F-15

Stay Point による簡略化と動的マップ分割を用いた学習型の軌跡の分類手法

織田 淳一†　　THAWONMAS, Ruck†
Junichi Oda　　Ruck Thawonmas

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てモデル化を行い,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.

†立命館大学大学院理工学研究科，Graduate School of Science and Engineering, Ritsumeikan University

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(図 1)．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, 二つのクラスタ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.

モバイル機器を利用して移動軌跡を取得, 解析し, それらをもとに, 推奨する移動経路や場所をユーザに提示するという研究もおこなわれている[16]. 集めたデータから, これまでのユーザの遷移履歴を, ウェブのリンク付けによるランキングシステムと同様の方法でランク付けを行い, 推奨される過去のユーザの遷移を表示するという方法を提案している.

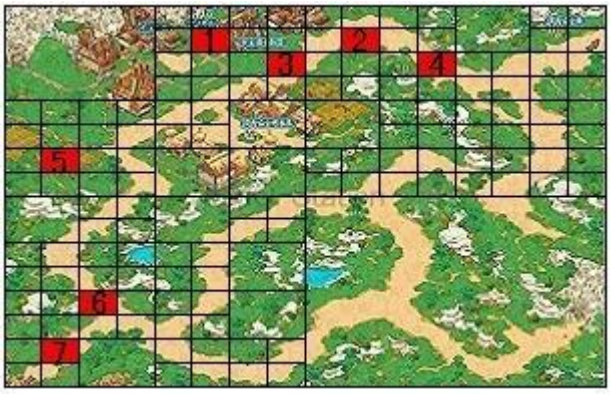

図 1. マップの動的分割とランドマーク(赤格子)の決定

3. 提案手法

状態遷移を用いた多くの手法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)のような時系列データであり，座標データの集合を指す．また，全軌跡数を N とする.

$$(x_i(0), y_i(0)),\cdots(x_i(j), y_i(j)),\cdots(x_i(T_i), y_i(T_i)) \quad (1)$$

ここで T_i は i (i = 1,..., 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, 対象となるエリアのサイズ以下の値をとる.

3.1 *Stay Point* の抽出

軌跡の簡略化を行う. *Stay Point* というのは, 一定の時間(T_{thresh}), あるいは一定の距離(D_{thresh})以内のすべての点を一点で表現する方法であり, 長く停滞している, もしくは大きく移動した場合に次の点が表示される. これにより, 各点が時間に関して, あるいは距離に対して一定範囲を示すことになり, 各点の持つ情報量が大きくなり, またデータを圧縮できる. *Stay Point* の座標は範囲内の点の平均の値が用いられる(2, 3).

$$s_{x_i} = \frac{1}{|P|} = \sum_{j=m}^{n} p_{x_i}(m) \quad (2)$$

$$s_{y_i} = \frac{1}{|P|} = \sum_{j=m}^{n} p_{y_i}(m) \quad (3)$$

ここで, P は $p_i(m), p_i(m+1), \cdots, p_i(n)$ を示し, $p_i(m)$ と $p_i(j)$ の距離の差は D_{thresh} 以内, $p_i(m)$ から $p_i(n)$ の時間の差は T_{thresh} 以内となる.

3.2 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(4)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 2)．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っ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.

$$d = \frac{1}{T_i}\sum_{j=1}^{T_i}\left\{\begin{array}{ll} 1, & if \begin{pmatrix} X_{high} \le x_i(j) \le X_{low} \\ and \\ Y_{high} \le y_i(j) \le Y_{low} \end{pmatrix} \\ 0, & otherwise \end{array}\right\} \quad (4)$$

$X_{high}, X_{low}, Y_{high}, Y_{low}$ 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 3）．

