

NAIST-IS-MT9851132

修士論文

2次元平面と3次元空間の組み合わせによる
モデリング環境

吉森 勇人

2000年2月20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。

吉森 勇人

審査委員： 横矢 直和 教授
千原 國宏 教授
竹村 治雄 助教授

2次元平面と3次元空間の組み合わせによる モデリング環境*

吉森 勇人

内容梗概

3次元形状のモデリングに用いられている3次元CGソフトウェアは2次元画面上で作業を行うのが一般的であるが、近年、人工現実感の分野では3次元空間でのモデリング作業を前提とした没入型モデラの開発が盛んに行われている。これら2つのモデリング環境はその作業空間の次元数に起因する利点と欠点をそれぞれに持っている。本研究では2次元平面と3次元空間でのモデリングを単一の作業空間で行うことが可能で、双方の利点を兼ね備えたモデリング環境を提案する。

本研究では表示装置として傾斜型ディスプレイを用いて提案環境を単一の作業空間として実現する。2次元平面でのモデリングには傾斜型ディスプレイ表面でのライトペンからの入力を、3次元空間でのモデリングには液晶シャッタ眼鏡による時分割両眼立体視と3次元位置・姿勢センサからの入力を用いる。ユーザはこの2種類の入力装置を組み合わせたデバイスを使いモデリングを行う。モデリング操作は各操作の持つ性質に応じてそれぞれの空間で行うことができる。例えば、ブール演算のような空間的な位置関係の把握、高い自由度の入力が必要とされる操作は3次元空間で行い、平面形状の作成のような2自由度しか必要としない操作は2次元平面で行う。また、掃引のような双方の性質を併せ持つ操作は両空間を同時に利用しながら行う。このように提案環境は、立体形状の視認が容易で、直感的で対話性が高いモデリングが行なえるという3次元空間の持つ利点と、入力の制御が容易で、安定した操作が行なえるという2次元平面の持つ利点を併せ持

*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-MT9851132, 2000年2月20日.

つ. 本論文では提案環境を実際に試作したシステムについて詳しく述べた後, その有効性をモデリング過程とモデリング結果を通して検証する.

キーワード

仮想現実感, 3次元形状モデリング, 没入型モデル, CAD

Combination of Two- and Three-Dimensional Space for Solid Modeling*

Hayato Yoshimori

Abstract

This paper proposes a hybrid 3-D modeling system that inherits advantages of both modeling on 2-D plane and modeling in 3-D space. To integrate 2-D and 3-D modeling environments into a unified modeling space, a slant rear-projection display is employed as a drafting table that gives users 2-D constraints and provides a 3-D work space by using LCD shutter glasses. To interact with objects in the unified modeling space, an input device which combines a lightpen for 2-D plane and a 3-D tracker for 3-D space is introduced.

In the proposed system, users can perform a number of modeling operations both on 2-D plane and in 3-D space according to the nature of each operation. The boolean operation, for example, is implemented to perform in 3-D space, because it becomes easier by giving users ability to understand the relationship between objects and to move objects with 6 degrees of freedom.

As a result, each operation in the proposed system becomes more powerful than those of conventional 3-D-CG softwares, so users can create complex models easily and efficiently.

Keywords:

virtual reality, 3-D modeling, immersive modeling system, computer aided design

*Master's Thesis, Department of Information Systems, Graduate School of Information Science, Nara Institute of Science and Technology, NAIST-IS-MT9851132, February 20, 2000.

目次

1. はじめに	1
2. 既存のモデリング環境と本研究の方針	3
2.1 3次元CGソフトの2次元平面でのモデリング	3
2.1.1 2次元平面でのモデリングの概要	3
2.1.2 2次元平面でのモデリングの利点と欠点	5
2.2 没入型モデラの3次元空間でのモデリング	8
2.2.1 3次元空間でのモデリングの概要	8
2.2.2 3次元空間でのモデリングの利点と欠点	10
2.3 3次元空間への制約の導入による操作性の向上	11
2.4 本研究の方針	13
2.4.1 2次元平面でのモデリングの利点を持つ2次元平面の実現	14
2.4.2 2次元平面と3次元空間の利点を活かしたモデリング	15
3. 2次元平面と3次元空間を組み合わせたモデリング環境の開発	17
3.1 プロトタイプシステムの概要	17
3.1.1 実行環境	18
3.1.2 ペン型操作デバイスの詳細	18
3.2 作業空間の構成要素	20
3.3 基本操作	25
3.4 オブジェクトの作成	27
3.4.1 平面形状の作成	28
3.4.2 形状指定による立体形状の作成	31
3.4.3 押し出しによる立体形状の作成	33
3.5 オブジェクトの変形	37
3.5.1 交差面の押し出しによる変形	37
3.5.2 頂点の編集による変形	38
3.5.3 仮想的な磁力による変形	41

3.5.4	ブール演算による変形	42
3.6	モデリング例	44
4.	検証と考察	53
5.	おわりに	57
	謝辞	59
	参考文献	60

図目次

1	平行投影図と直交投影図	7
2	既存のモデリング環境では作成しにくい立体形状の例	14
3	システム構成図	17
4	試作システムでのモデリングの様子	19
5	ペン型デバイスとカーソル	19
6	作業空間の構成要素	21
7	2次元平面の表示・非表示	22
8	2次元平面とオブジェクトの交差面	25
9	“Create” ボタンと構成要素	27
10	平面形状の作成	30
11	形状指定による立体形状の作成	31
12	“FixSec” による立体オブジェクト作成	34
13	頂点の追加による面の3角分割	35
14	“FixSec” の連続使用による立体形状の作成	36
15	交差面の押し出しによる立体オブジェクトの変形	39
16	2次元平面での頂点編集による立体オブジェクトの変形	40
17	磁力による平面オブジェクトの変形	43
18	“Boolean” ボタンと構成要素	44
19	ブール演算による立体オブジェクトの変形	45
20	りんごのモデリング過程	47
21	りんごのレンダリング結果	48
22	スタンドライトのモデリング過程	50
23	スタンドライトのレンダリング結果	51

表目次

1	ボタンの分類と各ボタンの機能	24
2	平面形状の作成方法とボタン名	28

3	立体形状の作成方法とボタン名	32
4	3つのブール演算とボタン名	44

1. はじめに

今日, 3次元CG(Computer Graphics)は, 映画やテレビ, 意匠設計, プレゼンテーションから家庭用ビデオゲームソフトウェアに至るまでのあらゆる映像表現分野に普及している. 当初は単なる直方体や球といった単純な立体形状しか表現できなかった3次元CGも, 計算機が高性能化した現在では, 恐竜や宇宙船といった複雑な立体形状を表現できるようになり, そのための複雑な3次元形状データを効率よく作成したいという欲求が高まっている.

一般にこれらの3次元形状データの作成(モデリング)に用いられる3次元CGソフトウェアでは, CRTモニタ等の2次元画面上で, マウス等の2自由度の入力装置を用いてモデリングを行う[1][2][3][4]. このような2次元平面でのモデリングでは, ユーザが同時に制御する自由度が1もしくは2と少ないため, ユーザは意図した通りに入力を行なうことができ, また, 机の上で操作するため安定した正確な操作が行える. しかし, この限定された自由度のため本質的に3次元の操作でも2次元の操作の組合せに分解して行う必要があり, 操作の直感性やモデリングの効率を低下させている[5]. また, 2次元画面での表示には奥行き情報が不足しており, 3次元形状の確認や空間的な位置関係の把握が困難であるという欠点もある[6].

一方, 2次元平面でのモデリングの持つこれらの欠点を排除するために, HMD(Head Mounted Display)や3次元マウス等の, 人工現実感(AR: Artificial Reality)あるいは仮想現実感(VR: Virtual Reality)の技術を駆使して, 仮想空間内でモデリングを行なう没入型モデラと呼ばれるモデリング環境が注目され, 様々なシステムが開発されている[5][6][7][8][9][10][11][12].

没入型モデラにおける3次元空間でのモデリングには, 両眼立体視により3次元形状を直接3次元的に確認できるという視認の容易性や, 多自由度のデバイスを用いることで3次元形状を直接3次元的に操作できるという操作の直感性などの利点がある[7]. その反面, 一般に人間には複数の自由度を同時に制御することは困難であり, また, 人間の腕の持つ空間指示能力が低いため, 正確な操作が困難であるという問題が指摘されている[8][13].

これを解決するために, 人間の物体操作に重要とされる力覚や触覚を仮想空間

に再現することで、ユーザの手の動きの自由度を制限する手法 [14][15][16] や、離散的配置制約や干渉回避等の、仮想物体の動きをソフトウェア的に制限する手法 [6][7][8][10] が提案され、3次元空間でのモデリングに導入されている。しかし、これらの制約を導入しても、3次元空間でのモデリングでは複雑な立体形状を作ることはできず、現状として、その利用範囲は、建築物の設計プロセスの初期段階 [10][12] や、大まかな形状を素早く作ることによるユーザの発想支援 [7][8] といった、単純な立体形状作成能力しか必要としないものに限定されている。

本研究の目的は、没入型モデラでは作れないような複雑さを持つ任意の立体形状を、3次元CGソフトウェアで作るよりも素早く容易に作成することが可能なモデリング環境を開発することである。そのために、2次元平面でのモデリングと3次元空間でのモデリングの持つそれぞれの利点に注目し、両者を単一の作業空間として組み合わせたモデリング環境を提案する [17][18]。

提案環境では、3次元空間に物理的な制約を持つ2次元平面を導入することで、それぞれのモデリングの持つ利点を単一の作業空間に実現する。また、両空間で共通して使用できる入力装置を使って、モデリング操作をその性質に適した空間で行えるようにすることで、提案環境を有効に利用できるようにする。

以下、2章で2次元平面でのモデリングと3次元空間でのモデリングの概要とその利点と欠点、および3次元空間でのモデリングに導入された制約について述べた後、本研究の提案を行なう。また3章では、2章での提案に従って試作したシステムの詳細や実装した諸機能について述べる。さらに4章では、試作システムを通して提案環境を考察し、最後にまとめと今後の課題について述べる。

2. 既存のモデリング環境と本研究の方針

前章で述べたように 3 次元形状データを作成するモデリング環境は、その作業空間の次元数によって 2 次元平面でのモデリングと 3 次元空間でのモデリングに大別することができる。本章ではこの分類に従って、まず 2.1 節において、2 次元平面でのモデリングの概要とその利点と欠点について述べる。次に 2.2 節において、2 次元平面でのモデリングの欠点を克服するべく研究されてきた、没入型モデルにおける 3 次元空間でのモデリングについて述べる。また、3.3 節では 3 次元空間でのモデリングの操作性を向上させるための制約について関連研究を示す。最後に 3.4 節において、本研究の方針について述べる。

2.1 3 次元 CG ソフトの 2 次元平面でのモデリング

2 次元平面でのモデリングとは、3 次元 CG ソフトウェアで用いられている 2 自由度の入力装置と 2 次元の表示装置を用いた作業環境における 3 次元形状データの作成手法のことである。通常 2 自由度の入力装置としてはマウスやタブレットが、表示装置としては CRT モニタが用いられる。

本節では、いくつかの 3 次元 CG ソフトウェアを例として 2 次元平面でのモデリングの概要を紹介した後、その利点と欠点について述べる。

2.1.1 2 次元平面でのモデリングの概要

2 次元平面でのモデリングにおいては、3 次元の作業空間は何らかの投影法を用いた投影図として 2 次元の画面に表示される。投影図には、X, Y, Z の各軸に沿った直交投影法により作成される 3 つの直交投影図 (3 面図)[1][2] や、任意の視点からの平行投影図 [3][4] などがあり、多くの 3 次元 CG ソフトウェアではこれらを組み合わせて用いている。また、画面には投影図として表示された作業空間に加えて、様々なモデリング操作と結び付いた多数のボタンやメニューなどが表示されている。

これらのボタンやメニューにより提供されているモデリング操作は、立体形状の作成に関する操作と、立体形状の変形に関する操作に大別される [19]。2 次元平

面でのモデリングでは、2自由度の入力装置を使って作業空間の投影図上でモデリング操作を実行する。実際のモデリング過程では、ユーザはまず大まかな立体形状を作成し、それに対して細かな変形を繰り返し加えることによって、望ましい立体形状に近づけて行く。ここでは、まずモデリング操作に必要なパラメータの指定方法について述べた後、実際のモデリング過程に従い、立体形状の作成操作、立体形状の変形操作の順に、それぞれ具体例を示しながら説明する。

まず、パラメータの指定方法について述べる。ユーザはモデリング操作をその操作が必要とする様々なパラメータを指定することで実行する。例えば、任意の位置にオブジェクトを移動させる操作に必要なパラメータは X, Y, Z の移動量であり、立体形状の拡大・縮小に必要なパラメータは、その中心点の座標と倍率である。ユーザは2自由度の入力装置を用いて、同時に2つのパラメータを対話的に指定することができる。また、2つを越えるパラメータを指定する場合、そのうち2つは2自由度の入力装置により対話的に指定し、残りのパラメータは、1) キーボードから数値を直接入力する、2) 別の投影図で指定する、3) スライダを用いる、4) キーボードの特定のキーを押しながらマウスを移動させる、等により指定する[20]。多くの3次元CGソフトウェアではこれらをユーザの好みにより使い分けることができる。

2つを越えるパラメータを指定する作業の具体例として、任意の1点の3次元位置を指定するという、3つのパラメータを必要とする操作を考える。ここで例として取り上げる2つの3次元CGソフトウェアでは、この操作をマウスを用いて対話的に行なうことができる。まず、3面図を用いる LightWave3D[1] では、1つの投影図上で位置を指定することで、3つのパラメータのうち2つを指定することができる。ユーザは、3つの投影図うち少なくとも2つの投影図上で位置を指定することで、3次元位置を指定することができる。一方、任意視点からの平行投影図を用いる Form・Z[3] では、3次元位置を指定するのに2つの操作を必要とする。まず、作業空間に存在する2次元の作業平面の位置と姿勢を、キーボードから数値を入力して設定する。次に、投影図に表示されたこの作業平面に対して、マウスにより位置を指定することで、対応する3次元位置を指定することができる。

次に、立体形状の作成操作について述べる。既存の3次元CGソフトウェアは、

それぞれに特色のある立体形状作成手法を用いている。多くの3次元CGソフトウェアに共通した手法として、予め用意されている直方体や球といった基本的な立体形状(プリミティブ)を、その形状を規定する複数のパラメータを指定することで作成する方法がある。また、プリミティブを利用する以外の立体形状作成手法としては、平面形状を移動または回転させ、その軌跡から立体形状を作成する手法(掃引)が、多くの3次元CGソフトウェアに備わっている。

プリミティブを利用する立体形状作成手法の例として、3D Studio MAX[2]を取り上げる。3D Studio MAXにはプリミティブとして直方体、球、円柱、円環体、カプセルなどが用意されている。3D Studio MAXでの直方体の持つパラメータは、長さ、幅、高さ、底面の中心などであり、円柱のそれは半径、高さ、側面の数などである。また、Form・Zでは先に述べた作業平面上で平面形状を作成した後、メニューから押し出し量を選ぶことで、平面形状を作業平面の垂直方向に平行移動させて、その軌跡から立体形状を作成することができる。

最後に、立体形状の変形操作について述べる。変形操作は、その3次元形状データの表現方法によってやり方が異なる。多くの3次元CGソフトウェアでは、3次元形状データを面、頂点、辺とそれらの接続情報として表現している。ユーザはこれらに選択・移動、削除・追加などの操作を行なうことで立体形状を変形していく。例えば、LightWave3Dでは頂点と面が操作対象であり、これらの移動が最も基本的な変形操作である。他にも曲げる、伸ばす、歪ませるなど数多くの変形操作が可能であるが、これらはすべて頂点を移動させることで実現されている。

2.1.2 2次元平面でのモデリングの利点と欠点

2次元平面でのモデリングには、1) 入力の制御が容易である、2) 安定した操作が行なえる、という2つの利点がある。これによりユーザは、2次元平面上での操作に関して高い操作性を得ることができる。以下、この2つの利点について詳しく述べる。

入力の制御が容易: 人間にとって複数の自由度を同時に制御するのは難しい作業であるが、少ない自由度ならば容易に制御できる [13]。2次元平面でのモデリングで使用する、マウスやタブレットなどの入力装置では、ユーザが同

時に制御する自由度は2と少ない。このため、ユーザは意図した通りの入力を行なうことができる。

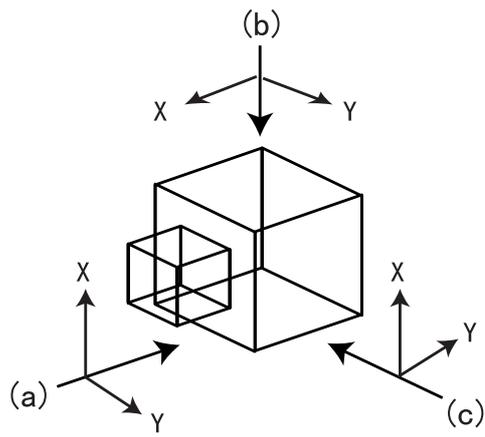
安定性が高い： 通常2自由度の入力装置は机のような平らな実物体の上で操作を行なうため、ユーザは操作時に摩擦力や抗力といった物理的な制約を受ける。このような物理的な制約により、ユーザは安定した操作を行なうことができる。

一方、2次元平面でのモデリングには、1) 立体形状の把握が困難、2) 操作の直感性やモデリングの効率が低い、という2つの欠点がある。以下、この2つの欠点について詳しく述べる。

