

ペンインターフェースを用いた群集アニメーションの作成

荻原佑輔[†] 尾下真樹[‡]

[†]九州工業大学大学院 情報工学研究科 情報科学専攻

[‡]九州工業大学 情報工学研究院 システム創成情報工学研究系

〒820-8502 福岡県飯塚市川津 680-4

E-mail: ogiwara@cg.ces.kyutech.ac.jp, oshita@ces.kyutech.ac.jp

近年の映画やテレビゲームでは、数千人単位の大規模な戦闘シーンや街中のシーンにおいて、群集モデルを用いて生成された群集アニメーションが利用されている。群集モデルとは、一定のルールによってエージェントを独立に操作する手法である。しかし、群集モデルを用いてアニメーションを生成するためには、エージェントの移動速度、他のエージェントとの距離といった群集パラメータを手作業で設定する必要があり、利用者の望むようなアニメーションの生成には手間がかかる問題がある。そこで、本研究ではマウスやペンタブレットを用いて、利用者が数本の軌道を入力するだけで、移動経路、移動速度、エージェント同士の標準距離、距離の補正量などの群集パラメータを自動的に決定し、群集アニメーションを生成する手法を開発した。

Pen-based Interface for Crowd Animation

Yusuke OGIWARA and Masaki OSHITA

Department of Systems Innovation and Informatics, Kyushu Institute of Technology

680-4 Kawazu, Iizuka-Shi, Fukuoka, 820-8502, Japan

E-mail: ogiwara@cg.ces.kyutech.ac.jp, oshita@ces.kyutech.ac.jp

Recently, crowd animation based on an agent model has been used for making a scene of large-scale battles and a scene in a town for movies and video games. Agent model is a method to operate each agent independently based on specific rules. In order to use an agent model, however, a user should specify crowd parameters such as path, velocity and distance manually. It is difficult for a user to specify appropriate for generating an animation that the user wants. In this paper, we propose a technique to set crowd parameters such as path, velocity, standard distance among agents and variation of distance automatically based on few trajectories that are given by a user using a mouse or a pen-tablet.

1. はじめに

近年、群集アニメーションは映画やテレビゲームで用いられている。例えば、映画「ロード・オブ・ザ・リング」では、大規模な戦闘シーンでエキストラの代わりにCGで作られたキャラクターを使用している。群集アニメーションは市販のソフトウェアや、ユーザが独自開発したシステムを用いて作成されている。しかし、これらのシステムを用いて群集アニメーションを作成するためには、ユーザは群集の初期位置、移動軌道、移動速度、キャラクター同士の間隔など様々なパラメータを手動で設定する必要がある。これは非直感的で非常に手間がかかる作業である。そこで、少ない手間で直感的にパラメータを設定できるインターフェースが求められている。

本研究は、ペンインターフェースを用いて群集アニメーションを直感的に作成できるシス

テムの開発を目的としている。

ユーザは、マウスやペンタブレットのような入力機器を用いて、群集の移動経路のサンプルになるような軌道を数本入力する。システムは入力された軌道に基づいて群集モデルのパラメータを自動的に計算する(図1(a))。計算された群集パラメータを用いて、エージェントモデルによる群集シミュレーションを行うことで、群集アニメーションを生成する(図1(b))。

システムは、入力された軌道から、目標経路、移動速度、エージェント同士の標準距離、距離の補正量の4つの群集パラメータを計算する。目標経路は、エージェントが移動する上での大まかな経路である。ただし、必ずしもエージェントは目標経路の上を移動するものではない。軌道を入力する速度に応じて群集の移動速度を決定する。また、入力された複数の軌道間の幅からエージェント同士の標準距離を決定する。

さらに、入力軌道のなめらかさでエージェント同士の標準距離に補正を与える。このようにユーザは軌道の入力に応じて、エージェントごとに独立したパラメータを決定する。群集パラメータは、全てエージェントの位置で変化する関数である。移動速度は全エージェントで共通で、残りのパラメータはエージェントごとに異なる値をとる。

