

# 身体動作の記述と表示のための実行制御環境の開発

平松 尚子<sup>†,☆</sup> 八村 広三郎<sup>†</sup>

我々は、人間の身体動作を記述し、それを蓄積、検索、表示することを目標に研究を行っている。人間の身体動作の記述方法には、動作の入力方法によって、様々な形式が考えられるが、これらにはそれぞれ一長一短がある。多様な動作を表現するためには、複数の記述法を併用し、必要に応じて使い分けることが望まれる。本研究では、動作情報をオブジェクト指向モデルによってカプセル化し、様々な形式で入力される動作情報を統一的に扱えるようにする。このため、基本的な動作モデルのテンプレートを作成し、動作エレメントライブラリを構築することによって、動作記述を簡素に行えるようにした。さらに、このような記述データを細かな単位に分割して編集したり、動作を表示したりするための、実行制御環境を開発した。本システムにより、民族舞踊やジェスチャーのような身体動作を効率的に記述し、またこれを再利用して新たな動作の記述に利用することができる。

## System for the Description and Display of Human Body Movement

NAOKO HIRAMATSU<sup>†,☆</sup> and KOZABURO HACHIMURA<sup>†</sup>

We have been studying about notation, storage, retrieval and display of human body movement. There are several types of notation for human body movement. Each of these notations have its own advantage and disadvantage. It is advisable that we use multiple types of notation according to the required level of description, and handle them in an integrated manner. We employed an object-oriented approach, for modeling the data of human body movement. By using the encapsulated object-oriented "Movement model", we can handle multiple types of movement description equally. We prepare a library of templates for basic movement notations, which enables a versatile description of human body movement.

### 1. はじめに

近年、様々な分野で、人間の身体動作をコンピュータシステムで扱う研究がさかんになっている。たとえば、コンピュータゲームなどのエンタテインメント分野、バーチャルリアリティ、ロボティクス、さらにマンマシンインタフェースなどへの応用がある。

一方、人文科学の分野でも、たとえば、民族舞踊のデータベース化と比較研究、身振りやしぐさなどの非言語コミュニケーションの研究への応用など、このようなシステムへの要求は高まっている<sup>1)</sup>。

身体動作を扱うことの課題には、まず、マルチリンク構造をした人体各部の動きを計測して入力し、これを適切な方法で記述することがある。また、三次元グ

ラフィックスの技術を用いて、このような記述データからグラフィックスで人体モデルの動作として表示することが必要である。さらに、記述データを記録・蓄積し、検索の対象としたり、入力されたデータを編集したりすることも、重要な課題となる。

身体動作の入力・記述方法については、いくつかの方式が考案され、利用されている<sup>2)</sup>。最近では、映画やアニメーションの作成によく利用されているモーションキャプチャの技術によって、実際の人体の各関節の位置の時間変化を記録することができるようになった。これによると、実際の動作を正確に記録・再現することができるが、このデータは位置座標の時間的系列であり、この生のデータだけから身体全体の様子を的確に読み取るのは容易ではない。取得したデータを編集したり、意味的にまとまりのある部分を取り出して再利用したりすることもこのままでは難しい。さらに、入力時の被験者とは異なる体型の人体モデルのための動作データとして直接適用できないのも大きな問題である。

<sup>†</sup> 立命館大学理工学部情報学科

Department of Computer Science, Ritsumeikan University

<sup>☆</sup> 現在、ミノルタ株式会社

Presently with Minolta Co., Ltd.

一方、舞踊の分野では、音楽における楽譜に相当する舞踊譜が考案され、比較的広く利用されている。舞踊譜の代表的なものには Labanotation がある<sup>3)</sup>。このような記法によれば、人体モデルの体型に依存することなく、たとえば、「右手を前に」のような記述ができる(実際の舞踊譜上では、これは図式記号で表現される)ので、記述された運動の意味や様子を把握するのが容易であり、また、このような運動単位の編集も考えやすい。しかし、一方で動作が図式記号化されたものであるため、Labanotation の譜面から、実際の人間の動作を正しく把握することには若干の習熟を要する。また、譜面としての Labanotation は、このままでは計算機システムで直接扱えるデータではないので、適当な入力システムにより、機械可読のデータに変換する必要がある。

以上のように、身体動作の入力・記述方法にはそれぞれ長所短所があるので、すべての分野で普遍的に利用できるものは存在しないといえる。このため、複数の記述法を併用し、必要に応じて使い分けられることが望まれる。また、どの方法をとるにしても、一度作成された動作データを効率的に編集し再利用するための、実行制御環境の整備が強く求められる。

このような背景にあつて、我々は人間の身体動作の入力・記述・表示を扱うための総合的な情報処理システムの構築を行うことを目標として研究を行っている<sup>1)</sup>。本論文では、このための1つの取組みとして、オブジェクト指向モデルによる動作記述と編集・表示のための実行制御環境について述べる。ここでは、動作記述と表示用のデータを生成する手法を、オブジェクト指向モデルによってカプセル化し、様々な形式で入力される動作データを統一的に扱えるようにした。また、基本的な動作オブジェクトのテンプレートを作成して、ライブラリ化することによって、動作の記述を容易にかつ簡潔に行えるようにした。

