

NAIST-IS-MT9751048

## 修士論文

# 仮想環境没入型モデラにおける アニメーション記述手法の提案と実装

財満 博昭

1999年2月12日

奈良先端科学技術大学院大学  
情報科学研究科 情報システム学専攻

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

財満 博昭

指導教官： 横矢 直和 教授  
千原 國宏 教授  
竹村 治雄 助教授

# 仮想環境没入型モデラにおける アニメーション記述手法の提案と実装\*

財満 博昭

## 内容梗概

計算機上で3次元アニメーションを記述する場合、従来は、特殊な言語によって記述する手法や2次元インタフェースを用いるため、操作が煩雑で分かりにくいという問題があった。そこで本研究では、直感的な3次元形状生成が可能な仮想環境没入型モデラにおいて、アニメーションを記述する場合に必要な機能について考察し、対話的にアニメーションを記述する手法を提案する。ここでいうアニメーションは、恒常的に動き続ける**静的アニメーション**とユーザのアクションに応答して起動される**対話的アニメーション**を含む。

本手法では、アニメーションの振り方やスピードなどの属性を、その属性の性質を示す形を持ったプリミティブ（**アニメーションプリミティブ**）として提供する。また、対話的アニメーションの記述に欠かせないアニメーションの起動条件などの属性についても、その属性を持ったプリミティブ（**イベントプリミティブ**）を提供する。本手法の特長は次の3点である。1) アニメーションの属性がプリミティブの形から直感的に把握できる、2) プリミティブを直接操作し、対話的にその属性を変更できる、3) 仮想物体のアニメーション記述は、仮想物体とプリミティブの簡単な組み合わせ操作で実現でき、複雑な操作の習熟を必要としない。本論文では、提案手法について詳しく述べた後、それを実装した試作システムを通して提案手法の有効性を考察する。

---

\*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-MT9751048, 1999年2月12日.

## キーワード

仮想現実感, 3次元ユーザインタフェース, アニメーション, インタラクション

# Interactive Design of Animation within an Immersive Modeling System\*

Hiroaki Zaima

## Abstract

In this paper, proposed is a design method of interactive animation within an immersive modeler that makes 3-D design intuitive. Interactive animation can be considered to have two attributes: the motion patterns and the triggers. The method provides two kinds of virtual objects corresponding to these attributes: animation and event primitives. The method has the following three advantages. First, a user can understand the types and attributes of the motion patterns visually from the shapes of these primitives. Second, a user can manipulate these primitives directly and can change the attributes of animation interactively. Third, virtual 3-D objects including various animation attributes can be created by simply combining shape, animation and event primitives, to design interactive animation. A prototype system has shown the effectiveness of the method for creating simple interactive animation in a short time.

## Keywords:

virtual reality, 3-D user interface, animation, interaction

---

\*Master's Thesis, Department of Information Systems, Graduate School of Information Science, Nara Institute of Science and Technology, NAIST-IS-MT9751048, February 12, 1999.

# 目次

1. はじめに	1
2. 関連研究:アニメーションの記述と没入型モデラ	4
2.1 アニメーションの記述	4
2.1.1 記述言語による手法	4
2.1.2 2次元 GUI による対話的な記述手法	5
2.2 没入型モデラ	7
2.2.1 没入型モデラの特徴	7
2.2.2 既存の没入型モデラ	9
3. 没入型モデラにおけるアニメーション記述に関する検討	11
3.1 仮想物体のアニメーション	11
3.1.1 静的アニメーション	11
3.1.2 対話的アニメーション	12
3.2 アニメーションの記述手法	13
3.2.1 静的アニメーションの記述手法	15
3.2.2 対話的アニメーションの記述手法	17
3.3 アニメーションプリミティブとイベントプリミティブ	20
3.3.1 モーションプリミティブ	20
3.3.2 パスプリミティブ	20
3.3.3 イベントプリミティブ	22
4. アニメーション機能を備えた没入型モデラの試作	24
4.1 システムの設計方針	24
4.2 システム概要	25
4.3 基本操作	28
4.4 組み合わせ操作	31
4.5 デザイン例	33

5. 考察	37
5.1 提案手法の妥当性の検証 . . . . .	37
5.2 アニメーションの多様化 . . . . .	39
6. むすび	42
謝辞	44
参考文献	45

## 目 次

1	3DCG ソフトの操作画面例 . . . . .	6
2	SKETCH の操作例 . . . . .	7
3	静的アニメーションの制作例 . . . . .	12
4	アニメーションの属性 . . . . .	14
5	アニメーションプリミティブの例 . . . . .	16
6	対話的アニメーションの例 . . . . .	18
7	イベントの連鎖 . . . . .	19
8	モーションプリミティブ . . . . .	21
9	パスプリミティブの例 . . . . .	22
10	イベントプリミティブ . . . . .	23
11	ハードウェア構成 . . . . .	25
12	作業の様子 . . . . .	26
13	試作システムにおけるシーングラフの例 . . . . .	27
14	プリミティブボックス . . . . .	30
15	イレイズボックス . . . . .	30
16	片手組み合わせ操作 . . . . .	32
17	両手組み合わせ操作 . . . . .	32
18	天体モデル . . . . .	33
19	びっくり箱 . . . . .	34
20	風車と飛行船 . . . . .	35
21	親子関係とリンク関係の例 . . . . .	40



## 1. はじめに

近年，計算機の処理能力の向上などに伴い，アニメーションを含んだ仮想世界やユーザが仮想物体に対して様々なアクションを起こすことのできる対話的な仮想世界を構築できるようになってきた．このような仮想世界では，例えば，常に回り続ける風車のような**静的アニメーション**や触れると人形の飛び出すびっくり箱，近付くと開く自動ドア，あるいは各階の呼び出しボタンによって昇降するエレベータなどの**対話的アニメーション**が含まれる．ここでいう静的アニメーションとは，ユーザのアクションによって起動されることのない恒常的なアニメーションを指し，動きのパターンを決定する個々の**アニメーション要素**から成る．また，対話的アニメーションとは，ユーザのアクションに応答して起動されるアニメーションの総称であり，個々のアニメーション要素と**起動イベント**から構成される．起動イベントとは，アニメーションを起動するよう関係付けられた仮想空間内の状態の変化を指す．

従来，このようなアニメーションを含んだ仮想空間を構築する場合，プログラム言語やスクリプト言語を用いて記述する手法 [1, 2, 3, 4, 5] や，3次元コンピュータグラフィックス (CG) ソフトなどを用いる手法 [6, 7, 8, 9] が採られてきた．しかし，記述言語による手法は特定の言語に習熟する必要があり，3次元CGソフトのようにCRTやマウスなどの2次元の入出力装置を用いる手法は操作が煩雑になりやすいという問題があった．

一方，直感的な操作により習熟を要せずに3次元世界を構築する手法として，仮想環境に没入しながら仮想物体の生成を行なう**仮想環境没入型モデラ**（没入型モデラ）が注目されている [10, 11, 12, 13, 14]．没入型モデラには，形状の直感的な把握や操作，対話的な仮想物体生成などの多くの利点があり，従来の3面図を編集して3次元形状を決定するようなモデラに比べて直感的で，分かりやすいモデリングが可能である．一般に没入型モデラには，3次元入力装置の精度，人間の空間指示能力の低さや仮想空間の計算処理の応答遅延などの要因により，仮想物体の正確な操作が困難であるという問題がある．しかし，人間の物体操作に重要とされる力覚や触覚をハードウェアにより同時に提示する手法 [15, 16, 17] や，離散的配置や干渉回避などのソフトウェアによる操作支援手法 [10, 11, 12, 13, 14]

などの導入により，作業内容によっては十分実用に耐え得る操作性を実現できる．

従来の没入型モデラは，仮想物体の形状の生成にのみ着目されているものが大半で，物体間に動きの制約のある物体やアニメーションを伴う仮想物体を生成できるものは少ない．開らは，関節や引き出しなどのような位置や姿勢が制限（**幾何制約**）された物体などの生成に焦点を置いた手法を提案している [18]．この手法は，予め用意された基本形状（**プリミティブ**）を組み合わせるという簡単な操作だけで，複雑な形状を生成できる没入型モデラ VLEGO[14] を拡張したもので，本来実体を持たない幾何制約をその属性を表す形を持ったプリミティブ（**幾何制約プリミティブ**）として提供することで，仮想物体に対する幾何制約の付加を形状の生成と同様の簡単な操作で可能とする．

本研究では，この手法を拡張して，没入型モデラの 3 次元形状生成における優位性を活かした，アニメーションの記述手法について検討する．具体的には，アニメーション要素と起動イベントを，**アニメーションプリミティブ** [19] と**イベントプリミティブ**として提供する手法を提案する [20]．これにより，アニメーション要素の属性を直感的に把握することができ，また，アニメーションプリミティブの位置・姿勢や形状自身を操作することで，対話的にアニメーション要素の属性の変更が可能になる．また，起動イベントについても，イベントプリミティブにより起動イベントを対話的に変更できる．さらに，仮想物体とこれらのプリミティブを組み合わせる簡単な操作で，アニメーション要素や起動イベントを決定できることにより，複雑な操作に習熟しなくても，静的アニメーションや対話的アニメーションの記述が可能になる．

以上に述べた提案手法を没入型モデラに導入することにより，仮想環境に没入したまま，仮想物体のアニメーションを直感的かつ対話的に記述できると考えられる．本論文ではさらに，提案手法に基づいて試作した没入型モデラの実装について述べ，提案手法の有効性を試作システムの使用から考察する．

以降，2 章では，従来のアニメーションの記述手法と没入型モデラについて述べ，3 章では，没入型モデラにおけるアニメーションの記述手法を検討し，提案する．さらに 4 章では，提案手法に基づき試作した没入型モデラについて述べ，これを用いてデザインしたアニメーションの例を示す．5 章では，試作システムを

通して提案手法の有効性について考察する。最後に、まとめと今後の課題について述べる。

## 2. 関連研究:アニメーションの記述と没入型モデラ

前章で述べたように，本研究の目的は，仮想物体の生成手法として没入型モデラに注目し，没入型モデラにおけるアニメーションの対話的な記述手法を開発することにある．本章では，2.1節において，アニメーションを記述する従来手法について詳しく述べる．また，2.2節では，没入型モデラの特徴について詳しく述べ，いくつかの既存の没入型モデラを紹介する．

### 2.1 アニメーションの記述

アニメーションを記述する従来手法として，プログラム言語やスクリプト言語のようななんらかの言語で記述する手法 [1, 2, 3, 4, 5] や，2次元グラフィカルユーザインタフェース (GUI) を用いて対話的にアニメーションをモデリングする手法 [6, 7, 8, 9] が挙げられる．以下，それぞれについて詳しく述べる．

#### 2.1.1 記述言語による手法

通常のプログラム言語で3次元形状のアニメーションを記述できるシステムとして，TBAG[1] や Obliq-3D[2] が挙げられる．TBAGは，C++言語をベースとし，インタラクションやアニメーションが可能な3次元グラフィックスアプリケーションを構築するための様々なクラスやインタフェースを提供するツールキットである．TBAGでは，3次元形状の位置・姿勢などの属性をプログラム言語で記述する．また，任意の属性間に関係式を定義することにより，対話的アニメーションを生成できる．下に，TBAGにおけるアニメーションの記述例を示す．ある3次元形状が変換行列  $xform$  を持っているとする．`assert` 関数によりこの式を有効化すると， $xform$  はこの式を満たすような値のみを取るようになる．すなわち， $xform$  は，時間を表す変数 `Time` と回転軸を表すベクトル `y_axis` を用いて `rot` 関数が出力する，時間によって変化する回転行列となる．

```
assert(xform == rot (y_axis, Time))
```

一方，Obliq-3Dは，3次元グラフィックスアプリケーションを構築するための独自のインタプリタ言語を提供する．Obliq-3Dにおいても，任意の属性間に関係

式を定義することで、対話的アニメーションを生成できる。このシステムでは、その言語で記述した内容は即座に 3 次元シーンに反映されるので、対話的に 3 次元形状やアニメーションの生成が可能である。

また、VRML[3, 4, 5]は、プログラミングの要素も含んだデータ記述言語で、ある瞬間の仮想物体の状態（キーフレーム）をいくつか与えるだけで、そのキーフレーム間の仮想物体の状態の遷移を自動的に補間するキーフレームアニメーションを行なえる。さらに、JAVA などの任意の言語で書かれたスクリプトを組み込むことで、より高度なアニメーションやインタラクションを含んだ仮想空間を構築できる。

これらのシステムでは、3 次元形状や対話的アニメーションをなんらかの言語で記述する。記述言語による手法は、高度なアニメーションやインタラクションを含んだ 3 次元グラフィックスを構築できる反面、思い描いた 3 次元のデザインを言語で正確に表現することが困難であるという問題がある。

### 2.1.2 2 次元 GUI による対話的な記述手法

一般に 3 次元 CG ソフトでは、2 次元 GUI によって精巧な 3 次元形状を対話的にデザインできるだけでなく、多彩なアニメーションを定義できる [6, 7]。例えば、キーフレームアニメーション、逆運動学の採用などによる高度なアニメーションの生成、「衝突検知」、「重力」、「弾性」、「摩擦」などの物理効果のシミュレーション、仮想物体の移動軌跡を指定する「モーションパス」などの記述ができる。また、キャラクターのアニメーションに特化した 3 次元 CG ソフトでは、キャラクターの足跡「フットステップ」を編集することで、キャラクターの歩行アニメーションを生成できる [8]。

