動的属性付き軌跡データの類似検索

大橋　英明† 　清水　敏之†† 　吉川　正俊††

† 京都大学大学院情報学研究科　〒 606 - 8501 京都府京都市左京区吉田本町
†† 京都大学大学院情報学研究科　〒 606 - 8501 京都府京都市左京区吉田本町
E-mail: †ohashi@db.soc.i.kyoto-u.ac.jp, ††{tshimizu,yoshikawa}@i.kyoto-u.ac.jp

あらまし　近年，センシング技術の発達等によって膨大な移動軌跡データが収集されており，軌跡データに対する研究が数多く行われている．その中で，軌跡間の類似を求める手法は軌跡データのクラスタリングや頻出パターン抽出などの基盤手法となっている．しかし，単に観測点の座標をもとに類似を求めるだけでは，有用な結果を得ることができない場合がある．例えば，サッカー選手の動きを軌跡データとして捉え，ある選手の動きと類似した動きをする選手を検索するような状況を想定する．検索結果を選手分析等に利用する場合，座標のみではなく，選手の移動速度やボールとの位置関係など，動的に変化する属性値を考慮して類似を求める必要があると考えられる．そこで我々は，オブジェクトの動的な属性値を付与した軌跡データ（動的属性付き軌跡データ）に対して，柔軟な類似検索を行うための手法の枠組みと，高速化のためのアルゴリズムを提案し，その有効性を検討する．

キーワード　軌跡，類似検索

1. はじめに

近年，センシング技術の発達により，大量の移動軌跡データが収集されている．収集された軌跡データの例として，歩行者の移動データ [15]，ハリケーンのトラッキングデータ [8]，スポーツのデータ [5], [12] 等があり，それぞれ経路の推薦，傾向の分析，選手の行動分析等に活用され得る．軌跡データの増加に伴い，分析を実現するためのクラスタリング [8] や外れ値探索 [7] などの手法が数多く提案されてきた．我々はこのような様々な分析手法の基盤となる軌跡の類似検索手法に焦点を当てる．

センサーで収集された移動軌跡データに対する類似検索手法として，移動軌跡データを位置情報に基づく二次元の時系列データとして捉え，軌跡間の類似関係を求めるような手法が考えられる．しかし，位置情報のみに基づく類似検索では有用な結果が得られない場合がある．例えば，あるサッカー選手の動きと類似した動きをする選手を検索するような状況を考える．この場合，単に座標のみではなく，選手の移動速度やボールとの位置関係など，動的に変化する属性値を考慮して類似を求めなければ，選手分析に用いることができるような有用な結果は得られないと考えられる．

本研究は，軌跡の動的属性を考慮した類似軌跡検索手法の枠組みを提案する．軌跡の動的属性とは時間経過に合わせて変化する属性（座標を除く）のことを指す．動的属性の中には，軌跡固有の情報から導出できるものと軌跡以外の情報と軌跡データの組み合わせから導出される属性が存在する．本研究では，前者を内的属性，後者を外的属性と表現する．内的属性の例として，隣接する二点の座標のみから大まかに算出できる「進行方向」などが挙げられる．外的属性の例として，サッカー選手の移動軌跡データを想定すると，ボールの位置情報と選手の移動軌跡データの組み合わせから導出できる「選手とボールの位置関係」などが挙げられる．

内的属性を扱った研究 [2], [9] は以前から数多く行われてきたが，外的属性を扱った研究は，複数のデータを横断利用することが重要視されてきた近年，注目されつつある [14]．なお，外的属性はデータ取得の観点から二種類に分類できると考えられる．一つ目は「軌跡と同時に記録され，観測点と紐づけられた属性」である．例えば，ハリケーンのトラッキングデータにおける気圧や，歩行者の移動軌跡データにおける心拍数などが例にあたる．二つ目は「軌跡を他のデータと結びつけることによって得られる属性」である．例えば，タクシーの移動軌跡データにおける走行路の制限速度情報や，スポーツ選手の軌跡データにおけるボールとの位置関係などが例にあたる．前者はタクシーの移動軌跡データと道路地図データを，後者は選手の移動軌跡データとボールの位置データを結びつけることによって取得できる．上記の二種類の属性は，属性値を取得する過程は異なるが，軌跡固有の情報から導出できないという事実を共通して持つため，本研究では同等のものとして扱う．

本研究では，内的属性，外的属性を考慮した拡張観測点間の距離を定義し，DTW(動的時間伸縮法) [1] を用いて，動的属性を考慮した類似度の算出を行う．軌跡データの中でも，スポーツデータのようにオブジェクトごとの挙動の差が激しい軌跡データの類似を求める手法として，DTW の他に LCSS [11], EDR [3] などが提案されてきた．しかし，本研究では詳細な距離の差を考慮して軌跡の距離を定義することを考えており，カウントベースの距離である LCSS や EDR ではなく，DTW を用いることが適切であると考えた．

また，本研究の手法に特化した高速化のためのアルゴリズムを提案した．このアルゴリズムの有効性を示すため，ハリケーンの実データと人工的に作成したデータを対象に実験を行った．その結果，アルゴリズムを用いなかった場合と比べて，3〜17 倍の速度向上が見れらた．