ユーザが入力する軌道は、主線と補助線(図2)の2種類である。主線は群集パラメータの目標経路と移動速度を決定する1本だけの軌道である。補助線は群集の広がりとはらつきを決定する複数本の軌道である。補助線は1本以上入力することができる。基本的には、主線を挟む形で2本以上の補助線を入力することを想定している。たとえ、補助線が1本だとしても、群集パラメータを求めることは可能である。しかし、群集アニメーションの作りやすさを考慮すると、2本以上が好ましい。例えば、軌道の幅が広いところでは群集は広がって移動し(図3(a), (b)), 狭いところでは狭くなって移動する(図3(c), (d))。主線を素早く入力したところでは群集は素早く移動し、ゆっくり入力したところではゆっくり移動する。また、補助線をなめらかに入力するとばらつきの小さい群集になる(図3(a), (c))。ばらつきが小さいと、エージェント同士の距離はほぼ同じ値になる。なめらかでない補助線を入力すると、エージェント同士の距離がばらついた群集になる(図3(b), (d))。

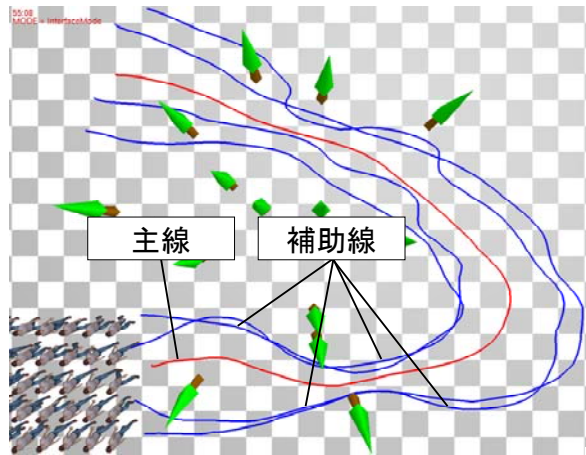


図2：入力軌道

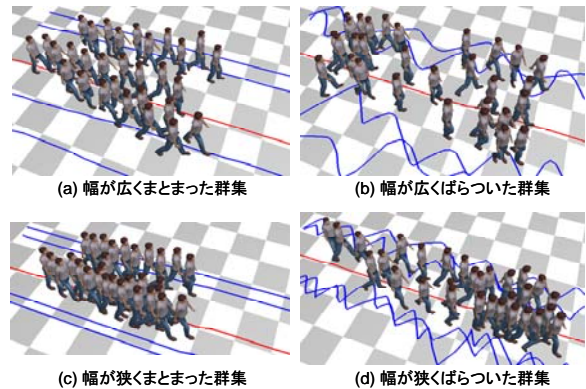


図3：入力軌道による群集の違い

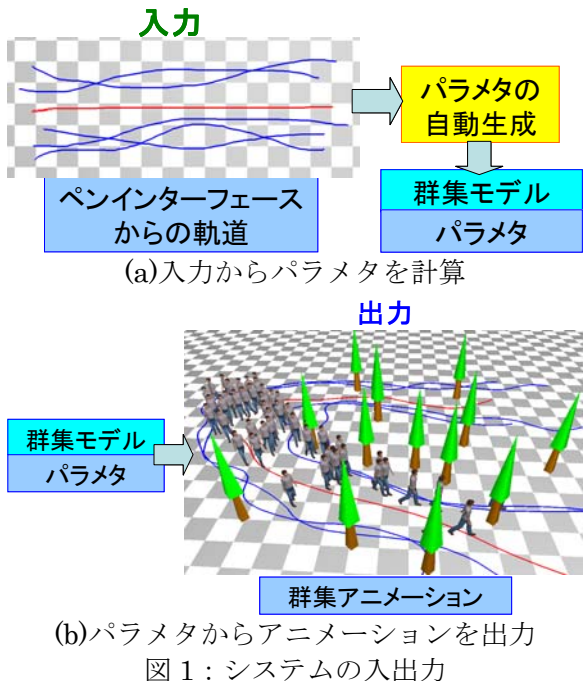


図1：システムの入出力

2. 関連研究

エージェント・モデル[1]は、群集アニメーション生成手法の中でも最も一般的な手法である。この手法は、一定のルールに従って各エージェントを独立に操作する。ルールの例として、リーダーに追従する、他のエージェントと一定の距離を維持するといったものがある。この研究[1]では、目標軌道に沿って移動するようなルールや、移動経路の特定の位置でエージェント同士の距離を変化させるルールは使われていなかった。本研究の群集モデルでは、これらのルールも取り入れている。