さて、身体動作の表示、すなわち人体アニメーションの生成については、キーフレーム法に代表される運動学的アプローチと、四肢や筋肉の物理特性を考慮して表示生成する動力的アプローチがある<sup>4)</sup>。最近では、自然な体の動きを生成するために、動力的アプローチがさかんになりつつある。たとえば、地形の三次元形状にあわせて歩行動作を生成するシステム<sup>5)</sup>や、時空間上の制約のもとで、人体モデルの動作を自動生成する手法<sup>6)</sup>などが報告されている。

これらは、三次元人体アニメーションを効率的に作成するという観点にたてば、確かに優れたものであるが、我々の研究の主目的は、与えられた様々な身体動

作の記述と蓄積・編集、検索などであり、アニメーションの作成それ自体については特に重要視していない。したがって、ここでは運動学的アプローチによる記述と表示を中心とする。

一方、現在 VRML コンソーシアムでは、VRML で扱う人体モデルの構造やアニメーションの手法などについて詳細な検討がなされている<sup>7)</sup>。これらは、将来においては三次元人体アニメーションのための有力なプラットフォームになると期待される。しかしながら、動作データの編集機能や制御環境については VRML の枠内では規定されておらず、EAI 機能を利用した、Java などによる別プログラムに依存している段階であつて、身体運動の記述と編集などにかかわるすべての課題が解決しているわけではない。

また、最近では三次元人体アニメーションの作成ができる、いくつかの三次元グラフィックスシステムが販売されており、動作データの編集も可能になっている。しかし、これらの機能にはそれぞれに一長一短があつて、体の部分ごとに運動を分割する機能、動作を時間単位ごとに細分化して扱う機能、これらをさらに複合化して取り扱う機能などの点で十分でなく、我々の目的である身体動作の記述と編集のためには、少なくとも現状では、どれも十分ではないと判断している。

## 2. 身体動作情報処理システム

### 2.1 システムの構成

我々の身体動作情報処理システムの全体は、図1に示すような構成になっている。ここでは、このシステムの概略について述べる。

このシステムは、図1に示しているように、大きく3つの部分に分けることができる。

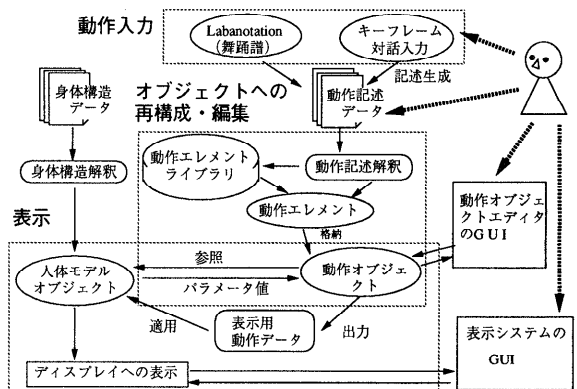


図1 身体動作情報処理システムの概要

Fig. 1 Structure of the human body movement processing system.

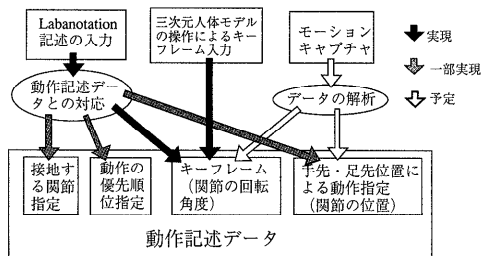


図2 動作記述データの作成

Fig. 2 Generation of the movement data.

- (1) 各種入力システムからの動作入力
- (2) 動作のオブジェクトモデルへの再構成・編集
- (3) 表示

このうち、今回は特に、オブジェクトモデルへの再構成・編集と、表示を行うための実行制御環境に重点をおいて、システムの実装を行った。

## 2.2 各種入力システムからの動作入力

まず、身体動作がシステムに入力され、「動作記述データ」と呼ばれるデータが生成される。動作記述データとは、各種入力手法から取得される身体動作の原データを変換して作成する中間的なデータで、後述する動作エレメントへの変換処理に適した記述形式で作成されている。

身体動作の入力については、様々な手法によるものを対象に考えており、特定の入力形式に依存しないように工夫しているが、現在のところ、以下の2つの方法で行っている。

- (1) 三次元人体モデルをグラフィック画面上で対話的に操ることによって、いくつかのキーフレームにおける身体姿勢を入力する方法
- (2) Labanotation 譜面データの入力・編集システムによる入力手法

これらのいずれも、現時点では独立したシステムとして作成されており、上述の動作記述データを出力する。

各身体動作入力手法のデータから動作記述データが作成されるまでの概略を図2に示す。

(1)の手法では、三次元人体モデルを対話的に操作することによって、各キーフレームにおける関節の回転角度を記録した動作記述データが生成される。

(2)のLabanotationに基づく入力方法では、Labanotation 譜面データの読み取り・入力・編集のために作成している別のシステム<sup>8)</sup>を用いる。この入力システムには、紙面に印刷されたLabanotation 譜面の自動読み取りを行うLabanReaderと、グラフィックスによる対話的な譜面データの入力を行うLabanEditor

とがある。現時点では、自動読み取りの認識率が十分でないので、LabanEditorによる対話入力の方を主に用いている。

LabanEditorでは、入力したLabanotationのそれぞれの図式記号に対応したテキストデータを書き出すようになっている。したがって、Labanotationのそれぞれの図式記号で表される動作に対して、あらかじめ、動作記述データとの対応表を用意しておき、これを参照しながらLabanEditorの出力データから動作記述データへの変換を行う。

