

道路ネットワークを用いた GPS 軌跡の特徴点を保持する圧縮アルゴリズムの提案

原木 司^{†,a}

廣田 雅春^{‡,b}

横山 昌平^{‡,c}

石川 博^{‡,d}

[†] 静岡大学大学院情報学研究科 [‡] 静岡大学創造科学技術大学院 ^{‡‡} 静岡大学情報学部情報科学科

a) *gs11039@s.inf.shizuoka.ac.jp* b) *dgs11538@s.inf.shizuoka.ac.jp* c) *yokoyama@inf.shizuoka.ac.jp*

d) *ishikawa@inf.shizuoka.ac.jp*

概要 GPS ログのデータサイズの増加に伴い、携帯端末からサーバへのデータ転送量の増加が課題となっている。また、GPS ログに対する分析の処理速度は、GPS ログのデータサイズが増加するにつれ低下するという課題がある。これらの課題に対する1つの解決方法としてGPS ログの圧縮手法があげられる。しかし、従来のGPS ログ圧縮手法では、GPS ログ内のユーザの移動速度の情報、GPS ログの軌跡形状、およびGPS ログの移動方向の情報などのGPS ログの特徴点を保持することが難しいという課題がある。そこで、本論文では、Opening Window Compression Algorithm, CriticalPoint Algorithm, および StayPoint の3つの手法と道路ネットワークの情報を用いることで、GPS ログの特徴点を保持しながら、圧縮率を向上させるGPS ログのオンライン圧縮アルゴリズムについて検討する。本手法について、Speed Error, SED Error, Compression Rate, および実行時間に関する評価を行い、従来の手法よりも効果的にGPS ログの圧縮を行えていることを示す。

キーワード データ圧縮, オンラインアルゴリズム, GPS ログ, 道路ネットワーク

1 はじめに

近年、スマートフォンのような手軽にGPS データが測位可能な携帯端末が普及している。それに伴い、それらの携帯端末を用いることで、ユーザの行動を記録したGPS ログが容易に取得可能になった。それにより、GPS ログを利用したWeb サービスが増加している。そのため、サーバ上でデータを利用する事が増加している。

GPS ログを利用したサービスなどを利用するためには、携帯端末からサーバにGPS ログを送信する必要がある。しかし、GPS ログのデータサイズが増加するに伴い、携帯端末からサーバへのデータ転送量の増加が課題の1つとしてあげられる。また、処理するGPS ログのデータサイズが増加するにつれ、GPS ログに対する分析の処理速度が低下するという課題がある。これらの課題に対する1つの解決方法として、GPS ログの圧縮に関する研究がある[1][2][3][4][11]。GPS ログを圧縮することにより、携帯端末からサーバへのGPS ログの転送量を少なくすることが可能となる。また、適切に圧縮されたGPS ログを用いることで、分析の処理量を減らすことが可能となる。

従来のGPS ログ圧縮手法では、GPS ログ内のユーザの移動速度の情報、GPS ログの軌跡形状、およびGPS ログの移動方向の情報を保持することが難しいという課題がある[10]。本手法では、GPS ログ内のユーザの移動速度の情報、GPS ログの軌跡形状、およびGPS ログの移動方向の情報をGPS ログの特徴点と呼ぶ。例え

ば、GPS ログからユーザの移動手段を推定する研究では、GPS ログからユーザの移動速度の平均値や、移動速度の最大値、最小値を求め、移動手段の推定に利用している[8][9]。また、観光地を移動したユーザのGPS ログの分析においては、ユーザがどの地点に留まっていたのかを分析するため、GPS ログの中からユーザの滞在地点を抽出する処理を行う。ユーザの滞在地点を抽出するためには、GPS ログの軌跡情報とユーザの移動速度の情報が利用される[5]。このように、GPS ログの特徴点は様々な処理に利用されている。しかし、GPS ログの圧縮によって、GPS ログの特徴点が失われたGPS ログを処理に利用した場合、圧縮前のGPS ログを利用した分析結果よりも分析の精度が低下する可能性がある。そのため、GPS ログを圧縮する手法において、GPS ログの特徴を出来る限り保持することが可能な圧縮アルゴリズムは、GPS ログを用いる研究に対して非常に重要であると考えられる。

