

生物集団シミュレーションで現れる運動状態の自動分類

猪股 能成¹ 高見 利也¹

概要: 本研究は人工知能技術の1つである群知能を理解するために、生物的なエージェントが持つパラメータなどの差異による群れとしての運動状態の変化を解析することが目的である。そのために、シミュレーションツールの開発と、そこから得られたデータを簡単な指標の時間変化や主成分分析による解析を行うことで、運動状態ごとの分類を目指した。簡単な指標の時間変化では、群れとしての振る舞いがある時とない時で明らかな違いが見られた。主成分分析の方では、2つの運動状態の分類に加え、初期状態から各運動へ遷移する様子も確認出来た。

Automatic classification for dynamic states emerged by simulation of biological group

YOSHINARI INOMATA¹ TOSHIYA TAKAMI¹

Abstract: The purpose of this study is to analyze the change in dynamic states of flocks induced by the difference of parameters of biological agents. Using a tool that we have developed for multi-agent simulations, we try to classify the dynamic state through the temporal change of simple indicators calculated from statistical information and the principal component analysis (PCA) of dynamic variables. By the temporal change of the indicators, we observed a clear difference between flocking states and other states. With the PCA, we achieved a successful classification of two different states, and observed transitions from a random state to a specific state.

1. はじめに

近年著しい発展をしている人工知能は、自動運転をはじめ、様々な分野で活躍している。人工知能とは、機械学習、画像認識、自然言語処理など、知的な振る舞いをする技術や研究のことであるが、その中の1つに群知能 (Swarm Intelligence) と呼ばれる技術がある。

群知能とは、単純な機能を持つエージェントの集まりが群れを創発させるアルゴリズムの総称である。群知能の大きな特徴は、たくさんのエージェントを動かす際に、中心的な制御構造、いわゆるリーダーを必要としないことで、各々のエージェントが周囲のエージェントと相互作用したり、環境と対話することによって群れとしての振る舞いを創発させる。具体的な群知能アルゴリズムには、粒子群最適化や蟻コロニー最適化 (ACO) などがあり、最適化問題、

ロボット工学 [1]、データマイニング [2] 等で活用されている。ACO は、蟻が自身の巣と餌を結ぶ最短経路を見つける様子から提案されたアルゴリズムで、主に経路探索問題で用いられる。ACO の他にも自然界からヒントを得た群知能アルゴリズムは多数存在しており、このような社会性生物は個々の能力が低いながらも群れ全体では高度な振る舞いを創発することが知られている。

一方で、実際の生物集団から創発される群れは、ACO に用いられた蟻の行列のほかにも、魚の渦運動、鳥の V 字飛行、蚊柱などのように構成している個体の数や形状がバラバラである。このような生物集団の運動を解析するために、カメラ等で実際の生物集団を計測した研究 [3] が行われているが、計測が困難である場合が多いため、シミュレーションを用いる場合が多い。例として、阪上氏らは、特定の運動状態について、実際に測定した値とシミュレーションでの値を比較した上で、測定できなかった項目をシミュレーションから得ている [4]。また、古場氏らは、実

¹ 大分大学
Oita University, Oita-shi, Oita 870-1192, Japan

際には撮影が困難な状況での群れの動きを、シミュレーションによって再現した [5]。これらの研究では、群れをシミュレーションするためのエージェントモデルとして、丹羽モデル [6] と Boid モデル [7] がそれぞれ用いられている。複雑な集団運動は個々のシンプルな相互作用によって創発されること [8] が知られており、丹羽モデル、Boid モデルをはじめとして、群れを再現するために考えられたシミュレーションモデルに相互作用を持たせているものが多い [9], [10], [11]。

群知能を利用したシステムの開発を行うために、システムで用いるアルゴリズムやエージェントの持つパラメータが変わることによって群れとしての振る舞いがどのように変化するかを解析することが必要があると考えている。そこで、群知能アルゴリズムを群れのシミュレーションモデルに置き換え、創発される群れの運動状態に対して一定の解析を行うことで、状態ごとの自動分類をしようと考えた。