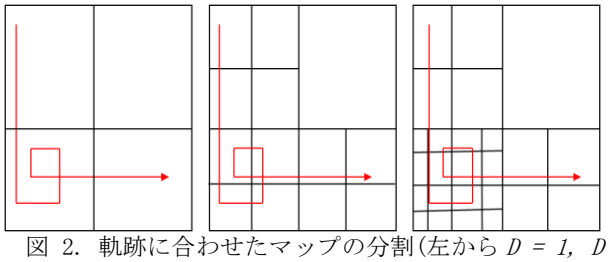

図 2. 軌跡に合わせたマップの分割（左から *D = 1, D =2, D = 3* を示す）

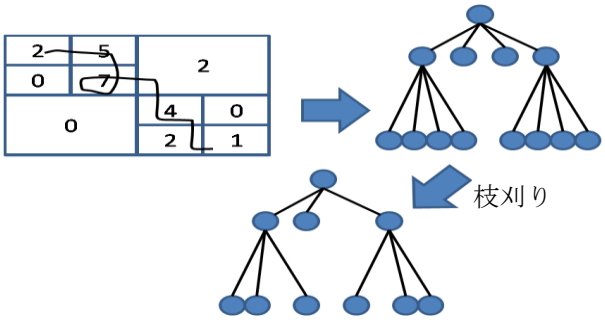

図 3. 動的マップ分割と 4 分木の対応，および枝刈り(エリア内の数値はエリア内に属するデータ数)

3.3 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 4). 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 5). 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).

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

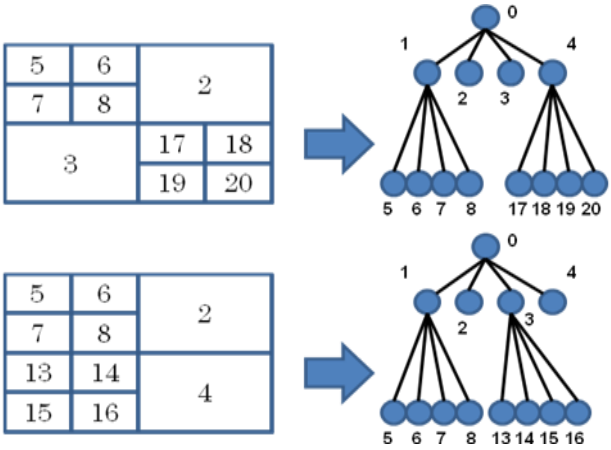

図 4. Z-ordering に基づくノード番号の割り当て(エリア内の数値はノード番号に対応)

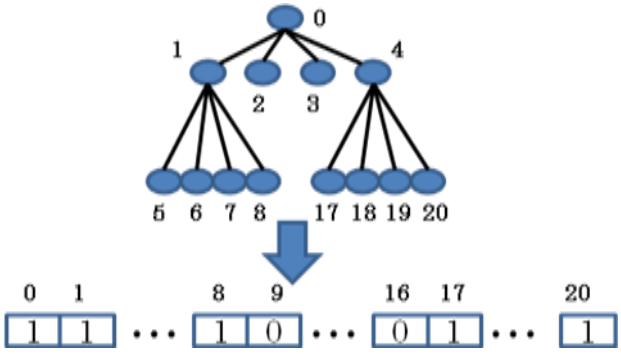

図 5. 4 分木とビット列の対応

以下に，動的マップ分割（4 分木作成）のアルゴリズム（*Dynamic Map Division*）を示す．このアルゴリズムは再帰アルゴリズムであり，*Recall* 文により再帰的に処理が行われる．

Procedure: Dynamic Map Division
Input: 軌跡データ (1)
構造を示す *bit* 列
分割レベル *D* 初期値は 0
エリア（ノード）番号 num 初期値は 0
対象となるエリア（ノード，初期は *root*）の
$X_{high}, X_{low}, Y_{high}, Y_{low}$

Return: 4 分木 *Quadtree*
1 *If* *D* < D_{max}
2 エリア全体を 4 分割
3 分割後の各エリアについて以下の処理を行う
4 エリアの分割レベルを増やす
5 エリア番号の割り当て
6 対応する *bit* 値の変更
7 (2) を用いて *d* を求める
8 *If* *d* > 閾値
9 *Recall*(対象のエリアをマップ分割)
10 *Return Quadtree*

4 行目では分割後出来上がった各エリアの分割レベルを 1 増やす処理を行っている．5 行目では(6)に従ってエリア番号を割り当てる．ここでの処理が Z-ordering にあたる．

$$num(k) = 4num + k \tag{6}$$

k は新しく作られる子ノードの番号をさし，1〜4の値をとる．

6 行目では長さ *B* のビット列の *num(k)*番目の値を 1 に変更し，マップ分割により *num(k)*番目のノードが作成されたことを示す．9 行目の *Recall* で再びこのアルゴリズム呼び出されている．このとき引数となるのが，軌跡データ，*bit* 列，4 行目で処理された分割レベル，*num(k)*，エリアの $X_{high}, X_{low}, Y_{high}, Y_{low}$ となる．

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(7). 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 6).

$$Dist_{ij} = \sum_{n=1}^{B} \left\{ bits_i(n) \oplus bits_j(n) \right\} \quad (7)$$

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.

