グラフィクスとCAD 96－7
（1999．8．23）

力学的手法によるアニメーションシステムの構築

安部麻里, 大野義夫 *
慶應義塾大学大学院理工学研究科計算機科学専攻

概要

本研究では力学的手法を用いたアニメーションシステムの提案をする．アニメーションの対象は，単一なキャラタの表現ではなく，動物の群れを代表とする多数のキャラクタの表現を対象としている．群集アニメーションにおいて，キャラクタ毎に動作記述を与えシミュレーションを行った場合，大変なコストがかかり，群集状態での行動を予測することが困難である．本研究ではパーティクルシステムをベースに，物体の形状モデリングと群集アニメーションにおけるキャラクタ，Actor の行動を力学的手法による統一されたアルゴリズムで記述，群集アニメーションを効率良く作成する手法を述べる．操作性に優れ，キャラクタの行動条件を逐一記述することなく直観的に分かりやすいシステムを設計者，アニメータ両者に提供する事が出来る．

Dynamics-based Animation System

Mari Abe, Yoshio Ohno
Department of Computer Science, Faculty of Science and Technology,
Graduate School of Keio University.

Abstract

The aggregate or crowded motion is of a great interest to a large number of animators. Simulation of such complex motion, however, is difficult if we specify code of conduct to each Actor. Moreover, when such approach is adopted, the behavior of each Actor, i.e., what action does it take next, where does it go next, and so on, is almost impossible to cast, not only for system designers, but also for animators.

In this paper we discuss some techniques for handling such motions intuitively based on dynamic method. This techniques, which originally come from the particle systems and ASAS, take account of shape modeling of obstacles and modeling of Actor's behaviors at the same time. These methods will be of great help for system designers and animators making crowd animation.

1 はじめに

アニメーション生成の代表的なアプローチを5つの観点から3つに分類した(表1).

パーティクルシステムを使った気体や流体の表現は，高度な専門知識を必要とする．炎，煙，霧，雲等を表現するために，温度，気圧，風，輝度等から物理学的なアプローチによってパーティクルの振舞いを計算し，シミュレーションする [3, 1]．正確なシミュレーションに近いが，直観的に分かりにくい．パラメータの調整は多くの経験と労力を必要する．

手法	パーティクル	群	行動ベース
数	多数	中間	少数
知能	無い	ある程度	ある
物理	考慮	ある程度	考慮せず
衝突	検知可能	回避	知的回避
制御	力の場	低レベルなルール	高レベルなルール

表 1: アニメーションシステムの比較

Reynolds [5] は，パーティクルシステムをベースに鳥の群を表現した．簡単な行動規則を個々の Actor(boid) に与え，現実に存在する群の現象に近いシミュレーションを行なった．　これは

*{mari|ohno}@on.cs.keio.ac.jp

ASAS(Actor/Scriptor Animation System) と呼ばれ, アニメーションシーケンスの記述をスクリプト, 動作主を Actor, つまり一フレーム毎に実行されるコードのかたまりとして, 独立して動く, 他の Actor と通信できる, 協調するという特徴を定義している [4]. 大域的な記述をせず個々のルールから全体を表現することを可能にした一方で, 障害物回避に対する Actor の振舞いに統一されたアルゴリズムがない [6]. 壁のような平面では, 新しい進行方向を現在の進行方向と平面との交差から求め, 回避行動を行なうが, 凸型物体では Actor の視点から見た 2 次元画像を処理する等, 外部に対する行動規則の記述方法は様々である. スクリプトの複雑性は物体の幾何形状を定義する際の, 定義の多様性に依存する.

一方内部の記述に対して複数のスクリプトを同時に使用する場合, システム設計者はスクリプトがお互いに依存せず, 独立した機能を記述しなければならない. 物体を避けながら右に曲がらなければならない場合, 『避ける』と『右に曲がる』というスクリプトは直交していないので Actor はしばしば進行方向を見失う. スクリプト同士の直交性を保ちつつ, それらを合成する場合, 行動が複雑になり Actor 数が増えるほど処理が複雑になる.

