

# 座標分布及び遷移確率に基づいた移動軌跡の比較方法の提案—オンラインゲームへの応用

織田 淳一\* Ruck Thawonmas\* Kuan-Ta Chen\*\*

\*立命館大学大学院 理工学研究科

\*\*中央研究院 資訊科學研究所

**抄録:** 位置測位技術の普及と高度化により一度に多量の移動軌跡データを取得することが可能となっている。それに伴い、移動軌跡データから有用な情報や知識の取得を行うデータマイニングの技術が重要視されており、移動軌跡の分類や行動の意味づけを行う研究が従来から行われている。本論文では、座標分布と遷移確率に基づいた比較手法に着目する。前者では各軌跡の特徴に合わせた状態群の定義方法を、後者では異なる状態群同士の比較方法を提案する。Quake II と Angel Love Online の軌跡データを用いた実験により、提案手法の優位性を確認する。

**キーワード:** 移動軌跡, 移動分析, 座標分布, 遷移確率, オンラインゲーム

## A Comparison Method for User Trajectories Based on Coordinate Distribution and Transition Probability – Applications to Online Games

Junichi Oda\* Ruck Thawonmas\* Kuan-Ta Chen\*\*

\*Graduate School of Science and Engineering, Ritsumeikan University

\*\*Institute of Information Science, Academia Sinica

**Abstract:** Nowadays, a large amount of movement trajectory data can be obtained due to the dissemination and advancement of the positioning technology. As a result of this, data mining technology has become a key to extract important information or knowledge from such data, and related research studies have been conducted. In this paper, we focus on methods for comparing user trajectories based on coordinate distribution and transition probability. In the former, we propose a method for defining a set of states that individually suit the characteristic of each trajectory and propose a method for comparing trajectories having different state sets in the latter. The effectiveness of our approach is validated with trajectory data from Quake II and Angle Love Online.

**Keywords:** Movement Trajectory, Movement Analysis, Coordinate Distribution, Transition Probability, Online Game

### 1. はじめに

無線ネットワーク技術や GPS などの位置測位技術の普及などにより、今日では人物や自動車などの移動体経路(移動軌跡)を得ることが容易となっている。それに伴い、多数の移動軌跡から有用な情報や知識の取得を行うデータマイニングの技術が重要視されており、移動軌跡を視覚化する手法、移動軌跡の特徴や関係性を抽出する手法や、移動軌跡を近似する手法の研究などが行われている。最近、MMOG(大規模多人数オンラインゲーム)または FPS(一人称視点シューティングゲーム)のプレイヤーの移動軌跡を対象とした研究[1-4]が報告されており、これらの研究成果を活用することでゲームの設計やプレイヤーに対するサービスの向上が期待できる。

移動体の移動軌跡の比較に焦点を当てた研究においては、得られた座標などの情報から類似度測定、または距離測定を行って移動軌跡の比較を行い、移動軌跡の分類や行動の意味付けによって移動体の現状や、これから先の行動を予測することが可能となる。移動軌跡解析の過程で移動軌跡を状態遷移として表現し比較する手法がいくつか提案されている。その多くは最初に対象となる空間を分割して出来上がった各エリアを状態として定義し、その状態群を最後まで変更させることはない。

しかし手法によっては、状態遷移に表現したときに過剰に軌跡が近似される、または状態内での細かい動きが表現されない場合がある。この問題は空間の分割回数を増やすことで防ぐこともできるが、処理コストが増大し性能を落とす可能性がある。そこで、軌跡の細かい特徴まで表現できる空間の分割を軌跡ごとに行い、

各軌跡によってできた異なる状態群の比較手法が必要であると考えられる。

本論文では移動軌跡に対して軌跡の座標データ(以後、データ)の分布に基づく分類と、遷移確率に基づく分類の二段階に分けて移動軌跡を比較する手法を提案する。データの分布に基づく分類においては、4分木に基づく動的マップ分割を採用し、4分木の構造を示すビット列同士のハミング距離により軌跡間の距離を求める。分類により出来上がったクラスタ内における軌跡に対しては、動的マップ分割により得られた軌跡ごとの分割マップをマルコフ連鎖モデルにおける状態群と見なしモデル化を行い、軌跡間の距離をとり再分類(遷移確率に基づく分類)する。軌跡ごとに動的分割を用いることにより、得られる状態群は軌跡の特徴をより細かく抽出するとともに、過剰な近似が施されることがない。また二段階の分類により、一度のクラスタリングで必要とする要素の数を減らし、軌跡同士が類似する、または類似しない理由を明確にできる。

4章で、Quake II というオンラインゲーム上で取得した軌跡の分類実験の結果を示す。実験データは人が実際に操作して得た軌跡と、Bot と呼ばれる自動制御プログラムにより操作されたキャラクタから得た軌跡が混在している。これらを、本手法を用いて分類する実験を行った。5章では、本手法の応用の1つとして、ALO というオンラインゲーム上で取得した軌跡から、プレイヤーの行動推測をする実験の結果を示す。本手法により分類されたプレイヤーの主な行動目的を、マップ情報と照らし合わせて推測した。

## 2. 関連研究

軌跡の比較手法として、[4]では、[1]の手法を改良する形で、MMOG における全プレイヤーの移動軌跡をもとに動的マップ分割とよばれる4分木の作成を行い、データの密度からランドマークとなるエリアを決定する。さらに、各プレイヤーの移動軌跡をランドマーク間の遷移とみなして遷移確率を求め、プレイヤー同士の遷移確率を比較することによりプレイヤーを分類する手法が提案されている。同手法は、マップの等分割を行う[1]の手法と比較して、属性が少ないことによって処理速度は速いが、元の軌跡からランドマーク間の遷移とする近似操作が過剰に働き、逸脱した行動をしたプレイヤーの移動軌跡が存在した場合に精度が失われる。

移動体追跡の一つの手法として、[5]では移動統計量のモデルとしてマルコフ連鎖モデルを採用しており、データキューブに似た移動ヒストグラムの論理表現を与えて多次元分析を可能としている。各状態には一意

の数値がZ-orderingに基づいて割り当てられ、単位時間の粒度の調整に、大きい単位時間で表現するロールアップ、小さい単位時間で表現するドリルダウンという操作を対応させている。しかし大量の数の移動オブジェクトを同時に追跡する場合、ヒストグラム構築時間がボトルネックとなる可能性がある。

