G-RoI: Automatic Region-of-Interest detection driven by geotagged social media data

LORIS BELCASTRO, FABRIZIO MAROZZO, DOMENICO TALIA and PAOLO TRUNFIO,
DIMES, University of Calabria

Geotagged data gathered from social media can be used to discover interesting locations visited by users called Places-of-Interest (PoIs). Since a PoI is generally identified by the geographical coordinates of a single point, it is hard to match it with user trajectories. Therefore, it is useful to define an area, called *Region-of-Interest* (*RoI*), to represent the boundaries of the PoI's area. *RoI mining* techniques are aimed at discovering Regions-of-Interest from PoIs and other data. Existing RoI mining techniques are based on three main approaches: predefined shapes, density-based clustering and grid-based aggregation. This paper proposes *G-RoI*, a novel RoI mining technique that exploits the indications contained in geotagged social media items to discover RoIs with a high accuracy. Experiments performed over a set of PoIs in Rome and Paris using social media geotagged data, demonstrate that G-RoI in most cases achieves better results than existing techniques. In particular, the mean F_1 score is 0.34 higher than that obtained with the well-known DBSCAN algorithm in Rome RoIs and 0.23 higher in Paris RoIs.

CCS Concepts: •**Human-centered computing** → **Social network analysis;** •**Information systems** → *Collaborative and social computing systems and tools; Web and social media search;*

Additional Key Words and Phrases: Places-of-Interest, Regions-of-interest, Geotagged social media, Social network analysis, RoI mining

ACM Reference Format:
Loris Belcastro, Fabrizio Marozzo, Domenico Talia and Paolo Trunfio, 2016. G-RoI: Automatic Region-of-Interest detection driven by geotagged social media data. *ACM Trans. Knowl. Discov. Data.* V, N, Article A (October 2017), 21 pages.
DOI: http://dx.doi.org/10.1145/0000000.0000000

1. INTRODUCTION

The widespread use of social media makes it possible to extract very useful information to understand the behavior of large groups of people. This is fostered by the large use of mobile phones and location-based services, through which millions of people every day access social media services and share information about the places they visit. In fact, data gathered from social media, such as posts from Twitter and Facebook or photos from Instagram and Flickr, are frequently geotagged. Geotagging is the process of adding geographic metadata (e.g., longitude/latitude coordinates) to text, photos or videos. It allows to locate the exact physical origin of shared information.

One of the leading trends in social media research is the analysis of geotagged data to determine if users visited or not interesting locations (e.g., touristic attractions, shopping malls, squares, parks), often called Places-of-Interest (PoIs). Since a PoI is generally identified by the geographical coordinates of a single point, it is hard to match it with user trajectories. For this reason, it is useful to define the so-called *Region-of-Interest* (*RoI*) representing the boundaries of the PoI's area [de Graaff et al.

Author's addresses: L. Belcastro and F. Marozzo and D. Talia and P. Trunfio, DIMES, University of Calabria, Rende (CS), Italy. Email:{lbelcastro, fmarozzo, talia, trunfio}@dimes.unical.it

2013]. The analysis of user trajectories through RoIs is highly valuable in many scenarios, e.g.: tourism agencies and municipalities can discover the most visited touristic places and the time of year when such places are visited [Bermingham and Lee 2014][Kurashima et al. 2010]; transport operators can discover the places and routes where is it more likely to serve passengers [Yuan et al. 2011] and crowed areas where more transport facilities need to be allocated [You et al. 2014].

RoI mining techniques are aimed at discovering Regions-of-Interest from PoIs and other data. Existing RoI mining techniques can be grouped into three main approaches: *predefined shapes* [de Graaff et al. 2013], *density-based clustering* [Zheng et al. 2012] and *grid-based aggregation* [Cai et al. 2014]. Predefined shapes techniques use fixed shapes, such as circles or rectangles, to represent RoIs. In many cases, the use of a predefined shape represents a naïve solution to the RoI mining problem, because a predefined shape is not able to handle PoIs having RoIs with different sizes and shapes. Density-based clustering techniques identify RoIs by clustering the data points according to a density criterion (i.e., number of data points per unit area). Such kind of algorithms are widely used because they are able to reach good results in many cases. However, density-based techniques may fail to distinguish regions that are very close to each other or that have different density. Grid-based aggregation techniques discretize the area in a regular grid and then aggregate the grid cells so as to form a RoI. The grid cells can be aggregate using different aggregation policies. Such kind of algorithms is very sensitive to parameters setting. Thus, may be hard to find a setting for identifying multiple RoIs with different characteristics in the same area.

