

疎な GPS 測位情報を対象にした 測位精度と短時間滞在除去に基づく滞在地推定手法

岩田 紗瑛^{1,a)} 新田 知之² 高山 敏典² 柳澤 政生¹ 戸川 望^{1,b)}

概要：GPS 機能を持ったスマートフォンや GPS ロガーの普及により、ユーザの位置情報の取得が容易となった。それに伴いユーザの行動を調査し、ユーザの趣味嗜好の推定や、過去に訪れた場所の推定、将来訪れる場所を予測する研究が注目されている。これらを実現するためには、GPS 測位に伴う測位誤差を除去し、正確な滞在地の推定が必要である。既存の滞在地推定手法では、GPS 測位間隔が非常に短く、測位端末のバッテリー消費が大きいという問題がある。測位間隔を長くすると、位置情報の数が少なくなる上、測位間隔が短い場合と同様にランダムな測位誤差が含まれるので、正確な滞在地推定が難しくなる。本稿では、疎な GPS 測位点の系列から、ユーザが滞在した場所の推定手法を提案する。提案手法は、測位点をクラスタリングする際、(a) 測位精度を用いたクラスタの有効半径を設定、これに基づきクラスタを結合し、(b) 移動か測位誤差が判別不能な短時間のクラスタを一旦削除し、再度クラスタを結合する操作を導入する。その結果、測位間隔が長い場合でも、高精度な滞在地推定を実現する。提案手法をあるユーザの 19 日分の測位点系列に適用したところ、既存手法と比較して、推定滞在地と真の滞在地との総距離誤差を 50%以上小さくすることを確認した。

1. はじめに

GPS 機能を持ったスマートフォンや GPS ロガーの普及により、ユーザの GPS 測位情報の取得が容易となった。それに伴い、GPS 測位情報を分析してユーザの行動を調査し、ユーザの趣味嗜好の推定や、過去に訪れた場所の推定、将来訪れる場所を予測する研究が注目されている [10]。行動調査にはユーザがいつどこに滞在したか正確に推定することが重要である。しかし、測位時の GPS 測位誤差を完全に除去することは不可能であり、しかも、GPS 測位情報には必ずランダムな測位誤差が含まれる [10]。GPS 測位誤差は、理想的な屋外条件で 10m 程度、都市部の屋内測位で 100m 近くになり、WiFi や基地局による測位では距離誤差が 600m に達するという報告がある [14–16]。GPS 測位情報に含まれる測位誤差をどのように判別し、正確な滞在地を推定できるかが大きな問題となる。

これまでの滞在地推定手法は、大きく分けて、密な測位情報を用いる手法と、疎な測位情報を用いる手法がある。密な測位情報を用いる手法の多くは、カーネル密度

推定 [12]、k-means 法 [9]、DBSCAN クラスタリング [8]、Mean-shift クラスタリング [7] を基本とする。矢部らは、1 秒ごとに GPS 測位された GPS 測位情報のカーネル密度を算出し、滞在地を推定している [2]。Ashbrook らは、1 秒ごとに GPS 測位された GPS 測位情報に k-means 法を適用し、滞在地を推定している [6]。西田らは 3 秒ごとに GPS 測位された GPS 測位情報に時間的に拡張した Mean-shift 法を適用し、滞在地を推定している [5, 10]。藤田ら、西野らはそれぞれ 5 秒、10 秒間隔で GPS 測位された GPS 測位情報に DBSCAN クラスタリングを適用し、滞在地を推定している [1, 4]。以上の研究 [1, 2, 4, 5, 10] は共通して、GPS 測位間隔が 1~10 秒と非常に短く、GPS 測位回数が非常に多いため、文献 [1, 2, 4] ではユーザに GPS ロガーを配布し、文献 [5, 10] ではユーザに GPS 測位専用の携帯端末を配布し GPS 測位情報を得ることを想定している。これらの手法では、測位専用端末を常時携帯し、端末を頻繁に充電しなければならないため、ユーザにとって大きな負担となる。多くのユーザが容易に滞在地推定を利用するためには、ユーザが常時携帯している携帯端末の GPS 機能を利用するのが望ましいと言われている [3, 13]。携帯端末の GPS 機能を用いるには、携帯端末本来の電話やメール等の使用を妨げない範囲で GPS 測位しなければならない。1~10 秒間隔での GPS 測位では、携帯端末のバッテリー消費

¹ 早稲田大学

Waseda University

² ゼンリンデータコム

ZENRIN DataCom

a) sae.iwata@togawa.cs.waseda.ac.jp

b) togawa@togawa.cs.waseda.ac.jp

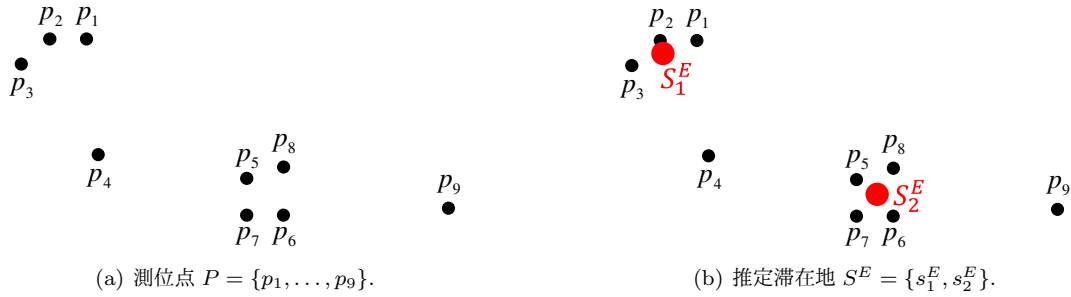


図 1 測位点集合から滞在地の推定.

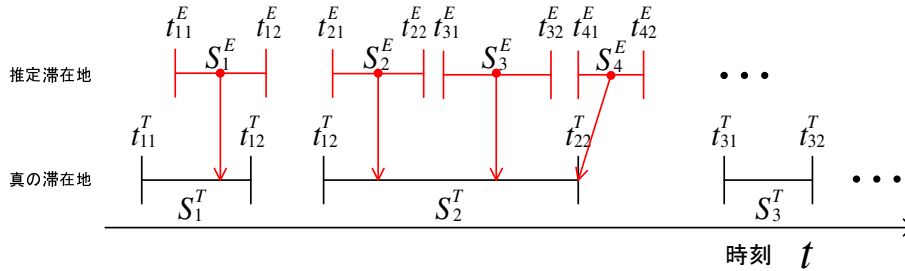


図 2 推定滞在地と対応する真の滞在地.