そこで、本研究では、道路ネットワーク上を移動したGPS ログを対象とし、Web アプリケーションなどでの利用に適した、処理速度が速く、軌跡の特徴点を保持するようなGPS ログの圧縮アルゴリズムを提案する。

2 関連研究

GPS ログの圧縮手法は、道路ネットワークを利用しない手法と、道路ネットワークを利用した手法の2種類のアプローチが挙げられる。

道路ネットワークを利用しないGPS ログの圧縮手法として、Douglas らのDouglas-Peucker Algorithm(DP

Algorithm)[2] や、それを改良した Meratnia らの top-down time-ratio algorithm(TD-TR Algorithm)[1] などのアルゴリズムがあげられる。これらの手法では、GPS ログの形状に着目して圧縮を行う。DP Algorithm や、TD-TR Algorithm では、GPS ログの始点と終点をつないだ直線と GPS ログの各点との Synchronized Euclidean Distance (SED)[1][2] を計算し、SED が閾値以上の GPS の測位点を圧縮後に抽出する測位点としている。その点を新たな始点として GPS ログの終点と直線でつなぐ。その直線と GPS ログのそれぞれの測位点との SED を計算し、圧縮点を探すという処理を閾値以上の GPS 測位点なくなるまで繰り返す。

道路ネットワークを利用した GPS ログの圧縮手法として、Schmid ら [3] の手法がある。Schmid らは、都市部の道路上を移動するユーザの GPS ログを対象とした圧縮手法を提案している。Schmid らの手法では、まず Map-Matching を行い、その結果を道路ネットワークと交通ネットワークと比較し、ユーザが移動している道路を特定する。次に、ユーザが移動した経路の交差点やバス停の位置を GPS ログの圧縮点として記録する。Schmid らの手法では、GPS 測位点が 115 点である GPS ログを圧縮することによって、6 本の道路に圧縮することが可能である。このときの圧縮率は 94.78% となっている。

本手法は、GPS ログの形状だけでなく、速度と角度の変化に着目した圧縮を行う点において、DP Algorithm や TD-TR Algorithm と異なっている。また、多くの手法では圧縮前に存在した静止状態を測位した GPS 点を圧縮後の GPS ログで保持できていないが、本手法では、静止状態を測位した GPS ログの特徴点を保持することが可能となっている。

3 提案手法

本手法では、GPS ログの特徴点を保持するために、形状に着目した圧縮手法である Opening Window Compression Algorithm(OW) [1]、速度と移動方向の変化に着目した圧縮手法である CriticalPoint Algorithm[4]、ユーザが滞在していた地点を抽出する StayPoint[5] の 3 つの手法を用いる。また、本手法では、ユーザが道路上を移動しながら測位した GPS ログを対象としている。GPS ログに対して Map-Matching を行い、ユーザの移動している道路を推定する。Map-Matching の結果に基づいて、OW を適用することで、ユーザの移動先を予測した結果を用いて、GPS ログの圧縮が可能となり、GPS の誤差の影響を小さくする。

