人々でも容易に3次元形状のモデリングを行うことができる．

以下に，従来の仮想彫刻モデラをその表現手法から，ボクセル表現を用いた仮想彫刻モデラ [7, 8] とパラメトリック表現を用いた仮想彫刻モデラ [9, 10] に分類し，それぞれについて説明する．

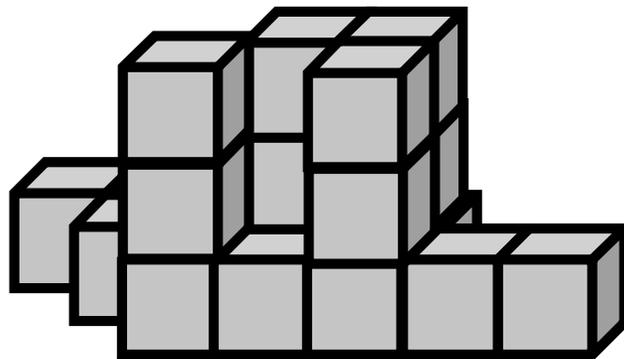


図 8 ボクセル表現

ボクセル表現を用いた没入型仮想彫刻モデラ [7, 8] では, ボクセルが形状の内部にあるか外部にあるかの情報を更新することによって変形を表現する. Galyeanらの仮想彫刻モデラ [7] では, 専用の3次元デバイスを用いて, いろいろな形の削り取りツールや盛り付けツールなどを操作し, 現実の彫刻のように3次元形状のモデリングを行う. 盛り付けは, 盛り付けツールが動いた軌跡にツールの形の形状を盛り付けることによって行う. これは, その軌跡上にあるボクセルにそのボクセルが形状の内部と外部のどちらにあるのかという情報を与えることによって

実現する．削り取りは，削り取りツールが動いた軌跡上の形状を，削り跡がツールの形になるように削り取ることによって行う．これは，その軌跡上にあるボクセルにそのボクセルが形状の外部にあるという情報を与えることによって実現する．

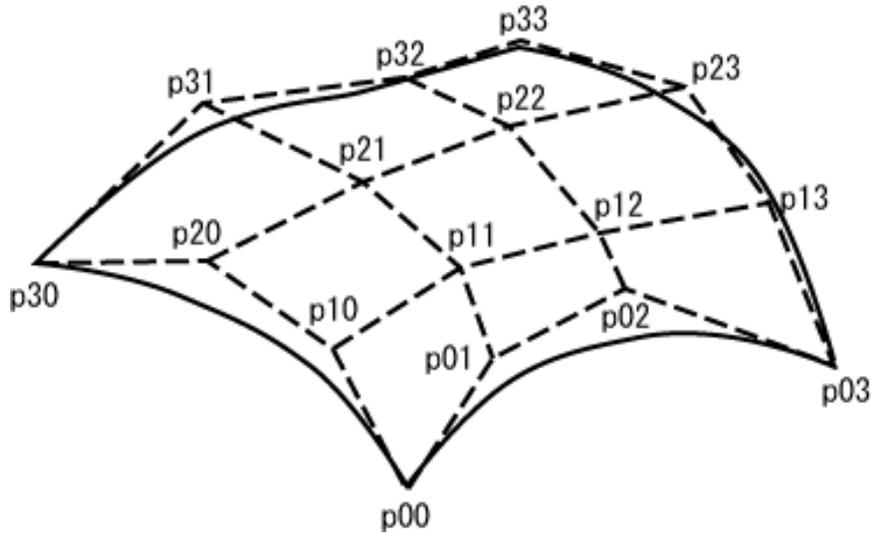


図 9 パラメトリック表現

#### パラメトリック表現を用いた仮想彫刻モデラ

パラメトリック表現 [11] とは，図 9 のように，制御点  $p_{ij}$  を用いたパラメータの陽関数形式で曲面を定義する表現方法である．パラメトリック表現には，双 3 次曲面、ベジエ曲面、有理ベジエ曲面、B スプライン曲面、NURBS 曲面、ベータスプライン曲面、クーンズ曲面、グレゴリ曲面など，様々な形式がある．代表的なパラメトリック曲面である B スプライン曲面 [11] は，格子状に並んだ制御点  $p_{ij}$  と  $u$  方向と  $v$  方向のノット列 (ノットベクトル)  $u_i, v_i$  によって定義される曲面である．ノットとは，曲面が定義されるパラメータ区間の境界の値である． $L \times K$  個

のパッチからなる  $n \times m$  次の B スプライン曲面はパラメータ区間  $u \in [u_n, u_{n+1}]$ ,  $v \in [v_m, v_{m+k}]$  によって定義される区分多項式曲面で, 式 (1) で表される.

式 (1) に示すように, B スプライン曲面上の点  $S(u, v)$  は, 制御点  $p_{ij}$  の加重平均になっている. したがって, パラメータ  $u, v$  が変化すると, 各制御点に対する重みである  $N_i^n(u)$  と  $N_j^m(v)$  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 B スプライン曲面となる.

$$S(u, v) = \sum_{i=0}^{n+L-1} \sum_{j=0}^{m+K-1} p_{ij} N_i^n(u) N_j^m(v) \quad (1)$$

ここで,  $N_i^n(u)$  と  $N_j^m(v)$  は,  $n$  次と  $m$  次の B-スプライン関数であり, ノット列を  $t_i$  とすると, 式 (2) で定義される.

$$N_i^n(t) = \frac{t - t_i}{t_{i+n} - t_i} N_i^{n-1}(t) + \frac{t_{i+n+1} - t}{t_{i+n+1} - t_{i+1}} N_{i+1}^{n-1}(t) \quad (2)$$

$$N_i^0(t) = \begin{cases} 1 & t \in [t_i, t_{i+1}) \\ 0 & t \notin [t_i, t_{i+1}) \end{cases}$$

パラメトリック表現を用いた没入型仮想彫刻モデラ [9, 10] では, 制御点を移動させたり, 制御点に対して比重値を設定して値を変更することにより, 変形を表現する. Raviv らの仮想彫刻モデラ [9] では, 3次元マウスを用いているいろいろな形の削り取りツールや盛り付けツールなどを操作し, 現実の彫刻のように3次元形状のモデリングを行うことができる. 盛り付けは, 盛り付けツールが動いた軌跡にツールの形の形状を盛り付けることによって行う. これは盛り付けられる形状がツールの形になるように軌跡上の制御点に対する比重値を変えることによって実現する. 削り取りは, 削り取りツールが動いた軌跡上の形状を, 削り跡がツールの形になるように削り取ることによって行う. これは削り跡がツールの形になるように軌跡上の制御点に対する比重値を変えることによって実現する.

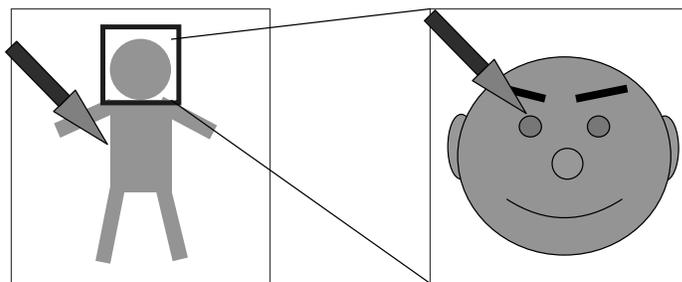


図 10 細かなパーツのモデリング

## 1.2 本研究の方針

仮想彫刻モデラ [7, 8, 9, 10] において、形状の部分的な拡大表示を行うことができれば、細かいモデリングが容易になると考えられる。例えば、人体のモデリングを行う場合に、図 10 のように顔を拡大表示できれば、目、口、鼻、耳、眉などのモデリングがしやすくなるし、手を拡大表示できれば、指や爪などのモデ

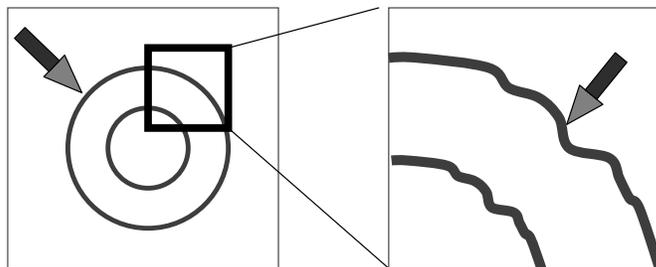


図 11 微妙に曲がった線状の削り取り

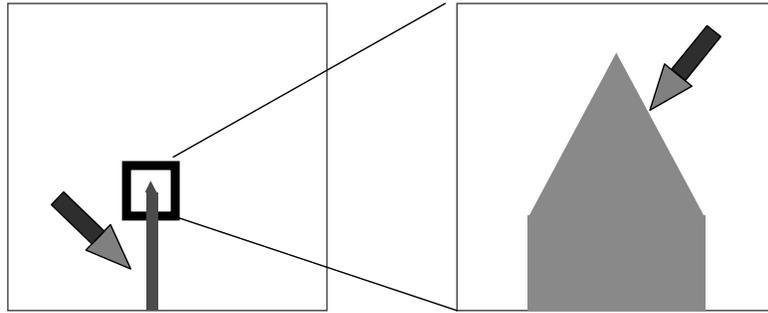


図 12 尖った形状のモデリング

やすくなる．また，鉛筆の先端などの尖った形状をモデリングする場合にも，図 12 に示すように，その部分を拡大できれば，尖り具合の調節がやすくなる．

しかし，従来の仮想彫刻モデラ [7, 8, 9, 10] において，形状の部分的な拡大表示を実現しようとする問題が生じる．また，加えられる変形の細かさに関して，従来の仮想彫刻 [7, 8, 9, 10] モデラでは，あらかじめ用意されたデータの解像度以上の変形しか加えることができず，それ以下の変形を加えられるようにしようとする問題が生じる．例えば，ボクセル表現を用いた仮想彫刻モデラ [7, 8] では，ボクセルを拡大することによって拡大表示を行えば，表示が粗くなる．また，ボクセルのサイズ以下の変形を加えることもできない．一方，図 13 のように，ボクセルを細かくする変換処理を行えば，表示が粗くならないようにしたり，もとのボクセルのサイズ以下の変形を加えることができるが，処理の増加や形状データの増加といった問題を招いてしまう．また，パラメトリック表現を用いた仮想彫刻モデラ [9, 10] においても，あらかじめ用意された制御点で表現できる細かさより細かい変形を加えるには，図 14 に示すように，制御点を増やす必要があり，ボクセル表現と同様に処理の増加や形状データの増加を招いてしまう．以上のように，ボクセル表現とパラメトリック表現では，形状の部分的な拡大表示

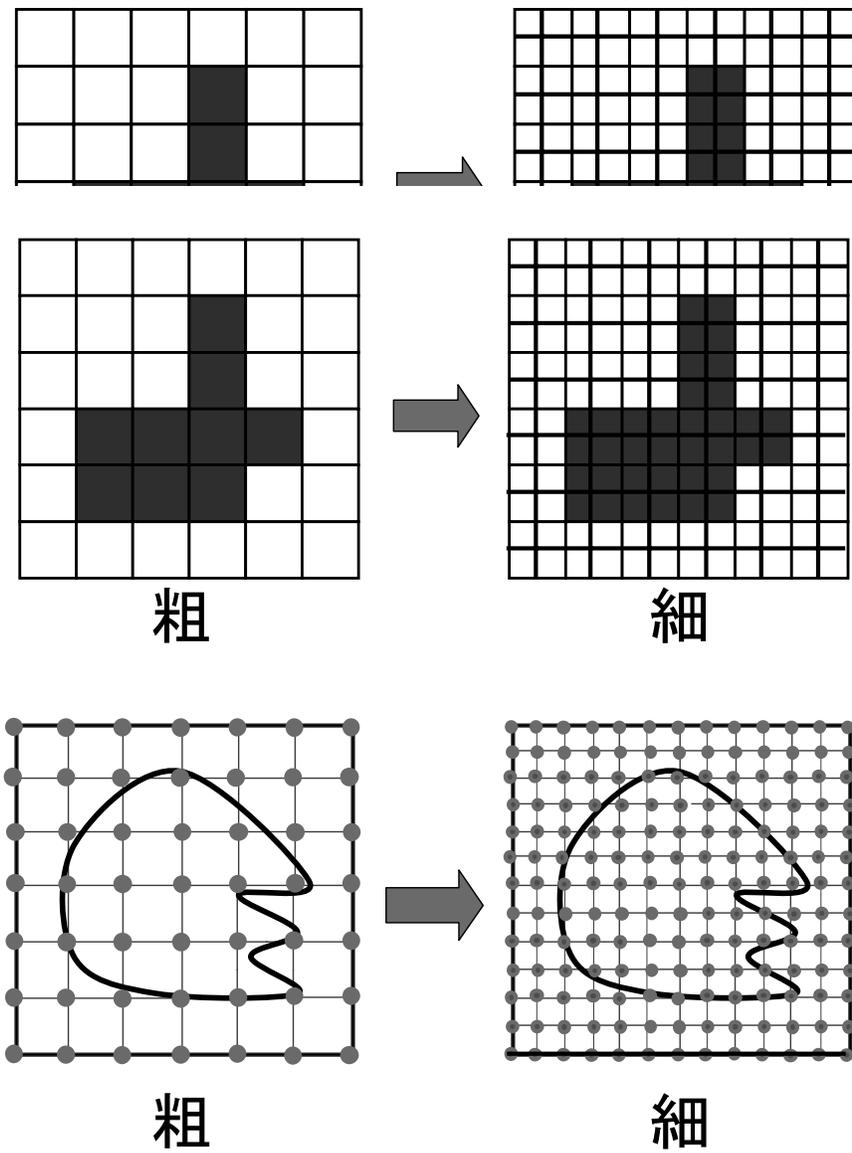


図 14 制御点の増加

やあらかじめ用意されたデータの解像度以下の変形を可能にしようとする、処理の増加や形状データの増加が生じてしまうという問題がある。

そこで、以下の二つの特徴を有することにより、これらの処理が不要で、CGデザイナーなどの熟練者以外の人々でも容易に細かい表現を含む3次元形状のモデリングを行えるモデラを開発する [12, 13]。