本論文の構成を以下に示す．第 2 節では，軌跡に関する関連研究の紹介を行う．第 3 節では，本研究の問題設定の詳細な説明を行い，第 4 節では，検索を高速化するためのアルゴリズムを提案する．第 5 節では，提案手法とそのアルゴリズムの有効性を検討するために行った実験について述べる．第 6 節では，本稿のまとめを述べる．

2. 関 連 研 究

本節では，内的属性を扱った既存研究，外的属性を扱った既存研究，軌跡の距離を扱った既存研究を紹介する．

2.1 内的属性を扱った研究

Pelekis らの研究 [9] は軌跡の時空間情報，速さの情報，進行方向の情報を考慮した類似検索手法を提案している．また，Buchin らの研究 [2] は軌跡を特徴付ける基準を用いて，軌跡を分割する枠組みを提案している．基準として，location, heading, speed, velocity, curvature, sinuosity, curviness と呼ばれる独自の指標を提案しており，これらの指標は全て軌跡固有の情報のみから算出される．以上二つの研究は内的属性のみを扱っている研究である．

2.2 外的属性を扱った研究

Zheng らの研究 [13] は，軌跡データと地図データを結びつけることによって，各観測点にラベルを付与した軌跡 (activity trajectory) を生成し，これに対する検索手法を提案している．本田らの研究 [16] は位置情報を伴う車両走行センサデータから車両走行の特徴を抽出し，要約・表現する手法を提案している．以上二つの研究は外的属性のみを扱っている研究である．

2.3 軌跡の距離を扱った研究

軌跡間の距離を表す指標として，DTW [1], LCSS [11], EDR [3] などが挙げられる．DTW はもともと信号解析に使われていた技術であったが，Berndt ら [1] によって時系列データマイニングに取り入れられた．DTW はマイニングの手法として有用である一方，コストが非常に高いことで知られており，コストを減らすための研究が数多く行われている [4], [6], [10]．LCSS は Vlachos ら [11] によって提案された手法である．最長共通部分列の考え方を取り入れており，DTW よりもノイズに強いという性質を持つ．EDR は Chen ら [3] によって提案された手法である．編集距離の考え方を取り入れており，LCSS よりも精度が高くノイズにも強い手法とされている．ただし，LCSS と EDR は二点間に定義された尺度がある閾値以上の場合に距離が 1 増加するような距離関数を定義している．つまり，LCSS と EDR は DTW と異なり詳細な距離の差を考慮できない．例えば，軌跡データに対して LCSS や EDR を単純に適用した場合，ある二点間のユークリッド距離が閾値よりわずかに大きい場合と非常に大きい場合を同じものとして扱ってしまう．これは，様々な尺度を用いて詳細な距離を測る本研究にとって不適当な性質であるため，本研究では DTW を利用する．また，Lee らの研究 [8] は，セグメント間の距離を複数の尺度を合成することによって導出している．Lee らは複数の尺度を合成する際に線形結合を用いているが，本研究も座標，内的属性，外的属性という複数の尺度を合成する際に線形結合を用いる．

3. 問 題 設 定

本研究は，動的属性付き軌跡集合が与えられた際に，その中から問合せとして一つの軌跡を選択し，その軌跡と最も類似した軌跡を出力するような問題を扱う．なお，軌跡間の類似の基準は利用者の意図によって異なると考えられる．例えば，サッカーの軌跡データに対して，ある利用者が過去の試合のデータを用いて選手分析をする状況を想定する．この時，ある選手の動きに対して，ドリブルの動きが類似した他選手の動きを得たいというニーズや，ポジションの取り方が類似した他選手の動きを得たいというニーズが想定される．前者の場合は，選手の進行方向や速度に重きをおくべきだと考えられ，後者の場合は，他選手やボールとの位置関係に重きをおくべきだと考えられる．このように，利用者の意図によって類似の基準は変化する．よって本研究では，様々なニーズに対応できるように，軌跡の入力に加えて，各動的属性に対する重みの入力も利用者に委ねることを想定している．

以下，本研究で取り扱う問題について詳細な定式化を行う．

3.1 動的属性付き軌跡

ある軌跡の識別子を i $(i = 1, 2, \ldots, n)$，軌跡の観測点の数を n, 軌跡の中で j 番目に観測された点の座標を $p_i(j)$ $(j = 1, 2, \ldots, n)$ とする．軌跡は時系列に並べられた観測点のリストで表現されるため，識別子 i を持つ軌跡 p_i は以下のように定義できる．

$$p_i =< p_i(1), p_i(2), \ldots, p_i(n) > \quad (1)$$

ただし本研究では，すべての観測点は等時間間隔で観測されていると仮定する．続いて，動的属性について具体例を交えて説明する．軌跡の動的属性とは時間経過に合わせて変化する属性（座標を除く）を指す．本研究では，動的属性を内的属性と外的属性に分別して考える．