3.1 Map-Matching

本手法では、Brakatsoulas ら [6] の Map-Matching の手法を利用する。Brakatsoulas らの手法は、オフライン

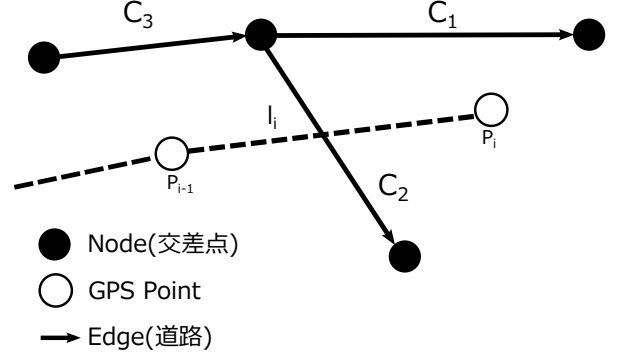


図 1 Map-Matching

データを対象としている。そのため、本手法では、この手法をオンラインデータに適用可能に変更する。Brakatsoulas らの手法を図 1 に示す。図 1 において、 C_1 、 C_2 、および C_3 は、道路の始点ノードから終点ノードへのベクトルを表す。道路ネットワークにおける、ノードとは交差点のことである。Brakatsoulas らの手法では、道路ネットワークから GPS ログの最初の GPS 測位点 P_{i-1} にもっとも近い道路 C_3 を求める。そして、GPS 測位点 P_i に対して、移動先の候補となる道路 C_1 、 C_2 を求め、(1)(2)(3) を用いて評価を行う。

$$S_d(P_i, C_j) = \mu_d - a \cdot \text{Dist}(P_i, C_j)^{n_d} \quad (1)$$

$$S_\alpha(P_i, C_j) = \mu_\alpha \cdot \cos(\alpha_{i,j})^{n_\alpha} \quad (2)$$

$$S = S_\alpha + S_d \quad (3)$$

μ_d 、 μ_α 、 n_d 、および n_α は、定数である。 P_i は GPS で測位された点の位置を表す。 C_j は移動先の候補の道路を表す。 $\text{Dist}(P_i, C_j)$ は、 P_i と C_j の地理的な距離を表す。 $\alpha_{i,j}$ は、 l_i と C_j との角度の差を表す。 l_i は、 P_{i-1} から P_i へのベクトルを表す。 $\cos(\alpha_{i,j})$ は、 $\alpha_{i,j}$ の余弦の値を表す。(1) は、 P_i と C_j との距離を評価する指標である。(2) は、 P_i と C_j との角度を評価する指標である。(3) の値が大きいくほど、候補とされている道路がユーザの移動先に適していることを表す。

Brakatsoulas らは、現在地点よりも時刻が後の GPS 測位点を事前に予測し、候補となる道路を求めることによって、Map-Matching の精度を向上させている。しかし、本手法では、GPS 測位点の予測をおこなうことによって、オフラインデータを対象としていた Brakatsoulas らの手法をオンラインデータに適用する。

3.2 Opening Window Compression Algorithms

はじめに、OW で利用される SED について述べる。SED は、圧縮後の GPS ログの点 P'_i と圧縮前の GPS ログの点 P_i との距離をを時間情報を考慮して求めたもの

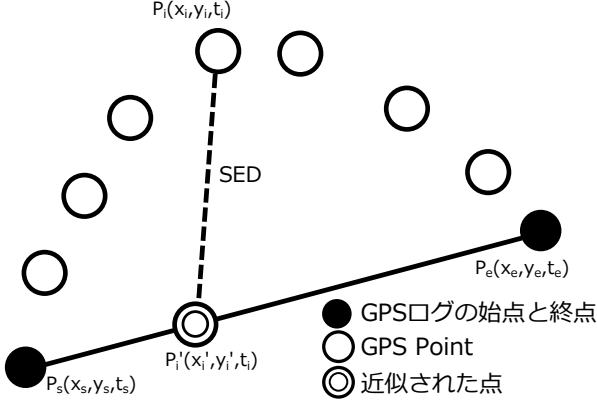


図 2 Synchronized Euclidean Distance

である。本手法では、SED を求めるため、Meratnia ら [1] の手法を用いる。SED の例を図 2 に示す。図 2 の P_i は GPS ログに含まれる軌跡の 1 点を表す。 x_i と y_i は P_i の座標、 t_i は P_i が取得された時間を表す。 P'_i は、 P_i を P_sP_e 上に近似した点である。 x'_i と y'_i は、 (4)(5) に基づいた (6)(7) によって求める。