- 形状の一部を拡大表示する時に、形状データを増やさずに滑らかに描画できる。
- 付加する形状以外の形状データを増やさずに、拡大した形状に対して、あらかじめ用意された解像度以下の変形を加えられる。

これら2つの特徴を持たせるために、本研究では、従来の仮想彫刻モデラ [7, 8, 9, 10] で用いていたボクセル表現やパラメトリック表現ではなく、陰関数表現を用いる。形状を局所的に細密なデータ構造を持つ陰関数で表現することにより、付加する形状データのみを細密にすることが可能であり、付加する形状以外の形状データを細かくしたり増やしたりせずに細かい変形を加えることができる。また、陰関数で表した形状データをポリゴンに変換してレンダリングを行う。この時、拡大の度合いに応じて、形状データを適切な数のポリゴンに変換することにより、形状データを増やすことなく描画が粗くなるを防ぐことができる。

以下では、まず、2章で提案する没入型仮想彫刻モデラ (以下、提案モデラ) の概要とその操作方法について述べる。次に、3章で提案モデラの実現手法として、3次元形状の表現手法と3次元形状の拡大表示手法について述べる。4章で提案モデラによるモデリング例を示し、CGデザイナーなどの熟練者以外の人々でも容易に細かい変形を加えられることを確認した後に、考察を加え、最後にむすびとする。

## 2. 没入型仮想彫刻モデラのインタフェース

提案モデラの特徴と主な機能を表1に示す。本章では、まず、2.1節において提案モデラの概要とともにハードウェアの構成について述べる。次に、2.2節で表1に示す3つの機能、盛り付け、削り取り、拡大表示について詳しく述べる。

表1 提案モデラの特徴と主な機能

特徴	形状の一部を拡大表示して細かい変形を加えることが可能
機能	7段階のサイズ変更が可能な盛り付けツールによる盛り付け
機能	7段階のサイズ変更が可能な削り取りツールによる削り取り
機能	部分指定ツールによる5段階の拡大表示

### 2.1 没入型仮想彫刻モデラの概要

図15のように、提案モデラではHMDを装着したユーザが、立体視によって仮想空間へ没入してモデリング作業を行う。ユーザには図16に示すような球状の



図15 モデリング作業の様子

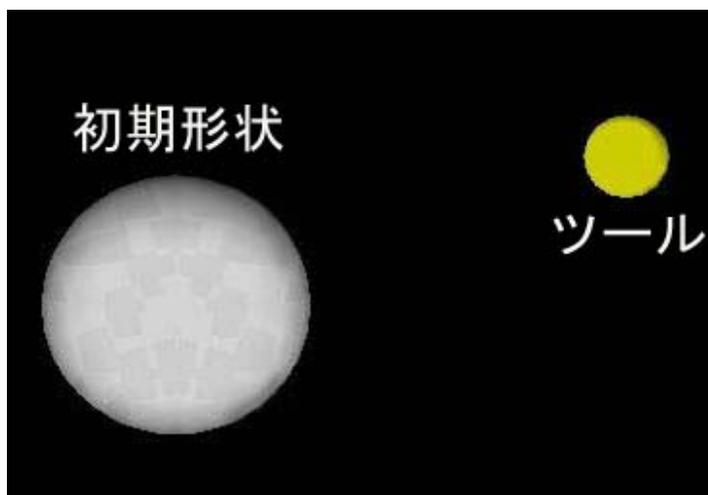


図 16 初期形状とツール

初期形状と球状の形状操作ツールが提供される。システムのハードウェア構成は図 17 のようになっており、仮想空間内の 3 次元形状の移動・回転を 3 次元マウスで行いながら、ペン型入力装置でツールを操作して、その形状に変形を加える。変形を加えるツールとして、盛り付けツールと削り取りツールがある。また、部分指定ツールを用いて形状の一部を拡大表示することができる。

なお、計算機は SGI 社 Onyx2(MIPS R10000, 195MHz, 16CPU) を用い、HMD として OLYMPUS 社 Mediamask を、3 次元位置センサとして Polhemus 社 3SPACE FASTRAK をそれぞれ接続して使用した。3 次元マウスとペン型入力装置は 3 次元位置センサを用いた自作品である。

## 2.2 没入型仮想彫刻モデラの操作

提案モデラでは、HMD を用いた立体視によって仮想空間へ没入したユーザが、盛り付け、削り取り、および、拡大表示の 3 つの機能を用いて 3 次元形状のモデ

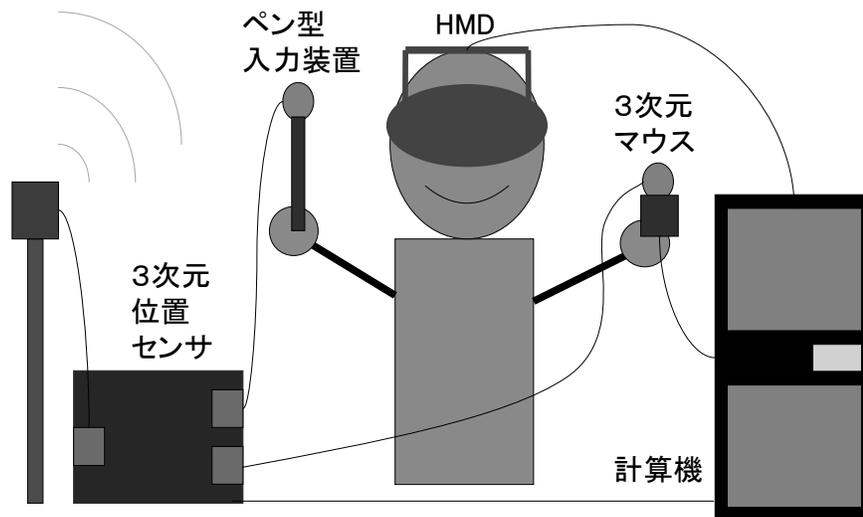


図 17 没入型仮想彫刻モデラのシステム構成図

リングを行う。全ての機能に共通する操作として、ツールの移動、ツールの種類の変更、モデリング中の3次元形状の移動・回転がある。これらの操作方法を表2に示す。なお、ツールの種類は、盛り付けツール、削り取りツール、部分指定ツールの順に変更する。次に、盛り付け、削り取り、拡大表示の3つの機能における個別の操作について詳しく述べる。

表 2 共通操作

操作の種類	入力装置の操作方法
ツールの移動	ペン型入力装置を移動させる
ツールの種類の変更	3次元マウスの右ボタンを押す
3次元形状の移動・回転	3次元マウスを移動・回転させる

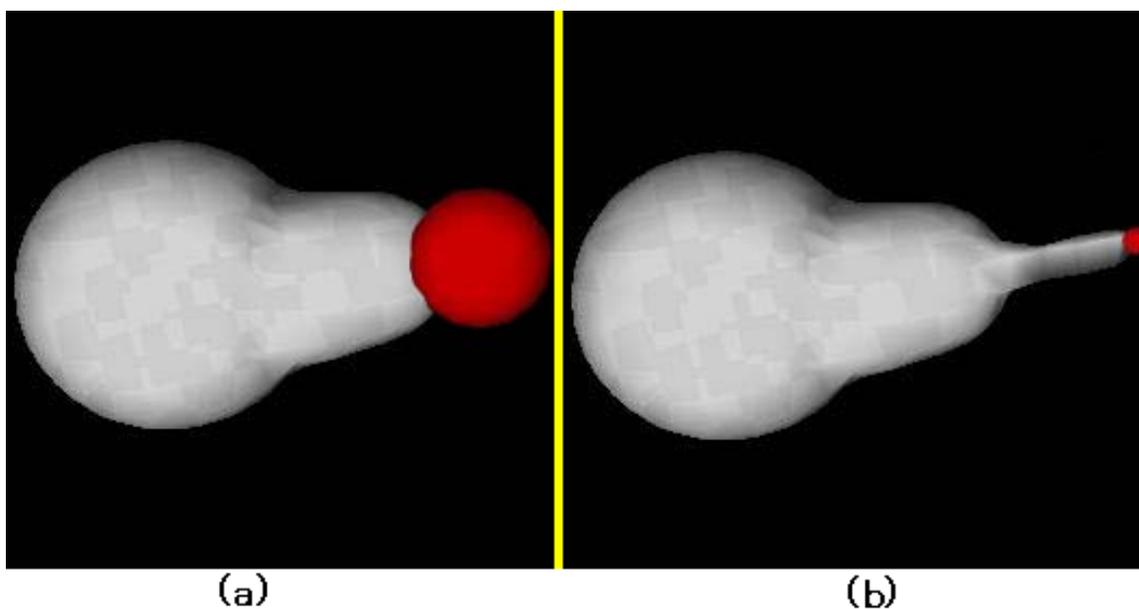


図 18 サイズ可変の盛り付けツール

### (1) 盛り付け

まず、3次元マウスの右ボタンを押すことで、盛り付けツールを選択する。盛り付けツールが動いた軌跡に形状が連続的に盛り付けられていく。盛り付けられた形状が他の形状と接触する場合は滑らかにつながる。また、3次元マウスの左ボタンを押すことで、ツールのサイズを変更し、盛り付ける形状のサイズを変えることができる。ツールのサイズは7段階用意されており、サイズは1段階増えると $\frac{3}{2}$ 倍大きくなる。先に示した図16のツールサイズは6段階目のサイズである。図18(a)は図16に示す初期形状に1番大きな7段階目の盛り付けツールで盛り付けた様子である。図18(b)はその盛り付けた部分に、2段階小さくした5段階目の盛り付けツールでさらに盛り付けた様子である。