これらの 3 次元 CG ソフトでは、一般に図 1 のように、3 次元物体の外観と同時に、3 次元物体の 3 面図や、その物体の構成図（一般にツリー構造）、様々な機能に対応する多数のボタンやメニューなどを CRT に表示し、それらに対してマウスやキーボードなどの入力装置によって操作を行なう。

また、アニメーションを記述することはできないが、2 次元の入出力装置を用いて簡単に 3 次元形状を生成できる手法を取り入れたモデリングツールに、

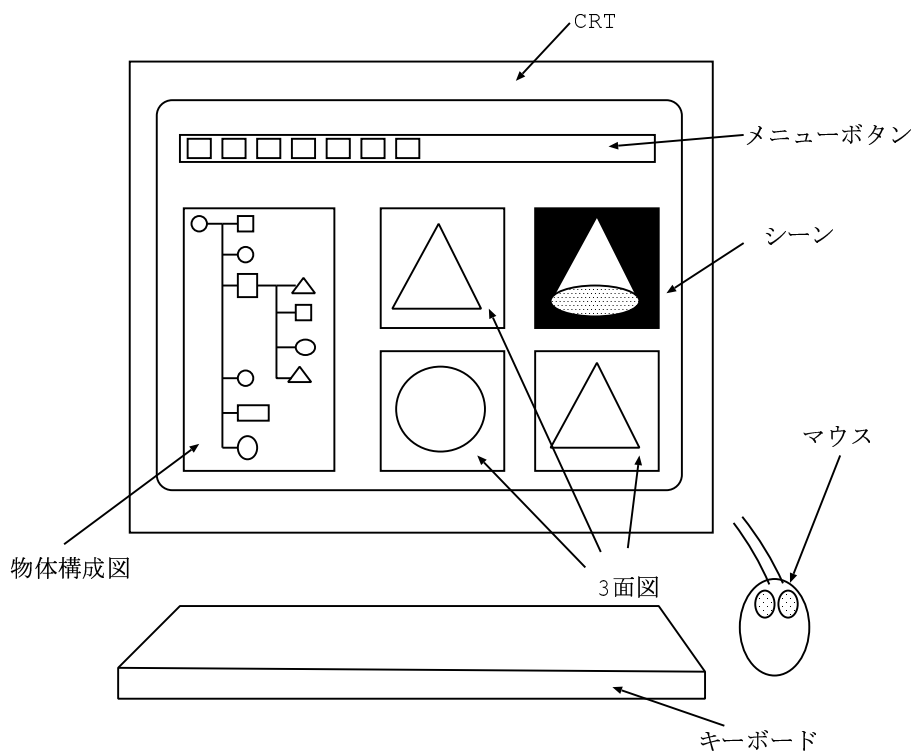


図 1 3DCG ソフトの操作画面例

SKETCH[9]がある。SKETCHでは、マウスもしくはタブレットを使用し、ある特定の1連のジェスチャを入力することにより3次元形状を生成・操作できる。例えば、図2左に示されるような3次元形状の生成のためのジェスチャや、図2右上に示したグループ化などの3次元形状操作のためのジェスチャが用意されている。また、図2右下に示されるように3次元形状の並進などを制限する制約も実装されている。

以上のような2次元の入出力装置を使用する手法は、頭部搭載式ディスプレイ(HMD)や3次元センサなどのような特殊な装置を必要としない反面、入力装置の2自由度と物体の持つ6自由度の対応関係を明示的に頻繁に変更する必要があるなど、3次元形状の視認や操作において直感性に欠けるという問題点もある。

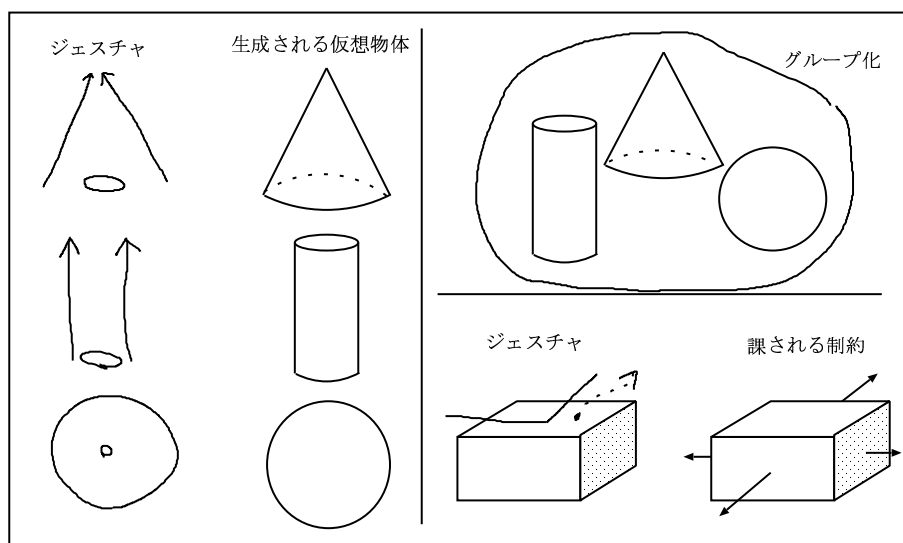


図 2 SKETCH の操作例

## 2.2 没入型モデラ

没入型モデラとは、人工現実感技術（VR）を応用して提示される仮想空間に没入しながら仮想物体を生成できるモデラを指す。ここ数年の計算機の処理能力の飛躍的な向上や、HMD、3次元センサなどのVRには欠かせない入出力装置の高性能化、様々な仮想環境内での仮想物体操作支援手法の開発などを背景として、実用的な没入型モデラが登場し始めた。本節では、没入型モデラの特徴について述べ、既存の没入型モデラをいくつか紹介する。

### 2.2.1 没入型モデラの特徴

没入型モデラは、マウスやCRTなどの2次元の入出力装置を使用する手法と比べて、より実世界に近い感覚で仮想物体を操作する環境を提供できるという利点を持つ。この利点は、1) 形状が直感的に把握できる、2) 形状が直感的に操作できる、3) 対話性の高い形状生成が行なえる、という3つの特長に起因する [14]。以下では、まずこれらの特長について述べる。

**形状の直感的把握：**没入型モデラでは、HMDや3次元センサなどの入出力装置を利用し、両眼立体視や運動立体視による仮想空間の3次元的な視認を可能としている。これにより、仮想物体の形状や仮想物体間の空間的な位置関係の把握を容易にする。

**形状の直感的操作：**没入型モデラでは、一般にグローブやボタンに3次元センサを組み合わせた3次元入力装置を用いて、仮想物体の直接操作が可能である。これにより、2次元入力装置では煩雑になりがちな3次元仮想物体操作を直感的に行なえる。

**対話的な仮想物体生成：**没入型モデラでは、仮想空間に没入しながら3次元仮想物体を実時間で操作できる。これにより、仮想物体に対する操作は直ちに反映され、その操作結果が即座に視認できるため、対話性に優れたモデリングが可能である。

これらの特長をもつ一方で、没入型モデラには仮想物体に対する正確な位置決め操作が困難であるという問題がある。この問題は、仮想空間の計算処理などによって起こる応答の遅延、3次元入力・表示装置の精度の低さなどによって引き起こされる仮想物体の提示位置のずれ、人間の空間指示能力の低さなどに起因する [13, 14]。近年、この問題点を解決するために、ハードウェアおよびソフトウェアによるいくつかの操作支援手法が開発されている。

ハードウェアによる操作支援手法として、人間の物体操作において重要とされる力覚と触覚を同時に提示する手法がある [15, 16, 17]。これらの手法は、特殊で大掛かりな装置が必要であることや、装置の可動範囲にユーザの行動が束縛されるという問題点があるが、より現実の物体操作に近い自然な仮想物体操作環境が実現できる。

ソフトウェアによる操作支援手法として、仮想物体の位置・姿勢を離散的に制約する離散配置制約（グリッド制約）や、仮想物体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自由度ある仮想物体の並進・回転の自由度を2次元並進や1次元回転などに制限する手法などが提案されている [10, 11, 12, 13, 14, 18]。



また、没入型モデラの操作性を向上させる手法として、両手による操作を可能とする手法がある。没入型モデラにおける両手操作には、並列操作により作業が効率化するといったことや、体性感覚として既知である両手の相対位置情報を利用した自然な両手操作を可能にする、といった利点があり、有効な操作支援となり得ることが指摘されている [21, 22, 23].

以上のような仮想物体操作支援手法や両手操作により、十分実用的な操作性を兼ね備えた没入型モデラも登場し始めた。次項では、ソフトウェアによる仮想物体操作支援手法や両手操作を導入している既存の没入型モデラのいくつかを紹介する。

### 2.2.2 既存の没入型モデラ

HMD を用いた初期の没入型モデラとして、3DM[10] が知られている。3DM では、ユーザは 6 自由度の入力装置を 1 個使用し、仮想環境に没入しながら 3 次元形状を生成できる。また、正確な仮想物体操作の支援として、3 次元カーソルの移動をグリッド単位に制限する機能を備えている。

また、Conceptual Design Space (CDS) [11] は、6 自由度の入力装置を 1 個使用し、その入力装置から仮想物体に発せられる光線 (Ray) と呼ばれる直線と交わる仮想物体への操作を可能としている。従って、離れた位置にある仮想物体を操作可能である。また、仮想物体の正確な位置合わせを支援するために、本来各々 3 自由度ある仮想物体の並進・回転操作の自由度を 1 自由度または 2 自由度に制限できる機能を備えている。

その他に、両手操作を取り入れた没入型モデラもいくつか存在する。World in Miniature (WIM) [12] は、仮想環境内での操作とナビゲーションを支援するユニークな手法である。WIM では仮想環境全体を表すミニチュアを提示し、仮想環境での変更はそのミニチュアに反映され、逆にミニチュア内での仮想物体への変更は仮想環境内の対応する仮想物体を変更する。また、ミニチュア内に提示されているユーザを移動させることで、視点を変更することができる。WIM は、仮想環境内において直接届かない位置にある仮想物体への操作を可能にし、また仮想環境の全体像の理解を助ける。

Chapel Hill Immersive Modeling Program (CHIMP) [13] も両手操作を採用している。CHIMP では、円錐型のスポットライトによって、操作対象物体を選択する仮想物体の遠隔操作や、視線方向で選択できるメニューなどの仮想環境内での作業を支援するための様々なインタラクション手法を取り入れている。また CHIMP では、CDS と同様に仮想物体の並進・回転操作の自由度を制限できる機能を備えている。

また、VLEGO[14] は、ブロック玩具を模倣し、円錐、球などの基本形状を持つ仮想物体（形状プリミティブ）を組み合わせるような簡単な操作だけで、複雑な 3 次元形状の仮想物体を生成できる没入型モデラである。VLEGO では、仮想物体の拡大・縮小や彩色などに両手を協調させて行なう操作を取り入れ、簡単に複雑な 3 次元形状を生成できる。また、正確な仮想物体操作を支援するために、仮想物体の位置や姿勢を離散的に制約する離散配置制約や、仮想物体同士の衝突を検出し、自動的に位置を補正する干渉回避などの機能を備えている。

これらの従来の没入型モデラでは形状の生成のみが扱われていたが、開らは VLEGO を拡張し、ドアや引き出しのように、その位置・姿勢が幾何的に制約された物体を、仮想物体生成と同様の簡単な組み合わせ操作で生成する手法を提案した [18]。この手法は、本来形状を持たない幾何制約にその種類や属性を示す形状を与え、特別な仮想物体（幾何制約プリミティブ）として提供することで、幾何制約の属性を直感的に把握・変更することを可能にする利点がある。また、彼らはこの手法が簡単なアニメーションの記述にも適用できることを示している。しかしながら、この手法では、アニメーションの記述能力が限られ、またインタラクションを含んだアニメーションには対応できない。次章では、開らの手法の利点を拡張し、多様な動きのパターンが記述でき、それらとユーザの起こすイベントの関連付けが可能な没入型モデラにおけるアニメーションの記述手法について検討を行なう。

### 3. 没入型モデラにおけるアニメーション記述に関する検討

没入型モデラにおいて仮想物体のアニメーションを扱えるようにすることで、没入型モデラの3次元形状生成における利点を活かした、直感的かつ対話的なアニメーションの記述が可能になることが期待できる。その結果、仮想環境に没入したまま、常に回り続ける風車やびっくり箱のような多様なアニメーションをデザインできる。本章では、本研究で焦点を当てているアニメーションの定義とそのアニメーションの持つ属性について考察し、それらを特別な基本仮想物体（プリミティブ）を用いて記述する方法を提案する。また、実際に用意した様々なプリミティブについて述べる。

#### 3.1 仮想物体のアニメーション

はじめに述べたように、本研究で扱うアニメーションはユーザのアクションとは関わりなく動き続ける静的アニメーションとユーザのアクションによって起動される対話的アニメーションである。本節では、それぞれのアニメーションの構成要素とその要素が持つ属性について考察する。