$$\Delta e = t_e - t_s \quad (4)$$

$$\Delta i = t_i - t_s \quad (5)$$

$$x'_i = x_s + \frac{\Delta i}{\Delta e}(x_e - x_s) \quad (6)$$

$$y'_i = y_s + \frac{\Delta i}{\Delta e}(y_e - y_s) \quad (7)$$

OW は、DP Algorithm や TD-TR Algorithm に類似した GPS ログの圧縮アルゴリズムである。しかし、OW は、DP Algorithm や TD-TR Algorithm と異なり、オンラインアルゴリズムである。

OW は、GPS ログの最初の点を *anchor* とし、GPS ログの先頭から 3 つ目の点を *float* とする。 *anchor* を p_s 、 *float* を p_e として、それぞれの *anchor* と *float* 間の GPS の点に対して SED を計算する。 SED が閾値以上になる点が存在しない場合、 *float* を次に側位された GPS の点に変更し、 *anchor* と *float* 間の GPS の点それぞれに対して SED の計算を行う。 SED が閾値以上になる点が存在する場合、 SED が閾値以上になった GPS の点を圧縮点とする Normal Opening Window Algorithm(NOPW) という手法によって選択された圧縮点を次の *anchor* とする。そして、それぞれの *anchor* と *float* 間の GPS の点に対して SED の計算を行い、圧縮点を探索する処理を繰り返す。図 3 に、OW を用いて圧縮した GPS ログの例を示す。

本手法と OW では、 *anchor*、 *float* の選び方が異なる。本手法では、 *anchor* と *float* に Map-Matching の結果を用いる。本研究では、道路上を移動しているユー

ザを対象としているため、ユーザの移動している道路を推定することによって、ユーザの移動経路を予測することが可能となる。本手法では、3.1 節で行った Map-Matching の結果をユーザの移動した道路とする。そして、その道路の始点を *anchor*、終点を *float* とする。新たな GPS ログが測位されるたび、その GPS の測位点に対して、それらの *anchor* と *float* を用いて SED を計算する。 SED の計算のために *float* に到着する時刻を計算する必要がある。本手法では、現在の GPS 測位点での速度に基づいて、同じ速度で *float* まで、直線で移動した場合の時間を計算し、それを *float* に到着する予想時刻とする。 SED が閾値以上になる点が存在しない場合は、新たな GPS ログの測位まで待機する。 SED が閾値以上になる点が存在する場合は、GPS ログの点を圧縮点とする。ユーザの移動している道路が変更されたときに、 *anchor* と *float* を次の道路の始点と終点に変更する。

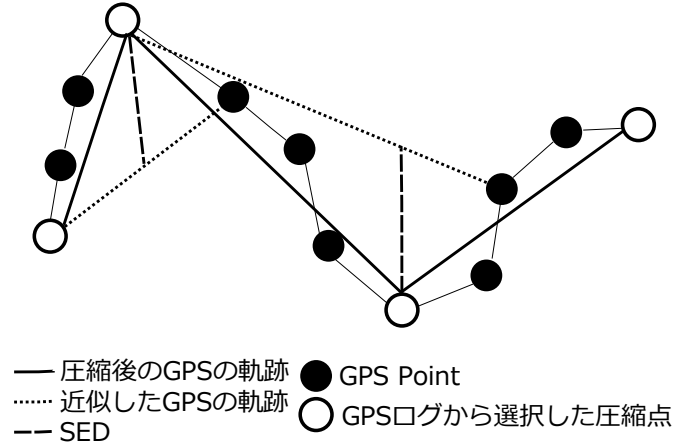


図 3 Opening Window Compression Algorithms

3.3 StayPoint

StayPoint とは、GPS ログの中で、ユーザが一定の範囲内で一定の時間以上、滞在していた地点を表す。 Stay-Point を求めるために、(8)(9)(10)(11) を満たす GPS ログを求める。