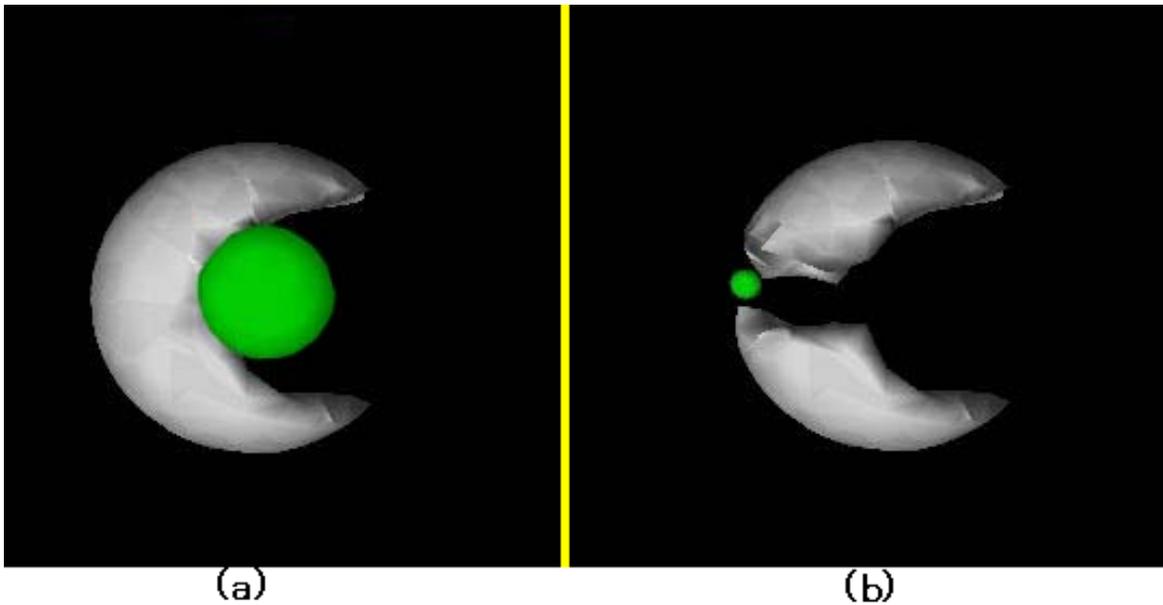


図 19 サイズ可変の削り取りツール

## (2) 削り取り

まず、3次元マウスの右ボタンを押すことで、削り取りツールを選択する。削り取りツールが動いた軌跡に形状が存在すれば、その形状を連続的に削り取っていく。削り口には、新たに表面が生成される。また、3次元マウスの左ボタンを押すことで、ツールのサイズを変更し、削り取る形状のサイズを変えることができる。ツールのサイズは7段階用意されており、サイズは1段階増えると $\frac{3}{2}$ 倍大きくなる。図 19(a)は図 16に示す初期形状の一部を1番大きな7段階目の削り取りツールで削り取った様子である。図 19(b)はその削り口の一部を、2段階小さくした5段階目の削り取りツールでさらに削り取った様子である。

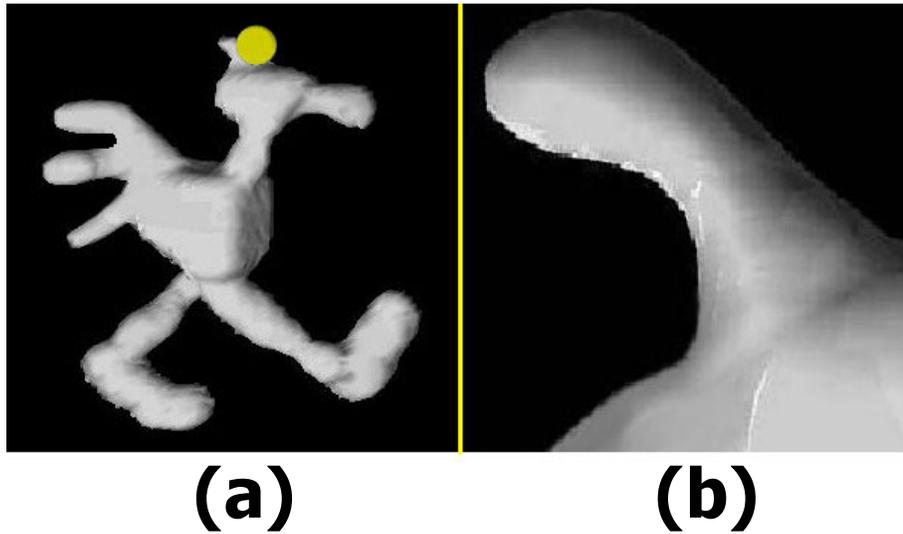


図 20 拡大表示機能

### (3) 拡大表示

まず、3次元マウスの右ボタンを押すことで、部分指定ツールを選択する。次に、細かい作業を行いたい部分を部分指定ツールで指定し、3次元マウスの左ボタンを押す。そうすると、その部分が取り出されて拡大表示される。図 20(b)は図 20(a)の鶏冠部分を拡大表示した状態である。拡大表示されても、表示が粗くなることはなく、ユーザは細かい形状を確認しながら、細かい変形を加えることができる。拡大の度合いは5段階用意しており、1段階増えると2倍拡大される。図 20(a)は最も拡大の度合いが小さい状態で、図 20(b)は図 20(a)の状態から2段階拡大した状態である。拡大の度合いは、マウスの左ボタンを押すことで変更できる。拡大の度合いに対して、ツールのサイズが大きすぎたり小さすぎたりすると作業が困難になる。このため、そのようなサイズのツールはユーザに提供されずに、拡大の度合いに適した作業しやすいサイズのツールだけが提供される。なお、拡大の度合いとツールサイズの組み合わせについては3.2.3項で詳しく述べる。

### 3. 没入型仮想彫刻モデルの実現手法

本章では提案モデルを実現するための手法として3次元形状の表現手法と3次元形状の拡大表示手法について述べる．まず，3.1節では3次元形状をどのようにして陰関数で表現するのかを述べ，3.2節で陰関数で表現した形状を拡大して表示する手法について述べる．

#### 3.1 形状の表現手法

本節では，3次元形状を陰関数を用いて表現する手法について述べる．まず，3.1.1項で形状の表現に用いる陰関数表現について述べた後，3.1.2項で付加する形状以外の形状データを増やさずに細かい変形を加えられるようにするための局所的に細密なデータ構造について述べる．

##### 3.1.1 陰関数による形状の表現

陰関数表現 [11] とは，陰関数形式で表現した関数を満たす3次元空間内の点の集合を3次元形状の表面とする表現手法であり，一般に式 (3) で定義される．

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

式 (3) において，陰関数  $f(P)$  に用いる関数によって様々な3次元形状の表現があり，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 (6) のような関数を用いた場合は，スケルタルサーフェス [14][15] と呼ばれる．

これらの陰関数表現では，3次元形状は表現式そのものが方程式であるので，交点や交線を容易に求めることができる．例えば，式 (4) で表した原点を中心とする半径  $r$  の球面  $f(P)$  と点  $p = (x_p, y_p, z_p)$  を通ってベクトル  $v = (x_v, y_v, z_v)$  に平行な直線  $p + tv$  があるとする．

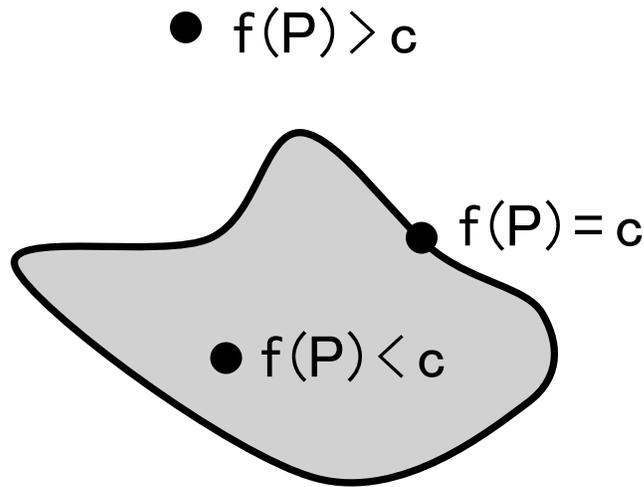


図 21 陰関数表現における形状の内外判定

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

これらの交点は，式 (5) で表される  $t$  についての 2 次方程式を解けば求められる．

$$(x_p + tx_v)^2 + (y_p + ty_v)^2 + (z_p + tz_v)^2 - r^2 = 0 \quad (5)$$

また，陰関数表現では， $f(P) = c$  となる点  $P$  の集合が表面上の点であり，表面をはさむ両側の点で関数  $f(P)$  を評価すると，一方は  $f(P) > c$  となり，他方は  $f(P) < c$  になる．簡単のため 2 次元で表した例を図 21 に示す．このため，交点や交線を求めるのに二分法 [16] などの収束計算を容易に利用することができる．この交点や交線を容易に求めることができるという特徴は，次節で述べる陰関数で表した 3 次元形状をポリゴンに変換する手法において有効である．

提案モデルでは，陰関数表現の中でもスケルトンと呼ばれる骨格要素を基にして形状を生成するスケルタルサーフェスを用いる．スケルタルサーフェスでは，形状はスケルトン  $S_i$  と距離場関数  $F_i$  によって決定され，式 (6) のような陰関数多項式で表現される．

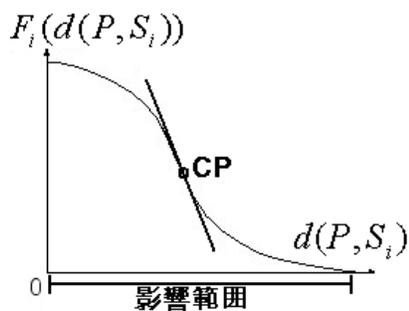


図 22 距離場関数の例

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

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

距離場関数  $F_i$  は典型的には図 22 に示すような形状となる関数が使われる。スケルトン<sup>①</sup>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。そのスケルトン

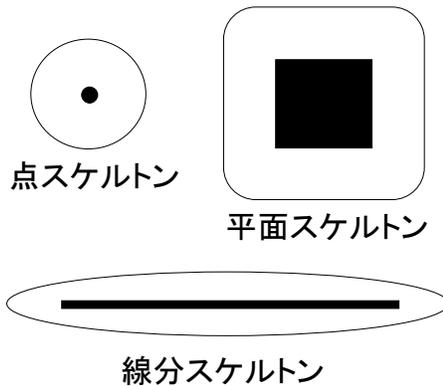


図 23 スケルトン

ケルトン  $S_i$  との距離である．すでに述べたように， $f(P) = c$  となる点  $P$  の集合が表面となる．

図 22 に示すような距離場関数の制御点 CP における傾きと影響範囲となる距離場関数の横軸のスケール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．形状と形状のつなぎ目の緩やかさの制御は制御点 CP における傾きの制御により行う．制御点 CP の傾きによる形状の変化を図 24 に示す．傾きが小さい場合はつなぎ目が緩やかになり，傾きが大きい場合はつなぎ目が急になる．形状の大きさの制御は影響範囲となる距離場関数の横軸のスケールの制御により行う．距離場関数の横軸のスケールが大きい場合は形状が大きくなり，距離場関数の横軸のスケールが小さい場合は形状が小さくなる．距離場関数の横軸のスケールによる形状の変化を図 25 に示す．

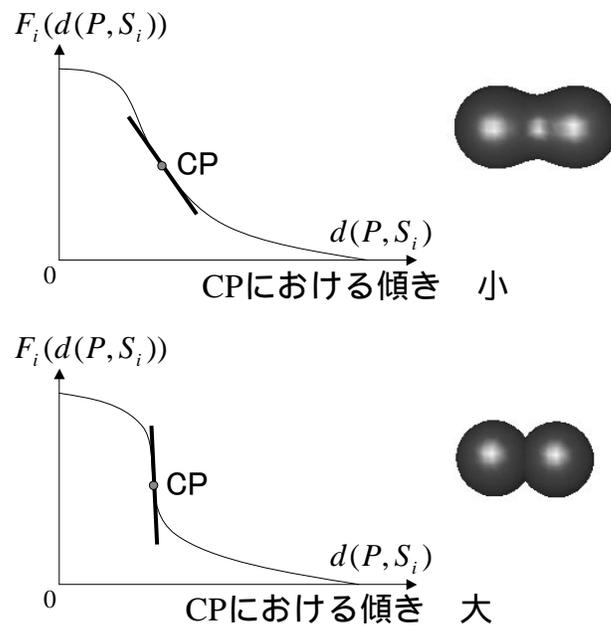


図 24 距離場関数の制御点 CP の傾きによる影響

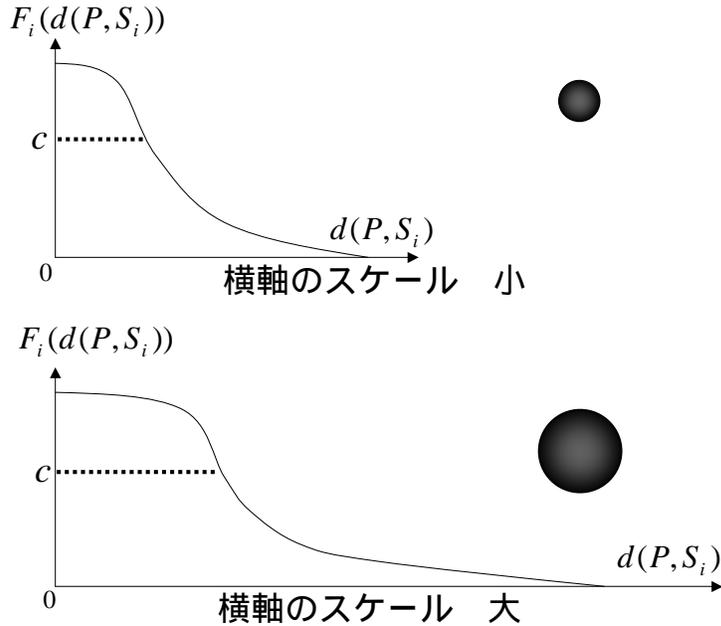


図 25 距離場関数の横軸のスケールによる影響

### 3.1.2 局所的に細密なデータ構造とその生成

提案モデルでは陰関数表現を利用して、形状のモデリングを行う。提案モデルでは、点スケルトンのみを用い、形状を表した式の和と差で、盛り付けと削り取りを表現する。例えば、式(7)で表される3つの点スケルトン( $S_1, S_2, S_3$ )からなる形状 $f_1(P)$ と式(8)で表される1つの点スケルトン( $S_4$ )からなる形状 $f_2(P)$ がある場合、形状 $f_1(P)$ への形状 $f_2(P)$ の盛り付けは、図26に示すように、 $f_1(P) + f_2(P)$ という加算によって表すことができ、変形後の形状 $f(P)$ は、式(9)で表現される。