##### 3.1.1 静的アニメーション

静的アニメーションとは、常に同じ振幅で動き続ける振り子などのように、インタラクションを含まず予め設定された動きを行なうアニメーションを指す。このようなアニメーションの種類は多数あるが、本研究では形状自身に変形するようなアニメーションは扱わず、剛体で構成される仮想物体間の相対位置関係が変化するようなアニメーションのみを扱う。

静的アニメーションの制作に必要な機能を検討するにあたり、図3に示すようなヘリコプターが飛び回るシーンを作る場合を例にとって考察する。まず、ヘリコプターの形状を形状プリミティブを組み合わせて生成する。この時、ヘリコプターのプロペラが回り続けるようにするために、本体とプロペラの間に取り付ける駆動部品が必要になる。そして、プロペラが回り続けるヘリコプターが完成す

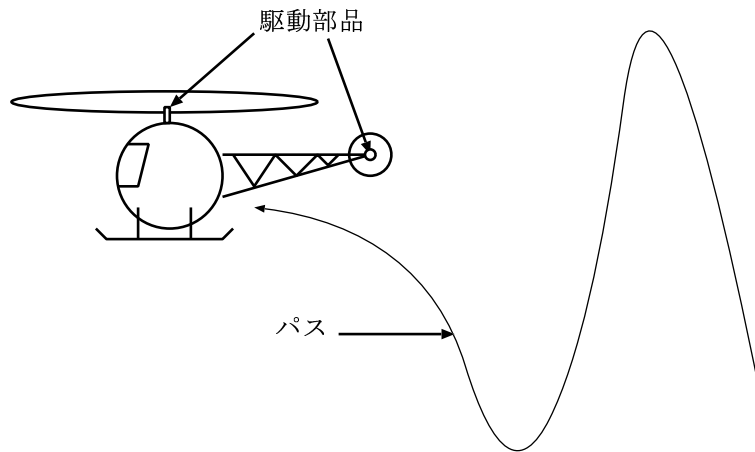


図 3 静的アニメーションの制作例

ると、次にヘリコプター全体が飛び回るように、ヘリコプターの移動経路（パス）を任意に設定できる機能が必要になる。また、駆動部品と移動経路は共に速度を設定できる機能が必要だと考える。

以上から、静的アニメーションのアニメーション要素は、その属性として**アニメーションパターン**と**アニメーションスピード**、そして、プロペラのようなアニメーションさせられる仮想物体（**サブジェクト**）を持つ。アニメーションパターンは予め設定された動きをする駆動部品と任意の経路を設定可能なパスが必要である。また、アニメーションスピードは個々のアニメーションパターンにおいて設定する機能が必要だと考えられる。

### 3.1.2 対話的アニメーション

対話的アニメーションとは、ユーザがクリックすることで歩き始める人形などのように、予め指定されたユーザのアクションによって起動するアニメーション

のことを指し、静的アニメーションと同様な個々のアニメーション要素と、これを起動するイベント（**起動イベント**）で構成される。

対話的アニメーションの制作に必要な機能をさらに検討するにあたり、先の例に似た場面として、ある仮想物体のボタンに触れる（ヒットする）とヘリコプターのプロペラが回りだし、そのボタンを押す（クリックする）とヘリコプターが飛び始めるという場合について考察する。この例では、アニメーション要素は回転するプロペラや飛び回るヘリコプターであり、起動イベントはボタンへのヒットやクリックである。すなわち、対話的アニメーションは、アニメーション要素と起動イベントで構成され、起動イベントは、その属性として、ボタンのようなアニメーションを起動する仮想物体（**トリガオブジェクト**）と、ヒットやクリックといった**起動条件**の種類を持つと考えられる。

なお、アニメーションの起動条件としては、仮想物体 A をクリックしてから仮想物体 B をクリックするといった、複雑なものに対応できることが望ましいが、本研究では、まず、基本的なものとして、仮想物体のクリックやヒットなどのユーザからのアクション、および他のアニメーションの終了によって連鎖的に起動するアニメーションのみを扱うものとする。

以上に述べた、静的アニメーションと対話的アニメーションの属性の関係を図 4 に示す。

### 3.2 アニメーションの記述手法

前節の考察に基づき、本節では、没入型モデラにおいて対話的にアニメーションを記述する手法について検討する。

先に述べたように、開らは幾何制約のような目に見えない属性を没入型モデラで扱っている [18]。開らの手法では、没入型モデラにおいてドアや引き出しのように、その位置や姿勢が制限（幾何制約）された物体を生成するため、本来形を持たない幾何制約という属性に実体を与え幾何制約プリミティブとして提供している。この手法は、次の 3 つの特徴を持つ。

1. 幾何制約を形状を持つプリミティブとして提供することにより、幾何制約の種類や属性を直感的に視覚化する。

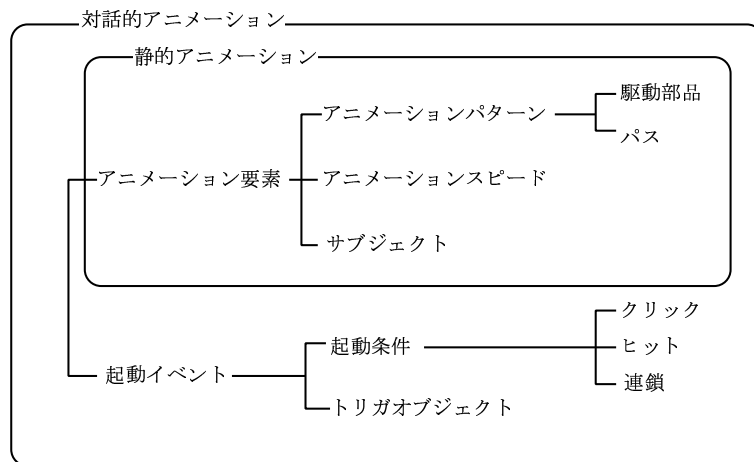


図 4 アニメーションの属性

2. 幾何制約プリミティブの位置や姿勢の変更や形状に対する操作により，幾何制約の属性値の対話的な変更を可能とする．
3. 形状プリミティブと幾何制約プリミティブを組み合わせることにより，仮想物体に対する幾何制約の付加を実現する．

開らの手法は，仮想物体の位置や姿勢に制約を与えるものであり，本研究で扱うアニメーションは，こうした幾何制約と同様の属性を時間軸方向で制御したものと捉えることができる．そこで，開らの手法を拡張してアニメーションの簡易な記述を可能とする手法について検討する．具体的には，没入型モデラにおけるアニメーション要素の記述手法が満たすべき要件として開らの手法に倣い，次の3点を設定する．

1. シーン全体のアニメーションを構成するアニメーションパターンごとに，その属性が直感的に把握できるように適切な形状を持つアニメーション記述用プリミティブ（アニメーションプリミティブ）を提供する．
2. アニメーションプリミティブに対する操作により，アニメーションスピードを含む各属性を対話的に変更できるようにする．

3. 実際にアニメーションさせたい仮想物体（サブジェクト）とアニメーションプリミティブを組み合わせることにより，サブジェクトを指定できるようにする．

次に，起動イベントの記述手法について検討する．開らの手法に倣った起動イベントの記述手法として，アニメーション要素に使われる既存のプリミティブの機能を拡張する方法と，新たな起動イベント記述用プリミティブを用意する方法が考えられる．本研究では，既存のプリミティブを改変する必要がなく，イベント記述に関する将来の機能拡張も容易であると考えられることから，起動イベント記述用のプリミティブとして新たに**イベントプリミティブ**を用意する方法を採用する．この際，イベントプリミティブは，アニメーション要素とトリガオブジェクト，および起動条件の種類を指定できなければならない．また，新たな操作体系を覚える必要のないよう，アニメーション要素やトリガオブジェクトは，これらの仮想物体とイベントプリミティブを組み合わせることで指定できるようにする．

以下では，まずアニメーションプリミティブ単体で実現できる静的アニメーションについて，次にアニメーションプリミティブとイベントプリミティブの組み合わせで実現される対話的アニメーションについて，以上の要件を満たすようなプリミティブの提供方針を検討する．

### 3.2.1 静的アニメーションの記述手法

本項では，静的アニメーションの機構を検討する．まず，構成要素であるアニメーション要素について考える．3.1.1項で検討したように，アニメーションプリミティブは，2種類のアニメーションパターンを持つ．これら2種類のアニメーションパターンに対応して，駆動部品は**モーションプリミティブ**として，パスは**パスプリミティブ**として提供する．以下では，アニメーション要素の属性である，アニメーションパターンやアニメーションスピード，およびサブジェクトについて，視覚化や関連付け操作について順に考える．

**アニメーションパターン**：駆動部品に関しては，幾何制約プリミティブと同様に，代表的パターンを予め用意し，それぞれについて異なる形状を持つモーショ

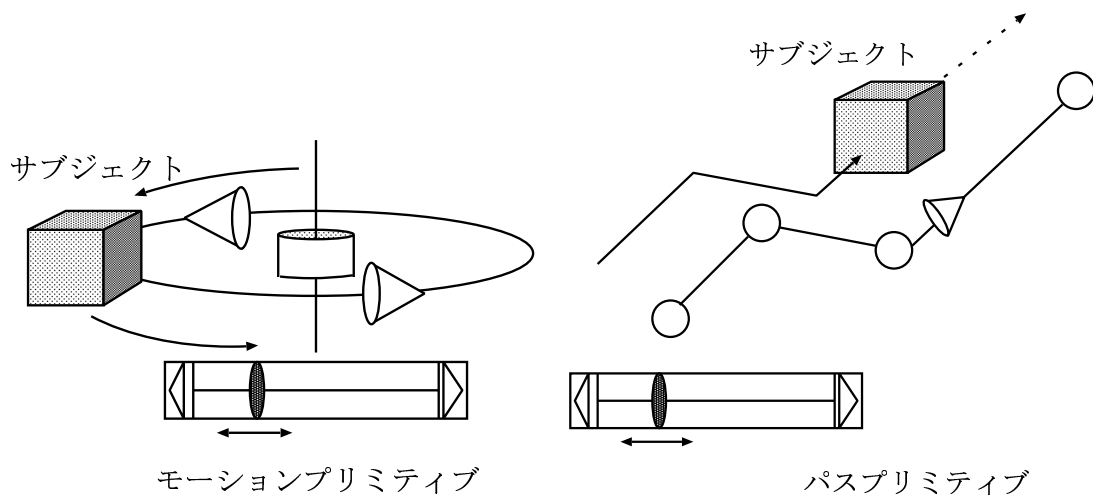


図 5 アニメーションプリミティブの例

ンプリミティブを用意する．モーシヨンプリミティブの具体例として，仮想物体がある特定の回転軸を中心に回転し続けるアニメーション（1次元回転アニメーション）を考える．この例では，図5左に示すように，その形状として，回転軸を表す線分と回転方向を示す矢印，およびアニメーションを把持する際の本体となる円柱を与える．

パスに関しては，図5右に示すように，ノードをつないだ折れ線状の形状を持つようなパスプリミティブを用意する．ノードの追加や移動などを行なうことで，パスプリミティブ全体の形状を変更する．

**アニメーションスピード**：アニメーションの方向や速さを示すために，アニメーションスピードに応じて運動する部品をアニメーションプリミティブに附属して視覚化する．図5左に示した1次元回転アニメーションの例では，回転方向を示す矢印がアニメーションスピードに応じて運動する．図5右に示したパスの例では，移動方向を示す矢印がアニメーションスピードに応じて運動する．また，各アニメーションプリミティブには，アニメーションスピードを変更するための附属部品としてスライダを用意する．



**サブジェクトとの関連付け**：幾何制約プリミティブと同様に、アニメーションプリミティブとサブジェクトを組み合わせることにより関連付けできるようにする。図 5 左に示した 1 次元回転アニメーションの例では、アニメーションプリミティブと直方体のサブジェクトを組み合わせることで、サブジェクトが回転軸まわりに常に回転する。図 5 右に示したパスの例では、パスプリミティブと直方体のサブジェクトを組み合わせることで、サブジェクトがパスの形状に沿って移動する。

このように、アニメーションプリミティブを用いることで、直感的な属性の把握、対話的な属性値の変更、および仮想物体に対するアニメーションの簡易な関連付けが可能になる。

### 3.2.2 対話的アニメーションの記述手法

対話的アニメーションは起動イベントとアニメーション要素から構成される。3.1.2 節で述べたように、起動イベントとしては、トリガオブジェクトを把持する（**ホールドトリガ**）、トリガオブジェクトに触れる（**タッチトリガ**）、という 2 種類に加えて、異なるアニメーション要素の終了を次のアニメーションの起動イベントとする（**連鎖トリガ**）場合の計 3 種類を考える。アニメーション要素については、先に述べたアニメーションプリミティブを利用する。

