

記号による記述を用いた身体モデルの動作制御

高木龍也 岡田義広 田中譲

北海道大学大学院 工学研究科

{tatsuya, okada, tanaka}@huee.hokudai.ac.jp

本稿では, Labanotation という動作の記述体系を用いた動作編集とアニメーションのシステムについて述べる. 人間の動作を CG で製作しようとした場合, そのパラメータの多さから多大な手間が必要となる. このシステムは記号により動作を記述することで, 動作指定の手間の削減を目的とする. また動作を階層的に扱うことで, ユーザの動作編集を助け, 動作の再利用性を向上する. 良く利用される動作はライブラリとして用意する. これらの動作編集の支援により, ユーザは動作のライブラリや自身又は他人が製作した動作を適宜利用し, 必要ならばそれらを再編集して利用し動作の記述を行なうことができる.

Motion Control of a Human Model Using Labanotation

Tatsuya Takagi, Yoshihiro Okada and Yuzuru Tanaka

Graduate School of Engineering,
Hokkaido University

North 13 West 8, Sapporo 060, Japan

This paper describes an animation system that enables users to edit human motions using Labanotation. Labanotation is a two dimensional graphical notation system for recording human movements so that users can define human motions easily by editing its graphical symbols. The graphical interface of this system has been developed using the IntelligentPad system. The IntelligentPad system provides media objects called pads which have a unique function and an image like a sheet of paper. It is possible to compose a composite pad by pasting one pad onto another pad. This animation system allows users to define and store partial motions as a composite pad. Users can reuse and re-edit pads.

1 はじめに

本稿では, Labanotation [3] という動作の記述体系を計算機上に導入することによって, CG による動作製作における手間の削減を目的とした, 動作編集とアニメーションのシステムについて述べる.

現在, CG による映像が一般的なものになったとはいえ, その製作には依然として多くの手間を必要とする. 特に動きのある人間を製作するには, その関節の多さからパラメータの調整に多くの労力と時間を必要とする. 関節角度を与えることで動作を制御するのではなく, 言葉や記号などのより抽象的な表現を用いて動作を指定し, これを関節角度に変換することで動作を制御することができれば, ユーザの手間を削減することができる.

本研究では Labanotation という身体動作の記述体系を用いることによって動作を記号的に扱う. 動作を記録するための記法は複数存在するが, 最も一般的であるのが Labanotation である. Labanotation では staff と呼ばれる音楽の譜表に相当する物の上に, 動作を表す記号を配置していくことで動作を記述する.

Labanotation を用いて同様のことを行なおうとした研究には [1, 2] などがあつた. しかし記号による動作の記述とはいえ, 動作を一から製作するのは多大な労力を必要とする. そのため [2] では基本動作の組合せによるマクロ的な動作が利用された. しかし動作のマクロ定義を行なうにはテキストのコードを編集する必要があつたため, ユーザが容易に動作を編集できるものではなかつた.

本研究では Labanotation による動作記述を行なうシステムを IntelligentPad 上に構築する. IntelligentPad はパッドと呼ばれる様々な機能部品をユーザが画面上で組み合わせていくことで, より複雑な機能を合成することができるシステムである. この IntelligentPad 上で Labanotation の記号や staff をパッドとして実現した. そして Labanotation に部品の重ね合わせという奥行きを付加することにより, 動作を階層的に扱う機構を導入した. これによりユーザは作成した一連の動作をマクロと同様に扱い, 再利用することが可能になる. ユーザは動作のライブラリや自身又は他人が製作した動作を適宜利用し, 必要ならばそれらを再編集して利用し動作の記述を行なうことができる.

2 Labanotation

Labanotation は動作を記述するための記法として最も有名なものである. 動作の記述は動作などを表す記号を譜表 (staff) に記述することで行なう. 図 1 に示すように, staff には垂直方向に線が引かれており, 各線の間の領域 (カラム) はそれぞれ図に記してある体の部位の動作を記述する領域である. 体の各部位の動作は各部位に対応するカラムに記号を配置することで行なう. staff の垂直方向は下から上に時間の経過を表しており, 記号の下端と上端がそれぞれ動作の開始する時と終了する時を表す.