同様にマルコフ連鎖モデルを用いたものとして[6, 7]がある。[6]では実際に店舗のレイアウトに合わせて手動で状態を定義し移動軌跡から二段階に分けて逸脱行動人物データ検出を行う手法を提案している。マルコフ連鎖モデルを用いて人物移動データをモデリングし、各人物移動データ間の確率的距離と、k-means法を用いたクラスタリング後の尤度を用いて、逸脱行動人物の検出を行っている。またクラスタ数については[8]で紹介されている評価指標を用いて最適なクラスタ数を求めている。クラスタリング前に、多次元尺度構成法による低次元空間への射影を行っており、これにより、各次元の意味が不明瞭となり各クラスタの軌跡の特徴が理解しにくくなる。[7]では軌跡の分布から空間をk-means法でクラスタリングし、分割された空間上で移動軌跡を状態列にモデル化する。状態列のDPマッチング法による比較で類似度を計算し、行動パターンの抽出を行っている。しかし、DPマッチング法は計算コストが大きくなり時間がかかるという問題点を持つ。

[9]では時系列データをPAA(Piecewise Aggregate Approximation)により時間軸で細分化したのち、正規化したデータを正規分布の面積が等しくなるように数値を文字により量子化して表現するSAX(Symbolic Aggregate Approximation),という手法を提案している。各アルファベット同士の差は正規分布の面積を分割した点同士の距離を使用しており、PAAの時間軸の粒度と文字の個数を決めることで、各アルファベット間の距離を定義できる。また、このアルファベット間の距離は常に実距離よりも小さくなることが証明されている。ただし、最適な粒度、文字数を求めることと、多次元への応用が困難となっている。

3次元空間に基づいたデータに関しては[10]ではMBB(Minimal Bounding Box), [11][12]ではMBR(Minimal Bounding Rectangles)と呼ばれる、属性ごとに近似する上限と下限を定め、その制限に基づいて軌跡を近似する手法を提案すると同時に、独自の近似手法に則した比較方法も提案している。これらの比較手法は時間ごとの差分を積分したものであるため、ノイズなどの影響で局所的に大きな差が出てしまうとその差分を保持したまま積分してしまう恐れがある。また、属性が増えるに従い各属性の制限が増え、効果的な近似が期待できないと推測される。

クラスタリングの目的として軌跡からその移動体が人ではなく、人工的にプログラムされたものによって制御されているキャラクタ(bot)かどうかを判別する研究がある[3]. この研究では, **Quake II** という FPS を対象として実際に bot により作られた移動軌跡とヒトにより作られた移動軌跡を比較している. クラスタリングの手法として **SVM(Support Vector Machine)** を使用しており, 二つのクラスに分類することに優れている SVM の特徴を効果的に用いている. しかし, SVM を多クラスの分類に用いる場合は工夫が必要となる.

### 3. 提案手法

状態遷移を用いた[1, 4-7]では最初に定義した状態群を用いてすべての軌跡を処理しているが, 軌跡によってどのエリアが重要であるか, 各エリアが占めるデータ数といった点は異なってくる. 複数の軌跡に対して 1 種類の状態群を割り当てるのではなく, 軌跡ごとにデータの特徴に合った状態群を作成して割り当てることで詳細な分析が可能と推測できる. しかし, これによりモデル化された軌跡間の比較が困難になる. そこで状態群の異なるモデル間の比較を後述する空間的ロールアップ処理に基づいて行う.

なお, ここで示す軌跡データとは(1)のような座標データの時系列データを指す. また, 全軌跡数を  $N$  とする.

$(x_i(1), y_i(1)), \dots, (x_i(j), y_i(j)), \dots, (x_i(T_i), y_i(T_i))$  (1)  
 ここで,  $T_i$  は  $i$  ( $i = 1, \dots, N$ ) 番目の軌跡のデータ長,  $(x_i(j), y_i(j))$  は軌跡  $i$  の  $j$  番目の座標を示す.  $x, y$  の値は 0 以上で, 対象となるエリアのサイズ以下の値をとる.

#### 3.1. 4 分木に基づいた動的マップ分割

各移動軌跡におけるデータの分布から, 軌跡ごとの分割マップを作成する. 対象のエリア全体を初期状態(分割回数 0)としてまず 4 分割する. 出来上がった各エリア内で(2)に従ってデータの密度  $d$  を求め, 密度が閾値以下となるまで各エリアについて 4 分割を繰り返す. また, エリアの分割回数を分割レベル  $D$  で表現する(図 1). 移動していない, 移動の少ない軌跡の場合, エリアが分割により小さくなくても密度が高くなり, 閾値を下回ることがなく分割が止まらない. そこで, 最大分割回数  $D_{max}$  を設定して制限を与える.

$$d = \frac{1}{T_i} \sum_{j=1}^{T_i} \left\{ \begin{array}{l} 1, \text{ if } \left( \begin{array}{l} X_{high} \leq x_i(j) \leq X_{low} \\ \text{and} \\ Y_{high} \leq y_i(j) \leq Y_{low} \end{array} \right) \\ 0, \text{ otherwise} \end{array} \right\} \quad (2)$$

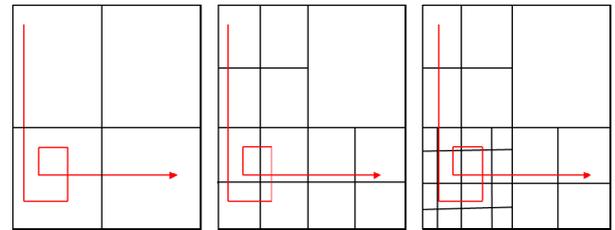


図 1. 軌跡に合わせたマップの分割(左から  $D = 1, D = 2, D = 3$  を示す)

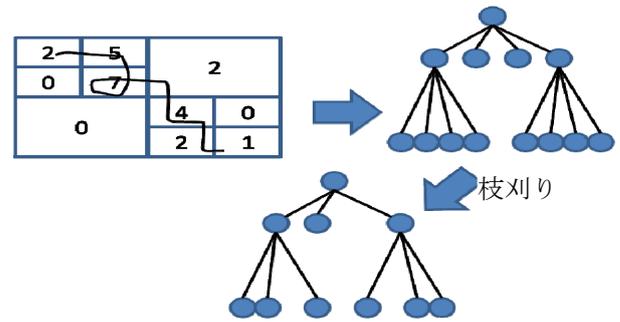


図 2. 動的マップ分割と 4 分木の対応, および枝刈り(エリア内の数値はエリア内に属するデータ数)

ここで,  $X_{high}, X_{low}, Y_{high}, Y_{low}$  はそれぞれ対象としているエリアの  $x, y$  軸の上限と下限を示す.

