アニメーション機能を装備したグラフィックソフトウェアの開発

坂本眞人[1)]・田辺真弥[2)]

Development of a Software for Graphics Equipped with Animation Function

Makoto SAKAMOTO[1)] and Shinya TANABE[2)]

Abstract

In this paper, we propose a software for graphics equipped with the function of animation, which if we draw a simple character by the mouse, we can animate the character on the screen. We call this software "Motion Paint". We think that many users will have some familiarity with the Motion Paint, and that it will be useful for a means of communication in the case of infant education, social welfare, and so forth.

Key Words : computer animation, affine transformation, in-betweening method, neural network

1．はじめに

コンピュータグラフィックス (CG) は、1963年にCGの父と言われる "Sutherland" がCGやCADの原型にあたる "Sketchpad" を開発したのが始まりと言われ、昨年でちょうど 40年の時が過ぎたことになる。以来、CGは目覚しい発展を遂げ、今日CGによるアニメーション作成はたいへん身近なものになってきている。Macromedia 社の "Flash" や Adobe 社の "Premiere" など、様々なCGソフトウェアの登場が映像作成の機会を広げ、アニメーション作品を閲覧する機会が多くなった。映画やテレビといった商業的な分野ではもちろんのこと、個人でも趣味としてアニメーション作成を楽しむことができるようになってきた。しかし、CGアニメーション作成の多くは複雑な操作と知識に加えて十分な時間とモチベーションを必要とし、初心者にとっては困難な作業である。

本稿では、誰もが気軽にアニメーションを作成できることを目的として開発したグラフィックソフトウェアを紹介する。従来のアニメーション作成ソフトウェアが詳細な表現のために多くの作成時間をかけるのに対し、本システムでは詳細な表現よりも簡単にすぐアニメーションが作成できる点を重視している。

著者は、同システムを "MotionPaint" と名付け、少ない説明で直感的に扱えることを目指している。このような直感的に扱える人に優しい技術の研究は各分野で行われ[1][2]、例えば、誰もが簡単に3次元モデルを生成できる手法を提案した『手書きスケッチによるモデリングシステム Teddy』[1] はソフトウェアの在り方という点で大変参考になった。

"MotionPaint" はその特性から、幼児教育や社会福祉等のコミュニケーションツールとし

1) 情報システム工学科助教授
2) 情報システム工学科学部生

ての応用、ビデオコンテ等の短時間でのラフなイメージ作成への利用などが期待でき、その他にもプログラミングについての関心度を高める広告塔的な役割も期待できる。なお、“MotionPaint”はＪＡＶＡの開発環境“JDK1.4”を用いて開発された[7]。

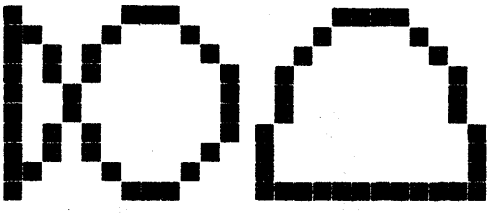
2．原理

2．1．入力

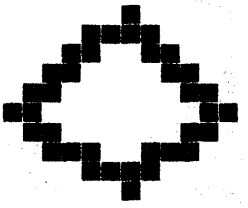
マウス等のポインティングデバイスを用い入力された手書き自由曲線の形状は点の集合として記録される。さらに、その形状は描画領域に対するワールド座標系から、点集合固有のローカル座標系と位置情報に変換される。

2．2．形状の比較

形状の比較にはニューラルネットワークを用いた［5］。ニューラルネットワークは、人の脳を模倣した学習機械である。今回は、入力層120、中間層50、出力層5の階層型ニューラルネットワークを用い、学習は誤差逆伝播法（BP法）を使い3000回行った。現在、サンプルデータとして用いている形状は、図1のような“魚”、“くらげ”、“エイ”の3種類である。



（a）魚　　（b）くらげ



（c）エイ

図1：サンプルデータ

入力された点集合は図1のサンプルデータと比較される。また、この他に“置物”を置くことができる。“置物”か否かの判断はニューラルネットワークではなく形状の大きさで下される。