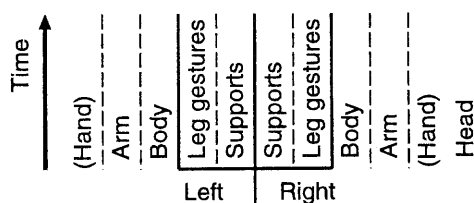


図 1: staff

最も基本的な記号は図 2 に示す direction と呼ばれる記号である. この記号は例えば腕のカラムに記述されると, 腕の付け根 (肩) に対して腕の先 (手) を指定した方向に向ける動作を示す. 水平面内の方向 (direction) は記号の形状によって表し, それぞれの方向に対してそれを表す形状の内部の模様で高さ (level) を表す.

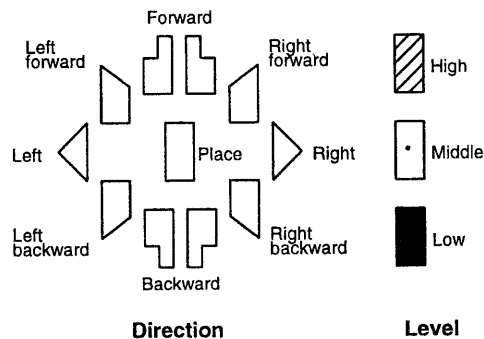


図 2: direction

このような記号を用いて例えば図 3 のように動作の記述を行なう. Backward を表す形状を Low を表す黒で塗った記号を staff の Left arm を表すカラムに記

述することで、左腕を後方低くに向ける動作を記述する。staffの垂直方向は下から上に時間の経過を表すので、記号の下端がその動作の開始を、上端がその動作の終了を表す。先の例では記号の下端における姿勢は不明であるが、その姿勢に関係なくその姿勢から左腕を後方低くに向ける動作を始め、記号上端における時間に左腕は後方低くに向けられその動作を終了する。Labanotationにおいて記号がないことは動作がないことを表す。したがってこの動作終了後、次の動作が記述されている位置までその姿勢は保たれる。

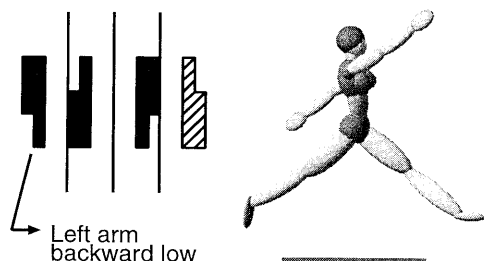


図 3: Labanotation による動作の記述例

3 IntelligentPad

Labanotation による動作の記述を行なうための GUI として IntelligentPad を使用した。

3.1 IntelligentPad の概要

IntelligentPad ではすべての情報を紙のイメージを持つ「パッド」と呼ばれる可視オブジェクトによって表現する。例えば数値や映像などのデータや、それら进行处理するためのプログラムもすべてパッドとして表現される。パッドには、他のパッドとデータのやりとりをするためのスロットと呼ばれるデータの受渡し口がある。ユーザは各パッドを画面上で直接貼りあわせることで自在にレイアウトができると同時に、このスロットを介してデータ連携を行なうことでパッド同士の機能合成を行なうことが可能である。これによりそれぞれのパッドが個別に持つ機能を合成し、より複雑な機能をもつ合成パッドを作ることができる。そしてこれら合成されたパッドは一つのパッドとみなすことができ、さらに他のパッドとの貼り合わせ操作を積み重ねていくことによって、より複雑な機能をもったパッドの合成や、ア