しかし、先述した研究では、様々な群れの運動状態に対して、定量的に解析によって分類を行った研究は見られなかった。したがって本研究では、シミュレーションで様々な運動状態を計測し、そこから得られたデータを用いて運動を分類するための解析を行うことを目的とする。

研究の方針としては、まず初めに、いくつかのシミュレーションモデルについて検討を行った上で、本研究で使用するシミュレーションモデルを決定する。そして、様々な運動状態を計測するためにシミュレーションを簡単に実行できて、データの収集を行いやすいツールの開発を行った上で、得られたデータから分析を行うことで運動状態ごとに分類できるかどうかを確認する。

第 2 節では、3 つのシミュレーションモデルの紹介と使用するモデルの決定を行う。第 3 節では、本研究で行う 2 つの分析方法を紹介し、第 4 節でシミュレーションツールの中身を説明し、実装する。そして、そのシミュレーションツールから得られたデータを用いて、分析を行った結果を第 5 節で確認し、第 6 節で結論と今後の課題を述べる。

2. 群れのシミュレーションモデル

本節では、群れの運動を再現するために、3 つのモデルについて具体的な内容を説明した上で、シミュレーションで用いるモデルの検討を行う。

2.1 Boid モデル

Boid モデル [7] は、1987 年に Reynolds 氏が提案した群れのシミュレーションモデルである。Separation (分離), Alignment (整列), Cohesion (結合) と呼ばれる 3 つの局所的な相互作用を各エージェントに持たせることで、そのエージェントの集合が群れらしい振る舞いをする。各相互作用の簡単な説明とイメージ (図 1) を以下に示す。

- **Separation** : 最も近くにいるエージェントに衝突し

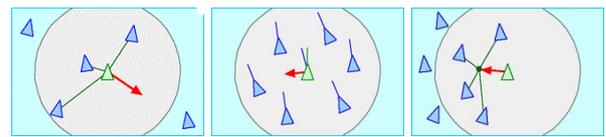


図 1 Boid モデルの相互作用 : 左から, Separation, Alignment, Cohesion (引用 : <https://www.red3d.com/cwr/boids/>)

Fig. 1 Interaction rules of boid model (Source: <https://www.red3d.com/cwr/boids/>)

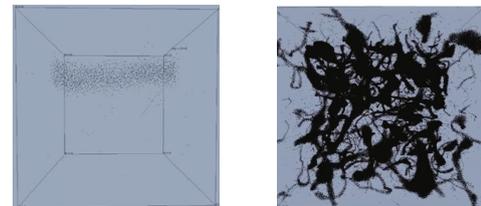


図 2 個体数が $2,048 (= 2^{11})$ と $524,288 (= 2^{19})$ の Boid シミュレーション ([13] Fig.1 より引用)

Fig. 2 Boids simulation; $N = 2^{11}$ and 2^{19} (Source: [13] Fig.1)

ないように進行方向をとる

- **Alignment** : 周囲のエージェントと進行方向を合わせる
- **Cohesion** : 他のエージェントがいる方向へ進行方向を変える

これらの相互作用は、その作用を適用させる範囲が各々設定されており、各エージェントは、この適用範囲を超えて相互作用をすることはない。Couzin 氏らによる研究 [10] では、Boid モデルと同等のモデルに対し、相互作用の適用範囲を適切に与えることによって、Swarm, Torus, Dynamic parallel group, Highly parallel group という 4 つの異なる運動状態が観測されている。

また、本武、池上両氏による発表では、およそ 50 万体の Boid でシミュレーションした様子が報告されている [12]。そこでは、数百・数千体のシミュレーション (図 2:左) では見られないような複雑な群れ (図 2:右) が観察されたと述べている。

2.2 坂井モデル

坂井氏は 1973 年に、群れが球状になって移動する状態を創発させるゴンズイを参考にして群れの運動モデルを発表した [9]。発表された論文内では、個体に働く力を 4 つに分けて説明し、その合力を一般的な運動方程式に適用させてシミュレーションを行っている。