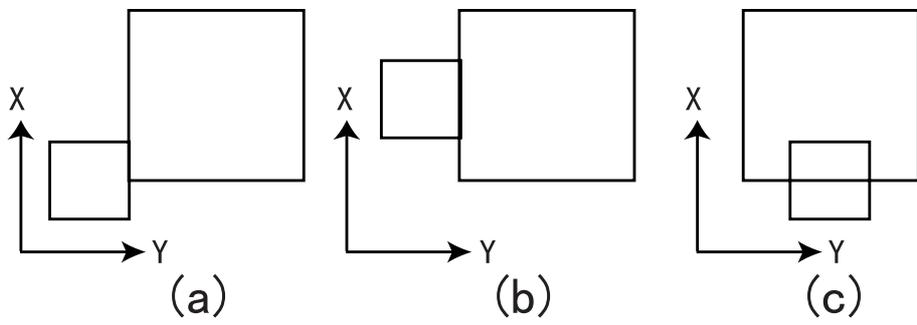
立体形状の把握が困難： 2次元平面でのモデリングでは、投影図を用いて立体形状を表示するために、その立体形状を3次元的に把握することが困難である。例として、隣接する大小2つの立方体の平行投影図と直交投影図を図1に示す。図1からわかるように、平行投影図だけでは2つの立方体の正しい位置関係を判断することはできない。一方、3つの直交投影図では両者の位置関係は一意に定まるが、これら3つの図からその3次元的な立体形状を想像するのは難しい。

操作の直感性・モデリングの効率が低い： 3次元空間での操作を2自由度の入力装置で行なうため、本質的に3次元の操作でも2次元の操作の組み合わせに分解しなければならない。例えば、作業空間を3面図により表示している環境では、立体形状をXYZ軸に関してすべて同時に移動させることはできない。ユーザは1つの投影図でXY軸に関する移動を行ない、次に別の投影図で残るZ軸方向の移動を行なう。このように、2次元平面でのモデリングでは、ユーザは意図する操作をそのまま3次元的に行なうことができないため、操作の直感性が不足している。また、作業の工程数が増えるためモデリングの効率が悪くなる。

以上のような2次元平面でのモデリングの持つ欠点を、人工現実感の技術を利用することで克服する試みとして没入型モデラが研究されてきた。次節では、没入型モデラの3次元空間でのモデリングについて述べる。



平行投影図



直交投影図

図 1 平行投影図と直交投影図

2.2 没入型モデラの 3 次元空間でのモデリング

没入型モデラとは、人工現実感技術を用いて提示される仮想空間に没入しながら立体形状を生成できるモデリング環境のことを指す。本論文では、このような没入型モデラにおけるモデリングを 3 次元空間でのモデリングと呼ぶ。本節では、3 次元空間でのモデリングの概要を既存の没入型モデラの例と共に示し、その利点と欠点を論じる。

2.2.1 3 次元空間でのモデリングの概要

没入型モデラでは仮想空間に没入しながらモデリングを行なう。仮想空間への没入感を得るためには正しい立体像の提示が重要であるため、没入型モデラでは視点追従を伴う両眼立体視により仮想空間を提示する。具体的には、磁気トラックを取り付けた HMD を用いて仮想空間を提示する手法や [6][8][9][10]、ディスプレイに左右眼画像を交互に表示し、それを磁気トラックを取り付けた液晶シャッター眼鏡を通して観察する手法 (時分割立体表示) [11][12] が用いられている。

仮想空間には 6 自由度の 3 次元入力装置に追従するカーソルが存在している。ユーザはこの入力装置を持った手を動かすことでカーソルを移動させ、同時に、入力装置に取り付けられたスイッチを押下する、またはキーボードを併用して特定のキーを押下することによってモデラに命令を伝えることができる。モデリング操作の選択は 2 次元平面でのモデリングと同様に、ボタンやメニューを通して行なわれる。

仮想空間でカーソルを用いてモデラに命令を伝える具体的な手順を、Butterworth らの開発した 3DM[6] を例として示す。3DM では、仮想空間に 3 次元化したアイコンの並んだメニューが浮遊しており、これを通してモデリング操作を選択する。ユーザは片手に把持した 6 自由度の 3 次元マウスに追従するカーソルをアイコンに接触させ、その状態でマウスのボタンを押下することで、そのアイコンに対応したモデリング操作を選択、または実行することができる。

没入型モデラでの立体形状作成手法の代表的なものは、2 次元平面でのモデリングと同様にプリミティブによるものである。ユーザは作成するプリミティブを選択し、それぞれに決められた作成手順に従った操作を行なってプリミティブを

作成する。他の作成手法としては、3つの頂点を指定して面を構成して行く方法や、線分や面を把持・移動させて、その軌跡から立体形状を作成する方法(掃引)がある。

例えば清川らの開発した VLEGO[7] では、作成可能なプリミティブを示す3次元アイコンがワイヤフレームの立方体に並んで格納されている。カーソルを3次元アイコンに接触させた状態で、3次元マウスのスイッチを押下すると、そのプリミティブが選択され、カーソルの先端にプリミティブが現れる。このカーソルの先端に現れたプリミティブは、押下したスイッチを解放するまでカーソルの動きに追従して移動する。ユーザが望ましい位置までプリミティブを移動させ、スイッチを解放することでプリミティブの作成は完了する。

また Deering らの HoloSketch[11] では、スイッチを押下してメニューから作成するプリミティブを選択すると、カーソルの位置に新しいプリミティブが作成される。この時押下したスイッチを解放するまでの間、3次元マウスの位置・姿勢の値の変化により、プリミティブの形状を規定するそれぞれのパラメータを変更することができる。ユーザは3次元マウスを移動させてパラメータを調節し、望ましい形状になった時点でスイッチを解放することで、プリミティブの形状を決定する。

一方 3DM では、プリミティブによる立体形状の作成に加えて、1) 3つの頂点を指定することで面を作成する、2) 既存の辺から掃引により面を作成する、の2つの立体形状作成手法が利用できる。ユーザはまず、メニューから 1), 2) のどちらかの手法を選択する。1) を選択すると、スイッチを押下することでカーソルの先端に頂点を作成、またはカーソルの先端にある既存の頂点を指定できるようになる。これにより、3つの頂点を作成、あるいは指定することで意図する3角形の面を作成することができる。また 2) を選択すると、既存の面を構成する辺を把持・移動し、その軌跡から任意の形状をした面を作成することができる。

以上のようにして作成した立体形状に対して、多くの没入型モデラで共通して行なえる操作としては、1) 立体形状全体の把持・6自由度の移動(3自由度の平行移動と3自由度の回転)、2) 拡大・縮小、3) 複数の立体形状のグループ化、などがある。また、いくつかの没入型モデラでは両手に3次元入力装置を持ち、両手操作

により立体形状を操作することができる [7][9][10].

例えば VLEGO では、両手で同一の立体形状 (VLEGO ではブロックと呼ばれる直方体) を把持した状態から、両手の間隔をさらに広げることで拡大、狭めることで縮小というように、両手の位置に対応して実時間でその大きさを変更できる。また、両手で 2 つのブロックをそれぞれ把持し、一方の上の面と、もう一方の下の面が接する状態にして、どちらか一方を解放することで 2 つのブロックをグループ化することができる。

2.2.2 3次元空間でのモデリングの利点と欠点

没入型モデラにおける 3次元空間でのモデリングでは、1) 立体形状の視認が容易である、2) 操作が直感的に行なえる、3) 対話性の高いモデリングが行なえる、という 3 つの利点がある。以下、この 3 つの利点について詳しく述べる。

視認の容易性: 視認の容易性とは、3次元の立体形状を直接 3次元的に視認できることを指す。没入型モデラでは、運動立体視を伴った両眼立体視により仮想空間を提示している。そのためユーザは、3次元的な視認に欠かせない奥行き情報を得ることができ、立体形状や立体形状間の位置関係を簡単かつ確実に把握できる。

操作の直感性: 操作の直感性とは 3次元の立体形状を直接 3次元的に操作できることを指す。没入型モデラでは、ユーザは 6自由度の 3次元入力装置を把持した手を、対象物の位置まで移動して直接操作を行なう。このような操作方法は、日常の実空間での作業感覚に近く直感的であり、作業効率に関しても優れている [21].

高い対話性: 没入型モデラでは、仮想空間に没入したまま立体形状を実時間で対話的に操作できる。すなわち、立体形状に対する操作は直ちにモデリング結果に反映され、その操作結果が即座に視認できるため、対話性の高いモデリングが行なえる。

これらの利点を持つ没入型モデラにおける 3次元空間でのモデリングは、前節で指摘した 2次元平面でのモデリングの欠点を克服している。しかし、1) 高自由

度の入力装置を制御することが難しい, 2) 正確かつ安定した操作を行なうのが難しい, という2つの欠点を持つため操作性が低い. 以下, この2つの欠点について詳しく述べる.

入力の制御が困難: 一般に, 複数の自由度を同時に制御することは困難である [13]. そのため, 6自由度の3次元入力装置を用いた操作では, ユーザの意図した通りの結果を得にくい. 例えば, Liang らは JDCAD[5] で, 先に述べた HoloSketch のプリミティブ生成手法と同様に, 3つ以上のパラメータでその形状が決定されるプリミティブの生成手法として, 磁気トラッカの位置・姿勢の値の変化により, それぞれのパラメータを変更するという手法を試みたが, 良い結果を得ることはできなかった. Liang らはその理由として, 複数の自由度を同時に制御することはユーザの手に負担を掛けるため難しいことや, 姿勢の制御は位置を制御するよりも難しいこと等を挙げている.

操作の正確性・安定性が低い: 人間の空間指示能力は本質的に低いため, ユーザは正確かつ安定した操作を行なうことができない. 例えば, 腕をあげて指先で空中を指示する場合, 指先は常に振動し安定しない. 実空間では触覚や力覚がこれを補い, 物理的に手が目標物体と接触する, あるいは把持している物体が他の物体と接触することを頼りに正確な選択や配置を行なっているが, 仮想空間でこれらの感覚を自然に提示することは困難である. 竹村らは, 力覚や触覚の伴わない仮想空間内でユーザが意志通りに3次元位置を指示できる精度は1cmから2cm程度と報告している [22].

この2つの欠点による3次元空間でのモデリングにおける操作性の低さは, 制約の不足に原因があると指摘されている [8]. 次節では, 3次元空間でのモデリングに制約を導入することで, 操作性を高める試みについて述べる.

2.3 3次元空間への制約の導入による操作性の向上

3次元空間でのモデリングの操作性を高めるために用いられる制約には, 大きく分けてソフトウェアによる制約と物理的な制約の2つがある. 既存の没入型モ

デラの多くはソフトウェアによる制約を用いており、複数用意されているソフトウェアによる制約を必要に応じて用いることができる。

ソフトウェアによる制約の代表的な手法としては、オブジェクトの位置・姿勢を離散的に制約する離散配置制約(グリッド制約)、オブジェクト同士の衝突を検出しその位置を補正する干渉回避、本来6自由度であるオブジェクトの移動(3自由度の平行移動と3自由度の回転)の自由度を2次元の平行移動や1次元の回転などに制限する手法などが挙げられる [6][7][8][10]。

例えば、Bowman らによる CDS(Conceptual Design Space)[8] では、オブジェクトを移動させる操作において、その自由度を1次元に限定する制約を利用することができる。ユーザはオブジェクト自体を把持することで6自由度の移動を行なうことができ、また、オブジェクトに取り付けられた十字型の制御ハンドル(作業空間の X, Y, Z 軸と対応している)のいずれかを把持することで、その軸に制限された移動を行なえる。

一方、物理的な制約には実際に力覚を再現してユーザの手の動きの自由度を制限する手法が提案されている [15][16]。平田らの SPIDAR II(Space Interface Device for Artificial Reality II)[14] では、ユーザが指に装着するポインタに4本の糸が取り付けられており、この糸を適宜拘束することで仮想物体に触った感覚を与える。この SPIDAR II と立体映像表示装置を組み合わせた仮想作業空間で、ユーザは2本の指にポインタを装着し、力覚を感じながら仮想物体を把持・移動させて積木を組み立てることができる。このような力覚を再現する手法では、仮想物体に対する正確な操作が行なえるが、装置が特殊で大がかりになる上、ユーザの作業範囲を装置の稼働範囲に束縛するという問題が指摘されている [7]。

より簡便な方法として、実物体である机、モニタの表面、手に持ったクリップボードなどの平面(実平面)を利用して、物理的な制約を提供する手法も考えられる [13]。この手法では、ユーザの手の動きを実平面上での2自由度の動きに制限することしかできないが、大がかりな装置を必要としないという利点がある。

このような実平面による物理的な制約を、3次元空間に導入したモデリング環境として、Zelevnik らによる ErgoDesk[23] がある。ErgoDesk では、表示装置である傾斜型ディスプレイの表面を実平面として利用することで、物理的な制約を提

供しており、ユーザはこの実平面上でライトペンを使って 2 自由度に限定された操作を行う。

一方、ErgoDesk でのモデリングは、一定の規則に従って描かれた 2 次元の図形を、ジェスチャ認識により自動的に 3 次元化するという手法 [24] を用いており、ほとんどすべてのモデリング操作を実平面上で行うように設計されている。同じ傾斜型ディスプレイを用いた、両眼立体視により提示される 3 次元空間では、磁気トラッカを用いて 3 次元形状の見回しと注釈の付加が行なえるが、没入型モデルのような 3 次元空間でのモデリングは考慮していない。

2.4 本研究の方針

前章で述べたように、本研究の目的は複雑な任意の立体形状を素早く容易に作成することが可能なモデリング環境を開発することである。本節に至るまで、1) 2 次元平面でのモデリング、2) 3 次元空間でのモデリング、3) 制約を導入した 3 次元空間でのモデリング、の順で既存のモデリング環境について述べてきたが、この 3 つの中に本研究の目的を達成しているモデリング環境は存在していない。

例として、図 2 に示す立体形状を作成することを考える。一般に 2 次元平面でのモデリングでは、掃引により次の手順で作成する。まず、1) 星型の平面形状を作成し、2) それを移動させる経路を指定し、3) 移動させた軌跡から立体形状を作成する。しかし、この場合 2) において複雑な 3 次元の経路を指定する必要があり、2 次元平面でのモデリングの投影図に対する操作では実現が困難である。また、3 次元空間でのモデリングで同じ手順により作成すると、2) の経路を指定する作業は容易に行なえるが、1) の星型のような複雑な平面形状の作成は難しい。

このように、既存のモデリング環境で図 2 のような立体形状を作るのが困難であるのは、その作業空間として、2 次元平面と 3 次元空間の 2 種類しか存在せず、そのため両者の持つ利点のどちらか一方しか利用できないからである。

そこで、本研究では 2 次元平面でのモデリングと 3 次元空間でのモデリングを単一の作業空間として組み合わせ、両者の持つ利点を同時に利用しながら、立体形状の作成が行なえるモデリング環境を提案する。提案環境では、既存のモデリング環境では作成するの難しかった図 2 のような立体形状も、まず、2 次元平面での

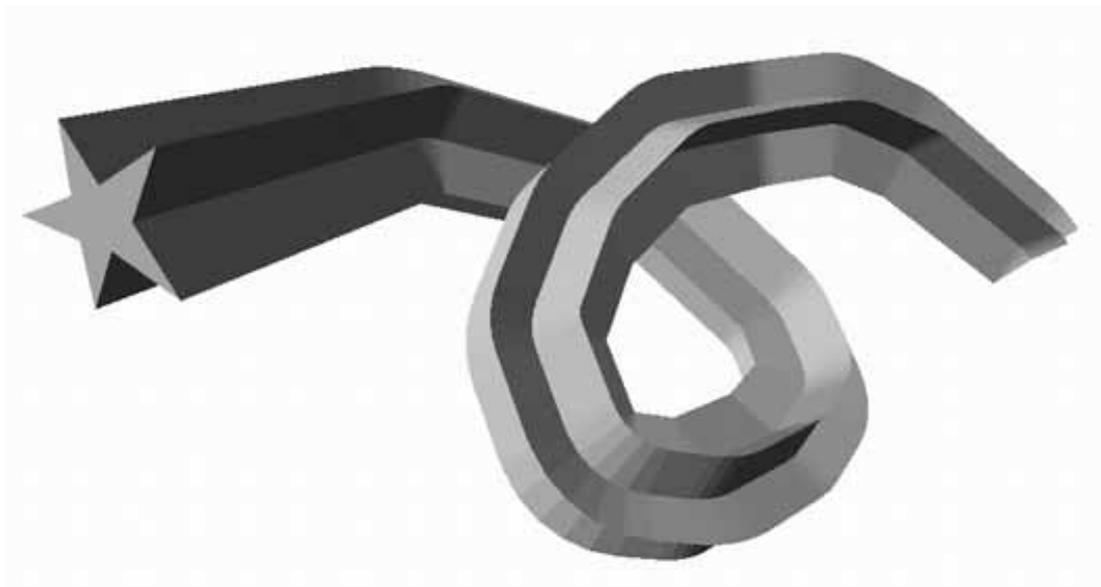


図 2 既存のモデリング環境では作成しにくい立体形状の例

モデリングにより星型の平面形状を作成し、次に、その掃引経路を 3 次元空間でのモデリングにより指定する、というように、両空間の利点を同時に利用することで、素早く容易に作成できると期待される。