2．3．アニメーション作成

アニメーションを描画する際は“java.awt”に含まれる“Polygonクラス”［6］を用いて形状を定義する。“Polygonクラス”は点集合をそのX値、Y値、点の数から多角形として定義し、プログラム上は変数として扱える。これにアフィン変換［3］を施してアニメートさせる。また、必要に応じて“Polygon変数”のセル画をいくつか作成する。以下、“魚”の動きを例に説明する。

（1）準備

点集合からPolygon変数“SH_fish”を生成する。また、“SH_fish”が尾ひれを動かす時とターンする時のセル画を用意する。今、入力データ“  ”を“魚”と判断された形状“SH_fish”とする。尾ひれの動きは、関数［4］

$$x' = r\sin\frac{x}{2r}\pi \qquad y' = r\sin\frac{y}{2r}\pi$$

を使いて作成する。一般に、同関数は平面を表す画像から球面への変形を行う時に用いられるが、このうち利用するのはX方向のみである。そして、曲率を変化させるために以下の式のように係数mを加える。なお、mには適当な値を試行錯誤的に与える。

$$x' = r\sin\frac{mx}{2r}\pi$$

まず、点集合の各々を図2のように平行移動させ、その後上式を用いて各点を写像させ、元の位置に平行移動させる。

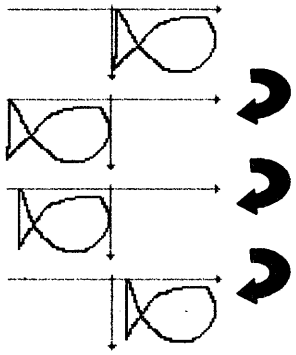


図２：点集合の平行移動

これらの処理によって得られた点の集合と、元の点集合との間で中間画像を数枚作り、尾ひれの動きのセル画とする（図３参照）。

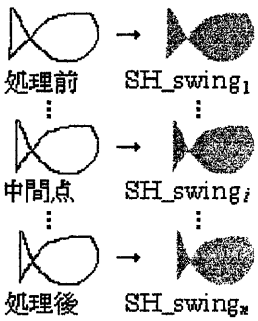


図３：SH_swing の作成

中間画像は処理前と処理後の間の点集合を線形内挿で求め、それを基にいくつかPolygon変数“SH_swing”を作成した。

ターン時のセル画は、2 つの絵を同時に描画することで表現している。点集合を⊿y だけ Y 方向正に平行移動させる。点集合のうち Y の値が点集合の横幅を超えたものを Pf、残りを Pb とし、Pf を水平方向に反転させる（Pf´）（図４）。

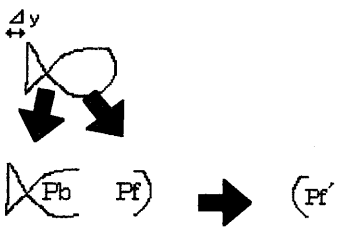


図４：ターン時のセル画の作成

このPb、Pf´をもとにPolygon変数SH_b1、SH_f1を作成する。そして SH_b1、SH_f1の順で描画する（図５）。

図５：SH_b1、SH_f1の描画
（実際は、SH_b1とSH_f1は同じ色）

こうして移動する距離を 2⊿y、3⊿y、…と繰り返しターン時のセル画を数枚つくる（図６）。

図６：ターン時のセル画の作成

（２）平行移動

“SH_fish”の位置“L(Lx,Ly)”は次の様なアルゴリズムで計算される。ただし、“Vx”は X 方向に１コマで移動する距離、“Vf”は加速の基準値、“direction”には初期値として 1 が代入されているものとする。また、“random()”は 0～1 の乱数を発生させる関数である。

```
Lx=Lx+direction*Vx;
Ly=Ly+sin θ *Vx;
Vx=Vx-A;
if(Vx<lim){
    rand=random();
    Vx=(rand+0.5)*Vf;
    θ=rand* θ f-( θ f/2);
}
```

X 方向には等加速度で減速しながら移動する。“Vx”がある値“lim”以下になると、“SH_swing”を描画し、その後“Vx”を十分大きな値に更新する。それと同時に θ も更新す

る。壁に当たった場合は、“direction”に-1を代入して進行方向を反転させる。その際ターン時のセル画を描画する。なお、“Vf”、“θf”、“lim”には適当な値を試行錯誤的に与える。

