情報処理学会論文誌　データベース　Vol.8　No.1　1–16 (Mar. 2015)

ソーシャルメディア上から収集したジオタグに基づく地理的特徴の抽出と評価

大森 雅己[1,a)]　廣田 雅春[2,b)]　石川 博[3,c)]　横山 昌平[1,d)]

受付日 2014年9月20日, 採録日 2015年1月9日

概要：写真共有サイトでは，多くの写真が共有されており，それらの写真には，ジオタグやタグが付与されている．これらのメタデータを用いて，ホットスポットや，タグが表す地理的特徴を抽出する研究がさかんに行われている．これらの研究において，地理的特徴として，ホットスポットの位置や，タグが表している領域が抽出されている．しかし，地理的特徴を抽出したいタグによっては，領域より線の方が適している場合がある（e.g. 海岸線，線路，高速道路）．たとえば，タグ「coastline」は，海岸付近で撮影された写真に多く付与されていると考えられる．そこで，我々は，タグ「coastline」から実際の海岸線を抽出できるのではないかという仮説を立てた．そして，海岸線のデータを用いて，大量の点群から線を抽出するアルゴリズムを提案し，実際の海岸線データと比較することで評価を行った．結果として，本手法で描いた海岸線の 64%～82%を実際の海岸線から 500 m 以内に描くことができた．また，本手法を海岸線以外のタグについても適用したところ，おおむねタグが表すと考えられる地理的特徴を得ることができた．

キーワード：ジオタグ，ソーシャルタギング，GIS，可視化，地理的特徴

Extraction and Evaluation of Geographical Characterization Based on Geo-tags on Social Media

Masaki Omori[1,a)]　Masaharu Hirota[2,b)]　Hiroshi Ishikawa[3,c)]　Shohei Yokoyama[1,d)]

Received: September 20, 2014, Accepted: January 9, 2015

Abstract: Many photographs have been shared on photo-sharing sites, and those are annotated with tags and geo-tags. Some studies have demonstrated extraction of hotspots and a geographical characterization which a tag represents using those metadata. However, in some cases, a line is more suitable than a region as a geographical characterization (e.g. coastline, railway, and highway). For example, photographs tagged with “coastline” are almost all taken at around a coastline. Therefore, we infered that coastlines can be extracted from photographs tagged with “coastline”. A novel method of extracting and drawing lines on a map was proposed and evaluated. Results show that the distance between lines of 64%–82% of our method and a coastline is less than or equal 500 m. In addition, we applied this method to other tags, and results represent a geographical characterization that the tag represents.

Keywords: geo-tag, social tagging, GIS, visualization, geographical characterization

1. はじめに

デジタルカメラやスマートフォンなどの GPS 搭載デバイスの普及により，ジオタグが付与された写真が，Flickr [1] や Panoramio [2] などの写真共有サイトで共有されている．

[1] 静岡大学大学院情報学研究科
Graduate School of Informatics, Shizuoka University, Hamamatsu, Shizuoka 432–8011, Japan
[2] 首都大学東京大学院システムデザイン研究科/日本学術振興会特別研究員 PD
Graduate School of System Design, Tokyo Metropolitan University/Research Fellow of the Japan Society for the Promotion of Science (PD), Hino, Tokyo 191–0065, Japan
[3] 首都大学東京システムデザイン学部情報通信システムコース
Information and Communications Systems, Faculty of System Design, Tokyo Metropolitan University, Hino, Tokyo 191–0065, Japan

[a)] gs13008@s.inf.shizuoka.ac.jp
[b)] hirota-masaharu@tmu.ac.jp
[c)] ishikawa-hiroshi@tmu.ac.jp
[d)] yokoyama@inf.shizuoka.ac.jp

ジオタグとは，写真の撮影位置の緯度経度などの位置情報を表すメタデータである．近年，ジオタグ付き写真が増加しており，Flickr では，2009 年 2 月に 1 億枚目のジオタグ付き写真が投稿された [3]．また，写真共有サイトでは，多くのユーザによって写真にタグを付与するソーシャルタギングが行われている．タグは，写真に関するテキストであり，撮影場所の地名や撮影対象であることが多い．

これらの，写真に付与されているジオタグとタグを用いて，地理的特徴を抽出する研究がさかんに行われている．地理的特徴を得ることにより，地理的な環境の状態などの地理空間についての知識を把握することが可能である．また，写真共有サイトのデータを用いるため，人々の興味についても把握することも可能である．Zhang ら [4] は，写真と写真に付与されたタグとタイムスタンプから雪や，緑に覆われている地域と時間を推定し，実際のデータと比較を行った．Thomee ら [5] は，画像処理の技術を用いて，タグが表す地理的な領域を可視化した．また，ホットスポットと呼ばれる，多くの人々が写真を撮影する場所を抽出する研究も行われている [6], [7]．Crandall ら [8] は，ウェブ上の大量のジオタグ付き写真に対して，密度に基づいたクラスタリングを適用し，ランドマークやホットスポットを抽出する手法を提案した．Kisilevich ら [9] も，密度に基づくクラスタリングにより，ホットスポットの抽出を行う手法を提案した．

これらの手法は，領域を抽出しているが，抽出するものによっては，領域では適さない場合がある．たとえば，海岸線や，線路，マラソンコースは，領域ではなく線の方が，形状として適している．そのため，地理的特徴として，領域を抽出するのみでなく，線状で抽出することも重要である [10]．そこで，我々は，これまでの研究とは異なるアプローチとして，写真共有サイトで共有されている写真群から，線状の地理的特徴を抽出する手法を提案する．

先ほどあげた海岸線や線路は地図からそれらの位置を知ることができる例である．本研究では，人々が写真に付与したタグを利用するため，人々が使う言葉の概念や人々が興味を持つ地域を抽出することも可能であると考えられる．たとえば，「beach」というタグで海岸線を描ければ，海水浴が行われ，人々が集まる浜辺だけでなく，海岸沿いで撮影された写真には「beach」というタグが付与されていることが分かる．また，道路を抽出する場合では，「road」，「street」，「avenue」といった近い意味の言葉の使われ方の違いを知ることができる．さらに，「ジョギングコース」や「散歩道」といったタグを用いれば，多くの人がその用途に使っている道を知ることができるといった使い方もできると考えられる．また，写真の撮影位置を用いるので，線を抽出できたかどうかで，その地域に人が訪れるか，人が写真を撮影するかという情報を知ることも可能である．

本研究の応用としては，線の周囲の写真に付与されたタグから，抽出された線に関する追加の情報を得ることがあげられる．例としては，線の一部の地域で多く使われる固有名詞から海岸名を抽出したり，「夕日」というタグが多く付与されているところでは，夕日が綺麗に撮影できる海岸であることを抽出したりすることが考えられる．さらに，線の抽出に利用した写真の撮影時刻を考慮することで，ルート推薦に応用ができると考えられる．

我々は，あるタグが付与された写真を撮影位置に基づいて地図上に配置したときに，タグが表す地理的な形状を示しているのではないかという仮説を立てた．たとえば，タグ「coastline」が付与された写真の撮影位置は，海岸線沿いに集まっており，写真の撮影位置から海岸線が把握できると考えた．この仮説を検証するために，Flickr から収集したジオタグとタグが付与された写真から海岸線を抽出し，評価を行った．本研究において，海岸線を選んだ理由としては，海岸線の形状は明確であり，海岸線の位置データが存在することや，海岸線は世界地図でよく目にするため，誰もが，本手法で描いた線が海岸線と近似しているかを一目で分かるということがあげられる．また，高解像度の実際の海岸線データは，OpenStreetMap [11] や，NOAA [12] から取得可能である．本論文では，実際の海岸線のデータと比較を行い，本手法の定量的な評価を行う．さらに，本手法を海岸線以外のタグに対しても適用し，海岸線以外のタグに対しても適用可能であることを示す．

海岸線を描くためには，まず，海岸線に関するタグが付与された写真の撮影位置が海岸線沿いに集まっているかを確認する必要がある．そこで，2 章では，前実験として，海岸線に関するタグが付与された写真の撮影位置が海岸線付近で撮影されているかを調査し，その結果について述べる．3 章では，大量の写真の撮影位置から線を描画する手法について説明する．4 章では，提案手法の実行結果とそれに対する考察を述べる．5 章では，提案手法で描いた線と実際の海岸線との距離を求め，提案手法の精度の評価を行う．6 章では，本研究の関連研究について述べる．7 章では，本研究で得られた成果のまとめと今後の展望について述べる．

2. ソーシャルタギングの地理的な信憑性

本章では，ソーシャルタギングの地理的な信憑性として，海岸線付近で撮影された写真に付与されると考えられるタグが実際に，海岸線付近で撮影された写真に付与されているかを調査する．写真の撮影位置から実際の海岸線までの最短距離を求め，距離が近い写真が多ければ，タグの地理的な信憑性が高いとする．

2.1 データセット

我々は，Flickr から約 2 億枚のジオタグ付き写真を収集した．その中から，Flickr から海岸線付近で撮影された写真に付与されると考えられるタグを選出し，それらのタグ

表 1　海岸線に関連するタグと，そのタグが付与された写真数
Table 1　Number of photographs having tags related to coastlines.

タグ	写真数
beach	2,488,923
sea	1,689,924
coastline	60,245
shoreline	47,114

表 2　それぞれのタグが付与された写真の撮影位置の分布
Table 2　Distribution of photographic locations from an actual coastline.