が大きすぎるという問題がある.

疎な測位情報を用いる手法として, GPS 測位端末にスマートフォンを使用し, GPS 測位間隔を 5 分とし滞在地を推定する手法がある [3, 13]. 山田らは各測位点に与えられる推定測位誤差から, 推定測位誤差の 2 倍を 1 辺に持つ正方形で測位点を表現し, 各正方形が重なった部分を滞在エリアと推定している. この手法は比較的大きな平面のエリア内にユーザが滞在したことを表現するだけで, 点として滞在地を推定することは難しい.

本稿ではユーザの携帯端末への負担が少ない, 疎な測位情報に注目して, 滞在地の推定を目指す. 測位間隔が短い場合, GPS 測位情報が多いため外れ値を測位誤差として除去しても十分なデータを保有することができる. しかし, 測位間隔が長い場合, GPS 測位情報の数が少ない上に, 測位間隔が短い場合と同様にランダムな測位誤差が含まれるので, 多くの既存手法のように外れ値を単純に除去するとデータ数が不足し, 正確な滞在地を推定することが難しくなる. いかにも外れ値を単純に除去せず正確な滞在地を推定するかが最大の問題となる.

本稿では, 疎な GPS 測位点の系列から, ユーザが滞在した場所の推定手法を提案する. 提案手法は, 測位点をクラスタリングする際, (a) 測位精度を用いたクラスタの有効半径を設定, これに基づきクラスタを結合し, (b) 移動か測位誤差か判別不能短時間のクラスタを一旦削除し, 再度クラスタを結合する操作を導入する. その結果, 測位間隔が長い場合でも, 精度の高い滞在地推定を実現する. 提案手法を用いた滞在地推定の実装と評価を行い, 提案手法の有効性を評価する.

本稿の貢献は以下の通りである.

- (1) 低頻度な測位による疎な位置情報に対して, (a) 測位精度を用いたクラスタの有効半径を設定, これに基づきクラスタを結合し, (b) 移動か測位誤差か判別不能短時間のクラスタを一旦削除し, 再度クラスタを結合する操作を提案することによって, より精度の高い滞在地推定を実現する.
- (2) 19 日間にわたるユーザの実測位点データに対して提案手法を適用したところ, 推定滞在地と真の滞在地の総距離誤差を既存手法の 50% 以下に削減できることを確認した. さらに, 各滞在地の滞在回数を計測したところ, 既存手法では滞在回数の誤差合計 149 回に対し, 提案手法は滞在回数の誤差合計 7 回となり, 正確な滞在地推定が実現された.

2. 滞在地推定問題

まず本章では滞在地推定問題を定義する. ユーザは各自所持する携帯端末で位置を GPS 測位するものとする. 測位点 p_i は, 測位時刻 t_i , 緯度 x_i , 経度 y_i , 測位精度 l_i ($l_i = 1, 2, 3$) によって与えられる. ここで測位精度 l_i とは, $l_i = 3$ のとき推定距離誤差 50m 以内, $l_i = 2$ で 50~300m, $l_i = 1$ で 300m 以上を意味する *1. 本稿で扱う測位精度 l_i は, 精度は高くないが, スマートフォンや従来の携帯電話など多くのユーザが利用できるものである [11]. 時系列順に並んだ N 個の測位点 p_i の系列 $P = \{p_1, p_2, \dots, p_N\}$ について, 各

*1 3.2.1 項で議論するように, 実際はより大きな測位誤差を持つ場合もある. 例えば, 測位精度 $l_i = 3$ であっても 500m 以上の大きな推定距離誤差を持つことや, 測位精度 $l_i = 1$ であっても 5m 程度の小さな推定距離誤差を持つこともある.

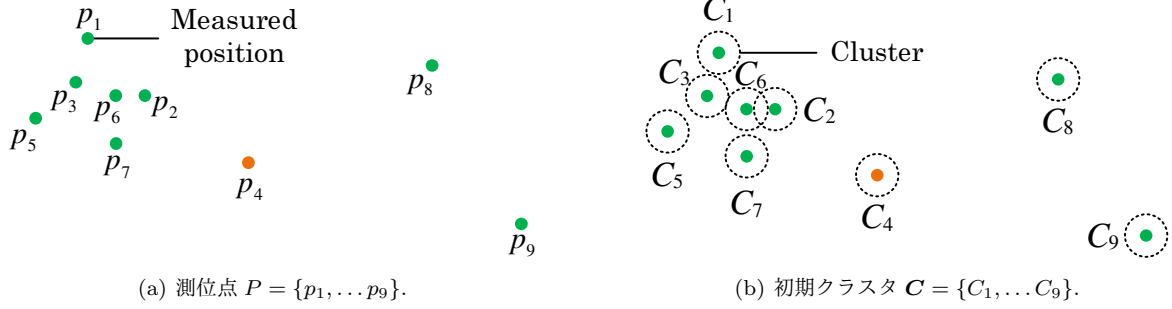


図 3 初期クラスタ生成の例.

測位点 p_i は 300 秒ごとに与えられる*2.

推定滞在地 s_j^E とは、滞在地の位置 (x_j^E, y_j^E) , 滞在開始時刻 t_{j1}^E , 滞在終了時刻 t_{j2}^E を持ち、滞在時間 $t_{j2}^E - t_{j1}^E = T_{stay}$ [秒] 以上とする. T_{stay} を最小滞在時間と呼ぶ. 時系列順に並んだ M 個の推定滞在地系列を $S^E = \{s_1^E, s_2^E, \dots, s_M^E\}$ と定義する. 同様に、真の滞在地 s_j^T とは、滞在地の位置 (x_j^T, y_j^T) , 滞在開始時刻 t_{j1}^T , 滞在終了時刻 t_{j2}^T を持ち、滞在時間 $t_{j2}^T - t_{j1}^T = T_{stay}$ [秒] 以上とする. 時系列順に並んだ L 個の真の滞在地系列を $S^T = \{s_1^T, s_2^T, \dots, s_L^T\}$ と定義する.