次に、起動イベントとアニメーション要素の関連付けについて考える。先に述べたように、本研究では、形状プリミティブやアニメーションプリミティブに変更を加えることなくイベントを扱えるようにするために、図 6 に示すように、トリガオブジェクトとアニメーションプリミティブ（図ではモーションプリミティブを用いたが、パスプリミティブを用いてもかまわない）へのポインタを持ち、これらのリンク関係を用いて、起動イベントをアニメーションプリミティブに通知する役目を持つ特別なプリミティブとして、イベントプリミティブを提供する。これらのリンク関係の指定は、トリガオブジェクトやアニメーションプリミティブとイベントプリミティブの組み合わせ操作で実現する。目に見えないリンク関係を明示するためには、例えば、これらの仮想物体を直線で結ぶことでリンク関係を視覚化することが望ましい。この例では、トリガオブジェクトをクリックす

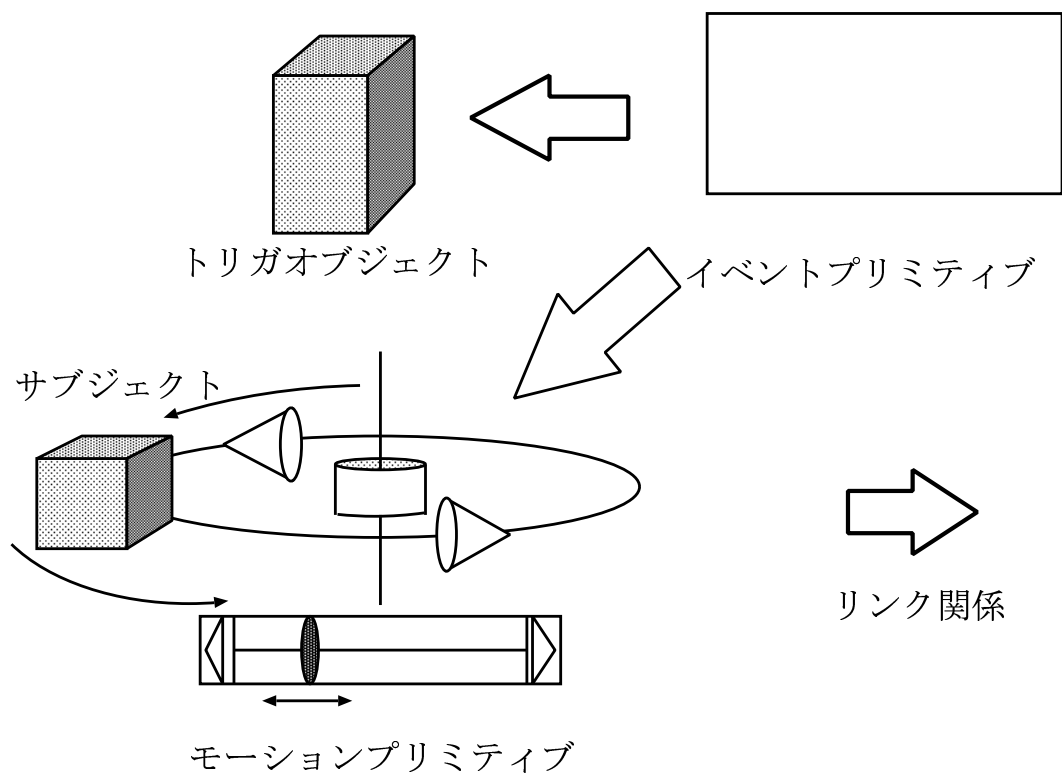


図 6 対話的アニメーションの例

ると、クリックを監視していたイベントプリミティブが1次元回転アニメーションプリミティブに起動イベントを通知し、サブジェクトが円運動を開始する。

イベントプリミティブを用いることにより、イベントを管理する様々な機能を容易に拡張することが可能となる。例えば、起動条件を選択するボタンを設けたり、編集中、タッチトリガなどが誤って起動されないようにするための編集モードと、実際にイベント待ちを行なう作動モードを切替えるボタンを設けるといったことが考えられる。

また、連鎖トリガに関しては、図7に示すようにイベントプリミティブ間にリンク関係を持たせることによって連鎖トリガを定義できるようにする。この場合、イベントプリミティブ1はクリックを起動条件とし、イベントプリミティブ2は連鎖を起動条件とする。トリガオブジェクト1をクリックすると起動条件を満た

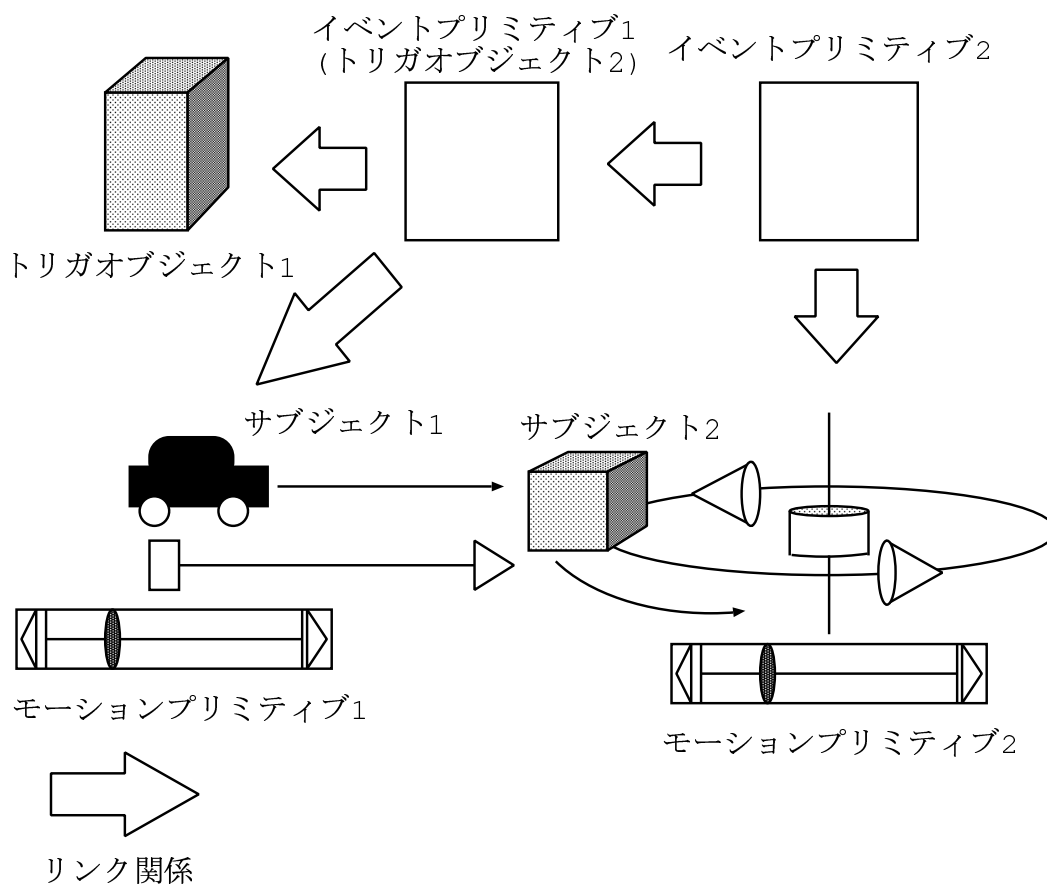


図7 イベントの連鎖

し、イベントプリミティブ1はモーションプリミティブ1（並進アニメーション）を起動させ、モーションプリミティブ1はサブジェクト1（車）をサブジェクト2（ブロック）に当たるまで移動させる。モーションプリミティブ1のアニメーションが終了すると、トリガオブジェクト2（イベントプリミティブ1）は起動条件を満たし、イベントプリミティブ2はモーションプリミティブ2（回転アニメーション）を起動させ、モーションプリミティブ2はブロックを回転させる。

### 3.3 アニメーションプリミティブとイベントプリミティブ

本節では、前節までの考察に基づき、アニメーションの生成に必要な機能を備えたアニメーションプリミティブとイベントプリミティブについて述べる。具体的には、アニメーションプリミティブとしてモーションプリミティブの実例とパスプリミティブの操作方法を説明した後、イベントプリミティブの機能を説明する。

#### 3.3.1 モーションプリミティブ

モーションプリミティブとしては様々なものが考えられるが、本論文では、頻繁に使用されると思われる直線運動と回転運動の2種類を扱う。また、直線運動に関してはさらに細分化し、3種類のモーションプリミティブを用意した。以下では、図8に示した4種類のモーションプリミティブ各々について説明する。

- (a) 直方体と円錐を結ぶ線分の長さだけ平行にサブジェクトを移動させる並進モーションプリミティブである。円錐部を移動することで、線分の長さと同時にサブジェクトの移動距離を変更できる。
- (b) 真中の球の座標に向かってサブジェクトを平行移動させる収束モーションプリミティブである。
- (c) 真中の線分に沿って、他のサブジェクトを単振動させる単振動モーションプリミティブである。その振幅は左右の四角錐を左右に移動することで変更できる。
- (d) 真中の線分を中心軸としてサブジェクトを回転させる回転モーションプリミティブである。

また、図8に示した全てのモーションプリミティブは、下に見える区画がスライダになっており、中の球を移動することでアニメーションのスピードを変更できる。

#### 3.3.2 パスプリミティブ

パスプリミティブは自由に設定した経路に沿って他の仮想物体を移動させる物体である。図9に示すように、パスプリミティブの形状はいくつかのノードとそ

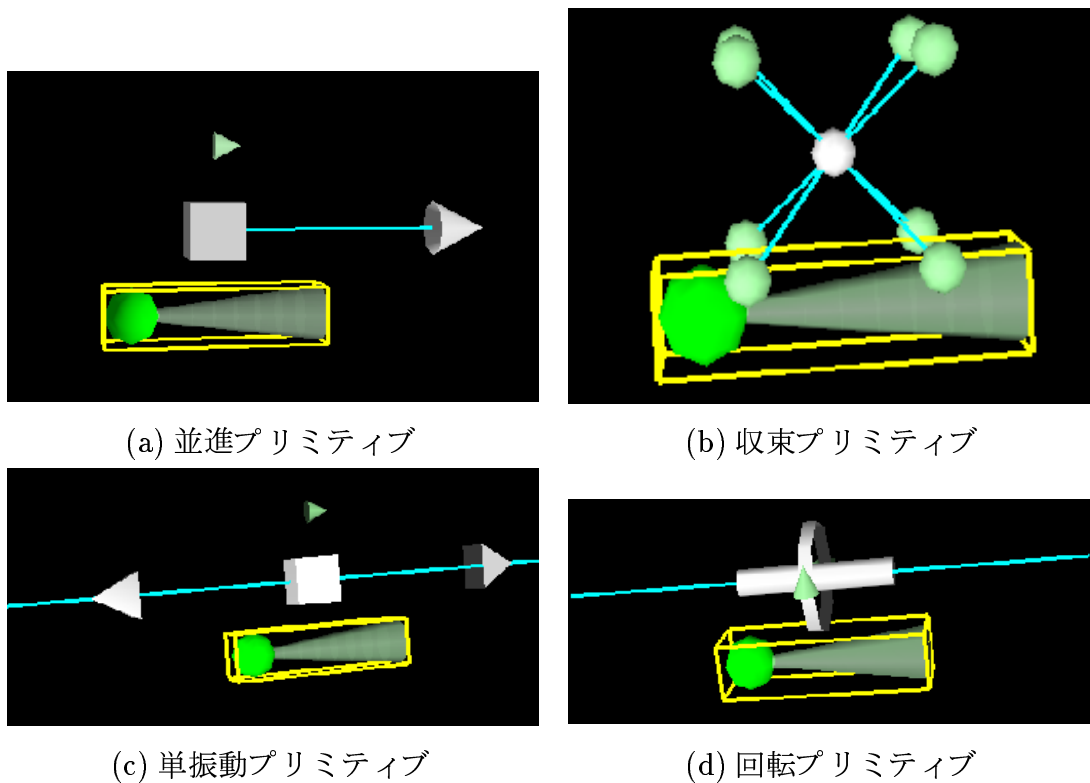


図 8 モーションプリミティブ

れを結ぶエッジ，そしてノードを管理する1つのパスボールで構成される．パスプリミティブは，図9の右下に見えるパスボールでノードを新たに生成していくことで生成する．この時，エッジは自動的に生成される．図9では，サブジェクトとして円錐をアニメーションさせている．なお，ノードは掴んで移動や追加・削除することができ，パスの形状を編集できる．サブジェクトはこのエッジに沿って移動する．また，ノード間を通過する時間は，ノード間の広さに関係なく同一である．モーションプリミティブと同様に下に見える区画がスピードを変更できるスライダーになっており，ノード間の通過スピードを変更できる．