高度な行動規則を内部表現として持ち, 群を表現したアニメーション手法に安生らの研究がある [2]. これは行動モデルや動作生成アルゴリズムを部品化し, 複数組み合わせることによって, 群でありながら高度な動作を生成する手法である.

本研究では, 従来群集アニメーションにおいて, 物体モデリングと Actor のふるまいを分離されたシステムとして扱うことで生じていたコストを軽減する. パーティクルにポテンシャルを与え, 物体と Actor を同一のパーティクルとみなし, アニメーション空間に力の場を与える事によって, 群集アニメーションシステムを構築する. 内部表現と外部表現の区別を出来るだけ無くし, 統一されたアルゴリズムで表現する. 表 1の分類で, パーティクルシステムと群の中間に位置づけする.

2 システム設計

2.1 システムの概要

表現する対象は Actor と物体であるが, 物体に幾何情報を与えない. その代わり物体をパーティクルの集まりとし, Actor はそのポテンシャルを認識することで行動を決定する. Actor 自身も同様にポテンシャルをもとにお互い通信する. Actor は速度に比例した視距離, 反比例した視野角をもっており, この視野によって物体, つまり障害物を認識する (図 1).

2.2 データ構造

各 Actor, 各障害物は表 2(a), (b) に示すデータを保持する。

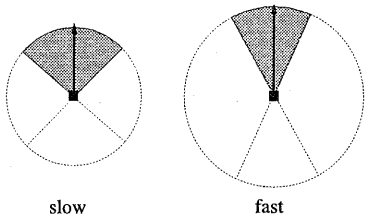

図 1: Actor の視野

現在位置	内部状態
速度ベクトル	速度に比例した視野角
最低速度	最小視野角
最高速度	最大視野角
ポテンシャル	

(a) Actor

現在位置	ポテンシャル

(b) 障害物

表 2: 保持するデータ

2.3 力学的手法

1. ポテンシャルから受ける力
速度をもった Actor がポテンシャルの場に飛び込む状態を設定する. まず式 (1) で障害物が Actor に及ぼすポテンシャル V を求める.

$$V = Aexp(-\lambda d) \tag{1}$$

ただし $d = \sqrt{x^2 + y^2}$ とする.

次に V をもとに Actor が受ける力 F を式 (2) で求め, F に従って 1timestep 毎に速度ベクトルを更新させる. 速度方向はポテンシャルの等高線の向きに設定する.

$$\begin{aligned} F &= -gradV \\ F_x &= \frac{\partial V}{\partial x} \\ F_y &= \frac{\partial V}{\partial y} \end{aligned} \tag{2}$$

ここで

V : 障害物が Actor に及ぼすポテンシャル
d : 障害物と Actor の距離
x, y: Actor の 2 次元座標
A : const

λ : const

とする.

視野に含まれる他の Actor や障害物の影響は大きく反映され, 遠方にある障害物の影響力は小さく反映されるように回避行動がモデル化される. (図 2).

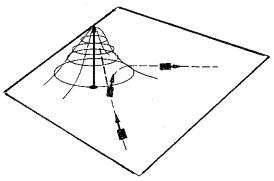

図 2: 単一の障害物回避

次に, ある程度大きさを持った物体を連続した障害物でモデル化する. 大きさを持った物体の表面は, 1 点で表される障害物の集合として表現する (図 3).

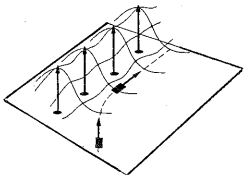

図 3: 大きさを持った障害物

Actor から認知される複数の障害物から, 作用する力をもとめ, 1timestep 毎に速度ベクトルを更新する (式 (3)).

$$F = \sum_{i=1}^{N} -gradV_i \qquad (3)$$

ここで

F : Actor の受ける力
N : 認知した障害物の数
V_i : 障害物 i が Actor に及ぼすポテンシャル

とする.

