

CGによる群集表現のための階層型キャラクター管理手法

三好雅則* 渡辺範人* 安生健一**

miyoshi@hrl.hitachi.co.jp

* (株)日立製作所日立研究所

** (株)日立製作所ビジュアルウェア事業推進部

〒319-1292 日立市大みか町 7-1-1

本稿では、多数の自律行動するCGキャラクターで構成される群集を効率的に管理する手法を提案する。木構造を用いてキャラクターを階層管理することで、キャラクターの行動決定に必要な情報を簡単に取捨選択できるようにする。これにより、キャラクターの行動を計算する時間を大幅に短縮できる、さらに映像表現能力を高めることも可能である。実験システムを通して、取り扱うキャラクターの数と演算時間を線形の関係にできること、および群れの疎密を調整できるなど表現力を高めることができることを確認した。

Hierarchical Character Management for Computer-Generated Crowd

Masanori Miyoshi*, Norito Watanabe* and Ken-ichi Anjyo**

miyoshi@hrl.hitachi.co.jp

*Hitachi Research Laboratory, Hitachi, Ltd.

** Visualware Planning Department, Hitachi, Ltd.

7-1-1, Omika-cho, Hitachi-shi, Ibaraki-ken, 319-1292 Japan

This paper proposes a new method for managing computer-generated crowds of autonomous CG characters. We employ the tree structure to manage the crowds, which makes it possible to select the required sets of information efficiently. We can reduce the amount of time to generate locomotion of the characters and get capability to represent more sophisticated crowds' movement.

1.はじめに

コンピュータグラフィックス(以下 CG)を用いて、多数のキャラクターが登場するアニメーション映像を制作するために、行動シミュレーションと呼ばれる手法が用いられている[1][2][3][4]。これは、あらかじめ個々のキャラクターが周囲の環境に応じて自律的に行動するように行動パターンを記述し、そのキャラクターの行動を計算機によって自動的に求めるものである。このような手法が、サイバースペース(ネットワーク上の仮想空間)や、映画など商業用の映像制作にも利用されつつある[5]。

これらの手法では、周りのキャラクターを観測して得た情報を用いて、キャラクターの行動を決定する。従って、キャラクター総数が増えると、観測すべきキャラクターの数が急激に増加し、シミュレーション時間が増大してしまう。観測すべきキャラクター数は、キャラクターの総数に対して二乗のオーダーで増加する。このことは、結果としてキャラクター数の多い映像の製作コストを上昇させる。また、各々のキャラクターに設けた知覚範囲内のキャラクターのみを観測の対象とすることで、処理対象のキャラクター数を減らす工夫もなされている。しかし、この方法では、知覚範囲外に存在する影響力の強いキャラクターの影響を考慮できない、知覚範囲内に大量のキャラクターが存在する場合には効果がないといった問題点が残る。

そこで、人間や動物などの群れには、リーダーと部下の関係が多く見られることに着目して、木構造を用いて群れを階層管理することにした。リーダーは部下を含む自分のグループの行動を決定し、部下に伝える。部下は、自分のグループ外のキャラクターを直接観測せずに、リーダーの指示を通してグループ外の情報を入手する。このため、上述した観測対象キャラクターの上昇を低減させることが可能になる。

2.階層型キャラクター管理

動物や人間の群れを観察する場合、何らかのグループ構造が見られる場合が多い。そして、そのグループにはリーダーと、それに従う部下の関係も見られる。すなわち階層関係が生じていると考えられる。例えば、猿の群れにはリーダーが存在し、その配下の猿たちはリーダーの意向や指示に従いながら、自分に与えられた権限の範囲内で自律行動をしている。群れ同士の抗争が生じた場合には、リーダーがどう対処すべきか群れの方針を決め、配下の猿に指示を出す。また、人間の場合には動物以上に、軍隊・会社など階層的な組織構造が見られることが多い。

本章では、このような階層的な組織を計算機上で表現する方法を説明する。

2.1 木構造を用いた階層管理

ここでは、前述したような階層的な組織を表現するために、木構造を用いることにした。キャラクターの群れを、対等に管理するのではなく、図1に示すように木構造を使い一つの親キャラクターに複数の子キャラクターを関連付け階層管理する。木構造を用いることで、特定のキャラクターに対して、親、子、兄弟の関係にあるキャラクターを容易に参照できるようになる。なお、本報告では、この階層構造において特定のキャラクターがどのような位置づけにあるのかを示す指標として順位レベルを導入する。順位レベルは、最上位のキャラクターを0として、その子を1といった具合に、子をたどるたびに1だけ値が増加するように定義する。また順位レベルの最大値を、木構造の深さと呼ぶことにする。図1の木構造の深さは2である。