$$m < i \leq n \quad (8)$$

$$Dist(p_m, p_i) \leq D_r \quad (9)$$

$$Dist(p_m, p_{n+1}) > D_r \quad (10)$$

$$Int(p_m, p_n) \geq T_r \quad (11)$$

p_i は、GPS ログに含まれる軌跡の 1 点を表す。 $Dist(p_m, p_i)$ は、 p_m と p_i の 2 点間のヒュベニの公式に基づいた距離を表す。 D_r は、距離の閾値を表す。 $Int(p_m, p_n)$ は、 p_m と p_n のタイムスタンプの差を表す。 T_r は、時間の閾値

を表す。これらの式を満たす GPS ログから StayPoint を算出する。StayPoint を s とすると、 $s = (x, y, t_a, t_l)$ で表される。 $s.x$ と $s.y$ は、 s の座標を表す。 $s.t_a$ は、ユーザが StayPoint に滞在を開始した時のタイムスタンプを表す。 $s.t_l$ は、ユーザが StayPoint の滞在を終えた時のタイムスタンプを表す。 $s.x$, $s.y$, $s.t_a$, および $s.t_l$ は、(12)(13)(14)(15) を用いて求める。

$$s.x = \sum_{i=m}^n p_i.x / |P| \quad (12)$$

$$s.y = \sum_{i=m}^n p_i.y / |P| \quad (13)$$

$$s.t_a = p_m.t_m \quad (14)$$

$$s.t_l = p_n.t_n \quad (15)$$

本手法では、StayPoint を用いてユーザが滞在していた地点を求める。そして、StayPoint を圧縮点とする。GPS ログ内から StayPoint を抽出し、圧縮点とすることで、圧縮前の GPS ログの静止状態という特徴を圧縮後も保持する。本手法では、 s を $sa = (x, y, t_a)$, $sl = (x, y, t_l)$ の2点で表す。 s を sa と sl の2点で表すことによって、従来の GPS ログと同じフォーマットで記述する。

3.4 CriticalPoint Algorithm

CriticalPoint Algorithm とは、オンラインアルゴリズムの GPS ログ圧縮手法である。GPS ログの連続する2点において、それらの角度が大きい場合や、記録される時間の差が小さい場合（ユーザの移動速度が速い）の2つを表す。 $SpeedDif(p_i, p_{i-1}) > Sr$, もしくは $AngleDif(p_i, p_{i-1}) > Ar$ を満たす p_i を CriticalPoint とする。 $SpeedDif(p_i, p_{i-1})$ は、 p_i と p_{i-1} との速度差を表す。 $AngleDif(p_i, p_{i-1})$ は、 p_i と p_{i-1} との角度差を表す。 Sr は、速度の閾値を表す。 Ar は、角度の閾値を表す。本手法では、CriticalPoint も圧縮点の1つとする。

3.5 圧縮点の統合処理

本手法では、OW, StayPoint および CriticalPoint の3つの手法を用いて圧縮点を求める。3つの手法で求められた圧縮点を、圧縮後の GPS ログとして統合する。このとき、重複する圧縮点を削除する。また、StayPoint や、CriticalPoint による圧縮点と、OW による圧縮点を比較すると、OW による圧縮点は、形状の特徴のみを保持しているが、StayPoint や、CriticalPoint による圧縮点は、形状の特徴に加えて、速度などの他の特徴も保持している。そのため、OW による圧縮点から一定の距離以内に、StayPoint や、CriticalPoint が存在する場合、その OW による圧縮点は削除する。

4 実験

本手法による GPS ログの圧縮結果が、圧縮率を維持しながら、GPS ログの特徴を保持しているかを評価する。実験データとして、SIGSPATIAL GIS CUP2012[7] で使われた、10 個の GPS ログとワシントン州の道路ネットワークを利用する。10 個の GPS ログのサイズの平均値は、約 50KB である。GPS 軌跡の点の数の平均値は、約 1443.6 点である。GPS の測位間隔は、1 秒である。

