Can Geo-tags on Flickr Draw Coastlines?

Masaki Omori[1], Masaharu Hirota[2], Hiroshi Ishikawa[3], Shohei Yokoyama[1]
[1]Graduate School of Informatics, Shizuoka University, Hamamatsu, Japan
[2]Graduate School of Science and Technology, Shizuoka University, Hamamatsu, Japan
[3]Information and Communications Systems, Faculty of System Design, Tokyo Metropolitan University, Hino, Japan
{gs13008,dgs11538}@s.inf.shizuoka.ac.jp, ishikawa-hiroshi@sd.tmu.ac.jp, yokoyama@inf.shizuoka.ac.jp

ABSTRACT

Many photos shared on photo-sharing sites are annotated with tags and geo-tags. Some studies have demonstrated extraction of the geographical characterization which a tag represents as regions using those metadata. However, in some cases (e.g. coastline), a line is more suitable than a region as a geographical characterization of a tag. Therefore, we proposed a novel method to extract lines as a region as a geographical characterization. Results show that the distance of a coastline and many lines of our method is less than 500 m. Although, in this paper, only the coastline has been evaluated, this method is applicable to other tags as well.

Categories and Subject Descriptors

H.2.8 [Database Management]: Database Applications—Data mining, Spatial databases and GIS; H.1.2 [Models and Principles]: User/Machine Systems—Human information processing; H.3.1 [Information Storage and Retrieval]: Content Analysis and Indexing

General Terms

Algorithms

Keywords

social tagging, GIS, visualization, geographical characterization

1. INTRODUCTION

Photo-sharing sites such as Flickr [1] include many geotagged and tagged photos. Geo-tags are annotated by digital cameras and smart phones equipped with GPS, and tags are annotated by social tagging from many users. Those metadata are used to extract geographical characterization.

[1]http://www.flickr.com/

SIGSPATIAL '14, November 04 - 07 2014, Dallas/Fort Worth, TX, USA
Copyright 2014 ACM 978-1-4503-3131-9/14/11 ...$15.00
http://dx.doi.org/10.1145/2666310.2666436

Table 1: Distribution of photographic locations from an actual coastline.

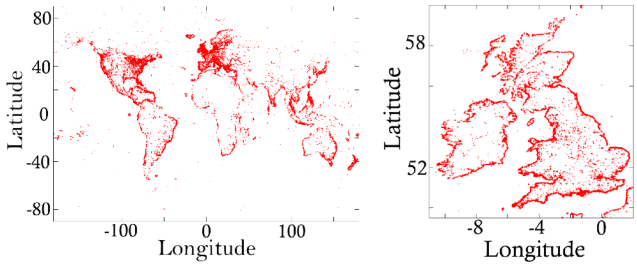
Tag	#photos	$\leq 100m[\%]$	$\leq 500m[\%]$
beach	2,488,923	51.34	80.44
sea	1,689,924	48.20	76.77
coastline	60,245	56.25	82.54
shoreline	47,114	51.92	70.93

Spatial knowledge such as locations in which people are interested and other aspects of the geographical environment can be ascertained from geographical characterization. Some studies have proposed a method to discover areas that a tag represents and hotspots, defined as places where many people take photos, using photos with tags and geo-tags [1], [2], [3], [4], [5], [6]. We also have studied a method to extract hotspots and to compute the relation of hotspots [7]. However, these approaches are unsuitable in all cases because some tags should not form regions but lines (e.g. "coastline" and "railway"). Therefore, we propose another approach of extracting geographical characteristics as a line from photo-sharing sites.

Our hypothesis is that allocation results of photos with a particular tag on a map based on its location visualizes the geographical characteristics such as the shape or split of the tag. Also, we inferred that photos tagged with "coastline" tag are almost all taken near a coastline, and the coastline can be extracted from the "coastline" tag. To verify this hypothesis, we attempt to extract a coastline from geo-tags and tags annotated to photos on Flickr, and evaluate it. We specifically examine the coastline because the coastline shape is clear and high-resolution data of coastlines are released on the Internet. Therefore, we evaluate the accuracy of our method quantitatively using these actual coastline data. In this paper, although we evaluate only coastlines, this method is applicable to other tags.

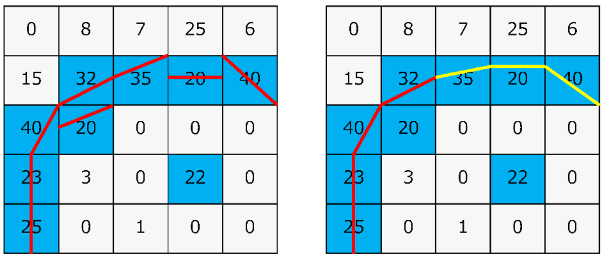
2. PRELIMINARY EXPERIMENT

To draw a coastline using locations where photos with the "coastline" tag were taken, the closeness of an actual coastline and locations where photos were taken are crucially important. Therefore, we investigate and quantify the geographical credibility of social tagging, defined as the distance between photographic locations with a tag and a place where the tag represents, using a coastline data as preliminary experiments. Here, high credibility means that most photos annotated with "coastline" tag are taken near coastlines.