推定滞在地 s_j^E に対応する真の滞在地 $s_{j'}^T$ とは、 s_j^E の滞在開始時刻 t_{j1}^E と、滞在終了時刻 t_{j2}^E の中心の時刻 $t_{jc}^E = \frac{t_{j1}^E + t_{j2}^E}{2}$ が、真の滞在地の滞在時間に包含される ($t_{j'1}^T \leq t_{jc}^E \leq t_{j'2}^T$) か、時間的に最も近傍の推定滞在地とする. 図 2 に例を示す. 推定滞在地系列 $S^E = \{s_1^E, s_2^E, s_3^E, s_4^E\}$ と、真の滞在地系列 $S^T = \{s_1^T, s_2^T, s_3^T\}$ が時系列順に示されている. 推定滞在地 s_1^E には、真の滞在地 s_1^T が、 s_2^E, s_3^E, s_4^E には s_2^T が対応付けられている.

推定滞在地 s_j^E に対応する真の滞在地 $s_{j'}^T$ に対して、これらの滞在期間の推定距離誤差 d_j を式 (1) で定義する.

$$d_j = \sqrt{(x_j^E - x_{j'}^T)^2 + (y_j^E - y_{j'}^T)^2} \quad (1)$$

このとき、推定滞在地の総距離誤差 D_{sum} を式 (2) で定義する.

$$D_{sum} = \sum_{j=1}^M d_j \quad (2)$$

総距離誤差 D_{sum} が小さいほど、推定距離誤差と滞在回数の誤差が共に小さいため、滞在地推定の精度が高いことを意味する. なお、図 2 の s_3^T のように、真の滞在地に推定滞在地が対応付けられないことは稀であり、ここでは単純に無視する.

以上の準備のもと、滞在地推定問題を以下のように定義する.

定義 1 (滞在地推定問題) 滞在地推定問題とは、測位点 p_i の系列 P (図 1(a)) から、滞在地 s_j^E の系列 S^E (図 1(b)) を出力する問題である. このとき総距離誤差 D_{sum} の最小化を目的とする.

3. 疎な GPS 測位情報を用いた滞在地推定

GPS 測位点には必ずランダムな誤差が含まれ、図 5 のように推定測位誤差にも誤差が含まれる. そのため、既存の滞在地推定手法では、短い測位間隔で大量の測位点を集めた上で、mean-shift や DBSCAN 等のクラスタリングを用いて、ある一定の距離を外れた測位点を外れ値として除去し、滞在地を推定している. しかし、疎な GPS 測位点系列で同様に処理すると、もともとの測位点の数が少ないためデータ量が不足し、正確な滞在地の推定ができなくなる.

そこで本稿では滞在地を推定する際、まず測位点から初期クラスタを生成した後 (Step 1), (a) 測位精度を用いたクラスタの有効半径を設定、これに基づきクラスタを結合し (Step 2, Step 3), (b) 移動か測位誤差か判別不能な短時間のクラスタを一旦削除し、再度クラスタを結合する操作を導入する (Step 4). そして、最終的に出力されるクラスタを推定滞在地とする. これらの処理を導入することにより、推定滞在地と真の滞在地の距離誤差を小さくすると同時に、測位誤差によりクラスタが分裂し、真の滞在地 1 つに対し、対応する推定滞在地が複数散在することを防ぐ.

3.1 初期クラスタ生成 (Step 1)

今、図 3(a) のように、測位点系列 $P = \{p_1, p_2, \dots, p_9\}$ が与えられたとする. まず、各クラスタ C_i を測位点 p_i で初期化する (図 3(b)). 全ての時系列順に並んだクラスタの系列を $C = \{C_1, C_2, \dots, C_9\}$ とする.

各クラスタ C_i に測位精度 l_i に応じたマージン $margin$ を式 (3) のように割り当てる.

$$margin = \begin{cases} 450 \text{ m} & (l_i = 1) \\ 380 \text{ m} & (l_i = 2) \\ 70 \text{ m} & (l_i = 3) \end{cases} \quad (3)$$

式 (3) では、多くの測位点をクラスタに含めるよう、各測位精度に対する距離誤差の累積分布において、95%の距離

*2 4 章の評価実験では、測位端末のバッテリー残量 50%以上では 300 秒ごとに測位され、バッテリー残量 50%未満では 600 秒ごとに測位された測位点系列を用いる.