評価指標として、SED Error, Speed Error, Compression Rate, および実行時間の4つを用いる。SED Error とは、圧縮前の GPS ログの形状と圧縮後の GPS ログの形状の差を表す指標である。SED Error は、圧縮前の GPS ログと圧縮後の GPS ログから算出した、SED の中央値と平均値を用いる。Speed Error とは、圧縮前の GPS ログの速度と圧縮後の GPS ログの速度の差を表す指標である。Compression Rate とは、GPS ログの圧縮率を表す指標である。Compression Rate は、(16) を用いて計算する。

$$CompressionRate = 1 - \frac{CP}{OP} \quad (16)$$

CP とは、圧縮後の GPS ログの GPS 測位点の数である。 OP とは、圧縮前の GPS ログの GPS 測位点の数である。実行時間とは、1 点の GPS ログを処理するために必要な時間を表す指標である。実行時間は、1 点の GPS ログを処理した際の処理時間の平均値とする。

それぞれの実験データに対して、本手法を適用し、GPS ログの圧縮を行う。本手法のパラメータとして、SED の閾値を用いる。これは、OW において圧縮点を求める際に利用されるパラメータである。そして、SED の閾値は、30~200 まで変化させる。また、それぞれの実験データに対して、Schmid らの手法を用いて、GPS ログの圧縮を行う。本手法を用いて圧縮した GPS ログと Schmid らの手法を用いてそれぞれの GPS ログに対して、評価指標を計算し、評価する。

4.1 実験結果と考察

表 1 実行時間による提案手法と STC の比較

	実行時間 [ms]
Proposed Method	0.221
STC	0.211

実験結果を表 1, 図 4, 図 5, および図 6 に示す。これらの図において、STC とは、Schmid らの手法を示す。ここで、STC は、提案手法のように SED に関するパラメータが存在しないため、それらの図においても、1 つの値となっている。また、ProposedMethod は提案手法

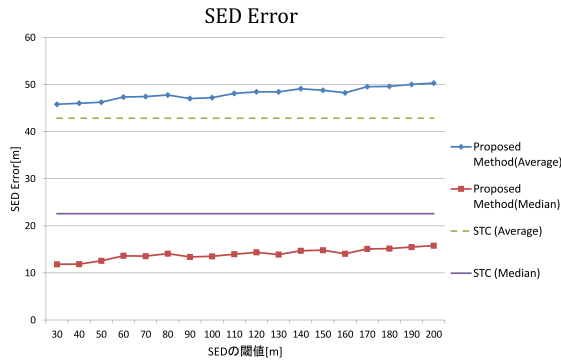


図 4 SED Error による提案手法と STC の比較

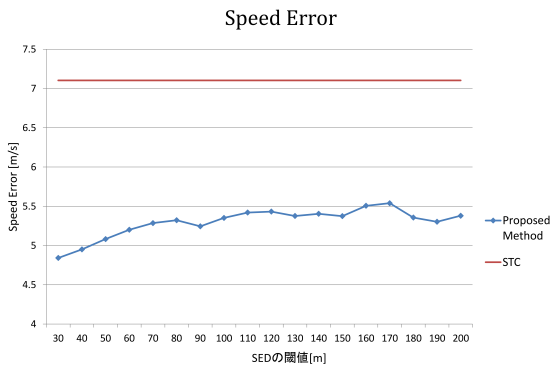


図 5 Speed Error による提案手法と STC の比較

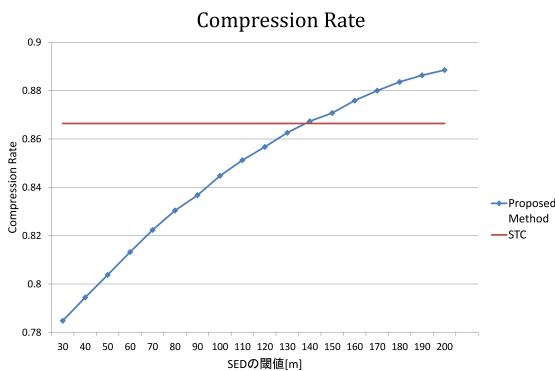


図 6 Compression Rate による提案手法と STC の比較

を表す. 図 4, 図 5 および図 6 の横軸は, SED の閾値を表す.