## 2.3 動作のオブジェクトモデルへの再構成・編集

次に、作成された動作記述データを、オブジェクトモデルとして再構成する。まず、動作記述データを解析し、動作ごとに細分化して分割する。続いて、分割した動作記述データに基づいて「動作エレメント」が生成される。動作エレメントとは、動作記述データを格納するためのオブジェクトモデルであり、これらは、動作エレメントのテンプレート集である「動作エレメントライブラリ」から生成される(図1)。

このようにして生成された動作エレメントは、さらに「動作オブジェクト」に格納される。動作オブジェクトは、複数の動作エレメントを全体的な身体動作としてまとめ、これらを統合的に管理するためのオブジェクトモデルである。この動作オブジェクトは、「動作オブジェクトエディタ」を用いることによって、動作エレメント単位での細かな編集を行うことが可能となっている。

動作エレメントや動作オブジェクト、動作オブジェクトエディタなどについては、3章以降で詳しく述べる。

## 2.4 表 示

オブジェクトモデルにより定義された身体動作を、ディスプレイに表示する際には、動作オブジェクトから「表示用動作データ」を生成し、これを「人体モデルオブジェクト」に渡す。表示用動作データとは、各時刻における人体モデルの姿勢を、セグメントの位置と各関節の回転角度で記述したものである。また、人体モデルオブジェクトとは、人体モデルの幾何構造や形状、そして現在の関節回転角度や位置などを保持するオブジェクトモデルである。

表示用動作データを受け取った人体モデルオブジェクトは、自身の姿勢とその変化、すなわち動作を、ディスプレイ上に表示する。また、人体モデルオブジェクトの状態は、動作オブジェクトに渡され、表示用動作データの生成過程にフィードバックされる。

動作の表示については6章で詳しく述べる。

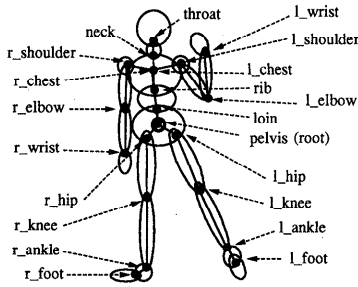


図3 人体モデルの階層構造と関節名  
Fig. 3 Joint names of human body model.

2.5 人体モデルの構造

本システムでは、図3に示すような、3自由度を持つ21の関節と21のセグメント、そしてルート（root）からなる人体モデルを使用する。rootとは、身体全体の向きや水平・垂直位置を決めるためのもので、人体モデルの腰（pelvis）の位置に設定されている。

人体モデルの各セグメントのサイズ、形状などの幾何情報は、人体モデルごとに定義でき、このデータは、独立した「身体構造データ」として保存することができる。これによって、様々な体型をした人体モデルを取り扱うことができる。

3. オブジェクト指向モデルを用いた動作の構造化

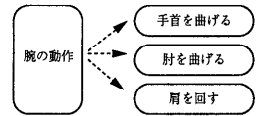
3.1 動作の分割とオブジェクト化

人体は多くの要素から構成されているため、その動作は複雑であり、これを記述するのは必ずしも容易でない。ここでは、身体動作を細分化して扱うことによって、効率的な動作記述ができるようにする。また、動作を組み合わせることで複合化するなど、動作の編集や再利用を効率的に行えるようにする。

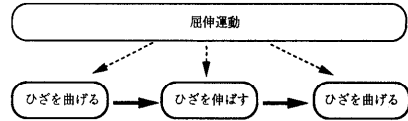
本研究では、図4に示すように、身体動作を体の部位ごと、あるいは時間方向に分割して考える。図4(a)の体の部位による分割では、「腕」「足」など、複数の関節からなる部位を1つの単位にした分割や、「手首」「肘」など、関節を単位にした分割を行う。また、同図(b)の時間方向の分割では、適当な時間間隔で、まとまりのよい動作に分割する。

分割した動作データを、図5のように、縦軸方向に体の部位、横軸方向に時間軸をとって二次元平面上に並べると、ある時刻にどの部位が動いているか、どの部位が時間経過とともに変化しているかなどが理解しやすくなる。

このようにして分割された動作は、それぞれがオブジェクト指向モデルによってカプセル化して表現され



(a) 体の部位による動作分割



(b) 時間による動作分割

図4 動作の分割

Fig. 4 Segmentation of body movement.

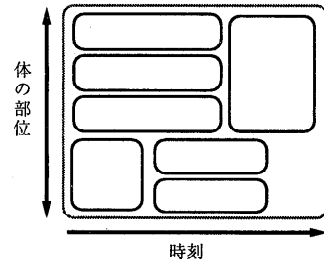


図5 分割した動作の二次元的配置

Fig. 5 Layout of segmented movement.

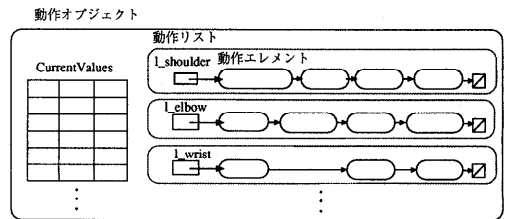


図6 動作オブジェクトの構造

Fig. 6 Structure of the movement object.