(a) The whole world. (b) Whole the UK.

Figure 1: Photographic location tagged with "beach".



(a) Example of unconnected lines. (b) Result of connecting lines from Figure 2(a).

Figure 2: Example of the procedure used to draw the coastline.

We collected about 200 million geo-tagged photos from Flickr, and picked out photos annotated with a tag related to coastline. Also, we used OpenStreetMap data [2] as actual coastline data. Table 1 presents results that reveal distances between the photographic location and the actual coastline location. The "beach" tag has the most photos and the high rate within 500 m in these tags. Therefore, we plotted the position of photos with the "beach" tag in Figure 1. We can realize the approximate shape of the coastline. Although a shape around the UK is not clear in Figure 1(a), Figure 1(b), extended the UK, represents the clear shape of the UK. Therefore, we extract actual coastlines using photos tagged with the "beach" tag, because it has sufficient quantity and credibility.

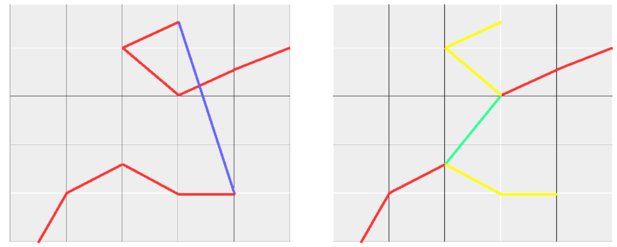
3. PROPOSED METHOD

Our method includes three phases. (1) We divide the area which we draw coastlines to grid, and draw coastlines, which is named a remarkable coastline, in each cell. (2) We connect unconnected coastlines between adjacent cells. (3) We complement the remaining unconnected coastlines that exist in isolated cells.

At first, remarkable coastlines are drawn. A selected area defined by a user is divided into a grid, and remarkable cells that definitely contain a coastline are detected from all cells. Then, a crossing line over the cell is calculated using geo-tags in each remarkable cell.

Algorithm 1 shows a pseudo-code of a method to decide remarkable cells and to draw remarkable coastlines. In addition, Figure 2(a) portrays an example of remarkable coastlines using this method ($\theta = 20$). A number shows the number of photos included in a cell. A blue cell is a remarkable cell. Remarkable cells are calculated using the difference of

[2]http://openstreetmapdata.com/data/coastlines



(a) Bad example of connecting lines between remote cells. (b) Example of connecting lines between remote cells with fixing.

Figure 3: Connecting coastlines between remote cells.

Algorithm 1 Drawing remarkable coastlines.

```
argument: θ /* a threshold value to decide remarkable
    cells */
remarkable_cells ⇐ φ
foreach cellα ∈ all cells do
  foreach cellβ ∈ around4(cellα) do
    if num(cellα) − num(cellβ) > θ then
      remarkable_cells ⇐ remarkable_cells ∪ cellα
    end if
  end for
end for

foreach cellC ∈ remarkable_cells do
  count ⇐ 0
  foreach cellA ∈ around8(cellC) do
    if is_remarkable_cell(cellA) then
      count ⇐ count + 1
    end if
  end for
  if count > 0 then
    cellα ⇐ max(around8(cellC))
    cellβ ⇐ max(around8(cellC)−cellα−around4(cellα))
    if num(cellβ) = 0 then
      cellB ⇐ opposite(cellC,cellα)
    end if
    drawline(center(cellC,cellα), center(cellC,cellβ))
  end if
end for
```

the number of photos between adjacent cells. A remarkable coastlines are drawn from the first cell in surrounding 8-cells towards the second cell in surrounding 5-cells, except the first cell and adjacent cells of the first cells, if even one has a remarkable cell in surrounding 8-cells. Here, a reason of excepting adjacent cells of first cells is to adapt bias of number of photos. Additionally, a remarkable coastline is not drawn if all of around 8-cells are not remarkable cells.