個体に働く力の簡単な説明を以下に示す。

- (i) 前向走性 常に前向きに進もうとする力
- (ii) 接近運動 仲間のいる方向への引力と、近づきすぎたときの反撥力
- (iii) 整列運動 仲間と向きを含めた速度を揃えようとする力
- (iv) 擾乱作用 確率的に進行方向を変える力

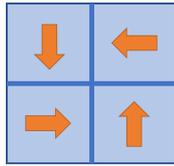


図 3 初期速度方向

Fig. 3 Velocity direction of first step

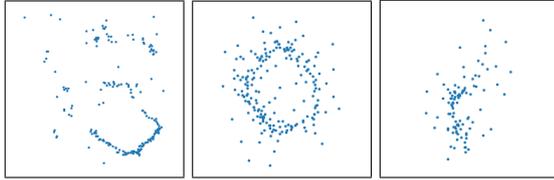


図 4 実装した Boid シミュレーション

Fig. 4 Simulation of implemented boids

論文内では、(i)~(iv) にそれぞれ与えられている係数を変化させてシミュレーションしており、その結果、主に 3 種類の運動が確認されていた。

2.3 Vicsek モデル

Vicsek 氏らは、粒子の速度がシンプルなルールやランダムなゆらぎによって決定されるような非平衡システムの相転移などを調べるためのモデルを発表した [11]。このモデルは以下の式によって定義される。

$$\mathbf{x}_i(t+1) = \mathbf{x}_i(t) + \mathbf{v}_i(t)\Delta t \quad (1)$$

$$\theta(t+1) = \langle \theta(t) \rangle_r + \Delta\theta \quad (2)$$

粒子 i は、自身を中心とした半径 r 以内に存在する粒子（自身を含む）の平均速度方向と、区間 $[-\eta/2, \eta/2]$ から一定の確率で選ばれる乱数との和によって、次のステップでの速度方向 $\theta(t+1)$ が決定する。なお、速度 \mathbf{v}_i の絶対値は一定である。このように簡単な式によって表せる Vicsek モデルはアクティブマター等の研究分野でよく用いられている。

また、論文内では、乱数の幅 η 、シミュレーションでの粒子の密度 ρ を適当に選びながらシミュレーションを行っており、エージェント全体が同じ方向を向いて進んでいたり、全く別の方向を進んだりと様々な結果が得られていた。

2.4 使用するモデルの決定

使用するエージェントモデルを決めるにあたって、モデルに求める条件を考えたい。まず、冒頭で述べたように、様々な運動状態を作れることが挙げられる。そして、それらの運動状態が乱数などによる偶発的なものではない方が良いと考えられる。それは、あるパラメータで行うシミュレーションにおいて追加のデータが必要になった際、再現性を維持していないと全データを再度取得する必要があると想定されるためである。

以上のことを踏まえ、本研究では Boid モデルを用いる

のが適切であると判断した。Boid モデルは、実装する目的などによって、エージェントの速度の大きさを可変にしたり、領域外に出ていこうとするエージェントの処理等が柔軟に決められる。そこで、本研究では以下のような条件でモデルを実装した。まず、各エージェントの速度ベクトルの大きさは一定に保ち、速度方向のみを更新していく。次に、シミュレーション領域から出ようとするエージェントは、速度方向を反転させ、領域内に閉じ込めておく。さらに、初期の進行方向は、位置座標に応じて図 3 のように決定する。これを決めた理由としては、シミュレーションの開始直後にエージェント同士が接触することを避けるためである。

Boid モデルは、相互作用に応じて速度を決定するのだが、その計算アルゴリズムを以下に示す。

- (1) 自身以外のエージェントとの距離を計算し、リスト化
- (2) 各相互作用の計算（適用範囲にエージェントがいなければ今の速度を維持）
- (3) 上の計算結果を合算し、加速度 \mathbf{a} を計算
- (4) 一般的な加速度の公式 $\mathbf{v}_{t+1} = \mathbf{v}_t + \mathbf{a}\Delta t$ を適用
- (5) \mathbf{v}_{t+1} を正規化し、大きさを一定にする