上記の動的マップ分割を 4 分木により実装する. 分割する前のマップの状態を *root* (ノードの番号 0) とみなし, マップの 4 分割を木の展開に対応させる. ノードによってはデータを保持しないものが存在する. そこで, データをもたないノードに対して枝刈りを行う(図 2).

以下に, 動的マップ分割(4 分木作成)のアルゴリズムを **Algorithm 1** に示す. このアルゴリズムは再帰的な関数 `map_division()` を定義している. この関数は 9 個の引数 `trajectory, T_i, tree, node_num, D, X_low, X_high, Y_low, Y_high` をとる. `trajectory` は対象の軌跡  $i$ ,  $T_i$  は対象の軌跡  $i$  の長さ, `tree` はマップ分割によりできたエリアの情報 (`node_num, D, X_low, X_high, Y_low, Y_high`, 葉ノード化を示すビット値)を格納するオブジェクト, `node_num` はノードの番号,  $D$  は分割レベル,  $X_{low}, X_{high}, Y_{low}, Y_{high}$  はそれぞれ, 対象エリアの  $x$  座標の最小値, 最大値,  $y$  座標の最小値, 最大値を示す. 動的マップ分割処理が呼び出された時, `tree` にはまだ分割されていないエリア(*root*)の情報が格納され, `node_num = 0, D = 0, X_low = 0, X_high =` エリア全体の  $x$  軸の最大値, `Y_low = 0, Y_high =` エリア全体の  $y$  軸の最大値が, 初期値として与えられる. ただし, 座標(0, 0)はエリアの左上で,  $(X_{high}, Y_{high})$  は右下となる.

**Algorithm 1** map\_division(*trajectory*,  $T_b$ , *tree*, *node\_num*,  $D$ ,  $X_{low}$ ,  $X_{high}$ ,  $Y_{low}$ ,  $Y_{high}$ )

- 1 **If**  $D = D_{max}$  //最大分割回数と同じであればこれ以上分割しない
- 2     *tree* に対象エリアの情報を格納
- 3     **return** *tree*
- 4 (2)より  $d$  を求める
- 5 **If**  $d >$  閾値 //密度が閾値以上のときマップ分割
- 6     *tree* に対象エリアの情報を格納
- 7      $X_{low}$ ,  $X_{high}$ ,  $Y_{low}$ ,  $Y_{high}$  から対象エリアの  $x$ ,  $y$  軸の中心の  $X_{middle}$  と  $Y_{middle}$  を求める
- 8     //1st child
- 9      $node\_num = 4node\_num + 1$  //子ノード番号算出
- 10    *tree*  $\leftarrow$  map\_division(*trajectory*,  $T_b$ , *tree*,  $node\_num$ ,  $D+1$ ,  $X_{low}$ ,  $X_{middle}$ ,  $Y_{low}$ ,  $Y_{middle}$ ) //子ノードのエリア分割を行う再帰処理
- 11    //2nd child
- 12     $node\_num = node\_num + 1$
- 13    *tree*  $\leftarrow$  map\_division(*trajectory*,  $T_b$ , *tree*,  $node\_num$ ,  $D+1$ ,  $X_{middle}$ ,  $X_{high}$ ,  $Y_{low}$ ,  $Y_{middle}$ )
- 14    //3rd child
- 15     $node\_num = node\_num + 1$
- 16    *tree*  $\leftarrow$  map\_division(*trajectory*,  $T_b$ , *tree*,  $node\_num$ ,  $D+1$ ,  $X_{low}$ ,  $X_{middle}$ ,  $Y_{middle}$ ,  $Y_{high}$ )
- 17    //4th child
- 18     $node\_num = node\_num + 1$
- 19    *tree*  $\leftarrow$  map\_division(*trajectory*,  $T_b$ , *tree*,  $node\_num$ ,  $D+1$ ,  $X_{middle}$ ,  $X_{high}$ ,  $Y_{middle}$ ,  $Y_{high}$ )
- 20 **else**
- 21    *tree* に対象エリアの情報を格納
- 22 **return** *tree*

アルゴリズムの概要を説明する. この処理ではエリア内でのデータの密度を調べ, 値がこちらで指定した閾値以上であれば, そのエリアを4分割する. 分割によりできた各エリアについても同じ処理を繰り返す. 実装には4分木を用いる. 4分木の構築を再帰呼び出しで実現する.

まず, 対象の分割レベル  $D$  が, こちらでユーザーの設定した最大分割回数  $D_{max}$  と等しい時, 分割を止め, 2行目で *tree* に対象エリアの情報の格納だけ行って *tree* を返す. 分割レベル  $D$  が  $D_{max}$  よりも低い場合, 5行目で密度  $d$  の比較を行う. 4行目で得られた密度  $d$  が, ユーザーの設定した閾値よりも大きい場合, 6-19行目までの処理を行う. 閾値以下の場合, 21, 22行目に示すように, 分割を行わずに *tree* に対象エリアの情報の格納だけ行って *tree* を返す処理を行う.

密度  $d$  が, ユーザーの設定した閾値よりも大きい場合, まず, 6行目で対象となるエリアの情報を *tree* へ格納し, 7行目で対象エリアの  $x$ ,  $y$  軸の中心,  $X_{middle}$ ,  $Y_{middle}$  を求める. 次に, 9行目で分割後の1番目のエリア(1番目の子ノード)の番号を求め, 10行目で再帰呼び出しを行い, このエリアに関するマップ分割処理を行う. このとき,  $D$  を1増やしておき, エリアの  $x$ ,  $y$  軸の最大値をそれぞれ,  $X_{middle}$ ,  $Y_{middle}$  にしておく. 11-19行目の処理も, 各分割後のエリアに対して同様の処理を行っている.