At Second, we connect unconnected remarkable coastlines. Figure 2(b) shows a result of connecting remarkable coastlines of Figure 2(a). Algorithm 2 shows pseudo-code of a method to connect unconnected remarkable coastlines between adjacent cells. Basically, end points of lines on a shared side are connected by shifting both end points to a middle point of those. However, if an end point is located on a vertex on a cell, then two cells that have a common side with the cell are compared by the numbers of photos, and a larger one is regarded as a side of the cell. Subsequently, remarkable coastlines for which both end points are uncon-

Algorithm 2 Connecting coastlines between adjacent cells.

```
foreach cellα ∈ all cells that have a line do
  foreach pointα ∈ both end points on cellα do
    cells ⇐ φ
    foreach cellA ∈ around(cellα) do
      if is_exist_point_in_cell(pointα,cellA) then
        cells ⇐ cells ∪ cellA
      end if
    end for
    cellβ ⇐ max(cells)
    side ⇐ get_side_between_cells(cellα,cellβ)
    points ⇐ get_points(side)
    pointα ⇐ center(points)
    redrawline(cellα)
  end for
end for

foreach lineα ∈ all lines for which both end points are
  not connected do
  delete(lineα)
end for
```

nected are deleted.

Finally, remaining unconnected coastlines that exist in isolated cells are complemented. A fundamentally important idea is that unconnected end points that are closed are connected sequentially. However, some cases are not suitable this method as shown Figure 3(a). Therefore, we fix a coastline as shown Figure 3(b). In Figure 3, a blue line shows a line of basic idea. Yellow lines show deleted lines by fixing. A green line shows a line with fixing.

Algorithm 3 shows a pseudo-code of a method to complement remaining unconnected coastlines that exist in isolated cells. A fixing method uses a direction of a line. A method for fixing is the following when $lineA$ and $lineB$ are connected. Herein, an end point of $lineA/lineB$ is named epA/epB. A cell that has $lineA/lineB$ is named $cellA/cellB$. (1) Directions of epA and epB are computed. Directions are defined as the side on which the end point of the line exists. For example, if epA is located on the left side of $cellA$, then the direction of epA is "left". In addition, if epB is located at the vertex of upper left, then the direction of epB is "top-left". (2) If the direction of epA and epB are facing each other, then $lineA$ and $lineB$ are connected by connecting epA and epB, and fixing is finished. Facing each other is defined as epA = "left" and epB = "right", or epA = "top-right" and epB = "bottom-left". (3) Repeat the procedure from (1) with shortening $lineA$ and $lineB$ by one cell alternately.

4. EXPERIMENT AND RESULTS

To examine the performance of our method, we collect 218,566 photos tagged with "beach" in the UK. Also, to examine whether our method is applicable besides a coastline, we collect 6,955 tagged with "shinkansen", which is a high-speed railway in Japan. Photos whose latitude or longitude is an integer value was excluded from these dataset because probably it is wrong data.

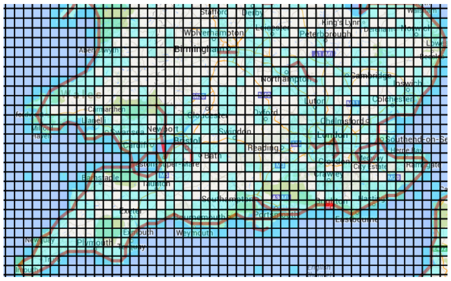
Figure 4, Figure 5 and Figure 6 show results on Google Maps [3]. In Figure 4 and Figure 5, each (a) shows a grid position, and (b) shows lines of our method in red lines at the UK. Figure 5 are obtained by reducing the grid size of Figure 4. The parameters are $\theta = 15$, $maxdist = 5$, $maxdel = 5$. In both Figure 4 and Figure 5, coastlines are almost reproduced. Comparing Figure 4 and Figure 5, Figure 5 reproduces details of shapes of a coastline more clearly than Figure 4 especially green arrow in Figure 5(b). However, the peninsula is extracted as an island by breaking lines (blue arrow). This is caused that our method connects coastlines that are broken off by the shortest distance.

In Figure 6, (a) shows the position of the railway, and (b) shows lines of our method in Japan. The parameters are $\theta = 3$, $maxdist = 5$, $maxdel = 5$. Although the number of photos are a few, most lines trace actual shinkansen railways. Part of lines are located at a place where there is not a shinkansen but there is a train. This is caused that some people annotate the "shinkansen" tag to a photo which reflects a train. In this paper, parameters were chosen by comparing and seeing only some patterns. Therefore, we

[3]https://maps.google.com/

Algorithm 3 Complement remaining unconnected coastlines.