これらを実装し、シミュレーションした様子を切り取ったものを図 4 に示す。

3. 解析方法

Boid のようなマルチエージェントシミュレーションでは、群れの状態はすべてのエージェントの座標と速度ベクトルで表される。しかしながら、2次元空間のシミュレーションでは、各ステップのデータは「エージェントの 4 倍」の次元を持つため、分析するには非常に膨大である。そこで、本研究では 2 つの方法でデータ次元の削減を行い、運動状態ごとに分類できるかどうかを確認する。

3.1 簡単な計算指標

エージェントシミュレーションでは、バラバラに配置された初期状態から特定の運動状態が創発することや、ある運動状態から別の運動状態へ遷移することを想定し、シミュレーションの各ステップごとに計算できるような指標を計算する。本研究では、群れにおける基本的な情報として考えられる以下の指標について解析を行う。

- 群れの半径： r
- 群れの中心と各個体間の距離の分散： S^2
- 縦・横方向での個体位置の分散： S_x^2, S_y^2
- 平均速度ベクトルの大きさ： $|\bar{\mathbf{v}}|$

それぞれの計算式は以下のとおりである。

$$r = \frac{\sum_{i=0}^N \sqrt{(\mathbf{x}_i - \bar{\mathbf{x}})^2 + (\mathbf{y}_i - \bar{\mathbf{y}})^2}}{N} \quad (3)$$

$$S^2 = \frac{\sum_{i=0}^N \{ \sqrt{(\mathbf{x}_i - \bar{\mathbf{x}})^2 + (\mathbf{y}_i - \bar{\mathbf{y}})^2} - r \}^2}{N} \quad (4)$$

$$S_x^2 = \frac{\sum_{i=0}^N (\mathbf{x}_i - \bar{\mathbf{x}})^2}{N}, S_y^2 = \frac{\sum_{i=0}^N (\mathbf{y}_i - \bar{\mathbf{y}})^2}{N} \quad (5)$$

$$|\bar{\mathbf{v}}| = \left| \frac{\sum_{i=0}^N \mathbf{v}_i}{N} \right| \quad (6)$$

ここでの $\mathbf{x}_i, \mathbf{y}_i, \mathbf{v}_i$ は、それぞれエージェント i の x 座標、 y 座標、速度ベクトルのことで、 N はエージェントの数を表している。これらの指標によって、 $4N$ 次元のデータが5次元まで削減することができる。

3.2 主成分分析

主成分分析とは、多数の変数からデータ全体におけるバラツキを表す変数を合成する解析方法である。データの次元を削減するためによく用いられているため、本研究でも採用する。主成分分析は、主成分と呼ばれる変数の合成を行い、求めた順に第1主成分、第2主成分、…、というように呼ばれる。あるデータから第1主成分を求める場合、元のデータ行列 \mathbf{X} と第1主成分に対応するベクトル \mathbf{w}_1 について、以下の条件を満たすような \mathbf{w}_1 を求めなければならない。

$$\mathbf{w}_1 = \arg \max_{\|\mathbf{w}\|=1} |\mathbf{X}\mathbf{w}| \quad (7)$$

第1主成分が求まり、第2主成分を求める場合には、元のデータ行列から第1主成分を除いたデータ行列 $\hat{\mathbf{X}} (= \mathbf{X} - \mathbf{X}\mathbf{w}_1\mathbf{w}_1^T)$ について同様の条件で求めることができる。

本研究では、シミュレーションで得られたデータから第3主成分まで求め、2次元平面と3次元空間に射影し、可視化を行う。その可視化されたデータが運動状態ごとに分類できるかどうかを確認する。