内的属性とは，軌跡の情報のみから導出できる属性のことを表す．Buchin らの研究 [2] では，heading, speed, curvature, sinuosity など多様な属性が独自に定義され，用いられていた．本研究では，特に「速さ」と「進行方向」を扱う．これら二つの属性は多様な軌跡において扱われ得る属性であると考えられるため，また導出が比較的容易であるため，本研究で取り扱うことにした．速さを示す属性値は，観測点が等時間間隔で観測されているという仮定より，隣接する二点間のユークリッド距離を用いて近似できると考える．よって，隣接する二点間のユークリッド距離を速さを示す属性値として扱う．また，進行方向を示す属性値も隣接する二点の座標から近似的に求めることができる．本研究では，これらの隣接する二点のうち，観測時刻が早い方の観測点に対して内的属性を付与する．

外的属性とは，外部の情報と軌跡の組み合わせから導出される属性のことを表す．例えば，ハリケーンの軌跡を考える場合，ハリケーンの中心気圧や最大風速などが外的属性に相当する．また，サッカー選手の軌跡を考える場合，他選手やボールとの位置関係などが外的属性に相当する．前者は「軌跡と同時に記録され，観測点と紐づけられた属性」であり，属性値を得るた

めの処理を必要としないが，後者は「軌跡を他のデータと結びつけることによって得られる属性」であり，属性値を得るための処理を必要とする．ただし，本研究では外的属性がどのように付与されたかは考慮せず，あらかじめ与えられるものとする．また，外的属性値も，内的属性値と同じく，各観測点に付与される．

以上を踏まえて，座標，内的属性値，外的属性値の組を拡張観測点と呼ぶことにする．

$$ep_i(j) = (p_i(j), iv_i(j), ev_i(j)) \tag{2}$$

$iv_i(j)$ は内的属性値，$ev_i(j)$ は外的属性値を表す．動的属性付き軌跡は時系列に並べられた拡張観測点のリストとして表現可能であり，本研究ではそのリストを拡張軌跡と呼ぶことにする．

$$ep_i =< ep_i(1), ep_i(2), \ldots, ep_i(n) > \tag{3}$$

なお，上述の例では，内的属性と外的属性が各々1 種類だけ存在するように定義されているが，複数属性への拡張も容易に行うことができると考えている．

3.2 拡張観測点間の距離

本研究は，拡張観測点間の距離を定義し，DTW を用いることによって，拡張軌跡間の距離を計算する．二つの拡張観測点間の距離は以下の式によって求めることができると考えた．

$$EDist(ep_i(k), ep_j(l)) = w_S \cdot d_S + w_I \cdot d_I + w_E \cdot d_E \tag{4}$$

$$d_S = Dist_S(p_i(k), p_j(l)) \tag{5}$$
$$d_I = Dist_I(iv_i(k), iv_j(l)) \tag{6}$$
$$d_E = Dist_E(ev_i(k), ev_j(l)) \tag{7}$$

w_S, w_I, w_E は重みを，$Dist_S$ は座標に関する距離を求める関数，$Dist_I$ は内的属性に関する距離を求める関数，$Dist_E$ は外的属性に関する距離を求める関数を表す．また，$w_S \geqq 0, w_I \geqq 0, w_E \geqq 0, w_S + w_I + w_E = 1$ とする．重みの最適な数値は利用者の意図ごとに異なると考えられる．なお，定式化の段階では，内的属性と外的属性にそれぞれ一つずつ重みを与えているが，内的属性を複数同時に扱う場合（速さと進行方向の二種類の属性を同時に扱うなど）や外的属性を複数同時に扱う場合（他選手との位置関係とボールとの位置関係の二種類の属性を同時に扱うなど）については，扱われる属性それぞれに対して重みを付与することで対応できると考えている．

4. アルゴリズム

4.1 目　　的

本研究は，利用者の意図に沿った柔軟な検索を実現するため，重みの入力を利用者に委ねることを想定している．しかし，重みを利用者入力とすると，重みが与えられるまで拡張観測点間の距離 (式 (4)) が一意に定まらないため，過去に収集されたデータ集合を用いて類似検索を行うような状況であっても，事前に DTW 距離を求めておくことができない．しかし，DTW の計算はコストが高いため，重みの入力ごとに DTW を逐一計算すると，計算に時間を要すると考えられる．そこで，我々は重みを用いた検索を行う本研究の問題設定に特化した，検索速度向上のためのアルゴリズムを提案する．

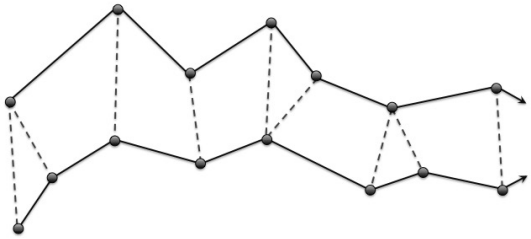

図 1: DTW

4.2 DTW について

アルゴリズムの説明を行う前に，DTW の概要を述べる．長さ k の時系列データ $P =< p(1), p(2), \ldots, p(k) >$ と長さ l の時系列データ $Q =< q(1), q(2) \ldots, q(l) >$ が存在した時，DTW 距離は以下のように定義できる．ただし，$i = 1, 2, \ldots, k$, $j = 1, 2, \ldots, l$ である．また，$p(i)$ と $q(j)$ 間の距離を $D(i, j)$ で表現する．

$$DTW(P, Q) = f(k, l)$$

$$f(i, j) = D(i, j) + min \begin{cases} f(i-1, j) \\ f(i, j-1) \\ f(i-1, j-1) \end{cases}$$

$$f(0, 0) = 0, f(i, 0) = f(0, j) = \infty$$

DTW は動的計画法に基づいた計算手法であり，長さの異なるデータ間の距離を定義することができる．図 1 は二つの軌跡上の観測点を DTW の定義に従い，破線を用いて対応付けた図である．この時，対応付けられた観測点間の距離の総和が DTW 距離となる．本研究では，上式に現れる $D(i, j)$ として式 (4) を用いることによって，二つの拡張軌跡 ep_i, ep_j の距離を導出する．

