式を用いたアニメーション・システム

3V-1

小野眞・宮田一乗
日本アイ・ビー・エム株式会社
東京基礎研究所

はじめに

コンピュータアニメーションは主として娯楽・教育などの分野で利用されてきたが，最近ではマルチメディアドキュメント，データベースなどコンピュータのユーザインタフェースとして欠かせないものとなり，その用途はますます広がりつつある。またアニメーション作成のアプローチも，よりリアルな表現を，よりインタラクティブな環境をとその研究分野も幅広い。私達は主に高等教育の場を想定し，非プログラマーのためのアニメーション環境の開発を試みた。キーフレームやプレイバックアニメーションでは表現しにくい数学や物理のアプリケーションを実行するために，式を用いてアニメーションを指定する方法をとりいれた。すなわちユーザは式を用いて物の形や属性，動きを指定する。本稿ではシステムの概要および式の効率的な評価の仕方について述べる。

システム概要

本システムは図1 に示すように，

1. アニメーションに用いるオブジェクトを定義する環境
2. 定義されているオブジェクトを用いて実際に動かしたり，パラメータを変更したりする環境

から成る。ユーザは(2)の環境において同じ定義でパラメータの異なるオブジェクトを複数呼び出すことも，タイムスケールやパラメータをインタラクティブに変更して再実行することもできる。

(2)で用いるオブジェクトは(1)の環境にて定義する。図1 で示すようにオブジェクトの形を定義する部分と動きを定義する部分，そして両者の関係を指示する部分とからなる。ユーザは式によって形状，色などの属性，そして動きを定義する。したがってそれらは全てユーザによって指定される（できる）ものだが，本稿では動きの指定を中心に述べる。ここで扱うことのできる式は陽関数，すなわち

```
y = f(x)
```

の形で表現できるものに限る。ユーザは任意の式を必要なだけ記述してオブジェクトの動きを指定できる。最終的に形状の移動・変化に使用したい変数と，形状を定義している変数（図1 においては円の形状を定義するx, y, r ）を順にクリックすることで両者を結合する。その結果，動きを示す変数の値が変化するにつれて，自動的に形状が移動・変化しアニメーションを形成する。このとき式は順不同で与えられ（すなわちプログラムと違い評価の順が与えられていない），またアニメーションに使われる変数も必ずしもすべて変化するとは限らない（たとえば惑星の公転を示す場合，位置は変化してもその半径は変化しない）ので，効率よく式を評価することが必要となる。このようなケースにおいては，遅延評価を用いて後ろむきに式を評価するのが一般的である。私達は与えられた式が陽関数であることを用いて，さらに効率的な遅延評価を行なったので，つぎに述べる。