しかしながら、主成分分析を用いるためには元のデータが行列である必要があるため、エージェント数の異なるシミュレーションデータをまとめて主成分分析することはできない。そこで、一旦多次元のデータを位置座標や速度ベクトルのごとに分散と共分散を計算した10次元のデータにしてから主成分分析する。

4. ツールの作成

2.4節で実装したBoidモデルを用いて、様々な運動状態をシミュレーションする。2.1節で述べたように、モデルのパラメータ等を変更することで運動状態を変化させられるため、ツールを開発することによって、それらを簡単に行えるようにしてデータを効率良く収集する。

以下では、パラメータの変更などといった、ツールに必要なGUIの作成と、シミュレーションの実行、表示などを行うプログラムの作成を行う。ツール内で行われる処理は図5に示す。

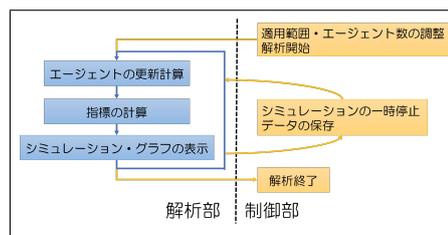


図5 ツールでの処理の流れ

Fig. 5 The flow of processing in our tool

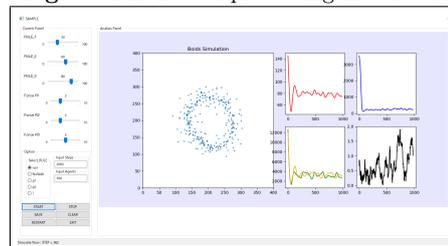


図6 シミュレーションツールの様子

Fig. 6 Snap of the simulation tool

4.1 制御部

Boidモデルは、3つある相互作用の適用範囲を適当に変えることで、いくつかの異なる運動形態を観察することができる。また、エージェントの数を増やすことによって複雑な運動を観察することも出来る。このことから、まず最初に、適用範囲とエージェントの数を容易に変えられるようなGUIの設置を行う。エージェントの相互作用範囲は、感覚的に調整できた方が良く考え、スライダーのGUIを各相互作用ごとに設置する。また、シミュレーションを行うエージェントの数の変更には、直接入力によって指定できるGUIを設置する。さらに、シミュレーションの開始・終了・再開などシミュレーションを制御する上で必要となる機能はボタンのGUIで実装する。

4.2 解析部

シミュレーションでは、2次元のBoidモデルを各エージェントに適用し、逐次、位置の移動に関する計算を行う。計算の順序としては、速度ベクトルの計算を全エージェントに対して行った後に、位置座標の更新を全エージェントで行う。このようにする理由は、Boidモデルが相互作用の適用範囲を持っているため、速度ベクトルの計算をする際に位置座標が更新されたエージェントと更新されていないエージェントが混在するのを防ぐためである。

画面上には、エージェント位置が更新されていく様子を表示させるとともに、その横には3.1節で定めた4つの指標を計算したグラフを表示させる。そうすることによって、シミュレーションで観測される運動状態の変化と、指標の時間変化を捉えやすくなり運動状態ごとに現れる特徴を見つけられるのではないかと考える。そのため、図5のように、エージェントの更新計算を行った後に各指標の計算を行い、画面表示するという処理の流れになっている。