$$f_1(P) = F_1(d(P, S_1)) + F_1(d(P, S_2)) + F_1(d(P, S_3)) \quad (7)$$

$$f_2(P) = F_2(d(P, S_4)) \quad (8)$$

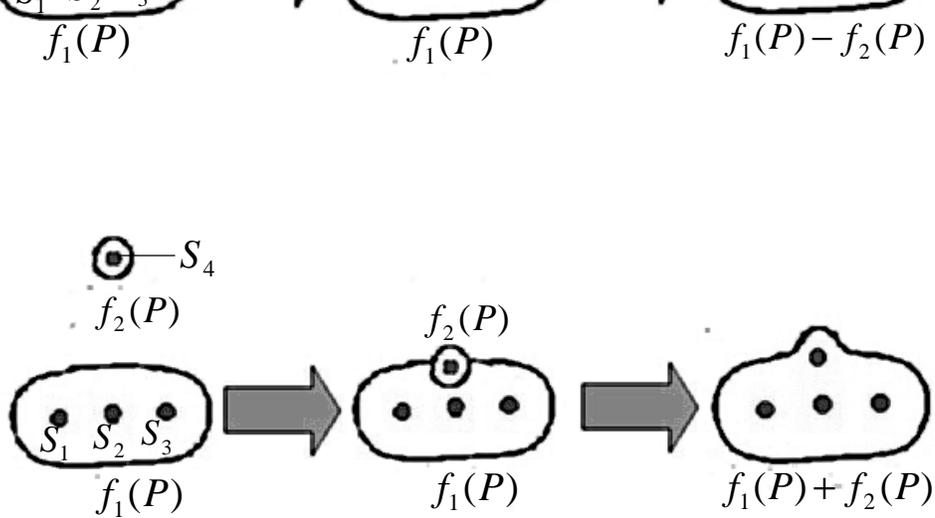


図 26 盛り付けの表現

$$\begin{aligned}
 f(P) &= f_1(P) + f_2(P) & (9) \\
 &= F_1(d(P, S_1)) + F_1(d(P, S_2)) \\
 &\quad + F_1(d(P, S_3)) + F_2(d(P, S_4))
 \end{aligned}$$

形状  $f_1(P)$  からの形状  $f_2(P)$  の削り取りは，図 27 に示すように， $f_1(P) - f_2(P)$

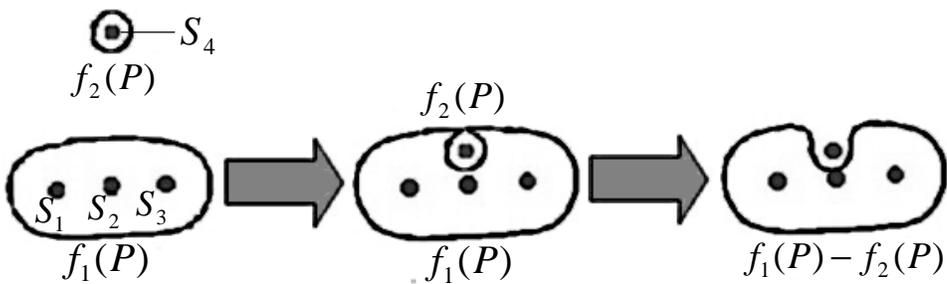


図 27 削り取りの表現

$$\begin{aligned}
 f_1(P) - f_2(P) &= f_1(P) - f_2(P) & (10) \\
 f_2(P) &= F_1(d(P, S_1)) + F_1(d(P, S_2)) \\
 &\quad + F_1(d(P, S_3)) - F_2(d(P, S_4))
 \end{aligned}$$

$f_1(P)$                        $f_1(P)$                        $f_1(P) - f_2(P)$

提案モデラでは、細かい変形を含む表現を行う場合は、図 28 に示すように、部

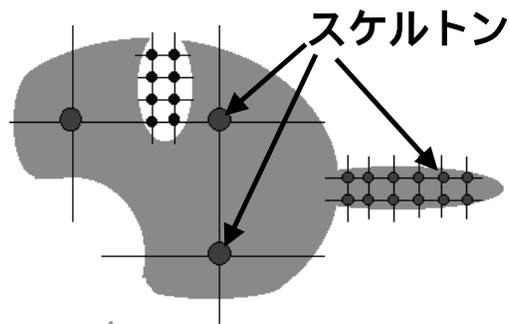
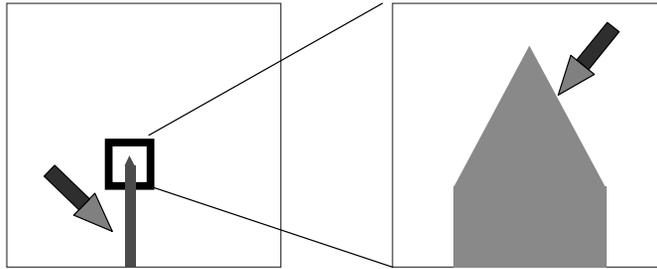


図 28 局所的に細密なデータ構造

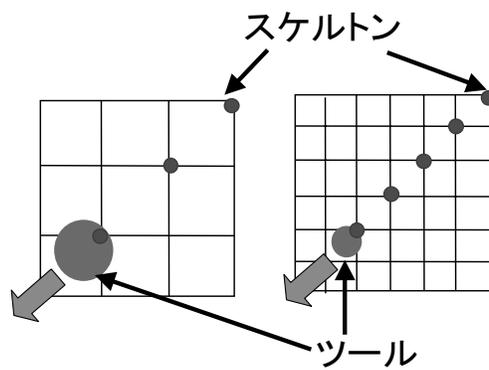


図 29 スケルトンの配置



幅大

幅小

7段階のツールサイズに合わせたスケルトン配置用格子の間隔は，ツールサイズが1段階増えると， $\frac{3}{2}$ 倍になる．その様子を簡単のため2次元で表したものを図30に示す．また，スケルトン配置用格子の間隔は対応するツールの直径の $\frac{2}{3}$ 倍である．提案モデルでは前項で述べた式(6)において， $c = 0.5$ と設定する．1段階から7段階までのツールサイズでは，式(11)で示すエルミート曲線[11]の横軸のスケールを，順に $\frac{1}{5}$ 倍， $\frac{3}{10}$ 倍， $\frac{1}{2}$ 倍，1倍， $\frac{21}{10}$ 倍， $\frac{26}{5}$ 倍， $\frac{66}{5}$ 倍したものを距離場関数として使用する． $v_i$ は位置 $p_i$ におけるベクトルであり，距離 $d$ に対する距離場関数の変化の様子とともに図31に示す．

$$F(d) = \begin{cases} p_0 H_0^3(d) + v_0 H_1^3(d) \\ \quad + v_1 H_2^3(d) + p_1 H_3^3(d) & (0 \leq d < 0.5) \\ p_1 H_0^3(d - 1.0) + v_1 H_1^3(d - 1.0) \\ \quad + v_2 H_2^3(d - 1.0) + p_2 H_3^3(d - 1.0) & (0.5 \leq d \leq 1.0) \end{cases} \quad (11)$$

$$\begin{aligned} H_0^3(t) &= (2t + 1)(1 - t)^2 \\ H_1^3(t) &= t(1 - t)^2 \\ H_2^3(t) &= t^2(t - 1) \\ H_3^3(t) &= t^2(3 - 2t) \end{aligned} \quad (12)$$

これらのスケールの変更の倍率については，経験的に盛り付ける形状や削り取る形状の大きさがツールサイズと同じになるように設定した．

以上のように，スケルトン配置用格子の細かさと距離場関数の変更により，付加するスケルトン以外の形状データを増やすことなく，細かい変形を加えることができる．

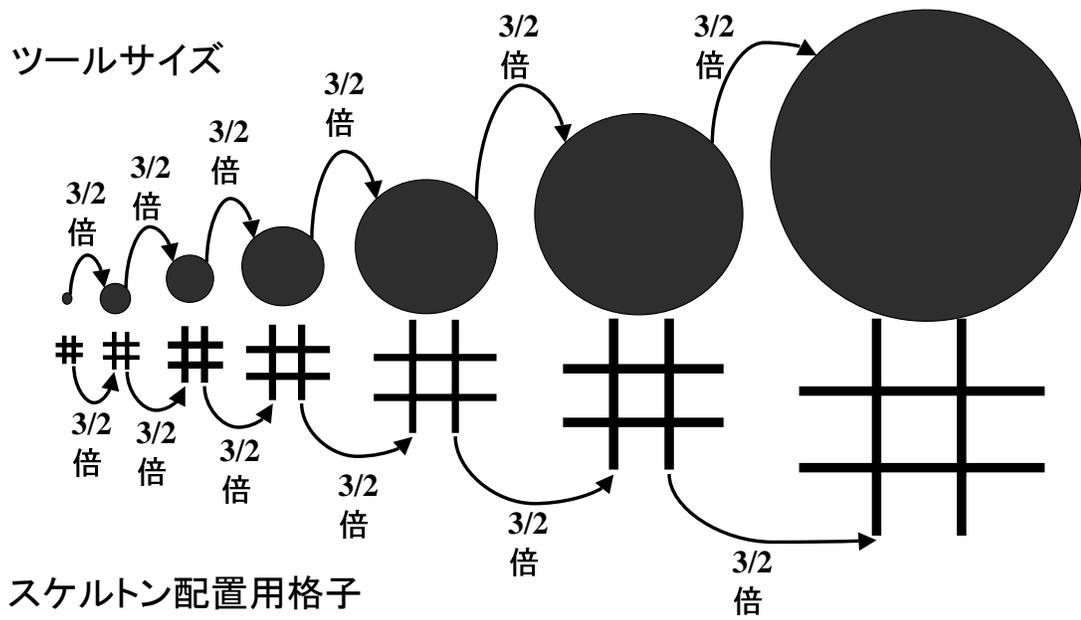
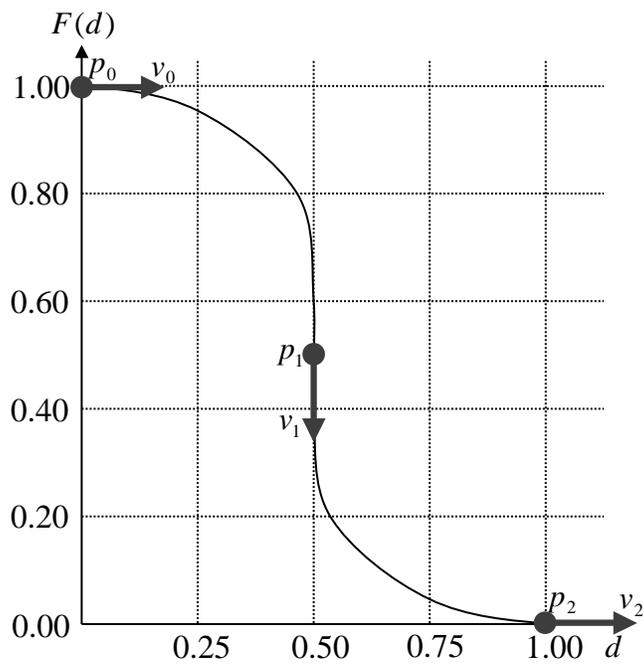