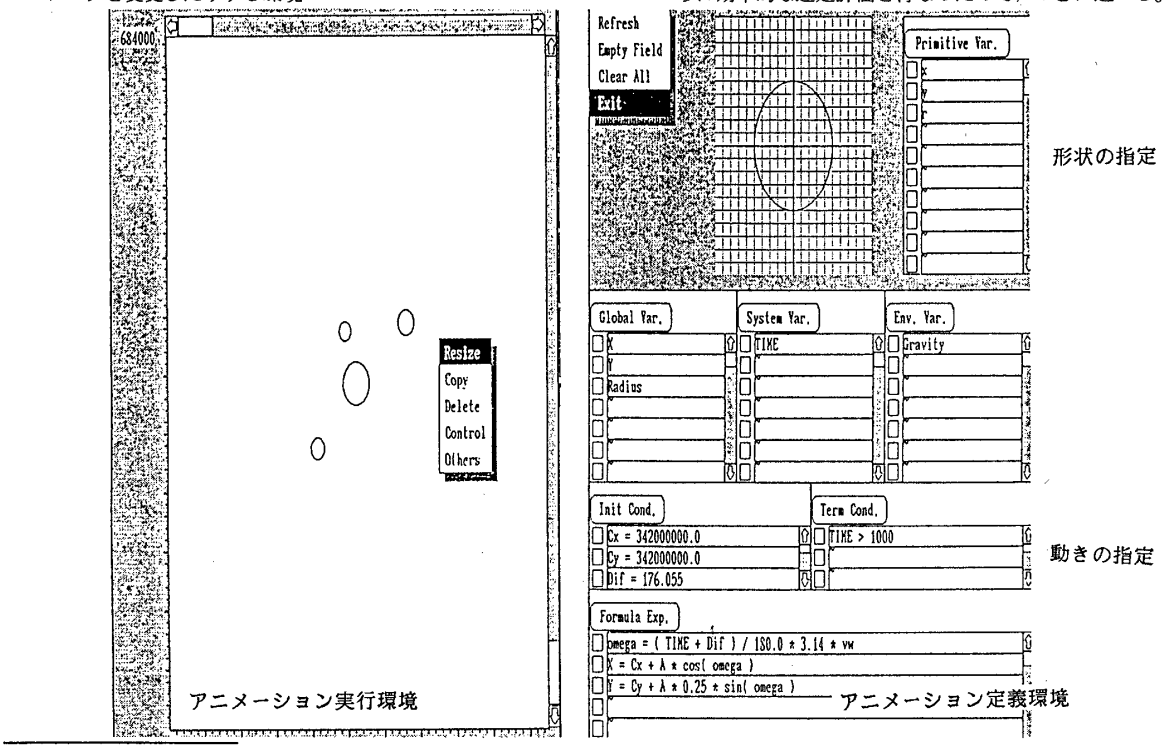


図 1: アニメーション・システム全体図

A formula based animation system
Makoto Ono, Kazunori Miyata
Tokyo Research Laboratory, IBM Japan, Ltd.

変数間の推移閉包を用いた最適化

まず私達は式の評価にあたり各変数の依存関係を求め依存グラフを作り，かつアニメーションのように同じ式を何度も評価することを考慮して，ある変数を書きかえたときにどの変数を再評価しなければならないかを示すグラフを求める事にした。たとえば図2 の場合，与えられた式(a) から各変数の依存関係を求めたグラフが(b) であり，そのグラフの推移閉包を求めることにより各変数に到達可能な変数を求める(c)。

実際の式の評価のアルゴリズムはつぎのとうりである。

1. 与えられた式から変数の依存関係の有向グラフを求めると同時に，どの式を用いればどの値を求められるか予め調べておく。
2. そのグラフの推移閉包(Transitive Closure)を求める。
3. 形状にパラメータに結合された値から後ろむきに値の評価をはじめる。
4. もし必要とする値が未決定(undetermined)であれば，その値を導く式を評価する。
5. 評価の途中において別の未決定の値を必要にする場合，再帰的に繰り返す。
6. 値が求まったならば，先の推移閉包グラフより，その変数より到達可能な変数を未決定にしてから，その値を代入する。

このような評価を行なうことにより，無駄な評価を行なわずに効率的にアニメーションをすすめることができる。

インプリメンテーション

上記システムを実現するにあたり私達はインタプリタを製作し，そのインタプリタのメモリアクセス，すなわちload/store命令実行時に，内部で自動的に最適化する方法を用いた。図2 (d)のようにload命令を実行する時に値が未決定ならば式の評価を自動的に行ない，またstore命令のときにはその変数から到達可能な変数を未決定に書き直す。

したがってオブジェクトの定義を行なう環境は，ユーザの与えた式・および結合状態より，自動的にそのインタプリタのプログラムをつくるプログラム・ジェネレータの形で開発された。

(a) 与えられた式

$x = a \cdot t + b$

$y = \alpha \cdot t^2 + \beta \cdot t + \gamma$

$a = g \cdot \cos\theta$

$\alpha = g \cdot \sin\theta$

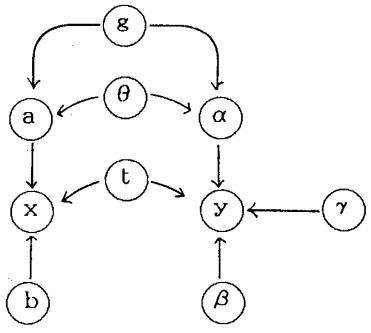
$b = 40$

$\beta = 40 \cdot b$

$\gamma = \alpha + \beta$

$g = 9.8$

(b) 式から作られた変数依存グラフ



(c) 各変数への到達可能変数

$x \leftarrow a, b, t, g, \theta$

$y \leftarrow \alpha, \beta, \gamma, t, g, \theta, b$

$a \leftarrow g, \theta$

$\alpha \leftarrow g, \theta$

$\beta \leftarrow b$

$\gamma \leftarrow \alpha, \beta, g, \theta$

(d) インタプリタによる実行

```
load x
    load a
        load g
        load θ
    load b
    load t
load y
    load α
    load β
    load γ
```

図 2: グラフを用いた式の評価とその実行

おわりに

高校や大学の物理・数学等においては，式を用いたアニメーション・システムはかなり有効であると考えられる。今後の課題としては，

1. グラフィックスエディタのなかで，作ったものを式で制御できるための情報を付加できるようにする
2. 陽関数をはじめ，もう少し一般的な式も扱えるようにする
3. ユーザインタフェースの整備

を行なっていきたい。

現在本システムはIBM6100ワークステーション上に，ハイパーメディアシステムの一部として開発されている。

文献

1. Finzer, W. & Gould, L.: *Programming by rehearsal.* BYTE **Vol.9, No.6** (1984) pp. 187-210.
2. Smith, R. B.: *Experience with the Alternative Reality Kit: An Example of the Tension Between Literalism and Magic.* IEEE CG & A **Vol.7, No.9** (1987) pp. 42-60.
3. Whilhelms, J.: *Toward Automatic Motion Control.* IEEE CG & A **Vol.7, No.4** (1987) pp. 11-22.