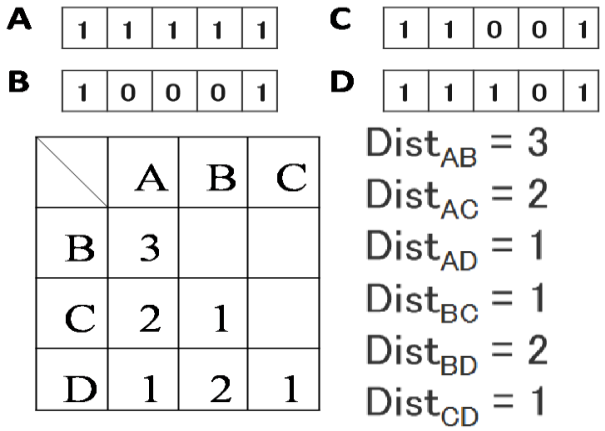

図 6. $N = 4$, $B=5$ としたときの距離行列作成例

3.4 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, 軌跡データの分布の違いに基づいた比較を行っている. したがって, 行動範囲が似ているもの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.

$$CH(K) = \frac{traceH/(K-1)}{traceU/(N-K)} \quad (8)$$

$$traceH = \sum_{k=1}^{K} N_k Dist_{C_k C} \quad (9)$$

$$traceU = \sum_{k=1}^{K} \sum_{i=1}^{N_k} Dist_{C_k k_i} \quad (10)$$

$$C = \min_n \frac{1}{N-1} \sum_{i=1, i\neq n}^{N} Dist_{ni} \quad (n = 1, \ldots, N) \quad (11)$$

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, 以降の操作を行わないようにする.

3.5 モデル間の比較

ここまでの操作により, 軌跡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(12)を行ってから, 状態間の遷移確率を求める.

$$\begin{pmatrix} x_i(0) \\ y_i(0) \end{pmatrix}, \cdots \begin{pmatrix} x_i(j) \\ y_i(j) \end{pmatrix}, \cdots \begin{pmatrix} x_i(T_i) \\ y_i(T_i) \end{pmatrix} \quad (12)$$
$$\Rightarrow (l_i(0)), \cdots (l_i(j)), \cdots (l_i(T_i))$$

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(13) (図 7).

$$bits_{com} = bits_i \ \wedge \ bits_j \quad (13)$$

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(14~16).

遷移前と遷移後の状態が l'の場合

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

遷移前の状態が l'の場合(m : 任意の遷移後の状態)

$$a_{l'm} = \frac{1}{A(l')} \sum_{i=1}^{A(l')} a_{l'(i)m} \quad (15)$$

遷移後の状態が l'の場合(m : 任意の遷移前の状態)

$$a_{ml'} = \sum_{i=1}^{A(l')} a_{ml'(i)} \quad (16)$$

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 7. 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 7 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 7 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(17)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(17)．

$$w_l = \frac{D_{\max} - D_l + 1}{D_{\max}} \tag{17}$$

$$Dist_{ij} = \frac{1}{h_{num}} \sum_{l=1}^{h_{num}} \sum_{m=1}^{h_{num}} w_l w_m \left| a_{lm}(i) - a_{lm}(j) \right| \tag{18}$$

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(18)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，行動範囲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遷移の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，行動範囲と行動遷移の似ているものが同じクラスタに収められる．

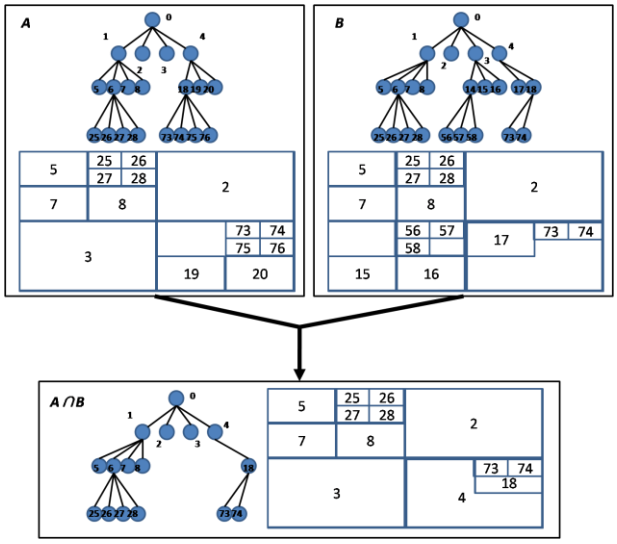

図 7. 共通するノードの抽出(エリア内の番号はノードの番号と対応する)