図 30 ツールサイズとスケルトン配置用格子



29

図 31 エルミート曲線による距離場関数

## 3.2 形状の拡大表示手法

本節では陰関数で表した3次元形状を拡大表示する手法について述べる。3.2.1項でスケルタルサーフェスを用いた3次元形状をユーザに実時間で提示する手法について述べる。次に、3.2.2項で描画の精度を変更し、描画を粗くすることなく、拡大の度合いを変更する手法について述べた後、3.2.3項で拡大の度合いとツールサイズの組み合わせについて述べる。

### 3.2.1 実時間処理のための高速描画

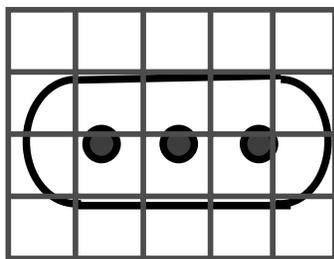


図 32 ポリゴン変換用格子

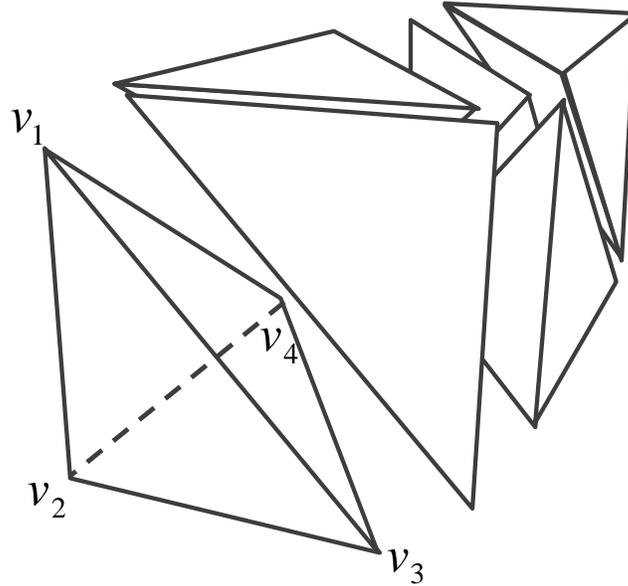


図 33 ポリゴン変換用格子の分割

ゴン変換用格子の頂点を，図 33 に示すように 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，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．そして，4 面体の各頂点座標  $v_i (i = 1 \sim 4)$  における陰関数の値  $f(v_i) (i = 1 \sim 4)$  を求める．求められた陰関数の値から，図 34 に示すように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．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 [16] 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し，収束させて根を求めるという手法である．そ

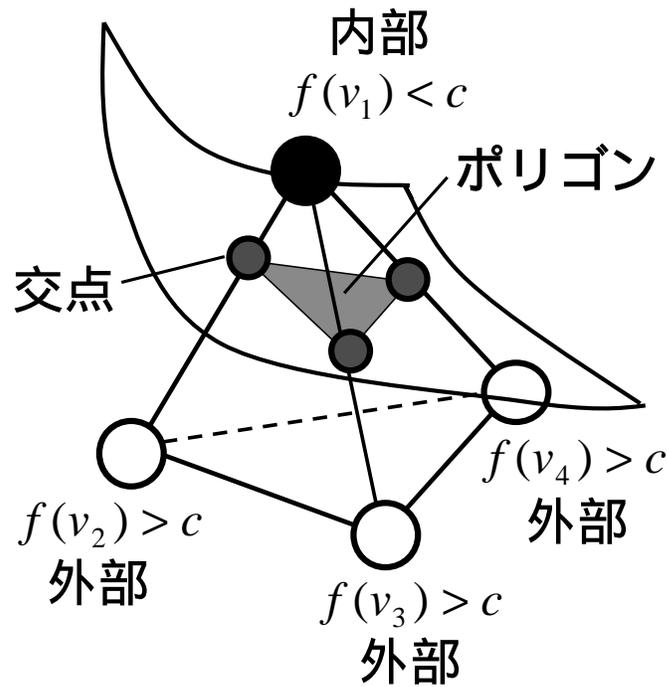


図 34 4 面体と形状との交点

して、求められた交点を基に、3 角形ポリゴンを生成し、ハードウェアを用いて高速に描画し、ユーザに提示する。

モデリング開始時や拡大の度合いを変更したときは、全てのポリゴン変換用格子に対して上述のポリゴン変換処理を行う。盛り付けや削り取りを行うときは高速化のために、変形する可能性のある範囲に限りポリゴン変換処理を行う。この範囲は経験的に、スケルトンが置かれた位置にあるポリゴン変換用格子を中心とする  $11 \times 11 \times 11$  個のポリゴン変換用格子とする。

### 3.2.2 描画精度の変更

生成するポリゴンのサイズを変更することにより、描画の精度を変える。生成するポリゴンのサイズの変更は、ポリゴン変換用格子のサイズを変えることにより行う。図 35 を例に、その手順を説明する。

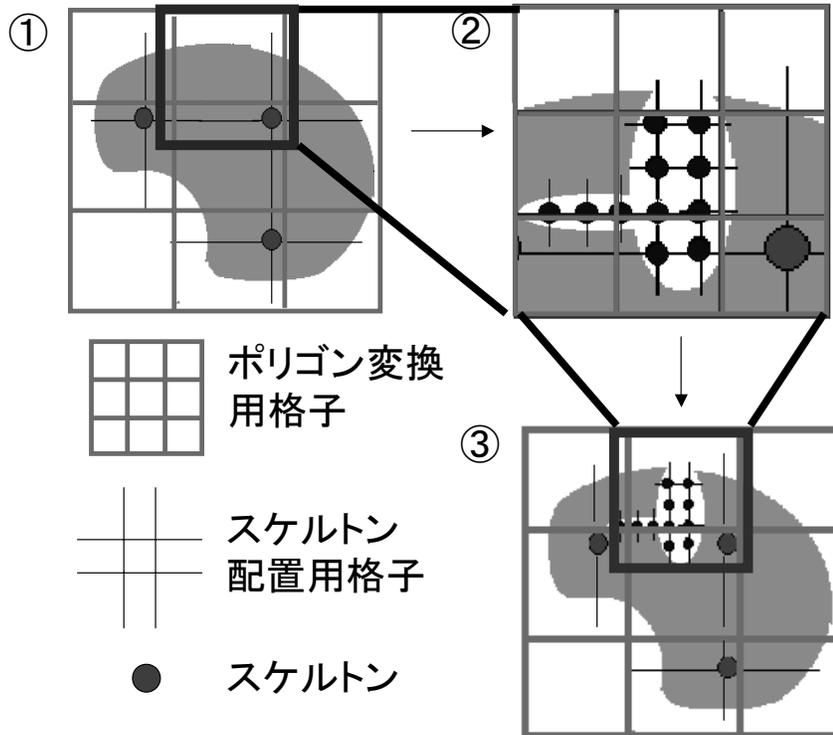


図 35 描画精度の変更

まず、取り出す領域に細かなポリゴン変換用格子を用意する( )。この領域は、拡大表示した時にユーザに提示される範囲のみを含むように設定している。そして、その領域に含まれる形状を小さなポリゴンに再変換して拡大する( )。作業

後に 3 次元形状の全体を表示するときは，再び粗いポリゴン変換用格子で全体をポリゴン変換する ( )．拡大の度合いをより大きくするときには，さらに小さなサイズのポリゴンに変換する．

5 段階の拡大の度合いに合わせたポリゴン変換用格子の間隔は，拡大の度合いが 1 段階増えると， $\frac{1}{2}$  倍になる．その様子を簡単のため 2 次元で表したものを図 36 に示す．一番大きなポリゴン変換用格子の間隔は，先に図 16 に示したツールの直径の  $\frac{1}{4}$  である．格子数はそれぞれ  $100 \times 100 \times 100$  である．以上のように，拡大の度合いに応じてポリゴン変換用格子のサイズを変更することにより，形状データを増やさずことなく，一部を取り出して描画を粗くせずに拡大する．

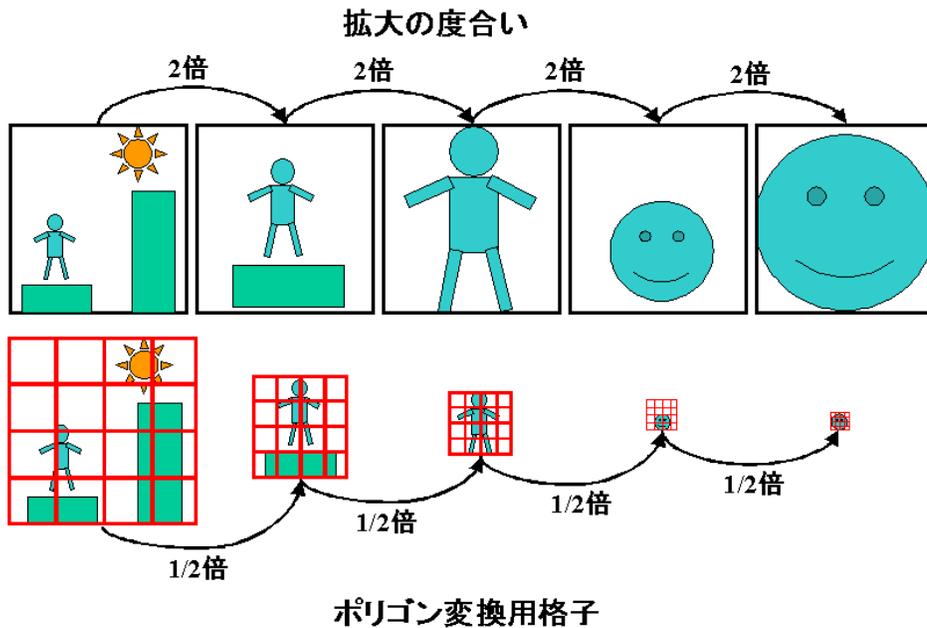


図 36 拡大の度合いとポリゴン変換用格子

### 3.2.3 拡大の度合いとツールサイズの組み合わせ

拡大の度合いが大きい場合に大きなサイズのツールで加えた変形を描画しようとする、ポリゴン変換処理を行う範囲を広くする必要があり、実時間性が損なわれる。また、拡大の度合いが小さいときに小さなツールを使うと、大量にスケルトンを置いてしまい、実時間性が損なわれる。そこで、拡大の度合いに対してツールサイズが大きすぎたり小さすぎたりすると作業が困難になるという理由に加えて、実時間性を保つために制限を設けるという理由で、図37のように、拡大の度合いごとに使えるツールサイズを限定している。1段階目の拡大の度合いでは7段階目、6段階目、5段階目以外のツールサイズは使用できず、2段階目の拡大の度合いでは6段階目、5段階目、4段階目以外のツールサイズは使用できない。このように、1つの段階の拡大の度合いで使えるツールサイズを3段階に限定している。

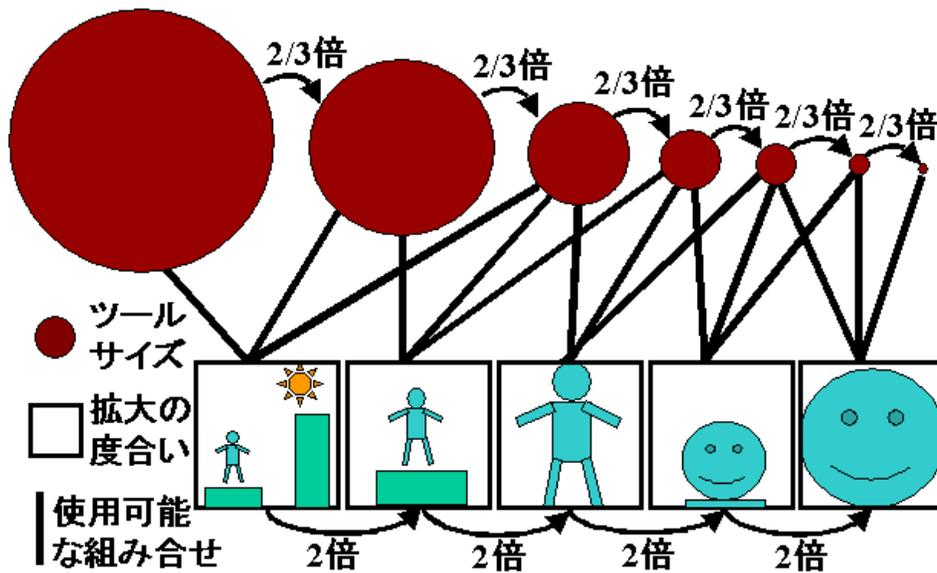


図 37 ツールサイズと拡大の度合い

## 4. モデリング例と考察

本章では提案モデラによるモデリング例を示し，モデリング例での経験を引用しながら提案モデラについての考察を行う．4.1 節でモデリング例の製作過程で行った作業の説明やモデリング例についての考察を述べる，4.2 節では提案モデラによる細かなモデリングや提案モデラの問題点について述べる．