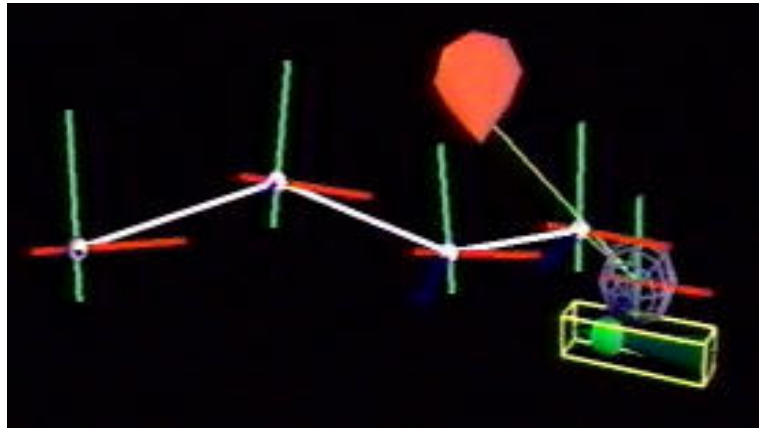
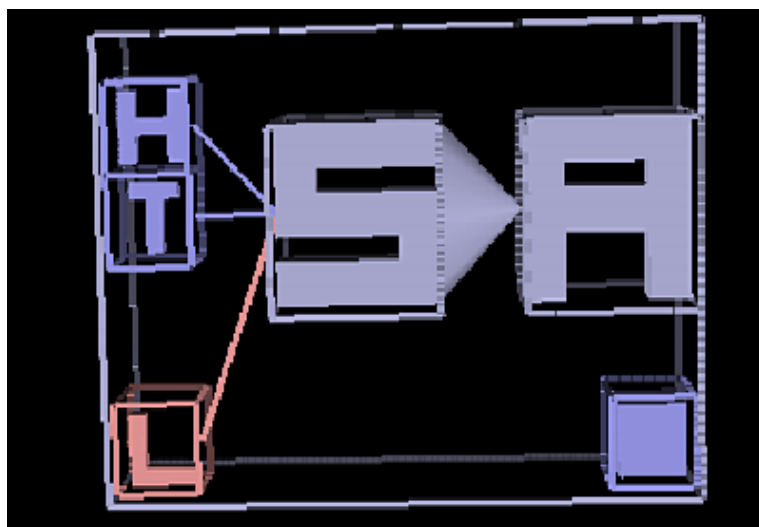


図 9 パスプリミティブの例

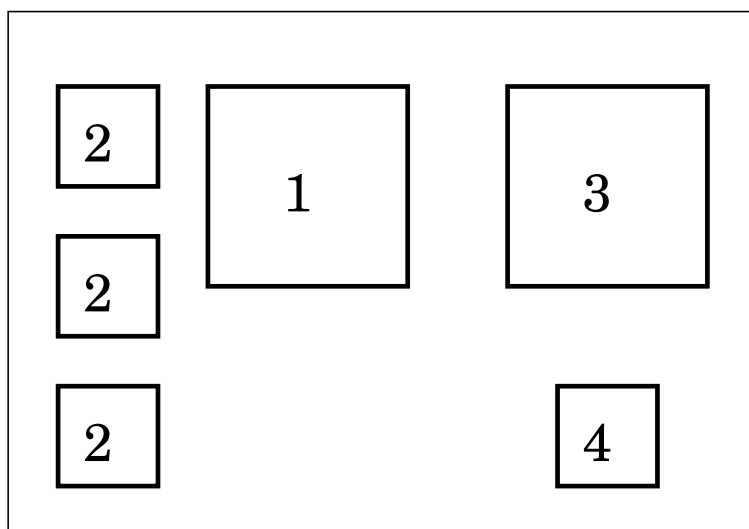
### 3.3.3 イベントプリミティブ

現在実装されているイベントプリミティブを図 10 (a) に示す。以下では、図 10 (b) の模式図を用いて、各部の名称と機能について述べる。

- 1 トリガオブジェクトに対するリンク関係を登録するために**スタートボックス**を用意した。スタートボックスをトリガオブジェクトに組み合わせることで、リンク関係を登録できる。
- 2 起動条件を決定するために、ホールドトリガ、タッチトリガ、連鎖トリガのそれぞれに対応した H,T,および L と書かれたボタン（**トリガボタン**）を用意した。いずれか1つをクリックすることで起動条件を決定する。また、トリガボタンはラジオボタンになっており、選択したボタンのみが青から赤色に変わる。初期状態では、連鎖トリガになっている。
- 3 アニメーションプリミティブに対するリンク関係を登録するために**アニメーションボックス**を用意した。アニメーションボックスをアニメーションプリミティブに組み合わせることで、リンク関係を登録できる。
- 4 トリガオブジェクトなどを登録する時などに、起動イベントが誤って発生しないようにするため、**セーフティボタン**を用意した。セーフティボタンは



(a) 実装例



(b) 模式図

図 10 イベントプリミティブ

トグルボタンになっており、押す毎に作動モード（赤）と編集モード（青）が切り替わる。初期状態では、編集モードになっている。

## 4. アニメーション機能を備えた没入型モデラを試作

前章において、没入型モデラにおけるアニメーションの記述手法として、アニメーションプリミティブやイベントプリミティブを用いる手法を提案した。本章では、提案手法の有効性を示すために試作した没入型モデラの実装について述べる。

### 4.1 システムの設計方針

本システムの開発の目的は、先に提案したアニメーションプリミティブやイベントプリミティブを用いる手法の有効性を探ることである。従って、これらのプリミティブの機能に主眼を置いた没入型モデラを試作する。そのためには、まず仮想物体とアニメーションプリミティブの組み合わせ操作について考慮しなくてはならない。これは、本システムでは後述のように、仮想物体の形状の生成、静的アニメーションや対話的アニメーションの記述の全てがこの組み合わせ操作によって行なわれるため、組み合わせ操作が仮想物体生成およびアニメーション記述の容易さを左右するからである。また、アニメーションさせる側とさせられる側があるという性質上、仮想物体とアニメーションプリミティブなどを組み合わせる際、両者の間に順序関係を持たせることができないなければならない。

また、仮想物体とアニメーションプリミティブなどを組み合わせる際、両者の位置を正確に合わせる必要がある。従って、仮想物体とアニメーションプリミティブの正確な位置合わせを支援する手法として、本システムでは、離散的配置制約を課すこととする。

以上のような設計方針に基づいて、3.3節で用意したアニメーションプリミティブとイベントプリミティブを導入した没入型モデラを試作し、提案手法の有効性を確認する。以降、4.2節で試作システムの概要について述べる。4.3節で本システムの基本操作の実際について、4.4節では組合せ操作について述べる。最後に、本システムを用いてデザインした仮想物体をいくつか紹介する。



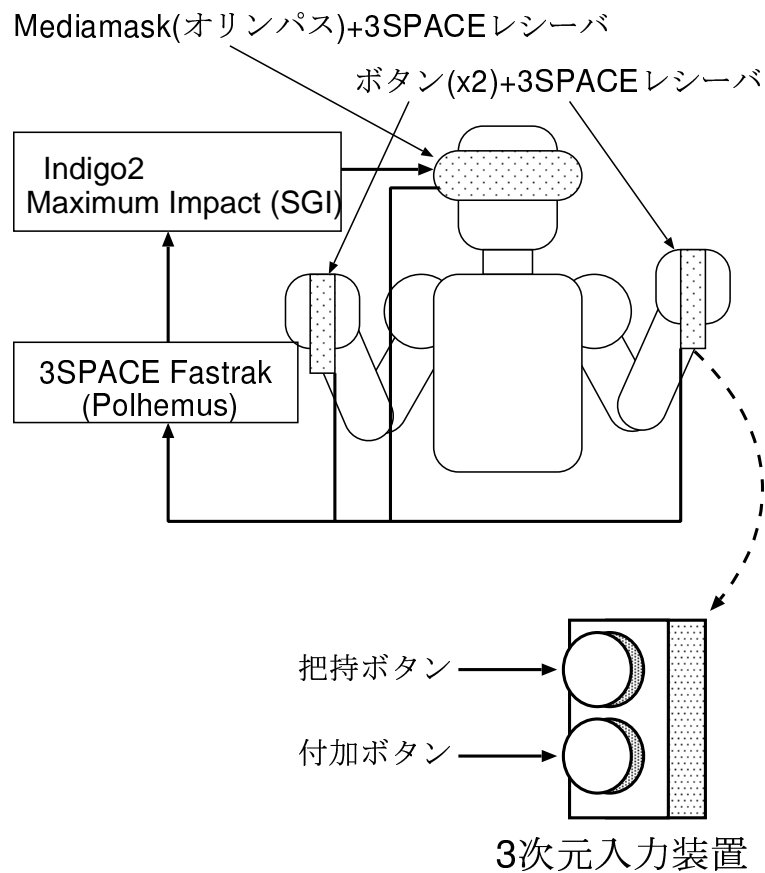


図 11 ハードウェア構成

## 4.2 システム概要

本節では、前節で述べた設計方針に基づいて試作した没入型モデラのハードウェア構成および仮想物体管理手法、さらに本システムで導入している仮想物体操作支援手法について述べる。

**ハードウェア構成:** 試作システムのハードウェア構成を図 11に示す。グラフィックスWS Indigo2 Maximum Impact (SGI 社) 上に試作システムを実装した。ユーザは、磁気トラッカ (Polhemus 社 3SPACE Fastrak) によって計測される視点に基づき提示された仮想空間を HMD (オリンパス社 Mediamask)



図 12 作業の様子

を通して両眼立体視する。操作者は両手用にそれぞれ用意された 3 次元入力装置を左右の手に 1 つずつ持ち、それらに追従する仮想空間中の 2 個の 3 次元カーソルによって仮想物体に対する操作を行なう。製作した 3 次元入力装置には磁気トラッカを組み込んでおり、その位置や姿勢を計測することにより 6 自由度の直接操作を可能としている。本システムを用いた作業の様子を図 12 に示す。また、図 11 下に示すように、3 次元入力装置にはそれぞれ 2 個のボタンが装備されている。以降、上方のボタンを**把持ボタン**、下方のボタンを**付加ボタン**と呼ぶ。それぞれの主な役割は次の通りである。

**把持ボタン**：仮想物体を把持し、並進や回転させるためのボタン。

**付加ボタン**：主に付加操作を実行するためのボタン。

これらのボタンを用いた具体的な操作については、4.3 節で述べる。

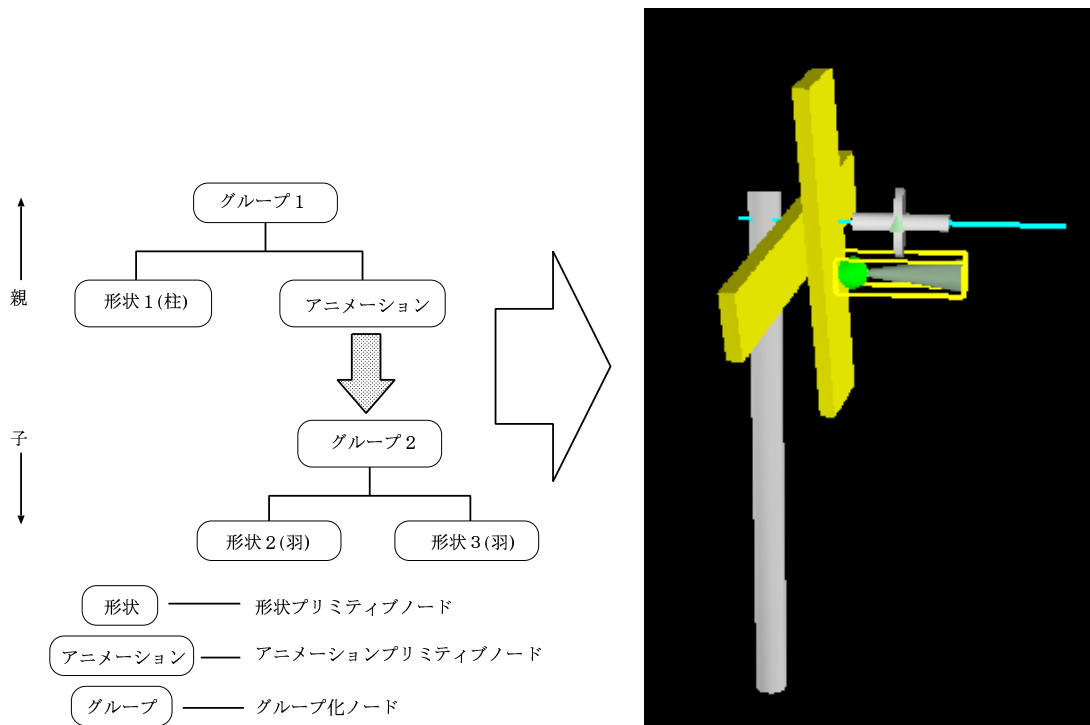


図 13 試作システムにおけるシーングラフの例

**シーングラフによる仮想物体管理：**本システムにおける仮想物体の形状は、1つ以上の球や直方体、円錐、円筒などの基本的な3次元形状（形状プリミティブ）から成る。つまり、予め用意された形状プリミティブを組み合わせることで、仮想物体の形状をデザインする。また、アニメーションを含んだ仮想物体は形状プリミティブに加えて、アニメーションプリミティブを含む。これらの仮想物体の管理には、シーングラフの概念を採用した [24]。本システムにおけるシーングラフの例を図 13左に示す。この例は、図 13右のような回転アニメーションによって回転し続ける風車を表す。グループ化ノードは複数のプリミティブを統括し、それらのプリミティブを1つの仮想物体として扱うことを可能とする形を持たないノードである。また、各ノード間には親子関係が存在し、アニメーションプリミティブに子がある場合、

そのアニメーションプリミティブは自身の属性に基づいて子をアニメーションさせる。例えば，図 13においては，アニメーションプリミティブは，グループ化ノード 2，すなわち形状 2 および形状 3 の 2 枚の羽をアニメーションさせる。