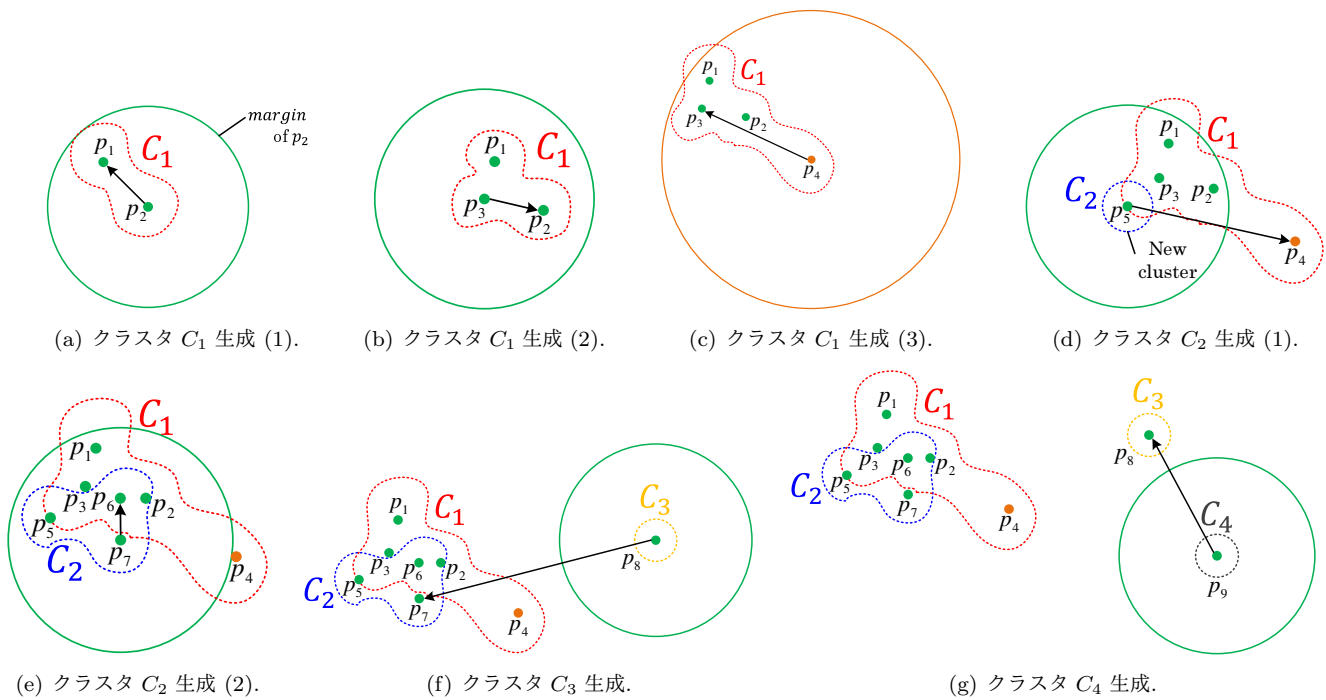


図 4 クラスタ生成の例.

表 1 各測位精度ごとの真の滞在地と各測位点までの距離の統計量.

	N	N'	mean [m]	min [m]	max [m]	var
$l_i = 3$	2482	2296	31.5	0.827	611	5809.5
$l_i = 2$	1833	348	65.1	0.0340	966	13064
$l_i = 1$	387	33	135	2.92	1738	31408

誤差をマージンとして採用した (図 5 参照. 詳細な議論は 3.2.1 項参照). これにより, 測位精度が良い測位点に対しては, マージンを狭く取ることによって滞在地推定の精度を高めることができ, 測位精度が悪い測位点に対しては, 広いマージンを設けることで測位精度が悪い測位点も滞在地に含めることができる.

次にクラスタ $C_i (i \geq 2)$ に対して, 前のクラスタ C_{i-1} に含まれる測位点系列 $P_{i-1}^C = \{p_m, p_{m+1} \dots, p_n\}$ のうち, 最も時刻の遅い測位点 p_n の位置が C_i のマージン以内の距離なら, C_{i-1} に C_i を結合し (図 4(a), 4(b), 4(c), 4(e)), そうでない場合, 結合しない (図 4(d), 4(f), 4(g)). 図 4(c) における測位精度が悪い測位点 p_4 は, マージンを広く持つので, 前のクラスタ C_1 に含めることができる.

最後に, 生成されたクラスタを時系列の順に C_1, C_2, \dots と再付番する. 以降, クラスタの結合, 削除により, クラスタの番号が変更された場合, 自動的にクラスタを時系列順に C_1, C_2, \dots と再付番する.

図 4 では, 初期クラスタ生成によって C_1, C_2, C_3, C_4 の 4 つのクラスタが生成される. ところが, 図 4 のように測位点 p_1, \dots, p_7 は同一の場所に滞在したものの, 測位誤差によって異なるクラスタが生成されたことが考えられる. そこで 3.2 節, 3.3 節では, 測位誤差により分裂したクラスタの結合方法を提案する.

3.2 測位精度を用いた有効半径の設定とこれに基づくクラスタ結合 (Step 2, Step 3)

本節では, まず各クラスタについて有効半径なる考えを導入する. 有効半径はクラスタの中心点を中心に, 半径がクラスタの推定距離誤差である円形の範囲である. これを導入し, 複数の初期クラスタを 1 つのクラスタに結合することで, 真の滞在時間に近い滞在時間を持つクラスタを生成することが期待できる.

3.2.1 各測位精度における測位点と真の滞在地との距離誤差の関係

クラスタの有効半径を設定するために, まず予備調査として, 測位精度と実際の距離誤差の関係を調査した. 図 5 に各測位点 p_i の測位精度 l_i における測位点 p_i と真の滞在地との距離の累積分布 (CDF) を示す. 横軸は距離誤差 [m], 縦軸が CDF である. 表 1 に統計量を示す. N は総測位点数, $mean$ は距離誤差の平均値, min は距離誤差の最小値, max は距離誤差の最大値, var は距離誤差の分散値である. N' は推定距離誤差の範囲内の測位点数で, 2 章の通り, 測位精度 $l_i = 3$ で推定距離誤差 50m 以内, 測位精度 $l_i = 2$ で推定距離誤差 50~300m, 測位精度 $l_i = 1$ で推定距離誤差 300m 以上を示す.

CDF の値が 95% に達する距離誤差は, 測位精度 $l_i = 3$ で 69.5m, 測位精度 $l_i = 2$ で 372.1m, 測位精度 $l_i = 1$ で

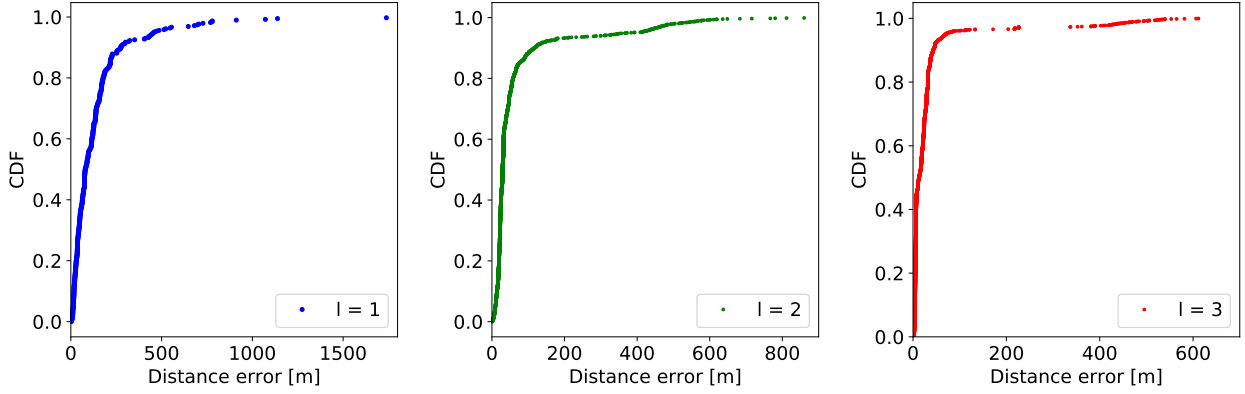


図 5 各測位精度における測位点と真の滞在地との距離誤差の CDF.