群集アニメーションのインターフェースに関する研究も行われている。Ulicnyら[2]は、ブラシ型のインターフェースを用いて、群集の一部もしくは全体に、移動の目標位置、移動速度、注視方向などのパラメータを用意に設定できるインターフェースを提案した。彼らの研究は、任意のエージェントへのパラメータ設定のしやすさに重点を置いており、群集の移動軌道は細かく設定できず、パラメータの値も手動で指定する必要がある。一方、本研究では、群集の移動に重点を置いており、目標経路や移動速度など

の移動に関連するパラメタを容易に設定できる。

Kwon ら[3]は、既存の群集の移動軌道を簡単なマウス操作によって編集することで、新しいシーンに適用する手法を提案した。この研究[3]では、軌道を生成するために、元となる群集の軌道をあらかじめ用意する必要がある。一方、本研究では、ユーザが入力した軌道に基づいて軌道を生成するため、元となる軌道は必要ない。

Karamouzas ら[4]は、1本の軌道を、テクスチャの生成で用いられる Parlin ノイズを用いて変化させ、移動軌道を生成する手法を提案した。この研究では、Parlin ノイズの係数をユーザが手動で指定する必要がある。一方、本研究では、生成した目標経路にノイズを与えてエージェントの移動に変化をつけることはできないが、エージェントの衝突回避等の外乱によって、同等の効果を得ることができる。

実在の人間の動作からルールを学習し、群集アニメーションを生成する研究も行われている。Couty ら[5]、Lee ら[6]、Lerner ら[7]は、動画から人間の移動軌道を得ることで、群集の歩行動作の動作ルールを抽出し、群集アニメーションを生成する。これらの研究では、人ごみの中をぶつからずに移動、門の間を並んで通り抜けることが可能となっている。また、実際のデータを利用しているため、手間はかかるが自然な動きを生成できる。本研究での入力軌道を、人間の移動軌道としてルールを抽出すれば、より自然な群集の移動を実現できると考える。

3. 群集パラメタ

本手法で求める群集パラメタは、目標経路 $\mathbf{P}_j \{ \mathbf{p}'_{j,1}, \mathbf{p}'_{j,2}, \dots, \mathbf{p}'_{j,K} \}$ 、移動速度 v_k 、エージェント同士の標準距離 s_k 、距離の補正量 $e_{j,k}$ により表せる。 \mathbf{p}'_k は 3次元のベクトルである。 $j \{ 1, 2, \dots, M \}$ は、エージェントの番号を表し、 M はエージェントの総数を表わす。 $k \{ 1, 2, \dots, K_j \}$ は、軌道の点の番号を表す。 K_j は軌道 i に含まれる点の数である。

目標経路 \mathbf{P}_j は、エージェントごとに異なる経路であり、エージェントが移動する際に目標位置とする点の集まりである。エージェントは目標経路上の点に近づく力の影響を受けて移動する。このパラメタは、ユーザが入力した軌道によって決まる大まかな経路である。エージェントは、目標経路に沿って移動するが、他のエージェントとの衝突回避や障害物との衝突回避を行う際は目標経路を外れる。

移動速度 v_k は、エージェントが移動する際の速度である。移動速度は、ユーザが主線を入力した際の入力速度に基づいて決定される。移動速度の値が大きい場所では群集は素早く移

動し、値が小さい場所では群集はゆっくり移動する。

エージェント同士の標準距離 s_k は、エージェントが移動する際に周囲のエージェントとどれだけの距離をとるかを表す値である。標準距離は、全てのエージェントで共通の値を持つ。

距離の補正量 $e_{j,k}$ は、エージェント同士の標準距離にどれだけのばらつきを与えるかを表す値である。距離の補正量は、エージェントごとに異なる値を持つ。

4. 群集パラメタの計算

群集パラメタは、入力軌道 $\mathbf{Q}_i \{ \mathbf{q}_{i,1}, \mathbf{q}_{i,2}, \dots, \mathbf{q}_{i,K_i} \}$ から求める。入力軌道の数を N とし、 $i \{ 1, 2, \dots, i, \dots, N \}$ である。 $i=1$ の軌道を主線とし、 $i \geq 2$ の軌道を補助線とする。また、生成される目標経路の点の数は主線の点の数と同数になる。求めた群集パラメタをもとに、群集の動きを生成する。

4.1 目標経路の生成

入力軌道（主線および補助線）、エージェントの初期位置 $\mathbf{p}_{j,0}$ 、エージェントの数から、各エージェントの目標経路を求める。なお、現在は、エージェントの初期位置とエージェントの数は、ユーザがあらかじめ手動で指定するものとしている。