ただし，例外として，イベントプリミティブのリンク関係がある。イベントプリミティブはシーングラフのどの位置にあっても，リンク関係によって親子関係を無視して影響を及ぼすことができる。

**仮想物体操作支援手法：**本システムでは，仮想物体の正確な位置合わせを支援するために，形状プリミティブやアニメーションの位置や姿勢を離散的に制約する離散的配置制約を導入する。すなわち，位置に関しては 1cm 間隔の格子点のみに，姿勢については各軸周りに 90 度単位でのみ配置可能とする。これにより，仮想物体同士，あるいは仮想物体とアニメーションプリミティブ間の相対位置を正確に決定できる。ただし，離散配置制約が必要ない場合は，離散配置制約を無効にできる。また，サブジェクトは自動的に離散配置制約が無効となり，その動きはアニメーションプリミティブに従う。

なお実装にあたり，プログラミング言語として C++言語 [25] を，そのライブラリとして OpenGL[26]，および独自のツールキット [27] を使用した。

### 4.3 基本操作

本システムでは，両手に持った 3 次元入力装置に追従する，仮想空間中の 2 個の 3 次元カーソルによって，仮想物体に対する操作を行なう。本節では，本システムに実装している仮想物体に対する基本的な操作の実際について述べる。

**選択：**3 次元カーソルの先端が仮想物体の内部にあるとき，その仮想物体は選択された状態になり，その仮想物体はハイライト表示される。

**把持：**任意の仮想物体を選択した状態で，把持ボタンを押すことで，その仮想物体を把持できる。把持ボタンのドラッグ中は，把持された仮想物体は 3 次元カーソルの移動に合わせて，離散的配置制約が課されている場合は離散

的配置制約に従いながら，並進および回転する．把持ボタンのドラッグを解除することでその時点の位置と姿勢で仮想空間中に再配置される．

**生成**：形状プリミティブやアニメーションプリミティブなどのすべてのプリミティブは，その種類に関わらず，図 14 に示すプリミティブボックスと呼ばれる棚の中に用意されており，プリミティブボックスの中の生成したいプリミティブを把持することで，そのプリミティブを新たに生成できる．

**複製**：ある仮想物体を把持した状態で，付加ボタンを押すことにより，元の仮想物体を把持したままその仮想物体の複製を仮想空間に生成できる．複製では，元の仮想物体へのリンクを生成するわけではなく，まったく新しい実体を生成するので，それぞれの仮想物体は独立して変更できる．

**消去**：把持している仮想物体を，仮想空間に浮かんでいる図 15 に示すイレイズボックスと呼ばれるワイヤフレームの直方体内に移動させ，把持ボタンを解放することで，その仮想物体を消去できる．

また，離散配置制約の有効・無効や属性プリミティブの可視・不可視を変更するために，以下の操作が実装されている．

**離散配置制約の有効・無効**：何も選択または把持していない状態で把持ボタンをクリックすることで，仮想物体に対する離散配置制約が，有効なときは無効に，無効なときは有効にできる．

**属性プリミティブの可視・不可視**：何も選択または把持していない状態で付加ボタンをクリックすることで，アニメーションプリミティブやイベントプリミティブが可視のときは不可視に，不可視のときは可視にできる．

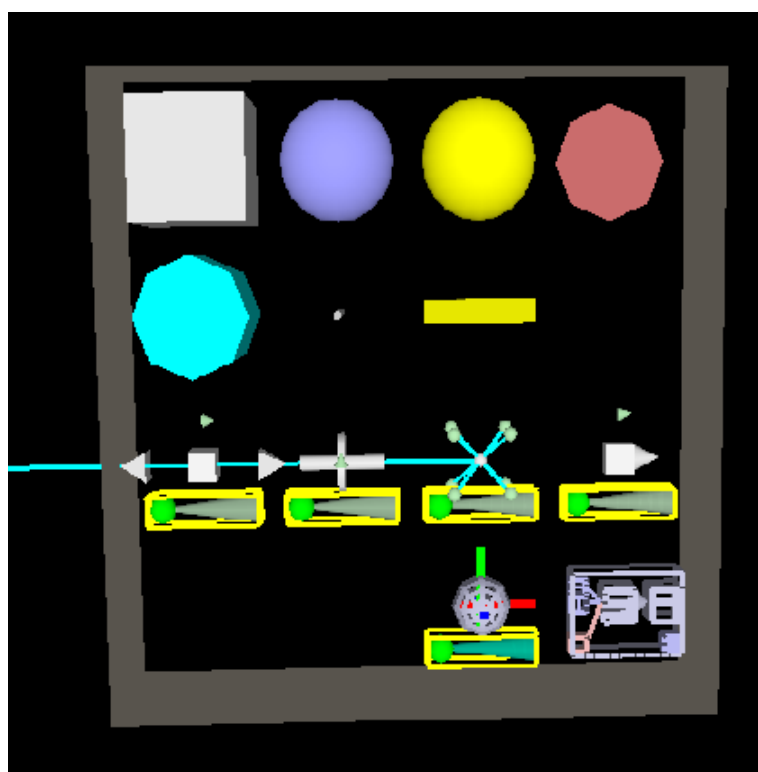


図 14 プリミティブボックス

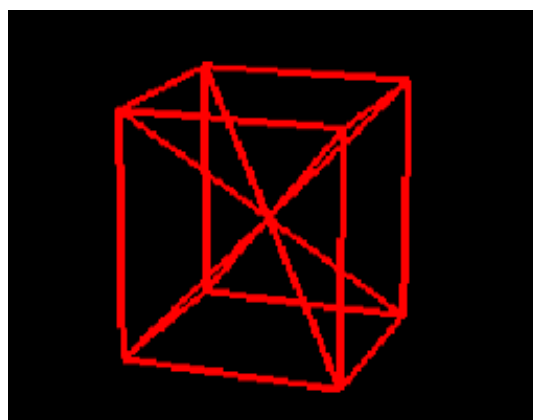


図 15 イレイズボックス

## 4.4 組み合わせ操作

本節では、仮想物体の形状生成、アニメーションプリミティブと仮想物体の関連付けなど多くの場合に用いられる組み合わせ操作について述べる。本システムでは、片手操作と両手操作の2種類の組み合わせ操作を用意した。なお、双方の操作の結果に違いはないので、ユーザは好みの方を使用すればよい。以下では、説明のために「ソースとターゲットの2つの仮想物体について、ソースをターゲットに対して組み合わせる」場合について、それぞれの具体的な操作法を述べる。

**片手操作：** 図 16に示すように、一方の3次元カーソルで、Aのようにソースを選択した状態で付加ボタンを押す。その後、付加ボタンを押しながら3次元カーソルをターゲットの内部まで持っていき、Cのようにターゲットを選択した状態で、付加ボタンを解放する。その結果、ソースのターゲットへの組み合わせが完了する。

**両手操作：** 図 17に示すように、一方の3次元カーソルでソースを把持あるいは選択し、他方の3次元カーソルでターゲットを把持あるいは選択した状態で、ソースオブジェクトに対応する3次元ポインタの付加ボタンをクリックする。その結果、ソースのターゲットへの組み合わせが完了する。

これらの組み合わせ操作の結果、ターゲットがアニメーションプリミティブの場合はソースがターゲットの子となる。それ以外の場合、ソースとターゲットはグループ化される。また、イベントプリミティブとの組み合わせでは、シーングラフに変化は起こらない。

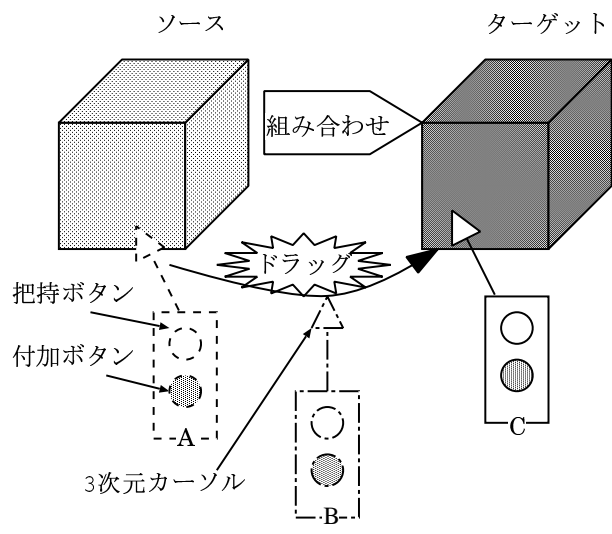


図 16 片手組み合わせ操作

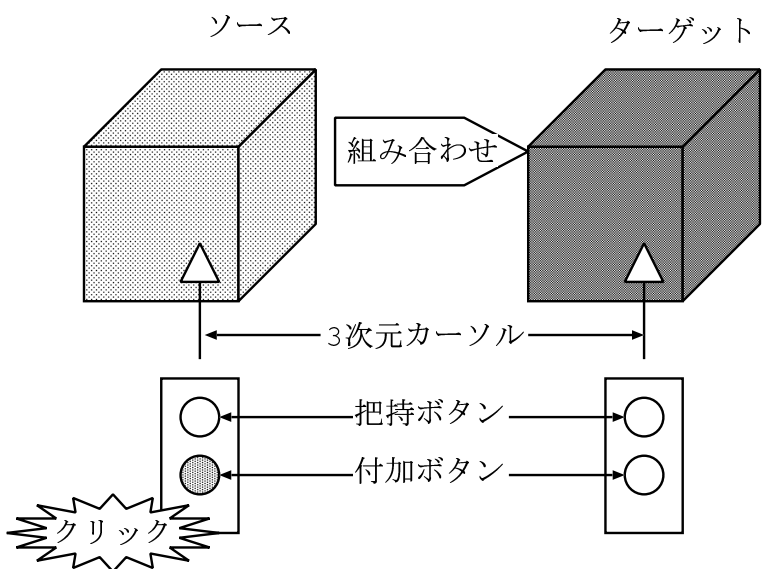


図 17 両手組み合わせ操作



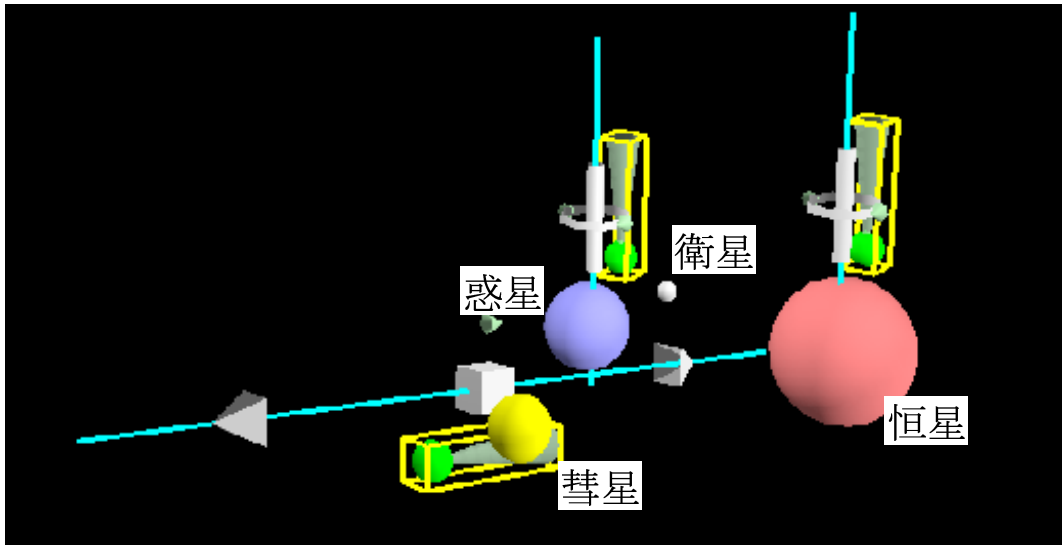


図 18 天体モデル

#### 4.5 デザイン例

本節では、提案手法の有効性を検討するために、試作システムを使用したアニメーションの例を示す。まず、静的アニメーションの例として、簡単な天体モデルを作成したものを図 18 に示す。右にある球（恒星）の周りを真中の球（惑星）が回り、惑星の周りを惑星の右上の球（衛星）が回るように作った。また、左にある球（彗星）は単振動するように作った。なお、制作時間は約 2 分であった。制作の手順としては、まず、恒星と惑星の上に見える回転プリミティブをそれぞれの子にする。次に、惑星は右の回転プリミティブ（恒星の子）の子に、衛星は真中の回転プリミティブ（惑星の子）の子に、彗星は単振動プリミティブの子にすることでシーンが完成する。

次に、対話的アニメーションの例として、簡単なびっくり箱をデザインしたものを図 19 に示す。箱の蓋を選択すると箱に入った人形が上に飛び出してくるようにつくった。なお、作成に要した時間は約 3 分であった。このびっくり箱のデザインの過程について述べる。まず、シーンを構成する部品をプリミティブボックスから取り出す（図 19 (a)）。次に、球や円錐を組み合わせてとんがり帽子をか