### 3.2. 4分木同士の比較と距離の定義

4分木の全ノードにはZ-orderingに基づいて一意に番号を割り当てる(図3). 数字の割り当て方法はすべての4分木に対して同じ方法をとるので, 二つの4分木において割り振られた番号が同じノードは, 同じエリアを示す. 4分木同士の比較はrootとなるノードから末端の全ノードに対して各ノードが存在するかどうかを示すビット列を用いて行う.  $n$ 番目のビットが  $n$ 番目のノードの有無を示す. 1の場合は存在し, 0の場合は存在しない(図4).

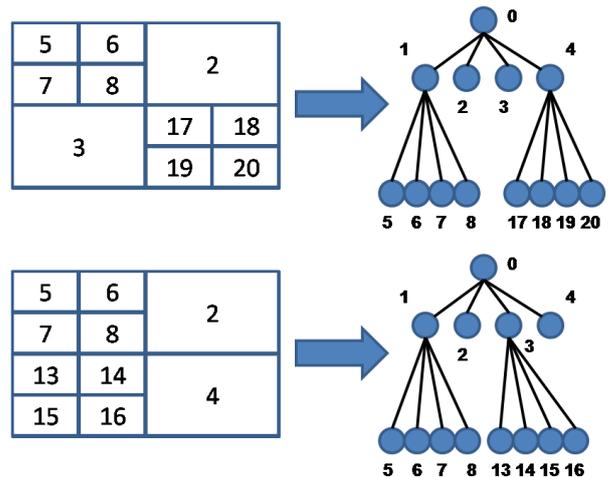


図3. Z-orderingに基づくノード番号の割り当て(エリア内の数値はノード番号に対応)

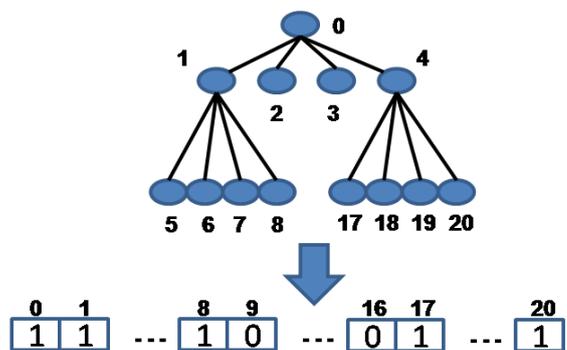


図4. 4分木とビット列の対応

比較の対象となる二つのビット列のハミング距離をとることで、4分木の構造の違い  $Dist$  が得られる(4)。また、 $D_{max}$  から必要となるビット列の長さ  $B$  が得られる(5)。ここでは  $Dist_{ij}$  の値を軌跡  $i$  と  $j$  の距離として定め、 $(N-1)(N-1)$ の距離行列を作成する(図 5)。

$$Dist_{ij} = \sum_{n=1}^B \{bit_i(n) \oplus bit_j(n)\} \quad (4)$$

$$B = \frac{4^{D_{max}+1} - 1}{3} \quad (5)$$

ここで、 $bits_i(n)$  は軌跡  $i$  のビット列の  $n$  番目のビット値を示す。

### 3.3. クラスタリングと逸脱した軌跡群の除外

作成した距離行列をもとに Ward 法[13]によるクラスタリングを行う。ここでの分類は4分木の構造の違い、つまり、軌跡データの分布に基づいた比較の結果を用いている。したがって、移動の分布が似ている軌跡が同じクラスタに分類される。

また、最適なクラスタ数を求めるのは一般的に困難である。そこで、本手法ではクラスタ数  $K$  の評価指標を用いることで、クラスタ数の妥当性を示すことにする。

$$CH(K) = \frac{traceH/(K-1)}{traceU/(N-K)} \quad (6)$$

$$traceH = \sum_{k=1}^K N_k Dist_{C_k C} \quad (7)$$

$$traceU = \sum_{k=1}^K \sum_{i=1}^{N_k} Dist_{C_k k_i} \quad (8)$$

$$C = \min_n \frac{1}{N-1} \sum_{i=1, i \neq n}^N Dist_{ni} \quad (n=1, \dots, N) \quad (9)$$

ここで、 $N_k$ はクラスタ  $k$  の要素数、 $k_i$ はクラスタ  $k$  の  $i$  番目の軌跡(ビット列)、 $Dist_{C_k k_i}$  はクラスタ  $k$  のセントロイドとクラスタ  $k$  の  $i$  番目のデータとの距離、 $Dist_{C_k C}$  はクラスタ  $k$  のセントロイド  $C_k$  と、全体データのセントロイド  $C$  の距離を示す。 $K \leq 20$  の範囲で、クラスタ内の最大データ数が全軌跡数の10%以下となるまで繰り返し、 $CH(K)$ の極大値の最小値をとる  $K$  を最適なクラスタ数として採用する[8]。

最後に、得られたクラスタごとに要素数の確認を行う。要素数が全移動軌跡数の10%以下であるクラスタを、逸脱した軌跡群とみなして除外し、以降の操作を

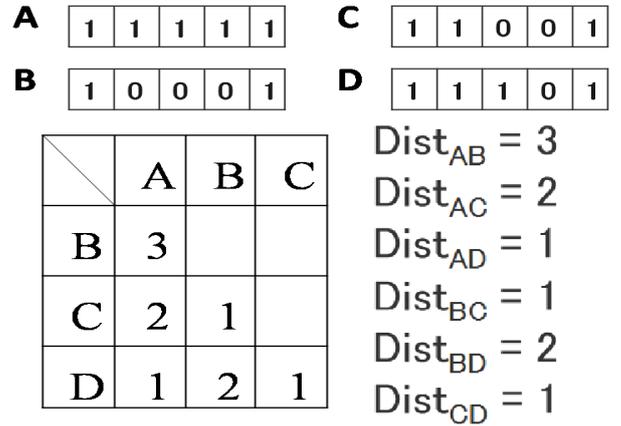


図 5.  $N=4, B=5$  としたときの距離行列作成例

行わないようにする。

### 3.4. モデル間の比較

ここまでの操作により、移動の分布の違いによる分類ができたが、これだけでは移動方向が全く逆の軌跡同士が同じクラスタに含まれてしまう。そこで、各クラスタ内で軌跡の遷移に着目した比較を行うことでこれらを分類する。