プリケーションの作成ができる。

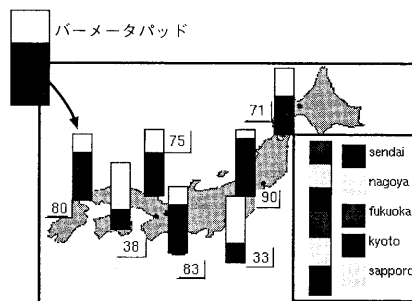


図 4: IntelligentPad

3.2 IntelligentPad の使用による利点

Labanotation による動作の記述を行なうためのシステムを、IntelligentPad 上で構築することで、スクロールバーや入力ボタンなどの基本的な機能を部品の貼り合わせのみで利用することができた。また IntelligentPad の特徴である機能部品の貼り合わせという簡単な操作で動作の記述を実現することができた。また IntelligentPad では機能部品を階層的に組み合わせることで複雑な機能を合成することができる。この機構により、動作を階層的に組み合わせることで複雑な動作を作成できる。このようにして作成した動作の階層構造はそのままパッドの階層構造になっているため認識しやすい。これらのことは記述された動作の再利用性の向上につながる。このように IntelligentPad の持つ特徴は Labanotation による動作記述システムと愛称が良いと思われる。

4 パッドを用いた動作編集

4.1 記号の部品化

IntelligentPad 上で Labanotation による動作の記述を行なうために、記号を部品化し LabanSymbolPad(以下 SymbolPad) というパッドを作成した。SymbolPad の一つである LabanDirectionPad(以下 DirectionPad) は、2章で説明した direction を表す記号をパッド化したもので、スロットに記号の情報を持つ。DirectionPad は direction や level ごとに別の部品が用意されているわけではなく、ユーザのクリックによってスロットの direction と level の値を変え、それにともない記号も変

化させる。例えば図5に示すように DirectionPad の左上部分をクリックすると direction を left forward に変更でき、繰り返し同じ部分をクリックすることにより level を次々と変えることができる。

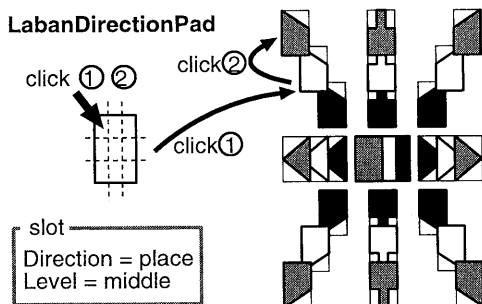


図 5: 記号の部品化

4.2 staff の部品化

staff は LabanStaffPad (以下 StaffPad) として部品化した。StaffPad は上に貼られた SymbolPad がグリッド上に並ぶように位置と大きさの調節を行ない、SymbolPad のスロットから記号の情報を収集し SymbolPad の位置情報とともに自分のスロットに保持する機能を持つ。

これらの部品に、スクロールさせるための部品などを組み合わせることで、Labanotation による動作の記述を行なうためのインタフェースを実現している。

ユーザは SymbolPad を StaffPad の適当な位置に貼り、SymbolPad をリサイズして動作の長さを決め、クリックして希望の記号に変えるという、マウスのみによる一連の操作を繰り返すことによって動作を記述していく。

4.3 動作の階層化

StaffPad に貼られるのは SymbolPad ばかりではない。複数の SymbolPad が貼られた StaffPad を、さらに StaffPad に貼ることも可能である。この場合も上に貼られた StaffPad がスロットに持つ記号の情報は下の StaffPad のスロットに保持される。

図6のように StaffPad を階層的に貼り合わせることによって、動作を階層的に扱うことが可能となる。

これにより次のような利点が得られる。

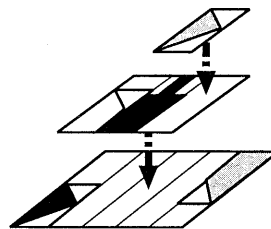


図 6: 動作の階層化

- 動作の階層構造はパッドの階層構造として認識できるため、動作がどのように構成されているか理解しやすい。
- 過去に製作した動作の一部分を再利用できる。
- 一連の動作を表す合成パッドを動作のライブラリとして利用できる。