This paper presents a novel RoI mining technique, called *G-RoI*, which differs from the existing approaches mentioned earlier as it exploits the indications contained in geotagged social media items (e.g. tweets, posts, photos or videos with geospatial information) to discover the RoI of a PoI with a high accuracy. Given a PoI p identified by a set of keywords, a geotagged item is associated to p if its text or tags contain at least one of those keywords. Starting from the coordinates of all the geotagged items associated to p, *G-RoI* calculates an initial convex polygon enclosing all such coordinates, and then iteratively reduces the area using a density-based criterion. Then, from all the convex polygons obtained at each reduction step, *G-RoI* adopts an area-variation criterion to choose the polygon representing the RoI for p.

Many experiments have been performed to assess the accuracy of G-RoI over real geotagged items extracted from Flickr, one of the most popular photo-sharing social media. The experimental results show that G-RoI is more accurate in identifying RoIs than existing techniques. Over a set of 24 PoIs in Rome, G-RoI achieves better results than existing techniques in 19 cases, with a mean precision of 0.78, a mean recall of 0.82, and a mean F_1 score of 0.77. In particular, the mean F_1 score of G-RoI is 0.34 higher than that obtained with the well-known DBSCAN algorithm. Further experiments have been performed over a set of 24 PoIs in Paris. Also in this case, G-RoI achieved best results in 18 cases, with a mean precision of 0.81, a mean recall of 0.66, and a mean F_1 score of 0.70 (0.23 higher than that obtained with DBSCAN). For the purpose of reproducibility, an open-source version of G-RoI and all the input data used in the experiments are available at https://github.com/scalabunical/G-RoI.

The remainder of the paper is organized as follows. Section 2 introduces the main concepts and the problem statement. Section 3 discusses related work. Section 4 describes the proposed methodology. Section 5 compares the performance of G-RoI with the main techniques in literature. Finally, Section 6 concludes the paper.

2. PROBLEM DEFINITION

A *Place-of-Interest* (*PoI*) is a specific location that someone finds useful or interesting. Generally, PoIs refer to business locations (e.g., shopping malls) or tourist attractions (e.g., squares, museums, theaters, bridges). PoIs are also named as *Point-of-Interest*.

For analyzing users' behavior, it is useful to understand whether a user visited or not a PoI. Since information on a PoI is generally limited to an address or to GPS coordinates, it is hard to match trajectories with PoIs. For this reason, it is useful to define the so-called *Region-of-Interest* (*RoI*) representing the boundaries of the PoI's area [de Graaff et al. 2013].

RoIs can be defined as "spatial extents in geographical space where at least a certain number of user trajectories pass through" [Giannotti et al. 2007]. Thus, RoIs represent a way to partition the space into meaningful areas and, correspondingly, to associate a label to a place. In literature, RoIs are also named as *regions of attraction* [Zheng et al. 2012] or *frequent (dense) regions* [Altomare et al. 2016].

A *geotagged item* is a piece of information (e.g. tweet, post, photograph or video) to which geospatial information were added. Specifically, a geotagged item g includes the following features:

- *text*, containing a textual description of g.
- *tags*, containing the tags associated to g.
- *coordinates* consists of *latitude* and *longitude* of the place from where g was created.
- *userId*, identifying the user who created g.
- *timestamp*, indicating date and time when g was created.

A geotagged item can be associated to a PoI $\mathcal{P}$ if its text or tags refer to $\mathcal{P}$. The goal of G-RoI is finding a suitable RoI $\mathcal{R}$ that describes the boundaries of $\mathcal{P}$'s area, by analyzing a set of geotagged items associated to $\mathcal{P}$.

3. RELATED WORK

Existing techniques to find RoIs can be grouped into three main approaches: *predefined shapes*, *density-based clustering* and *grid-based aggregation*. Table I reports approaches, algorithms, and goals of the main related work.