4.3 提 案 手 法

基本的な考え方は下界 (lower bound) を利用した検索対象の絞り込みである．座標，内的属性，外的属性に関する DTW 距離を，拡張軌跡間の距離を求めるときに用いた重みを使って線形結合した値を，拡張軌跡の類似検索における下界として用いることができる．まずは，例を参照しながらこの事実が成立することを確認する．

図 2 は軌跡 i と軌跡 j の座標に関する距離，内的属性に関する距離，拡張観測点に関する距離の行列と DTW を実行をした際のワーピングパスを示している．ただし，$(w_S, w_I, w_E) = (0.5, 0.5, 0)$ として拡張観測点間の距離 (式

$p_1(4)$	4	1	7	6
$p_1(3)$	1	3	5	2
$p_1(2)$	3	5	4	3
$p_1(1)$	1	2	11	3
	$p_2(1)$	$p_2(2)$	$p_2(3)$	$p_2(4)$

(a) 座標に関する距離行列

$iv_1(4)$	12	6	9	6
$iv_1(3)$	8	5	14	4
$iv_1(2)$	1	2	12	3
$iv_1(1)$	1	3	6	6
	$iv_2(1)$	$iv_2(2)$	$iv_2(3)$	$iv_2(4)$

(b) 内的属性に関する距離行列

$ep_1(4)$	8	3.5	8	6
$ep_1(3)$	4.5	4	9.5	3
$ep_1(2)$	2	3.5	8	3
$ep_1(1)$	1	2.5	8.5	4.5
	$ep_2(1)$	$ep_2(2)$	$ep_2(3)$	$ep_2(4)$

(c) 拡張観測点に関する距離行列

図 2: ワーピングパス

(4)) を求めている．この時，各々のワーピングパスから，座標に関する距離に基づく DTW 距離 (*15*)，内的属性に関する距離に基づく DTW 距離 (*22*)，拡張観測点に関する距離に基づく DTW 距離 (*20*) が導出され，$20 \geqq 0.5 \cdot 15 + 0.5 \cdot 22 + 0$ であるため，確かに先ほどの事実が成立している．

図 2(a) と図 2(b) に示されたワーピングパスは座標に関する距離と内的属性に関する距離の総和が最小となるように選択されたパスである．ゆえに，図 2(c) のパスを用いて，図 2(a) と図 2(b) の距離行列から導出される総コスト (*17* と *23*) は，各々の DTW 距離 (*15* と *22*) 以上の大きさになる．また，図 2(c) のパスを用いた場合の各々の総コスト (*17* と *23*) を，重み $(w_S, w_I) = (0.5, 0.5)$ を用いて合成した値 (*20*) が拡張観測点間の DTW 距離 (*20*) と一致するのは自明である．よって，拡張観測点間の距離に基づく DTW 距離は座標に関する距離に基づく DTW 距離と内的属性に関する距離に基づく DTW 距離を合成した値よりも大きくなる．以下，厳密な証明を行う．まずは次の補題を証明する．

補題 1. 拡張観測点に関する距離行列を用いて DTW 距離を求める際のワーピングパスを WP_{ep} とする．WP_{ep} を用いて，座標，内的属性値，外的属性値の距離行列からコストを導出すると，それらのコストは距離行列から導出される DTW 距離よりも大きくなる．

証明. DTW 距離を求める際のワーピングパスを以下のリストで表す．

$$WP_N =< WP_N(1), WP_N(2), \ldots, WP_N(n) >$$

N には ep, p, iv, ev のいずれかが代入される．WP_N は拡張観測点 (ep)，座標 (p)，内的属性 (iv)，外的属性 (ev) のいずれかに関する距離行列を用いて DTW 距離を導出する際のワーピングパスが表される．長さ k の時系列データ $P_N = p_N(1), p_N(2), \ldots, p_N(k)$ と長さ l の時系列データ $Q_N = q_N(1), q_N(2), \ldots, q_N(l)$ 間の DTW 距離を求める時，ワーピングパスの要素 $WP_N(m)$ は $(m = 1, 2, \ldots, n)$，DTW の定義によって対応付けられる二つの観測点の観測番号の組となる．

$$WP_N(m) = (i, j) \tag{8}$$

ただし，$i = 1, 2, \ldots, k$, $j = 1, 2, \ldots, l$ とする．

$p_N(i)$ と $q_N(j)$ 間の距離を $D_N(i, j)$ で表現する．以下，ワーピングパスを引数にとり，そのワーピングパスを用いて距離行列から総コストを求める関数を定義する．