2. 速度ベクトルの算出

式 (3) より力のベクトルを求めた後, 一定時間内に力の方向へ新しい速度ベクトルを変更するよう, 1timestep 毎の速度変化を求める. 速度方向はポテンシャルの等高線沿いに設定する. 現段階では速度方向変更にかかる時間をスクリプトの状態からパラメータによって調整する. 障害物との衝突を予測した状態では 16step で方向転換し, 他の Actor と正面衝突の危険性を予測した場合では, 素早く方向転換しなければならないため, 8step で方向転換する. 速度ベクトルの絶対値における変化は, スクリプトの状態によって一定の加速度を設け減速, 加速を行う. スクリプトについては第 2.4節で詳しく説明する.

2.4 スクリプトの組み込み

ポテンシャルによる行動モデルを補完するため, いくつかのスクリプトを提案する. スクリプトは Actor によって選択される. それぞれの状態が重複する場合には優先順位によって 1 状態を選択する (表 3). Actor は最高速度, 最低速度をもっているため無限に加速したり停止したりすることはない.

優先順位	状態	速度変化
1	Actor 認知 (正面衝突)	減速
2	Actor 認知	減速
3	障害物認知 (前方)	減速
4	障害物認知 (両脇)	減速
5	安全	加速

表 3: スクリプトの状態

1. Actor 認知 (正面衝突)
   ある Actor の視野に他の Actor が真正面から向かって来る時, お互いが同じ方向に速度ベクトルを変更しないように, 左に方向転換する.

2. Actor 認知
   ある Actor の視野に他の Actor が存在する時, 他の Actor に対する自分の相対速度を求め, 新たにその Actor が自分にとって危険な状態であるかどうか判定する. 例えば視野に入っているが並行に移動している Actor は影響を考慮する必要はない.
   危険であると判断された時, 他の Actor で作成されるポテンシャルの場に従って速度ベクトルを算出する.
   この処理によって同じ速度ベクトルをもつ Actor は無視し, 前をすすむ Actor が遅い時は速度を下げるような表現ができる. 相対速度で他の Actor を評価することによって, Actor を障害物と等価に扱うことができる.
   式 (4) で相対速度を計算する。

$$v_r = -v_a + v \qquad (4)$$

ここで

v_r: 他の車を考慮した車の相対速度
v_a: 他の車 a の絶対速度
v : 車の絶対速度

とする.

相対速度を考慮する前の様子と後の様子を図4, 図5に示す。

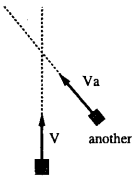

図 4: 相対速度を考慮する前

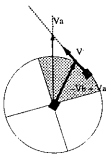

図 5: 相対速度を考慮した後

3. 障害物認知 (前方)
第 2節で述べた方法で物体を回避しながら, 減速する.

4. 障害物認知 (両脇)
両側面に同時に障害物が存在するとき, Actor はポテンシャルによる速度方向の変更をしない. 狭路を進む様な状況では不安定な動きをしない.

5. 安全
最高速度に達するまで一定の加速度で加速する.

3 結果

C 言語, X ライブラリを使用した (図 6).

- 他の車と正面衝突の危険性があるとき
進行方向に向かって左側に回避することができた (図 7).

- 他の車に追突の危険性があるとき
他の Actor が視野に入ったとき, 相対速度を計算して回避することができた (図 8). 前方が遅い車, 後方速い車である.