る。カプセル化したオブジェクトを「動作エレメント」と呼ぶ。動作エレメントの中には、動作に関する情報（すなわち、動作の開始時刻、終了時刻、動作のパラメータ）と、表示用動作データ生成のための手続きが格納される。動作をカプセル化することで、異なった手法によって記述された動作を統一的に扱ったり、細分化した動作単位で編集を行ったりすることを可能にする。

このようにカプセル化された動作エレメントをまとめて管理するために、オブジェクト「動作オブジェクト」を定義する。このオブジェクトは、時刻情報に応じて、内部に保持している細分化された動作の中から、その時刻に行われるべき動作を選定したり、表示用動作データ生成に必要な手続きを実行させたりする。

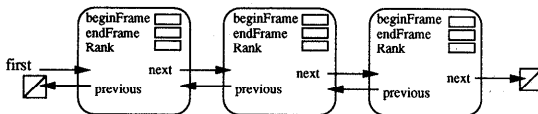


図7 動作エレメントの構造と動作リストの構成

Fig. 7 Structure of the movement elements and the movement list.

### 3.2 動作オブジェクトの構造

図6に動作オブジェクトの構造を示す。動作オブジェクトは、動作エレメント、動作リスト、そして局所変数 *CurrentValues* からなる。

動作エレメントは、前節で述べたとおり、分割された動作をカプセル化して保持するオブジェクトであり、動作に関するパラメータと、表示用動作データを生成する手続きを持っている。図6で、長円によって示されているのが、動作エレメントである。

動作リストは、動作エレメントを関節ごとに時系列に並べ、リストにして管理するオブジェクトである。動作リストは、21（人体モデルの関節の数）に1（人体モデルのroot）を加えた数だけ用意される。

局所変数 *CurrentValues* は、現時点における、各関節の回転角度や、人体モデルのrootの位置や方向などを保持するものである。時間の進行によって、*CurrentValues* の内容は更新される。

### 3.3 動作エレメント

#### 3.3.1 動作エレメントの構成

動作エレメントには、いくつかの型（タイプ）が用意されており、タイプの違いによって、内部に格納しているパラメータや表示用動作データを生成する手続きは、異なる。しかし、外部から見たときの動作エレメントの振舞いを共通にすることで、動作エレメントを統一的に扱えるようになってきている。

図7に、動作エレメントの構造と動作リストの構成を示す。すべての動作エレメントは、そのタイプにかかわらず、開始時刻 *beginFrame*、終了時刻 *endFrame*、そして、動作を算出する際の優先順位 *Rank* というパラメータを持っている。また、動作リストを構成するために、前の動作エレメントへのリンク *previous* と、次の動作エレメントへのリンク *next* を持っている。

動作エレメントの優先順位（ランク）は、動作を生成する際、計算を順序付けるために設定されており、ランク付けは、各動作エレメントタイプが表現する動作の依存関係に基づいて行っている。すなわち、ランクは動作を算出する際に演算が実行される順番であり、最高のものが0で、以下、1, 2... と数字が増えるに従って、ランクは低くなる。ランクの高い動作エレ

表1 動作エレメントのタイプ

Table 1 Types of movement elements.

タイプ	ランク	機能
Default	0	関節角度などの初期値を指定
KeyFrame	0	キーフレームにおける関節角度を指定
FramePlus	0	キーフレームにおける関節角度の増分を指定
Support	1	接地した関節を指定し、それに連動して、rootの位置を決める
LocalPlus	1	キーフレームにおけるroot位置をローカル座標で指定
FootControl	2	接地していない足のankleの位置を指定

メントによる人体モデルの姿勢の変化は、よりランクの低い動作エレメントの演算に反映される。

#### 3.3.2 動作エレメントのタイプ

主な動作エレメントのタイプとしては、表1のようなものを用意している。

表1に示した6種類の動作エレメントのうち、Default, KeyFrame, FramePlus, LocalPlusの4つのエレメントは、キーフレームで入力された動作を表現するために用いる。しかし、これらのエレメントによる、任意の時刻での関節の回転角を補間によって求める単純なキーフレーム法では、補間により生成される動作、特に足先・手先などの末端位置の動作を適切にコントロールすることが難しい。また、身体のサイズの違いにより、末端部分の位置が大きく変化する。特に脚部では、接地面との関係もあり、足先位置のわずかな変動が結果の表示において顕著に現れ、地面に足がめり込むなどの不自然な現象を生じさせる。

そこで、身体のサイズに関係なく、かつ自然な動作、特に脚部の動作を表現するために用意したのが、SupportエレメントとFootControlエレメントである。Supportエレメントは、接地した関節の動作に連動した身体全体の移動と、接地処理を行う。FootControlエレメントは、足先位置の指定によって脚部全体の動作を生成する。

以上のような動作エレメントを使って、歩行の際の脚部の動作を記述・生成する例を以下に示す。図8は、歩行動作中のあるキーフレームでの足の状態を示している。接地する方の足の姿勢（関節の回転角度）は、KeyFrameエレメント（ランク0）で指定する。rootの位置は、Supportエレメント（ランク1）で地面に接地する関節を指定することで、脚部の動作に連動して決められる。前に踏み出す方の足の動作は、FootControlエレメント（ランク2）で指定する。

この記述を元に演算が行われる際には、ランクが高

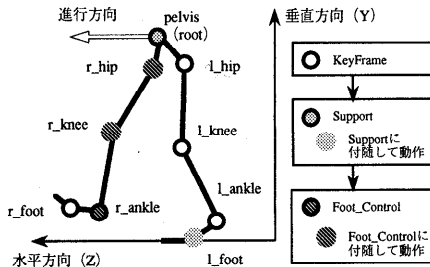


図 8 足の動作指定

Fig. 8 Specification of leg motion.