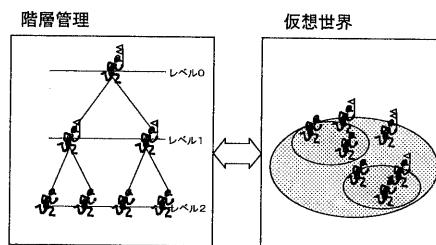


図1 木構造による階層管理

次に、この木構造を使った行動シミュレーションの手順を説明する。前章で述べたように、通常の自律的なキャラクターは、自分の周辺のキャラクターを観測して得た情報を用いて、自分自身の行動を決定する。従って、キャラクター総数が増えると、観測すべきキャラクターの数が急激に増加し、シミュレーション時間が増大してしまうという欠点があった。そこで、木構造を利用して観測すべきキャラクターを限定することにした。ここで、木構造で上位のキャラクターの情報は親から、下位のキャラクターの情報は自分と兄弟から入手できると仮定すれば、群れ全体を考慮したキャラクターの行動が、その親と兄弟だけを観測すれば決定できると考えた。この仮定により、キャラクター数と観測すべきキャラクターの数は線形の関係にできると推測できる。これを実現するために、本報告で取り扱うキャラクターに次の機能を持たせることにした。

- (1) 上位のキャラクターの情報を下位に伝えるために、自分の判断結果を命令という形で子に伝える。
- (2) 下位のキャラクターの情報を上位に伝えるために、自分が統括するグループの状態を把握して、その情報を他のキャラクターに提供する。

2.2 階層管理下での進行方向の決定

次に、階層管理を使った行動シミュレーションの具体例を示す。ここでは、簡単のために、各々のキャラクタは、自分の統括するグループが、他のグループやキャラクタと衝突しないように進行方向だけを定める場合を考えることにした。このままでは、直ちに複雑な振舞いを見せる猿や人間の群集へは適応できない。しかし、キャラクタ同士の衝突回避を表現する比較的単純な行動シミュレーションにおいても、階層管理手法の有効性が示せると考えて、このような限定を設けることにした。

ここでは、進行方向決定のために、魅力度を用いた行動モデルを採用した[4]。これは、キャラクタ同士の好き嫌いの度合いに対応する値をキャラクタに割り振っておき、この値をもとに引力や斥力を発生させる手法であり、キャラクタ同士が群れる行動や、衝突を回避する行動を表現することが可能である。図2に示すように、キャラクタAの進行方向は、キャラクタB, Cとの関係によって生じる好みベクトル v_b と v_c の和 v で決まる。好みベクトルは、キャラクタ間の位置関係によって引力になったり斥力になったりする。例えば、両者が衝突しそうほど近づいたら、衝突を回避するため大きな斥力となる。

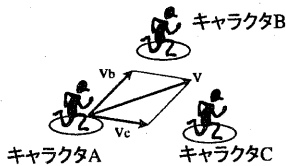


図2 魅力度による進行方向の決定

この考え方を、階層構造を持つ群集に適応する場合の進行方向の決定手順を次に説明する。図3には、キャラクタBの進行方向の決定方法を示している。

(1) 親から子へ命令の伝達

自分の意志を子に伝えるために、後述する手順(3)に従って決定した好みベクトルを自分の子に対して与える。ここで、各キャラクタに個性を与えるために、従順度と呼ぶスカラー量を導入する。従順度は親キャラクタの命令に従う度合いを示すもので、親から与えられる好みベクトルの大きさを変化させる。この値が1の場合は、親に完全に従い、0の場合は、親を完全に無視する。また、子に対して好みベクトルを与えるときに、自分と子との幾何学的な関係を基にしてベクトルの大きさを変化させても良い。例えば、子との距離が遠くなるほどベクトルの大きさを小さくすれば、距離に応じて命令に従う度合いが変化する様子を表

現できる。図では、これらを考慮した結果の好みベクトルを v_o としている。

(2) 親と兄弟の情報の収集

親と兄弟の存在領域を求める。ここでは、存在領域として、キャラクタに対する外接円を考える。親の存在領域は、親単体について求める。また、兄弟の存在領域は、それぞれに子孫が存在しない場合には単体の存在領域を、子孫が存在する場合には、それらをすべて含む存在領域を求める。図の例では、キャラクタAとキャラクタCの存在領域を収集している。親であるキャラクタAの存在領域は、その単体の存在領域 A_p 。キャラクタCの存在領域は、そのすべての子孫を含む存在領域Cとなる。