Y方向はX方向の sinθ倍で推移する。“SH_fish”が上下の壁にあった場合は、描画領域の中央に向かうようθを変化させる。ここで、“SH_fish”の縦幅、横幅に考慮する。

(3) 拡大縮小と回転

“SH_fish”がターンして向きが変わると、“SH_fish”をX方向に-1倍し、“SH_fish”を水平方向に反転させる。回転は常に角度θだけ回転させている。ただし、拡大縮小や回転はともに最初に作成した“SH_fish”に対して行うものであり、拡大縮小及び回転後の情報は“SH_fish”には上書きしない。

(4) セル画の使用

描画対象を“SH_fish”から“SH_swing”やターン時のセル画に変えるタイミングを紹介する。“SH_swing”は“Vx”が“lim”以下になった直後から数コマにわたって描画される。この時、“SH_swing”に対してもアフィン変換による平行移動を施す。“SH_swing”が描画されている数コマも“Vx”は減少し続けているので、“lim”はこのことも考慮して決めなければならない。ターン時のセル画は“SH_fish”が壁にぶつかった時に描画する。

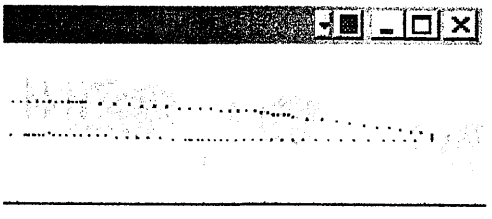


図7：セル画の描画（各黒点はコマの中心）

3．実装

本研究では、以下の表1の環境下でプログラムを作成し、実行した。図5は実際に描いた例である。

OS	Windows2000
プログラム言語	Java
ポインティングデバイス	マウス

表1：開発環境

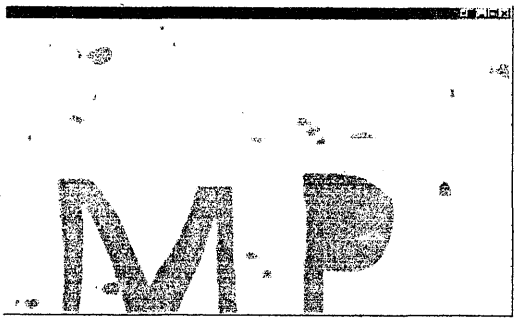


図8：出力例

4．考察

プログラム作成や実装を通じ、いくつかの改良点や問題点が浮上した。その中でも特に問題を感じたのは、形状の判断の是非である。つまり、意図した形状とは違ったアニメーションが割り当てられてしまうケースが多々あった。これには形状判断の方法、我々の形状に対する意識、サンプルデータの妥当性、ポインティングデバイスなどの原因が考えられる。

5．アンケート

“MotionPaint”の操作性の検証を兼ねたアンケートを行った。対象人数は23人である。ここではそのうちの問1から問5までの結果を紹介する。

問1．形は思ったとおりに判断されましたか。

1．思ったとおり	2
2．まずまず思ったとおり	13
3．どちらともいえない	4
4．あまり思ったとおりではない	2
5．ぜんぜん駄目	2

問2．上で1以外だった方に質問です。その原因は何だと思いますか。(複数選択可)

1．形判断プログラム	11
2．マウスが描きづらい（ポインティングデバイス）	8
3．自分の絵心	8
4．わからない	0
5．その他 ()	2

問3．操作は簡単で、すぐに使い方が理解できましたか。

1．わかりやすかった	18
2．ある程度はわかりやすかった	5
3．どちらともいえない	0
4．少しわかりづらかった	0
5．わからなかった	0

問4．おもしろいと感じましたか。

1．おもしろい	16
2．そこそこおもしろい	7
3．わからない	0
4．そんなにおもしろくない	0
5．おもしろくない	0

問5．アニメーションは自然でしたか。また、駄目な点はどこですか。

1．自然	9
2．そこそこ自然	10
3．どちらともいえない	1
4．少し不自然	1
5．不自然	2

問1．『形は思ったとおりに判断されましたか。』では個人差はあるものの、多くの人が形とアニメーションのずれを感じていると分析できる。その原因として挙げてもらった問2．『上で1以外だった方に質問です。その原因は何だと思いますか。(複数選択可)』をみると選択肢1～3がほぼ均等に多い。選択肢1は第4章の『形状判断の方法』、『サンプルデータの妥当性』に、選択肢2は『ポインティングデバイス』に、選択肢3は『我々の形状に対する意識』にそれぞれ対応していると考えることができる。だとすれば、形とアニメーションのずれは上記の全ての要素が原因で起こっているといえる。やはり、改良点をしぼるのではなく、要素全てを視野に入れ検討する必要があるようだ。また、選択肢5の『その他』として「傾きに弱い」等の意見が寄せられていた。確かに、手書き入力である以上、多少の傾きは発生して当然である。このことも考慮にいれる必要があると思われる。