い順に、接地している脚の動作→root の位置→前に踏み出す脚の動作、という順番で計算される。以上を繰り返して、左右の足を交互に接地し前に踏み出させることで、連続した自然な歩行動作を記述することができる。

脚部で適用したのと同じ機能を、腕部や胴部の動作についても導入することは可能である。しかし、これらでは、脚部に比べてキーフレーム法を使用することによる不都合が少ないため、現時点では、いくつかのKeyFrame エlementを用いて記述している。

### 3.3.3 動作エレメントライブラリ

動作エレメントライブラリとは、動作エレメントのテンプレートを、クラスライブラリとして保持するものである。動作エレメントライブラリには、それぞれの動作エレメントのタイプについて、固有の記憶領域や表示用動作データを生成する手続きが定義される。また、すべての動作エレメントに共通の要素は基底クラスで定義され、他の動作エレメントタイプに継承される。

動作エレメントライブラリを使用することによって、多様な表示用動作データ生成手法を併用したり、記述の簡略化を行ったりすることができる。

### 3.4 動作オブジェクトの生成

最初、生成された時点での動作オブジェクトには、動作エレメントがまったく含まれていない。そこで、動作エレメントを新しく生成し、動作オブジェクトに格納していく必要がある。図9に、動作オブジェクトに動作エレメントが格納されるまでの過程を示す。以下に、これについて順に説明する。

(1) 動作オブジェクトに格納する動作エレメントのタイプやパラメータを指定する。これには、動作記述データを解析した結果や、5章で述べる動作オブジェクトエディタによる入力を使用する。

(2) 動作エレメントライブラリを参照して、(1)での指定に対応したタイプの動作エレメントが新しく生成され、パラメータの値が格納される。

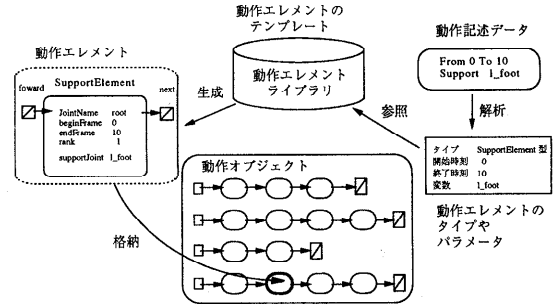


図 9 動作オブジェクトへのデータ格納

Fig. 9 Movement object.

(3) 生成された動作エレメントが動作リストに加えられる。

### 3.5 動作オブジェクトの編集

動作オブジェクトに含まれる動作エレメントを操作することによって、動作オブジェクトの編集を可能にする。ここでいう動作オブジェクトの編集とは、動作リスト内での動作エレメントの順序を入れ替えたり、動作エレメントを追加・削除したり、動作エレメント内のパラメータ値を変更したりすることである。また、複数の動作エレメントを複合化し、まとめて取り扱うようにすることもできる。

動作オブジェクトの編集についての詳細は、5章で述べる。

## 4. オブジェクト指向モデルを用いた表示用動作データの生成

ここでは、前章で述べたように構成されている動作オブジェクトによって、実際の画面表示に使用する表示用の動作データを生成する仕組みについて述べる。

### 4.1 動作オブジェクトによる表示用動作データ生成

動作オブジェクトは、時刻情報に対応した表示用動作データを算出し、出力する。図10に、その手順を示す。

(1) まず、動作オブジェクトに時刻情報が入力される。すると、動作オブジェクトは、関節ごとの動作リストをたどり、入力された時刻に対応する記述を格納した動作エレメントを選び出す。動作オブジェクトは、選択された動作エレメントのランクを調べる。

(2) ランク N の動作エレメントが活性化され、演算が行われる。動作エレメントは、算出したデータを、動作オブジェクトが持つ変数 CurrentValues に格納する。

(3) CurrentValues の値が人体モデルオブジェクトに適用される。N に 1 を加え (活性化する動作エレメントのランクを下げ)、ステップ (2) に戻る。

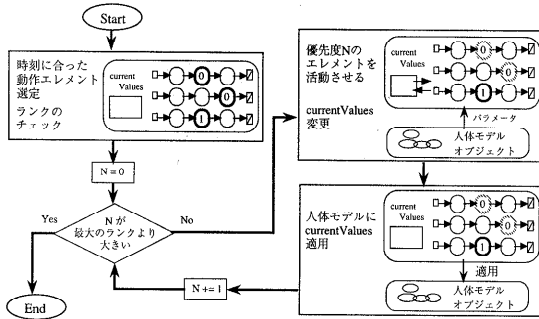


図 10 動作オブジェクトによる表示用動作データ生成

Fig. 10 Creation of the movement data by movement object.