ここからの操作はクラスタごとに行う。対象の軌跡に関しては動的マップ分割により得られた分割マップを状態群とみなして、マルコフ連鎖モデルに基づいたモデル化(10)を行ってから、状態間の遷移確率を求め

$$(x_i(0), y_i(0)), \dots, (x_i(j), y_i(j)), \dots, (x_i(T_i), y_i(T_i)) \Rightarrow l_i(0), \dots, l_i(j), \dots, l_i(T_i) \quad (10)$$

ここで、 $l_i(j)$  は軌跡  $i$  の  $j$  番目の状態(エリア)を示す。

動的マップ分割により得られた分割マップは軌跡ごとに結果が異なる。そこで比較する二つの軌跡において、4分木の構造を示すビット列の論理積により共通する4分木の構造を抽出する(11)(図 6)。

$$bits_{com} = bits_i \wedge bits_j \quad (11)$$

ここで、 $bits_{com}$  は二つの軌跡から抽出した、共通する4分木の構造を表すビット列を示す。

共通するノードの抽出により、分割レベルが下がるエリアが生じるので、分割レベルが下がったエリアのマルコフ連鎖モデルのパラメータを、状態群の変化に合わせて変更する。この操作は空間的ロールアップ処理に対応する[5]。ロールアップ処理は以下のように行う。共通するノードのパラメータは、分割レベルが下がる前のパラメータを操作することで得られる(12~14)。

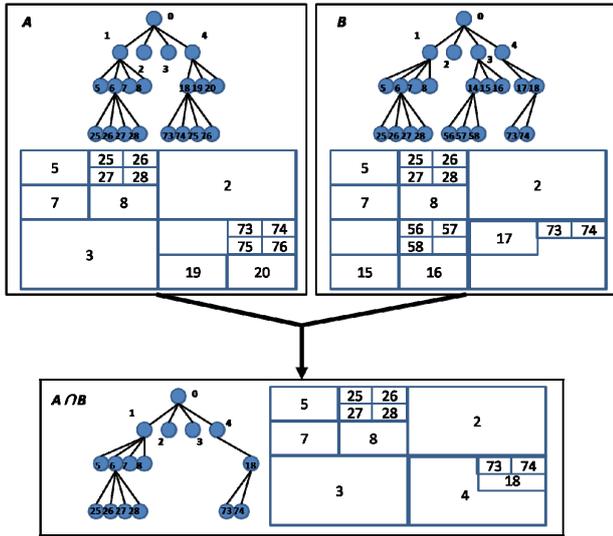


図 6. 共通するノードの抽出(エリア内の番号はノードの番号と対応する)

遷移前と遷移後の状態が  $l'$  の場合

$$a_{l'l'} = \frac{1}{A(l')} \sum_{i=1}^{A(l')} \sum_{j=1}^{A(l')} a_{l'(i)l'(j)} \quad (12)$$

遷移前の状態が  $l'$  の場合 ( $m$ : 任意の遷移後の状態)

$$a_{l'm} = \frac{1}{A(l')} \sum_{i=1}^{A(l')} a_{l'(i)m} \quad (13)$$

遷移後の状態が  $l'$  の場合 ( $m$ : 任意の遷移前の状態)

$$a_{ml'} = \sum_{i=1}^{A(l')} a_{ml'(i)} \quad (14)$$

ここで、 $l'$  は分割レベルの下がった後の統合されたエリアの番号を意味し、 $l'(k)$  は共通するノードの抽出により削除されたノードで、統合後のエリア  $l'$  のパラメータに関与するエリアを示す(図 6. において、 $l'=3$  とすると、 $l'(k)$  は 15, 16, 56, 57, 58 のエリア番号を示す). 以後、これらのエリアの数を  $A(l')$  で表し、図 6 の例では  $A(3)=5$ . ロールアップ処理により 4 分木において葉ノードでないノードもエリアとして扱われる場合が生じる(図 6 ではエリア 4, 18 がこれに該当する).

次に、ロールアップ処理により変更された二つの軌跡の各遷移確率の差をとる. ここで、各エリアの分割レベルはエリアごとに異なり、エリアによってはロールアップ処理により分割レベルが低くなり、詳細な情報が失われた可能性がある. エリア  $l$  の分割レベルに合わせて重み  $w_l$  を (15) のように与え、軌跡  $i$  と軌跡  $j$  の遷移確率の差の重み付き平均を軌跡間の距離として定める(16).

$$w_l = \frac{D_{\max} - D_l + 1}{D_{\max}} \quad (15)$$

$$Dist_{ij} = \frac{1}{h_{num}} \sum_{l=1}^{h_{num}} \sum_{m=1}^{h_{num}} w_l w_m |a_{lm}(i) - a_{lm}(j)| \quad (16)$$

ここで  $D_l$  はエリア  $l$  の分割レベル、 $D_{\max}$  は最大分割数、 $Dist_{ij}$  は軌跡  $i$  と軌跡  $j$  の距離、 $a_{lm}$  はエリア  $l$  からエリア  $m$  への遷移確率、 $h_{num}$  はロールアップ後の二つの軌跡の共通する状態群の数を示す. 軌跡によってはロールアップ処理により統合されたエリアが、一度も分割されていない状態(分割回数 0 の初期状態)になる. この場合は、遷移確率の差が 0 となり好ましくないので、特別に差を(16)の取りうる値の最大値 2 を距離として与える.

最後に、同じクラスタ内のすべての軌跡の組に対して距離をとり、この距離を用いて 3.3 で示した方法でクラスタリングを行う. 3.3 の時点で、移動の分布が似ている軌跡が分けられており、ここでのクラスタリングでエリア間の遷移が似ているものが同じクラスタに分けられるので、結果として、移動の分布とエリア間の遷移が似ているものが同じクラスタに収められる.

#### 4. Quake II による BOT 判別実験

Quake II とは、*id Software* により開発されたネットワーク対戦型ビデオゲームであり、FPS(First-Person Shooting Game)のジャンルに属する. Quake II により生成された移動ログを使用する(図 7). ゲーム空間内において、プレイヤーは特定のキャラクタを操作し、所持している武器を使用して敵を撃つ動作を行う. 多くのプレイヤーが一度にゲームに参加することができ、チームとして他のプレイヤーと協力して参加することもできるが、「death-match games」と呼ばれるものにおいては、参加している他のプレイヤーをできるだけ倒すという目的があり、Quake II ではこの「death-match games」のルールに基づいてプレイする人が多く、人気が高い.