Predefined shapes. This approach uses predefined shapes (circles, rectangles, etc.) to represent RoIs. For example, Kisilevich et al. [Kisilevich et al. 2010a] define RoIs as circles of fixed radius centered on a set of PoIs whose center coordinates are known. Spyrou and Mylonas [Spyrou and Mylonas 2016] used circular RoIs to extract popular touristic routes from Flickr. Specifically, circular shapes are used to translate a trajectory of geospatial points into a sequence of RoIs. Cesario et al. [Cesario et al. 2015] used rectangles to define RoIs representing stadiums for a trajectory mining study. In particular, the RoI of a stadium is the smallest rectangle enclosing the stadium's area. De Graaff et al. [de Graaff et al. 2013] use Voronoi tessellations [Voronoi 1908] to define RoIs starting from a set of geographical coordinates representing PoIs.

Density-based clustering. With this approach, RoIs are obtained by clustering a set of geographical locations. For instance, Crandall et al. [Crandall et al. 2009] used the Mean shift clustering algorithm [Cheng 1995] to group the locations of a set of Flickr photos. The RoI is the polygon enclosing the cluster points. Zheng et al. [Zheng et al. 2012] used DBSCAN [Ester et al. 1996] to discover tourist attraction areas from a set of Flickr photos. DBSCAN was adopted for three main reasons: i) it tends to identify regions of dense data points as clusters; ii) it supports clusters with arbitrary shape; iii) it has a good efficiency on large-scale data. DBSCAN was also used by Altomare et al. [Altomare et al. 2016], with the goal of detecting the regions that are more densely visited based on data from GPS-equipped taxis. Kisilevich et al. [Kisilevich

Table I. Comparison with related algorithms.

Related work	Approach	Algorithm	Goal
Kisilevich et al. [Kisilevich et al. 2010a]	Pred. shapes	Circle with fixed radius	Mine travel sequences from Flickr photos
Spyrou-Mylonas [Spyrou and Mylonas 2016]	Pred. shapes	Circle with fixed radius	Extract popular touristic routes from Flickr photos
Cesario et al. [Cesario et al. 2015]	Pred. shapes	Rectangle enclosing PoIs	Trajectory mining from Twitter data
De Graaff et al. [de Graaff et al. 2013]	Pred. shapes	Voronoi tessellations	RoI extraction from cadastral data
Crandall et al. [Crandall et al. 2009]	Density	Mean shift clustering	Organize a large collection of geotagged Flickr photos
Zheng et al. [Zheng et al. 2012]	Density	DBSCAN	Discover interesting places from Flickr photos
Altomare et al. [Altomare et al. 2016]	Density	DBSCAN	Detect RoIs based on data from GPS-equipped taxis
Kisilevich et al. [Kisilevich et al. 2010b]	Density	P-DBSCAN	Discover attractive areas from collections of Flickr photos
Giannotti et al. [Giannotti et al. 2007]	Grid	Popular Regions	Mine rectangular RoI shapes from trajectory data
Cai et al. [Cai et al. 2014]	Grid	Slope RoI mining	Mine arbitrary RoI shapes from Flickr trajectory data
Cesario et al. [Cesario et al. 2016]	Grid	Grid cell aggregation	Discover mobility patterns from Instagram photos
Shi et al. [Shi et al. 2014]	Grid	DCPGS-G	Mine RoIs from historical geo-social networks

et al. 2010b] used a variant of DBSCAN, named P-DBSCAN, to cluster photos taking into account the neighborhood density (i.e., the number of distinct photo owners in the neighborhood) and exploiting the notion of adaptive density for fast convergence towards high density regions. Density-based approaches need a method to assign a meaning to each RoI found. There are different ways to perform this task. Zheng et al. [Zheng et al. 2012] and Yin et al. [Yin et al. 2011] assign a name to each cluster by taking the most frequent keyword in the geotagged items. Ferrari et al. [Ferrari et al. 2011a] automatically associate to each RoI the zip code of the data points in the cluster center.

Grid-based aggregation. This approach discretizes the area under analysis in a regular grid and extract RoIs by aggregating the grid cells. For example, Giannotti et al. [Giannotti et al. 2007] divide an area into grid cells and then count the trajectories passing through each cell. Grid cells whose counters are above a certain threshold are expanded to form rectangular shaped RoIs. Cai et al. [Cai et al. 2014] argued that rectangular expansion produces RoIs that may contain uninteresting low-density cells. For this reason, they proposed a hybrid grid-based algorithm, called Slope RoI, to mine arbitrary RoI shapes from trajectory data. Cesario et al. [Cesario et al. 2016] split the EXPO 2015 area in a grid and associated grid cells to PoIs representing pavilions, in order to discover the behavior and mobility patterns of users inside the exhibition. Shi et al. [Shi et al. 2014] map geotagged data into grid cells, and then group the cells taking into account spatial proximity and social relationship between places.