これまでのプログラムでは、歪な魚を防ぐために閾値を高めに設定していたが、今回のアンケートをみてわかる通り、歪な形が多くなってでもたくさんの可能性を受理させる方が心地よい操作性につながると推測される。

問3．『操作は簡単で、すぐに使い方が理解できましたか。』問4．『おもしろいと感じましたか。』ではある程度満足のいく結果が得られた。これは当初の目標の一つである『誰もが気軽にアニメーションを作成できる』が達成されていると言える。

問5．『アニメーションは自然でしたか。また、駄目な点はどこですか。』では選択肢1『自然』、選択肢2『そこそこ自然』が多数を占めていた。が、選択肢3～5の『どちらともいえない』『少し不自然』『不自然』という意見もあり、改良の余地があることは否めない。原因として挙げられた『駄目な点』では、「色が1色だったから」、「大きさに合った動きが欲しい」、「微妙な形の判断がされていないから」、「各々にスピードの違いがないから」、「動きが単調だから」、「バイタリティーが感じられない」などがあった。

以上から、形状の判断、アニメーションの質には問題があるものの、楽しく簡単に操作できていると思われる。

6．おわりに

本稿では落書き感覚でのアニメーション作成システムの一つを紹介した。開発には“JDK 1.4”を用いてコードは1900行程度である。一般に、落書きや走り書きを通して新たなアイデアが浮かんでくることは多い。また、誰でも気軽に使えるという点からも、このようなソフトウェアの必要性は十分あると考えられる。

今後は、各機能の追加と動きの種類の充実をはじめ、操作性の向上、各アルゴリズムの見直しなど、より洗練されたプログラムにしていくことが課題である。終わりに、アンケートその他に協力して頂いた諸氏に深く感謝の意を表する。

参考文献

［1］五十嵐 健夫、松岡 聡、田中 英彦、“手書きスケッチによるモデリングシステムTeddy” 情報処理学会プログラミングシンポジウム, 箱根小涌園(神奈川県足柄下郡), pp.31-38, 2000.

［2］椎尾 一郎、山本 吉伸、"コミュニケーションツールのための簡易型AR システム" インタラクティブシステムとソフトウェア VIII（日本ソフトウェア科学会 WISS2000） pp. 117-124, 近代科学社, ISBN4-7649-0285-0, 2000.

［3］大沢 裕、河原 哲夫、後藤 道子、 坂上 勝彦、佐藤 誠、曽根 光男、長 幸平、鳥脇 純一郎、西村 宣起、福江 潔也、増田 功、松岡 龍治、八木 伸行、 安岡 善文、山本 和彦、山本 正信、 “イメージプロセッシング〈画像処理標準テキストブック〉” CG－ARTS 協会、1997.

［4］伊藤 裕之、大平 智弘、岡野 宏美、長船 龍太郎、小城 浩之、北山 由紀雄、倉田 和夫、源田 悦夫、後藤 道子、小林 昭世、斉藤 綾、櫻井 修、白石 学、須長 正治、堤 江美子、鶴野 幸子、鶴野 玲治、寺山 裕策、土門 良裕、 樋澤 明、渕上 季代絵、細谷 健、松隈 浩之、面出 和子、守屋 一夫、若林 尚樹、脇山 真治、渡辺 誠、“ディジタルイメージクリエーション－デザイン編 CG－” 財団法人画像情報教育振興協会（CG－ARTS 協会)、2001.

［5］舩橋誠壽、吉原郁夫、“システム制御情報ライブラリー〈２０〉 システム制御のための知的情報処理” 朝倉書店、1999.

［6］http://www.javadrive.jp/

［7］http://java.sun.com/