また、Quake II 内では移動やアクションの履歴を容易に残すことができ、この機能により多くのプレイヤーがネット上で自分の技術などを公開したり議論したりすることができる. 人工知能の分野において、上記の



図 7. Quake II プレイ画像

ように公開されたヒトの移動軌跡を学習データとしてプログラムされたシステムによってキャラクタを操作し、ヒトらしい動きをさせる手法が提案されている[14].

移動ログにはヒトが実際にプレイしたことのできたもの(*player*)と、*bot* とよばれるプログラムにより自動で動くキャラクタ(*Crbot*, *Eraser*, *Ice*)により得られたログがある(図 8). 本研究では[3]で使用されたデータと同じ *player* と *bot* のデータを使用して実験を行う。*player* のデータはネット上で公開されているログのうち「*The Edge*」と名付けられたマップ上の移動軌跡を取得した。このマップ上でのプレイヤーの目的はできるだけ多くの他プレイヤーを倒すという「*death-match games*」に基づく動きを示している。*bot* のデータに関しては、まず実際に三種類の *bot* を用意し、「*The Edge*」マップ内に 2~6 の *bot* を三種類からランダムに選びだして約 20 時間動作させる。これを数十回行って得られたデータを *bot* のログデータとしている。

「*The Edge*」マップは中央部分にホール上の広い空間を有している。この空間は *player* にとっては攻撃されやすい場所と判断され、軌跡の分布の密度が低いのにに対して、*bot* にとっては移動しやすい場所と判断されるため、軌跡の分布の密度が高い。3 種類の *bot* の軌跡は一定のアルゴリズムに従って行動したものであるため、同じ経路上を繰り返し移動する傾向があるのに対し、*player* はエリア全体をまんべんなく移動する傾向がある。それぞれの *bot* の特徴として、*Crbot* はマップの左下側への移動が少なく、移動パターンが複数存在する。*Eraser* は移動パターンが少なく、狭い通路では通路の中心を移動する特徴がある。*Ice* は待ち伏せして敵を攻撃する機会を待つ傾向がある。軌跡の比較により、この 4 種類の軌跡の判別ができるかどうかを実験する。

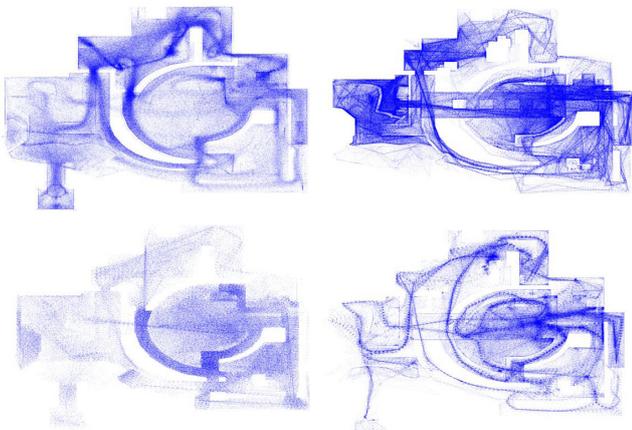


図 8. *player* と *bot* の軌跡(左上:*player*, 右上:*Crbot*, 右下:*Eraser*, 左下:*Ice*)

#### 4.1. 比較対象とする別の手法

本手法を評価するために、同じデータを用いて他の手法との比較を行う。比較する手法は[1]で用いられている、全データを用いた動的マップ分割手法であり、マップ上のデータ分布から共通の分割マップを作成する。これにより得られた分割マップを状態群とし、その次は提案手法と同じく、各軌跡に対してマルコフ連鎖モデルに基づいたモデル化を行い、得られた遷移確率の比較を行う(16)。 (16)で得られた値に基づいて、Ward 法によるクラスタリングを行う。

また、提案手法では座標分布に基づく分類と、遷移確率に基づく分類の 2 回に分けて軌跡のクラスタリングを行っている。二度にわたってクラスタリングを行うことを評価するために、各比較手法を単独で行った結果と比較する。各手法を用いるにあたり、こちらで設定した変数とログデータの詳細について記述する。この値は比較するすべての手法において共通の値である。

エリアの最大分割回数：5

分割を判断する密度の閾値：5%

マップのサイズ(2240, 1050)

軌跡データ中に欠損値やエリア外と思われる数値を示すデータポイントは除外。

データ長は *player* と *bot* で大きく異なる。ヒトは約 2 時間に対して *bot* のデータは約 20 時間。本手法ではマップ内の分布をとるので、実験でデータ長は変更しない。

#### 4.2. 評価指標と結果

出来上がったクラスタ内のデータについて、クラスタ  $k$  内のプレイヤーの種類のばらつきをはかるために、*player*, *Crbot*, *Eraser*, *Ice* の種類に関する平均情報量 (エントロピー)  $H(k)$  をとり、全クラスタのエントロピーの和  $H$  を求める。これにより得られた  $H$  を評価指標として用いる。

表 1 より、座標分布に基づく分類と、遷移確率に基づく分類の両方を施した手法のエントロピーの和は 0 となり、他の手法は 1.5 を超える値を示した。これはほかの手法では出来上がったクラスタ内に *bot* と *player* が

表 1 各手法の平均情報量

クラスタリング手法	$H$
全軌跡による動的マップ分割	1.58
座標分布に基づく分類	1.56
遷移確率に基づく分類	1.58
座標分布に基づく分類と遷移確率に基づく分類 (提案手法)	0

混在するクラスタが出来上がったが、本手法では正確に分割することができたことを示している。また、全データによる動的マップ分割の結果では、*bot* と *player* の判別はできたが、*bot* の種類を分類することはできなかった。遷移確率に基づく分類のみの場合は、作られたクラスタ数は 2 個であったが、一つの軌跡を除いた他のすべての軌跡が一つのクラスタに含まれるという結果となり、全手法の中でも結果を得るまでの時間が最も長かった。