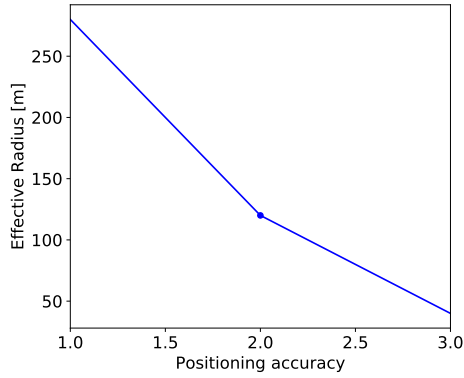


図 6 有効半径.

459.5m で、CDF の値が 90% に達する距離誤差は、測位精度 $l_i = 3$ で 43.0m、測位精度 $l_i = 2$ で 116.3m、測位精度 $l_i = 1$ で 276.6m だった。

3.1 節の初期クラスタ生成では、測位点単体の推定距離誤差を用いるため、距離誤差が大きくなりやすいので CDF 95% の距離誤差を用いた。3.3 節のクラスタ結合では、複数の測位点を持つクラスタを結合するため、測位点単体よりも測位誤差が小さいことが期待されるので、CDF 90% の距離誤差を用いた。

3.2.2 クラスタの有効半径

クラスタ C_i に含まれる測位点系列から、 C_i の中心点 (x_{c_i}, y_{c_i}) を算出することを考える。今、クラスタ C_i に含まれる n 個の測位点系列を時系列順に $P_i^C = \{p_{i_1}, \dots, p_{i_n}\}$ とする。各測位点は測位精度 l_k を持つため、測位精度ごとに重み w_k を考え、加重平均することで C_i の中心点を式 (4) のように求める。

$$x_{c_i} = \frac{\sum_{k=i_1}^{i_n} w_k x_k}{\sum_{k=i_1}^{i_n} w_k}, y_{c_i} = \frac{\sum_{k=i_1}^{i_n} w_k y_k}{\sum_{k=i_1}^{i_n} w_k} \quad (4)$$

本稿では w_k の値を、各測位精度 l_k に対する距離誤差の平均の逆比とする。表 1 より、測位精度 $l_k = 3$ のときの距離誤差の平均は 31.5m であり、測位精度 $l_k = 2$ のときの距離誤差の平均は 65.1m、測位精度 $l_k = 1$ のときの距離誤差の平均は 135m である。つまり測位精度 $l_k = 3$ のときの距

離誤差の平均を 1.0 としたとき、測位精度 $l_k = 2$ では 2.1、測位精度 $l_k = 1$ では 4.3 となるため、重みを式 (5) のように定義した。

$$w_k = \begin{cases} 1 & (l_k = 1) \\ 2 & (l_k = 2) \\ 4 & (l_k = 3) \end{cases} \quad (5)$$

クラスタ C_i に含まれる測位点の中で、最も時刻が早い測位点 p_{i_1} の時刻 t_{i_1} を C_i の滞在開始時刻、最も時刻が遅い測位点 p_{i_n} の時刻 t_{i_n} を C_i の滞在終了時刻とする。クラスタ C_i の滞在時間は、 $(t_{i_n} - t_{i_1})$ とする。クラスタ C_i の測位精度 l_{c_i} は、クラスタ内に含まれる測位点の測位精度の平均とする (式 (6))。

$$l_{c_i} = \frac{\sum_{k=i_1}^{i_n} l_k}{i_n} \quad (6)$$

このときクラスタの有効半径 (ER: Effective Radius) を次のように提案する。まず、クラスタ C_i の測位精度 l_{c_i} の値がちょうど $l_{c_i} = 1, 2, 3$ のとき、有効半径 ER を式 (7) で与える。

$$ER = \begin{cases} 280 \text{ m} & (l_{c_i} = 1) \\ 120 \text{ m} & (l_{c_i} = 2) \\ 40 \text{ m} & (l_{c_i} = 3) \end{cases} \quad (7)$$

これらの値は、各測位精度に対する測位誤差の累積分布 (図 5) において、90% の距離誤差を表す値となっている。ここでは、そのままの測位点よりも精度で重み付けしたクラスタの中心点の方が誤差が少ないと考えられるため、95% の距離誤差ではなく、90% の距離誤差を採用した。

l_{c_i} の値がちょうど 1, 2, 3 にならないとき、式 (8) のように 1 次関数で補間し、有効半径 ER を与える (図 6)。

$$ER = \begin{cases} -80 l_{c_i} + 280 & (1 \leq l_{c_i} < 2) \\ -160 l_{c_i} + 440 & (2 \leq l_{c_i} \leq 3) \end{cases} \quad (8)$$