$$WDist_{N1}(WP_{N2}) = \sum_{m=1}^{m=n} D_{N1}(WP_{N2}(m)) \tag{9}$$

ただし，$N1, N2$ には ep, p, iv, ev のいずれかが代入される．$WDist_{N1}$ は拡張観測点 (ep)，座標 (p)，内的属性 (iv)，外的属性 (ev) のいずれかに関する距離行列から総コストを求める関数を表す．このとき，WP_{N1} は $N1$ の距離行列の中で最小となるように選択されたパスであるため，以下の不等式が成立する．

$$WDist_{N1}(WP_{N2}) \geqq WDist_{N1}(WP_{N1}) \tag{10}$$

式 (10) より，以下の不等式が成立する．

$$WDist_p(WP_{ep}) \geqq WDist_p(WP_p)$$
$$WDist_{iv}(WP_{ep}) \geqq WDist_{iv}(WP_{iv})$$
$$WDist_{ev}(WP_{ep}) \geqq WDist_{ev}(WP_{ev})$$

よって，補題 1 が成立する．

次に定理の証明を行う．

定理 1. 二つの拡張軌跡 ep_i, ep_j と座標，内的属性，外的属性の各々に対応する重み w_S, w_I, w_E が与えられたとき，$DTW(ep_i, ep_j) \geqq w_S \cdot DTW(p_i, p_j) + w_I \cdot DTW(iv_i, iv_j) + w_E \cdot DTW(ev_i, ev_j)$ が成立する．

証明. 補題 1 より，次の式が成立する．

$$\begin{aligned}
&DTW(ep_i, ep_j) \\
&= WDist_{ep}(WP_{ep}) \\
&= w_S \cdot WDist_p(WP_{ep}) + w_I \cdot WDist_{iv}(WP_{ep}) \\
&\quad + w_E \cdot WDist_{ev}(WP_{ep}) \\
&\geqq w_S \cdot WDist_p(WP_p) + w_I \cdot WDist_{iv}(WP_{iv}) \\
&\quad + w_E \cdot WDist_{ev}(WP_{ev}) \\
&= w_S \cdot DTW(p_i, p_j) + w_I \cdot DTW(iv_i, iv_j) \\
&\quad + w_E \cdot DTW(ev_i, ev_j)
\end{aligned}$$

よって，定理 1 が成立する．

定理 1 を利用したアルゴリズム 1 について述べる．

まず前処理として，各々の尺度に関する DTW 距離を計算する (PREPROCESS PHASE)．この計算は重みを与えられるよ

りも前に実行することができる．また，入力軌跡を与えられる以前であっても，すべての軌跡の組み合わせに対して DTW 距離を導出しておくことで前処理を完了できる．次に，前処理で求められた DTW 距離を利用して類似検索を行う (SEARCH PHASE)．重みと DTW 距離から下界 (lb) を算出し，下界が，以前に算出された中で最も短い距離の値 ($dist_{min}$) よりも小ければ，拡張軌跡間の真の距離 ($true\_dist$) を計算し，大きければ，その計算手順を省略する．これにより，検索候補を大きく絞り込むことができる．

なお，アルゴリズム 1 では，座標に関する距離，一つの内的属性に関する距離，一つの外的属性に関する距離という合計三つの尺度を扱っているが，内的属性を複数同時に扱う場合や外的属性を複数同時に扱う場合についても，対応する重みを用意することによって，同様のアルゴリズムを用いることができる．

Algorithm 1 FastExtendedTrajectorySimilaritySearch

INPUT: extended trajectory q
INPUT: $\{c_1, c_2, \ldots, c_n\} \in C$ of candidate extended trajectories
INPUT: w_S, w_I, w_E
OUTPUT: the index of nearest extended trajectory regarding q

▷PREPROCESS PHASE
for $i \leftarrow 1$ to $|C|$ **do**
  $D.p[i] \leftarrow DTW(q.p, c_i.p)$
  $D.iv[i] \leftarrow DTW(q.iv, c_i.iv)$
  $D.ev[i] \leftarrow DTW(q.ev, c_i.ev)$
end for
▷SEARCH PHASE
$dist_{min} = \infty$
for $i \leftarrow 1$ to $|C|$ **do**
  $lb \leftarrow lower\_bound(D[i], W)$ ▷ calculate LB from D and W
  if $lb < dist_{min}$ **then**
    $true\_dist \leftarrow DTW(q.ep, c_i.ep)$
    if $true\_dist < dist_{min}$ **then**
      $dist_{min} \leftarrow true\_dist$
      $index \leftarrow i$
    end if
  end if
end for
return $index$

5. 実　　験

5.1 実験データについて

ハリケーンの実データと人工的に生成したデータを用いて提案アルゴリズムの有効性を示す．始めにそれぞれのデータの概要を説明し，続けてデータに適合した座標，内的属性値，外的属性値の扱い方と距離を求める関数について議論する．

5.1.1 データの概要

ハリケーンの軌跡データとして，Unisys Weather(注1) の大西洋上のハリケーントラックデータを用いた．このトラックデータは軌跡の識別子，タイムスタンプ，緯度，経度，風速，気圧の情報を持つ．本実験では，6 時間ごとに記録された値が欠損していない軌跡データ 450 件を用いる．また，風速と気圧を外的属性として扱う．