本研究では、このような提案環境を実現するために、2 次元平面でのモデリングの利点を持つ 2 次元平面を 3 次元空間に実現し、両空間での利点を活かしたモデリングを可能にする手法を提案する。以下、それぞれを実現するための方針を示す。

2.4.1 2 次元平面でのモデリングの利点を持つ 2 次元平面の実現

2 つの空間を単一の作業空間として組み合わせるためには、操作を 2 次元に拘束する 2 次元平面を 3 次元空間中に実現する必要がある。また、この 2 次元平面は、2.1 節で 2 次元平面でのモデリングの利点として述べた、1) 入力制御が容易である、2) 正確かつ安定した操作が行なえる、の 2 点を満たすものでなければならない。

操作を 2 次元に拘束する手段としては、2.3 節で述べたような、ソフトウェアに

よる制約と物理的な制約がある。これらの制約を利用して、それぞれ次のような方法で 3 次元空間中に 2 次元平面を実現できる。

1. 3 次元空間中に 2 次元平面を設定し、3 次元入力装置から得られる 3 次元位置を、その 3 次元位置から 2 次元平面に下ろした垂線の足に変換する
2. 3 次元空間に設定した 2 次元平面がユーザの手に与える力覚を再現することで 2 次元平面を提供する
3. 何らかの実平面を利用した物理的な制約により 2 次元平面を提供する

本研究では、これらの実現方法のうち、3. の実平面を用いた物理的な制約を採用する。その第 1 の理由は、2. と比較して大がかりな装置を必要としないことであり、第 2 の理由は、次に述べる Schmandt の実験が示すように、物理的な制約により提供される 2 次元平面の方が、1. のソフトウェアによる制約により提供される 2 次元平面よりも、提案環境が求める 2 次元平面としては優れているからである。Schmandt は、複数の層からなる VLSI 回路の設計タスクを立体視による 3 次元の作業空間で行い、入力装置としてタブレットを用いた場合（操作対象とする層を選択する機能を持つ）と、ソフトウェア制約であるグリッド制約を適用した磁気トラッカを用いた場合とで比較して、前者の有効性を示した [25]。

Schmandt が実験に用いたタスクの特徴は、操作対象が 3 次元空間中の 2 次元平面上に限定されており、その平面上での安定した精度の高い入力が必要とされている点にある。タブレットという実平面を用いた物理的な制約は、入力を 2 自由度に制限するだけでなく、実世界と同様の摩擦力や抗力により高い操作性を提供する。そのため、実際には何の支えも無い 3 次元空間で操作するソフトウェアによる制約よりも、このタスクに関して有効であったと思われる。

2.4.2 2 次元平面と 3 次元空間の利点を活かしたモデリング

本研究では、実平面を利用して実現する 2 次元平面と 3 次元空間で、両空間でのモデリングの持つ利点を同時に利用しながら、モデリングが行なえるようにする。そのようなモデリングを可能とするために満たすべき要件として、次の 3 つを挙げる。

1. 両空間の共存: 2次元平面と3次元空間を、物理的に単一の作業空間として実現し、さらに、その作業空間において、両空間を常に同時に存在させる。

もし、両空間が物理的に離れた場所に存在していると、ユーザは両者のモデリングを行なうために、物理的な移動を必要とする。また、物理的には単一の空間として実現しても、その中で2つの空間が別々に存在していると、何らかの切替え操作が必要となる。

本研究では、両空間を共存させることで、物理的な移動や、切替え操作を必要とせずに、両空間でのモデリングを行なうことができるようにする。

2. 単一の操作デバイスの導入: 2次元平面でのモデリングと3次元空間でのモデリングの双方において共通して使用できる操作デバイスを導入する。これにより、2次元平面と3次元平面で、それぞれに異なったデバイスを用いる環境で起こり得るデバイスの持ち変えに伴う物理的負荷と、各操作で使用するデバイスの判断に伴う認知的負荷を防ぐ [26]。

3. 適切な空間でのモデリング操作の実装: 個々のモデリング操作はそれに適した作業空間の次元数が異なるため、モデリング操作をその性質に適した空間で行なえるように実装する。2次元平面でのモデリングの持つ利点を必要とするモデリング操作は2次元平面に実装し、3次元空間でのモデリングの持つ利点を必要とするモデリング操作は3次元空間に実装する。

これまで述べて来た方針に従って開発した試作システムについて、次章で詳しく述べる。

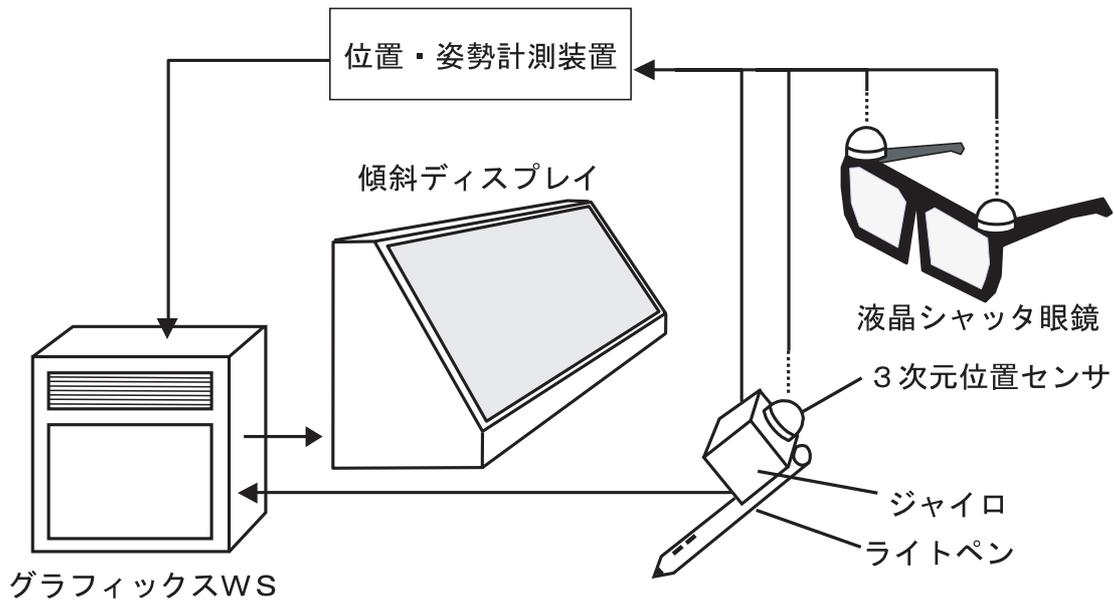


図 3 システム構成図

3. 2次元平面と3次元空間を組み合わせたモデリング環境の開発

前章では、2次元平面と3次元空間を組み合わせたモデリング環境を提案し、提案環境を実現するための方針を示した。本章では、その方針に従い提案環境を試作したシステムの詳細と、実装したモデリング操作の実際について詳しく述べる。

なお実装にあたり、プログラミング言語として C++言語 [27] を、グラフィックスライブラリとして OpenGL[28]、および独自のツールキット [29] を使用した。

3.1 プロトタイプシステムの概要

本研究では、2.4節で述べたように、実平面による物理的な制約を利用して3次元空間中に2次元平面を実現する。プロトタイプシステムでは、この2次元平面を ErgoDesk と同様に傾斜型ディスプレイとライトペンによって実現した。この傾斜型ディスプレイを中心に構成される本システムの機器構成を図3に、モデリ

ング作業の様子を図 4に示す。

本節では、まず、構成図中の機器の説明を行い、次に、ユーザがモデリングに使用するペン型操作デバイスの詳細を示す。

3.1.1 実行環境

プロトタイプシステムの全体構成は図 3に示す通りである。2次元平面での高い操作性と広い作業空間を実現するため、画像表示装置として 86 インチ背面投影式傾斜型ディスプレイ (ソリッドレイ社製) を採用した。位置・姿勢計測装置には、傾斜型ディスプレイの鉄製の筐体から影響を受けない点を重視して、姿勢計測装置のジャイロセンサと位置計測装置の超音波センサを融合させた IS-600 (InterSense 社製) を用いた。ユーザは 2 つの超音波センサをフレームの両端に取り付けた液晶シャッタ眼鏡 CrystalEyes (StereoGraphics 社製) を通して仮想空間を両眼立体視する。また同時に、ユーザは超音波センサにより計測された視点に基づき、実時間で 3 次元空間を運動立体視できる。なお、本システムでは、ユーザの視点位置を液晶シャッタ眼鏡に取り付けた超音波センサの計測位置から、垂直下方向に 5cm ずらした位置であると仮定している。

また、入力装置としてライトペン LP-213 (MicroSpeed 社製) と IS-600 を組み合わせたペン型デバイスを製作した。これらの計測装置からの入力を処理し、傾斜型ディスプレイに描画を行うグラフィックスワークステーションには、Onyx2 IR2 (SGI 社製) を用いた。

3.1.2 ペン型操作デバイスの詳細

2次元平面での操作と 3次元空間での操作を単一のデバイスで行なうことができるように、図 5に示すペン型デバイスを導入した。2次元平面での入力位置は、ライトペンのペン先とし、指示位置は傾斜型ディスプレイ画面上にドット型カーソルにより示される。3次元空間での入力位置は、位置・姿勢センサから求めたペン先から 10cm 延長した位置とし、通常矢印型のカーソルにより 3次元空間内に示される。この矢印型カーソルは、その柄の部分が根本に近くなるほど透明になり、ペン先とは直接接続しないように描画されている。なお、これらの入力は、ペ

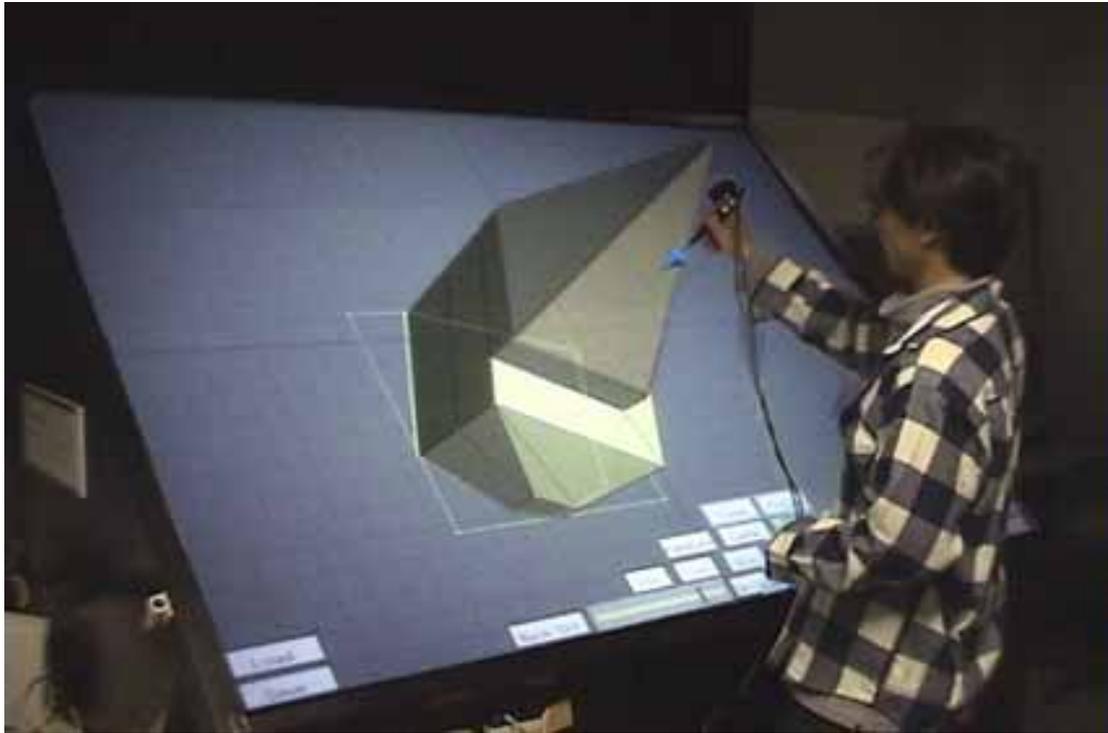


図 4 試作システムでのモデリングの様子

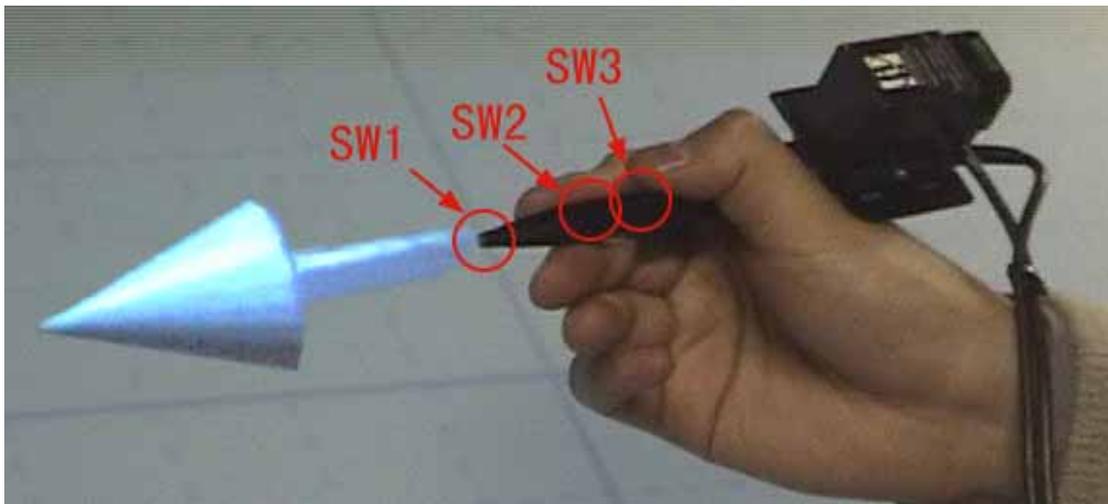


図 5 ペン型デバイスとカーソル

ン先がディスプレイ表面から 5mm 以内にあるときは 2 次元平面, それ以外では 3 次元空間が対象となる.

また, 図 5 に示すように, ライトペンにはスイッチがペン先に 1 つ, 側部に 2 つ配置されており, ペン先から順に SW1, SW2, SW3 と呼ぶ. SW1 はペン先をディスプレイ表面に押しつけることによって押下され, SW2 と SW3 は親指によって押下する. すべてのモデリング操作はこれらのスイッチにより行うことができ, 他の操作デバイスを必要としない.

3.2 作業空間の構成要素

図 6 に示すように, 作業空間は操作対象であるオブジェクトと 3.2 節で述べたカーソルの他に, 2 次元平面, オブジェクトと 2 次元平面の交差面, 複数のボタンから構成される. 本システムでは, 従来の CAD や 3 次元 CG ソフトウェアと同様に, ボタンを通してシステムに命令を伝える. これらのボタンに対する操作は, 本質的には 2 自由度しか必要としていない [8]. そのため, 本システムではユーザが正確に素早くシステムに命令を伝えることができるように, ボタンを 2 次元平面上に配置した. 以下では, それぞれの構成要素について説明する.

2 次元平面: 2 次元平面は物理的には傾斜型ディスプレイの表面であり, システムの作業空間においては XY 平面にあたる. 作業空間の座標系は, 傾斜型ディスプレイの中心を原点として, ディスプレイ表面に垂直な方向を Z 軸としている. 2 次元平面には X 軸と Y 軸を示す十字線と, 5cm の間隔で格子状に並ぶグリッドが描画されており, それ以外の領域は灰色で塗りつぶされている. また, ユーザは 2 次元平面の表示・非表示を “Back Off” ボタンにより切替えることができる. 図 7 に示すように, 2 次元平面を表示している状態 (図中 (a)) では, 2 次元平面より奥の領域は 2 次元平面に遮られて見ることができないが, “Back Off” ボタンで非表示にすることで奥の領域も見ることができるようになる (図中 (b)).