表2は提案手法の詳細結果を示すもので、1st は座標分布に基づいた分類の結果を示し、2nd はクラスタ内での遷移確率に基づく分類の結果であり、*C*, *E*, *I* はそれぞれ *Crbot*, *Eraser*, *Ice* を意味する。表2から、座標分布に基づいた分類だけでは *player* と *Crbot*, *player* と *Ice* の軌跡が同じクラスタとして分類されているが遷移確率に基づく分類により正確に分類ができるようになった。また、1stの4番目のクラスタは遷移確率に基づく分類が行われていない。これは、このクラスタに含まれている軌跡の数が全軌跡数の 10%を下回っていたため、逸脱した軌跡と判別されたことを示す(図9)。

### 5. ALO による実験

本手法の応用の例として、ALO のログを用いてプレイヤーの行動分析を行う。クラスタリングによる結果とマップの情報とを照らし合わせた行動推測の方法を示す。ALO (ANGEL LOVE ONLINE) とは、台湾のゲーム会社 USERJOY TECHNOLOGY (宇峻科技) が開発したオンラインゲームで、MMOG に分類される 2D 方式の RPG である。プレイヤーは ALO 内で、アバターと呼ばれる自分の分身を示すキャラクタを操作して、ALO 内で用意

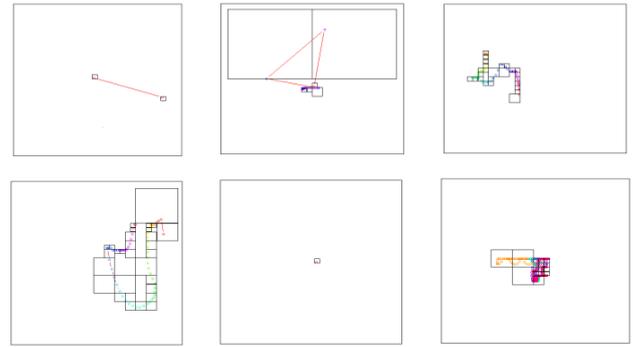


図 9. 逸脱した軌跡と判別された軌跡の例

されたクエストという目標を達成するために行動したり、ALO を通じて他のプレイヤーとコミュニケーションをとることができる。このゲーム内の桜貝村とよばれるマップで、約 70 時間、1 秒間隔で複数のプレイヤーの軌跡を記録したログを用いて実験を行った。プレイヤーの総数は 394、各プレイヤーのデータ長はプレイヤーごとに異なり、また、ここでのプレイヤーの行動目的は統一されていない。桜貝村マップの情報とクラスタリング結果とを照らし合わせて、行動目的を分析する。

桜貝村マップの特徴は、マップの左側、右上、右下にはプレイヤーを攻撃する敵が点在しており、右中央付近は町となっていてプレイヤーをサポートするさまざまな施設がもうけられている(図 10)。また、図 10 の青い部分は海を表しており、プレイヤーが通ることはできない。

座標分布に基づいた分類では 2 クラス、遷移確率に基づく分類で出来上がったクラスタ数は 21 となった。ここで、各クラスタの特徴を得るために、クラスタごとに全軌跡による動的マップ分割を採用し、軌跡の分布に合わせて分割マップを作成する。その分割マップを状態群とみなして状態間の遷移確率を求め、値の高いものから順に 10 個まで矢印で示した図を作成した(図 11, 12)。図 11, 12 はそれぞれ座標分布に基づいた分類でクラスタ 1, 2 に分類されたクラスタのうちの 1 つを図示している。ここで、ALO のログはデータ間隔

表 2 座標分布に基づく分類と遷移確率に基づく分類

	軌跡の種類別の個数					$H(k)$	
	1st	2nd	<i>player</i>	<i>C</i>	<i>E</i>	<i>I</i>	1st
1	1	27	0	0	0	0	0
	2	9	0	0	0		0
2	1	9	0	0	0	0	0
	2	31	0	0	0		0
3	1	9	0	0	0	0.83	0
	2	0	25	0	0		0
4	1	15	0	0	0	0	0
5	1	5	0	0	0	0.72	0
	2	0	0	0	20		0
6	1	0	0	13	0	0	0
	2	0	0	8	0		0



図 10. ログをとった ALO の桜貝村マップ

が 1 秒と非常に短く、遷移の多くが同じエリアへの遷移と判断されてしまうため、クラスタごとの遷移の表示では、同じエリアへの遷移は省いている。

マップの情報と図 11 の分割マップを照らし合わせて、このクラスタに分類されたプレイヤーの行動推測を行う。分割マップの結果からこのクラスタ内のプレイヤーはマップの左側での行動が少ないことがわかる。また、遷移がマップの下側に多いのが特徴的で、村と敵が分布しているマップの下側との遷移が目立つ。これより、このクラスタのプレイヤーはマップの下側で敵と戦闘し、村内の施設でサポートを受けるというのを繰り返したと推測できる。

マップの情報と図 12 の分割マップを照らし合わせて、このクラスタに分類されたプレイヤーの行動推測を行う。削除されている分割マップがないことから、このクラスタ内のプレイヤーの移動範囲は、マップの全体に広いことがわかる。そのなかでも、村の上側と下側に、小さく分割されたエリアと遷移が特徴的である。これより、このクラスタに分類されたプレイヤーは、移動範囲は広いが、行動の目的は村の上下にいる敵との戦闘ではないかと考えられる。

また、軌跡の分布は多いが、遷移回数は少ない場所(図 11, 12 において、エリアは小さく分割されているがそのエリアへの遷移、またはそのエリアからの遷移が描かれていない場所)は、プレイヤーが動いておらず、

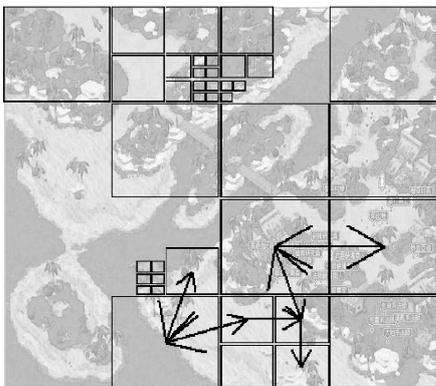


図 11. クラスタごとの動的マップ分割と遷移の一例

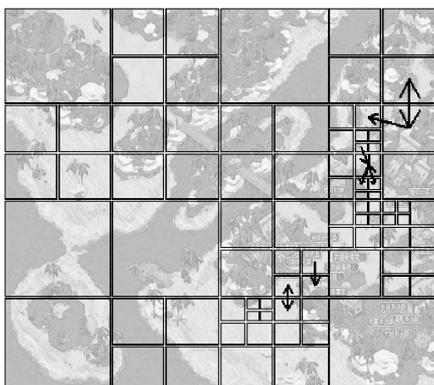


図 12. クラスタごとの分割マップと遷移の一例

同じ座標、もしくはエリア内にとどまっていることが多いことを意味する。このゲームでは、プレイヤーが一定時間座り続けることで徐々に体力など、敵との戦いで消費したものを回復できる機能がある。敵が存在するエリア付近での停滞は、プレイヤーがその場所で回復していることを示し、施設内ではサポートを受けることにより動けないのではないかと推測できる。