タグ	平均誤差 [m]	100 m 以内 [%]	500 m 以内 [%]
beach	7,293.33	51.34	80.44
sea	8,307.95	48.20	76.77
coastline	3,921.20	56.25	82.54
shoreline	23,510.43	51.92	70.93

が付与された写真を収集した．**表 1** は，写真の収集に用いたタグとそのタグが付与された写真の数を表している．これらの写真の撮影位置と実際の海岸線との距離を算出した．その際に，実際の海岸線として，OpenStreetMap の海岸線データを利用した．

2.2　結果・考察

写真の撮影位置と実際の海岸線との距離を調査した結果を**表 2** に示す．平均誤差は，それぞれのタグが付与された写真の撮影位置と正解の海岸線との距離の平均誤差，100 m 以内と 500 m 以内は，それぞれのタグが付与された写真ごとに，海岸線から 100 m 以内で撮影された写真の割合，500 m 以内で撮影された写真の割合を示している．

写真の撮影位置と実際の海岸線の位置との平均誤差は，どのタグにおいても数千 m 以上であるが，約 70%から 80%の写真が海岸線から 500 m 以内で撮影されていることが分かる．平均誤差が大きい原因は，ノイズとなる写真の存在が考えられる．ノイズとなる写真として，GPS を使わずにユーザによって付与されたジオタグが付与された写真，ユーザのミス，もしくは，閲覧数を上げるために，故意に写真の内容と関係がないタグが付与された写真などがある．これらの写真の撮影位置は，海岸線の位置とは関係なく様々な場所に存在する．他にも，地名など，別の意味として付与されているものがある．たとえば，タグ「beach」が付与されている写真には，Long Beach というアメリカの都市で撮影されたものが存在する．地名として Long Beach をタグ付けする場合に，「long」と「beach」のように分けて 2 つのタグが付けられる場合がある．このような場合も，海岸線から離れた場所に写真がある原因である．これらのことから，海岸線から遠い場所にも写真が多く存在し，写真の撮影位置と実際の海岸線の位置との平均誤差が大きくなっている．

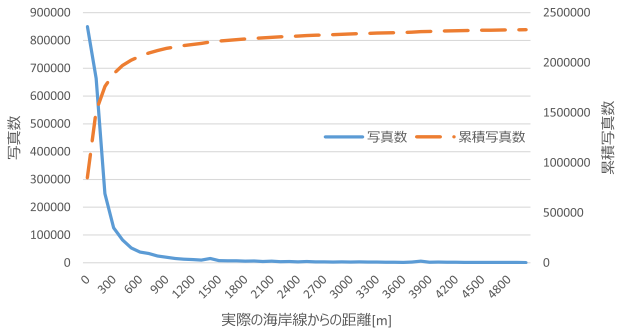


図 1　タグ「beach」が付与された写真の海岸線からの距離
Fig. 1　Distance between photographic location tagged with “beach” and coastline.

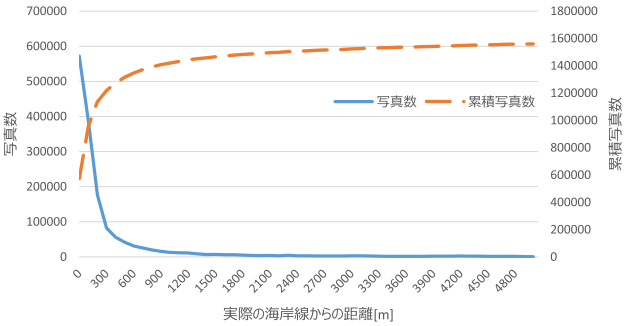
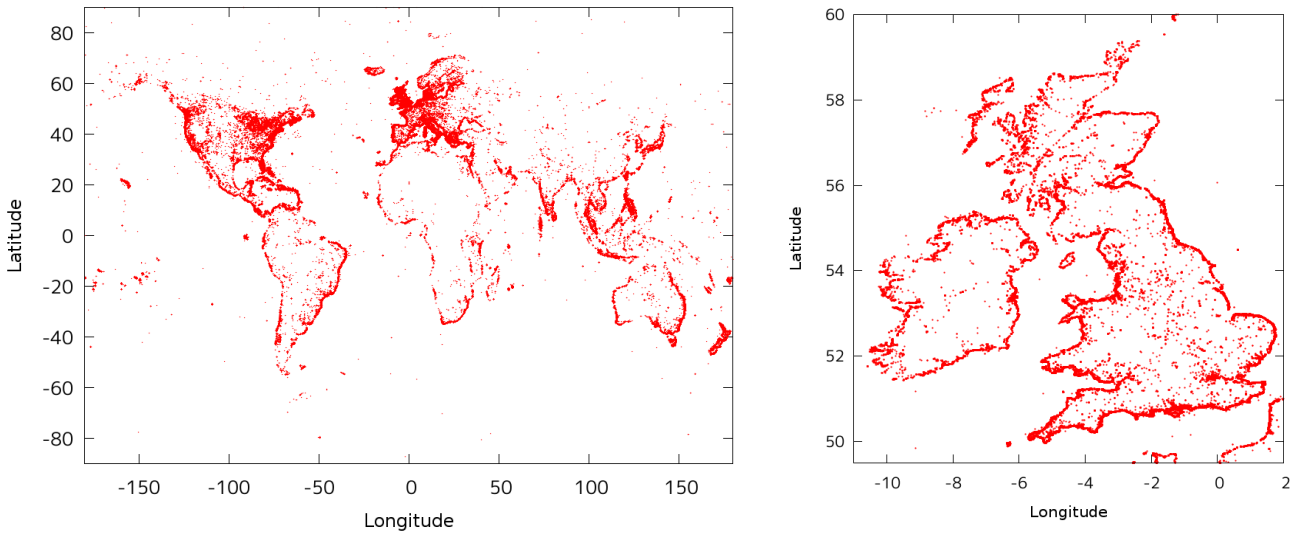


図 2　タグ「sea」が付与された写真の海岸線からの距離
Fig. 2　Distance between photographic location tagged with “sea” and coastline.

また，タグ「sea」の平均誤差は，タグ「beach」の平均誤差よりも大きい．その理由として，タグ「sea」は，海岸沿いの道路や山の上，海上などの，海岸から離れているが，海が写っている写真に付与される場合があるためと考えられる．また，海岸線付近で撮影された写真においても，海岸線から多少離れた場所で撮影された写真が多い．その理由としては，写真の撮影位置と撮影対象との距離，潮の満ち引き，GPS の誤差などが考えられる．その中でも撮影対象との距離が主な理由として考えられる．海岸付近で写真を撮影するとき，浜辺など陸から海岸を撮影することや船上から陸を撮影することはあっても，海岸線上に立って周囲を撮影することはほとんどないと考えられる．そのため，海岸線の位置と写真の撮影位置に数百 m 程度の差があると考えられる．

図 1，**図 2**，**図 3**，**図 4** は，OpenStreetMap の海岸線と写真の撮影地点の距離ごとの写真数のヒストグラムとその累積分布関数である．青色の実線がヒストグラム，橙色の破線が累積分布関数である．横軸は，撮影位置と海岸線との距離，縦軸は，写真数を表している．距離は 100 m 間隔で集計しており，図 1 の距離 0 m が表している値の 85 万とは，0 m〜100 m の距離で撮影された写真が 85 万枚であることを表している．累積分布関数は，その値までの累



(a) 世界全体
The whole world.

(b) イギリス周辺の拡大図
Whole of the UK.

図 5　タグ「beach」が付与された写真の撮影位置
Fig. 5　Photographic location tagged with “beach”.

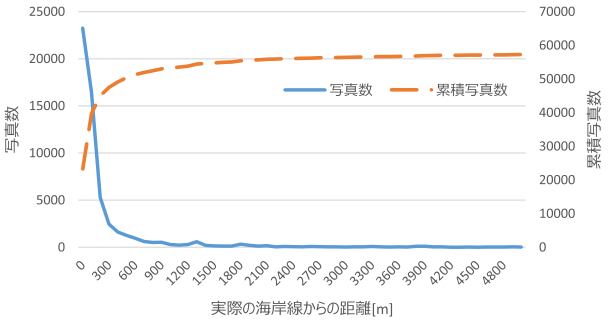


図 3　タグ「coastline」が付与された写真の海岸線からの距離
Fig. 3　Distance between photographic location tagged with “coastline” and coastline.

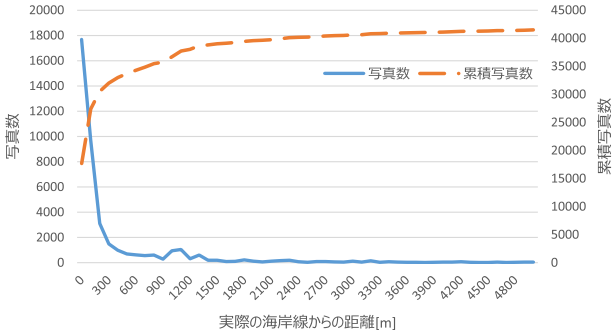


図 4　タグ「shoreline」が付与された写真の海岸線からの距離
Fig. 4　Distance between photographic location tagged with “shoreline” and coastline.