ボタン: 2 次元平面上に配置されているボタンには, 機能の名称が書かれており, 2 重の長方形で囲まれた形をしている. ユーザはライトペンのスイッチ SW1

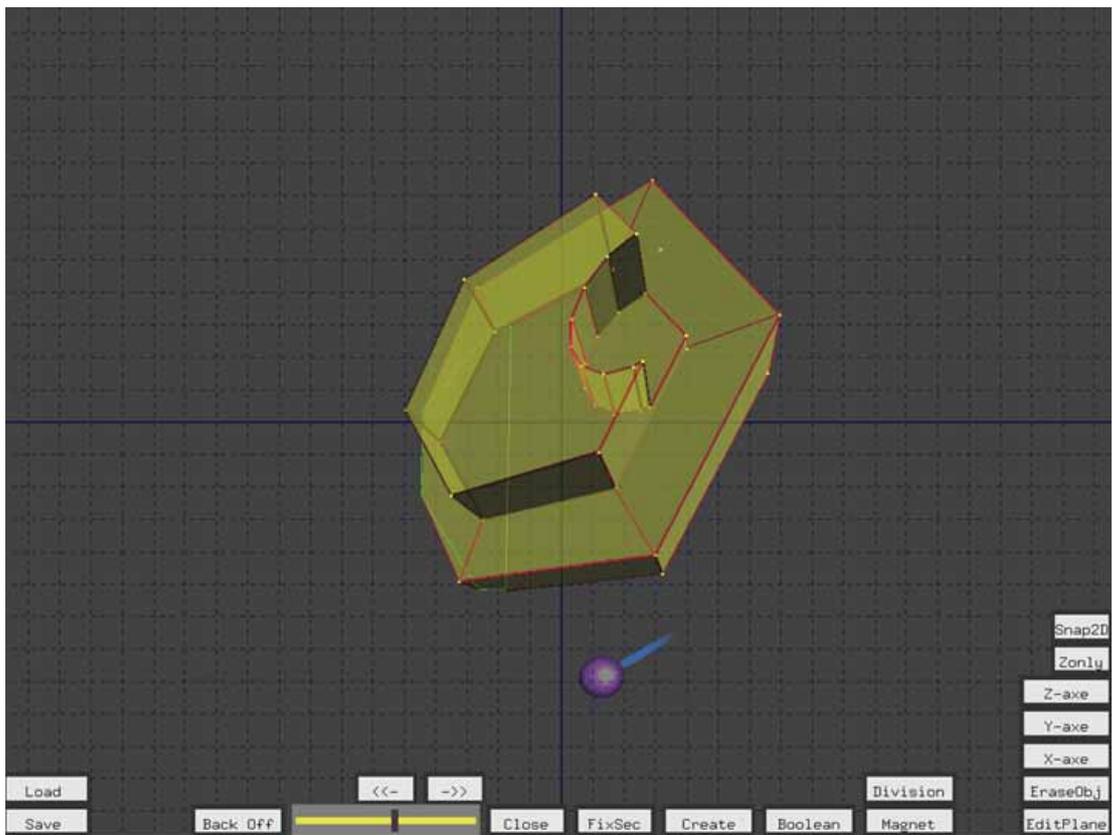
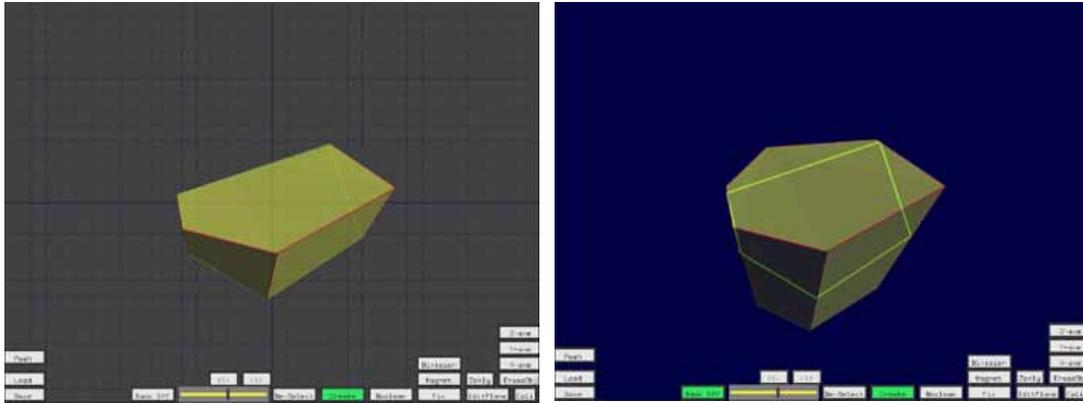


図 6 作業空間の構成要素



(a) 2次元平面の表示

(b) 2次元平面の非表示

図 7 2次元平面の表示・非表示

でボタンを直接押すことで、各ボタンに割り当てられた機能を実行することができる。また、ボタンはその機能によりトグルボタン、コマンドボタン、ツールボタンの3つに分類される。まず、各ボタンの分類と、その機能の簡単な説明を表1に示し、以下では、これらの3種類のボタンの詳しい説明を行なう。なお、各ボタンの提供する機能を利用してオブジェクトを作成、変形する方法は次節以降で詳しく述べる。

トグルボタン: トグルボタンは押す度に ON・OFF が切り替わるボタンであり、先に述べた2次元平面の表示・非表示を切替える“Back Off”ボタン等がこれにあたる。すべてのトグルボタンは OFF の状態から始まり、これを SW1 で押すと ON の状態に切替わる。トグルボタンの内側の長方形は、ON の状態では緑色に描画され、OFF の状態では灰色に描画されている。

コマンドボタン: コマンドボタンは単一のタスクを実行するボタンである。ここでいう単一のタスクとは、タスクが開始されてから終了するまでの間、ユーザが何も行なう必要が無いタスクを指す。例えばコマンドボタンである“Division”ボタンを押すと、選択状態にある面が自動的に3角分割され、ユーザはボタンを押す以外の操作を行なう必要はない。

なお、コマンドボタンは SW1 がボタンの上で押されている間、内側の長方形の色が緑色に変化し、SW1 を離すと元の灰色に戻る。

ツールボタン：コマンドボタンが単一のタスクを実行するのに対して、ツールボタンは一連のタスクの開始をシステムに知らせる働きをする。例えば、オブジェクトの消去を行なうツールボタン “EraseObj” は、押しただけではオブジェクトは消去されず、ボタンを押した後、ユーザがカーソルをオブジェクトと接触させることでオブジェクトを消去することができる。なお、一連のタスクを実行している間は、ボタンの内側の長方形は緑色で描画され、ボタンを再び押すか、一連のタスクを終了させるボタンを押すことで、ボタンの色は元の灰色に戻る。また、ツールボタンのうち、“Create” と “Boolean” はプルダウン・メニューとして働く。これらのボタンを押すと、一連のタスクを実行している間、タスクに関連した複数のボタンが新たに表示される。

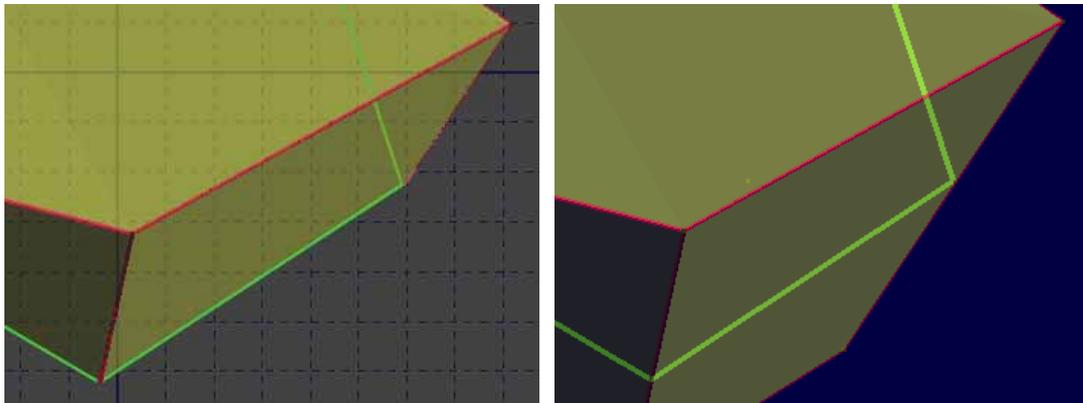
スライダ：スライダはつまみを表す小さな長方形と、つまみの可動範囲を示す横長の長方形、それらを囲む長方形の 3 つで描画されている。ユーザはつまみを SW1 でドラッグして移動させることで、選択状態にあるオブジェクトの透明度を変更することができる。つまみを左端に移動させると、透明度は最も高くなり、オブジェクトは表示されない。その状態から、つまみを右に移動させるにつれて、透明度は低下し、徐々にオブジェクトが見えるようになる。右端まで移動させると、透明度は最も低くなり、オブジェクトは完全に不透明となる。

交差面：2次元平面とオブジェクトの交差面は、2次元平面の表示・非表示に関わらず、その輪郭線が常に強調して表示されている。この様子を示すために、図 7の交差面を拡大して図 8に示す。本システムではこの交差面を、後述するオブジェクトの作成、変形操作に利用する。

表 1 ボタンの分類と各ボタンの機能

ボタンの種類	名前	機能
トグルボタン	Back Off	2次元平面の表示・非表示
	Zonly	オブジェクトの移動方向制約の有効・無効
	Snap2D	2次元平面へのスナップの有効・無効
ツールボタン	Create *	オブジェクトの作成
	Magnet	磁力による変形
	Boolean *	ブール演算による変形
	EraseObj	オブジェクトの消去
コマンドボタン	FixSec	交差面の作成
	Close	オブジェクトの作成の終了
	Division	選択面を3角分割
	Save	オブジェクトの保存
	Load	オブジェクトの読み込み
	EditPlane	選択面・頂点を2次元平面と一致させる
	X-axe	X軸に関するオブジェクトの姿勢制御
	Y-axe	Y軸に関するオブジェクトの姿勢制御
	Z-axe	Z軸に関するオブジェクトの姿勢制御
	◀-	オブジェクトの縮小, 側面数の減少等
-▶	オブジェクトの拡大, 側面数の増加等	

* はプルダウン・メニュー



(a) 2次元平面の表示時

(b) 2次元平面の非表示時

図 8 2次元平面とオブジェクトの交差面

3.3 基本操作

本システムにおけるすべてのモデリング操作は、選択状態にあるオブジェクト、頂点・面に対して効果を及ぼす。また、オブジェクトの選択状態と頂点・面の選択状態は同時に存在し得る。従って、あるオブジェクトが選択状態にあり、かつその頂点・面が選択状態にある場合が起こり得る。この場合は、オブジェクトではなく、頂点・面のみが操作対象となる。

3.2節で述べた通り、これらのすべての操作はペン型デバイスに追従するカーソルを用いて行なう。通常、ユーザは操作対象とするオブジェクトを選択し、把持・移動により操作しやすい場所に配置して目的の操作を行なう。

以下では、いくつかのオブジェクトに対する基本操作について述べる。

オブジェクトの選択: カーソルの先端を選択したいオブジェクトの内部に入れることで、そのオブジェクトを選択することができる。非選択状態のオブジェクトは、面だけで描画されているが、選択状態のオブジェクトは、非選択状態のオブジェクトと区別するために、面に加えて辺と頂点で描画される。なお、選択状態にあるオブジェクトは常に1つである。ユーザが非選択状態のオブジェクトを新たに選択すると、選択状態にあったオブジェクトは自動的に非選択状態に移行する。

頂点の選択: オブジェクトの頂点を選択するには、カーソルの先端をそれらに近付け SW2 を押下する。これにより、2次元平面ではドット型カーソルを中心とする半径 5mm の円の内部に存在する頂点を、3次元空間では矢印型カーソルの先端を中心とする半径 1cm 球体の内部に存在する頂点を選択することができる。なお、この領域に複数の頂点が存在する場合は、円または球体の中心に最も近い頂点を選択される。

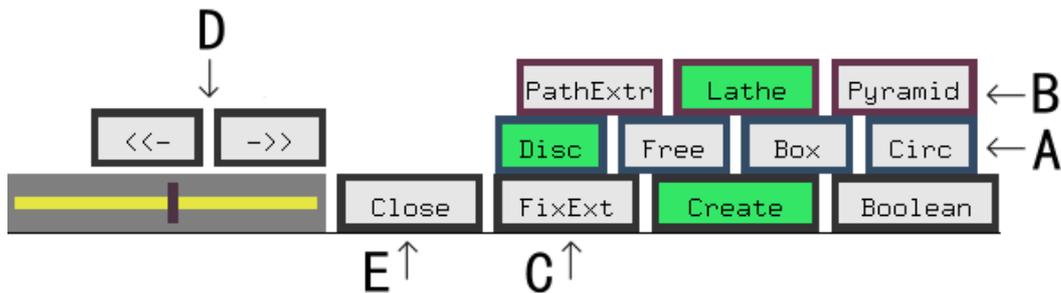
また、ユーザに選択可能な頂点を知らせるために、カーソルを中心とする領域に存在する頂点は領域外の頂点が単なる点で描画されているのに対して、小さな黄色の立方体として描画される。また、頂点の選択を実行すると、これらの立方体は緑色に変化し、選択状態にあることを示す。これらの選択状態にある頂点は、再びカーソルを近づけて SW2 を押下することで、選択状態を解除することができる。

面の選択: オブジェクトを構成する面を選択するには、頂点の選択と同じく、カーソルの先端をそれらに近付け SW2 を押下する。これにより、ペン型デバイスのペン先から 15cm 延長した直線と交差する面を選択することができる。非選択状態にある面は黄色で描画されているが、この直線と交差する面は、薄い赤色で描画される。これにより、ユーザは SW2 を押下することで選択される面を予め知ることができる。また、選択された面は濃い赤色で描画され、非選択状態にある面と区別することができる。選択状態の解除は、再びカーソルを近づけて SW2 を押下することでなされる。

把持・移動: 把持・移動の対象となるのは、頂点・面が選択されていない場合は選択状態にあるオブジェクト自体であり、選択されている場合はその頂点・面である。ユーザは SW2 と SW3 を同時に押下することでこれらを把持し、SW2 と SW3 を解放するまで把持し続けることができる。把持した状態で、2次元平面においてカーソルを移動させると、オブジェクトを2自由度で平行移動させることができ、また、3次元空間においてカーソルを移動させると、6自由度(3自由度の平行移動と3自由度の回転)で移動させることができる。



(a) “Create” ボタンを押す前



(b) “Create” ボタンを押した後

図 9 “Create” ボタンと構成要素

オブジェクトの消去: オブジェクトを消去するには、まずツールボタン “EraseObj” を押す。次に、消去したいオブジェクトを選択することで、そのオブジェクトを消去することができる。

オブジェクトの拡大・縮小: コマンドボタン “->>” と “<<-” のどちらかを押すと、選択状態にあるオブジェクトを、それぞれ拡大・縮小することができる。なお、拡大・縮小の中心点は、オブジェクトの一部が 2 次元平面と交差しているなら、交差面の重心であり、それ以外では、オブジェクトの重心である。

3.4 オブジェクトの作成

本システムで作成されるオブジェクトには、平面形状単体からなる平面オブジェクトと、その平面形状を 3 次元化することで作成される立体オブジェクトがある。ユーザは図 9(a) の状態にあるツールボタン “Create” を押して、これらのオブジェクトの作成を開始する。プルダウン・メニューとして働く “Create” を押すと、図 9(b) に示すように、複数のボタンが新たに表示される。

表 2 平面形状の作成方法とボタン名

Disc	反時計周りに指定した頂点から自由形状を作成
Free	ペン先の軌跡から自由形状を作成
Box	対角の 2 頂点から長方形を作成
Circ	中心と端点から正円を作成

平面オブジェクトは、図中 A に並ぶ 4 つのボタンを用いて、2 次元平面で平面形状を作成し、図中 E の “Close” ボタンを押すことで作成される。一方、立体オブジェクトは、作成した平面形状を “Close” せず、1) 図中 B に並ぶボタンを用いて作成する立体形状を指定する方法、2) 図中 C の “FixSec” ボタンと基本操作で述べた把持・移動により、平面形状を押し出すことで拡張する方法、のいずれかによって立体形状に拡張した後に、“Close” ボタンを押すことで作成される。

なお、本論文では “Close” を押すまでの作成課程にある平面オブジェクトを平面形状、同じく作成課程にある立体オブジェクトを立体形状と呼ぶことでこれらを区別する。以下では、まず平面形状の作成手順を説明し、次に 2 通りの立体形状の作成方法について詳しく説明を行なう。

3.4.1 平面形状の作成

本システムでは自由形状、長方形、正円の 3 種類の平面形状が作成できる。このうち自由形状には、反時計周りに入力した頂点から作る方法と、ペン先の軌跡から作る方法の 2 通りの作成方法がある。ユーザは、作成する平面形状を図 9 中 A に並ぶ 4 つのボタンにより指定した後、それぞれの平面形状を後述する手順に従って作成する。各ボタンと作成できる平面形状の関係を表 2 に示す。

平面形状を作成した後、それ単体からなる平面オブジェクトを作成したければ “Close” ボタンを押す。また、その平面形状を元に立体オブジェクトを作成したい場合は、“Close” せずに後述する 2 つの立体形状拡張方法のいずれかを行なう。

まず、4 つのボタンそれぞれにおける平面形状の作成手順を述べる。

Disc: SW1 により反時計まわりに頂点を指定して行くことで自由形状を作成

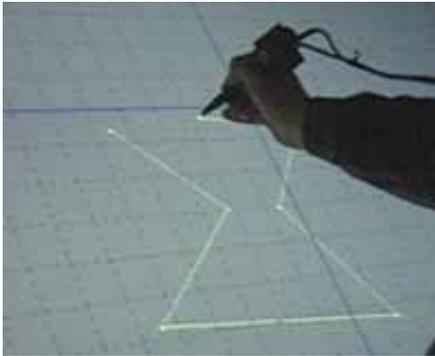
できる。はじめの頂点を指定した後では、ユーザが頂点を指定する毎に、図 10(a) のように 2 つの頂点を結ぶ辺が作成される。また、最後に指定した頂点と現在のカーソルの間には、両者を結ぶ辺がラバーバンドとして表示される。これにより、ユーザは自分の作成している平面形状の現状を確認することができる。最後に、はじめに指定した頂点を再び SW1 で押すことで、閉じた自由形状を作成することができる。なお、閉じた立体形状とした後に “Close” を押すことで、図 10(b) のように平面オブジェクトにすることができる。

Free: 2次元平面に SW1 を押しつけたままカーソルを移動させることで、そのカーソルの軌跡から自由形状を作成できる。カーソルの軌跡は逐次細かい線分の集まりとして近似され、図 10(c) のように折れ線として表示されて行く。SW1 を 2次元平面から離すことで、折れ線の最初の頂点と最後の頂点をつないだ閉じた自由形状を作成することができる。