表 1 より, 本手法の実行時間と, Schmid らの手法の実行時間を比較すると, 本手法の方が処理速度が遅いが,, それらの差は 0.1ms であり, 本手法は GPS ログを処理するために十分な処理速度であると考えられる.

図 4 に, 提案手法と STC の SED Error の結果を示す. 図 4 において, SED Error の平均値に関して, 本手法は STC に比べて, SED Error の値が大きい. しかし, SED Error の中央値では, 本手法は STC に比べて, SED Error の値が小さい. これらのことから, 本手法は STC に比べて, SED Error の値が極端に大きい値となる場合もあるが, ほとんどの GPS ログの圧縮点に関して, STC よりも SED Error の値が小さいと考えられる. そのため, STC に比べて, 本手法は, 形状の特徴を保持していると考えられる.

本手法が, 極端に大きな SED Error の値をとる原因として, 本手法の OW で用いる *float* の到着時刻の算出方法があげられる. 本手法では, GPS 測位点から *float* まで, ユーザが直線で移動した場合の到着時刻を計算しているが, 道路は常に直線ではなく, 曲線が存在する. そのため, *float* の予想到着時刻と実際の到着時刻に差が生まれる. また, 道路上には渋滞や信号待ちなど停車や減速を必要とする場面が存在し, 一定の速度で, *float* まで移動できるとは限らない. そのため, 現在の速度を用いて, *float* への到着時刻を予想すると大きな誤差が生まれてしまう場合が存在する. それにより, 適切な SED の値と大きく異なる SED の値が算出される場合がある. 結果として, 適切な圧縮点ではない GPS 測位点が, OW の圧縮点として選択されるため, 極端に大きな SED Error の値が存在すると考えられる. もう 1 つの原因として, GPS ログに対する Map-Matching の結果の誤りによって, 適切な SED が求められない場合があげられる. 本手法では, オンラインアルゴリズムの Map-Matching を用いているため, オフラインアルゴリズムに基づいた Map-Matching に比べて, Map-Matching の精度は低い. そのため, ユーザが通過していない道路が, Map-Matching の結果として選択されることで, 適切な *float* が選択されない場合がある. それにより, OW において適切な圧縮点を選択できず, SED Error の値が適切な値より大きくなると考えられる.

図 5 に, 提案手法と STC の Speed Error の結果を示す. 図 5 において, 全ての SED の閾値において, 本手法は Speed Error の値が STC よりも下回っている. また, SED の閾値を変化させた場合でも, Speed Error の値はほとんど変化していない. これは, 本手法が, SED の閾値による影響をあまり受けず, STC よりも速度の

特徴を保持可能であることを表している。

図 6 に、提案手法と STC の圧縮率の結果を示す。本手法は、SED の閾値を変化させることによって、圧縮率を変化させることが可能である。本手法の圧縮率を、STC の圧縮率以上に变化させた場合 (SED の閾値の値が 140 以上の場合) において、STC と SED Error と Speed Error の値を比較する。本手法の圧縮率が STC の圧縮率を上回った場合でも、速度と形状の特徴を保持可能な事が図 4, 図 5 によりわかる。このことから、STC に比べて、本手法は圧縮率と、速度と形状の特徴保持において優れていると考えられる。