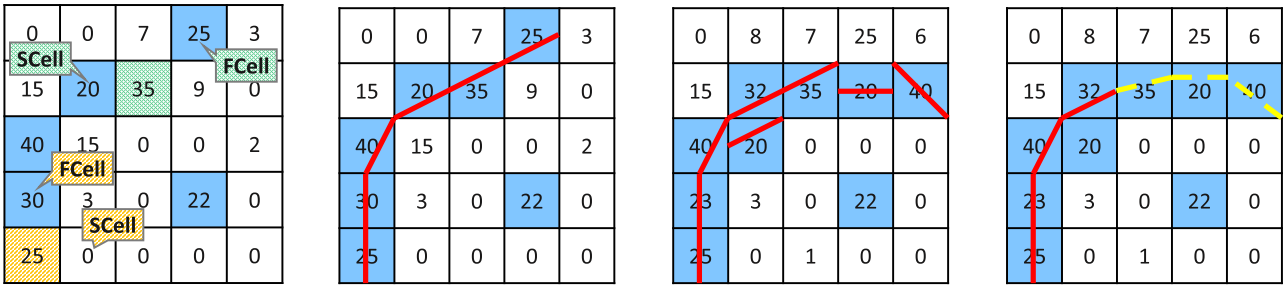
計値を表しており，900 m の距離が表しているのは，0 m～1000 m の距離で撮影された写真数である．

図 1～図 4 のすべてにおいて，ロングテールの形状をしており，ほとんどの写真が海岸線付近で撮影されていることが分かる．また，300 m から 600 m 付近から写真数が減少している．この 4 つのグラフを比較すると，図 4 だけが 1500 m 付近まで他と少し異なる形状をしている．また，表 2 の「shoreline」の 500 m 以内の割合は約 70%であり，4 つの中では最も低い．これは，「shoreline」という単語が海岸以外の川や湖の岸辺に対しても用いられる単語であるということが原因として考えられる．今回用いた OpenStreetMap の海岸線データには，川や湖は含んでいない．そのため，湖岸など海から離れた地域で撮影されている写真の分，他のタグと比較して，海岸付近で撮影された割合が下がっていると考えられる．

表 1 および表 2 より，タグ「beach」が付与された写真は最も枚数が多く，海岸線付近で撮影されている割合も高い．そこで，タグ「beach」が付与された写真を緯度経度に基づいてプロットした．世界全体での結果を**図 5** (a) に，イギリス周辺の拡大図を図 5 (b) に示す．図 5 (a) から，それぞれの大陸のおおよその形が把握できるが，写真の枚数が多いヨーロッパ大陸やイギリス付近では，図 5 (a) から海岸線の形状を把握することは困難である．しかし，拡大すると，図 5 (b) のように，海岸線の形状が把握することができる．これらのことから，十分な写真数があり，海岸線付近で撮影された写真の割合が高いタグ「beach」を用いて，海岸線の再現を試みる．

3. 提案手法

本章では，大量の写真の撮影位置から線を描く手法について述べる．本手法は，(1) 線を描く地域をグリッドに分割し各セルの中で線を引く，(2) 隣接するセル間で線をつなぐ，(3) 離れたセル間で線をつなぐ，という 3 つの段階



(a)
FCell と SCell の選択
Selection of FCell and SCell.

(b)
基準線がつながっている例
Example of a connected line.

(c)
基準線がつながっていない例
Example of unconnected lines.

(d)
図 6(c) の基準線を修正した結果
Result of unconnected lines in Figure 6(c) are modified.

図 6　基準線の描画と修正
Fig. 6　Example of the procedure used to draw lines.

からなる．

3.1　地域の分割および基準線の描画

本節では，線を描く地域の分割と本手法の基準となる線を引く手順について述べる．線を引く流れは，(1) グリッドに分割，(2) 線を引く候補となるセルの決定，(3) 周囲のセルに含まれる写真数に基づいて，候補となったセル内に線を引く，である．ここで，線を引く候補となるセルを候補セル，各セルの中に引く線を基準線と呼ぶ．

本手法を用いて線を引く例を**図 6** に示す．セル内の数値はそれぞれのセルの範囲内で撮影された写真の枚数，色が塗られたセルは候補セルを表している．また，候補セルの決定と基準線の描画の手順を Algorithm 1 に示す．ここで，引数 θ は閾値を，変数 *remarkable_cells* は候補セルを表している．また，around4(*cell*) は *cell* の上下左右のセルを返す関数，around8(*cell*) は *cell* の周囲 8 セルを返す関数，num(*cell*) は *cell* に存在する写真数を返す関数，is_remarkable_cell(*cell*) は *cell* が候補セルの場合に true，そうでない場合に false を返す関数，max(*cells*) は *cells* の中で最も写真数が多いセルを返す関数，opposite(*cellA*,*cellB*) は *cellA* に対して，*cellB* と反対方向に位置する隣接セルを返す関数，center(*cellA*,*cellB*) は *cellA* と *cellB* の中間点を返す関数，drawline(*pointA*,*pointB*) は *pointA* と *pointB* を結ぶ線を描画する関数である．

まず，海岸線を再現したい任意の地域を選択する．この際に，グリッドの最外側には写真がほとんど存在しないことを想定しているため，線を描く領域より，多少広く地域を選択することが好ましい．地球全土を対象とすることも可能だが，計算量の都合上，本論文では，地域を選択し必要な地域にのみ線の描画を行う．次に，それぞれのセルの範囲内で撮影された写真の枚数を求める．その後，それぞれのセルに対して，隣接する上下左右のセルと写真の枚数を比較し，少なくとも 1 つのセルとの枚数の差が閾値 θ 以上の場合は，枚数が多い方のセルを候補セルとする．図 6 の例は，$\theta = 20$ とした場合である．

Algorithm 1 基準線の描画
Drawing basic lines.

```
1: argument: θ
2: remarkable_cells ⇐ φ
3: foreach cellα ∈ all cells do
4:     foreach cellβ ∈ around4(cellα) do
5:         if num(cellα) − num(cellβ) > θ then
6:             remarkable_cells ⇐ remarkable_cells ∪ cellα
7:         end if
8:     end for
9: end for
10:
11: foreach cellC ∈ remarkable_cells do
12:     count ⇐ 0
13:     foreach cellA ∈ around8(cellC) do
14:         if is_remarkable_cell(cellA) then
15:             count ⇐ count + 1
16:         end if
17:     end for
18:     if count > 0 then
19:         cellα ⇐ max(around8(cellC))
20:         cellβ ⇐ max(around8(cellC)−cellα−around4(cellα))
21:         if num(cellβ) = 0 then
22:             cellB ⇐ opposite(cellC,cellα)
23:         end if
24:         drawline(center(cellC,cellα), center(cellC,cellβ))
25:     end if
26: end for
```

次に，候補セル内での線の描画を行う．それぞれの候補セルに対して，周囲の 8 つのセルの中に 1 つでも候補セルが存在した場合，図 6 (a) のように，周囲の 8 つのセルのうち，最も写真数の多いセル（FCell）と 2 番目に写真数が多いセル（SCell）を算出する．ただし，SCell は，FCell と隣接しないセルから選出される．これは，あるセルにおいて片側にのみ写真数が偏っていた場合の対策である．図 6 (a) の緑の交差の模様，黄色の斜線の模様で塗られたセルは，同模様の FCell および SCell と対応している．左下にある黄色の斜線が引かれた，写真数 25 のセルの FCell は，上側

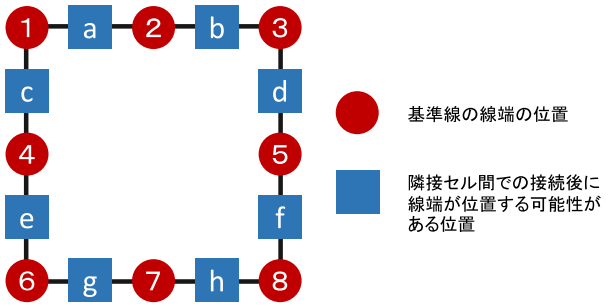


図 7　セル内に引かれる基準線の線端の位置
Fig. 7　Location of end points of a basic line in a cell.

の写真数 30 のセルであり，SCell は，右側の写真数 0 のセルとなる．右上の写真数 3 のセルは，FCell と隣接するため SCell とならない．

FCell と SCell が決定したら，FCell から SCell へ向かって線を引く．線を引くときの線の始点と終点の位置は，図 7 の数字が書かれた円がある 8 カ所のいずれかである．たとえば，FCell が上のセル，SCell が右下のセルであった場合は，図 7 の 2 から 8 へ線が引かれる．周囲の 8 セルすべてが候補セルでない場合は，そのセルには線を引かない．FCell および SCell が 1 つに定まらない例外時には，次のような処理を行う．まず，FCell が 2 つであり，かつ隣接していない場合は，FCell から SCell に線を引くのと同様に 2 つの FCell 間で線を引く．また，図 6 (a) の最左下のセルのように，SCell の枚数が 0 枚の場合は，FCell から中心のセルに対して反対側にあるセルへ向かって線を引く．次に，写真数が 0 でない，FCell または SCell が複数ある場合について述べる．このような場合は，FCell または SCell の中で候補セルであるものだけを FCell または SCell として扱い，上記の条件で線を引く．それでも FCell または SCell が複数である場合は，周囲のセルから適切な線の位置を決定することが困難であるため，そのセルには線を引かない．図 6 (b) は，図 6 (a) から線をつないだ結果である．

また，FCell と SCell が存在しないという場合は存在しない．仮に，周囲 8 セルすべての写真数が 0 枚の場合は，8 セルすべてが FCell である．ただし，このような場合は，8 セルすべてが非候補セルであるため，FCell および SCell の算出処理をする必要はない．

3.2　隣接するセル間の線の接続

本節では，隣接するセル間の線をつなげる手順について述べる．前節で述べた方法では，図 6 (b) のように隣接するセル間でつながっているとは限らず，図 6 (c) に示すように，セル内の線と隣のセルの線がつながっていない場合がある．そのため，図 6 (d) のように線を修正し，つなぐ処理を行う．

隣接するセル間で線をつなぐ手順を Algorithm 2 に示す．ここで，is_exist_point_in_cell(*point*,*cell*) は *cell* 内に

Algorithm 2 隣接するセル間の線の接続
Connecting lines between adjacent cells.