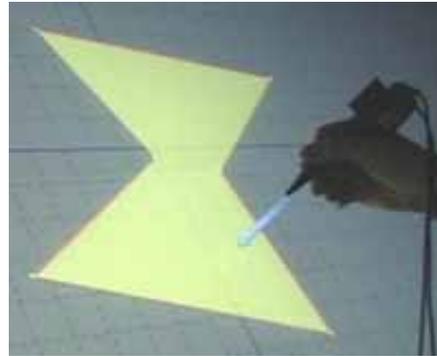
Box: 2つの頂点を指定することで、それらに対角の頂点とする、各辺がそれぞれ X 軸、Y 軸と平行な長方形を作成できる。1つ目の頂点を指定すると、図 10(d) のように現在のカーソルの位置を対角の頂点とする長方形がラバーバンドとして表示される。これにより、ユーザは2つ目の頂点を指定する前に、作成される形状を知ることができる。最後に2つ目の頂点を指定することで、4つの頂点が作成され、それらを結んだ長方形を作成することができる。

Circ: 中心点と端点の2つを順に指定することで、正円を作成することができる。ユーザはまず中心点を指定する。中心点を指定すると図 10(e) のようにラバーバンドとして現在のカーソルの位置を端点とする正円が表示される。ユーザはこのラバーバンドを参考にしながら端点を指定することで、正 14 角形で近似された円を作成することができる。なお、この 14 という数字は、図 9中 D の 2つのボタンにより増加・減少させることができる。

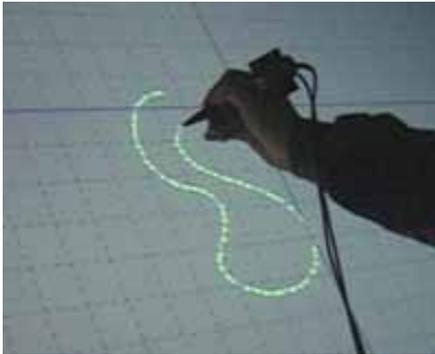
以上の 4 種類の方法によって作成した平面形状は、基本操作として述べた頂点の選択、把持・移動に加えて、頂点の追加、形状全体の拡大・縮小により編集することができる。これらの編集操作の具体的な操作手順を次に示す。



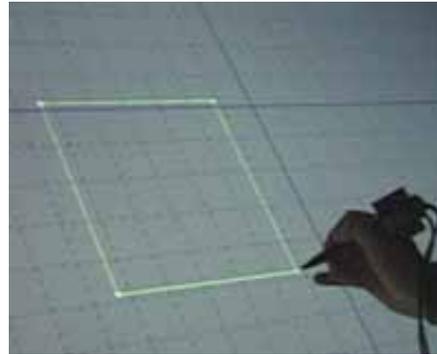
(a) “Disc” による平面形状の作成



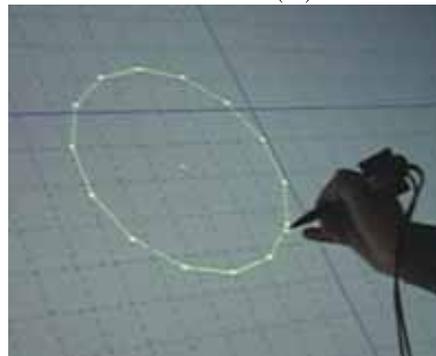
(b) “Close” による平面オブジェクト



(c) “Free” による平面形状の作成



(d) “Box” による平面形状の作成



(e) “Circ” による平面形状の作成

図 10 平面形状の作成

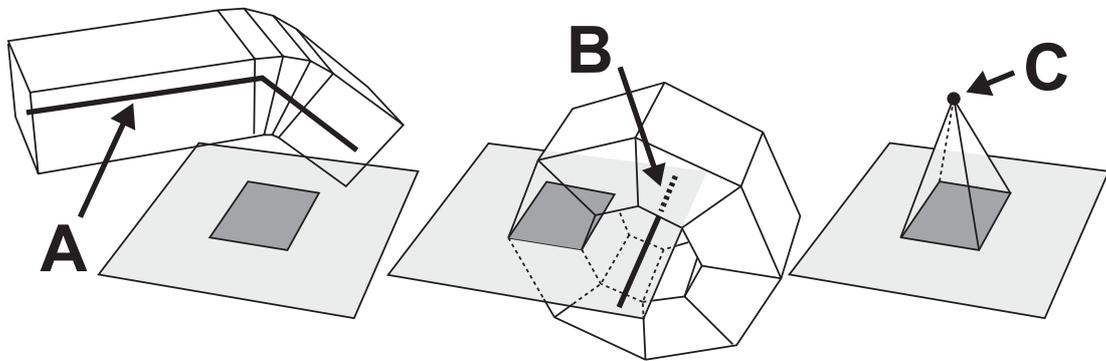


図 11 形状指定による立体形状の作成

頂点の追加: カーソルの位置と平面形状を構成する任意の辺との距離が 5mm 以内の場所で、SW3 を押下することで頂点を追加できる。なお、追加される位置は、カーソルの位置からその辺に下ろした垂線の足である。

拡大・縮小: 拡大・縮小には 2 通りの手段がある。一つは、平面形状が作成されると自動的に表示されるバウンディング・ボックスを用いる方法である。ユーザはバウンディング・ボックスの各頂点に対して選択、把持・移動を行なうことで、平面形状を任意の縦横比で拡大・縮小することができる。もう一つは、図 9 中 D の 2 つのボタンを用いる方法である。この 2 つのボタンにより、平面形状の縦横比を保ったまま拡大・縮小することができる。

3.4.2 形状指定による立体形状の作成

平面形状を立体形状に拡張する第 1 の方法は、図 11 に示すような、錐体、回転体、掃引体の 3 つの予め決められた立体形状に、規定の拡張手順に従って拡張する方法である。まず、この 3 つの立体形状と、図 9 B 列に並ぶ 3 つのボタンとの関係を表 3 に示す。

ユーザが図 9 中 B に並ぶ 3 つのボタンのいずれかを押して作成する立体形状を指定すると、掃引体では 3 次元空間に掃引経路 (図 11 中 A) を指定できるようにな

表 3 立体形状の作成方法とボタン名

PathExtr	掃引体の作成
Lathe	回転体の作成
Pyramid	錐体の作成

り、また、回転体では 2 次元平面に回転軸 (図 11中 B) が、錐体では 3 次元空間に頂点 (図 11中 C) が現れる。立体形状作成中は常に、その元になる平面形状に対して先に述べた編集操作を行なうことができ、また同時に、掃引経路、回転軸、頂点に対しても後述する編集操作を行なうことができる。

ユーザはこれらの編集操作を行ないながら、作成される立体形状を 3 次元空間において立体視により確認しながら作業を行なえる。最後に “Close” ボタンを押すと、その時点での立体形状が立体オブジェクトとして作成される。各立体形状の具体的な作成手順は次のようになる。

PathExtr: ユーザは SW3 を押下して頂点を作成して行くことで、図 11中 A のような掃引経路を折れ線として指定し、平面形状をその経路に沿って押し出した掃引体を作成することができる。3 次元空間において作成された頂点は順に連結され、折れ線として表示される。同時に、その時点での折れ線を掃引経路とする掃引体も表示される。

ユーザは掃引経路を編集するために、基本操作で述べた頂点の選択、把持・移動に加えて、頂点の追加を行なうことができる。頂点の追加は、頂点の選択と同様に SW2 を押下することで行なわれる。カーソルと経路との距離が 1cm 以内の場所で SW2 を押下すると、経路の途中に頂点を追加することができる。

また、ユーザは頂点が選択されていない状態で、把持・移動を行なうことで、掃引経路全体を 6 自由度で移動することができる。このとき経路の移動に合わせて掃引体も更新され、ユーザは全体の形状を見回すことができる。

Lathe: “Lathe” ボタンを押すと、図 11中 B のような線分が 2 次元平面上の Y 軸上に現れる。同時に、その線分を回転軸とする平面形状の回転体が側面の

数を 12 として作成され、3 次元空間に表示される。ユーザは回転軸の両端の頂点を選択し、把持・移動させることで、これを 2 次元平面上の任意の位置に配置できる。また、作成される回転体の側面の数を図 9 中 D の 2 つのボタンにより増加・減少させることができる。なお、この回転軸の移動と側面の数の変更による回転体の形状の変化は、3 次元空間において実時間で確認できる。

Pyramid: “Pyramid” のボタンを押すと、図 11 中 C のように平面形状の重心の垂直上方向 20cm の位置に頂点が追加される。同時に、追加された頂点と平面形状の各頂点を結んだ錐体が作成され、3 次元空間に表示される。追加された頂点は最初から選択状態であり、ユーザは把持・移動操作によりその位置を変更することができる。なお、この頂点の移動は 3 次元空間に表示されている錐体の形状に実時間で反映される。

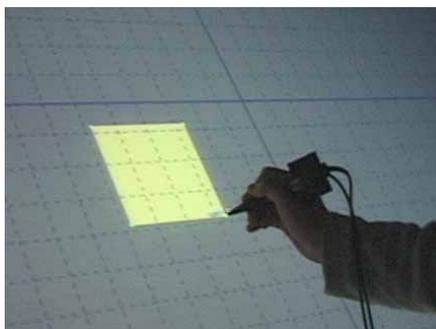
3.4.3 押し出しによる立体形状の作成

平面形状から立体形状を作成する第 2 の方法は、基本操作で述べた把持・移動と、図 9 中 C の “FixSec” ボタンを交互に用いて、平面形状を押し出しながら立体形状に拡張する方法である。以下にその詳細を示す。

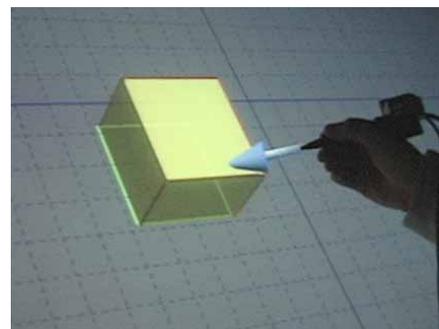
ユーザが SW2 と SW3 を同時に押下して平面形状を把持すると、即座に平面形状は垂直下方向に疑似的に無限の長さで押し出される。この押し出された立体形状を、把持・移動により任意の位置に配置して “FixSec” ボタンを押すことで、立体形状を 2 次元平面で分断し、2 次元平面より手前の領域を残すことができる。ユーザはこのとき生成される 2 次元平面との交差面（輪郭線のみで面としては存在していない）を平面形状と同様に編集して、立体形状を変形させることができる。交差面に編集を加えた後に再び立体形状を把持すると、今度は交差面が垂直下方向に押し出される。

以上の操作を繰り返して、望ましい立体形状を作成した後、最後に “Close” ボタンを押すと、立体形状は閉じた立体オブジェクトになる。このとき、2 次元平面との交差面は、“FixSec” のように輪郭線としてではなく、実際の面として作成され、立体オブジェクトの底面となる。

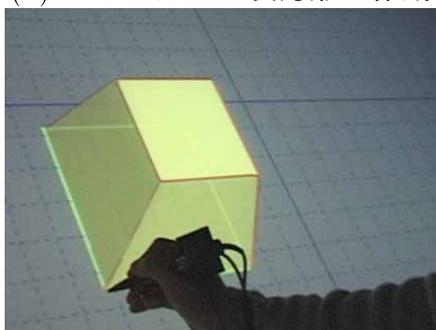
例として、長方形から立体オブジェクトを作成する過程を図 12に示す。ユーザはまず、2次元平面で長方形(a)を作成し、把持・移動することで、この長方形を垂直下方向に押し出した四角柱(b)を作成する。次に、(b)の四角柱に“FixSec”を使用して交差面を作成し、それを拡大して(c)のようにする。これを再び把持・移動することで(c)の交差面を押し出し、(d)の立体形状を作成する。最後に、“Close”を押すことで、この立体形状は立体オブジェクトになる。



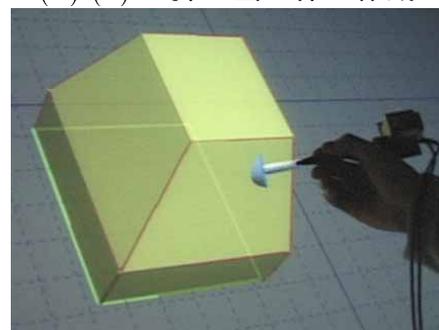
(a) “Box” による長方形の作成



(b) (a) の押し出し体の作成



(c) 作成した交差面の拡大

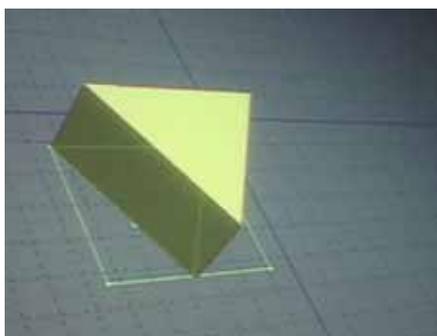


(d) (c) の押し出し体の作成

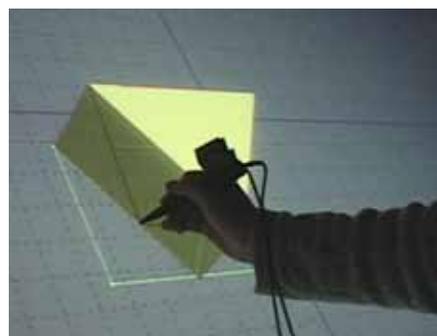
図 12 “FixSec” による立体オブジェクト作成

“FixSec” により生成される交差面に編集操作として頂点の追加を行ない、追加した頂点を移動させると、その頂点は面を構成する他の頂点とは同一平面上に存在しなくなる。これを防ぐために、頂点の追加を行なうと、頂点を追加された面は自動的に3つの頂点からなる面に分割(3角分割 [30])される。この過程を図 13に示す。まず、平面形状として三角形を作成し、把持・移動により三角柱(a)を作成

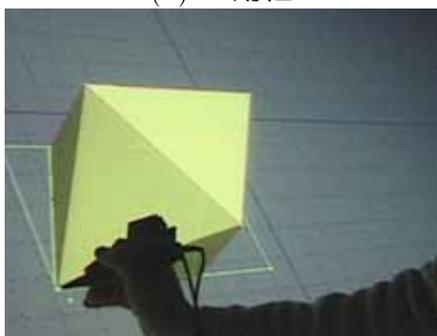
する。ここで、“FixSec” を押して作成した交差面に頂点を追加すると自動的に 3 角分割が起こり (b) のようになる。追加した頂点を選択し、把持・移動を行なって (c) のようにし、これを再び把持・移動すると (d) の立体形状ができる。



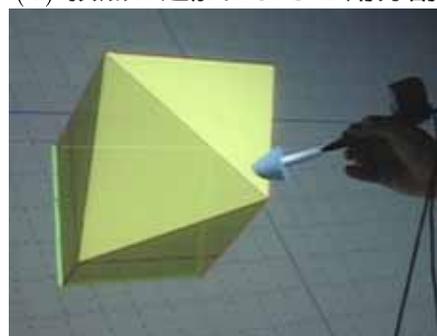
(a) 三角柱



(b) 頂点の追加による 3 角分割



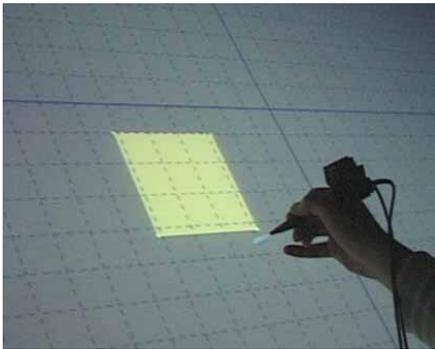
(c) 追加した頂点の移動



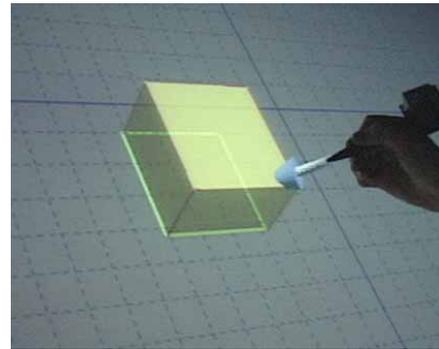
(d) (c) の押し出し体の作成

図 13 頂点の追加による面の 3 角分割

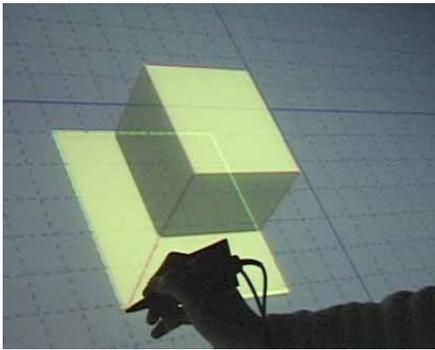
なお、“FixSec” を押した後、把持・移動を行わずに再び “FixSec” を押すと、同じ交差面が 2 度作られ、後で作られた交差面のみが編集対象となる。これを利用した立体形状の作成過程を図 14 に示す。まず、平面形状として長方形 (a) を作成し、把持・移動により四角柱 (b) を作成する。ここで、“FixSec” を 2 回連続して押すと、2 次元平面上に 2 つの交差面が作成される。このとき、操作対象となるのは後で作られた交差面のみであるので、これを拡大すると (c) のように 2 次元平面と平行な段をつくることができる。これを再び把持・移動すると (d) の立体形状ができる。