図 4, 図 5, および図 6 から、本手法は、SED の閾値値の増加に従って、SED Error と Speed Error の値は増加し、圧縮率は増加する。これは、圧縮率の増加に伴って、速度と形状の特徴の保持率が低下することを表している。そのため、本手法を GPS ログの圧縮に対して利用する場合、圧縮率と特徴の保持についてはトレードオフであることがわかる。また、本手法の特徴として、SED の閾値によって圧縮率を変化させることが可能であることがあげられる。そのため、GPS ログの速度と形状を利用するような処理を行う際に、その処理に必要な特徴の値に従って、圧縮率を変化させることが可能であるため、GPS ログの圧縮率が高く、圧縮された GPS ログに対する処理の性能をほとんど低下させないような GPS ログの圧縮が可能である。

5 まとめと今後の課題

本論文では、GPS ログの特徴点を保持するために、形状に着目した圧縮手法である Opening Window Compression Algorithm(OW), 速度と移動方向の変化に着目した圧縮手法である CriticalPoint Algorithm, ユーザが滞在していた地点を抽出する StayPoint の 3 つの手法を用いたオンライン圧縮アルゴリズムを提案した。提案手法と STC に対して、評価実験をおこない STC に比べて提案手法が、GPS ログの特徴点を保持していることを確認した。実行時間に関しても、オンラインアルゴリズムとして十分な実行時間であることを確認した。また、SED の閾値を増加させるに伴い、圧縮率は向上するが、速度や形状の特徴を保持できなくなるという傾向が確認された。

今後の課題として、提案手法の中で、ユーザの移動速度や、Map-Matching の精度により、SED の精度が低下する可能性があるため、それについての対処があげられる。

謝辞

本研究の一部は科学研究費補助金 (課題番号 22700094) の助成による。

参考文献

- [1] Meratnia, N. and de By, R. A.: Spatiotemporal compression techniques for moving point objects, Proc. of the Extending Database Technology, pp. 765–782, 2004.
- [2] Douglas, D. H. and Peucker, T. K.: Algorithms for the reduction of the number of points required to represent a line of its caricature, The Canadian Cartographer, Vol. 10, Number 2, pp. 112–122, 1973.
- [3] Schmid, F., Richter, K. F. and Laube, P.: Semantic trajectory compression, Proc. 11th International Symposium, SSTD, Vol.5644, pp. 411–416, 2009.
- [4] Barbeau, S., Labrador, M. A., Perez, A., Winters, P. Georggi, N., Aguilar, D. and Perez, R.: Dynamic management of real-time location data on GPS-enabled mobile phones, The Second International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies, pp. 343–348, 2008.
- [5] Zheng, V. W., Zheng, Yu., Xie, X. and Yang, Q.: Collaborative location and activity recommendations with GPS history data, Proc. of the 19th World Wide Web International Conference, pp. 1029–1038, 2010.
- [6] Brakatsoulas, S., Pfooser, D., Salas, R. and Wenk, C.: On map-matching vehicle tracking data, Proc. of the 31st Very Large Data Bases International Conference, pp. 853–864, 2005.
- [7] ACM SIGSPATIAL Cup 2012.: <http://depts.washington.edu/giscup/>.
- [8] Stenneth, L., Wolfson, O., Yu, P. S. and Xu, B.: Transportation mode detection using mobile phones and GIS information, Proc. of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, pp. 54–63, 2011.
- [9] Zheng, Y., Liu, L., Wang, L. and Xie, X.: Learning transportation mode from raw gps data for geographic applications on the web, Proc. of the 17th World Wide Web International Conference, pp. 247–256, 2008.
- [10] Muckell, J., Patil, V., Ping, F., Hwang, J., Lawson, C. T. and Ravi, S. S.: SQUISH: an online approach for GPS trajectory compression, Proc. of the 2nd International Conference on Computing for Geospatial Research & Applications, pp. 1–8, 2011.
- [11] Chen, M., Xu, M. and Franti, P.: Compression of GPS trajectories, The Data Compression Conference, pp. 62–71, 2012.