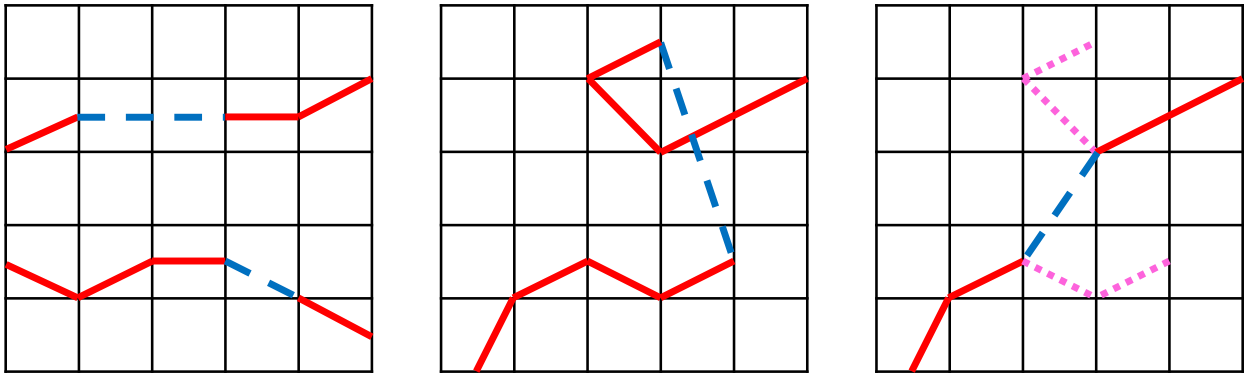
```
1: foreach cellα ∈ all cells that have a line do
2:   foreach pointα ∈ both end points on cellα do
3:     cells ⇐ φ
4:     foreach cellA ∈ around8(cellα) do
5:       if is_exist_point_in_cell(pointα,cellA) then
6:         cells ⇐ cells ∪ cellA
7:       end if
8:     end for
9:     cellβ ⇐ max(cells)
10:    side ⇐ get_side_between_cells(cellα,cellβ)
11:    points ⇐ get_points(side)
12:    pointα ⇐ center(points)
13:    redrawline(cellα)
14:  end for
15: end for
16:
17: foreach lineα ∈ all lines for which both end points are not
    connected do
18:   delete(lineα)
19: end for
```

point が存在すれば true，そうでなければ false を返す関数，get_side_between_cells(*cellA*,*cellB*) は *cellA* と *cellB* が共有する辺を返す関数，get_points(*side*) は *side* 上にあるすべての線端を返す関数，center(*points*) は *points* の中間点を返す関数，redrawline(*cell*) は *pointα* の変更を適用し *cell* 内の線を再描画する関数である．これらの *point* とは，セルの辺上に位置する線端のことである．

隣接するセル間の線をつなぐ方法は，いずれかの辺上に，隣接セルの基準線とつながっていない基準線の線端が 2 つあった場合に 2 つの線端をそれらの中間点に移動することで基準線をつなぐ．ただし，線端がセルの頂点上にある場合は，頂点を始点とする 2 つの辺を共有する 2 つの隣接セルに含まれる写真数を比較する．そして，写真数が多い方のセルと共有する辺上に線端があると見なしてつなぐ処理を行う．基準線の線端の位置は，同じ辺上の 2 点の中間点になるため，図 7 のアルファベットが書かれた四角形の位置に位置する可能性がある．たとえば，図 7 の 1 から 7 に線が引いてあり，隣接する上側のセルの線端が 3 に位置している場合は，1 と 3 の中間点である 2 に線端を移動させる．つまり，2 から 7 に引かれた線となる．また，上側のセルの線端が 2 にあった場合は，1 と 2 の中間点である a に線端を移動させ，a から 7 への線となる．すべての隣接するセル間で線をつないだ後，基準線の両端が隣のセルとつながっていない線を削除する．図 6 (d) は，図 6 (c) の基準線を隣接するセル間でつないだものである．黄色の破線が修正された箇所である．

3.3　離れたセル間の線の接続

本節では，離れたセル間の線をつなぐ手順について述べ



(a) 単純につなぐことが可能な例
Example of connecting lines easily.

(b) 誤ってつないだ例
Example of wrongly connecting lines between remote cells.

(c) 修正を行いつないだ結果
Result of connecting lines between remote cells with fixing.

図 8　離れたセル間で線をつなぐ例
Fig. 8　Example of connecting lines between remote cells.

る．前節までの手順では，離れたセル間の線はつながれていない．そのため，離れたセル間で線をつなぐ処理を行う．離れたセル間の線をつなぐ例を**図 8** に示す．基本的には，図 8 (a) のように，つながっていない線端どうしを近い順につなぐ．図 8 の赤色の実線が前節までの手順で描いた線，青色の破線が赤色の線を距離が近い順につないだ線である．このとき，図 8 (b) に示す例のように，単純に近い線をつなぐと，適切でないと考えられる線が引かれる場合がある．そのため，図 8 (c) に示すように修正を行う．図 8 (c) のピンク色の点線が図 8 (b) の赤色の実線から削除した線であり，青色の破線が修正後の最終的につながれた線である．

離れたセル間の線をつなげる手順を Algorithm 3, Algorithm 4 に示す．ここで，引数 *maxdist* はつなぐことを許可する最大距離，*maxdel* は削除を許可する最大の線数，*disttable* はつながっていないすべての線端どうしの距離を保持するテーブルである．また，sort_order_by_distance_ascending(*disttable*) は *disttable* を距離の昇順で並べ替える関数，get_first_point(*row*) は *disttable* の行 *row* から一方の線端を返す関数，get_second_point(*row*) は *disttable* の行 *row* から get_first_point とは異なる線端を返す関数，get_distance(*row*) は *disttable* の行 *row* から距離を返す関数，not_connect(*point*) は線端 *point* がいずれかとつながっていなければ true，そうでなければ false を返す関数，get_connect_point(*point*) は線端 *point* とつながっている線端を返す関数，get_another_point(*point*) は基準線の *point* と逆側の線端を返す関数である．lat(*point*), lon(*point*) はそれぞれ *point* の緯度，経度を返す関数であり，left_side(*point*), right_side(*point*), top_side(*point*), bottom_side(*point*) はそれぞれ線端 *point* がセルの左側の辺，右側の辺，上側の辺，下側の辺の上にあれば true，そうでなければ false を返す関数である．

Algorithm 3 離れたセル間の線の接続 (1)
Complement remaining unconnected lines (1).

```
 1: argument: maxdist
 2: argument: maxdel
 3: argument: disttable
 4: sort_order_by_distance_ascending(disttable)
 5: foreach row ∈ disttable do
 6:     p1 ⇐ get_first_point(row)
 7:     p2 ⇐ get_second_point(row)
 8:     if not_connect(p1) and not_connect(p2) and
        get_distance(row) < maxdist then
 9:         for 1 to maxdel do
10:             if try_connect(p1,p2) then
11:                 break
12:             end if
13:             tmp1 ⇐ p1
14:             p1 ⇐ get_connect_point(get_another_point(p1))
15:             if try_connect(p1,p2) then
16:                 break
17:             end if
18:             tmp2 ⇐ p1
19:             p1 ⇐ tmp1
20:             p2 ⇐ get_connect_point(get_another_point(p2))
21:             if try_connect(p1,p2) then
22:                 break
23:             end if
24:             p1 ⇐ tmp2
25:         end for
26:     end if
27: end for
```

最初に，つながっていないすべての線端から，距離がパラメータ *maxdist* 未満のつながっていないすべての線端までの距離を求める．そして，距離が近い順につなぐ処理を行う．ただし，一方の線端がすでに別の線端とつながっている場合はつなぐ処理を行わない．線をつなぐ際に，線の向きに基づいて線の修正を行う．まず，線端がセル上の

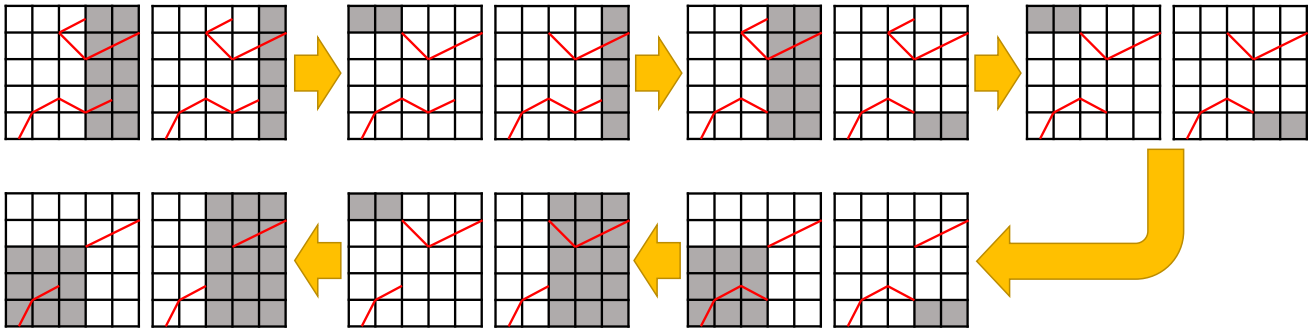


図 9　図 8 (b) から図 8 (c) までの流れ
Fig. 9　Flow from Fig. 8 (b) to Fig. 8 (c).