エージェントの目標経路は、ユーザが入力した軌道を補間して生成する。現在では、軌道を補間する方法として、エージェントの初期位置に一番近い入力軌道の影響を強く受けて生成するようにしている。生成される目標経路の形状は、入力された主線と補助線の影響を受ける。しかし、補助線からは、大まかな形状の影響だけ受けるようにし、細かい形状は後述の距離の補正量を決定する際に用いる。

まず最初に、補助線の大まかな形状を得るために、平滑化を行う。補助線を一定間隔 δ で平均値を計算し、その値を平滑化した補助線 $\bar{\mathbf{q}}_{i,k}$ とする。ただし、 $(2 \leq i \leq N)$ の範囲をとり、補助線の点の間隔は十分小さいものとする。

$$\bar{\mathbf{q}}_{i,k} = \frac{1}{\delta} \sum_{d=-\delta}^{\delta} (\mathbf{q}_{1,k+d} - \mathbf{q}_{i,k+d}) \quad (1)$$

次に、目標経路の形状を決めるため、各軌道への距離の比率を決定する。エージェントの初期位置から各軌道までの距離を l_i とし、入力軌道の数を N 、 $i \{ 1, 2, \dots, i, \dots, N-1, N \}$ は軌道の番号とする。距離の比率を $o_{j,i}$ とすると、

$$o_{j,i} = \frac{1}{l_{j,i}^2 \times L_j} \quad (2)$$

$$\text{ただし } L_j = \sum_{i=1}^N \frac{1}{l_{j,i}^2} \quad (3)$$

となる。

最後に、距離の比率 $o_{j,i}$ に応じて移動経路の点 $\mathbf{p}'_{j,k}$ を求めることで、エージェント j の移動経路 $\mathbf{P}_j \{ \mathbf{p}'_{j,1}, \dots, \mathbf{p}'_{j,k}, \dots, \mathbf{p}'_{j,K_1} \}$ を生成する。 $\mathbf{p}'_{j,k}$ は、主線上の対応点 $\mathbf{q}_{1,k}$ から、平滑化した各補助線上の $\mathbf{q}_{1,k}$ に対応した点 $\bar{\mathbf{q}}_{i,k'}$ へのベクトルを $\mathbf{V}_{i,k}$ とすると、以下の式で求められる。

$$\mathbf{p}'_{j,k} = \sum_{i=1}^N (\mathbf{V}_{i,k} o_{j,i}) + \mathbf{q}_{1,k} \quad (4)$$

4.2 群集の移動速度

移動速度は、群集全体で共通の値を持つ。移動速度は、主線の入力速度から計算する。

移動経路は、入力された点の座標 \mathbf{q}_k とその点が入力されたときの時刻 t_k を用いて求める。ユーザが入力した軌道の線分の長さ Δl_k (メートル) と、入力にかかった時間 Δt_k から入力速度を求める。軌道の線分は $\Delta l_k = |\mathbf{q}_k - \mathbf{q}_{k-1}|$ で求め、入力にかかった時間は $\Delta t_k = t_k - t_{k-1}$ で求める。これらの値から、群集の移動速度 v_k を、以下の式で求める。

$$v_k = \frac{\Delta l_k}{\Delta t_k} \quad (5)$$

4.3 エージェント同士の標準距離

エージェント同士の標準距離は、入力された軌道の両端の幅が広いときに大きい値になり、幅が狭いときに小さい値になるようにする。ただし、際限なく小さくまたは大きくすることは、得策ではない。この理由は、エージェント同士の距離を小さくしすぎると、エージェント同士のめりこみが生じ、大きくしすぎても、エージェント同士が軌道の両端の幅以上に離れることは現実的ではないためである。

まず、軌道の両端の幅 W_k を求める。軌道の幅は、主線と平滑化された補助線を用いる。これは、ユーザが、距離の補正量を決めるために入力した、補助線の細かい形状の影響を与えないようにするためである。まず、幅を求めるために、主線の点 $\mathbf{q}_{1,k}$ における、両端の点 $\bar{\mathbf{q}}_{L,k}$ と $\bar{\mathbf{q}}_{R,k}$ を求める。主線を基準として、全軌道上の k の位置にある点 $\bar{\mathbf{q}}_{i,k}$ の中から、主線上の点 $\mathbf{q}_{1,k}$ に一番遠い点を片側の点 $\bar{\mathbf{q}}_{L,k}$ とする。その