また，上記のハリケーンデータにならって，軌跡の識別子，タイムスタンプ，x 座標，y 座標，二つの外的属性を属性として持つようなデータを生成し，それらを人工の軌跡データとした．軌跡は 1000 件で，それぞれの軌跡の長さは範囲 (30:70) からランダムに決定した．x 座標，y 座標，二つの外的属性は文献 [10] を参考にして，以下のようなランダムウォークモデルから生成した．

$$v(i) = v(i-1) + dif(i)$$

ただし，軌跡の先頭 ($v(0)$) の x 座標と y 座標は範囲 (-5:5) からランダムに生成された小数，二つの外的属性は範囲 (-5:5) からランダムに生成された整数を用いている．また，直前のデータとの差分 ($dif(i)$) として，x 座標，y 座標は範囲 (-1:1) からランダムに生成された小数，二つの外的属性は範囲 (-5:5) からランダムに生成された整数を用いている．人工データをハリケーンデータの形に合わせるため，x 座標と y 座標が小数，二つの外的属性が整数となるように生成した．

5.1.2 座　　標

式 (1) では一般化のため，座標を $p_i(j)$ として表現していたが，ハリケーンおよび人工データはオブジェクトの位置を二次元上の座標として表現できるため，$p_i(j)$ は $(x_i(j), y_i(j))$ と表すことができる．また，本実験では座標に関する距離を二次元のユークリッド距離とする．よって，座標に関する距離は以下の式で算出した．

$$\text{座標に関する距離} = \sqrt{(x_k(l) - x_i(j))^2 + (y_k(l) - y_i(j))^2} \quad (11)$$

5.1.3 内的属性値

ハリケーンデータと人工データの両方において，内的属性として「速さ (speed)」と「進行方向 (direction)」を用いる．拡張軌跡上で隣接する二点間の座標の距離を速さの属性値として扱う．よって，速さの属性値は以下の式で算出できる．

$$iv_i(j).speed = \sqrt{(x_i(j+1) - x_i(j))^2 + (y_i(j+1) - y_i(j))^2} \quad (12)$$

次に進行方向の属性値について定義する．拡張軌跡上で隣接する二点のうち，観測時刻が早い方の観測点を始点，遅い方の観測点を終点と呼ぶ．x 軸と並行かつ始点を端点に持ち，x 軸正方向に伸びる半直線と，始点・終点を結ぶ線分がなす角度を進行方向の属性値として扱う．進行方向の属性値は以下の式で算出できる．

$$iv_i(j).direction \quad (13)$$

(注1)：http://weather.unisys.com/hurricane/

$$= \begin{cases} \theta & (x_i(j+1) > x_i(j)) \\ \theta + \pi & (x_i(j+1) < x_i(j), y_i(j+1) \geqq y_i(j)) \\ \theta - \pi & (x_i(j+1) < x_i(j), y_i(j+1) < y_i(j)) \\ \frac{\pi}{2} & (x_i(j+1) = x_i(j), y_i(j+1) > y_i(j)) \\ -\frac{\pi}{2} & (x_i(j+1) = x_i(j), y_i(j+1) < y_i(j)) \end{cases}$$

ただし，

$$\theta = \arctan\left(\frac{y_i(j+1) - y_i(j)}{x_i(j+1) - x_i(j)}\right) \qquad (14)$$
$$(\text{値域は} -\frac{\pi}{2} < \arctan(x) < \frac{\pi}{2})$$

ところで，式 (3) が $1 \leqq j \leqq n$ の範囲で成立するように記述していたが，速さや進行方向を考慮する時，軌跡終端の観測点の内的属性値を定義できない．よって，実験を行う際は，終端の観測点の速さと進行方向として直前の速さと進行方向の値を用いた．また，隣接する二点が一致している場合も直前の値と同じ値を用いた．

続いて，内的属性に関する距離を考える．速さに関する距離と進行方向に関する距離は，それぞれの属性値の差で表現できるものと考えた．また，速さに関する距離を重視する検索者もいれば，進行方向に関する距離を重視する検索者もいると考えられるので，それぞれに重み付けが必要である．

以上より，本研究では内的属性に関する距離を以下の式で扱う．

内的属性に関する距離
$$= w_{iname} \cdot |iv_i(j).iname - iv_k(l).iname| \qquad (15)$$

ただし，$iname$ は内的属性を表し，w_{iname} は内的属性値に掛け合わせる重みを表す．

5.1.4 外的属性値

まず，ハリケーンの外的属性について説明する．本実験では外的属性として，ハリケーンの気圧と風速を用いる．これらのデータは「軌跡と同時に記録され，観測点と紐づけられた属性」であり，特殊な前処理を必要としない．どちらの属性の距離も属性値の差で表すことができると考えた．また，内的属性に関する距離と同じく，検索者によって重視する属性は異なるため，重み付けが必要である．人工データの外的属性もハリケーンデータの外的属性の形に合わせて生成したため，同様に扱う．