4. Quake による分類実験

QuakeII 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 8)．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でできたもの(*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 9). 本研究では[15]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．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．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行動する傾向があるのに対し，*player* の軌跡は広い範囲に分散する傾向がある．軌跡の比較により，この 4 種類の判別ができるかどうかを実験する．

図 8. Quake II プレイ画像

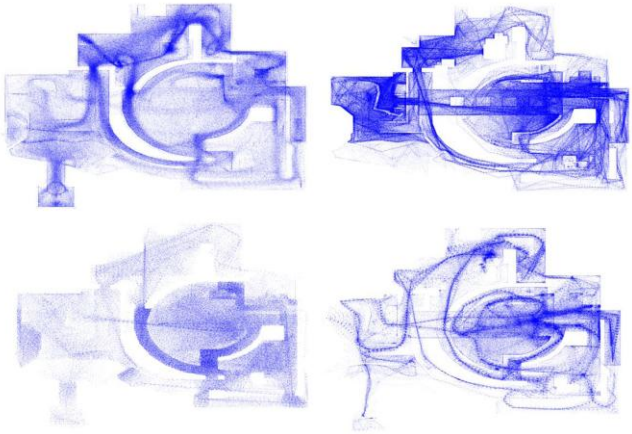

図 9. *player* と *bot* の軌跡(左上：*player*, 右上：*Crbot*, 右下：*Eraser*, 左下：*Ice*)

4.1 テストデータの抽出

全軌跡は 171 人分あり, そのうち ***player*** の軌跡は 105, ***Crbot*** の軌跡は 25, ***Eraser*** の軌跡は 21, ***Ice*** の軌跡は 20 となっている. 実験では各種類から 10%分(***player***:10, ***Crbot***:2, ***Eraser***:2, ***Ice***:2)の軌跡をランダムで選択してテストデータとして使用し, 残りを学習用として使用する.これを 10 回繰り返して実験し, 学習データの分類における情報エントロピー(19, 20), テストデータの分類の精度(21)と再現率(22)を算出する.

$$H = \frac{1}{C}\sum_{i=1}^{C} h_i \tag{19}$$

$$h_i = -\sum_{i=1}^{4}\left(\frac{Type_i}{C_i}\log_2\frac{Type_i}{C_i}\right) \tag{20}$$

ここで H は学習データの分類結果の情報エントロピーの平均, C は分類によりできたクラスタ数, h_i は各クラスタの情報エントロピー, Type は軌跡の種類を意味し, クラスタ内にある ***player***, ***Crbot***, ***Eraser***, ***Ice***, それぞれの数を示す. C_i はクラスタ内にあるすべての軌跡の数を示す.

$$Accuracy = \frac{Type\text{が正しかったテストデータの数}}{\text{全テストデータの数}} \tag{21}$$

$$\mathrm{Re}callRatio = \frac{Type\text{が正しかったbotデータの数}}{\text{テストデータ中の}bot\text{データの数}} \tag{22}$$

また, テストデータと学習データにより出来上がったクラスタとの比較方法は, クラスタ内の軌跡すべてを用いて, 動的マップ分割を行い, 分割マップと遷移確率を得たのち, テストデータのそれぞれの分割マップと遷移確率を比較する方法をとる. 分割マップの構造の差, 遷移確率の差がともに小さいクラスタに, テストデータが分類される..

4.2 実験

実験にあたり, 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.

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.

Stay Point の閾値について, T_{thresh} は 1, 3, 5 分の 3 通り, D_{thresh} については, 縦横それぞれの最大値(2240, 1050)から 0.3, 0.5, 0.7 倍した範囲を D_{thresh} と定義して実験を行った. 同時に, *Stay Point* を用いていないものも実験した.