後、 k に最も近い、 $\bar{\mathbf{q}}_{i,k}$ で一番遠い点をもう片側の点 $\bar{\mathbf{q}}_{R,k}$ とする。軌道の幅は、両者から計算できる。

$$W_k = |\bar{\mathbf{q}}_{L,k} - \bar{\mathbf{q}}_{R,k}| \quad (6)$$

エージェント同士の標準距離 $s_{j,k}$ は、軌道の幅 W_k に比例して変化する。軌道の幅が十分狭いときの幅を W_N 、十分広いときの幅を W_W とし、そのときのエージェント同士の標準距離をそれぞれ s_N 、 s_W とする。これらの値は固定とする。エージェント同士の標準距離を、以下の式で決定する。

$$s_{j,k} = \begin{cases} s_N & (W_k \leq W_N) \\ s' \times (W_{j,k} - W_N) + s_N & (W_N < W_k < W_W) \\ s_W & (W_k \geq W_W) \end{cases} \quad (7)$$

ただし、 $s' = \frac{s_W - s_N}{W_W - W_N}$ である。

4.4 距離の補正量

このパラメタは、エージェント同士の標準距離にばらつきを与えるパラメタである。距離の補正量は、補助線がなめらかであれば小さく、なめらかでなければ大きくなる。

まず、全てのエージェントで共通のばらつき E_k を求める。 E_k は、主線と各補助線との距離の変化量 $G_{i,k}$ の分散値から求める。まず、主線上の点 $\mathbf{q}_{1,k}$ と、対応した補助線 i 上の点 $\mathbf{q}_{i,k'}$ との微分を次の式で求める。

$$G_{i,k} = \frac{|\mathbf{q}_{1,k} - \mathbf{q}_{i,k'}| - |\mathbf{q}_{1,k+1} - \mathbf{q}_{i,k'+1}|}{|\mathbf{q}_{1,k} - \mathbf{q}_{1,k+1}|} \quad (8)$$

次に、 $G_{i,k}$ の平均値 $AVG_{i,k}$ を軌道ごとに求める。ただし、 $G_{i,k}$ を中心に一定間隔 δ で平均をとる。

$$AVG_{i,k} = \frac{1}{\delta} \sum_{d=-\delta}^{\delta} G_{i,k+d} \quad (9)$$

この平均値を使って各距離の変化量の分散値 $VAR_{i,k}$ を求める。分散値は、

$$VAR_{i,k} = \frac{1}{K_i} \sum_{k=1}^{K_i} (G_{i,k} - AVG_{i,k})^2 \quad (10)$$

となり、全体の分散値 VAR_k は、 N を入力された軌道の数とすると

$$VAR_k = \frac{1}{N} \sum_{i=1}^N VAR_{i,k} \quad (11)$$

となり、これに係数 α をかけたものが共通のばらつき E_k になる。

$$E_k = \alpha \text{VAR}_k \quad (12)$$

次に、エージェントごとに異なる距離の補正量 $e_{j,k}$ を求める。エージェントごとに違いを出すため、 E_k に乱数をかける。乱数の範囲は 0.5 から 1 で、これは、距離の補正量を乱数によって小さくしないためである。 E_k に共通の補正量の最大値 E_{MAX} で割った割合で求める。 E_{MAX} は定数とする。

$$e_{j,k} = \text{random}(0.5,1) \times \frac{E_k}{E_{MAX}} \quad (13)$$

$\text{random}(0.5,1)$ は 0.5 から 1 までの乱数を生成する関数とする。

5 群集アニメーションの生成

本手法では、エージェントモデル[1]により各エージェントの動きを生成する。本手法のエージェントモデルでは、軌道に追従する力、他のエージェントとの衝突を回避する力、障害物との衝突を回避する力を用いる。各エージェントごとにこれらの 3 つの力ベクトルを求め、全ベクトルの合計により移動方向を決定する。力は方向と大きさを持つ 3 次元のベクトルである。これらの力は毎ステップごとに計算する。

5.1 目標経路に追従する力

目標経路に追従する力は、エージェントが目標経路上の目標位置から離れているほど大きくなる。目標位置とは、エージェントが目標経路に追従する際に、目標とする座標である。