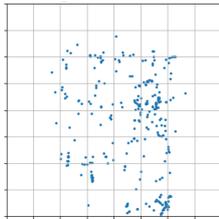


図 7 分散状態
 Fig. 7 Scattered state

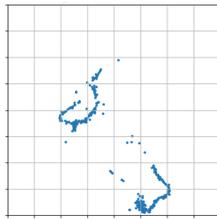


図 8 ひも状運動
 Fig. 8 String-like state

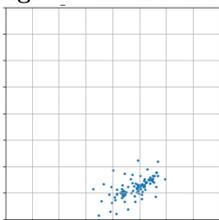


図 9 並進運動
 Fig. 9 Parallel state

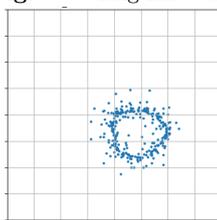


図 10 渦運動
 Fig. 10 Torus

ここまでで述べてきたものを Python で実装し、実際にツールを動かしている様子が図 6 である。

5. 解析結果

5.1 シミュレーション結果

我々は、エージェントの相互作用範囲または個体数をいくつかの値に変えてシミュレーションを 2000 ステップずつ行った。

5.1.1 エージェント数が一定で、相互作用範囲を変えた場合

エージェント数を 300 体に固定して、相互作用範囲を様々な値に変えてシミュレーションした。3つの相互作用範囲がいずれも狭い場合では、群れとしての振る舞いが見られず、エージェントはバラバラに動いているような状態(分散状態・図 7) だった。しかし、相互作用範囲が広くなるにつれて、エージェント同士が近づいたり、同じ方向へ進むようになる。ある相互作用範囲の時には、ひも状の形ができたり、なくなったりする不安定な状態(ひも状運動・図 8) から渦のように群れ全体が回転しているような状態(渦運動) に遷移したパターンが見られ、それよりも大きな範囲になると、渦運動を行い続ける状態(図 10) が観察された。

5.1.2 相互作用範囲が一定で、エージェント数を変えた場合

こちらでは、エージェントが 300 体の時に安定して渦運動が見られた相互作用範囲を用いて、エージェント数を 10~1000 体に変えてシミュレーションした。その結果、エージェント数が少ない場合は、渦運動ではなく、群れ全体が同じ方向を向き空間内を移動している状態(並進運動・図 9) が見られ、数が増えるにつれて渦運動がみられるようになった。

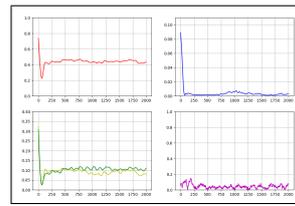


図 11 渦運動
 Fig. 11 Torus

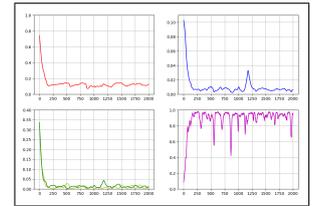


図 12 並進運動
 Fig. 12 Parallel state

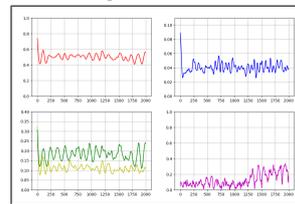


図 13 分散状態
 Fig. 13 Scattered state

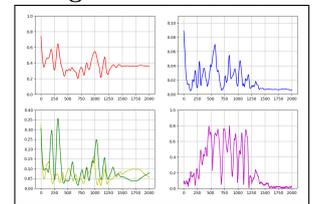


図 14 ひも状→渦運動
 Fig. 14 String-like → Torus

5.2 簡単な計算指標

3.1 節で定めた計算指標の時間変化を運動状態ごとに比較を行う。今回観察できたグラフのいくつかを図 11~14 に示す。各図の左上(赤)は式(3)、右上(青)は式(4)、左下(黄・緑)は式(5)、右下(紫)は式(6)から計算されたグラフである。

渦運動や並進運動といった、群れとしての運動状態が見られ続けた時(図 11 と図 12) は、右下以外のグラフ(位置に関する指標)の時間変化に大きな変動は見られなかったが、分散状態(図 13) では比較的大きな変動が見られた。ひも状運動から渦運動に遷移したパターン(図 14) のグラフでは、渦運動になってから値の変動が安定するようになるのが確認された。

各図右下にある平均速度ベクトルの大きさの時間変化では、並進運動が 1 に近い値を比較的推移しているのに対して、渦運動では 0 に近い値で推移した。これは、並進運動をしているエージェントはすべて同じ方向を向いて動いているため、平均をとっても本来のエージェントが持つ速度の大きさ(=1)に近い値で推移したと考えられる。一方で、渦運動はある点を中心に時計回り(または反時計回り)でエージェントが動いているため、反対側にいるエージェントの速度ベクトルと相殺される形でこの指標が計算されたためだと考えられる。