Algorithm 4 離れたセル間の線の接続 (2)
Complement remaining unconnected lines (2).

```
 1: declare function delete_connect_line (point)
 2:   while is_not_null(point) do
 3:     tmp ⇐ get_connect_point(get_another_point(point))
 4:     delete (get_line(point))
 5:     point ⇐ tmp
 6:   end while
 7: end function
 8:
 9: declare function try_connect (p1,p2)
10:   if aim_to(p1,p2) and aim_to(p2,p1) then
11:     delete_connect_line(get_connect_point(p1))
12:     delete_connect_line(get_connect_point(p2))
13:     redrawline(p1,p2)
14:     return true
15:   end if
16:   return false
17: end function
18:
19: declare function aim_to (p1,p2)
20:   result ⇐ true
21:   if left_side(p1) and lon(p1) < lon(p2) then
22:     result ⇐ false
23:   end if
24:   if right_side(p1) and lon(p1) > lon(p2) then
25:     result ⇐ false
26:   end if
27:   if top_side(p1) and lat(p1) > lat(p2) then
28:     result ⇐ false
29:   end if
30:   if bottom_side(p1) and lat(p1) < lat(p2) then
31:     result ⇐ false
32:   end if
33:   return result
34: end function
```

どの位置にあるかに基づいて，線端が向いている方向を求める．方向は，セルの頂点の場合は，左上，右上，左下，右下の 4 方向，それ以外の場合は，左，右，上，下の 4 方向のいずれかとなる．(1) 基準線が引かれたセルから見て，この方向にもう一方の線端があるかを調べる．ここで，双方ともに線端が向いている方向にもう一方の線端がある状態を向かい合っていると定義する．(2) 線端どうしが向かいあっている場合は，距離が *maxdist* 未満ならば，その 2 つの線端を直線でつなぐ．(3) 向かい合っていない場合は，どちらか一方の線を 1 セル分短くし，もう一度 (2) を行う．(4) それでも向い合っていなければ，短くした線を元に戻し，もう一方の線を 1 セル分短くし，(2) を行う．(3) と (4) のどちらでも向かい合わない場合は，双方の線を 1 セル分短くし，線を削ったセル数がパラメータ *maxdel* を超えるか，線をつなぐまで (2) から (4) を繰り返す．*maxdel* を超えた場合は，短くした線を元に戻し，次に距離が近い線端間の接続処理に移る．

この手順で図 8 (b) から図 8 (c) にいたるまでの流れを**図 9** に示す．それぞれの左側の図は，上側の線が向いている方向，右側の図は，下側の線が向いている方向を表している．色が塗られたセルに，もう一方の線端が位置しているときが，向かい合っている状態である．図 9 の左下の流れの最後では，左右両方の図で色が塗られたセルにもう一方の線端があるため，図 8 (c) のように，その状態で線をつなぐ．

3.4　提案手法の考察

本研究では，グリッドを用いて各セル内で基準線を引き，基準線をつなげるというアプローチをとった．グリッドを用いた理由は，まず，GPS によって計測された位置情報には，誤差が含まれることや同じ対象を撮影した場合でも撮影位置に多少の散らばりがあることがあげられる．そのため，写真の撮影位置をグリッドに写像することで，これらの影響を低減できると考えられる．また，大量の写真を扱うため，1 つ 1 つの写真に対して処理を行うことに比べ，グリッドに分割し，それぞれのセルに対して処理を行うことで，大幅に計算量を削減できるという利点がある．さらに，セルごとに処理を並列化することも可能であり，グリッドを用いることで効率的に処理を行うことができる．このような利点があるため，グリッドを用いたアプローチは，ジオタグを用いた研究でよく用いられている手法となっている [13], [14], [15], [16], [17], [18]．そこで，本研究の線を抽出する手法においてもグリッドを利用した．

グリッドを用いたため，分割されたそれぞれのセル内で独立して線を引く処理を行うが，それだけでは，3.2 節で

述べたように隣接セルと線がつながらない場合がある．そのため，隣接するセル間で線を修正し，つなぐ処理を行う．仮に，海岸線上で多数の写真が偏りがなく撮影されていれば，隣接するセル間で線をつなげば，海岸線は 1 つの線として抽出できるはずである．しかし，実際には，人々が訪れない地域，写真が撮影されない地域では，線が引かれないため，1 つの海岸線が途切れた複数の線として抽出されてしまう．そこで，3.3 節に述べた離れたセル間で線をつなぐことで，こういった地域の補完を行う．これらのことから，本研究では，3 段階のアプローチをとった．

本研究では，1 章で述べたように，線状の特徴の抽出対象として海岸線を用いた．そのため，海岸線には存在せず，他の線状のものに存在する形状は，対応していない．たとえば，線の分岐があげられる．海岸線は分岐することはないが，線路や道路では，Y 字路や交差点といった線の分岐が存在する．本手法では，分岐に対応していないため，分岐があるものに適用すると，「y」の字のような分岐が「ソ」の字のように抽出され，隙間ができてしまう．この問題は，本手法で復元した近接する 2 つの線が接続するかどうかを推定し，接続すると判定した線をつなぐことにより，分岐を含む線を抽出できると考えられる．このアプローチを用いれば，現時点の手法に手を加えることなく，追加の処理を行うことで，分岐の抽出ができると考えられる．しかし，別途，実験と評価を要するため，今後の課題とする．

4. 実行結果

本章では，本手法を Flickr から取得した写真に適用した実行結果について述べる．本論文では，2 章で，写真数が多く海岸付近で撮影された写真の割合も高い結果を得られたタグ「beach」を用いて，海岸線の描画を行う．また，海岸線以外のタグについても本手法を適用し，海岸線以外で有効であるかを検証する．

4.1 データセット

本論文では，**表 3** のデータセットに対して，提案手法を適用した．各データセットは，2 章と同じデータセットから，タグと地域でフィルタリングしたものである．琵琶湖は，写真枚数が少ないため，琵琶湖を表すと考えられる複数のタグを用いた．これらのデータセットにおいて，撮影位置の緯度経度のいずれかが整数値のものは除外した．それらのほとんどは，GPS によるものでなく，ユーザの手入力によるものであり，実際の撮影位置から大きくずれている可能性が高いためである．なお，本論文では Flickr の写真を用いるが，ジオタグとタグのみを用いるため，Twitter [19] などのジオタグを付与可能な他のソーシャルメディアサイトにおいても適用可能である．

表 3　データセットの種類
Table 3　Kind of datasets.

タグ	場所	写真数	抽出対象
beach	ハワイ	12,747	海岸線
beach	イギリス	218,566	海岸線
琵琶湖 & biwako & lakebiwa & biwalake	滋賀県周辺	3,152	琵琶湖の外周
border	ヨーロッパ	12,817	国境
shinkansen	日本	6,955	新幹線の路線図

4.2 結果・考察

海岸線の描画結果は，OpenLayers [20] を用いて Google Maps [21] 上に表示した．パラメータの決定は，それぞれのデータセットにおいて，写真数や撮影位置の分布からいくつかの値で実行し，人目で最も良いと判断した値を用いた．

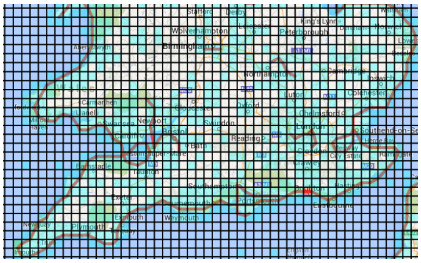
4.2.1 タグ「beach」

図 10 にハワイのマウイ島，**図 11** にイギリス南部のセルサイズが大きい場合，**図 12** にイギリス南部のセルサイズが小さい場合の結果を示す．それぞれの図の (a) は，分割したグリッドの位置，(b) は，本手法による海岸線を赤色の線で示している．ハワイのパラメータは，$\theta = 20$，$maxdist = 5$，$maxdel = 5$，イギリスのパラメータは，$\theta = 15$，$maxdist = 5$，$maxdel = 5$ とした．図 10 (b) では，島の西側は，おおむね海岸線に沿って線が引かれているが，東側は，ほとんど線が引かれていない．これは，写真の枚数が少なく，線を引く候補となる条件の閾値を超えていないことが原因である．また，海岸線から離れた内陸部に線が引かれている．現在の手法では，線の両端が隣接するセルの線とつながっていない場合のみ線の削除を行う．内陸部に残っている線は，2 つ以上のセルの線がつな

(a) 分割したグリッド
Position of grid.

(b) 本手法によって描かれた線
lines by our method.

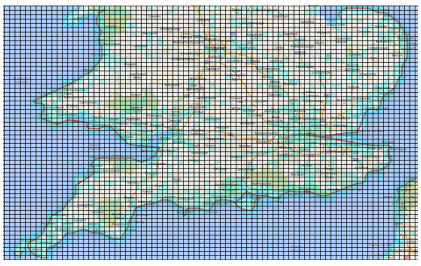
図 10　海岸線の抽出：ハワイのマウイ島付近
Fig. 10　Results of extracting coastlines at Maui Island.



(a) 分割したグリッド
Position of grid.

(b) 本手法によって描かれた線
lines by our method.

図 11　海岸線の抽出：イギリス南部（セルサイズ大）
Fig. 11　Results of extracting coastlines at the UK (large cell).



(a) 分割したグリッド
Position of grid.

(b) 本手法によって描かれた線
lines by our method.

図 12　海岸線の抽出：イギリス南部（セルサイズ小）
Fig. 12　Results of extracting coastlines at the UK (small cell).