図 38 モデリング例：雪だるま

### 4.1 モデリング例

提案モデラによるモデリングの例として，雪だるま，鳥，カニ，イルカの4つを製作した．まず，図 38 の雪だるまの製作過程を順に説明する．まず，図 39 に



図 39 顔と帽子のモデリング

示すように，先に図 16 に示した初期形状に一番大きな 7 段階目の盛り付けツールで顔を盛り付け，6 段階目の盛り付けツールで帽子を盛り付けた．続いて，5 段階目の盛り付けツールで腕を盛り付け，6 段階目の盛り付けツールで手を盛り付けた後，図 40 に示すように部分指定ツールで顔を指定し，図 41 に示すように顔を 2 段階目の度合いに拡大した．そして，4 段階目の盛り付けツールで鼻を盛り

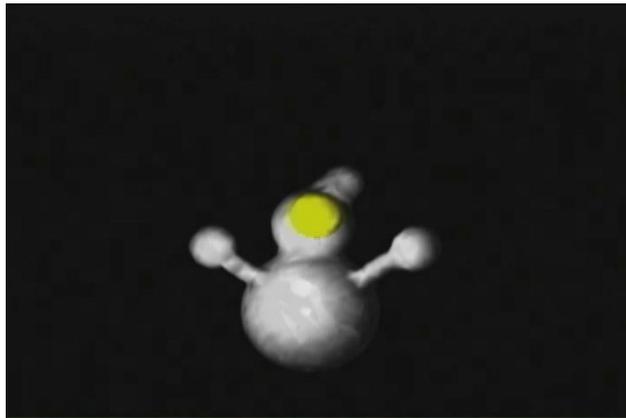


図 40 拡大部分 (顔) の指定



図 41 顔の拡大表示 (拡大の度合い：2段階)

付け，4段階目の削り取りツールで口と目を削り取った．図 42 は目を削り取っている様子である．次に，3段階目の盛り付けツールで眉を盛り付けた後，図 43 の四角で囲んだ帽子の一部を拡大し，図 44 のように，一番小さな 1段階目の削り取りツールで文字を彫った．全体表示の図 38 では帽子に彫った文字は確認できなくなるが，再び拡大表示すれば，確認することができる．顔のパーツなど細かい



図 42 口，鼻，目のモデリング

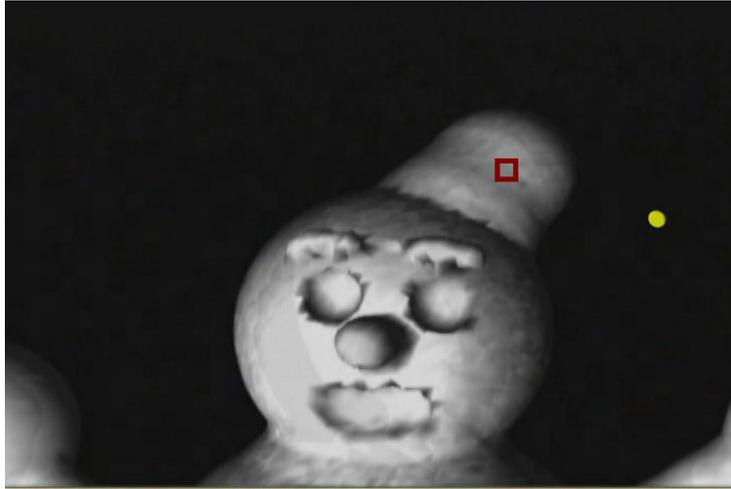


図 43 雪だるまの顔



図 44 図 43 の四角で囲んだ部分の拡大表示 (拡大の度合い: 5 段階)

形状を含むものの、ツールと同じ球状の部位が多く、比較的形状が単純であったため、特に手間取ることもなく、モデリングにかかった時間は6分程度であった。

次に、鳥、カニ、イルカについて、提案モデラの特徴である形状の一部を拡大表示して細かな変形を加える機能に焦点を絞って説明を行う。図45の鳥は、まず、拡大の度合いが最も小さい状態で、胴体、頭部、クチバシ、足の概形のモデリングを行った。次に、翼を取り出して2段階の度合いに拡大した状態で、図46のように5段階目の削り取りツールで模様をつけた。さらに図47のように、図46の四角で囲んだ部分を3段階の度合いに拡大した状態で、3段階目の削り取り

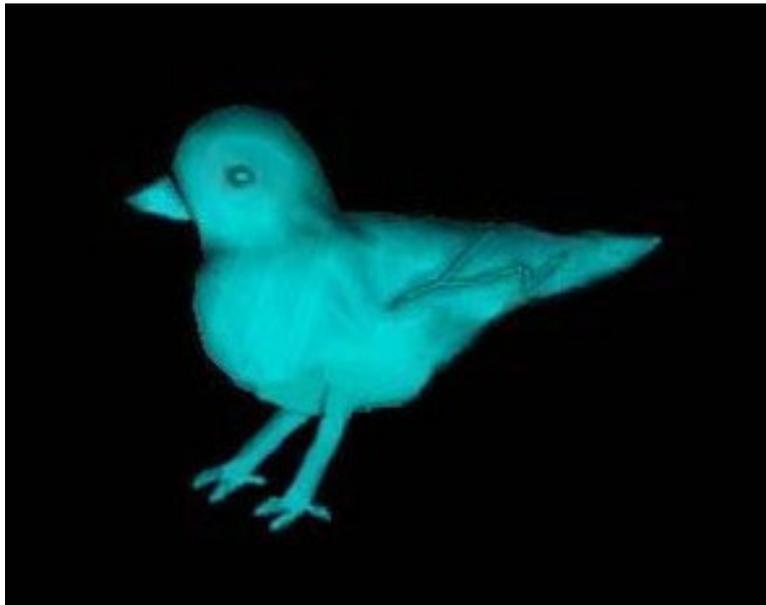


図 45 モデリング例：鳥

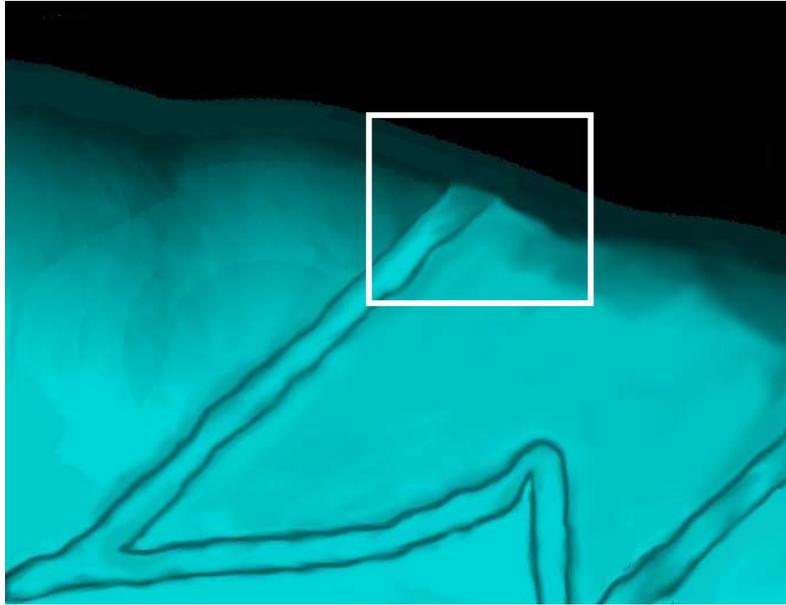


図 46 翼の拡大表示 (拡大の度合い: 2 段階)

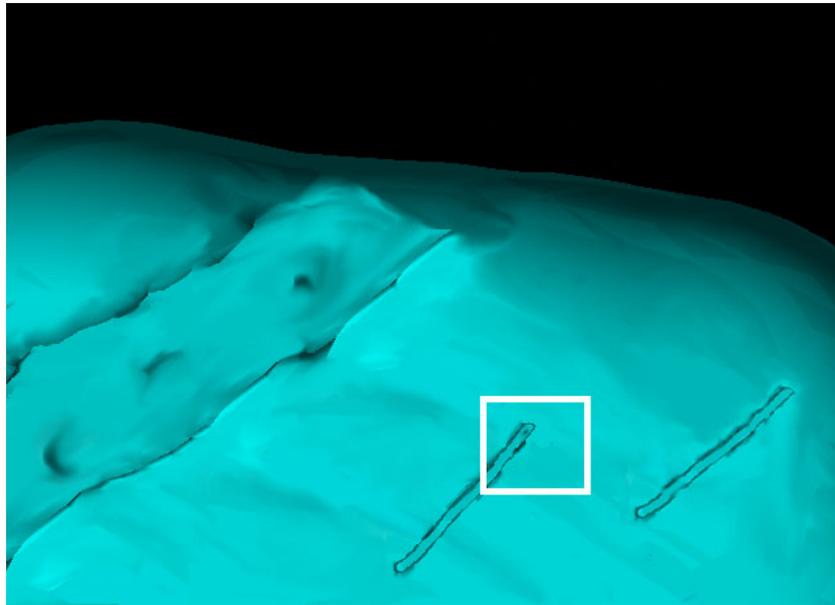


図 47 図 46 の四角で囲んだ部分の拡大表示 (拡大の度合い: 3 段階)



図 48 図 47 の四角で囲んだ部分の拡大表示 (拡大の度合い：5 段階)

ツールで細かい模様をつけた。同様に，図 48 に示すように，図 47 の四角で囲んだ部分を 5 段階の度合いに拡大して，1 段階目の削り取りツールでさらに細かい模様をつけた。全体表示の図 45 では細かい模様は分からないが，再び拡大表示すれば，細かい模様を確認することができる。模様を付けるときに長い距離を削り取る必要があり，削り取る深さを一定にするために慎重にモデリングを行ったが，それ以外は特に手間取ることもなく，モデリングにかかった時間は 10 分程度であった。

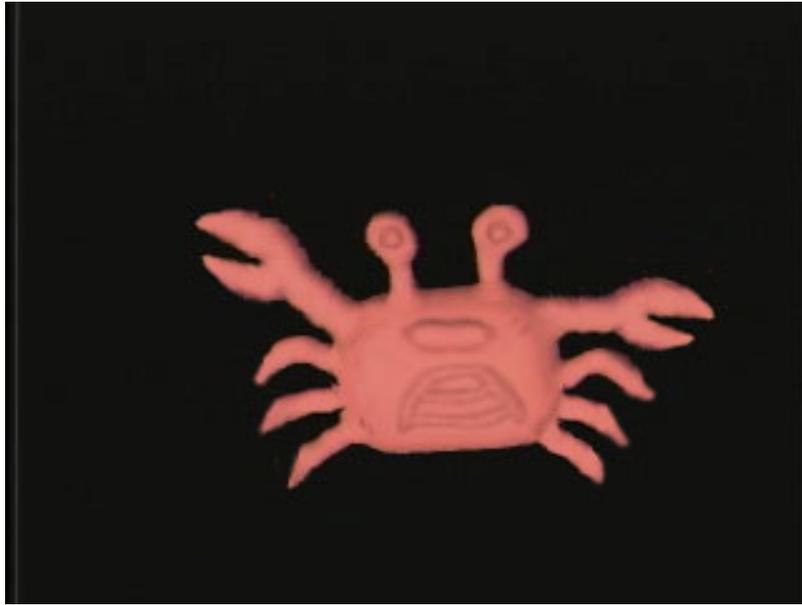


図 49 モデリング例：カニ

図 49 のカニは、まず、拡大の度合いが最も小さい状態で、胴体、突起、足の概形のモデリングを行った。次に、突起を 2 段階目の度合いに拡大した状態で、図 50 のように 5 段階目の削り取りツールで目を掘った。次に、図 51 のように足の先端を 4 段階目の度合いに拡大して 3 段階目の削り取りツールで先端を尖らせた。このように大きなツールで盛り付けた形状の一部を拡大して小さな削り取りツールで尖らせることができる。全ての足の先端を尖らせるのに数回拡大の度合いの変更をする必要があったが、モデリングにかかった時間は 10 分程度であった。