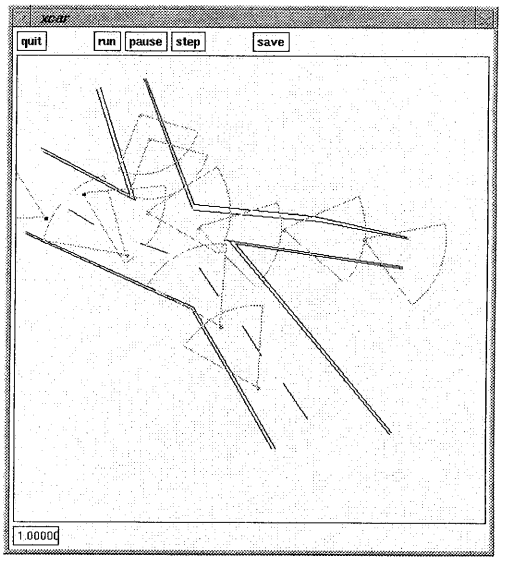

図 6: 実装画面

- 障害物を認知したとき
大きさをもつ障害物を回避することが確認できた (図 9).

- 両脇に障害物を認知したとき
狭路を通り抜けることが確認できた (図 10).

4 まとめ

本論文ではパーティクルシステムをもとに, ポテンシャルの概念を用いた力学的なアプローチによる群集アニメーションシステムを提案した.
障害物と Actor を, ポテンシャルをもつパーティクルとしてモデリングし, 物体の幾何形状によらない行動モデルを記述する事が出来た.
効果はそれぞれ以下の通りである.

- アニメータ:
  - 物理方程式のような専門知識がなくても一つのアルゴリズムを理解するだけでよい
  - スクリプトベースではなく力学的な要素を加えることによって直観的に分かりやすくなった
  - ポテンシャルを独立したプリミティブのように扱って, ユーザ側で要素を設定することができた

- システム設計者:
  - Actor の行動・物体の表現が単一のアルゴリズムによって簡単に表現できた

今後の課題として, 以下の事があげられる.

- ポテンシャルの変形
  ポテンシャルのバリエーションを増やすことによってActorの行動パターンを表現出来るように, 様々なポテンシャルを設計する必要がある.
- 3次元拡張
  現段階では2次元上の動きしか想定していないので3次元拡張が必要である.
- スクリプトの組み込み
  インタプリタを用いて, プログラム実行中にスクリプトモジュールを組み変え可能にすると柔軟性に富んだシステムになる.

謝辞

本研究の一部は日本学術振興会平成9年度未来開拓学術研究推進事業「システム生命を有する知的システムの構築」(JSPS-RFTF97I00103)の援助を受けている.

参考文献

[1] James D. Foley, Andries van Dam, Steven K. Feiner, and John F. Hughes. *Computer Graphics - Principles and Practice:2nd edition.* Addison-Wesley, 1990.

[2] 三好雅則, 渡辺範人, 安生健一. CGによる群集表現技術と映像制作への応用. Visual Computing グラフィックスとCAD 合同シンポジウム '98, pp. 7–12, 1998.

[3] W Reeves. Particle systems - a technique for modeling a class of fuzzy objects. *SIGGRAPH '83*, pp. 359–376, 1983.

[4] Craig W. Reynolds. Computer animation with scripts and actors. *acm SIGGRAPH '82 Proceedings*, pp. 289–296, 1982.

[5] Craig W. Reynolds. Flocks, herds, and schools: A distributed behavioral model. *acm SIGGRAPH Proceedings*, pp. 25–34, 1987.

[6] Craig W. Reynolds. Not bumping into things. *In Notes for the SIGGRAPH 1988 Course Developments in Physically-Based Modeling*, 1988.