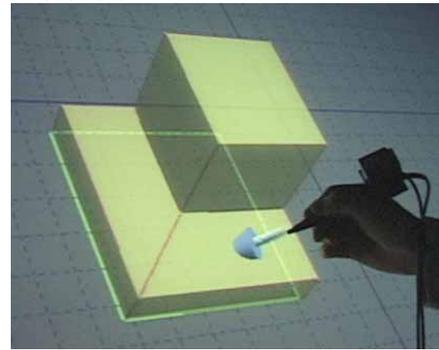
(a) “Box” による長方形の作成



(b) (a) の押し出し体の作成



(c) 交差面の拡大



(d) (c) の押し出し体の作成

図 14 “FixSec” の連続使用による立体形状の作成

3.5 オブジェクトの変形

本システムには、一旦作成したオブジェクトを変形する手段として、1) 2次元平面との交差面の押し出しによる変形、2) 頂点の編集による変形、3) 仮想的な磁力による変形、4) ブール演算による変形、の4つがある。これらの変形操作は、立体オブジェクトと平面オブジェクトのどちらに対しても適用可能である。また、標準状態である頂点の編集以外の変形操作は、ボタンにより選択・実行される。本節では、これらの変形手法を詳細に説明する。

3.5.1 交差面の押し出しによる変形

前節の押し出しによる立体形状の作成で説明した“FixSec”と“Close”を利用して、2次元平面との交差面を押し出すことでオブジェクトを変形させることができる。押し出しによる立体形状の作成における操作手順と異なるのは、平面形状を把持・移動することから始めるのではなく、ユーザが既存のオブジェクトを2次元平面と交差するように配置し、“FixSec”を押すことから始める点である。

オブジェクトを把持し、2次元平面と交差するように配置した状態で“FixSec”を押すと、オブジェクトは2次元平面で分断され、2次元平面より手前の領域が残される。また、オブジェクトを構成する面の1つが2次元平面と一致するように、オブジェクトを配置した場合は、“FixSec”を押すことでその面が除去される。この結果作成された2次元平面との交差面を、ユーザは平面形状と同様に編集して立体形状を変形させることができる。さらに再び把持すると、この作成された交差面が垂直下方向に押し出される。このように、“FixSec”により作成された交差面に編集を加え、把持・移動により押し出すという操作を繰り返し、最後に“Close”により交差面を閉じて変形操作を終了する。

例として、図15(a)～(f)に球体の底を押し出して台を付け足す過程を示す。はじめ(a)の状態にある球体を把持し、球体の底が2次元平面と交差するように配置して(b)の状態にする。ここで“FixSec”ボタンを押すことで交差面を作成し、把持・移動を行なうことで(c)のように交差面を押し出す。(c)の状態では“FixSec”を押して作成した交差面に拡大操作を行なって(d)のようにする。再び把持・移動を行なうことで拡大した交差面を押し出す。最後に“Close”を押すことで底面

が作成され、最終的に (f) に示すようなオブジェクトが生成される。

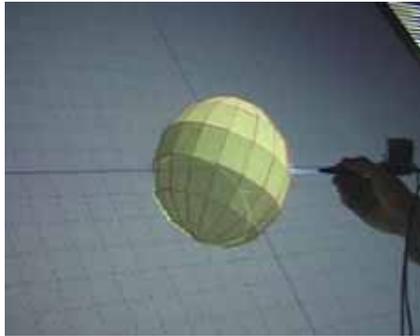
3.5.2 頂点の編集による変形

頂点の編集によるオブジェクトの変形には、3次元空間で行なう方法と、2次元平面で行なう方法の2つがある。3次元空間では、3次元空間中のオブジェクトの頂点を基本操作に従って直接移動させることで編集を行なう。2次元平面では、オブジェクトと平面との交差面に対して、頂点の追加、把持・移動、交差面の拡大・縮小などにより編集を行なう。この交差面は、3.3節で述べたように、その輪郭線が常に強調して表示されている。変形操作時には、この輪郭線を SW1 で直接押すことで、オブジェクトの各面を2次元平面で分断し、実際の操作対象とすることができる。

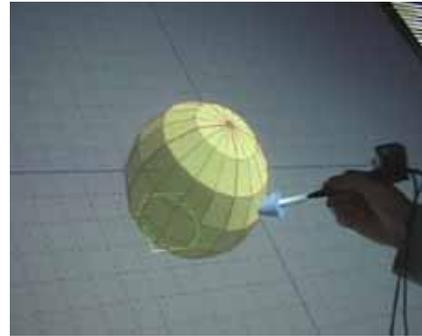
以上のような2次元平面での頂点編集による変形方法では、ユーザが望む交差面を簡単に得られることが重要である。しかし、基本操作で述べたような、オブジェクトの把持・移動操作では、大まかな位置と姿勢は容易に得られるが、微妙な調整は難しい。そのため、次の4種類の補助機能を持つボタンを提供している。

1. オブジェクトのローカル座標系の特定の軸を、作業空間の座標系の対応する軸と一致させるコマンドボタン “X-axe”, “Y-axe”, “Z-axe”
2. 選択した3つの頂点、または1つの面を2次元平面と一致させるコマンドボタン “EditPlane”
3. オブジェクトの動きを Z 軸方向に限定するソフトウェア制約の有効・無効を切替えるトグルボタン “Zonly”
4. 移動中のオブジェクトを選択状態にある頂点において2次元平面にスナップさせるソフトウェア制約の有効・無効を切替えるトグルボタン “Snap2D”

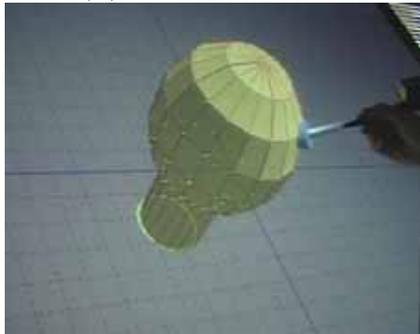
なお、オブジェクトのローカル座標系はオブジェクト作成操作の最後に “Close” を押した時点で作成される。作成された時点では作業空間の座標系とローカル座標系は一致しており、その後はオブジェクトの移動・回転に併せて変化する。



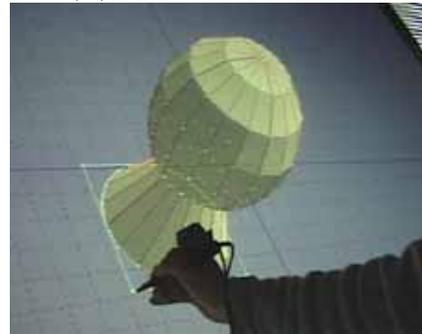
(a) 球体の初期状態



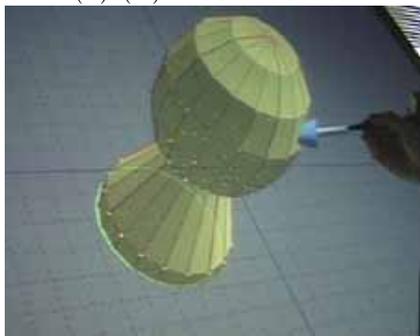
(b) 押し出す面の設定



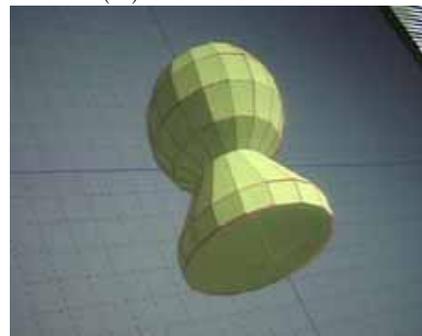
(c) (b) の押し出し体



(d) 交差面の拡大

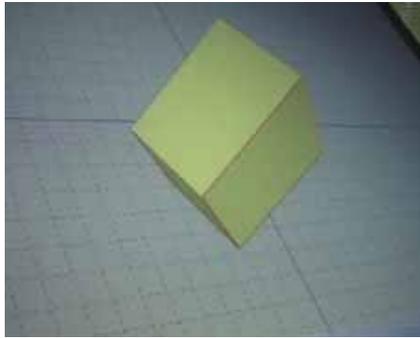


(e) (d) の押し出し体

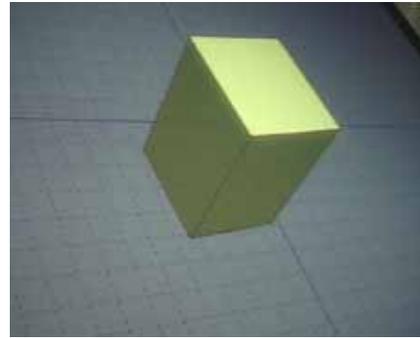


(f) (e) に “Close” を適用

図 15 交差面の押し出しによる立体オブジェクトの変形



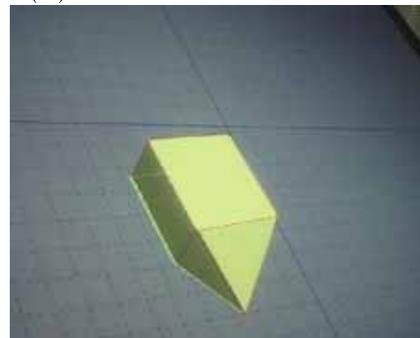
(a) 変形前のオブジェクト



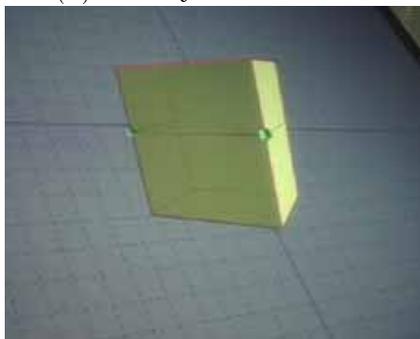
(b) “Z-axis” による姿勢変更



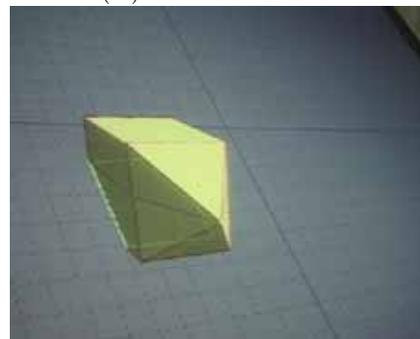
(c) “Zonly” による移動



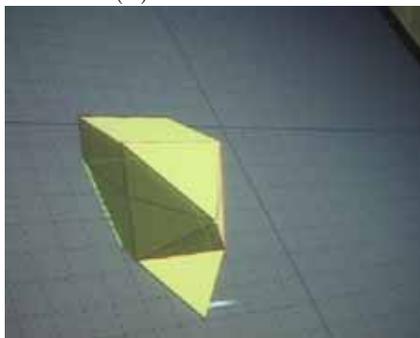
(d) 交差面の編集



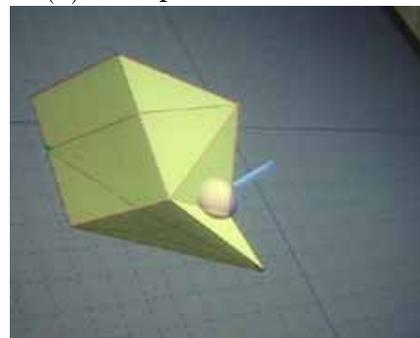
(e) 頂点の選択



(f) “Snap2D” による移動



(g) 交差面の編集



(h) 変形後のオブジェクト

図 16 2次元平面での頂点編集による立体オブジェクトの変形

これらの補助機能を利用しながら、2次元平面での頂点編集によりオブジェクトを変形させる過程を図 16(a) ~ (h) に示す。はじめ直方体は (a) のように傾いた状態で配置されている。この状態で “Z-axis” を押すと、直方体のローカル座標系の Z 軸と作業空間の Z 軸が一致するように直方体の姿勢が変化して、(b) のようになる。なお、この姿勢の変化は、500ms 程度の期間のアニメーションを伴って行なわれる。(b) の状態で “Zonly” ボタンを押して、直方体の移動を Z 軸方向に限定した状態で把持・移動を行ない、(c) のように配置した。ここで 2次元平面との交差面を SW1 で直接押すことで操作対象となった交差面に、頂点を追加・移動して (d) のようにした。追加した頂点の反対側にある 2つの頂点 ((e) の緑色の立方体) を選択する。“Zonly” を無効に、“Snap2D” を有効にした状態で、把持・移動を行ない、選択した 2頂点を 2次元平面にスナップさせて、(f) のように配置した。ここで再び 2次元平面との交差面を SW1 で押し、(g) のように頂点の追加・移動を行なった。この例では、最終的に直方体は (h) のような形状に変形された。

3.5.3 仮想的な磁力による変形

磁力による変形では、オブジェクトの一部を引っ張る・押し込むことで、表面に滑らかな出っ張りや窪みを付けることができる。ツールボタン “Magnet” を押して磁力による変形を指定すると、カーソルは矢印から、磁力の影響範囲を示す半透明の球形に変化する。ユーザは SW2 で球を大きく、SW3 で球を小さくして磁力の影響範囲を調節し、SW2 と SW3 を同時に押下しながらカーソルを移動させることで磁力による変形を実行する。球の内部に存在する頂点はカーソルの移動量に対して、球の中心からの距離に反比例した量で移動する。カーソルの移動によるオブジェクトの変形の様子は立体視により実時間で確認できる。

以上のような磁力による変形を用いて、例えば 4つの頂点からなる平面オブジェクトに滑らかな出っばりを付けるためには、予めその面を細分化してメッシュ状にしておく必要がある。そのために本システムでは、選択状態にある面をコマンドボタン “Division” により 3角分割することができる。ユーザは面を選択して “Division” を繰り返し適用することで、面を細かく分割することができる。

図 17に、“Division” を繰り返し適用してメッシュ状にした平面に対して、磁力

による変形を適用する様子を示す。まず、(a)のように変形させる領域が球の内部に入るようにする。このとき球の内部に存在する頂点は小さな立方体として描画される。(a)の状態では SW2 と SW3 を同時に押下しながら、カーソルを手前に動かすことで、(b)→(d)のように徐々に出っばりを付けて行くことができる。また、同じく(a)の状態からカーソルを奥に動かすことで、(c)→(e)のように徐々に窪みを付けて行くことができる。

3.5.4 ブール演算による変形

ブール演算では複数のオブジェクトを組み合わせて、複合形状を生成することができる。ブール演算の対象となるのは選択状態にあるオブジェクトと、その1つ前に選択状態にあったオブジェクトの2つである。本システムでは和・差・積の3種類のブール演算が行なえる。

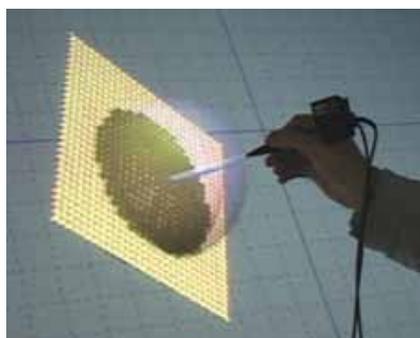
和: 2つのオブジェクトを結合して1つのオブジェクトに合成する。このとき重なり合う内側の領域は取り除かれる。

差: 1つ前に選択状態にあったオブジェクトから、現在選択状態にあるオブジェクトとの重なり合う領域を切り取る。同時に、現在選択状態にあるオブジェクトは消去される。

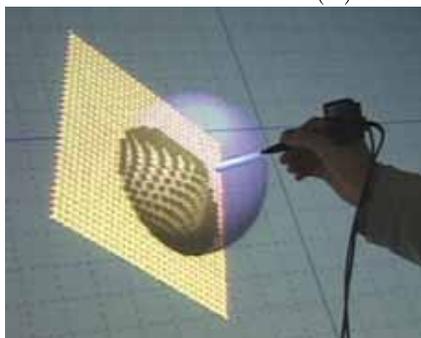
積: 2つのオブジェクトの重なり合う領域だけを残す。

ツールボタン“Boolean”を押すと、図18に示すように新たに3つのボタンが現れる。これらのボタンの名前と3種類のブール演算の関係を表4に示す。ユーザはオブジェクトを把持・移動して、適切な位置関係になるように配置する。なお、“Boolean”を押した状態では、ブール演算の対象となる2つのオブジェクト間の交差面の輪郭線が、常に強調して表示されている。最後に、3つのボタンのいずれかを押すことで対応するブール演算を実行する。

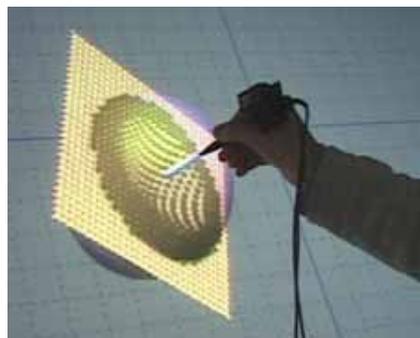
具体例を図19(a)～(e)に示す。まず、円筒形と直方体をそれぞれ選択、把持・移動させて(a)のように配置し、“Boolean”ボタンを押す。この工程はオブジェクトを適切な位置関係に配置すると同時に、2つのオブジェクトを選択することで、それらをブール演算の対象として指定する役割も持つ。