外的属性に関する距離を考える．内的属性と同じく，各属性の距離はそれぞれの属性値の差で表現でき，また，重視する属性は利用者によって異なる．

以上より，外的属性に関する距離は以下の式で定義できる．

外的属性に関する距離
$$= w_{ename} \cdot |ev_i(j).ename - ev_k(l).ename| \qquad (16)$$

ただし，$ename$ は外的属性を表し，w_{ename} は外的属性値に掛け合わせる重みを表す．

5.2 性能評価

本実験では、観測点間に定義される距離として，座標，速さ，進行方向，二つの外的属性の計五つの属性に関する距離を扱う．また，一つの軌跡の識別子と五つの属性に関する距離に対応した五つの重みをランダムに組み合わせたデータ 10 件を入力として用いる．なお，重みの総和は 1 となるように生成した．提案したアルゴリズムが有効であることを示すため，文献 [6] を参考に以下の三つの指標を用いて評価を行った．

（1） 下界の大きさ ($Tightness$)

（2） 枝刈りされた軌跡の数 ($PruningPower$)

（3） 類似検索の実行速度

表 1: 実験結果（ハリケーンデータ）

	Tightness	Pruning Power	実行時間 (s)	LB を用いた実行時間 (s)
1	0.807	0.884	14.88	1.94
2	0.800	0.951	10.98	0.77
3	0.827	0.960	17.49	1.21
4	0.810	0.924	14.20	1.75
5	0.868	0.960	8.84	0.57
6	0.815	0.860	15.8	3.08
7	0.738	0.833	13.54	2.64
8	0.811	0.958	14.70	0.84
9	0.770	0.902	11.67	1.34
10	0.798	0.951	10.89	0.67

表 2: 実験結果（人工データ）

	Tightness	Pruning Power	実行時間 (s)	LB を用いた実行時間 (s)
1	0.699	0.824	68.92	12.42
2	0.704	0.897	75.27	8.02
3	0.790	0.918	64.73	5.58
4	0.655	0.755	72.37	17.38
5	0.679	0.822	80.59	15.02
6	0.528	0.630	68.56	24.14
7	0.701	0.822	79.60	14.52
8	0.756	0.930	79.90	6.10
9	0.723	0.887	66.73	7.75
10	0.795	0.887	65.65	7.62

ハリケーンデータを用いた実験結果を表 1 に，人工データを用いた実験結果を表 2 に示した．ただし，下界と DTW 距離を算出する過程で発生する計算機由来の誤差によって，下界の値が DTW 距離よりも大きくなるという事態を避けるため，本実験では下界を本来の値よりも小さな値に，DTW 距離を本来の値よりも大きな値に修正している．

Intel Core i7 3.60GHz，16GB メモリの PC を用いて，Ubuntu 14.04.3 上で，Python3.4.3 を用いて提案手法を実装した．また，事前に算出した DTW 距離を格納するために，postgres9.3.10 を利用した．

5.2.1 下界の大きさ

下界は元の値に近ければ近いほど有効に働く．よって，下界の値を本来の DTW 距離の値で割った値 ($Tightness$) は評価指標として有用である．

$$Tightness = \frac{\text{下界の値}}{\text{本来の } DTW \text{ 距離}} \qquad (17)$$

10 件の入力に対して，入力軌跡とそれ以外の軌跡の組み合わせについて $Tightness$ を求め，平均を取った値を表 1 と表 2 に示した．ハリケーンデータは 0.738〜0.868，人工データは 0.528〜0.795 となっている．ハリケーンデータの $Tightness$ が人工データの $Tightness$ と比べて大きい理由は，ハリケーンデータに座標や外的属性の変化の傾向が類似しているものが多く含まれるからだと考えられる．下界は線形和によって算出できるため，高速に計算することができる．それにもかかわらず，下界の値は元の DTW 距離の約 7,8 割の値となっており，下界が有効に働いていると考えられる．

5.2.2 枝刈りされた軌跡の数

下界によって枝刈りされた軌跡が多ければ多いほど，実行速度は速くなる．よって，枝刈りされた軌跡の数を元の軌跡の数で割った値 ($PruningPower$) は評価指標として有用である．

$$PruningPower = \frac{\text{枝刈りされた軌跡の数}}{\text{元の軌跡の数}} \tag{18}$$

10 件の入力に対する PruningPower を表 1 と表 2 に示した．ハリケーンデータは全軌跡データの約 83%〜96%を，人工データは約 63%〜約 93%を枝刈りしており，提案アルゴリズムが効果的に作用していると考えられる．また，$Tightness$ が大きいものほど $PruningPower$ が大きくなるという傾向が見られた．

5.2.3 類似検索の実行速度

最後に，実行速度について述べる．10 件の入力に対する通常の実行速度と下界を用いた場合の実行速度を表 1 と表 2 に示す．ハリケーンデータは約 5.1 倍〜17 倍，人工データは約 2.8 倍〜13 倍の速度向上が見られた．ハリケーンデータにおいて枝刈りされた軌跡の数が人工データと比べて多かったため，ハリケーンデータにおける実行速度が人工データにおける実行速度と比べて大きく向上していると考えられる．