有効半径は以後クラスタを結合する際の閾値となる。

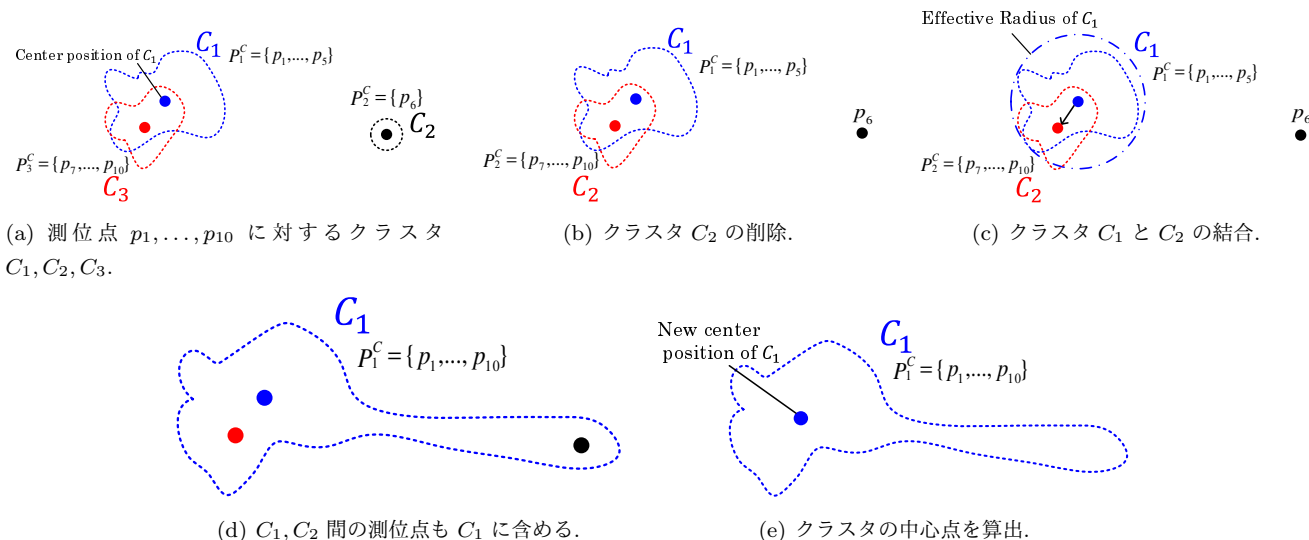


図 7 短時間滞在除去操作.

3.2.3 有効半径に基づくクラスタ結合

クラスタの有効半径をもとに、クラスタ C_i の有効半径以内に次のクラスタ C_{i+1} の中心点が入っていた場合、クラスタ C_i にクラスタ C_{i+1} を結合することができる。例えば図 4 では、測位点 p_1, \dots, p_7 の系列が測位誤差の大きな測位点 p_4 により、クラスタ C_1, C_2 に分割されている。有効半径を利用したクラスタ結合により、 C_1 の有効半径以内に C_2 の中心点が入っているの、これらをクラスタ C_1 に結合することができる。

3.3 クラスタ結合と短時間滞在除去 (Step 4)

有効半径に基づくクラスタ結合を実行したとしても、1~2 個の測位点で構成される小さなクラスタが多数存在する。実際、4 章の実験結果では、有効半径に基づくクラスタ結合を実行した後、クラスタは 474 個生成されるが、このうち 302 個は 1~2 個の測位点で構成される小さなクラスタとなっている。小さなクラスタは含まれる測位点の数が少なすぎるため、真の滞在地を推定することが難しい。例えば、1 個の測位点から構成されるクラスタは測位誤差や移動の可能性が高い。2 個の測位点から構成されるクラスタは測位誤差の可能性もあるが、測位誤差により分裂した他のクラスタの一部の可能性もある。

そこで、移動か測位誤差か判別不能な短時間のクラスタを一旦削除し、再度クラスタを結合する手法法 (以降、短時間滞在除去と呼ぶ) を提案する。短時間滞在除去処理は、まず第 1 段階として 1~2 個の測位点から構成される小さなクラスタを対象に実行し、続いて第 2 段階として最小滞在時間 T_{stay} 未満のクラスタすべてを対象に実行する。結果的に、すべてのクラスタは T_{stay} 以上の滞在時間を持ち、しかも各クラスタは十分な測位点を持つため、真の滞在地に近い滞在地を推定することが期待される。

3.3.1 短時間滞在除去 (第 1 段階)

まず、生成されたクラスタ系列 C について、クラスタに含まれる測位点が 1 あるいは 2 のものに対して短時間滞在除去処理を実行する。図 7(a) のようにクラスタ系列 $\{C_1, C_2, C_3\}$ が与えられたとする。図 7 の各色の点はクラスタの中心点を表す。クラスタ C_2 は 1 つの測位点によって構成される測位点系列 $P_2^c = \{p_6\}$ を持つクラスタなので、クラスタ C_2 の滞在時間は 0 である。 C_2 のように滞在時間の短いクラスタは測位誤差である可能性が高いため、一旦クラスタを削除する (図 7(b))。このとき、クラスタ C_2 は削除されるが、 C_2 に含まれていた測位点 p_6 は削除されないものとする。また、クラスタは再付番され、図 7(a) でクラスタ C_3 は図 7(b) でクラスタ C_2 となる。

次にクラスタ C_1 の有効半径に C_2 が含まれる場合 (図 7(c))、クラスタ C_1 とクラスタ C_2 を結合する (図 7(d))。ここで、クラスタを結合する際、クラスタ C_i に測位点 p_i, p_j ($i < j$) が含まれるとき、 $i < k < j$ を満たす測位点 p_k も C_i に含める。この結合処理により、既存手法では測位誤差とみなされ削除されていた測位点 p_6 も新たなクラスタに含まれることになる。疎な測位情報を用いる場合、測位点そのものは少ないが、提案手法ではできるだけ多くの測位点を利用することができる。

そして式 (4) を用いて新しいクラスタ C_1 に含まれる測位点 p_1, \dots, p_{10} の中心点を算出する (図 7(e))。 p_6 のような測位誤差と考えられる測位点の重みを考慮することで (式 (5))、より真の滞在地に近い中心点を生成できる。

結果的に測位誤差または移動中の可能性が高い測位点を持つクラスタが削除され、図 7(c), 7(d) の結合処理が適用される。この処理により全てのクラスタは 3 個以上の測位点を持つことになり、クラスタの中心点と真の滞在地が近づくことが期待できる。

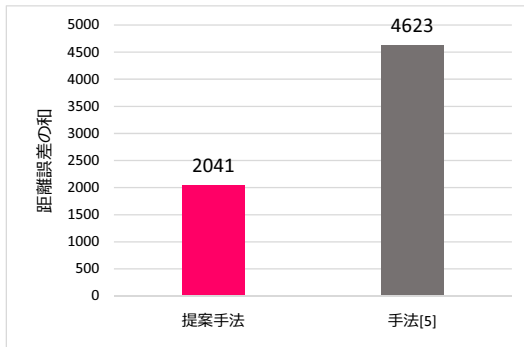


図 8 各手法による推定滞在地の距離誤差の総和。

3.3.2 短時間滞在除去 (第 2 段階)