このように簡単に計算できる指標でも、時間変化を追うことで運動状態ごとに特徴があることが確認できた。そのため、時間変化を加味した秩序変数を作成することで運動状態の自動分類を実現できるのではないかと考えられる。

5.3 主成分分析

相互作用範囲が一定で、エージェント数を変えた場合から得られたデータを主成分分析した。3.2 節で述べたように、データの次元数が異なるため、分散・共分散を計算した 10 次元のデータに替えて行い、その結果を図 15 に示す。

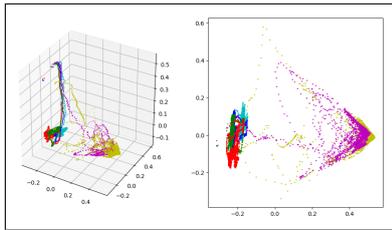


図 15 主成分分析 (相互作用範囲一定)
Fig. 15 Result of PCA
(Interaction range is constant.)

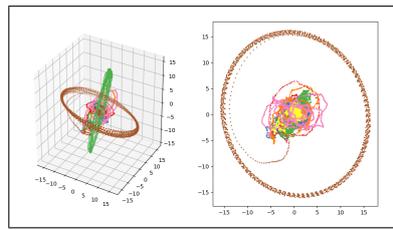


図 16 主成分分析 (エージェント数一定)
Fig. 16 Result of PCA
(Agent number is constant.)

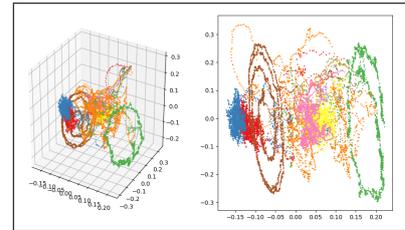


図 17 主成分分析 (エージェント数一定)
Fig. 17 Result of PCA
(Agent number is constant.)

図の右半分は第2主成分までをプロットしたもので、左半分は第3主成分までプロットした結果となっている。このシミュレーションでは、並進運動と渦運動が見られたのだが、これらが主成分分析によって大きく分かれたプロットされる結果となった。図の紫・黄でプロットされているのが並進運動で、赤などでプロットされているのが渦運動である。また、第3主成分まで表示させると、エージェントがランダムに配置された状態(図の左上)から各状態への遷移するところも見られた。

一方で、エージェント数が一定で、相互作用範囲を変えた場合から得られたデータを主成分分析した結果を図16に示す。第3主成分までプロットされた結果を見ると、茶色と緑色の点が他よりも突出している。この2つは、渦運動の中でも、よりエージェント群が1本の紐のような形状が見られたパターンだった。中心付近で混在しているパターンの中には、幅を持った渦運動や分散状態などが含まれており、先述したパターンのみが誇張される結果となった。また、こちらのデータについても、分散と共分散を計算したデータに置き換えて主成分分析した結果を図17に示す。

元のデータをそのまま主成分分析したときに比べて、各々のパターンが分かれるようにはなったが、紐に近い形状をした渦運動のパターンである茶色と緑色の点が近くにプロットされなかった。他のパターンに関しても、運動状態ごとに分類できるまでは至らなかった。この原因としては、主成分分析が3.2節で述べたように、「データ全体におけるバラツキを表す変数を合成する解析方法」であることが考えられる。改善するためには、データの与え方を工夫するか、主成分分析以外の解析方法の検討が必要だろう。

6. まとめ

6.1 結論

本研究では、創発される様々な群れの運動状態について、シミュレーションツールを用いてデータを収集し、2つの方法で解析を行うことで運動状態ごとの分類を目指した。

簡単な計算指標を用いた場合では、位置に関する指標と速度に関する指標のそれぞれで時間変化に運動状態の分類につながるような違いが見られた。エージェントの位置に関する指標では、群れとして振る舞いが安定である場合と