6. おわりに

我々は動的属性を考慮した軌跡の類似検索のための枠組みを提案した．この枠組みによって，軌跡が持つ様々な属性（内的属性，外的属性）を利用した柔軟な検索が可能となった．加えて、検索速度を向上させるためのアルゴリズムを示し，実験によってそのアルゴリズムの有効性を示した．

今後の拡張の方向性として，大きく三つの方向性が考えられる．

（1） 距離関数の改良

（2） 部分軌跡を考慮した検索

（3） 前処理の高速化

まずは「距離関数の改良」について述べる．本研究では，拡張観測点間の距離を線形和の形で表現したが，データによってはその形が不適当な場合もあると考えられる．また，実世界での利用を想定した時に，重みの入力が利用者にとって不明瞭でコストの高い作業となり得るため，これを解決するためにも改良を行う必要がある．

続けて「部分軌跡を考慮した検索」について説明する．本研究では，軌跡全体を類似検索の対象としていたが，部分軌跡を扱わなければ有用な結果が得られない場合がしばしばあると考えられる．例えば，サッカーの選手の軌跡を入力とした際に，試合全体での動きが漠然と類似した選手よりも，ある一部の動きが強く類似した選手の方が分析対象として有用であると考えられる．よって，軌跡の一部分を扱えるように拡張することは有用であると考えられる．

最後に「前処理の高速化」について述べる．提案したアルゴリズムの中で，あらかじめ各属性に対する DTW 距離を求めるという前処理を行っている．入力軌跡の長さを k, 検索対象の軌跡の長さの平均を l, 属性の数を m，軌跡の数を n とした時，前処理の計算量は $\Theta(klmn^2)$ となり，軌跡の数が増えた際，計算量が爆発的に増加してしまう．実利用に耐えうるようにするためには，前処理計算の効率化を検討しなければならないと考えられる．

文　献

[1] Donald J Berndt and James Clifford. Using dynamic time warping to find patterns in time series. In *KDD workshop*, Vol. 10, pp. 359–370, 1994.

[2] Maike Buchin, Anne Driemel, Marc van Kreveld, and Vera Sacristán. An algorithmic framework for segmenting trajectories based on spatio-temporal criteria. In *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pp. 202–211, 2010.

[3] Lei Chen, M Tamer Özsu, and Vincent Oria. Robust and fast similarity search for moving object trajectories. In *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, pp. 491–502, 2005.

[4] Xudong Gong, Yan Xiong, Wenchao Huang, Lei Chen, Qiwei Lu, and Yiqing Hu. Fast similarity search of multidimensional time series via segment rotation. In *Proceedings of the Database Systems for Advanced Applications*, pp. 108–124, 2015.

[5] Joachim Gudmundsson and Thomas Wolle. Towards automated football analysis: Algorithms and data structures. In *Proceedings of the 10th Australasian conference on Mathematics and Computers in Sport*, 2010.

[6] Eamonn Keogh and Chotirat Ann Ratanamahatana. Exact indexing of dynamic time warping. *Knowledge and information systems*, Vol. 7, No. 3, pp. 358–386, 2005.

[7] Jae-Gil Lee, Jiawei Han, and Xiaolei Li. Trajectory outlier detection: A partition-and-detect framework. In *Proceedings of the 24th International Conference on Data Engineering*, pp. 140–149, 2008.

[8] Jae-Gil Lee, Jiawei Han, and Kyu-Young Whang. Trajectory clustering: a partition-and-group framework. In *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, pp. 593–604, 2007.

[9] Nikos Pelekis, Ioannis Kopanakis, Gerasimos Marketos, Irene Ntoutsi, Gennady Andrienko, and Yannis Theodoridis. Similarity search in trajectory databases. In *Proceedings of the 14th International Symposium on Temporal Representation and Reasoning*, pp. 129–140, 2007.

[10] Yasushi Sakurai, Masatoshi Yoshikawa, and Christos Faloutsos. Ftw: fast similarity search under the time warping distance. In *Proceedings of the 24th ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pp. 326–337, 2005.

[11] Michail Vlachos, George Kollios, and Dimitrios Gunopulos. Discovering similar multidimensional trajectories. In *Proceedings of the 18th International Conference on Data Engineering*, pp. 673–684, 2002.

[12] Chikashi Yajima, Yoshihiro Nakanishi, and Katsumi Tanaka. Querying video data by spatio-temporal relationships of moving object traces. In *Visual and Multimedia Information Management*, pp. 357–371. Springer, 2002.

[13] Kai Zheng, Shuo Shang, Nicholas Jing Yuan, and Yi Yang. Towards efficient search for activity trajectories. In *Proceedings of the 29th International Conference on Data Engineering*, pp. 230–241, 2013.

[14] Yu Zheng. Trajectory data mining: an overview. *ACM Transactions on Intelligent Systems and Technology (TIST)*, Vol. 6, No. 3, p. 29, 2015.

[15] Yu Zheng, Lizhu Zhang, Xing Xie, and Wei-Ying Ma. Mining interesting locations and travel sequences from gps trajectories. In *Proceedings of the 18th international conference on World wide web*, pp. 791–800, 2009.

[16] 本田崇人, 松原靖子, 根山亮, 櫻井保志. 車両走行センサデータからの自動パターン検出. 第 8 回 Web とデータベースに関するフォーラム論文集, 2015.