がっているため，削除されていない．そのため，線を削除する条件の修正を検討する必要があると考えられる．

図 11 (b) と図 12 (b) を見ると，イギリスでもおおむね海岸線に沿って線が描けている．図 11 (b) では，線が交差している場所が存在する．これは，離れたセルをつなげる際に線端の向きだけを利用しており，つなぐ候補となる 2 点が向かい合ってさえいれば，その 2 点間に線があるかどうかを考慮していないためである．海岸線の場合は，交差する箇所は存在しないが，道路の交差点のような形状を抽出する場合，交差が生じるべきである．そのため，この問題は，セルサイズを小さくすることで対処を行う．

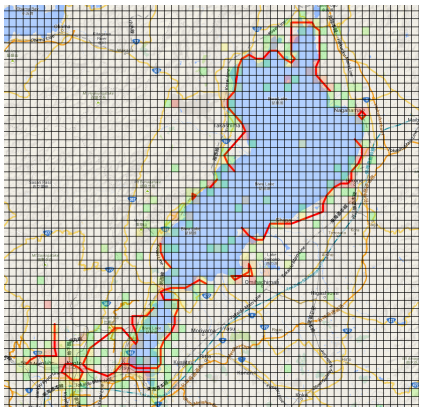
図 11 (b) と図 12 (b) を比較すると，図 12 (b) では，図 11 (b) にあった内陸部の線がなくなっており，海岸線の細部も再現できている．ウェーマス付近のポートランドやブリストル海峡のような細かい凹凸部分も再現できている（図 12 (b) 緑矢印）．しかし，南西の端のペンザンス付近は，線が途切れて島のように表示されている（図 12 (b) 青矢印）．セルサイズが小さいと内陸部の線が存在しない理由は次のように考えられる．セルサイズが大きいと，ある程度散らばった位置で撮影された写真が 1 つのセルにまとめられる．しかし，セルサイズが小さいと，写真が複数のセルに散らばるため，写真枚数が少ない地域は線を引く候補となる可能性が低い．ノイズとなる写真（e.g. 内陸部で撮影された beach と関係のない写真に「beach」とタグ付けされた写真）の撮影位置は，海岸付近で撮影された写真に比べると，あちこちに散らばって撮影されており，撮影位置は線状にはなっていないと考えられる．そのため，内陸部で線を引く候補となるセルになった場合でも，隣接するセルに線を引く候補となるセルが存在しないため線は引かれない．海岸線上に写真が少ない地域があった場合は，本手法の線を引く段階では線は途切れるが，離れた線をつなぐ際につなぐことができる．しかし，これは海岸線が直線である場合であり，半島になっている場所では，途切れた場所を最短距離でつなぐと半島部分が島のようになってしまう場合がある．そのため，図 11 (b) の南西部のような現象が起きると考えられる．

4.2.2　タグ「beach」以外のタグ

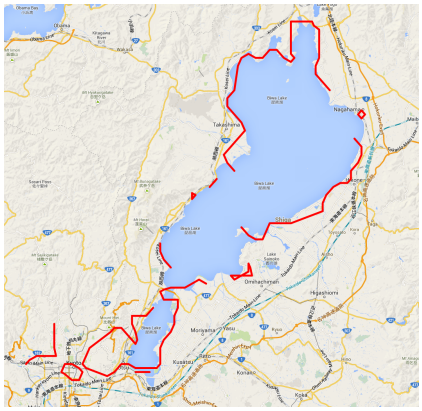
図 13 に琵琶湖，**図 14** に border，**図 15** に shinkansen の結果を示す．琵琶湖データセットのパラメータは，$\theta = 2$, $maxdist = 5$, $maxdel = 5$, border データセットのパラメータは，$\theta = 3$, $maxdist = 5$, $maxdel = 5$, shinkansen データセットのパラメータは，$\theta = 3$, $maxdist = 5$, $maxdel = 5$ とした．それぞれの (a) は，分割したグリッドの位置を，(b) は，本手法による海岸線を赤色の線で示している．**図 16** (a) に実際のヨーロッパの国境，図 16 (b) に実際の新幹線の路線図を示す．

図 13 (b) より，琵琶湖と陸の境界がおおむね描けているのが分かる．このデータセットの写真数は，表 3 に示すように 3,152 枚と少ないことに加え，ほとんどの写真は，琵琶湖の南西にある京都の都市部で撮影されている．これらのことから，写真枚数が少ない場所でも線が描けることが分かる．

図 14 (b) より，国境に沿って線が引かれている部分もあるが，国境周辺の地域以外にも線が多いことが分かる．これは，border という英単語は，国境以外にもあらゆる堺目として使われる言葉であり，国境よりも小さい州のような



(a) 分割したグリッド
Position of grid.



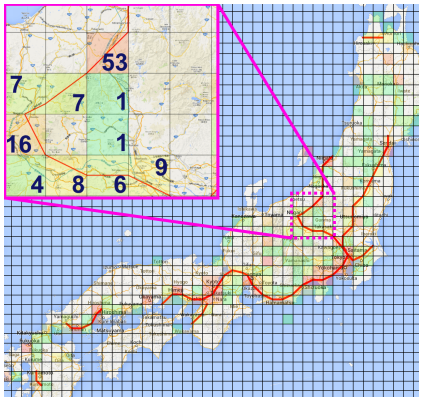
(b) 本手法によって描かれた線
lines by our method.

図 13　琵琶湖の結果
Fig. 13　Results of the Lake Biwa.

(a) 分割したグリッド
Position of grid.

(b) 本手法によって描かれた線
lines by our method.

図 14　タグ「border」の結果
Fig. 14　Results of the “border” tag.



(a) 分割したグリッド
Position of grid.

(b) 本手法によって描かれた線
lines by our method.

図 15　タグ「shinkansen」の結果
Fig. 15　Results of the “shinkansen” tag.

(a) ヨーロッパの国境
Border in Europe.

(b) 新幹線の路線図
Position of the shinkansen railway.

図 16　国境と新幹線の実際の位置
Fig. 16　Actual position of border and shinkansen railway.

地域の境界などにも用いられることが原因と考えられる．また，イギリスに多くの線が引かれている．これは，イギリスは，他のヨーロッパの国より写真数が多いため，写真数は少ないが国境付近で撮影されている写真に合わせてパラメータを指定すると，イギリスのほとんどのセルが候補セルとなり，線が描かれてしまうことが原因として考えられる．

図 15 (b) より，新幹線の路線がおおむね描けていることが分かる．大阪から和歌山，東京から千葉の線が引かれた場所は，新幹線は通っていないが，電車が通っており，電車を撮影した写真や，電車から撮影した写真にも誤って「shinkansen」とタグ付けしたことが原因として考えられる．図 15 (a) の左上は，上越新幹線と長野新幹線の路線がある地域の拡大図であり，数字は各セル内の写真数を表している．この地域では，2 つの新幹線の路線が 1 つの線として描かれている．原因としては，上越新幹線が通っているセルで撮影された写真は 1 枚であり，写真枚数が少ないため，適切な線を描くことができなかったと考えられる．これについては，今後，ジオタグ付き写真が増加することで対応できると考えられる．しかし，写真枚数が十分な地域では，写真数に対して適切と考えられる線が描けている．

これらのことから，本手法は，海岸線だけでなく海岸線以外のタグに対しても有効であることが分かる．そのため，タグが表す地理的特徴を線状で抽出する手法として，提案手法は有効であると考えられる．

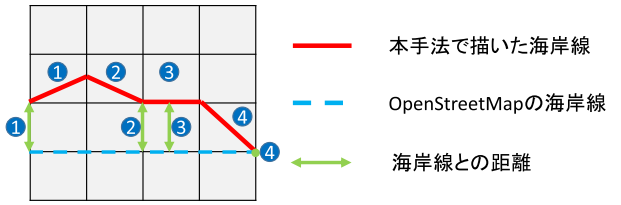


図 17　評価方法
Fig. 17　Evaluation method.

5. 評価

本章では，前章で示した提案手法の実行結果と実際の海岸線を用いて評価を行う．評価は，実際の海岸線と提案手法による海岸線の距離の算出，理想的な海岸線データに対して提案手法を適用した場合と実際のノイズが含まれたデータに対して提案手法を適用した場合の比較を行う．評価は，実際の海岸線データには，OpenStreetMap のデータを用い，タグ「beach」の結果に対して行う．

5.1 実際の海岸線との距離

提案手法によって描かれた線から実際の海岸線までの最短距離を求め，定量的に精度の評価を行う．この距離が近いほど，提案手法の性能は高いことを示している．

図 17 は，最短距離の計算の例である．赤色の実線が，提案手法による海岸線，青色の破線が OpenStreetMap の海岸線であり，緑色の矢印線が，提案手法の海岸線から OpenStreetMap の海岸線までの最短距離を示している．1，2，3 の最短距離は，それぞれセルの 1 辺の長さであり，4 の最短距離は，2 つの線が接しているため 0 m となる．

表 4，**表 5**，**表 6** に図 10 (b)，図 11 (b)，図 12 (b) で提案手法により引いた線と実際の海岸線との距離の分布を示す．表 4，表 5，表 6 のすべてにおいて，0 m～250 m に引かれた線が最も多い．250 m～1 km に引かれた線は，2 章で述べたように，約 20%の写真は，実際の海岸線から 500 m より離れた地点で撮影されており，線を引くために，用いた写真の撮影位置が海岸線から多少ずれていることや線を引く基準の位置が図 7 で説明したセル内の決まった 16 カ所であることが，主な原因として考えられる．また，1 km 以上にある線は，主に内陸部に引かれた線が原因であると考えられる．

表 5 と表 6 を比較すると，4.2 節で述べたように，セルサイズが小さい方が 0 m～250 m の割合が高く，高い性能であることが分かる．しかし，セルサイズの違いは，解像度の違い，つまり，Google Maps などのズームレベルにあたるもので，セルサイズが大きいほど粗くなっているが，セル内に 1 つの直線を引く条件において，適切な線が引かれている．**図 18** (a) と図 18 (b) は，ある海岸線に提案手法を適用した例を表している．図 18 (a) は，セルサイズが小さい場合，図 18 (b) は，セルサイズが大きい場合であり，赤色の実線が提案手法による線，青色の破線が実際の海岸線を表している．この海岸線では，図 18 (a) のように線が引けることが好ましい．しかし，図 18 (b) の提案手法による線は，突起部分を無視しているが，このセルサイズの場合においては最適な線である．図 17 による評価手法では，図 18 (a) は，11 個のセルに線が引かれており，すべての線が海岸線と接している，もしくは交差しているため，0 m の線が 11 本となる．そして，図 18 (b) は，0 m の線が 1 本となる．セルサイズが異なるので，線の数は異な

表 4　図 10 (b) の線と実際の海岸線との比較結果
Table 4　The shortest distance of actual coastline and lines of Fig. 10 (b).