最後に、最小滞在時間 T_{stay} [秒] 未満のクラスタを削除し、再度短時間滞在除去処理を実行する。

上記 2 段階の短時間滞在除去処理により、大きな GPS 測位誤差によるクラスタの分裂を防ぎ、正確な滞在時間を持つクラスタを生成することが期待できる。

3.4 提案手法の概要

3.1, 3.2, 3.3 節の議論より、以下に提案手法をまとめる。

Step 1 測位精度を元に初期クラスタを生成する。

- 1.1 クラスタ C_i を測位点 p_i で初期化する。
- 1.2 各クラスタ C_i について測位精度によってマージンを定め、クラスタ C_{i+1} が C_i のマージン以内に含まれるなら C_i と C_{i+1} を結合し C_i とする。

Step 2 クラスタに含まれる測位点から、各クラスタの中心点を式 (4) で計算し、滞在時間、測位精度の平均 (l_{c_i}) を算出し、 l_{c_i} からクラスタの有効半径を算出する。

Step 3 時系列順に並んだあるクラスタ C_i と次のクラスタ C_{i+1} の滞在中心点を比較し、 C_i の有効半径以内に C_{i+1} の中心点が含まれるなら、 C_i と C_{i+1} を結合する。

Step 4 短時間滞在除去処理を次の 2 段階で処理する。

- 4.1 1, 2 個の測位点から構成されるクラスタを削除し、Step 2, 3 を処理する。このとき、クラスタ C_i に測位点 p_i, p_j ($i < j$) が含まれるとき、 $i < k < j$ を満たす測位点 p_k もクラスタ C_i に含める。
- 4.2 滞在時間 T_{stay} [秒] 未満のクラスタを削除し、Step 2, 3 を処理する。同様に、クラスタ C_i に測位点 p_i, p_j ($i < j$) が含まれるとき、 $i < k < j$ を満たす測位点 p_k もクラスタ C_i に含める。
- 4.3 各クラスタの中心点を求め、これを滞在地と推定する。

4. 評価実験

提案した滞在地推定手法を、実測位点データに適用し、提案手法を評価する。加えて、測位によるバッテリー消費を

評価する。

4.1 滞在地推定の評価結果

滞在地推定に用いた実測位点データを説明し、提案手法と既存手法を比較する。

4.1.1 実測位点データと実験条件

実測位点データとしてユーザ 1 人の 19 日間の測位点を集めた。測位に使用した携帯端末は SONY 社 XPERIA Z2 である。測位アプリケーションはバックグラウンドで継続して動作し、測位間隔に合わせてシステムのアラームサービスを使用して測位し測位点の情報を記録する。この測位アプリケーションでは、測位端末のバッテリー残量が 50% 以上では 300 秒ごとに測位され、バッテリー残量が 50% 未満では 600 秒ごとに測位される。最小滞在時間 T_{stay} は 1200 秒とする。得られた測位点は 4173 点で、ユーザがその間に 14 箇所に合計 36 回滞在した。

4.1.2 比較手法

比較手法として西田らの時空間 Mean-shift クラスタリングによる滞在地推定手法 [5] を用いた。この手法は Mean-shift クラスタリングの拡張による滞在地推定手法で、高頻度な GPS 測位情報を入力すると、滞在地を推定することができる手法である。この手法は我々の知る限り滞在地の推定精度が最も良い。文献内に示されたパラメータのうち、最低滞留時間 ψ^{time} は提案手法と合わせ 1200 秒とした。

4.1.3 評価結果

提案手法と既存手法 [5] にユーザの GPS 測位点 4173 個の系列を与え滞在地を推定した。まず、手法 [5] と提案手法の総距離誤差を評価すると、提案手法が 2041m、手法 [5] が 4623m となった (図 8)。手法 [5] はクラスタの生成マージンを 20m と固定しているため、クラスタが生成された場合、距離誤差は小さい。しかし、測位誤差により測位点が離れると、ユーザが真の滞在地に滞在中でもクラスタの消失や分裂が起こりやすい。それに対し、提案手法では、測位精度により結合するマージンや有効半径を適応的に変化させることで、多くの場合ユーザが滞在した時間全ての測位点をクラスタに含めることができる。

次に手法 [5] と提案手法について、全ての場所に対する真の滞在回数と推定した滞在回数を比較した (図 9)。図 9 では、各滞在地について、青いバーが真の滞在回数、赤いバーが提案手法で推定した滞在回数、灰色のバーが既存手法で推定した滞在回数を示し、各数字は推定した滞在回数の結果である。手法 [5] では、高頻度な測位情報をもとにしているため、測位誤差を全てクラスタから除外する。そのため、測位誤差により滞在地から遠くに測位されると、その測位点が除外され滞在地とするクラスタが分裂したり、クラスタが形成されず滞在時間が短くなることが多い。提案手法はクラスタの分裂を防ぎ実滞在時間に近い滞在地を生成した。

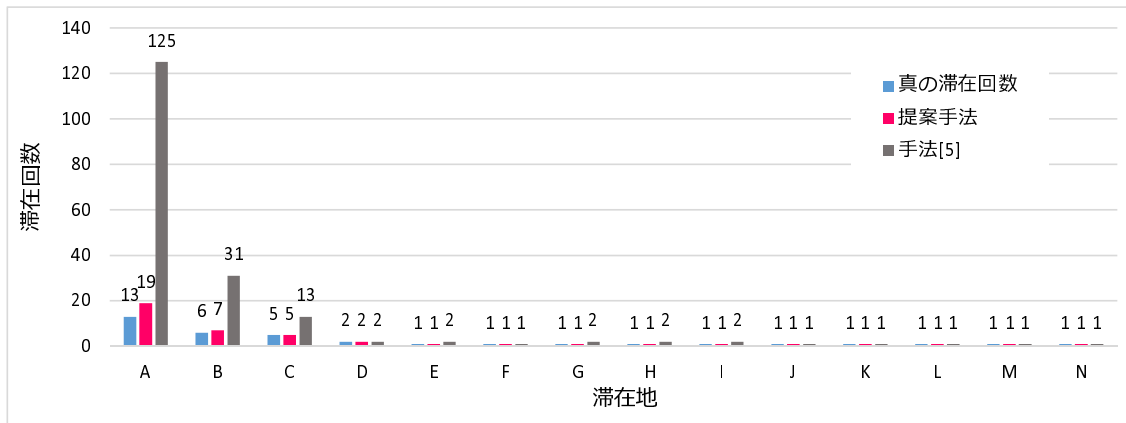


図 9 各場所に対する提案手法と既存手法の滞在推定回数.