ユーザは動作のライブラリや過去に製作した動作を適宜利用し、必要ならばそれらに変更を加えて再利用して動作を編集することができる(図7)。

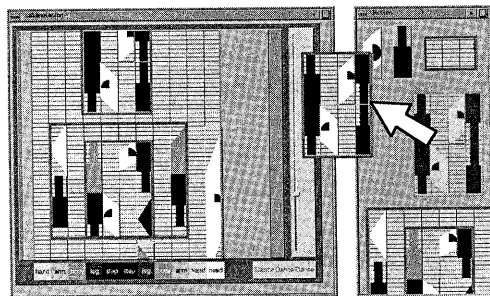


図 7: パッドを用いた動作編集

5 Labanotation の解釈

5.1 Labanotation データからの関節角度の生成

パッドを用いた動作の編集の結果、最下層の StaffPad のスロットにはすべての記号のデータが保存される。このデータを次のような、体の部位、動作開始フレーム、動作終了フレーム、記号情報、の形式でファイルに出力する。

```
# Left Arm
1 12 direction forward low
```

```

14 20 direction place middle
# Left Leg
1 12 rotation right 45
20 29 direction forward high

```

この Labanotation データから動作を生成する。動作は動作終了時の関節角度と動作開始時の関節角度からキーフレーム補間により生成する。動作終了時の関節角度は、各体の部位について各記号に対応する関節角度をあらかじめ決めておくか、動作開始時の関節角度を用いて決定する。このため動作開始時の関節角度によって同じ記号でも異なる動作が生成される。

例えば先の Labanotation データならば、Left Arm に対して forward low という動作をさせる場合、動作終了時(第 12 フレーム)の肩の関節角度(以下角度の単位は $^{\circ}$)を $(0, 45, 0)$ 、肘の関節角度を 0 と決めておき、これと動作開始時(第 1 フレーム)の各関節角度をキーとして補間することにより動作を生成する。2章で説明したように、Labanotation において記号がないことは動作がないことを表すので、この動作終了後、次の動作が記述されている第 14 フレームの前までその姿勢は保たれる。次の place middle では肘関節角度が 150 と決められているため第 14 フレームから第 20 フレームへかけて 0 から 150 へ補間される。この時 place middle は肩関節角度に影響しないので肩関節角度は前の値を保つ。

フレーム番号	肩関節角度	肘関節角度	
20	$(0, 45, 0)$	150	↑
⋮	⋮	⋮	place middle
15	$(0, 45, 0)$	37.5	
14	$(0, 45, 0)$	18.75	↓
13	$(0, 45, 0)$	0	
12	$(0, 45, 0)$	0	↑

5.2 重心の移動

これまでの処理ですべての関節角度が生成される。しかしこれではその場で体を動かすだけで空間を移動できない。Labanotation において重心の移動は Supports カラムにおいて記述される。そこで多関節モデルの制御パラメータとして各関節角度の他に腰の位置(重心とする)を設け、これを Supports カラムの記号で制御する。

Labanotation における Supports カラムの記述は他のカラムに比べて特殊である。図 8 に示すように Supports カラムに direction 記号が書かれた場合、その方向に足を踏みだし重心を移動することを表す。また他のカラムでは記号がないことは動作がないことを表すが、Supports カラムでは記号がないことは接地をしなくする動作を表す。このため同図のように動作終了時点で左足は接地しない。

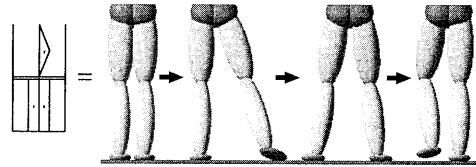


図 8: Supports

図 8 に示す動作のように Supports カラムにおいて一つの記号でなされる動作は複数の基本動作から成り立っていると思われる。そこで Labanotation に変更を加え、Supports カラムには足が接地しているかいないかを記述することとする。パッドを合成することにより複雑な動作を作成できることは先に述べたとおりであるから、従来の Supports カラムにおける動作は接地と Leg gestures カラムにおける動作の合成で記述する。例えば同図のように右足を右に踏み出すには右足を右に出し、体を右に押し出すために左足を左に(蹴り)出す。この間左足が接地していることを Supports カラムで記述する。

以下では接地の記述から腰の位置を決定する処理について、どちらかの足が接地している場合とどちらの足も接地していない場合に分けて説明する。

5.2.1 足が接地している場合の体の位置

Supports カラムに記号が記述されている場合、記号が記述された方の足は接地していることを表す。この場合、接地している足の爪先とかかとの位置で比較を行ない、低い方が地面と同じ高さになるように腰の高さを決定する。腰の水平面上の位置は、接地点が固定されるように決定する。図 9 に示すように、各関節角の変化によって、爪先とかかとのうち接地している方の位置が $d\vec{x}$ だけ変化した時、腰の位置を $-d\vec{x}$ 移動させることで接地点を固定する。