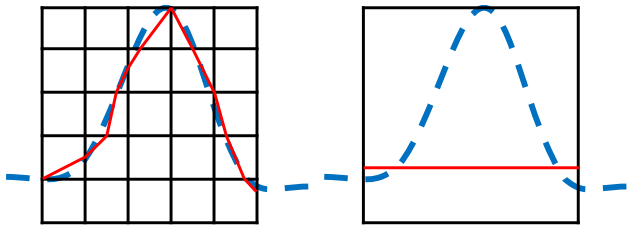
距離	線の数
0 m～250 m	42（75%）
250 m～500 m	4（7%）
500 m～750 m	6（10%）
750 m～1 km	2（3%）
1 km～	2（3%）

表 5　図 11 (b) の線と実際の海岸線との比較結果
Table 5　The shortest distance of actual coastline and lines of Fig. 11 (b).

距離	線の数
0 m～250 m	112（59%）
250 m～500 m	10（5%）
500 m～750 m	9（5%）
750 m～1 km	4（2%）
1 km～	55（29%）

表 6　図 12 (b) の線と実際の海岸線との比較結果
Table 6　The shortest distance of actual coastline and lines of Fig. 12 (b).

距離	線の数
0 m～250 m	174（73%）
250 m～500 m	22（9%）
500 m～750 m	14（6%）
750 m～1 km	10（4%）
1 km～	20（8%）



(a) セルサイズ小
Small cell.

(b) セルサイズ大
Large cell.

図 18　セルサイズと線が引かれる位置
Fig. 18　Size of cell and location of line.

るが，どちらも 0 m であり，この評価手法では，解像度の違いに影響されないことが分かる．

表 5 および表 6 を比較したときに，表 5 の方が性能が劣っているのは，1 km 以上の内陸部に引かれた線が原因である．そこで，表 5 および表 6 から，1 km 以上の内陸部と思われる線を除き，割合を再計算すると，それぞれの距離の割合は，ほぼ同じとなる．このことから，セルサイズが異なっていても，海岸線付近においては，高い精度で線が描けていることが分かる．

5.2　頑健性の評価

次に，ハワイの結果を用いて，ノイズを含むデータに対してどの程度有効であるかを示すため，人手で生成した理想的なデータセットとの比較を行った．人手で生成したデータセットは，2 つのデータセットを用意した．1 つは，Google Maps を用いて，島の周囲の海岸線から人手によって 200 カ所の緯度経度を取得して生成したデータセットである．緯度経度は，島の外周全体から 1 km〜2 km の間隔で取得し，実際のデータのような偏りがないようにした．データ数を 200 とした理由は，セルサイズと同程度の間隔で取得し，1 つのセル内に少なくとも 1 つのデータが位置するようにするためである．もう 1 つは，図 10 に用いた，実際の撮影位置から，人手で海岸線から離れた位置の写真を取り除いたものである．200 件のデータセットと異なり，実際の写真の撮影位置を使っているため，図 10 (b) で線が引かれていない地域には，ほぼ写真が存在しない．ここで，200 件のデータセットをデータセット A，人手で海岸線から離れた写真を除去したデータセットをデータセット B，範囲内の全写真を含むデータセットをデータセット C と呼ぶ．それぞれのデータセットの写真数は**表 7** のとおりである．

それぞれのデータセットに対して，提案手法を適用し，5.1 節に述べた方法で距離を求めて比較を行う．パラメータは，$maxdist = 5$，$maxdel = 5$ はすべてのデータセットで固定であり，写真総数が異なるため，θ がデータセットにより異なる．データセット C は，図 10 と同じ $\theta = 20$ とした．データセット A は，1 つのセル内に位置するデータが 1 つの場合があるため，$\theta = 1$ とした．データセット B は，データセット C と同じ $\theta = 20$ とした．

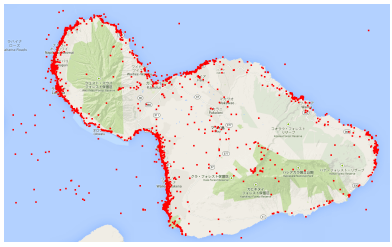
それぞれのデータセットの写真の撮影位置を**図 19** に，結果を**図 20** に示す．ただし，データセット C は，4 章と

表 7　データセットの写真数
Table 7　The number of photographs in each datasets.

データセット	写真数
A	200
B	10,085
C	12,747

(a) データセット A の写真の撮影位置
Photographic location of dataset A.

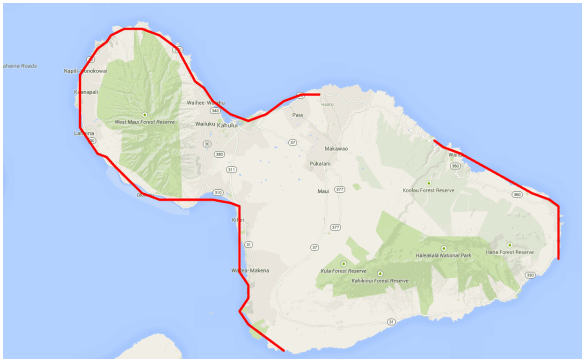
(b) データセット B の写真の撮影位置
Photographic location of dataset B.



(c) データセット C の写真の撮影位置
Photographic location of dataset C.

図 19　それぞれのデータセットの写真の撮影位置
Fig. 19　Photographic location of each dataset.

(a) データセット A の結果
Result of dataset A.

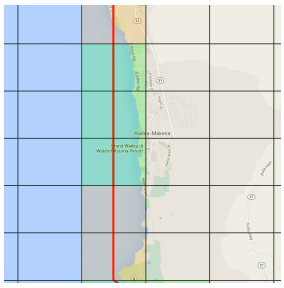


(b) データセット B の結果
Result of dataset B.

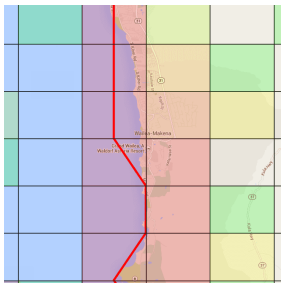
図 20　それぞれのデータセットの線の位置
Fig. 20　Location of line of each dataset.

表 8　データセットの写真数
Table 8　The number of photographs in each datasets.

距離	データセット A	データセット B	データセット C
0 m～250 m	72（86%）	42（84%）	42（75%）
250 m～500 m	7（8%）	6（12%）	4（7%）
500 m～750 m	2（2%）	1（2%）	6（10%）
750 m～1 km	1（1%）	1（2%）	2（3%）
1 km～	2（2%）	0（0%）	2（3%）



(a) データセット A の拡大図
Zoom of dataset A.



(b) データセット C の拡大図
Zoom of dataset C.

図 21　データセット A で 1 km となる原因
Fig. 21　Reason of exceeding 1 km in dataset A.

同じパラメータであるため，図 10 (b) が結果となる．それぞれの海岸線からの距離は，**表 8** のとおりである．データセット A では，島の外周をほぼ等距離で緯度経度を取得したため，島の全体が描けている．そのため，表 8 において，線の数の合計が最も多くなっている．データセット A，データセット B はノイズを含まないため，ともにデータセット C より良い結果となっている．しかし，さほど大きな違いはなく，ノイズを含むデータであっても，島の外周に写真が多くあれば正確に線が引けることが分かる．

データセット A は，海岸線上の緯度経度のみをデータセットとしているが，1 km 以上にも線が存在する．これは，海岸線がセルの端に位置したことが原因となっている．**図 21** (a) にデータセット A で 1 km 以上となった箇所を示す．写真の位置は，線が引かれた，色が塗られた縦に並んだセル上にあるため，セルの中央に縦に線が引かれている．この場合，セルの中心からセルの右端までが約 1 km であるため，1 km 以上の距離となった．図 21 (b) は，データセット C における同じ場所の結果である．このデータセットでは，明らかに海岸線上でない写真も含まれるため，海岸線付近の陸地の写真の影響で，一部陸地よりになっている．このため，データセット C では，この場所の線は，1 km 以上となっていない．データセット B においても，この場所の線はデータセット C と同じであり，1 km 以上は 0 となっている．このことから，正確に海岸線上にプロットされたデータセットよりも海岸線から離れた位置のデータを含むデータセットの方が良い結果となる場合があることが分かる．しかし，このようなケースは稀であり，海岸線から離れた位置のデータの影響で，実際の海岸線に近づくより実際の海岸線から遠ざかることの方が多い．また，セルサイズを小さくすることで，このようなことが起きた場合の精度の差を低減できる．

6.　関連研究

位置情報が付与されたデータを用いて地理的特徴を発見する研究は，ここ数年間でさかんに行われている．Sengstock ら [22] は，Flickr の写真のジオタグとタグを用いてランドマークのような地理的特徴を抽出する手法を提案した．Sakaki ら [23] は，ジオタグ付きツイートから地震や台風といった自然災害を抽出し，震央の推定を行った．また，Kamath ら [24] は，ジオタグ付きツイートを用いて，ハッシュタグの広がりと特定のハッシュタグが使われる地域を調査した．Yin ら [25] は，車や食べ物の分布といった地域ごとの特徴を Flickr のジオタグ付き写真を用いて発見した．これらの多くの既存研究では，地理的特徴として，領域を抽出している．抽出する地理的な特徴によっては，海岸線のように，領域より線が適している場合がある．そこで，我々は，地理的特徴として線を抽出する．これまでにも，大量の点から線を抽出する研究は，行われている．Liu ら [26] は，GPS 軌跡データから道の形状を線として抽出する手法を提案している．我々は，GPS の軌跡データではなく，写真の撮影位置から線を抽出する．そのため，データが連続していないことや，線が引かれるべき場所であるが，写真が撮影されていない地域があるなどの違いがあり，GPS 軌跡から線を引く手法を本研究に用いることは適切でない．