The proposed G-RoI technique does not belong to the approaches described earlier and it differs from them in three main respects:

- Differently from approaches using predefined shapes, G-RoI defines RoIs as polygons that are more accurate to model the variety of shapes a PoI can have.
- Density- and grid-based approaches may have troubles in distinguishing RoIs associated to PoIs that are very close to each other [Cai et al. 2014]. In fact, these approaches cluster data points (or aggregate cells) based on their proximity, even if they belong to different PoIs that are close to each other. As a result, two or more adjacent PoIs may be associated to the same RoI. In contrast, G-RoI accurately iden-

tifies different RoIs even in the presence of adjacent PoIs, as demonstrated by the experimental results presented in Section 5.
- Density- and grid-based approaches algorithms depend on the setting of multiple parameters (e.g., *eps* and *minNumPoints* for DBSCAN, *cell size* and *minimum support* for Slope RoI). For this reason, it is not easy to find parameters that produce accurate RoIs over different locations with a variety of shapes and data points distributions. In contrast, as shown in Section 5, G-RoI is accurate in identifying RoIs over locations characterized by a variety of shapes and data points distributions, using always the same value for its configuration parameter (a distance threshold between 0 and 1).

Summarizing, the main advantages of G-RoI with respect to the other techniques can be outlined as follows: i) G-RoI exploits data preprocessing based on keyword selection, which allows it to deal with more precise and clearer input data points; ii) G-RoI exploits a density-based criterion to identify the set of candidate RoIs and an area-variation criterion for choosing the most accurate RoI, which allow it to capture regions of interests independently from their density and shape.

Out of the above comparison are all the works that aggregate social geotagged data into regions defined either manually or through external services [Chaniotakis and Antoniou 2015; Ferrari et al. 2011b]: *manually* defining the boundaries of the PoIs (e.g., as polygons on a map); ii) *automatically*, using public web services (e.g., OpenStreetMap[1]) that provide the geographical boundaries of a place given its name.

4. METHODOLOGY

Let a PoI $\mathcal{P}$ be identified by one or more keywords $K = \{k_1, k_2, ...\}$. Let G_{all} be a set of geotagged items. Let $G = \{g_0, g_1, ...\}$ be the subset of G_{all}, obtained by applying a *G-RoI preprocessing* procedure that selects from G_{all} only the geotagged items associated to $\mathcal{P}$, i.e., the text or tags of each $g_i \in G$ contains at least one keyword in K. Let $C = \{c_0, c_1, ...\}$ be a set of coordinates, where c_i represents the coordinates of $g_i \in G$. Thus, every $c_i \in C$ represents the coordinates of a location from which a user has created a geotagged item referring to $\mathcal{P}$. Let cp_0 be a convex polygon enclosing all the coordinates in C, obtained by running the convex hull algorithm [Barber et al. 1996] on C, described by a set of vertices $\{v_0, v_1, ...\}$.

To find the RoI $\mathcal{R}$ for $\mathcal{P}$, the G-RoI algorithm uses two main procedures:

— *G-RoI reduction*. Starting from cp_0, it iteratively reduces the area of the current convex polygon by deleting one of its vertex. A density-based criterion is adopted to choose the next vertex to be deleted. The density of a polygon is the ratio between the number of geotagged items enclosed by the polygon, and its area. At each step, the procedure deletes the vertex that produces the polygon with highest density, among all the possible polygons. The procedure ends when it cannot further reduce the current polygon, and returns the set of convex polygons $CP = \{cp_0, ..., cp_n\}$ obtained after the n steps that have been performed.