(3) 進行方向の決定

手順(1)(2)で得た情報を基に、キャラクタBの進行方向を決定する。まず、親と兄弟の存在領域を使って、好みベクトルを求める。ここでは、存在領域同士が干渉する場合には、大きな斥力を生じるようにして、グループ同士の衝突を回避することにする。図の例では、親であるキャラクタAと兄弟のキャラクタCに対して、好みベクトル v_a , v_b を求めている。最後に、これらのベクトルと親から与えられてベクトル v_o を合成して、最終的な好みベクトル v を求める。これがキャラクタBの進行方向になる。

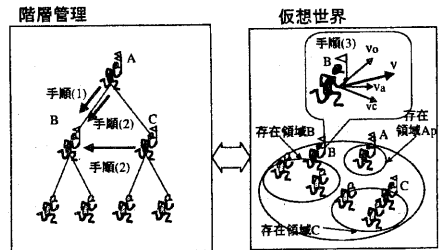


図3 進行方向の決定手順

以上の手順を、木構造の最上位から全てのキャラクタに再帰的に適応することで、すべてのキャラクタの進行方向を決定することができる。

2.3 動的な木構造の変更

前節の手順は、グループ単位での衝突回避を考慮しながら進行方向を決定するものである。映像表現上、グループ単位での衝突回避が望ましい場合もあれば、より小さな単位での衝突回避が必要な場合もある。ここでは、このような場合にも対応できるように、必要に応じて階層構造を動的に変更することを考える。

ここで、親とその子孫からなるグループを、その親の管理グループと呼ぶことにする。図4に示

すように、キャラクタ B の管理グループが、兄弟の関係にあるキャラクタ C の管理グループと衝突しそうになったとき、階層関係を保ったままではグループ同士が干渉しないように回避行動をとる。ここで、キャラクタ B の子を、キャラクタ B の兄弟になるように木構造を変更すれば、キャラクタ B の管理グループは解消され、キャラクタ B とその子は対等の関係になり、キャラクタ単位でグループ C と衝突回避することになる。

また、グループを解消するだけでなく、復帰させることも可能である。例えば、グループ同士が干渉しそうになるためにグループ関係を解消した場合を考える。しばらく時間が経過した後、以前のグループ関係を保ったとしてもグループ間で干渉が生じないならば、以前のグループ関係を再び復帰させれば良い。そのためには、グループ解消時にグループ解消前の構造を保持しておき、復帰時には、その構造に戻るように木構造を変更すれば良い。

このように、動的に木構造を変更することで、衝突判定の粒度を変えることが可能になる。実際には、事前にグループ関係の解消・復帰の条件を決めておき、それに従ってシミュレーション実行時に木構造を変更すれば良い。

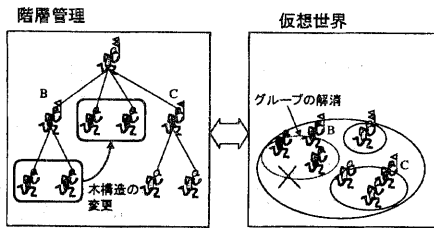


図 4 木構造の変更によるグループの解消

3. 結果

本報告で提案する群集管理機構の効果を確認するために、実験システムをグラフィックスワークステーション(CPUは200MHzのR4000)上に実装した。群集に階層管理を適用することで、キャラクタの行動を計算する時間を短縮できること、および映像表現力が高まることを確認するために、このシステムを使って実験を行った。

3.1 計算時間の比較

実験システムを使って計測したキャラクタ数と計算時間の関係を図 5 に示す。階層管理しない場合と、する場合について調べた。ここで、群集管理については、階層の深さを 3 とし、子を持つ場合には、その数がすべて等しくなるように木構造を構築した。また、計算時間とは、キャラクタの行動をシミュレーションするに要した CPU 時

間であり、描画時間等は含んでいない。ここでは、シミュレーション開始時から 100 フレーム分の平均時間を採用した。

図 5 によると、階層管理しない場合は、キャラクタの数が增加すると、計算時間が急激に上昇することが分かる。一方、階層管理する場合は、キャラクタ数と計算時間は、ほぼ比例の関係にあり、計算時間の上昇を押さえることができている。よって、キャラクタ数が多くなればなるほど、階層管理機構が有効に働き、シミュレーション時間を大幅に短縮できることになる。