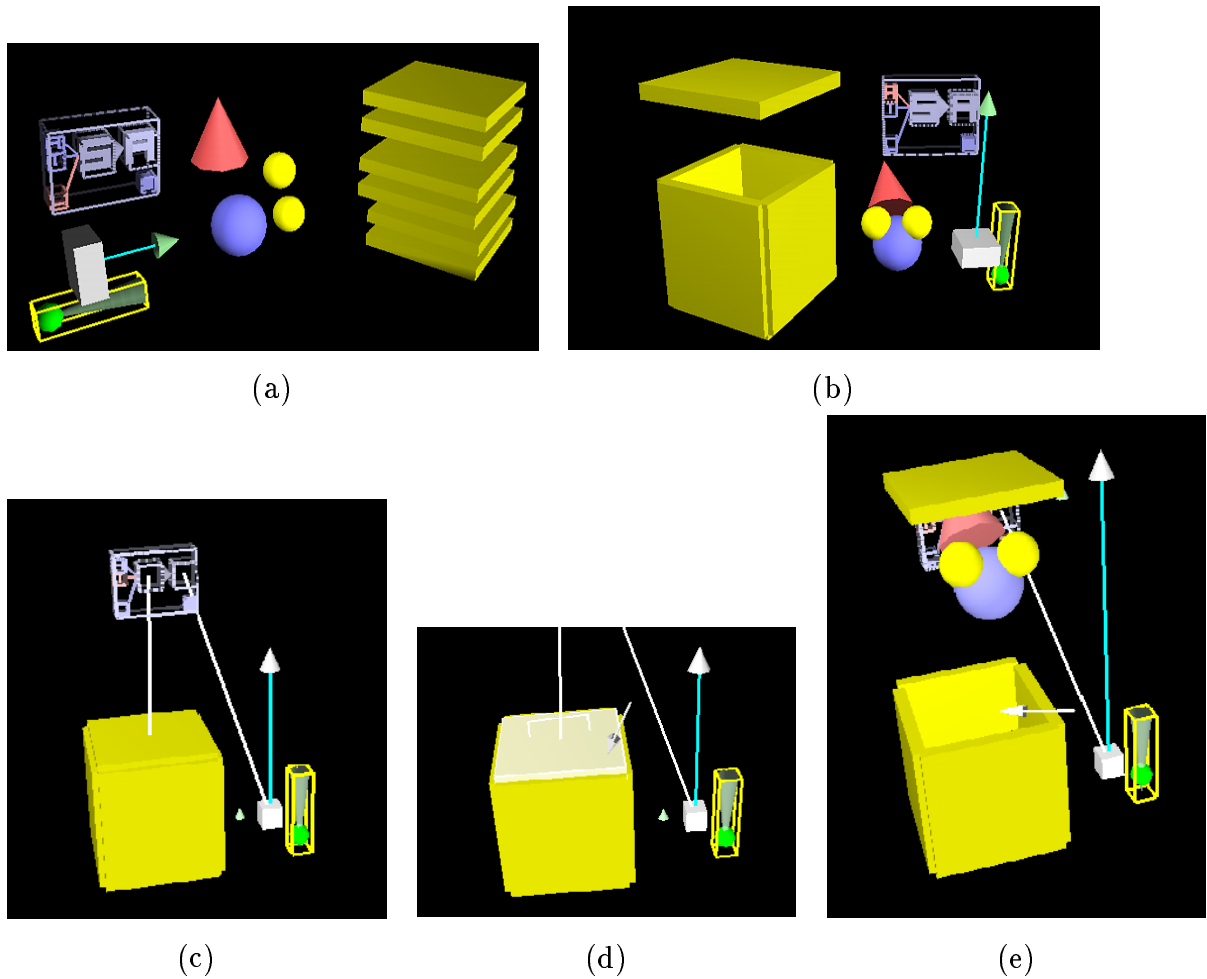
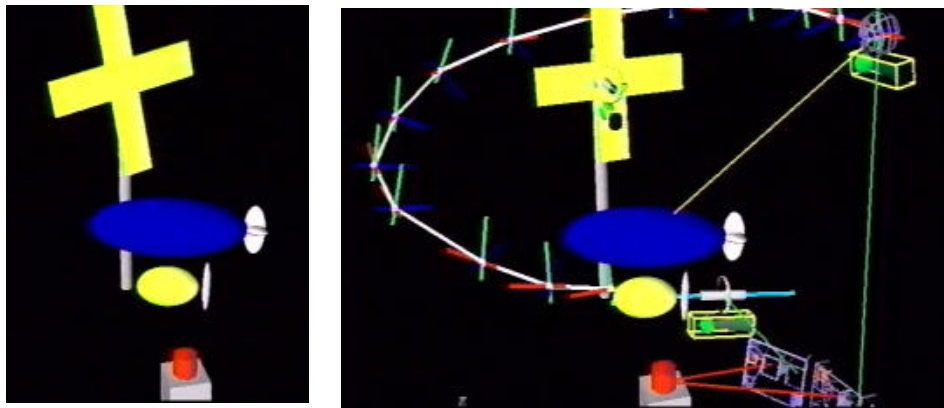


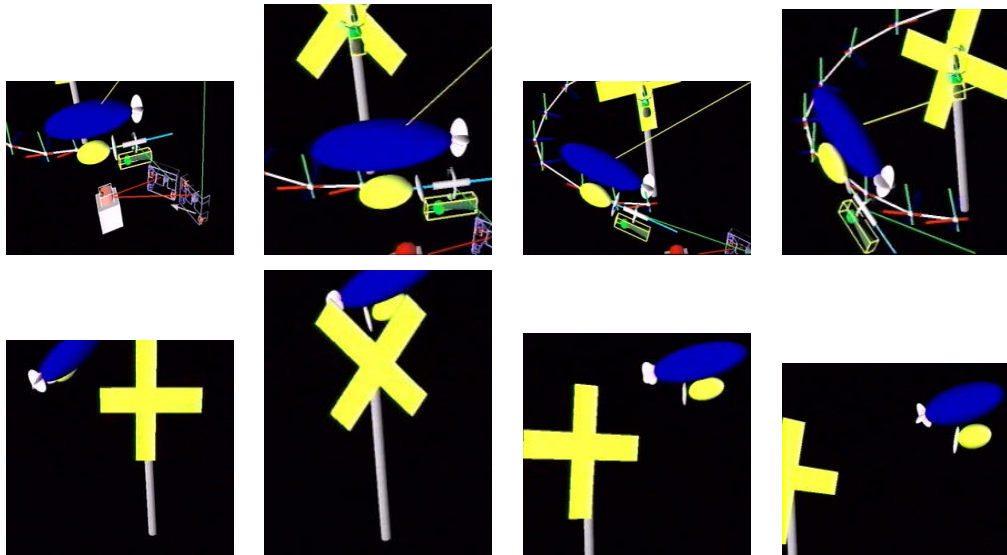
図 19 びっくり箱

ぶった人形を作り，直方体を組み合わせて箱を作る（図 19 (b)）．さらに，人形と箱の蓋にあたる形状プリミティブを並進アニメーションプリミティブに組み合わせ，並進アニメーションプリミティブとイベントプリミティブを組み合わせて，リンク関係を作る．また，箱をトリガオブジェクトとしてイベントプリミティブに登録する．起動イベントの種類を決定し，イベントの記述が完了した様子を図 19 (c) に示す．最後に，イベントプリミティブを作動モードにしておく．このようにして作られたびっくり箱に対して，箱の蓋を選択した様子を図 19 (d) に，



(a)

(b)



(c)

図 20 風車と飛行船

その結果人形が飛び出した様子を図 19 (e) に示す。

最後に、静的アニメーションと対話的アニメーションが含まれたシーンのデザイン例として、回転し続ける風車とユーザのアクションに反応して動く飛行船を作成した。図 20 (a) 上にある風車の羽はユーザのアクションとは無関係に回転し、図 20 (a) 中央にある飛行船は図 20 (a) 下にあるボタンをヒットするとプ

ロペラが回転を始め、ボタンをクリックすると飛行船全体が風車の周りを移動し始めるように作った。なお、作成に要した時間は約5分であった。このデザインの過程について述べる。まず、風車の羽とその中央にある回転プリミティブを組み合わせることで回転し続ける風車を生成できる。次に、タッチトリガを起動条件とするイベントプリミティブは、プロペラと親子関係にある回転プリミティブおよびボタンとリンク関係にする。また、ホールドトリガを起動条件とするイベントプリミティブは、飛行船と親子関係にあるパスプリミティブおよびボタンとリンク関係にする。デザインが完了した様子を図20(b)に示す。そして、ボタンをヒットし、クリックすることでアニメーションが起動された様子を図20(c)に示す。

## 5. 考察

本章では，試作した没入型モデラの使用から，提案したアニメーション記述手法について考察する．また，今後提案手法においてアニメーションの種類を多様化するための指針について考察する．

### 5.1 提案手法の妥当性の検証

本節ではまず，3.2節で述べた，アニメーションプリミティブを提供する際に考慮すべきであるとした，1) アニメーションの形状，2) アニメーションの属性の変更手法，3) 仮想物体とアニメーションプリミティブの組み合わせ操作手法，の3点，および4) イベントプリミティブの組み合わせ操作手法，5) イベントプリミティブの属性の変更手法，の2点の計5点について試作システムの使用からそれぞれ考察する．

- 1) **アニメーションプリミティブの形状:** 試作システムでは，モーションプリミティブについては個々にその属性を表現する独自の形状を与えたため，アニメーションの種類を容易に判別でき，またその属性を直感的に把握できた．しかし，本研究で提供するモーションプリミティブの形状は筆者の独断で与えたものであり，より多くの人にとって分かりやすい形状に改善するためには，評価実験などを行なう必要がある．また，パスプリミティブについては，ユーザが自由に経路を設定できるが，ノードを直線で結ぶ以外の選択肢を用意する必要がある．例えば，パラメトリック曲線を用いて，コントロールポイントを操作することで柔軟な移動経路の設定が可能になると考える．
- 2) **アニメーションの属性の変更手法:** 試作システムでは，モーションプリミティブ自身やパスプリミティブのノードの並進や回転，移動距離やスピードを変更するための部品を操作することによって，アニメーション要素の属性を視覚的に確認しながら対話的に変更できた．なお，アニメーションパターンの種類に応じてその属性も異なるので，アニメーションの属性の変更手法

はアニメーションの種類に依存している。今後アニメーションパターンの種類に応じた適切な属性変更手法を取り入れるべきである。また、パスプリミティブについては、速度や自由な移動経路を設定することができたが、パスプリミティブ全体の位置や姿勢の変更ができなかった。例えば、エッジを掴むことなどにより、パスプリミティブ全体の位置や姿勢を変更をできることが望ましい。

- 3) **仮想物体とアニメーションプリミティブの組み合わせ操作手法:** 試作システムでは、仮想物体とアニメーションプリミティブを組み合わせるために、片手操作と両手操作の2種類を実装した。組み合わせる際にその方向を意識しなければならないが、これらの組み合わせ操作を用いて、簡易に仮想物体に対してアニメーションを記述できた。しかし、両手操作では同時に2個の仮想物体を把持または選択しなければならない、その状態を維持するのが困難であるため、両手操作を用いるより片手操作を用いる方が容易であるように感じられた。今後、組み合わせ操作に関しては評価実験による評価実験による定量的な検討を行なう必要がある。
- 4) **イベントプリミティブの組み合わせ操作:** 試作システムでは、トリガオブジェクトやアニメーションプリミティブの指定などに仮想物体の形状生成と同様の組み合わせ操作を採用した。操作手法が統一されたことによって、習熟の困難さは軽減されたと考える。
- 5) **イベントプリミティブの属性の変更手法:** 試作システムでは、リンク関係の登録に用いるトリガオブジェクトやアニメーションプリミティブの指定にスタートボックスとアニメーションボックスを用いたが、その形状(アルファベット)はその機能を推測する上で必ずしも分かりやすい訳ではなかった。今後より分かりやすい提示方法を検討することが必要だと思われる。例えば、テキストなどによって分かりやすいアイコンを提示することが考えられる。トリガボタンやセーフティボタンについても、同様にして分かりやすく機能を提示できると考える。今後、イベントプリミティブの属性を増やす際には、さらに属性の提示方法を慎重に検討すべきである。

以上のように、改善すべき点は残されているものの、試作システムを用いると、4.5節で示したようなアニメーションを容易に短時間で記述できる。現システムでは、まだアニメーションの種類は少ないが、提案手法がアニメーションの直感的かつ対話的なデザイン環境の構築に有効であることを試作システムを通して確認できた。

## 5.2 アニメーションの多様化

3.2節で、アニメーションをプリミティブとして提供する枠組みを提案した。しかし、3.3節で紹介したプリミティブだけでは、アニメーションの種類としては少ない。本項では、提案手法であるプリミティブを用いる手法において今後、アニメーションを多様化するための指針を考察する。

まず、静的アニメーションと対話的アニメーションの多様化に共通する問題として、アニメーションパターンをどのような方針で増やすかについて考察する。アニメーションパターンを増やすには、次に2つの方針が考えられる。