— *G-RoI selection*. It analyses the set of convex polygons CP returned by the *G-RoI reduction* procedure, and selects the polygon representing RoI $\mathcal{R}$ for PoI $\mathcal{P}$. An area-variation criterion is adopted to choose $\mathcal{R}$ from CP. Given CP, the procedure identifies two subsets: a first subset $\{cp_0, ..., cp_{cut-1}\}$ such that the area of any cp_i is significantly larger than the area of cp_{i+1}; a second subset $\{cp_{cut}, ..., cp_n\}$ such that the area of any cp_i is not significantly larger than the area of cp_{i+1}. The procedure returns cp_{cut} as RoI $\mathcal{R}$. This corresponds to choosing cp_{cut} as the corner point of a discrete

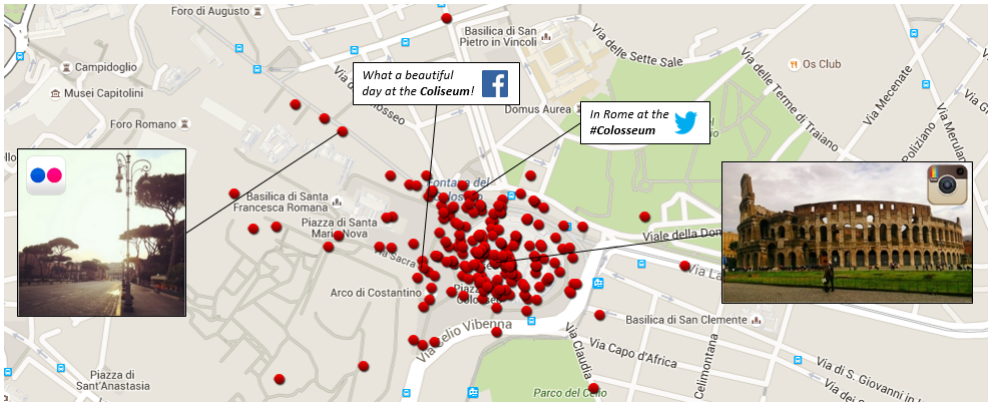
[1] https://www.openstreetmap.org/

L-curve [Hansen 1992] obtained by plotting the areas of all the convex polygons in CP on a Cartesian plane, as detailed later in this section.

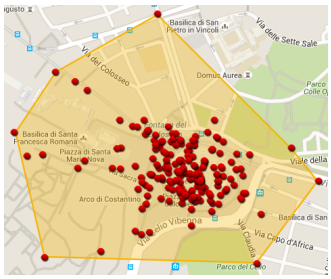
It is worth noting that the G-RoI algorithm was designed to identify regions represented by polygons, i.e., regions having non-null areas. In the extreme situation in which all the points are perfectly in a line, the algorithm cannot identify any polygon and therefore terminates immediately without returning any RoI. However, it must be mentioned that, since the coordinates (latitude and longitude) of each point are available at a very fine grain (six decimal digits, corresponding to a resolution of a few decimeters), it is very unlikely that all the input points are perfectly aligned, and in our experiments on real data this extreme case was never found.

4.1. Example

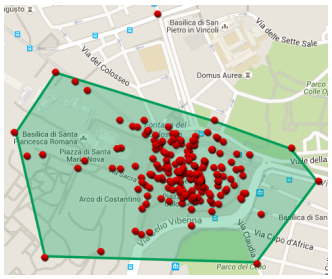
For the sake of clarity and for the Reader's convenience, before going into algorithmic details, we describe how the *G-RoI reduction* and *selection* procedures work through a real example. We collected a small sample of 200 geotagged items from different social networks (Flickr, Twitter, Instagram and Facebook), referring to the *Colosseum* in Rome and posted at a maximum distance of 500m from it.



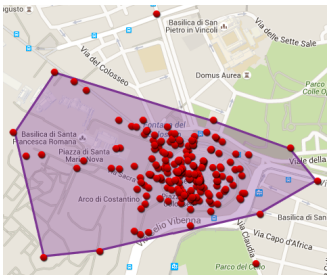
(a) Collection of geotagged items.



(b) Initial convex polygon cp_0.



(c) Generating cp_1 by deleting one vertex from cp_0.



(d) Generating cp_2 by deleting one vertex from cp_1.

Fig. 1. G-RoI reduction on Colosseum's geotagged items.