このようにして、ランクの低い動作エレメントにおける演算には、それよりランクの高い動作エレメントによる人体モデルの姿勢の変化が反映される。

4.2 動作エレメントによる表示用動作データ生成

活性化された動作エレメントは、自らが持つパラメータや手続きを使用して、入力された時刻に対応する演算を行う。ここでは、例として、Support エレメントが行う処理について説明する。

Support エレメントは、固定する（接地している）関節を指定する際に使用される動作エレメントである。Support エレメントは、指定された関節が動作開始時刻での位置・方向のまま固定されるように、root の位置と方向を調整する。

固定する関節として l\_foot が指定されていると、Support エレメントは、現時点における l\_foot と root の相対位置と方向を算出する。続いて、この結果と、動作開始時点での l\_foot の位置・方向から、現時点で root がとるべき位置や方向を算出し、これを更新する。また、足先のセグメントが地面に平行になるように、l\_foot の回転角度を算出する。

5. 動作オブジェクトの編集システム

5.1 動作オブジェクトエディタ

動作オブジェクトエディタは、動作記述データから動作オブジェクトを作成し、オブジェクトに含まれる動作エレメントの様子を可視化して表示・編集するためのシステムである。

編集対象の動作記述データを指定して動作オブジェクトエディタを起動すると、動作オブジェクトエディタは動作オブジェクトを生成し、指定した動作記述データを読み込んだ後、動作オブジェクトの内容をウィンドウ上に表示する。図 11 に、動作オブジェクトエディタの表示画面を示す。この画面の縦軸は、各関節に対

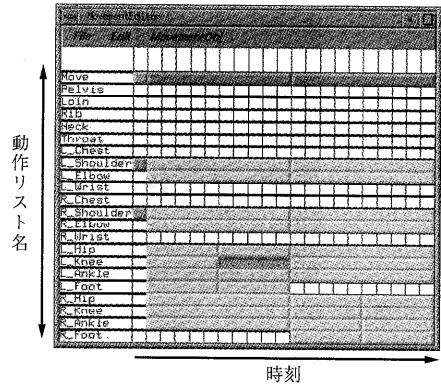
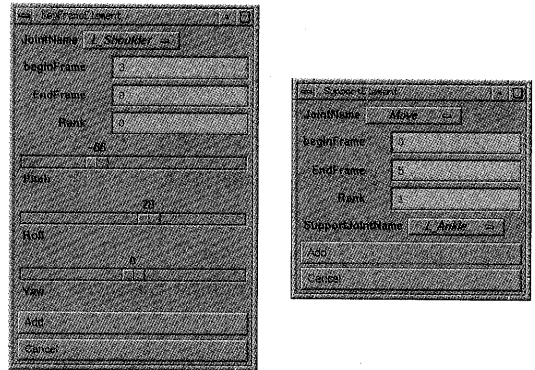


図 11 動作オブジェクトエディタによって表示された動作オブジェクト

Fig. 11 Appearance of movement object editor.



(a) KeyFrameエレメント (b) Supportエレメント

図 12 動作エレメントの編集

Fig. 12 Edit dialog for movement elements.

応した動作リスト（関節）名、横軸は時間を表している。ボタン状に表示されている長方形のバーが、それぞれの動作エレメントである。動作エレメントの型の違いは、ボタンの色で表されている。

動作エレメントに対応したボタンをダブルクリックすると、動作エレメント内の変数を編集するためのダイアログが表示される。ダイアログ内のスケールなどを操作して、動作エレメント内の変数の値を設定することができる。図 12 は、代表的な 2 つの動作エレメント、KeyFrame エレメントと Support エレメントのダイアログ画面を示している。

5.2 動作エレメント型の呼び出し

動作を新しく作成するときや、既存の動作に新しい要素を加えるときには、図 13 のダイアログにより、動作エレメント型を選択した後、動作エレメントライブラリを呼び出して動作エレメントを新規に作成する。

ダイアログ上のリストから動作エレメント型を選択

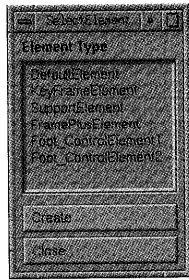


図 13 動作エレメント型選択ダイアログ  
Fig. 13 Movement element selection dialog.

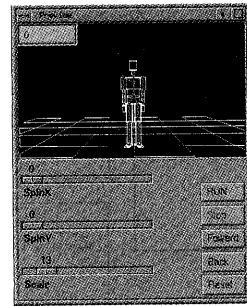


図 14 プレビュー画面  
Fig. 14 Preview window.

し、Create ボタンをクリックすると、対応する動作エレメント型のエレメントが生成され、エレメントに応じた編集ダイアログが表示される。編集後に Add ボタンを選択すると、生成された動作エレメントは、ダイアログで指定した動作リストに加えられる。

5.3 動作エレメントの複合化

動作の編集を効率化するために、複数の動作エレメントを複合化して、まとめて取り扱うことができるようにしている。複合化された動作エレメントを「複合動作エレメント」と呼ぶ。

複数の動作エレメントをマウスクリックにより選択した後、Edit メニューの Combine Elements を使って、動作オブジェクトを複合化することができる。

複合動作エレメントは、動作全体の開始時刻、終了時刻をパラメータとして持っている。複合動作エレメントの開始時刻、終了時刻を変更すると、複合動作エレメントに含まれているすべての動作エレメントの開始時刻と終了時刻が、複合化した動作全体の開始時刻、終了時刻との比率に従って変換される。これにより、複合化した動作を時間的に引き伸ばしたり、縮めたりすることが容易にできる。

5.4 動作のプレビュー

動作オブジェクトの編集結果は、図 14 のような、ワイヤーフレームによる簡易な人体モデルを使用したプレビュー画面で確認することができる。

5.5 動作オブジェクトの保存

動作オブジェクト編集の結果は、動作記述データとして保存することができる。動作記述データは、いくつかの記述形式に対応している。たとえば、表示用動作データを適当な時刻でサンプリングしてキーフレームとして記述する形式や、動作エレメントのタイプや内部のパラメータをそのまま記述する形式などがある。

