

2P-3 アニメーションにおける経路自動生成法

泰間 健司

清水 雅久

三洋電機(株) 情報通信システム研究所

1. はじめに

最近のハードウェアの急速な進歩により、コンピュータアニメーションも多くの分野で利用されるようになってきた。しかし、コンピュータアニメーションを制作する際には、移動物体の行動制御はほとんどアニメータの手作業によることが多いため、多数の物体が入り組んだ複雑な環境では、アニメータの負担が大きくなり、行動制御が困難になってくることが多い。そのため、行動制御の自動化の研究が盛んになっている^{1) 2) 3) 4) 5)}。本稿では、他の物体との相互干渉(相対速度・位置関係など)を考慮することにより、自動的に移動物体を加速・減速したり、あるいは進行方向を変えることにより、障害物との衝突を自動的に回避し、経路を決定するアルゴリズムの提案を行う。

2. 従来の自動衝突回避手法

従来、自動衝突回避問題については、ロボットの分野などでポテンシャル法がよく利用されているが、この方法には、引力と反発力が釣り合う点で移動物体が停留してしまう、多数の物体が複雑に配置されている空間ではポテンシャル関数の計算が困難である、などの様々な問題が発生する。

又、移動物体を点と考え、移動物体の大きさの分だけ、障害物を膨張させた新たな環境下で、点物体が移動する経路を考える手法が提案されているが⁶⁾、障害物を膨張させる計算が困難であるため、静的な環境下での経路生成への適用例が中心である。

一方 Reynolds は、単純な環境において、移動する物体から視野内に障害物が見えればその進行方向を転換し衝突を避けるようなアルゴリズムを提案している⁷⁾。

3. 自動衝突回避アルゴリズム

本アルゴリズムは、物体の形状情報とその物体の移動速度(大きさ及び方向)など動的な変化を伴う行動情報とを基に、障害物を検知し衝突回避を行うものである。

3-1. アルゴリズムの特長

本アルゴリズムの特長は、移動物体と障害物の距離を考慮しており、たとえ進行方向上に障害物があろうとも、十分遠い距離にある障害物は無視し、無駄な計算時間を省いている。又、移動物体は、進行方向の一点上のみを注視しているのではなく、視界内にある障害物の進路及び速度などの局所情報を活用し、どの方向に進路を取るべきか総合的に判断する。更に、移動物体や障害物の形・大きさが動的に変化しても、適切に移動物体の経路を見い出す事ができる。

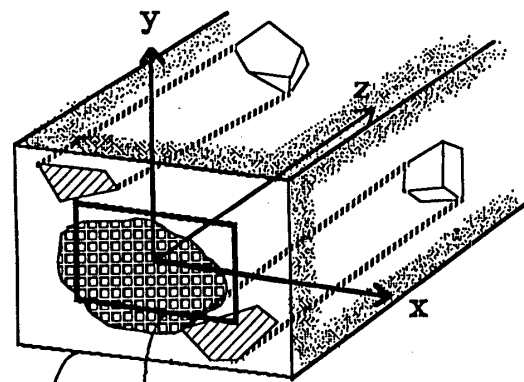
3-2. アルゴリズムの概要

正のZ軸方向に進む移動物体とある方向に運動する障害物とが存在する3次元空間において、以下の手法で物体の進行方向を決定する。

- 1) その位置のZ座標が正であるような(移動物体より前方にある)障害物に対して衝突危険性を求める。
- 2) 衝突危険性がゼロになる障害物は、除去する。
- 3) 視野内に障害物が検出されなければ、大局情報(到達すべきゴールの位置)に基づき進行方向を決定する。
- 4) 視野内に障害物が存在すれば、局所情報(移動物体と各障害物の速度・位置、その距離)に基づき衝突回避制御を行い、次に大局情報を加味して進行方向を決定する。

3-3. 視野、環境視野

移動物体の現在位置を原点とし、進行方向が正のZ軸であるような3次元空間座標系で、移動物体をXY平面上に平行投影する。XY平面上の移動物体の写像を含む方形領域を視野と呼び、更にXY平面上でこの視野を包括する方形領域を環境視野と呼ぶ(図1)。



環境視野 視野

図1: 視野、環境視野

本アルゴリズムでは、視野と環境視野を利用して、移動物体と障害物との衝突判定及び衝突回避方向の決定を行う。例えば、XY平面上に写像された障害物領域が、移動物体領域と交差するならば、この障害物は移動物体と近距離にあり、このまま直進すれば近い将来衝突する可能性が高いということになる。又、これら両領域が交差しなくても、障害物の運動を考えるならば、領域の配置関係から将来移動物体が各障害物と衝突するかどうかの可能性を求めることができる。

Self Routing Animation

Kenji TAIMA Masahisa SHIMIZU
SANYO ELECTRIC Co., Ltd.