時刻 t でのエージェント $j \{1, 2, \dots, j, \dots, M\}$ 自身の位置 $\mathbf{p}_{j,t}$ から、目標経路上の目標位置 $\mathbf{p}'_{j,k}$ に向かう力は

$$\mathbf{F}_{\text{trajectory},j,t} = (\mathbf{p}'_{j,k} - \mathbf{p}_{j,t}) \left(1 - \frac{R}{|\mathbf{p}'_{j,k} - \mathbf{p}_{j,t}|} \right) \quad (14)$$

となる。

5.2 他のエージェントと衝突回避する力

他のエージェントと衝突回避する力は、エージェントが他のエージェントと衝突をしないために、距離をとろうとする力である。エージェント同士が衝突を回避する距離よりも近い場合、エージェントには離れる力が生じる。

エージェント同士が衝突を回避する距離 $r_{j,k}$ は、エージェント同士の標準距離に距離の補正量をかかけたものである。

$$r_{j,k} = s_{j,k} \times e_{j,k} \quad (15)$$

他のエージェント (位置 $\mathbf{p}_{1,t}$) との距離が、衝

突を回避する距離より近い場合は回避する。つまり、 $|\mathbf{p}_{1,t} - \mathbf{p}_{j,t}| < r_{j,k}$ なら回避する力が発生する。エージェント 1 との衝突を回避する力は

$$\mathbf{F}_{\text{avoid},j,t} = -(\mathbf{p}_{1,t} - \mathbf{p}_{j,t}) \left(1 - \frac{r_{j,k}}{|\mathbf{p}_{1,t} - \mathbf{p}_{j,t}|} \right) \quad (16)$$

となる。この力を残りのエージェント全てに対して計算し、合計した値が衝突回避する力となる。

$$\mathbf{F}_{\text{avoid},j,t} = \sum_{j=1}^M \mathbf{F}_{\text{avoid},j,t} \quad (17)$$

となる。

5.3 障害物を回避する力

障害物を回避する力は、エージェントが障害物に衝突しないために、その障害物と距離をとろうとする力である。

障害物 x (位置 $\mathbf{p}_{\text{block},x}$) に衝突しない最小半径を $r_{\text{block},x}$ とする。つまり、

$|\mathbf{p}_{\text{block},x} - \mathbf{p}_{j,t}| < r_{\text{block},x}$ なら回避する力が発生する。障害物 x との衝突を回避する力は

$$\mathbf{F}_{\text{block},j,x,t} = -(\mathbf{p}_{\text{block},x} - \mathbf{p}_{j,t}) \left(1 - \frac{r_{\text{block},x}}{|\mathbf{p}_{\text{block},x} - \mathbf{p}_{j,t}|} \right) \quad (18)$$

となる。この力を残りの障害物全てに対して計算し、合計した値が障害物を回避する力となる。障害物の数を X とすると

$$\mathbf{F}_{\text{block},j,t} = \sum_{x=1}^X \mathbf{F}_{\text{block},j,x,t} \quad (19)$$

となる。

5.4 エージェントの移動

エージェント j は、各力の合計 $\mathbf{F}_{j,t}$ の方向に移動速度 v_k で移動する。合計した力は

$$\mathbf{F}_{j,t} = w_{\text{trajectory}} \mathbf{F}_{\text{trajectory},j,t} + w_{\text{avoid}} \mathbf{F}_{\text{avoid},j,t} + w_{\text{block}} \mathbf{F}_{\text{block},j,t} \quad (20)$$

となる。ここで $w_{\text{trajectory}}, w_{\text{avoid}}, w_{\text{block}}$ は各力の重みである。現在は我々の実験では、衝突回避に関する重みを大きくすることは重要であるため、 $w_{\text{trajectory}} : w_{\text{avoid}} : w_{\text{block}} = 1 : 2 : 2$ としている。

移動量 $\mathbf{D}_{j,t}$ は

$$\mathbf{D}_{j,t} = v_k \frac{\mathbf{F}_{j,t}}{|\mathbf{F}_{j,t}|} \quad (21)$$

となる。この移動量を現在のエージェントの位置に加えることで、エージェントを移動させる。

次のステップのエージェントの位置 $\mathbf{p}_{j,t+1}$ は

$$\mathbf{p}_{j,t+1} = \mathbf{p}_{j,t} + \mathbf{D}_{j,t} \quad (22)$$

となる. エージェントの移動後に, エージェントの位置と目標位置との距離が, 目標経路への追従距離 R 以下のときに, 目標位置を次の点に更新する. 以上の計算を各ステップで行うことによってアニメーションを生成する.