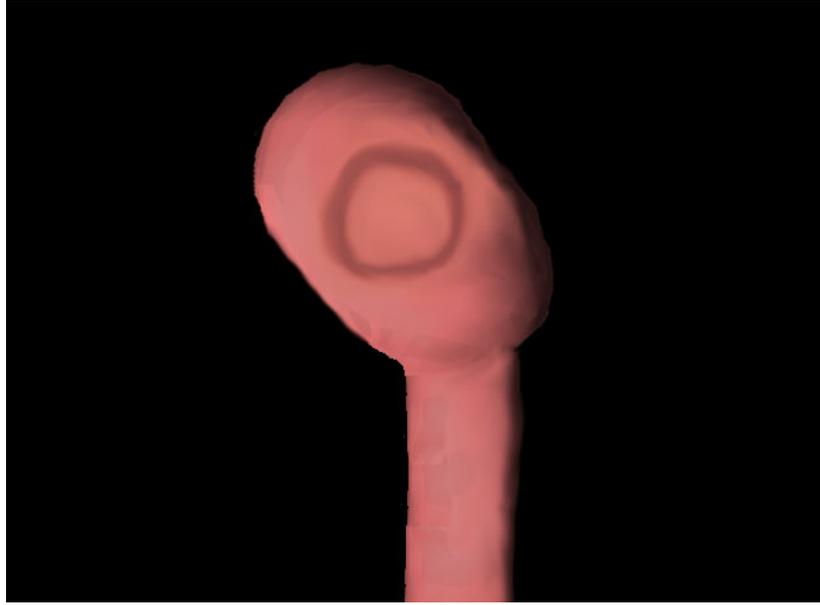


図 50 目の拡大表示 (拡大の度合い: 2 段階)

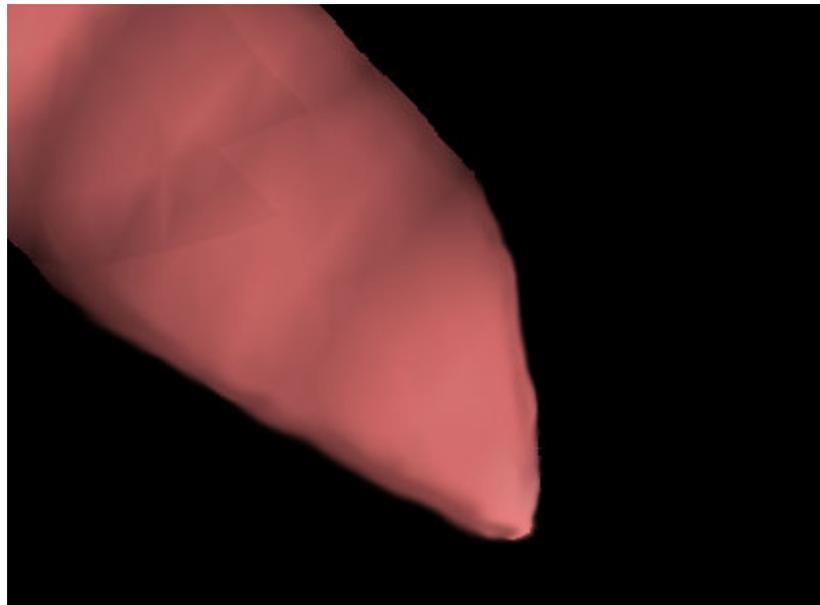


図 51 一番下の右足の拡大表示 (拡大の度合い: 4 段階)

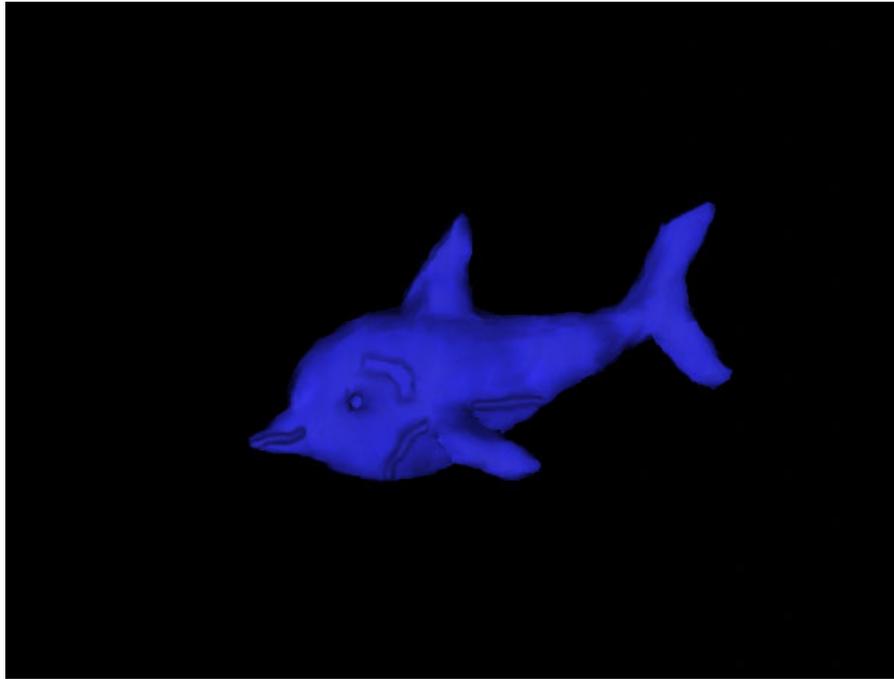


図 52 モデリング例：イルカ

図 52 のイルカは、まず、拡大の度合いが最も小さい状態で、胴体、クチバシ、ひれなどの概形のモデリングを行った。次にクチバシを 2 段階目の度合いに拡大して、5 段階目の削り取りツールで口を彫った。5 段階目の削り取りツールは 2 段階目の度合いに拡大しなくても使用できるが、微妙な曲がり具合の線を彫りたかったので、作業がしやすいように拡大した。その他にも、ひれを尖らせる作業などを行ったが、モデリングにかかった時間は 8 分程度であった。

以上のように、提案モデラによって、図 38 の雪だるま、図 45 の鳥、図 49 のカニ、図 52 のイルカを短時間でモデリングすることができた。盛り付けた部分にさらに細かく盛り付ける、形状の先端を尖らせる、微妙な曲がり具合の線を彫る

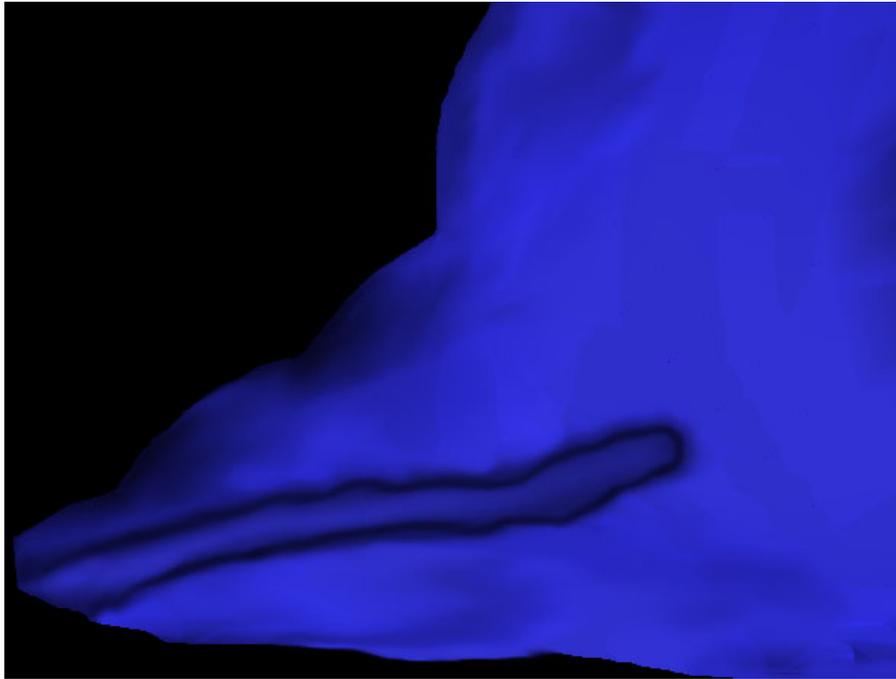


図 53 口の拡大表示 (拡大の度合い：2段階)

などの細かなモデリングを，拡大の度合いとツールのサイズの変更により容易に行うことができた．

なお，本実験では，CPU をポリゴン変換処理に 6 つ，その他の処理に 1 つ用いた．4.2 節で述べたように，変形を加えたときに処理を行うポリゴン変換用格子は  $11 \times 11 \times 11$  個に限定しているため，ポリゴン変換に関してはモデリング中に処理量が増加することはない．しかし，スケルトンの増加とともに陰関数の計算量は増加する．図 38 の雪だるま，図 45 の鳥，図 49 のカニ，図 52 のイルカでは 800 ~ 1000 個のスケルトンを配置しており，これらの状態の画像の更新レートは約 10fps であった．

## 4.2 考察

提案モデラによる細かい表現を含む形状のモデリングについてモデリング例での経験を引用しながら検証を行う。

提案モデラは、3次元形状に細かい変形を実環境の彫刻における動作と同じような動作で加えることができる。そのため、幾何や位相に関する数学的知識やCGに関する専門知識がなくても、容易に細かい表現を含む形状のモデリングを行うことができる。しかし、盛り付け時や削り取り時に反力がないためにツールと3次元形状との接触が分かりにくく、深く削りすぎたりしないように注意深く作業をする必要があった。これは、ハプティックインタフェース [21] を導入することにより解決することが可能であると考えられる。そのためには、反力計算や接触判定を行う方法について検討しなければならない。

形状の一部に細かい変形を加えるときは、その部分を拡大し見やすくした状態で、ツールを小さくして細かい変形を加える。拡大するときはポリゴンの数を増やすことにより、描画が粗くなるのを防いだ。拡大の度合いは7段階あるが、モデリング例を通して、全ての拡大の度合いにおいて描画が粗くならないことを確認することができた。また、拡大表示をした見やすい状態では、形状の先端を尖らせる、微妙に曲がった線を彫るなどの細かいモデリングが行いやすいことを確認した。

拡大表示は部分指定ツールで拡大する部分を指定して3次元マウスの左ボタンを押すことにより容易に行うことができる。ツールサイズの変更も3次元マウスの左ボタンを押すことで容易に行うことができる。しかし、拡大の度合いとツールサイズはどちらもボタンを押すごとに1段階ずつ変更するため、いくつかの段階を越えて変更したいときは不便さを感じた。拡大の度合いとツールサイズの段階をいくつでも越えて変更できるようなインタフェースを実現することにより不便さを解消すれば、作業効率を上げることができると考えられる。そのようなインタフェースの実現のためには、ボタンの数が多いデバイスやグラフィカル・ユーザ・インタフェース、例えばツールで押すことができる仮想空間に設置したCGのボタンなどの導入について検討する必要がある。

提案モデラでは、一度削り取った部分に盛り付けることができない。これは、スケルトン配置用格子の頂点にスケルトンを置くことで削り取りを実現していることが原因である。前節の4つのモデリング例には、削り取りを行った部分に盛り付けを行う必要がある形状が含まれなかったために不便さを感じることはなかった。提案モデラでは、4つのモデリング例以外にも様々な形状のモデリングを行うことができるが、削り取りを行った部分に盛り付けを行うことができれば、より多くのパターンの形状を表現することができると考えられる。よって、削り取りを行った部分に盛り付けを行えるようにするために、すでに置かれたスケルトンの消去や更新を行う方法を試みた。例えば、4章で述べた手法では、削り取りは図54に示すように、盛り付けによりスケルトンが置かれている頂点に減算用のスケルトンを重ねて置くことで実現し、それ以上のスケルトンを置くことはできなかった。しかし、図55に示すように、スケルトンを消去することで削り取りを表現すれば、何度でもスケルトンを置くことができるので、削り取った部分に盛り付けることが可能である。また、この手法は、4章で述べた手法に比べてスケルトンの増加を減らすことができる。しかし、これは同じ頂点にスケルトンが置かれることが前提であるため、同じサイズの盛り付けツールと削り取りツールを使用した場合にしか適用できない。よって、盛り付けツールのサイズと削り取りツールのサイズが異なる場合にも、スケルトンの消去や更新によって削り取りを行う方法を検討した。大きなツールで盛り付けた形状  $f_1(P)$  に対して、小