1. アニメーションプリミティブの種類を増やす。
2. アニメーションプリミティブとサブジェクトの関連付けを拡張する。

まず、1の方針の場合、すなわち、予め必要なアニメーションパターンの数だけ、豊富な種類のアニメーションプリミティブを用意することが考えられる。この場合、そのアニメーションプリミティブ1個だけで、そのアニメーションの属性を表しているのので、容易にアニメーションの属性を把握でき、また変更できるという利点がある。しかし、アニメーションパターンの増加に応じて、アニメーションプリミティブの種類も増えるので、その種類が膨大になる可能性がある。そこで、ユーザが頻繁に使用すると判断したアニメーションパターンを何度も利用できるように、ユーザが作成した任意のパスプリミティブやモーションプリミティブ同士を組み合わせたものをプリミティブボックスへ登録し、これを再利用できるような、拡張性のある機構を用意することが望まれる。こうすることで、予め用意するアニメーションプリミティブの種類を抑えることができると考える。

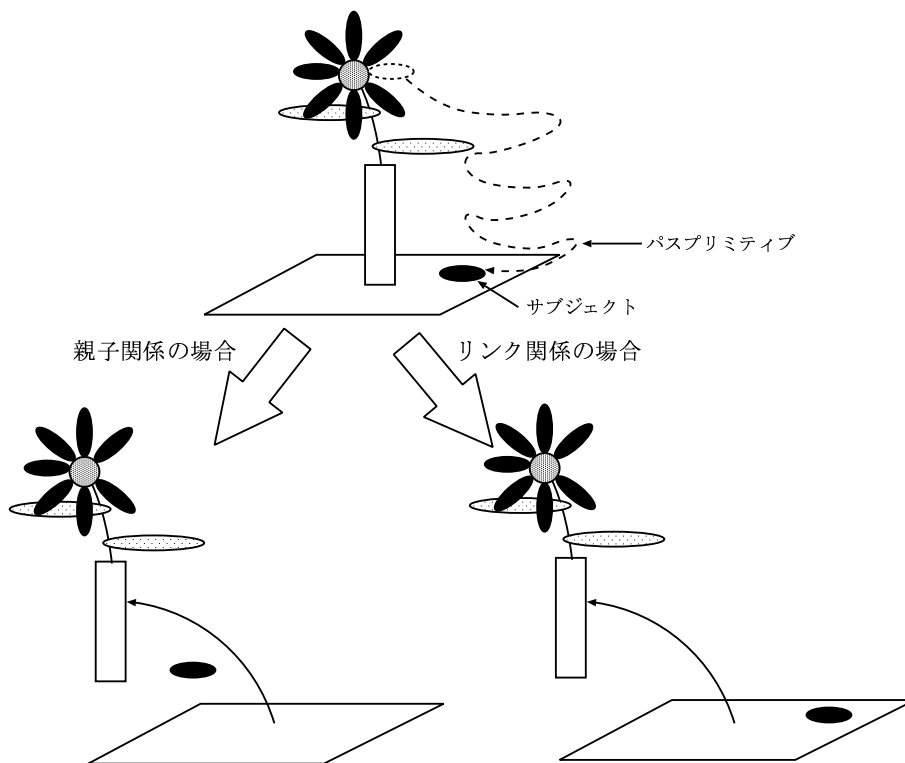


図 21 親子関係とリンク関係の例

次に、2の方針について検討する。この方針は、アニメーションプリミティブとサブジェクトの関係を従来のシーングラフにおける親子関係だけでなく、シーングラフに依存しないリンク関係にも対応するというものである。これにより、シーングラフの親子関係を切ったり、新たな親子関係を作るようなアニメーションができる。例えば、図21のように、パスプリミティブによるアニメーションを終えて散ってしまった花卉（サブジェクト）が花びんに追従するような現象を防ぐことができる。

次に、対話的アニメーションを別の角度から多様化するために、起動イベントを多様化することを考える。起動イベントの種類を増やすには、イベントプリミティブの属性を増やす方針と起動条件を複雑化するという方針が考えられる。まず、イベントプリミティブの属性を増やす方針の例として、時間パラメータを導



入することにより，タイマによって指定時刻になるとアニメーションを起動したり，起動条件を満たしてからアニメーションを起動するまでのタイムラグなどの記述が可能になると考える．また，起動条件を複雑化する方針の例として，複数のトリガオブジェクトを指定し，それらの起動イベントの論理演算による複合的な起動条件を設定できれば，複雑な対話的アニメーションを記述できると考える．このような論理演算は，新たに論理演算用のプリミティブとして用意する方法とイベントプリミティブに組み込む方法が考えられるが，どちらが優れているかは評価実験などを行なって決定する必要がある．なお，起動イベントを多様化する以上の2つの方針は排他的ではなく，両者を同時に用いることも可能である．

## 6. むすび

本研究では，仮想環境没入型モデラにおいて，アニメーションを記述する場合に必要な機能について考察し，アニメーションの属性を対話的に変更することのできるプリミティブを提供する手法を提案した．提案手法の特徴として，以下の3点が挙げられる．

1. アニメーションパターンごとに，その属性が直感的に把握できるように適切な形状を持つアニメーションプリミティブが提供されている．
2. アニメーションプリミティブに対する操作により，アニメーションスピードを含む各属性を対話的に変更できる．
3. 実際にアニメーションさせたい仮想物体（サブジェクト）とアニメーションプリミティブを組み合わせることにより，サブジェクトを指定できる．

1. の特徴により，ユーザは直感的にアニメーションの属性を理解でき，また，2. の特徴により，その属性を対話的に変更できる．そして，3. の特徴により，ユーザは特別な習熟を必要とせず，仮想物体のアニメーションを記述できる．

さらに，提案手法を導入し，アニメーションプリミティブを用いた没入型モデラを試作した．試作システムでは，仮想環境に没入しながらアニメーションを簡単かつ効率的に作成できることを確認した．本システムで利用できるアニメーションパターンは限られているため，作成できるアニメーションも単純なものに限られるが，動きのある簡単な仮想世界を短時間で作成したい場合には有効である．本手法を拡張していくことで，扇風機やエレベータ，自動ドアのような多様なアニメーションとインタラクションを含んだ仮想物体を記述することができると考えられる．

今後，試作システムをより実用的にするための指針として，システムの汎用性の向上，および操作性の向上が考えられる．現在のシステムでは，アニメーションの記述に重点を置いたため，形状プリミティブの色や形，大きさなどの変更は取り入れておらず，仮想物体の形状をデザインする際，予め用意されたプリミティブしか利用できない．VLEGO[14]では，仮想物体のサイズ変更や彩色などのため

に、両手を協調させて行なう自然な操作を導入している。また、筆者の所属する研究室では、仮想環境における曲面変形手法を現在開発中である [28]。試作システムをモデリング環境としてより汎用的にするためには、このような形状プリミティブの変形手法を取り入れることが有効であると思われる。また、試作システムの操作性を向上させるためには、干渉回避などの操作支援手法を導入することが有効であると考えられる。さらに、離散的配置制約の有効・無効やアニメーションプリミティブの可視・不可視の変更操作についても、現在の操作方法では、仮想物体を掴み損なった場合などに誤ってモード変更するという誤操作が多発するため、モードを制御するためのツールを仮想空間に用意する必要があると考える。最後に、今後の課題としては、以下の4点が挙げられる。

- 仮想物体の変色や弾性変形などのアニメーションも範疇にいた、アニメーションパターンの拡充。
- アニメーションプリミティブとサブジェクトの関連付けの拡張。
- 時間パラメータなどのイベントプリミティブの属性の拡充。
- 論理演算などの複雑な起動条件への対応。

これらは、アニメーションの種類を多様化するためのものであるが、提案手法は試作システムで実装したプリミティブ以外のアニメーションにも応用可能である。しかし、アニメーションパターンの種類を増強する場合、考えられるアニメーションパターンの種類は膨大であり、仮に全てのアニメーションに本手法を取り入れてプリミティブとして表現しても、逆にアニメーションの記述が非直感的で煩雑になる可能性がある。また、アニメーションプリミティブの種類が膨大になる可能性もある。アニメーションプリミティブ同士を組み合わせることにより、アニメーションパターンの種類を拡充する手法などを取り入れ、アニメーション記述手法の実用性をさらに向上していく必要がある。

## 謝辞

本研究の全過程を通して，懇切なる御指導，御鞭撻を賜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 助教授に厚く御礼申し上げます。

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

本研究を通じて，終始熱心に相談に応じて頂き，多くの時間と労力を割いて下さったソフトウェア基礎講座 清川 清 氏に深く感謝致します。

常日頃より温かい御助言を頂いたソフトウェア基礎講座 大隈 隆史 氏，並びにソフトウェア基礎講座の諸氏に深く御礼申し上げます。

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

## 参考文献

- [1] C. Elliott, G. Schechter, R. Yeung, and S. Abi-Ezzi: “TBAG: A High Level Framework for Interactive, Animated 3D Graphics Applications,” Proc. of SIGGRAPH’94, pp.421-434, 1994.
- [2] M. A. Najork and M. H. Brown: “Obliq-3D: A High-Level, Fast-Turnaround 3D Animation System,” IEEE Trans. on Visualization and Computer Graphics, pp.175-193, 1995.
- [3] R. Lea, 松田, 宮下: JAVA+VRML Java と VRML2.0 で作るインタラクティブ 3D ワールド, プレンティスホール出版, 1997.
- [4] J. Hartman and J. Wernecke: The VRML 2.0 Handbook, Addison Wesley Developers Press, 1996.
- [5] 武藤: “VRML の応用例と最新動向”, NTT 技術ジャーナル : NTT コミュニケーション情報誌, Vol.8, No.10, pp.98-101, 1996.
- [6] Ray Dream Studio 5 ユーザガイド, MetaCreations Corporation, 1997.
- [7] LightWave 3D version 5.5J 日本語ユーザガイド, NewTek Inc, 1997.
- [8] Character Studio User’s Guide and Tutorials, Autodesk Inc, 1997.
- [9] R. C. Zeleznik, K. P. Herndon, J. F. Hughes: “SKETCH: An Interface for Sketching 3D Scenes,” Proc. of SIGGRAPH’96, pp.163-170, 1996.
- [10] J. Butterworth, A. Davidson, S. Hench and T. M. Olano: “3DM: A Three Dimensional Modeler Using a Head-Mounted Display,” Proc. of ACM Symposium on Interactive 3D Graphics, pp.135-139, 1992.
- [11] D. A. Bowman and L. F. Hodghes: “User Interface Constraints for Immersive Virtual Environment Applications,” Graphics, Visualization and Usability Center Technical Report GIT-GVU-95-26, Georgia Institute of Technology, 1995.

- [12] R. Stoakley, M. J. Conway and R. Pausch: "Interactive Worlds in Miniature," Proc. of ACM SIGCHI'95: the Conference on Human Factors in Computing Systems, pp.262-272, 1995.
- [13] M. R. Mine: "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.
- [14] 清川, 竹村, 片山, 岩佐, 横矢: "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.
- [15] 平田, 水口, 佐藤, 川原田: "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), No.8, pp.1788-1795, 1993.
- [16] 岩田, 早川: "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.
- [17] Y. Nishino, Y. Kunii and H. Hashimoto: "20 DOF Haptic Device for the Interaction with Virtual Environments," Proc. of ICAT'97, pp.85-92, 1997.
- [18] N. Hiraki, K. Kiyokawa, H. Takemura and N. Yokoya: "Imposing Geometric Constraints on Virtual Objects within an Immersive Modeler," Proc. of ICAT'97, pp.178-183, 1997.
- [19] 財満, 清川, 竹村, 横矢: "仮想環境没入型モデラにおける対話的アニメーション付加手法の検討", 日本バーチャルリアリティ学会第3回全国大会論文集, pp.47-48, 1998.
- [20] 財満, 清川, 竹村, 横矢: "仮想環境没入型モデラにおけるインタラクティブアニメーションのモデリング手法の検討", 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), Vol.98, No.616, pp.9-16, 1999.
- [21] A. Hinckley, R. Rausch, J. C. Goble and N. F. Kassell: "A Survey of Design Issues in Spatial Input," Proc. of ACM Sympo. UIST'94, pp.213-222, 1994.

- [22] E. Sachs, A. Roberts and D. Stoops: “3-Draw: A Tool for Designing 3D Shapes,” IEEE Computer Graphics & Applications, Vol.11, No.6, pp.18-26, 1991.
- [23] C. Shaw and M. Green: “Two-Handed Polygonal Surface Design,” Proc. of ACM Sympo. UIST’94, pp.205-212, 1994.
- [24] P. S. Strauss and R. Carey: “An Object-Oriented 3D Graphics Toolkit,” Proc. of ACM Sympo. UIST’94, pp.341-349, 1992.
- [25] 林: 新 C++入門, ソフトバンク株式会社, 1998.
- [26] M. Woo, J. Neider, T. Davis: OpenGL Programming Guide Second Edition, Addison-Wesley Developers Press, 1997.
- [27] 世利, 大隈, 清川, 竹村, 横矢: “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報告 (電子ディスプレイ), Vol.97, No.526, pp.95-100, 1998.
- [28] 松宮, 清川, 竹村, 横矢: “陰関数表現を用いた仮想空間没入型曲面形状モデラ”, 日本バーチャルリアリティ学会第 3 回全国大会論文集, pp.53-56, 1998.