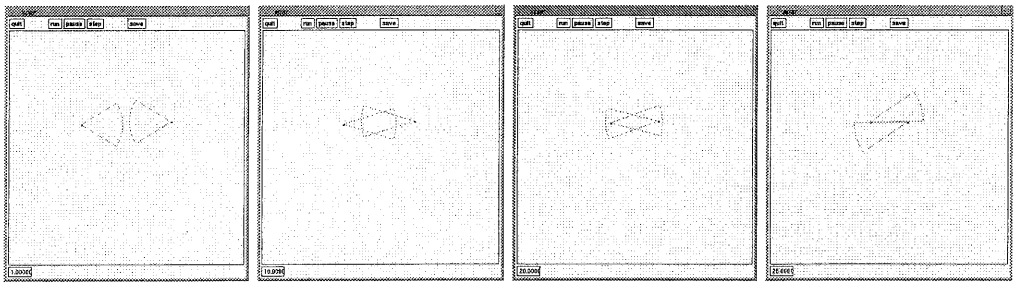

図 7: Actor 認知 (正面)

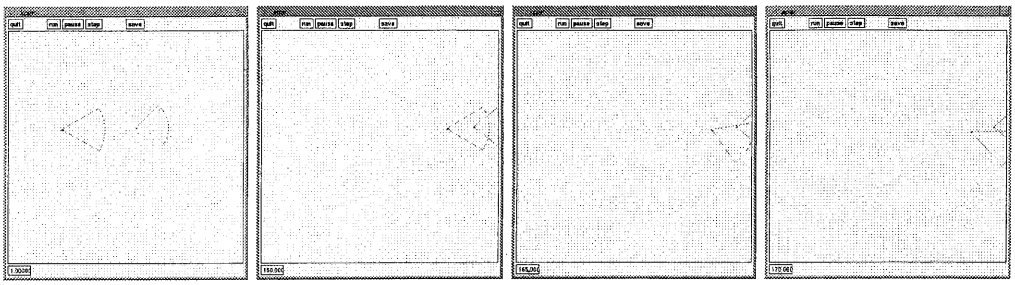

図 8: 追突

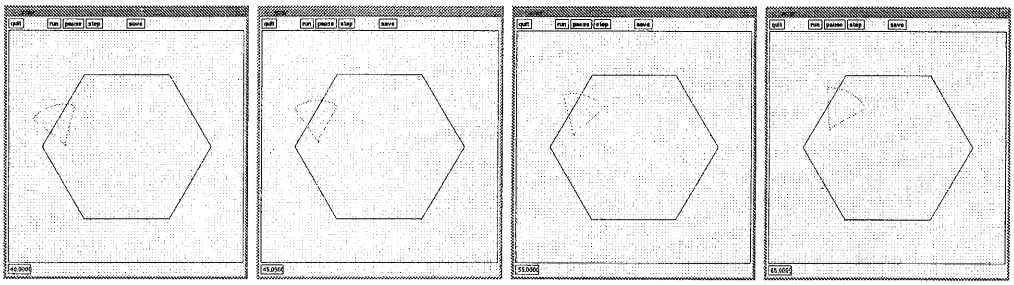

図 9: 障害物認知 (前方)

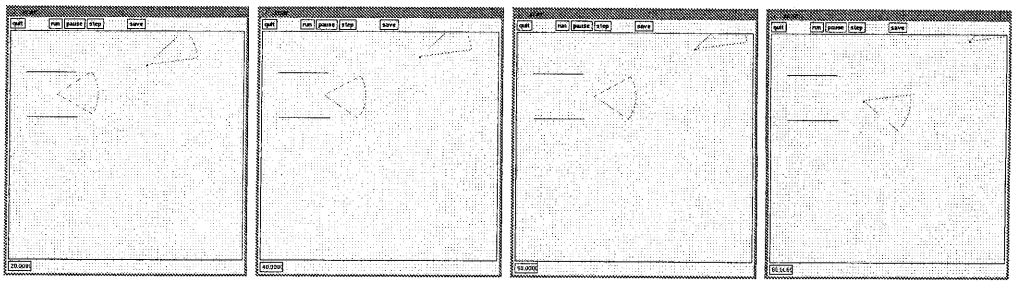

図 10: 狭路