4.2 結果

実験結果を図 10 に示す. *Stay Point* を用いていないものに比べて, いくつかの組み合わせで, エントロピーが低く, また, 精度, 再現率がともに大きい値を示す結果が得られた. 特に, T_{thresh} が 1, D_{thresh} が 0.7 のとき, *Stay Point* を用いていないもののデータ容量が 49.3MB なのに対して, 3.77MB という小さなデータ容量で同様の成果が得られた

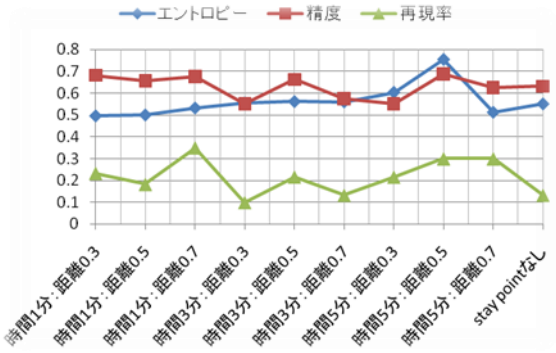

図 10. 実験結果.

5. おわりに

Quake の実験結果から, データ容量を大幅に削減しても, 元のデータで得られるクラスタリングの精度を損なわないことが示された. *Stay Point* の抽出により, あまり重要でないデータが削減されたことが理由であると推測できる. また, 容量を小さくすることで, 処理時間の短縮にも貢献できる.

今後は, クラスタリングの精度を上げるとともに, 行動予測についての学習モデルの構築につながるような取り組みをしていきたい.また, 移動軌跡から得られるほかの情報(加速度など)を用いた分類や, 視覚化についても研究を進めていきたい.

参考文献

[1] Ruck Thawonmas, Masayoshi Kurashige and Kuan-Ta 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] Kuan-Ta Chen, Hsing-Kuo Kenneth Pao, and Hong-Chung Chang,
"Game Bot Identificationbased on Manifold Learning," Proceedings of ACM NetGames 2008, 2008.
[4] 倉重 正義, "MMOG におけるランドマー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, May 2008, pp. 1727-1742.
[8] T. Calinski and J. Harabasz, "A dendrite method for cluster analysis," Communication in Statistics, vol.3, pp.1-27, 1974.
[9] Jessica Lin, Eamonn Keogh, Stefano Lonardi, and Bill Chiu, "A Symbolic Representation of Time Series, with Implications for Streaming Algorithms," DMKD03:8th, ACM SIGMOD, Workshop on Research Issues in Data Mining and Knowledge Discovery, 2003.
[10] Sigal Elnekave, Mark Last, and Oded Maimon, "A Compact Representation of Spatio-Temporal Data," Seventh IEEE International Conference on Data Mining, pp. 601-606, 2007.
[11] Micail Vlachos, Marios Hadjieleftheriou, Dimitrios Gunopluos, and Eamonn Keogh, "Indexing Multi-Dimensional Time-Series with Support for Multiple Distance Measures," SIGKDD’03, August 24-27, 2003.
[12] Aris Anagnostopoulos, Michail Vlachos, Marios Hadjieleftheriou, Eamonn Keogh, and Philip S. Yu, "GlobalDistance-Based Segmentation of Trajectories," KDD’06, August 20-23, Philadelphia, Pennsylvania, USA, 2006.
[13] J.H. Ward, "Hierarchical Grouping to optimize an objective function," Journal of American Statistical Association, 58(301), pp. 236-244, 1963.
[14] Christian Thurau, Christian Bauckhage, and Gerhard Sagerer, Synthesizing Movements for Computer Game Characters, In C.E. Rasmussen, H.H. Bülthoff, and et. al., editors, Pattern Recognition (DAGM’04), volume 3175 of LNCS, pages 179–186. Springer, 2004.
[15] K.-T. Chen, A. Liao, H.-K K. Pao, H.-H. Chu. Game Bot Detection Based on Avatar Trajectory. In Proceedings of IFIP ICEC 2008, 2008.
[16] Zheng, Y., Zhang L., Xie, X. and Ma, W. Y.Mining. Interesting Locations and Travel Sequences From GPS Trajectories For Mobile Users. Submitted to WWW 2009
[17] Ruck Thawonmas, Junichi Oda and Kuan-Ta Chen, "Analysis of User Trajectories Based on Data Distribution and State Transition: a Case Study with a Massively Multiplayer Online Game Angel Love Online," Proc. of the 10th International Conference on Intelligent Games and Simulation (GAME-ON 2009), Dusseldorf, Germany, pp. 56-60, Nov., 2009.