## 6. おわりに

Quake II の実験結果から各手法単独ではクラスタリングが不十分であり、本手法のように 2 段階に分けてクラスタリングを行う方法が効果的であることを示すことができた。ロールアップ処理を用いた、遷移確率に基づく分類の単独処理に要した時間に比べ、本手法の方が短い時間で結果を得られたことより、座標分布に基づいた分類が、ロールアップ処理時の一度に比較する軌跡の数を減らし、結果を得るまでの必要とする時間を減少させることができたと考えられる。また、ALO の実験では、応用の一例として本手法によるクラスタリング結果とマップの情報を関連付けたプレイヤーの行動推測の方法を示した。軌跡の分布と遷移に分けることにより、プレイヤーが動いているエリアと、停滞しているエリアとの区別も、クラスタごとの特徴の描写により視覚的にわかりやすく表現することができた。より多くのマップ情報(施設に関する詳細や敵の分布など)があれば、プレイヤーの行動についてより具体的な推測ができると考えられる。

軌跡データから得られる情報として速度があげられる。今後の発展として、この移動速度を含めた移動軌跡の分類による、より具体的で正確な行動分析や、3次元空間上でのマップ(空間)分割への応用があげられる。また、本手法の汎用性を評価するためにほかの時系列データを用いて実験し、他の手法との比較をしていきたいと思う。

## 参考文献

- [1] R. Thawonmas, M. Kurashige and K.T. Chen, "Detection of Landmarks for Clustering of Online-Game Players," *International Journal of Virtual Reality*, Vol. 6, No. 3, pp. 11-16, 2007.
- [2] 平野将康, Ruck Thawonmas, "仮想空間におけるユーザの移動量に注目した移動通路抽出法," *ゲーム学会和文論文誌*, Vol. 2, No. 1, pp. 20-27, 2008.
- [3] K.T. Chen, H.K.K. Pao, and H.C. Chang, "Game Bot Identification Based on Manifold Learning," *Proceedings of the 7th ACM SIGCOMM Workshop*

on Network and System Support for Games, pp. 21-26, 2008.

[4] 倉重 正義, "MMOGにおけるランドマアの自動決定とプレイヤーのクラスタリング," 立命館大学理工学研究科情報理工専攻修士論文, 2008年3月.

[5] 石川 佳治, 町田 陽二, 北川 博之, "マルコフ連鎖モデルに基づく移動ヒストグラムの動的構築法," 電子情報通信学会論文誌 D, Vol. J90-D, No. 2, pp. 311-324, 2007.

[6] 鈴木直彦, 平澤宏祐, 田中健一, 小林貴訓, 佐藤洋一, 藤野陽三, "人物移動軌跡データ群における逸脱行動人物検出及び行動パターン分類," 電子情報通信学会論文誌 D, Vol. J91-D, No. 6, pp. 1550-1560, 2008.

[7] 神田 崇行, 塩見 昌裕, 野村 竜也, 石黒 浩, 萩田 紀博, "RFID タグを用いた科学館来館者の移動軌跡の分析," 電子処理学会論文誌, Vol. 49, No. 5, pp. 1727-1742, 2008.

[8] T. Calinski and J. Harabasz, "A dendrite method for cluster analysis," Communication in Statistics, Vol. 3, No. 1, pp. 1-27, 1974.

[9] J. Lin, E. Keogh, S. Lonardi, and B. Chiu, "A Symbolic Representation of Time Series, with Implications for Streaming Algorithms," Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, pp. 2-11, 2003.

[10] S. Elnekave, M. Last, and O. Maimon, "A Compact Representation of Spatio-Temporal Data," Proceedings of the 7th IEEE International Conference on Data Mining, pp. 601-606, 2007.

[11] M. Vlachos, M. Hadjieleftheriou, D. Gunopluos, and E. Keogh, "Indexing Multi-Dimensional Time-Series with Support for Multiple Distance Measures," Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 216-225, 2003.

[12] A. Anagnostopoulos, M. Vlachos, M. Hadjieleftheriou, E. Keogh, and P.S. Yu, "GlobalDistance-Based Segmentation of Trajectories," Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 34-43, 2006.

[13] J.H. Ward, "Hierarchical Grouping to optimize an objective function," Journal of American Statistical Association, Vol. 58, pp. 236-244, 1963.

[14] Christian Thureau, Christian Bauckhage, and Gerhard Sagerer, "Synthesizing Movements for Computer Game Characters," Lecture Notes in Computer Science, Vol. 3175 (DAGM'04), C.E. Rasmussen et al. (Eds.), pp. 179-186, 2004.

## 著者紹介



織田 淳一

2009年3月立命館大学知能情報学科卒業。現在、同大学大学院理工学研究科情報理工専攻博士課程前期課程に在籍。移動軌跡解析の研究に従事。



Ruck Thawonmas

1994年東北大学大学院工学研究科情報工学専攻博士課程修了。博士(工学)。理化学研究所などを経て、2004年4月より立命館大学情報理工学部知能情報学科教授。ゲームAI, オンラインゲームにおける知的獲得に興味をもつ。IEEE (Senior Member), ゲーム学会などの学会の各会員。



Kuan-Ta Chen

Since 2006 August he is an assistant research fellow at the Institute of Information Science and the Research Center for Information Technology Innovation (joint appointment) of Academia Sinica in Taiwan. His research interests include Internet & Multimedia Quality of Experience (QoE) management, Internet measurement, VoIP, network security, and online games. He is a member of ACM, IEEE, IICM, and CCISA.

ここは学会で  
記載します。

2009年〇月〇日 論文受理  
2009年△月△日 採録決定