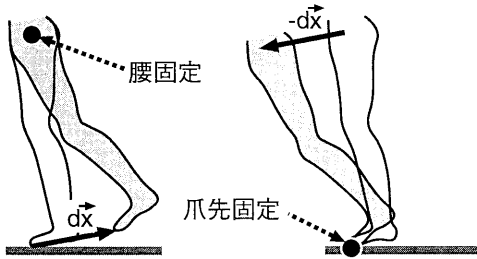


図 9: 足が接地している場合の体の位置

5.2.2 両足とも接地していない場合の体の位置

どちらの Supports カラムにも記号が記述されていない場合、どちらの足も接地していない、つまりジャンプしていることを表す。ジャンプ中にも Leg gestures カラムなどに記号を記述することで動作することが可能なので、ジャンプ中に姿勢が変化し、ジャンプ終了時の腰の高さが変わる可能性がある。そこでジャンプ開始時と終了時の腰の高さを考慮し、ジャンプ中は鉛直方向に重力による等加速度運動をすると仮定すると、ジャンプの期間からジャンプ中の腰の高さを決定することができる。腰の水平面上の位置は、ジャンプ中、等速度運動するとしてジャンプ直前の速度を維持するようにして決定する。

以上のように Labanotation データを解釈することにより生成した関節角度と腰の位置により作成したアニメーションの 1 シーンを図 10 に示す。

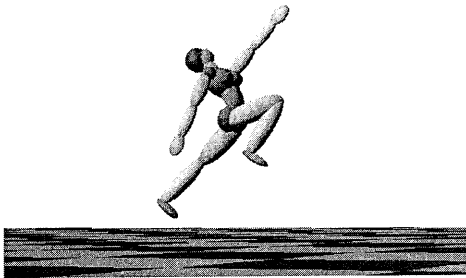


図 10: アニメーションの 1 シーン

6 おわりに

本稿では、Labanotation という動作の記述体系を計算機上に導入することによって、CG による動作作成における手間の削減を目的とする、動作編集およびアニメーションのためのシステムについて述べた。

動作は体の各部位の動きの合成と見ることができるとともに簡単な動作の連続と見こともできる。Labanotation は人間の動作を体の部位で分割し、それをさらに時間で分割した基本動作となるものを記号として体系化したものである。このため staff は横(体の部位)と縦(時間)の広がりを持っており、動作の記述はこの 2 次元平面を記号で埋めていく作業となる。

動作を 2 次元平面として表現したものが Labanotation であったのに対し、本研究では、ここに部品の重ね合わせという奥行き情報を付加することにより動作を階層的に扱い編集する機構を導入した。これにより動作の再利用性を向上させることができた。またこの時、動作の階層構造は IntelligentPad システムにおけるパッドの階層構造として認識できるため、動作の構造が分かりやすく編集も行ないやすい。ユーザは動作のライブラリや自身又は他人が製作した動作を適宜利用し、必要ならばそれらに変更を加えて再利用して動作を編集することができるようになった。

今後は、ユーザが複数の記号で記述した動作を一つの新しい記号で表わせるようにし、記述された複合動作の構造を意識することなくそれを再利用可能にする機構を実現する予定である。また IntelligentPad の 3 次元拡張である IntelligentBox 上でこのシステムを実現することも考えたい。

参考文献

- [1] Barenholtz, J., Wolofsky, Z., Ganapathy, I., Calvert, T.W. and O'Hara, P., "Computer Interpretation of Dance Notation", *Proceedings of the 3rd International Conference on Computing in the Humanities*, pp. 235 - 240 (1977)
- [2] Calvert, T.W., Chapman, J. and Patla, A., "Aspects of the Kinematic Simulation of Human Movement", *IEEE Computer Graphics and Applications*, 2, 9, pp. 41 - 50 (1982)
- [3] Hutchinson, A., *Labanotation*, Theatre Arts Books, New York (1970)