そうでない場合で明らかな違いが見られ、速度に関する指標の方では、安定した振る舞いが見られた渦運動と並進運動で対照的な違いを見ることができた。

また、主成分分析を用いた場合、2種類の運動状態を含んだデータでは、各運動状態が大別するだけでなく初期状態からの遷移までを確認することができ、有効であったと考えられる。しかしながら、多くの運動状態を含んだデータだと、1つの運動状態のみが分類できるような結果となり、各運動状態ごとに分類するまでには至らなかった。

6.2 今後の課題

今回は2つの解析を通して、グラフの形状や、プロットされた点の分布などで視覚的に分類を試みた。今後は、これらを取り入れつつ、より定量的な解析を行うことで運動状態の自動分類ができるようにしたいと考えている。そのための課題として、以下の3つをあげる。

まず、解析方法の再検討である。主成分分析は、解析結果で述べたように多くのパターンを含んだ場合に上手く分けられない可能性がある。そのため、動的モード分解[8]などといった別の解析方法についても適用を検討していくことが必要である。

2つ目に、今回シミュレーションできたパターン以外の運動状態を新たに観測することである。今回は主に4つの運動状態が観測できたが、実際にはまだ多くの運動状態が存在すると考えられる。それらに対して、今回用いた方法を含めた解析方法の有用性を検討するためにも、運動状態のパターンは増やす必要がある。

3つ目は、Boidモデルシミュレーションの見直しである。Boidモデルの速度ベクトルを更新する際、他のエージェント全てとの距離を計算し、範囲内にいるエージェントとの相互作用を計算しているため、計算量は $O(N^2)$ である。本研究では、最大1000体のエージェントを使用した。本武氏らによる研究[12]でも触れられているように、エージェント数が増えると計算コストが大幅に増えてしまう。今後、さらに多くのエージェントを使ったシミュレーションを行うためには、計算量を抑えるためにアルゴリズムの改善が必要である。さらに、本研究で実装したシミュレーションは、エージェントが領域外に出ることができな

い閉鎖的な空間で行っていたが、上へ出て行ったエージェントが下から出てくるといった周期的なものに変更することが可能である。そうすることによって、閉鎖的な空間で見られなかった運動が見られるようになり、先述した新たな運動状態の観測ができる可能性がある。

謝辞 本研究はJSPS 科研費 JP18K11338 の助成を受けたものです。

参考文献

- [1] 二上将直 他: 工学院大学研究報告, 第 105 号, pp.29-36 (2008).
- [2] D. Martens, B. Baesens, T. Fawcett: Machine Learning, Vol. 82, pp.1-42 (2011).
- [3] 水口毅 他: 可視化情報 Vol.37 No.144, pp.14-19 (2017).
- [4] 阪上雅昭, 寺山慧, 炭谷竜太: 数理解析研究所講究録, No.1900, pp.110-117 (2014).
- [5] 古場健士, 藤村誠, 山口恭弘: 情報処理学会第 77 回全国大会講演論文集, No.1, pp.707-708 (2015).
- [6] Hiro-Sato Niwa, Journal of Theoretical Biology, Vol.181, pp.47-63 (1996).
- [7] Craig W. Reynolds: Computer Graphics, Vol.21, No.4, pp.25-34 (1987).
- [8] K Fujii, T Kawasaki, Y Inaba, Y Kawahara: PLoS Computational Biology 14(11): e1006545 (2018).
- [9] 坂井澄子: 生物物理, 13 巻, 2 号, pp.82-90 (1973).
- [10] I. D. Couzin *et al.*: Journal of Theoretical Biology, Vol.218, pp.1-11 (2002).
- [11] Tamás Vicsek *et al.*: Physical Review Letters, Vol.75, 1226 (1995).
- [12] Y. Mototake, T. Ikegami: SWARM 2015, pp.446-450 (2015).
- [13] T. Ikegami *et al.*: Philosophical Transaction of the Royal Society A, Vol.375, pp.1-15 (2017).