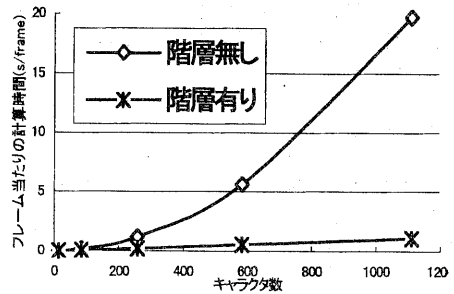


図 5 キャラクタ数と計算時間の関係

3.2 映像の作成

実験システムを使って、障害物を回避する馬の群れの行動を模擬した。映像表現を比較するために、次の 4 つの場合について映像を作成した。

- 階層管理しない場合(図 6)
- 静的な木構造で階層管理をする場合(図 7)
- レベル 1 の親子関係の解消を許す動的な木構造で階層管理をする場合(図 8)
- レベル 1,2 の親子関係の解消を許す動的な木構造で階層管理をする場合(図 9)

ここでは、左上から右方向に代表的な画像を並べてある。キャラクタ数は、馬以外に障害物も含めて 156 である。階層管理する場合は、階層の深さは 3 とし、子を持つ場合には、その数が 5 となるように木構造を構築した。また、馬を囲む円錐はキャラクタの管理するグループの存在領域を示している。画像の右端の円錐は、障害物を表わす。キャラクタは、これと衝突しないように回避行動をとる。

これらの図からは、それぞれの管理手法に応じて、障害物と衝突しないように回避行動をとる馬の群れの動きが表現できている。特に、動的な木構造の場合には、円錐の障害物と円錐の管理領域が接触することで、親子関係を解消するように木構造が変更されて、子の行動が親から解放されていることが分かる。

また、これらの映像を作成するのに要した時間を表 1 に示す。ここでの計算時間は、シミュレーション開始時から 500 フレームのシミュレーションに要した CPU 時間である。

この表から、階層管理をする場合には、しない場合と比べて演算時間を約 1/4 と大幅に短縮できていることが分かる。特に、動的木構造により階層管理する図 9 の場合には、最終的に、障害物との衝突回避は、キャラクタ単位で計算されるため、動き自体も図 6 の階層管理しない場合と比べても遜色なく、さらにキャラクタの疎密の状況も指定でき、有効な手法である。さらに、図 5 を考慮すると、群れの規模が大規模になるほど、本手法の効果が大きくなると考える。

表 1 計算時間の比較

群集管理手法	計算時間(s)
階層無し	230.0
静的階層	62.0
動的階層(第1レベル)	63.9
動的階層(第1,2レベル)	68.8

4. おわりに

木構造を用いて多数の CG キャラクタを階層管理する手法を提案した。本手法の特徴は、キャラクタの行動を計算する時間を大幅に短縮できること、および群れの疎密を調整できるなど映像表現能力を高めることである。本手法の有効性を、実験システムを通して確認した。

今回は、魅力度を用いて決定した進行方向だけ

を親の意向として子に反映させた。進行方向以外にも、自律キャラクタに対して与える命令などを通して、親の意向を子に伝えることができる。今後は、この様な拡張を通して、多彩な群集表現を実現する必要がある。

謝辞

馬の群れの映像作成で利用したモデル、アニメーションデータの作成にあたり、(株)アイデンティファイの松木靖明氏、アニメータの堀大治氏をはじめとするスタッフの方にご協力を頂いたことに感謝致します。

参考文献

- [1] C. W. Reynolds. "Flocks, herds and schools: A distributed behavioral model", Proc. ACM SIGGRAPH, 21(4):25-34, 1987
- [2] 内木哲也、丸一威雄、所真理雄：能動的キャラクタを用いたアニメーション生成手法、第 2 回 NICOGRAPH 論文コンテスト論文集, p.197(昭 61)
- [3] X. Tu and D. Terzopoulos, "Artificial fishes: Physics, locomotion, perception, behavior", Proc. ACM SIGGRAPH, p. 43-50, 1994
- [4] 鷗沼宗利、安生健一、武内良三：群衆行動のモデリングー仮想都市空間における人の群れと環境のインタラクションー, 電気学会論, Vol. 115-C, No.2, '95 (平 7)
- [5] E. Guaglione et al., "The Art of Mulan", ACM SIGGRAPH Course Notes 39, 1998