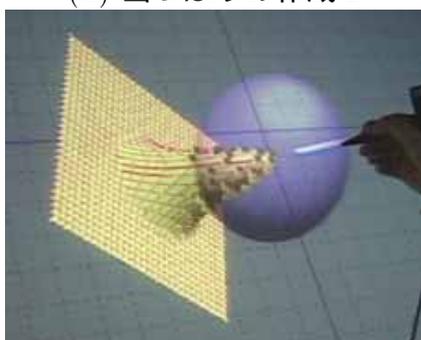
(a) 変形前の平面



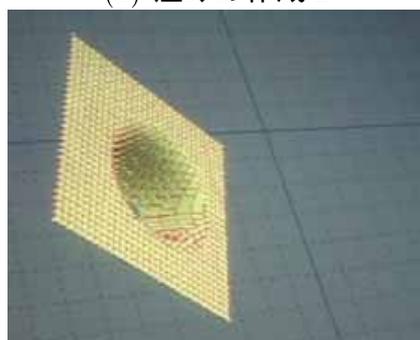
(b) 出っぱりの作成 1



(c) 窪みの作成 1



(d) 出っぱりの作成 2



(e) 窪みの作成 2

図 17 磁力による平面オブジェクトの変形

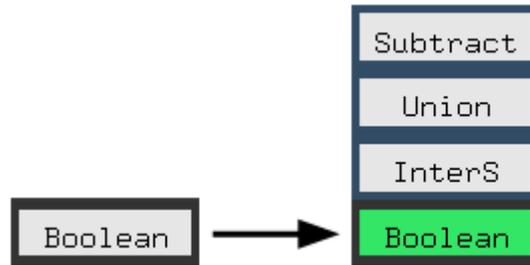


図 18 “Boolean” ボタンと構成要素

表 4 3つのブール演算とボタン名

Subtract	差のブール演算を実行
Union	和のブール演算を実行
InterS	積のブール演算を実行

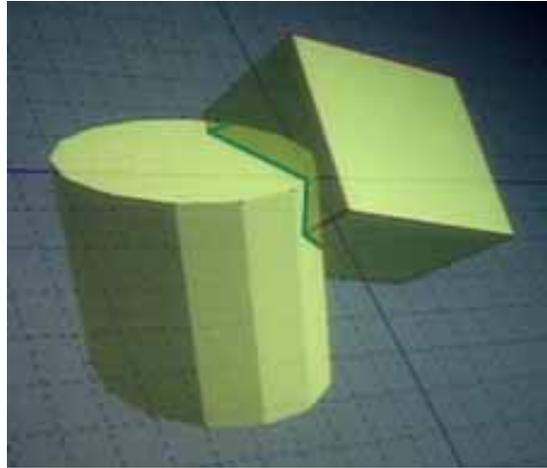
(a) の状態で “Subtract” ボタンを押して差のブール演算を実行すると、円筒形から現在選択状態にある直方体が引かれて (b) のようになる。逆に円筒形を選択状態にしてから “Subtract” ボタンを押すと、(c) のように直方体から円筒形を引くことができる。

一方、(a) の状態で “Union” ボタンを押して和のブール演算を実行することで、(d) のように 2 つを結合させて 1 つのオブジェクトにすることができる。

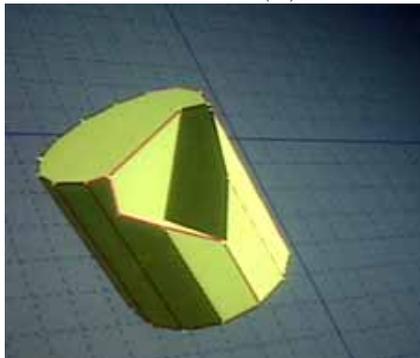
また、(a) の状態で “InterS” ボタンにより積のブール演算を実行することで、(e) のように両者の重なり合う領域を残すことができる。なお、(b)、(c) ではブール演算による変形箇所が見えるように、それぞれオブジェクトを移動させている。

3.6 モデリング例

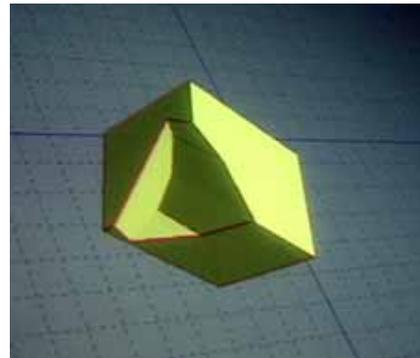
本節では、試作システムを用いたモデリング例を通して、本システムの有効性を示す。



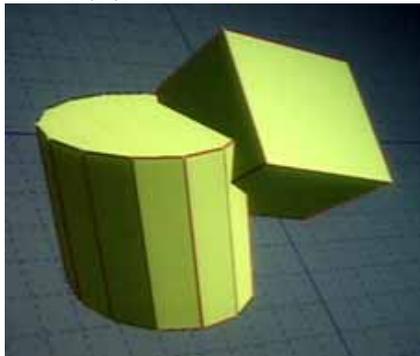
(a) ブール演算前のオブジェクト



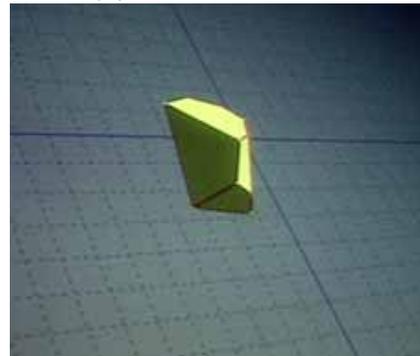
(b) 差の演算結果 1



(c) 差の演算結果 2



(d) 和の演算結果



(e) 積の演算結果

図 19 ブール演算による立体オブジェクトの変形

リンゴのモデリング

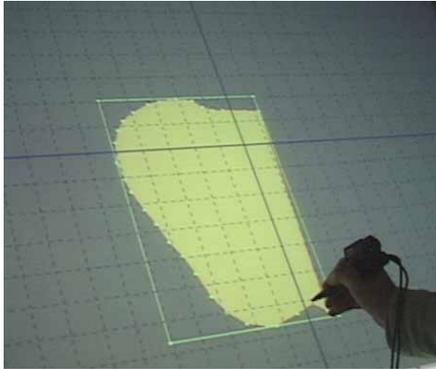
本システムを用いたモデリングの例として、りんごをモデリングした際の工程を図 20(a) ~ (f) に示す。また、作成した形状データを市販のレンダリングソフトで描画した結果を図 21 に示す。

りんごは実、茎、葉、虫食い痕の 4 つで構成されている。まず、図 (a) ~ (f) に沿って順に各工程の概要を示す。

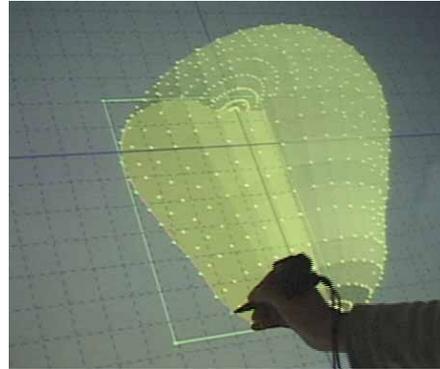
- a) 2次元平面で“Disc”により、りんごの断面の平面形状を作成する。
- b) a) の平面形状から“Lathe”により回転体を作成し、側面の数を“ $\rightarrow\gg$ ”により増加させる。これをりんごの実とする。
- c) b) のりんごの実に磁力による変形 (“Magnet”) を用いて凹凸をつける。
- d) 2次元平面で葉の形をした平面オブジェクトを作成し、“Division”により 3 角分割を繰り返し適用した後、磁力による変形で歪曲させる。
- e) 2次元平面で作成した正円を把持・移動で手前に引き出し、“FixSec”により交差面を作成する。作成した交差面を編集し、再び把持・移動で手前に引き出す。この操作を繰り返し、最後に“Close”することでりんごの茎を作成する。
- f) まず、“Circ”による正円を“Lathe”で回転させた球体に、“Magnet”を用いて適当な凹凸を与え、これを虫食いの痕をつけるためのオブジェクトとする。次に、実、葉、茎のそれぞれの大きさを、“ $\ll-$ ”、“ $\gg-$ ”により調節した後、和のブール演算により結合させる。最後に、この結合したオブジェクトと、虫食い痕のオブジェクトとの間で差のブール演算を適用することで、図 21 のりんごが完成する。

スタンドライトのモデリング

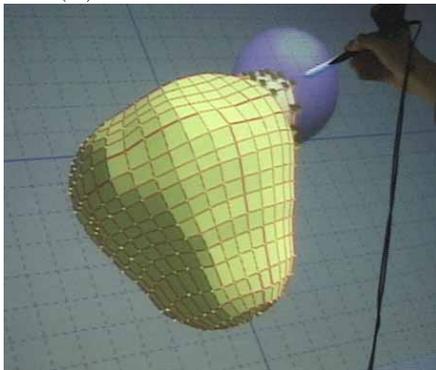
本システムを用いたモデリングの例として、スタンドライトをモデリングした際の工程を図 22(a) ~ (h) に示す。また、作成した形状データを市販のレンダリングソフトで描画した結果を図 23 に示す。



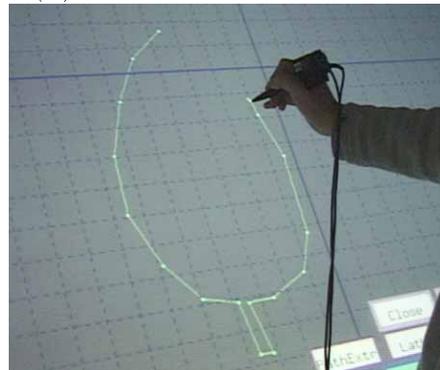
(a) りんごの断面の作成



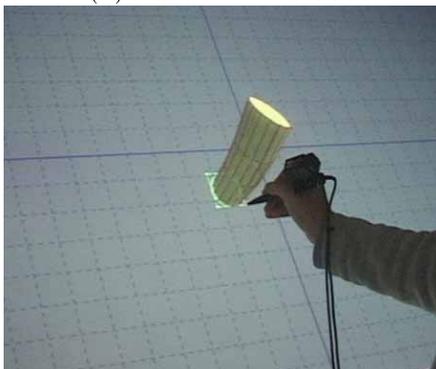
(b) “Lathe” による実の作成



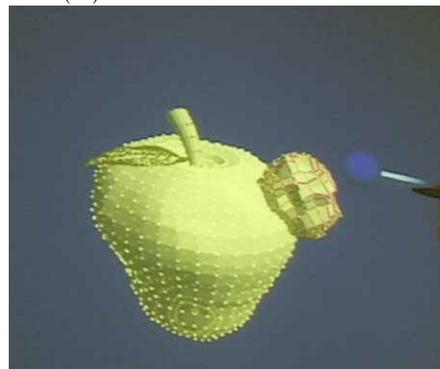
(c) 磁力による変形



(d) 葉の平面形状の作成



(e) 茎の作成



(f) 適切な位置に配置

図 20 りんごのモデリング過程

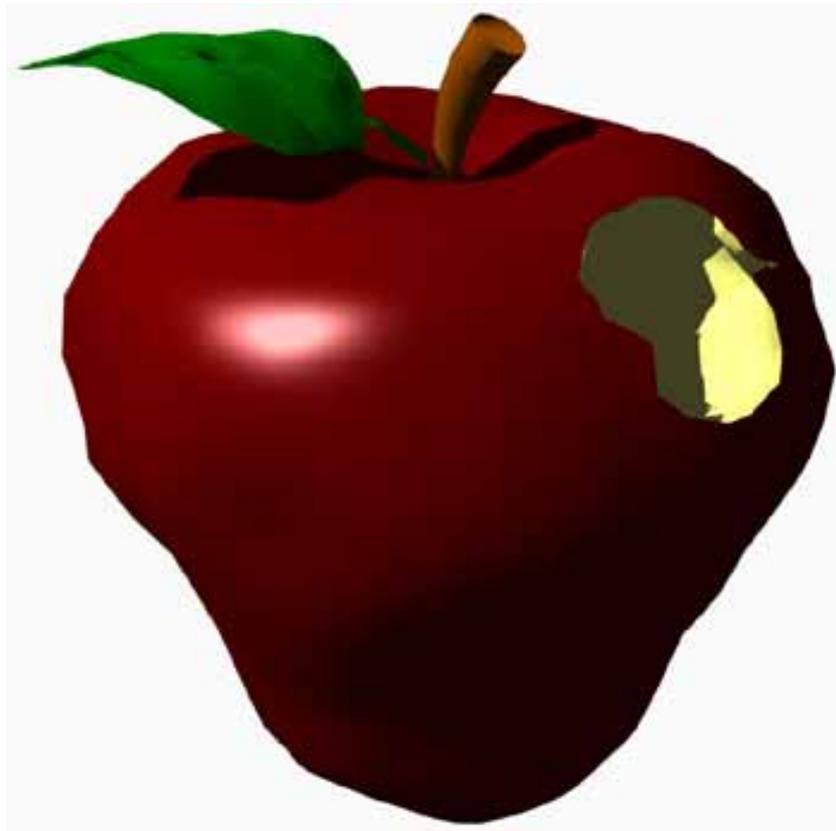


図 21 りんごのレンダリング結果

スタンドライトは、台、首、傘、電球の4つから構成されている。まず、図(a)～(h)に沿って順に各工程の概要を示す。

- a) 2次元平面で”Circ”により正円を作成し、それを”Lathe”で回転させて球体を作る。磁力による変形を使用して球体の片側を引き出す。これを電球とする。

2次元平面で正円を作成し、それを”Zonly”を有効にした状態で、把持・移動させて少しだけ手前に引き出す。次に、”FixSec”により交差面を作成し、それを拡大する。再び把持・移動させて少しだけ手前に引き出し、”FixSec”を2回連続して使用した後、交差面を拡大させ、最初に作った正円(上面)と平行な段を作る。最後にもう一度、手前に引き出し、”Close”を押す。これをスタンドの台とする。

- c) (b)の上面を構成する頂点から3つを選択し、”EditPlane”により上面を2次元平面と一致させる。次に、2次元平面と一致した上面に頂点を追加し、把持・移動を行なって、スタンドの台の一部に出っぱりをつける。なお、(c)では2次元平面を非表示にしている。
- d) 2次元平面で正円を作成し、”PathExtr”で経路を指定して掃引体を作成する。これをスタンドの首とする。
- e) d)で作成したスタンドの首を、経路上の頂点を選択、把持・移動させて曲げる。さらに、経路に頂点を追加し、それを移動させて曲がり具合を調整する。
- f) 2次元平面を非表示にした状態で、c)のスタンドの台とe)の首を適当な位置に配置して、和のブール演算を行なう。
- g) 2次元平面を表示に戻し、その上で”Disc”により自由形状を作成する。これを”Lathe”で回転させて傘を作る。
- h) 2次元平面を非表示にした状態で、f)の和のブール演算の結果と、g)の傘を適切な位置関係に配置する。両者を見比べながら、傘の大きさを、”->>”、

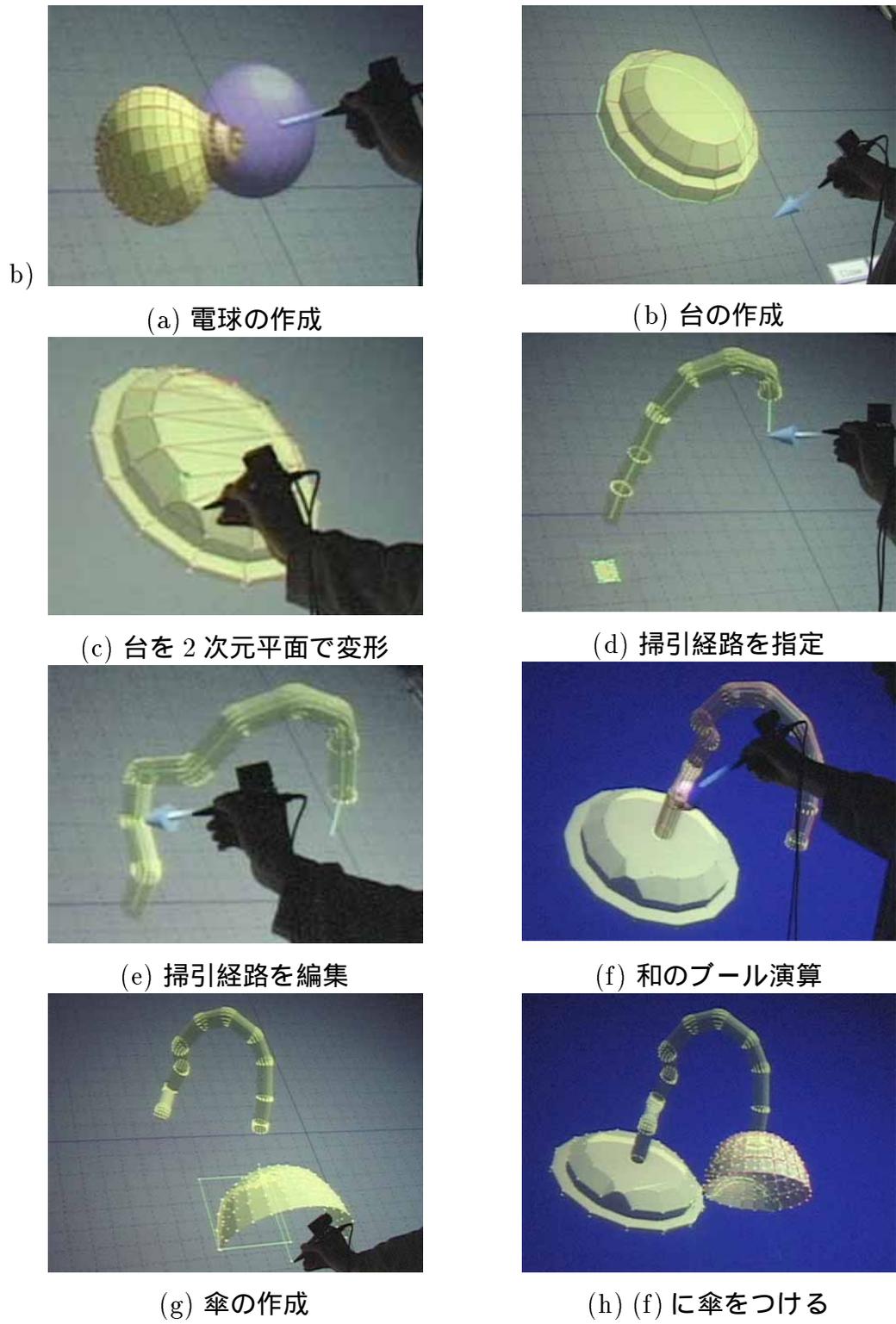


図 22 スタンドライトのモデリング過程



図 23 スタンドライトのレンダリング結果

“<<-”により拡大・縮小して調節する。最後に、電球を適切な位置に配置して、和のブール演算を適用することで、スタンドライトが完成する。

4. 検証と考察

本章では、3章で述べた試作システムが、2.4節で述べた本研究の方針を実現しているかを考察する。2.4節で述べた本研究の方針を要約して以下に示す。

1. 2次元平面でのモデリングの利点を持つ2次元平面を3次元空間中に実現する。
2. 2次元平面でのモデリングと3次元空間でのモデリングの双方の持つ利点を同時に利用しながらモデリングを行なえるようにする。これには次の3つの要件を満たす必要がある。
 - (a) 2次元平面と3次元空間を単一の作業空間として実現し、そこに両空間を共存させる。
 - (b) 両空間の操作を単一の操作デバイスにより行なえるようにし、ユーザの負担を軽減させる。
 - (c) 個々のモデリング操作をその性質に適した空間に実装する。