3-4. 衝突危険性

障害物 i の座標を $T_i(X, Y, Z)$ 、運動速度を V_o (v_x, v_y, v_z)、移動物体の速度を $V(0, 0, V)$ 、障害物と移動物体との距離を D 、並びに距離の定数を K_D とした場合、各障害物 i に対する衝突危険性 $P(i)$ を以下のように定義する。

$$P(i) = F(D) \cdot G(V, v_z) \cdot H(X, Y, V_o)$$

但し、

$$1) \quad F(D) = \begin{cases} 0 & : D \geq K_D \\ 1/D^2 & : D < K_D \end{cases}$$

$$2) \quad G(V, v_z) = \begin{cases} \text{正の } Z \text{ 軸方向に関して、} & : (V - v_z)^2 \\ \text{移動物体のほうが速い} & \\ \text{遅い、又は同じ} & : 0 \end{cases}$$

$$3) \quad H(X, Y, V_o) = \begin{cases} Z \text{ 軸から遠ざかる} & : \frac{1}{\sqrt{X^2 + Y^2}} \cdot \frac{1}{|X|v_x + |Y|v_y} \\ Z \text{ 軸に近づく} & : \frac{|X|v_x + |Y|v_y}{\sqrt{X^2 + Y^2}} \\ \text{変わらない} & : \frac{1}{\sqrt{X^2 + Y^2}} \end{cases}$$

3-5. 局所情報による衝突回避方向候補の選定
次のような手法で候補の決定を行う (図2)。

- 1) 各障害物領域の頂点と原点とを結ぶ直線がその障害物領域と接するような頂点を検出する。
- 2) 検出された上記各頂点と原点とを結ぶベクトルの単位ベクトルを求める。同図の場合、 OP' 、 OQ' 、 OR' 、 OS' となる。
- 3) 隣り合うベクトルの内、同じ障害物領域に属さないベクトルをペアにする。この場合 (OP' 、 OS')、(OQ' 、 OR') となる。
- 4) 求める回避方向の候補は、これらペアベクトルの和で表される。同図の場合、

$$OA1 = aOP' + bOS'$$

$$OA2 = cOR' + dOQ'$$

- 5) 上記の候補を表す式の変数 a 、 b 、 c 、 d を衝突危険性 $P(A)$ 、 $P(B)$ により求める。即ち、

$$OA1 = OP' / P(A) + OS' / P(B)$$

$$OA2 = OR' / P(A) + OQ' / P(B)$$

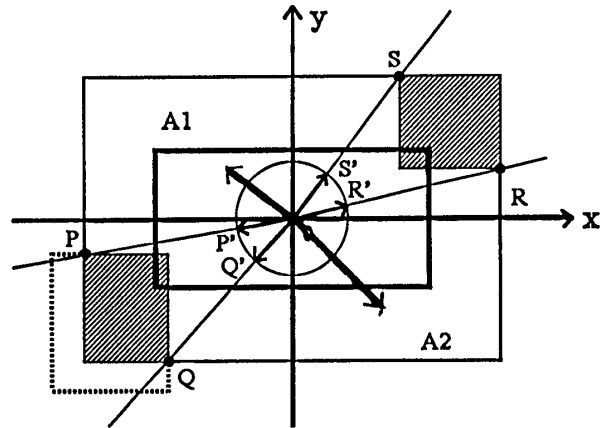


図2：衝突回避方向の選定

3-7. 大局情報に基づく経路生成

上述のようにして得られた衝突回避方向ベクトルの複数の候補の中からゴール G に向かうベクトル OG に最も近いベクトルを選択する。但し、ゴールがある一定距離内にあり、しかも環境視野内に存在する場合は G を XY 平面に平行投影した点 G' に物体を移動し、進行方向ベクトルをゴール (正の Z 軸方向) に向ける。もしここで、視野内に障害物がなければゴールに向かい直進し、あれば前述の局所情報による衝突回避を繰り返す。

4. 終わりに

大局的情報も考慮しつつ、物体の形状情報と、移動速度 (大きさ及び方向)、位置など動的な変化を伴う行動情報を基に自律的に障害物を回避しながらゴールを目指すことができるようなアルゴリズムの提案を行った。今後、このアルゴリズムの有効性を評価し、より一般性の高いアルゴリズムに改良していく予定である。

参考文献

- (1) Wilhelms, J., "Toward Automatic Motion Control", IEEE Computer Graphics and Applications, Vol. 7, #4, April 1987, pp. 11-22
- (2) 久保田ほか, 「ローカルプランニングによる移動ロボットの経路探索」日本ロボット学会誌 7 巻 4 号, pp. 3-10, (1989年 8 月)
- (3) K. Fujimura et al. "Path Planning Among Moving Obstacles Using Spatial Indexing" IEEE International Conference On Robotics and Automation Vol. 2/3, pp. 1662-1667
- (4) K. Kant et al. "Toward Efficient Trajectory Planning", International Journal of Robotics Research, pp. 72-89
- (5) 泰間ほか, 「アニメーションにおける衝突回避アルゴリズム」平成元年電気関係学会関西支部連合大会論文集 pp. G243 (1989)
- (6) T. Lozano-Perez et al. "An Algorithm for Planning Collision-Free Paths among Polyhedral Obstacles" Communications of the ACM Vol. 22 No. 10, (Oct. 1989)
- (7) Reynolds, C. W., "Flocks, Herds, and Schools: A Distributed Behavioral Model" Computer Graphics Vol. 21, #4, July 1987, (ACM SIGGRAPH '87 Proceedings), pp. 25-34