本研究では，写真に付与されているタグを用いて線を抽出しているが，適切でないと考えられるタグが付与されていることがある．すでに述べたように，海岸線から 500 m 以内で撮影された写真はおよそ 8 割である．また，Kennedy ら [27] は，タグに示された視覚的な特徴が写真内に写っている可能性はおよそ 5 割であることを示した．写真のタグや写真に写っているものを推定するため，タグと写真の意味の関係を推測する研究が行われている．たとえば，Hirota ら [28] は，画像特徴量を用いた写真の検索結果からタグを推定する手法を提案した．さらに，Lee ら [29] は，画像の類似度やタグの共起を使ってタグの改善を行った．我々の手法は，適切なタグが付与されている写真が多く，ノイズとなるタグが少ないほど精度が向上すると考えられる．また，写真の撮影地点の地名や写真の被写体など，タグが付与された理由を正しく分類することも精度の向上に寄与すると考えられる．

7.　おわりに

本論文では，海岸線付近で撮影された写真に付与されると考えられるタグが，実際に海岸線付近で撮影された写真

に付与されていることを示し，それらの写真から，海岸線を描くアルゴリズムを提案した．また，タグ「beach」が付与された写真を用いて，提案手法と実際の海岸線データとの比較を行い，提案手法の評価を行った．提案手法による海岸線の 64%～82%は，実際の海岸線から 500 m 以内に描けた．さらに，海岸線以外のタグについても，良好な結果を得られ，写真の撮影位置とタグからタグが表している地理的特徴を線状で抽出することができた．

今後の課題として，アルゴリズムの性能の向上と拡張があげられる．現在のアルゴリズムでは，グリッド内の写真の分布は利用していない．セル内の分布を用いて線の位置を調整することで，より性能が向上すると考えられる．また，候補セルの決定や基準線を引くときにセル内の写真数を用いているため，写真数が多い地域と少ない地域に影響を受ける．そのため，写真の絶対値ではなく，周辺の写真枚数に応じた条件に変更するなど，あらゆる分布に対応できるように修正する必要性があると考えられる．また，今回の実行結果では，パラメータは人目である程度良い結果を得られたものを選んだ．今後は，タグごとの検索結果の写真の枚数や分布によって最適なパラメータを調査する必要があると考えられる．さらに，現在の手法では，Y 字分岐路や，P 字の道路などの形状に対応していないため，複雑な形状にも対応させる必要がある．

参考文献

[1] Flickr, available from ⟨http://www.flickr.com/⟩ (accessed 2015-01-20).

[2] Panoramio, available from ⟨http://www.panoramio.com/⟩ (accessed 2015-01-20).

[3] 100,000,000 geotagged photos (plus), available from ⟨http://code.flickr.net/2009/02/04/100000000-geotagged-photos-plus/⟩ (accessed 2015-01-20).

[4] Zhang, H., Korayem, M., Crandall, D.J. and LeBuhn, G.: Mining Photo-sharing Websites to Study Ecological Phenomena, *Proc. 21st International Conference on World Wide Web*, pp.749–758 (2012).

[5] Thomee, B. and Rae, A.: Uncovering Locally Characterizing Regions within Geotagged Data, *Proc. 22nd International Conference on World Wide Web*, pp.1285–1296 (2013).

[6] Shirai, M., Hirota, M., Yokoyama, S., Fukuta, N. and Ishikawa, H.: Discovering Multiple HotSpots Using Geotagged Photographs, *Proc. 20th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pp.490–493 (2012).

[7] Hirota, M., Shirai, M., Ishikawa, H. and Yokoyama, S.: Detecting Relations of Hotspots using Geo-tagged Photographs in Social Media Sites, *Proc. Workshop on Managing and Mining Enriched Geo-Spatial Data*, pp.7:1–7:6 (2014).

[8] Crandall, D.J., Backstrom, L., Huttenlocher, D. and Kleinberg, J.: Mapping the world's photo, *Proc. 18th International Conference on World Wide Web*, pp.761–770 (2009).

[9] Kisilevich, S., Mansmann, F. and Keim, D.: P–DBSCAN: A density based clustering algorithm for exploration and analysis of attractive areas using collection of geo-tagged photos, *Proc. 1st International Conference on Computing for Geospatial Research & Application*, pp.38:1–38:4 (2010).

[10] Omori, M., Hirota, M., Ishikawa, H., Yokoyama, S.: Can geo-tags on Flickr Draw Coastlines?, *Proc. 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems* (2014).

[11] OpenStreetMap Data, available from ⟨http://openstreetmapdata.com/data/coastlines⟩ (accessed 2013-04-15).

[12] NOAA National Geophysical Data Center, available from ⟨http://www.ngdc.noaa.gov/mgg/shorelines/shorelines.html⟩ (accessed 2015-01-20).

[13] Parikh, M., Varma, T.: Survey on Different Grid Based clustering Algorithms, *International Journal of Advance Research in Computer Science and Management Studies*, Vol.2, Issue.2, pp.427–430 (2014).

[14] Zhao, Q., Shi, Y., Liu, Q., Fränti, P.: A Grid-growing Clustering Algorithm for Geo-spatial Data, *Pattern Recognition Letters* (2014).

[15] Edla, D. and Jana, P.: A grid clustering algorithm using cluster boundaries, *2012 World Congress on Information and Communication Technologies* (*WICT*), pp.254–259 (online), DOI: 10.1109/WICT.2012.6409084 (2012).

[16] Liu, H., Wei, L.-Y., Zheng, Y., Schneider, M. and Peng, W.-C.: Route Discovery from Mining Uncertain Trajectories, *2011 IEEE 11th International Conference on Data Mining Workshops* (*ICDMW*), pp.1239–1242 (online), DOI: 10.1109/ICDMW.2011.149 (2011).

[17] Shi, J., Mamoulis, N., Wu, D. and Cheung, D.W.: Density-based Place Clustering in Geo-social Networks, *Proc. 2014 ACM SIGMOD International Conference on Management of Data*, *SIGMOD '14*, New York, NY, USA, pp.99–110, ACM (online) DOI: 10.1145/2588555.2610497 (2014).

[18] Liu, Z., Yan, H. and Han, H.: Mining Large-Scale Social Images with Rich Metadata and Its Application, *Journal of Software*, Vol.7, No.4 (online) (2012).

[19] Twitter, available from ⟨https://twitter.com/⟩ (accessed 2015-01-20).

[20] OpenLayers, available from ⟨http://openlayers.org/⟩ (accessed 2015-01-20).

[21] Google Maps, available from ⟨https://maps.google.com/⟩ (accessed 2015-01-20).

[22] Sengstock, C. and Gertz, M.: Latent Geographic Feature Extraction from Social Media, *Proc. 20th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pp.149–158 (2012).

[23] Sakaki, T., Okazaki, M. and Matsuo, Y.: Earthquake shakes Twitter users: real-time event detection by social sensors, *Proc. 19th International Conference on World Wide Web*, pp.851–860 (2010).

[24] Kamath, K.Y., Caverlee, J., Lee, K. and Cheng, Z.: Spatio-temporal dynamics of online memes: a study of geo-tagged tweets, *Proc. 22nd international conference on World Wide Web*, pp.667–678 (2013).

[25] Yin, Z., Cao, L., Han, J., Zhai, C. and Huang, T.: Geographical topic discovery and comparison, *Proc. 20th International Conference on World Wide Web*, pp.247–256 (2011).

[26] Liu, X., Biagioni, J., Eriksson, J., Wang, Y., Forman, G. and Zhu, Y.: Mining large-scale, sparse GPS traces for map inference: comparison of approaches, *Proc. 18th*

ACM SIGKDD International Conference on Knowledge discovery and data mining, pp.669–677 (2012).
[27] Kennedy, L.S., Chang, S.-F. and Kozintsev, I.V.: To search or to label?: predicting the performance of search-based automatic image classifiers, *Proc. 8th ACM International Workshop on Multimedia information retrieval*, pp.249–258 (2006).
[28] Hirota, M., Fukuta, N., Yokoyama, S. and Ishikawa, H.: A Robust Clustering Method for Missing Metadata in Image Search Results, *Journal of Information Processing*, Vol.53, No.3, pp.537–547 (2012).
[29] Lee, S., De Neve, W. and Ro, Y.M.: Tag refinement in an image folksonomy using visual similarity and tag co-occurrence statistics, *Signal Processing: Image Communication*, Vol.25, No.10, pp.761–773 (2010).

大森 雅己

1990 年生．2013 年静岡大学情報学部卒業．同年同大学大学院情報学研究科情報学専攻に入学し，現在に至る．日本データベース学会学生会員．

廣田 雅春　（正会員）

1988 年生．2014 年静岡大学創造科学技術大学院情報科学専攻修了．首都大学東京大学院システムデザイン研究科日本学術振興会特別研究員（PD）．博士（情報学）．Web マイニング，マルチメディア，地理情報システムの研究に従事．電子情報通信学会，日本データベース学会，ACM 各会員．

石川 博　（フェロー）

首都大学東京システムデザイン学部教授．東京大学理学部卒業．富士通研究所，都立大学教授，静岡大学教授を経て 2013 年より現職．東京大学博士（理学）．データベース，データマイニング，ソーシャルメディア等の研究に従事．国際論文誌 ACM TODS，IEEE TKDE，国際会議 IEEE ICDE，VLDB 等論文多数．最近の著書に『ソーシャルビッグデータサイエンス入門』（コロナ社，2014），『Social big data mining』（CRC Press，2015）等．情報処理学会坂井記念特別賞，科学技術庁長官賞（研究功績者）受賞．情報処理学会フェロー，電子情報通信学会フェロー．ACM，IEEE 各会員．

横山 昌平　（正会員）

静岡大学情報学研究科講師．産業技術総合研究所，静岡大学情報学部助教を経て 2012 年より現職．2006 年東京都立大学大学院工学研究科修了，博士（工学）．データ工学，特にジオ・ソーシャルデータの分析に関する研究に従事．電子情報通信学会，日本データベース学会各正会員．情報処理学会論文誌データベース編集委員（幹事補佐）．

（担当編集委員　風間 一洋）