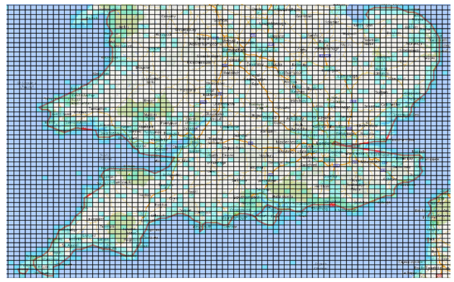
```
argument: maxdist /* an integer value to decide maxi-
  mum distance to permit connection */
argument: maxdel /* an integer value to decide maxi-
  mum number of deletable lines */
argument: disttable /* a table that includes all pairs of
  end points and distance between unconnected lines */
sort_order_by_distance_ascending(disttable)
foreach row ∈ disttable
 p1 ⇐ get_first_point(row)
 p2 ⇐ get_second_point(row)
  if not_connect(p1) and not_connect(p2) and
  get_distance(row) < maxdist then
  for 1 to maxdel do
   if try_connect(p1,p2) then
    break
   end if
   tmp1 ⇐ p1
   p1 ⇐ get_connect_point(get_another_point(p1))
   if try_connect(p1,p2) then
    break
   end if
   tmp2 ⇐ p1
   p1 ⇐ tmp1
   p2 ⇐ get_connect_point(get_another_point(p2))
   if try_connect(p1,p2) then
    break
   end if
   p1 ⇐ tmp2
  end for
 end if
end for

declare function try_connect (p1,p2)
  if aim_to(p1,p2) and aim_to(p2,p1) then
    delete_connect_line(get_connect_point(p1))
    delete_connect_line(get_connect_point(p2))
    redrawline(p1,p2)
    return true
  end if
  return false
end function
```



(a) Position of grid

(b) Our method

Figure 4: Results on coastlines (large grid).



(a) Position of grid

(b) Our method

Figure 5: Results on coastlines (small grid).

(a) Actual lines

(b) Our method

Figure 6: Results on shinkansen.

Table 2: Distance between actual coastline and lines of 4(b).

Distance	#lines in Fig. 4	#lines in Fig. 5
$0 - 250m$	112 (59%)	174 (73%)
$250 - 500m$	10 (5%)	22 (9%)
$500 - 750m$	9 (5%)	14 (6%)
$750m - 1km$	4 (2%)	10 (4%)
$1km-$	55 (29%)	20 (8%)

should find the most suitable parameter, because parameters are not optimized strictly.

5. EVALUATION

We compare the actual coastlines and lines of our method, and evaluate the accuracy of our method quantitatively. We calculate the shortest distance between a line of our method and the nearest actual coastline from our method every each line in a cell. Here, we use OpenStreetMap data as the actual coastline data.

Table 2 shows the distributions of the shortest distance in Figure 4(b) and Figure 5(b). The shortest distance of some lines is over 250 m. The cause is described in Section 2: about 20 % photos are located at far away (over 500 m) from the actual coastline. In other words, the dataset to draw coastlines includes some error. This fact results from lines that are drawn inland in Figure 4(b). However, lines drawn at from 0 m to 250 m are the most numerous in both results. This shows that our method has good performance.

6. CONCLUSION

As described in this paper, we proposed and evaluated a novel method to draw lines from many points as another approach of extracting geographical characterization. Then we demonstrate that most coastlines by our method are close to the actual coastline, and our method can extract geographical characterization that a tag represents from geo-tags and tags. As future works, we will improve performance and adapt to complex shaped lines (e.g. Y-split) by discovering suitable parameter for each tag and improving algorithm.

7. REFERENCES

[1] H. Zhang, M. Korayem, and D. J. Crandall, Mining Photo-sharing Websites to Study Ecological Phenomena, In Proceedings of the 21st International Conference on World Wide Web, pages 749-758, 2012

[2] B. Thomee and A. Rae, Uncovering Locally Characterizing Regions within Geotagged Data, In Proceedings of the 22nd International Conference on World Wide Web, pages 1285-1296, 2013

[3] C. Sengstock and M. Gertz, Latent Geographic Feature Extraction from Social Media, In Proceeding of the 20th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, pages 149-158, 2012

[4] T. Sakaki, M. Okazaki, and Y. Matsuo, Earthquake shakes Twitter users: real-time event detection by social sensors, In Proceedings of the 19th International Conference on World Wide Web, pages 851-860, 2010

[5] K. Y. Kamath, J. Caverlee, K. Lee, and Z. Cheng, Spatio-temporal dynamics of online memes: a study of geo-tagged tweets, In Proceedings of the 22nd International Conference on World Wide Web, pages 667-678, 2013

[6] Z. Yin, L. Cao, J. Han, C. Zhai, and T. Huang, Geographical topic discovery and comparison, In Proceedings of the 20th International Conference on World Wide Web, pages 247-256, 2011

[7] M. Shirai, M. Hirota, S. Yokoyama, N. Fukuta, and H. Ishikawa, Discovering Multiple HotSpots Using Geo-tagged Photographs, In Proceedings of the 20th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, pages 490-493, 2012