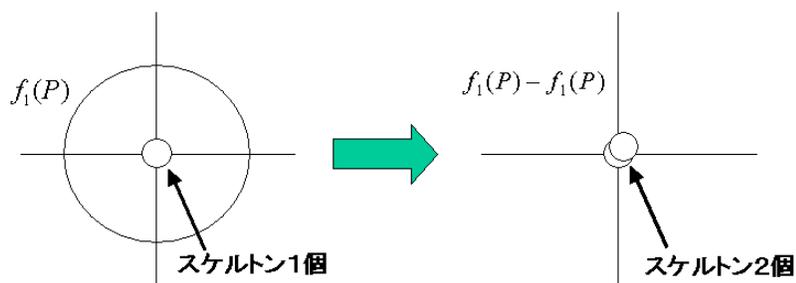


図 54 スケルトンを置いた場合

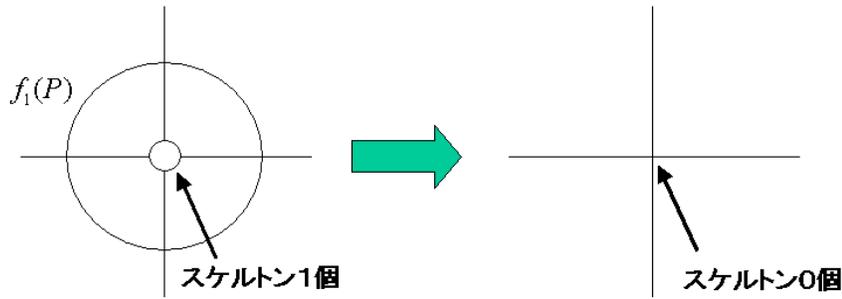


図 55 スケルトンを消去した場合

小さなツールで形状  $f_2(P)$  を削り取る場合は，図 56 に示すように  $f_1(P) - f_2(P)$  という演算を行う．小さなツールで盛り付けた形状  $f_2(P)$  に対して，大きなツールで形状  $f_1(P)$  の削り取りを行う場合は，図 56 に示すように盛り付けによるスケルトンを消去してから減算用のスケルトンを置く．大きなツールで削り取った部分 -  $f_1(P)$  に対して，小さなツールで形状  $f_2(P)$  を盛り付ける場合は，図 57 に示すように  $-f_1(P) + f_2(P)$  という演算を行う．小さなツールで削り取った部分  $f_2(P)$  に対して，大きなツールで形状  $f_1(P)$  の盛り付けを行う場合は，図 57 に示すように削り取りによるスケルトンを消去してから増算用のスケルトンを置く．この方法により，異なるサイズのツールを使用した場合でも，削り取った部分に盛り付けることができる．しかし，この方法では，盛り付けツールのサイズを変更しなくても，場所によって盛り付けられる形状の大きさが変わってしまう場合がある．つまり，以前に削り取りが行われていない部分に盛り付けた形状  $f_2(P)$  と図 57 に示すように以前に削りとりが行われた部分に盛り付けた形状  $-f_1(P) + f_2(P)$  では大きさが異なる．よって，両方の場合で生成される形状の大きさを同一にする必要があり，それぞれの場合で距離場関数を変更する方法などを検討する必要がある．

以上のように，提案モデラは細かい表現を含む 3 次元形状のモデリングを簡単な操作で行うことができるが，いくつか改善すべき点が残されている．これらの点を改善することにより，操作性や表現力をさらに高めることができると考えられる．

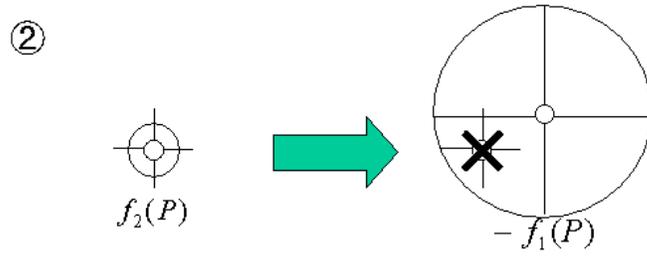
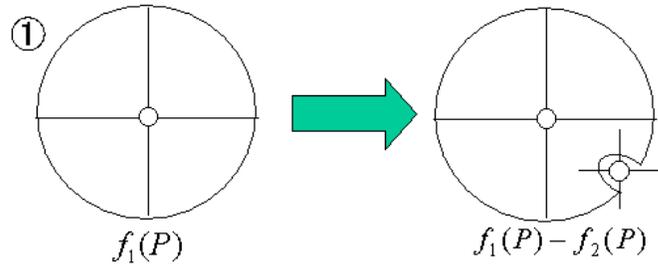


図 56 異なるツールサイズの削り取り

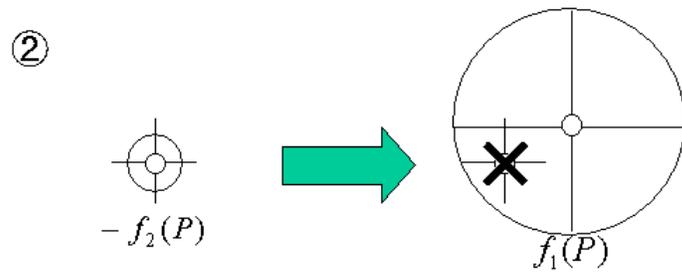
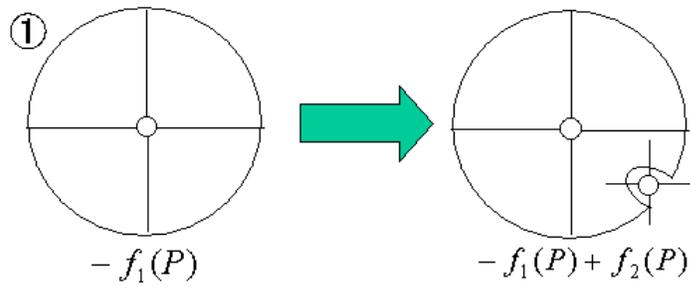


図 57 異なるツールサイズの盛り付け

## 5. むすび

本論文で提案したモデラでは，HMD を用いた立体視によって仮想空間に没入し，簡単な操作方法による盛り付けや削り取りといった直感的な動作で 3 次元形状のモデリングを行うことができる．盛り付けは，盛り付けツールが動いた軌跡に形状を生成することにより行う．削り取りは，削り取りツールが動いた軌跡に存在する形状を切削することによって行う．このように，提案モデラは，1) 幾何や位相に関する数学的な知識や CG に関する専門的な知識がなくても使用できる，2) 実環境における動作と同じ動作で 3 次元形状の変形操作を容易に行える，3) 3 次元形状がどのように変形するかを容易に予想することができる，4) 3 次元形状を立体的に見ることにより 3 次元形状の視認を容易に行える，というモデリングを容易にするための 4 つの要件を満たしている．

従来の仮想彫刻モデラ [7, 8, 9, 10] で用いられているボクセル表現やパラメトリック表現では，描画を粗くせず拡大表示を行うことやあらかじめ用意されたデータの解像度以下の変形を加えることを可能にしようとする，処理の増加や形状データの増加を招いてしまう．そこで，提案モデラでは，従来の仮想彫刻モデラ [7, 8, 9, 10] で用いているボクセル表現やパラメトリック表現ではなく，陰関数表現を用いた．3 次元形状を局所的に細密なデータ構造を持つ陰関数で表現することにより，付加する形状データのみを細密にすることを可能にした．また，陰関数で表した形状データをポリゴンに変換してレンダリングを行うが，拡大の度合いに応じて適切な数のポリゴンに変換することにより，拡大時に描画が粗くなるのを防いだ．これらの手法を用いて開発した提案モデラは，描画を粗くせずに 3 次元形状の部分的な拡大表示を行うことが可能で，これにより細かいモデリングを容易に行うことができる．また，加えられる変形の細かさに関しても，従来の仮想彫刻モデラ [7, 8, 9, 10] のように，あらかじめ用意されたデータの解像度以上の変形しか加えられないという制限はない．

以上のように，提案モデラは没入型仮想環境において細かい表現を含む 3 次元形状のモデリングを容易に行うことができる仮想彫刻モデラである．本論文においては，提案モデラを用いて実際にモデリングを行ったモデリング例を示し，細かい表現を含む 3 次元形状を短時間でモデリングできることを確認した．しかし，

提案モデラは盛り付けや削り取りといった直感的な作業方法でモデリングを行うため、精密さや正確さに欠ける。よって、モデリング例に示したような精密さや正確さを必要としない3次元形状のモデリングには適しているが、工業製品のような精密さや正確さを必要とする3次元形状のモデリングには適していない。

本県球では陰関数表現をポリゴン表現に変換する手法を用いたため、レイトレーシング法の実装を行わなかったが、提案モデラでモデリングを行った3次元形状にレイトレーシング法を適用すれば、3次元形状の高精彩な画像の取得ができる。レイトレーシング法を用いた高精細な画像では、提案モデラでモデリングをした細かい形状までも確認できる。

## 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，大阪大学 サイバーメディアセンター 情報メディア教育研究部門 竹村 治雄 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 山澤 一誠 助手に深く感謝致します．

本研究の全過程において，終始暖かい御助言，御指導を頂き，大学院生活の指針となって頂いたソフトウェア基礎講座 松宮雅俊氏に心より感謝致します．

物心両面において常に温かい御支援を頂くと共に，親身に立った御助言を頂いた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 元事務補佐員 福永 博美 女史，事務補佐員 北川 知代 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] Shade R5 チュートリアル, ExpressionTools, 2001.
- [3] LightWave 3D version 5.5 日本語ユーザーガイド, NewTek Incorporated, 1997.
- [4] form・Z RenderZone 徹底解説, 株式会社建設知識, 1997.
- [5] Ray Dream Studio 5 ユーザガイド, MetaCreations Corporation, 1997.
- [6] 3ds max リファレンス 日本語版, Autodesk, 2001.
- [7] T.A. Galyean and J.F. Hughes: "Sculpting: An Interactive Volumetric Modeling Technique," *Proc. ACM SIGGRAPH'91*, pp.264-274, 1991.
- [8] Introduction to FreeForm Modeling, SensAble Technologies, <http://www.sensable.com/>
- [9] A. Raviv and G. Elber: "Three Dimensional Freeform Sculpting via Zero Sets of Scalar Trivariate Functions," *Proc. 5th Sympo. on Solid Modeling and Applications*, pp.246-257, 1999.
- [10] R.A. Noble and G.J. Clapworthy: "Sculpting and Animating in a Desktop VR Environment," *Proc. IEEE Computer Graphics International*, pp.187-195, 1998.
- [11] 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.
- [12] 海部陸, 松宮雅俊, 横矢直和: "局所的に細密なデータ構造を持つ陰関数表現を用いた没入型モデラ," 日本 VR 学会第 6 回大会論文集, pp.469-470, 2001.
- [13] 海部陸, 松宮雅俊, 横矢直和: "局所的に細密な陰関数表現による拡大作業機能を持つ没入型モデラ," 電子情報通信学会技術研究報告, Vol.101, No.624, pp.13-18, 2002.

- [14] B. Wyvill: “ Animation and Special Effects, ” *Introduction to Implicit Surfaces*, ed. J. Bloomenthal, Morgan Kaufmann Publishers, pp.242-269, 1997.
- [15] 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.
- [16] W.H. Press, S.A. Teukolsky, W.T. Vetterling and B.P. Flannery, 丹慶勝市, 奥村晴彦, 佐藤俊郎, 小林誠 (訳): NUMERICAL RECIPES in C 日本語版, 技術評論社, 1993.
- [17] J.C. Hart: “ A Geometric Method for Antialiased Ray Tracing of Implicit Surfaces, ” *The Visual Computer*, Vol.12, No.10, pp.527-545, 1996.
- [18] G. Wyvill: “ Ray Tracing Implicit Surfaces, ” *Introduction to Implicit Surfaces*, pp.166-195, Morgan Kaufmann, 1997.
- [19] J. Bloomenthal: “ An Implicit Surface Polygonizer, ” in P. Heckbert(ed.), *Graphics Gems IV*, Academic Press, pp.324-349, 1994.
- [20] 松宮雅俊, 竹村治雄, 横矢直和: “ 自由形状モデリングのための陰関数曲面を用いた仮想粘土細工システム, ” *情処学論*, Vol.42, No.5, pp.1151-1160, 2001.
- [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.