以下、それぞれの方針について、前章で示したモデリング例での経験を引用しながら検証を行なう。

2次元平面でのモデリングの利点を有する2次元平面の実現

本システムでは、傾斜型ディスプレイの表面を実平面として利用することで、物理的な制約による2次元平面を実現している。この物理的な制約により、2次元平面上でユーザが同時に制御する自由度は2と少なくなり、容易に入力を制御することができた。

また、次の2つの理由により安定した操作を行なうことができた。

- ユーザが2次元平面上での操作に用いるペン型デバイスは、傾斜型ディスプレイの表面と直接接触することで摩擦力や抗力をもたらす。
- 86インチの傾斜型ディスプレイ表面の作り出す2次元平面は十分広く、また45度傾斜した状態で設置されているため、ユーザは製図台の上で図面を作成するように自然な体勢で作業できる。

結果として、前章で示したりんごをモデリングにおけるりんごの断面、葉の平面形状、スタンドライトの傘の断面を作成する際に、複雑な平面形状を意図した通りに作ることができた。また、一旦作成した平面形状に対しての細かな修正も容易に行なえた。

以上のように、2.1 節で述べた、1) 入力の制御が容易である、2) 安定した操作が行なえる、という2次元平面でのモデリングの2つの利点を、本システムにおける2次元平面も同じく持つことができた。

両空間の共存による単一の作業空間でのモデリング

ここでは、本システムが2次元平面と3次元空間を共存させることで、物理的な移動や切替え操作を必要とせず、単一の作業空間でモデリングを行なうことができたかどうかを確認する。

本システムでは、2次元平面を実現するのに利用した傾斜型ディスプレイを用いて3次元空間を実現している。ユーザは、傾斜型ディスプレイに交互に提示される左右眼画像を超音波センサを取り付けた液晶シャッター眼鏡を通して観察することで、運動立体視を伴う両眼立体視をすることができる。これにより、2次元平面でのモデリングと3次元空間でのモデリングを、ユーザは物理的に移動することなく、単一の作業空間で行なうことができた。

また、例えばりんごの茎を作成する過程や、スタンドライトの台に出っぱりを作る過程では、オブジェクトと2次元平面との交差面に対して操作を行なった。このように、2次元平面でのモデリングを、オブジェクトと2次元平面の交差面に対する操作として導入することで、作業空間の切替え操作を必要としない、ソフトウェア的な意味においても両者が共存した環境を実現できた。

一方、作業空間としての3次元空間が狭い点が問題となった。本システムでの3次元空間は、ユーザの視点と傾斜型ディスプレイの4角を結んだ錐体に限定されているため、作業空間としては十分な広さを持つことができなかった。そのため、例えばスタンドライトの首の部分を作成する過程では、作業空間の広さを考慮しながら経路を指定して行く必要があった。

単一の操作デバイスの導入によるユーザの負担の軽減

ここでは、単一の操作デバイスを導入することで、ユーザに物理的な負担や認知的な負担を掛けなくすることができたかを確認する。

本システムでは、2次元平面でのモデリングと3次元平面でのモデリングに共通して用いる操作デバイスとして、ライトペンと超音波センサ、ジャイロセンサを組み合わせたペン型デバイスを導入した。このペン型デバイスを用いた操作では、2次元平面で用いるデバイスと、3次元空間で用いるデバイスを持ち変える必要が無かった。従って、持ち変えに伴う物理的な負荷を感じずに、両空間でのモデリングを行なうことができた。

また、すべての操作をこのペン型デバイスで行なえるため、各操作毎に使用するデバイスを判断に伴う認知的な負荷を感じることなく、モデリングを行なうことができた。さらに、このペン型デバイスでは、3次元空間でのモデリングにおいて仮想物体に対して直接的に操作を行なうのと同じ感覚で、2次元平面に表示されている交差面に対しても直接的に操作することができた。このような、首尾一貫した操作方法によりユーザは、どちらの空間でモデリングを行なっているかに注意を払う必要は無かった。

一方、ペン型デバイスの形状に関してはいくつか問題があった。まず、ジャイロセンサが重たいため安定感が悪く持ちにくいものであり、長時間の使用には不向きであった。また、把持・移動操作は、ペン型デバイスの側部にあるスイッチを押下しながら行なうため、そのペン型デバイスを軸とした回転は困難であった。

また、ペン型デバイスの位置計測に用いた超音波センサには、次のような問題があった。ペン型デバイスの位置は、取り付けた超音波センサからの超音波を、天井に設置した受信部が受け取ることで計測される。そのため、超音波センサを常に上を向けて操作する必要があり、ユーザはその点に注意を払う必要があった。結果として、操作の自由度が限定されてしまった。

適切な空間でのモデリング操作の実装

ここでは、モデリング操作をその性質に応じて適切な空間に実装できたかを

確認する。

本システムで2次元平面に実装したモデリング操作は、2次元平面でのモデリングの利点である、1) 入力の制御が容易である、2) 安定性が高い、の2つの利点により容易に行なえるようになる操作である。例えば、前章でのりんごを例にとると、りんごの断面の2次元平面上での作成は、これら2つの利点により容易に行なうことができた。また、ボタンやスライダに対する操作も、正確に意図した通り行なうことができた。

一方、本システムで3次元空間に実装したモデリング操作は、3次元空間でのモデリングの利点である、1) 視認の容易性、2) 操作の直感性、3) 高い対話性、のいずれか、あるいはすべてにより容易に行なえるようになる操作である。例として、個別に作成したりんごの実、葉、茎、虫食い痕を、適切な位置関係に配置してブール演算を適用する工程を考える。ユーザは、視認の容易性によりオブジェクト間の位置関係を素早く判断し、その判断に従って効率良く直感的に6自由度の移動を行なえ、さらにその結果を即座に確認できた。同じ操作を2次元平面でのモデリングで行なうと、一度に2自由度の移動しかできないため、細かな移動操作を組み合わせることで繰り返し行なう必要があり、非常に手間のかかる作業である。

以上のように、どちらか一方の空間では難しい作業が、もう片方の空間で容易に行なうことができたという点から、適切な空間に実装できていることが確認できた。なお、以上の考察は筆者による主観的な評価によるため、今後これを裏付けるための評価実験を行なう必要がある。

5. おわりに

本論文では、2次元平面と3次元空間を組み合わせたモデリング環境を提案した。提案環境は2次元平面でのモデリングの持つ、入力制御が容易である、安定した操作が行なえる、という2つの利点と、3次元空間でのモデリングが持つ、立体形状の視認が容易である、操作が直感的に行なえる、対話性の高いモデリングが行なえる、という3つの利点を、単一の作業空間に持ち合わせる。

本研究では、そのような提案環境を実現するために、1) 2次元平面でのモデリングの利点と同等の操作性を有する2次元平面を3次元空間中に実現し、2) 両空間での利点を活かしたモデリングを可能にする手法を提案した。さらに2)を実現するために満たすべき要点として、a) 両空間が共存した単一の作業空間でのモデリング、b) 単一の操作デバイスの導入によるユーザの負担の軽減、c) 適切な空間でのモデリング操作の実装、の3点を示した。

以上の方針に従って開発した試作システムでは、傾斜型ディスプレイの表面を実平面として利用することで、高い操作性を持つ2次元平面を実現し、また、同じ傾斜型ディスプレイを用いた時分割立体表示により3次元空間を実現した。これにより、両空間を物理的に単一の作業空間とすることができた。また、2次元平面での操作を3次元空間に立体視表示されたオブジェクトと、2次元平面との交差面に対する操作として導入することで、切替え操作を必要とせずに両空間でのモデリングを行なえるようにした。

さらに、ライトペンと位置・姿勢計測装置を組み合わせたペン型デバイスを、両空間で共通して使用する操作デバイスとして用いることで、ユーザに物理的、認知的な負荷を掛けずにモデリングを行なえるようにした。以上のような試作システムに、モデリング操作をその性質に応じて適当な空間に実装した。提案環境の有効性は、試作システムでのモデリング例を通して確認することができた。本システムに対する考察により明らかになった問題点としては、3次元空間の狭さ、ペン型デバイスの扱いにくさ等が挙げられる。

今後の課題としては、3次元空間の狭さを解決するために、傾斜型ディスプレイの表面より奥の領域を有効に利用する手法の開発が挙げられる。一般に複雑なオブジェクトを作るには、それを細かな部品に分けて作成して行く。本システムで

は直接操作が行なえる表面より手前の領域が狭いため、これらの部品を配置しておくことで作業空間が混雑してしまう。表面より奥の領域では直接操作することができないが、これらの部品を格納しておくには有効な空間であると考えられる。

本システムにおいて作成できる立体形状は限られたものであった。モデリング環境において作成できる立体形状は、その環境が実装しているモデリング操作の種類に大きく依存する。従って、本システムに更にモデリング操作を実装することで、多様な立体形状を作成することができると考えられる。

謝辞

本研究の全過程を通して、懇切なる御指導、御鞭撻を賜わった、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、有益な御助言を賜わった、像情報処理学講座 千原 國宏 教授に暑く御礼申し上げます。

本研究の全過程を通して、懇切丁寧なる御指導、御鞭撻を賜わった、ソフトウェア基礎講座 竹村 治雄 助教授に心より感謝の意を表します。

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 助手に深く感謝いたします。

物心両面において常に暖かい御支援を頂いたソフトウェア基礎講座 博士後期課程の 松宮 雅俊 氏、神原 誠之 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、日頃より暖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 女史に心より感謝いたします。

参考文献

- [1] *LightWave 3D Version 5.5 Reference Manual*, NewTek, Inc, 1997.
- [2] *3D Studio MAX RELEASE 3 Reference Manual*, Autodesk, Inc, 1999.
- [3] 鳥谷部 真: form・Z RenderZone 徹底解説, 株式会社建設知識, 1997.
- [4] *AutoCAD Release 14 ユーザガイド*, オートデスク株式会社, 1997.
- [5] Liang, J. and Green, M.: “JDCAD: A Highly Interactive 3D Modeling System,” *Proc. 3rd Internatioinal Conf. on CAD and Computer Graphics*, pp.217–222, 1993.
- [6] Butterworth, J., Davidson, A., Hensch, S. and Olano, T. M.: “3DM: A Three Dimensional Modeler Using a Head-Mounted Display,” *Proc. ACM Sympo. on Interactive 3D Graphics (GI '92)*, pp.135–139, 1992.
- [7] 清川 清, 竹村 治雄, 片山 喜章, 岩佐 英彦, 横矢 直和: “両手操作を用いた仮想物体モデラ VLEGO”, *電子情報通信学会論文誌 (A)*, Vol.J80-A, No.9, pp.1517-1526, 1997.
- [8] Bowman, D. A. and Hodges, L. F.: “User Interface Constraints for Immersive Virtual Environment Applications,” *Graphics, Visualization and Usability Center Technical Report*, GIT-GVU-95-26, 1995.
- [9] Stoakley, R., Conway, M. and Pausch, R.: “Virtual Reality on a WIM: Interactive Worlds in Miniature,” *Proc. ACM CHI'95 Conf. on Human Factors in Computing Systems*, pp.265–272, 1995.
- [10] Mine, M.: “Working in a Virtual World: Interaction Techniques Used in the Chapel Hill Immersive Modeling Program,” *UNC Chapel Hill Computer Science Technical Report TR96-029*, 1996.

- [11] Deering, M. F.: “The Holosketch VR Sketching System,” *Communications of the ACM*, Vol.39, No.5, pp.54–61, 1996.
- [12] Hill II, L. C., Chan, C. and Cruz-Neira, C.: “Computer Aided Design in an Immersive Environment: The Virtual Architectural Design Tool (VADeT),” <http://www.icemt.iastate.edu/~lchill/cpe575paper/cpe575.html>, 1997.
- [13] Hinckley, K., Rausch, R., Goble, J. C. and Kassell, N. F.: “A Survey of Design Issues in Spatial input,” *Proc. ACM Sympo. on User Interface Software and Technology*, pp.213–222, 1994.
- [14] 平田 幸広, 水口 武尚, 佐藤 誠, 川原田 弘: “組立操作のための仮想作業空間”, 電子情報通信学会論文誌 (D-II), Vol.J76-D-II, No.8, pp.1788–1795, 1993.
- [15] 岩田 洋夫, 早川 敬介: “仮想物体の把持操作を行なうグローブ型フォースディスプレイの開発”, 第 11 回ヒューマンインタフェースシンポジウム論文集, pp.395–400, 1995.
- [16] 野間 春生, 宮里 勉: “協調仮想物体操作のための仮想拘束感覚の提示手法”, 日本バーチャルリアリティ学会第 2 回大会論文集, pp.284–287, 1997.
- [17] 吉森 勇人, 松宮 雅俊, 竹村 治雄, 横矢 直和: “2 次元/3 次元空間のシームレスな融合によるモデリング環境”, 日本バーチャルリアリティ学会第 4 回大会論文集, pp.353–354, 1999.
- [18] 吉森 勇人, 松宮 雅俊, 岩佐 英彦, 竹村 治雄, 横矢 直和: “2 次元/3 次元空間の組み合わせによるモデリング環境”, 電子情報通信学会技術研究報告 (マルチメディア・仮想環境基礎研究会), 2000(掲載予定).
- [19] Foley, J., van Dam, A., Feiner, S. and Hughes, J.: *Computer Graphics*, Addison-Wesley Publishing Company, Inc, 1990.
- [20] Mine, M.: “A Review and Analysis of Through-the Window Computer-Aided Modeling Systems,” *UNC Chapel Hill Computer Science Technical Report TR94-070*, 1994.

- [21] 竹村 治雄, 伴野 明: “時分割両眼立体視を用いた 3 次元指示入力操作の評価”, 電子情報通信学会論文誌 (D-I), Vol.J74-D-I, 1, pp.33–38, 1991.
- [22] 竹村 治雄, 北村 喜章, シャネゾン・アラン, 岸野 文郎: “仮想現実環境における物体配置タスクの一操作補助手法”, テレビジョン学会誌, Vol.48, No.10, pp.1312–1317, 1994.
- [23] Forsberg, A. S., LaViola Jr, J. J. and Zeleznik, R. C.: “ErgoDesk: A Framework for Two- and Three-Dimensional Interaction at the ActiveDesk,” *Proc. 2nd International Immersive Projection Technology Workshop*, 1998.
- [24] Zeleznik, R. C., Herndon, K. P. and Hughes, J. F.: “SKETCH: An Interface for Sketching 3D Scenes,” *Proc. SIGGRAPH'96*, pp.163–170, 1996.
- [25] Schmandt, C. M.: “Spatial Input/Display Correspondence in a Stereoscopic Computer Graphics Workstation,” *Proc. ACM SIGGRAPH '83*, pp.253–262, 1983.
- [26] Hinckley, K., Goble, J. C., Pausch, R. and Kassell, N. F.: “New Applications for the Touchscreen in 2D and 3D Medical Imaging Workstations,” *Proc. SPIE Medical Imaging '95, Image Capture, Formatting, and Display*, 1995.
- [27] Ellis, M. A. and Stroustrup, B.: *The Annotated C++ Reference Manual*, Addison-Wesley Publishing Company, Inc, 1992.
- [28] Woo, M., Neider, J. and Davis, T.: *OpenGL Programming Guide Second Edition*, Addison Wesley Developers Press, 1997.
- [29] 世利 至彦, 大隈 隆史, 清川 清, 竹村 治雄, 横矢 直和: “汎用 3 次元ユーザインタフェースツールキットの実装”, 電子情報通信学会技術研究報告 (電子ディスプレイ), Vol.97, No.526, pp.95–100, 1998.
- [30] Seidel, R.: “A simple and Fast Randomized Algorithm for Computing Trapezoidal Decompositions and for Triangulating Polygons,” *Computational Geometry Theory & Applications*, Vol.1, No.1, pp.51–64, 1991.

[31] *StereoGraphics Developers' Handbook*, StereoGraphics Corporation, 1997.