6. 実験結果と考察

本システムを用いて, ユーザが思った通りの群集アニメーションが作成できるかを実験した (図 5). 実験に使用したコンピュータのスペックは, CPU は 3.20GHz, メモリは 1024MB, グラフィックボードは GeForce6600GT である. 作成するシーンには, 群集の移動速度, 群集の広がり, 群集のばらつきを変化させる箇所を設け, 群集の数は 50 人とする (図 4).

実験の結果, 図 4 のような入力軌道を与えることで, ユーザの意図通りの動きを実現することができた. なお, 本システムは, パラメタ設定の容易さから, 数千人規模の大群集を扱うことに適していると考えられる.

7. 今後の課題

本研究では, 数本の軌道入力により群集パラメタを決定することで, ユーザが用意に群集アニメーションを作成できるようなシステムを提案した. 今後の課題として, 軌道上の任意の位置で, エージェントに移動しながらの会話や剣や槍等の武器を使った戦闘などの行動を行わせることができるようなインターフェースも追加したいと考える. 具体的には, 目標経路上に動作を設定し, 他のエージェントと連携した動きを生成する.

また現在, エージェントの初期位置は手動で設定する必要がある. エージェントを配置する手間を考えると, 初期位置も入力軌道から自動的に決定することが望ましい. そこで, ユーザが軌道を入力後に, エージェントの数を手動で指定する, 入力軌道の開始位置の両端の間に, エージェントを自動的に配置する機能を追加する.

加えて, ユーザは移動経路を始点から終点まで指定する必要がある, 複雑な移動経路や広い場所での移動を実現するのに多少手間がかかる. そのため, 長い移動経路なしで群集アニメーションを作成できることが望ましい. そこで, エージェントが曲がる頻度などのパラメタを追加し, 短い入力軌道からそれらを求めるようにする.

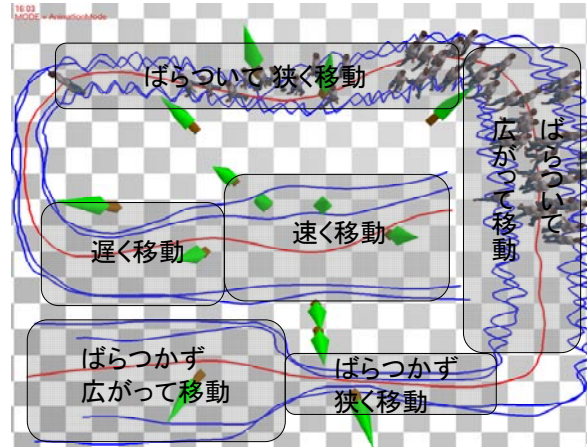


図 4 : 群集のコース

参考文献

- [1] Creg W. Reynolds, "Flocks, Herds, and Schools: A Distributed Behavioral Model", *SIGGRAPH'87*, pp.25-34, 1987.
- [2] Branislav Ulicny, Pablo de Heras Ciechowski, Daniel Thalmann, "Crowdbrush: Interactive Authoring of Real-time Crowd Scenes", *ACM SIGGRAPH2004*, pp. 243-252, 2004.
- [3] Taesoo Kwon, Kang Hoon Lee, Shigeo Takahashi, "Group Motion Editing", *ACM SIGGRAPH2008*, 2008.
- [4] Ioannis karamouzas, Mark H. Overmars: "Adding variation to path planning", *Comp. Anim. Virtual Worlds 2008*, pp.283-293, 2008.
- [5] N. Courty, T. Corpetti, "Crowd Motion Capture", *Computer Animation and Virtual Worlds*, Volume 18, Issue 4-5, pp. 361-370, 2007.
- [6] K. H. Lee, M. G. Choi, Q. Hong, J. Lee, "Group Behavior from Video: A Data-Driven Approach to Crowd Simulation", *ACM SIGGRAPH/Eurographics symposium on Computer animation*, pp. 109-118, 2007.
- [7] A. Lerner, Y. Chrysanthou, D. Lischinski, "Crowds by Example", *Computer Graphics Forum*, Volume 26, Number 3, pp. 655-664. 2007.