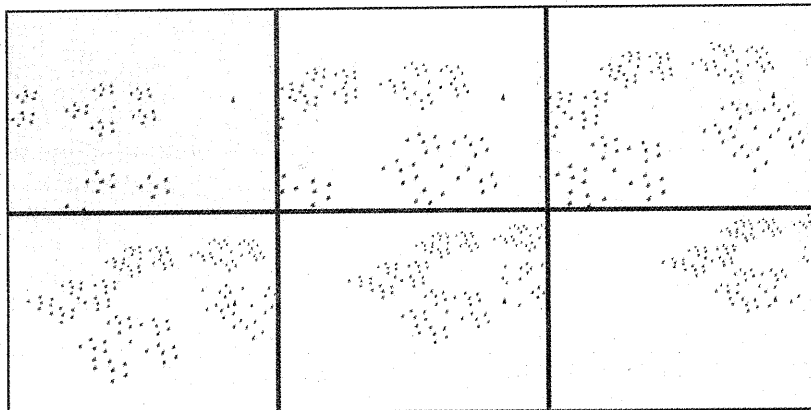


図 6 階層構造無し

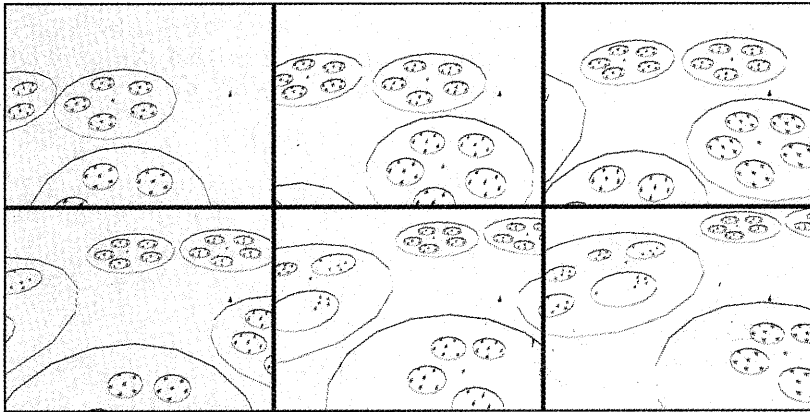


図 7 映像例 (静的木構造)

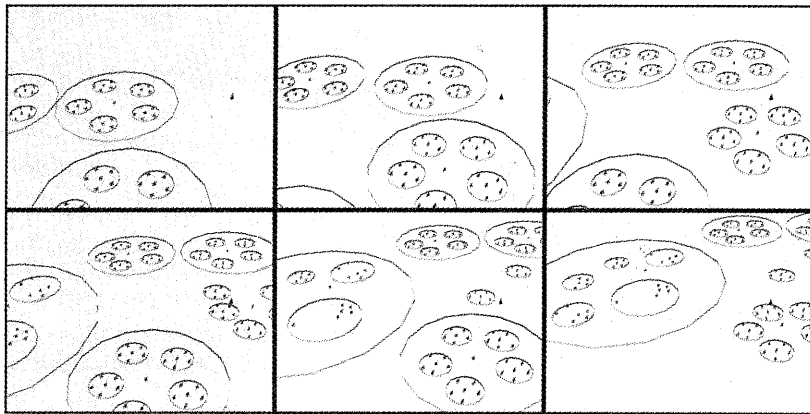


図 8 映像例 (動的木構造、レベル1の親子関係を解消)

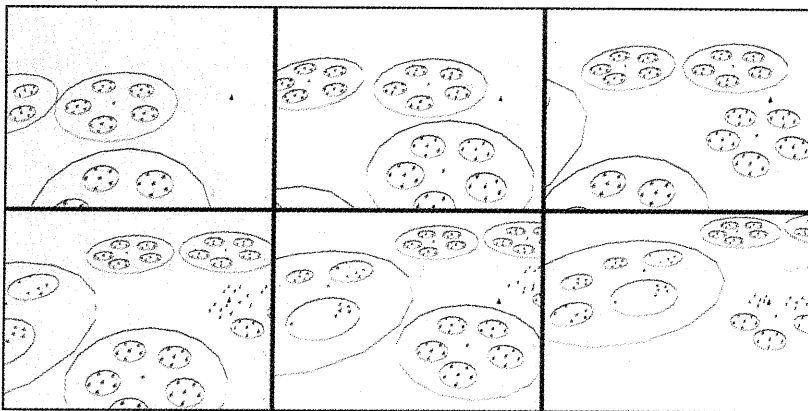


図 9 映像例 (動的木構造、レベル1,2の親子関係を解消)