6. 動作表示システム

動作表示システムは、4 章で述べたような手順で表

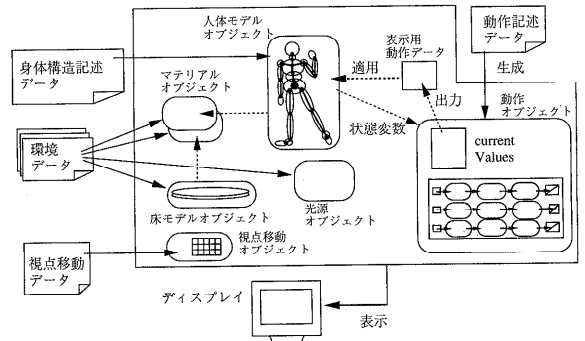


図 15 動作表示システム  
Fig. 15 Display system of human body movement.

示用動作データを算出し、それを基に、人体の動作を、照明光やシェーディングを適用した三次元グラフィックスとしてディスプレイ上に表示するシステムである。動作表示システムの概要を図 15 に示す。

動作表示システムは、表示用動作データのほかに、身体構造記述データ、環境データ、視点移動データなどを読み込む。身体構造記述データは、2.5 節で述べたように、表示に利用する人体モデルの各セグメントに関する情報を記述したものである。環境データは、床の形状や材質、光源などに関する情報を記述したものである。視点移動データは、視点の位置や方向の変化について記述したものである。

動作表示システムは、これらのデータから、人体モデルオブジェクト、床モデルオブジェクト、光源オブジェクト、マテリアルオブジェクト、視点移動オブジェクトを生成する。いずれのオブジェクトも、三次元グラフィックスをディスプレイ上に表示するために必要な手続きを保持している。

動作表示システムは、以下のような手順で動作の表示を行う。

- (1) 身体構造記述データから人体モデルオブジェクト、環境記述データから床モデルオブジェクトや光源



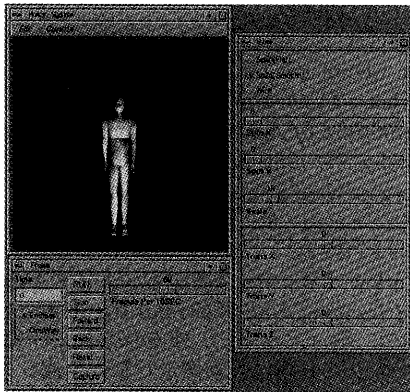


図 16 動作表示システムのインタフェース  
Fig. 16 Interface of the display system.

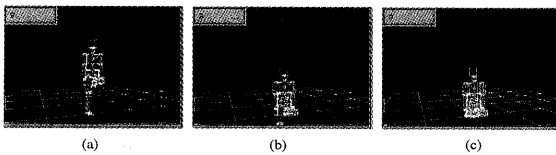


図 17 屈伸動作の生成  
Fig. 17 Generation of knee-flexing movement.

オブジェクト、視点移動データから視点移動オブジェクトを作成する。また、動作記述ファイルから、動作オブジェクトを作成する。

(2) 動作オブジェクトに時刻情報を与える。動作オブジェクトは 4 章で述べたような手順で、人体モデルオブジェクトなどを参照しながら表示用動作データを算出する。

(3) 動作オブジェクトが算出した表示用動作データを人体モデルオブジェクトに適用し、その形状を三次元グラフィックスとしてディスプレイに表示する。

動作表示システムのインタフェースは、図 16 に示すように、三次元人体モデルを表示するウィンドウと表示の制御を行う制御ウィンドウから構成されている。

## 7. システム実行例

### 7.1 動作の新規生成

動作オブジェクトエディタを使用して、膝の屈伸運動の動作を新規に生成する例について述べる。

まず、重心を支える足の動作（ここでは左足）を、KeyFrame エlement で定義する（図 17 (a)）。続いて、Support Element を使用して、左足の足先 (l\_foot) を接地させるように指定する。（図 17 (b)）。最後に、root の動きと一致するような右足の動作を、FootControl Element を使って指定する（図 17 (c)）。

以上の手順で動作オブジェクトを作成した後、その

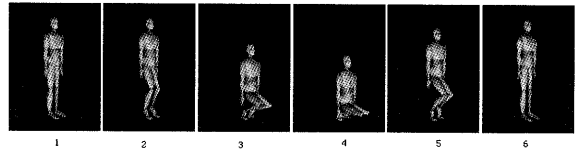


図 18 屈伸動作の表示  
Fig. 18 Display of knee-flexing movement.

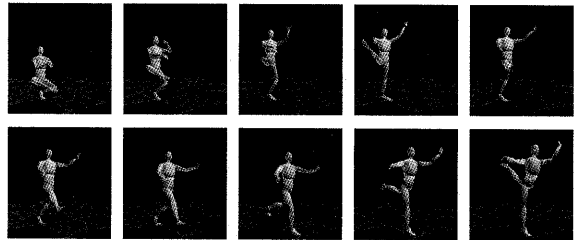


図 19 Support エlement の追加によるダンス動作の編集結果  
Fig. 19 Refined dance movement by adding "Support element".