4.2 バッテリ消費

従来手法で仮定される密な位置測位と、提案手法のような疎な位置測位によるバッテリー消費を評価するため、ASUS社のNexus 7 (2013) を用い、測位によるバッテリー消費を調査した。Nexus 7はGPSセンサを搭載し、バッテリー容量は3950mAhである。バッテリー消費の調査では、3秒、300秒の測位間隔において、バッテリー残量100%からバッテリー残量が50%になるまでの時間を計測した。

3秒間隔でGPS測位した場合、バッテリー残量が50%になるのに約13時間要した。Nexus 7のバッテリー使用状況ログから、CPU利用時間36分、GPS使用時間が13時間、算出された消費電力が904mAhであった。

300秒間隔でGPS測位した場合、バッテリー残量が50%になるのに約49時間半要した。バッテリー使用状況ログから、CPU利用時間7秒GPS使用時間が18時間半、算出された消費電力が1115mAhであった。

上記の結果より高頻度な測位ではバッテリー消費が低頻度に比べて約4倍大きい。ユーザの多岐にわたる携帯端末の利用を考慮すると、ユーザの携帯端末で測位する場合、数分から数十分以上の測位間隔で低頻度に測位する必要があると言える。この点において、提案手法は疎な測位情報から精度良く滞在地を推定することに成功している。

5. おわりに

本稿では、疎な測位情報から滞在地を推定する手法を提案した。提案手法では(a)測位精度を用いたクラスタの有効半径を設定、これに基づきクラスタを結合し、(b)移動か測位誤差か不明な短時間滞在を除去し再度クラスタに含めるクラスタリングによって、疎な測位情報でも滞在地を高精度の推定した。19日間にわたるユーザの実測位点データに対して提案手法を適用したところ、推定滞在地と真の滞在地の総距離誤差を既存手法の50%以下に削減できることを確認した。さらに、各滞在地の滞在回数を計測したところ、既存手法では滞在回数の誤差合計149回に対し、

提案手法は滞在回数の誤差合計7回となり、正確な滞在地推定が実現された。今後は、推定した滞在地から訪問POI(Point of Interest)を抽出するアルゴリズムを開発する予定である。

参考文献

- [1] 西野正彬, 瀬古俊一, 青木政勝, 山田智広, 武藤伸洋, 阿部匡伸, “滞在地遷移情報からの行動パターン抽出方式の検討,” 情報処理学会研究報告ユビキタスコンピューティングシステム (UBI), vol. 2008, no. 110 (2008-UBI-020), pp. 57–64, 2008.
- [2] 矢部直人, 有馬貴之, 岡村祐, “GPSを用いた観光行動調査の課題と分析手法の検討,” 観光科学研究, vol. 3, pp. 17–30, 2010.
- [3] 山田直治, 磯田佳徳, 南正輝, 森川博之, “GPS搭載携帯電話を用いた移動経路履歴に基づく訪問地・経由地予測システム,” 研究報告ユビキタスコンピューティングシステム (UBI), vol. 2010, no. 8, pp. 1–8, 2010.
- [4] 藤田将成, 手塚博久, 武藤伸洋, 南弘征, 水田正弘, “GPS移動履歴からの接触可能性キーワード抽出法と嗜好推定法の提案,” 行動計量学, vol. 40, pp. 3–15, 2013.
- [5] 西田京介, 戸田浩之, 倉島健, 内山匡, “確率的訪問POI分析: 時空間行動軌跡からのユーザモデリング,” マルチメディア, 分散協調とモバイルシンポジウム 2013 論文集, vol. 2013, pp. 334–345, 2013.
- [6] D. Ashbrook and T. Starner, “Learning significant locations and predicting user movement with GPS,” in *Proceedings of the Sixth International Symposium on Wearable Computers, 2002. (ISWC 2002)*, pp. 101–108. IEEE, 2002.
- [7] D. Comaniciu and P. Meer, “Mean shift: A robust approach toward feature space analysis,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, 2002.
- [8] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, “A density-based algorithm for discovering clusters in large spatial databases with noise.” in *Kdd*, vol. 96, no. 34, pp. 226–231, 1996.
- [9] J. A. Hartigan and M. A. Wong, “Algorithm as 136: A k-means clustering algorithm,” *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 28, no. 1, pp. 100–108, 1979.
- [10] K. Nishida, H. Toda, T. Kurashima, and Y. Suhara, “Probabilistic identification of visited point-of-interest for personalized automatic check-in,” in *Proceedings of*

the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing, pp. 631–642. ACM, 2014.

- [11] NTT Docomo, “測位方法,” <https://www.nttdocomo.co.jp/service/search/usage/gps/>.
- [12] S. J. Sheather and M. C. Jones, “A reliable data-based bandwidth selection method for kernel density estimation,” *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 683–690, 1991.
- [13] N. Yamada, Y. Isoda, M. Minami, and H. Morikawa, “Incremental route refinement for gps-enabled cellular phones,” in *Proceedings of the Fifth International Conference on Mobile Computing and Ubiquitous Networking (ICMU 2010)*, pp. 87–93, 4 2010.
- [14] P. A. Zandbergen, “Accuracy of iphone locations: A comparison of assisted gps, wifi and cellular positioning,” *Transactions in GIS*, vol. 13, no. s1, pp. 5–25, 2009.
- [15] P. A. Zandbergen, “Comparison of wifi positioning on two mobile devices,” *Journal of Location Based Services*, vol. 6, no. 1, pp. 35–50, 2012.
- [16] P. A. Zandbergen and S. J. Barbeau, “Positional accuracy of assisted gps data from high-sensitivity gps-enabled mobile phones,” *Journal of Navigation*, vol. 64, no. 3, pp. 381–399, 2011.