内容を動作記述データとして保存する。その後、この動作記述データを動作表示システムに読み込ませ、グラフィックス表示させた結果を図 18 に示す。

### 7.2 既存の動作記述データの編集

ここでは、すでに作成されている動作記述データに修正を加える例について述べる。

2.2 節で述べた三次元人体モデルを対話的に操作して動作入力をするシステムを用いて、キーフレーム形式でダンスの動作を作成し、動作記述データとして保存したものがあある。その動作記述データを読み込んで動作オブジェクトを生成し、動作表示システムで表示を行うことができる。しかしこのままでは、両足が同時に平行移動したり、地面に足がめり込むなど、3.3.2 項で述べたような不自然な動きがいくつか観察された。

そこで、動作オブジェクトに Support Element を追加して、地面に接地する関節を指定し、これに root の動きを連動させる。これで、それぞれの時点で、接地した関節を中心に体全体が動作する自然な動作を得ることができる。生成されたアニメーションの一部を、図 19 に示す。この場面では、左足を軸にして立ち上がった右足を踏み出した後、右足を支えにしてポーズをとっているが、各時点で接地した足先が固定され、これに従って、root の位置と Foot の動きが自動的に調整されている。

このように、キーフレーム法などで作成された既存の動作記述データに適切な動作 Element を追加することにより、環境条件に応じた、より自然な動作に修正することができる。

## 8. おわりに

本研究では、総合的な身体動作情報処理システムの実現を最終目標として、身体動作を記述し、表示するための手法の開発と、実行制御のためのシステムの実装を行った。本システムでは、様々なタイプの動作エレメントを併用して動作オブジェクトを作成し、これを表示用動作データ生成に利用する。これによって、多様な動作記述と柔軟な編集が可能な実行制御環境を構築することができた。

動作オブジェクトエディタでは、動作エレメントを関節ごとに、時系列に並べて可視化することで、どの時刻で、どの関節がどのような種類の動作を行うかが容易に確認できる。また、動作エレメントライブラリを利用することで、様々な形式の表示用動作データ生成手法を利用することができるようになった。さらに、キーフレームによる表示用動作データ生成だけでは表現できない動作も記述・生成することができた。それに加え、動作エレメントの複合化を可能にし、複数の動作の時刻情報を一括して変更することができた。

残された課題には、以下のようなものがある。

まず、現在のエレメントタイプだけでは、身体、特に腕部や胴体などの複雑な動作を生成する際、使用する動作エレメントの数が多くなる傾向がある。そこで、表現力を強化した動作エレメントのタイプを追加し、少ないエレメントで多様な動作が表現できるようにする必要がある。

また、動作を編集する際、動作エレメント内のどのパラメータを変更すればどのような動作が生成されるのかが判断しにくい。そこで、動作エレメント内のパラメータ変更にインタラクティブに連動するプレビューを用意するなど、動作編集の操作性の向上をはかる必要があると考える。

このほかに、動作エレメントライブラリを利用しやすくするために、動作エレメントの機能を可視化したボタンを利用するなど、ユーザインタフェースの工夫が必要であると考えている。

謝辞 本研究の一部は文部省科学研究費補助金(課題番号 07207116)の援助によって行われた。記して謝意を表する。

## 参 考 文 献

- 1) 八村広三郎: 身体運動情報の処理, 情報処理学会研究報告, 人文科学とコンピュータ, 90-CH-6,

pp.1-8 (1990).

- 2) Calvert, T.: *Composition of Realistic Animation Sequences for Multiple Humman Figures, Making Them Move: Mechanics, Control, and Animation of Articulated Figures*, Badler, N.I., Barsky, B. and Zeltzer, D. (Eds.) pp.35-50, Morgan Kaufmann (1991).
- 3) Hutchinson, A.: *Labanotation, Theater Arts Books* (1977).
- 4) Kerlow, I.V.: *The Art of Computer Animation and Imaging*, Van Nostrand Reinhold (1996).
- 5) 筒口 拳, 末永康仁, 渡部保日児, 下原勝憲: 3次元シーン内の人物像歩行動作生成システム, 情報処理学会論文誌, Vol.38, No.4, pp.787-796 (1997).
- 6) Rose, C.F., et al.: Efficient Generation of Motion Transitions using Spacetime Constraints, *Proc. SIGGRAPH'96*, pp.147-154 (1996).
- 7) <http://ece.uwaterloo.ca/~h-anim>
- 8) 吉田康行, 松岡洋介, 八村広三郎: 舞踊譜 Labanotation に基づく身体運動の処理—譜面読取り LabanReader と譜面エディタ LabanEditor, 情報処理学会研究報告, 人文科学とコンピュータ, 98-CH-38, pp.61-68 (1998).

(平成 10 年 8 月 31 日受付)

(平成 11 年 1 月 8 日採録)



平松 尚子

1973 年生。1998 年立命館大学大学院理工学研究科情報システム学専攻修士課程修了。同年ミノルタ(株)入社。立命館大学大学院在学中に身体運動情報処理についての研究に従事。電子情報通信学会会員。



八村広三郎(正会員)

1948 年生。1971 年京都大学工学部電気工学科第二学科卒業。1976 年同大学院工学研究科博士課程修了。工学博士。国立民族学博物館第五研究部助手, 京都大学情報処理教育センター助教授, 同大学工学部助教授を経て, 1994 年より立命館大学理工学部情報学科教授。画像処理, コンピュータグラフィックスの教育・研究に従事。画像検索, 感性情報処理, マルチメディアシステム等に興味を持つ。電子情報通信学会, 画像電子学会, 日本 ME 学会各会員。