In their posts and photos, the social network users identify the *Colosseum* with different keywords. The Geonames website[2] reports the names used in different languages to identify the Colosseum, such as *Coliseum*, *Coliseo*, *Colise*, and synonymous such as *Flavian Amphitheatre* or *Amphitheatrum Flavium*. All the geotagged items in our sample contain at least one of such keywords. From these items, the 200 coordinates shown in Figure 1(a) are extracted. Given the coordinates, the *G-RoI reduction* procedure calculates the initial convex polygon cp_0 (shown Figure 1(b)), and then iteratively reduces the area. Figure 1(c) shows polygon cp_1 obtained after the first step by deleting one of the vertices from cp_0. Similarly, Figure 1(d) shows polygon cp_2 obtained after cp_1. The *G-RoI reduction* procedures iterates until it cannot further reduce the current polygon. The output of the procedure is the set of convex polygons $CP = \{cp_0, cp_1, ..., cp_n\}$ obtained at each step. Figure 2 shows with different colors all the convex polygons in CP, including the one chosen as RoI $\mathcal{R}$ by the subsequent *G-RoI selection* procedure.

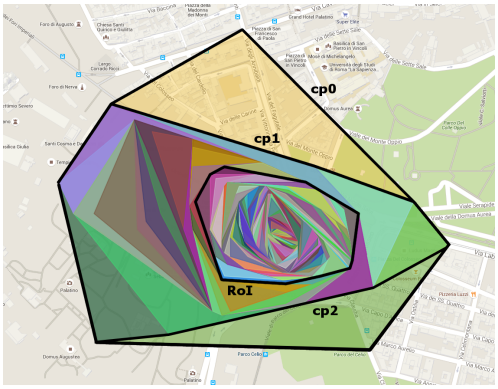


Fig. 2. Set of convex polygons in CP identified by the RoI reduction procedure, with indication of RoI $\mathcal{R}$ chosen by the RoI selection procedure.

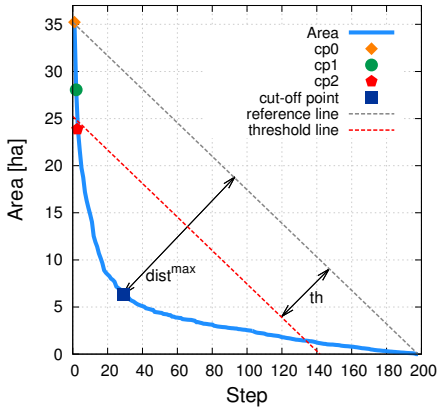
The *G-RoI selection* procedure analyzes CP to choose RoI $\mathcal{R}$ among all the convex polygons in it. To this end, the procedure extracts from CP an ordered set of Cartesian points $P = \{(0, A_0), (1, A_1), ..., (n, A_n)\}$.

An element $p_i \in P$ is a point (i, A_i), where i is the step in which cp_i was generated, and A_i is the area of cp_i. Figure 3(a) plots all the points in P in our example. The graph shows how much the area decreases with the steps performed by the *G-RoI reduction* procedure. The graph can be divided in two parts:

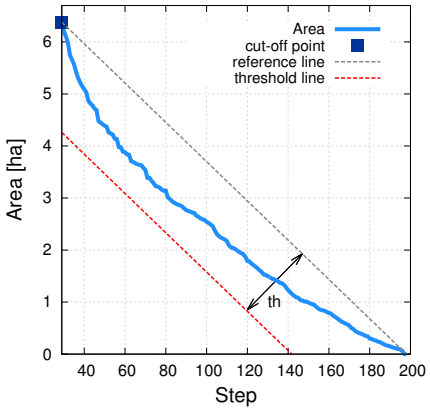
— The first part, from step 0 to a cut-off point p_{cut} (not included), decreases quickly, because at each step the *G-RoI reduction* procedure cuts a significant portion of area.
— The second part, from p_{cut} to step n, decreases slowly, because at each step the *G-RoI reduction* procedure cuts only a small portion of area.

The *G-RoI selection* procedure identifies the point p_{cut} that is located at the maximum distance ($dist^{max}$) from the *reference line* joining the first point and the last point under analysis (p_0 and p_n), as shown in Figure 3(a). If the set of points $\{p_{cut}, ..., p_n\}$ follows a linear trend as shown in Figure 3(b), i.e., there is no point below a *threshold line* at distance th from the reference line joining the points p_{cut} and p_n, then the procedure returns the polygon corresponding to p_{cut} as RoI $\mathcal{R}$ (see Figure 3(c)). Otherwise, the

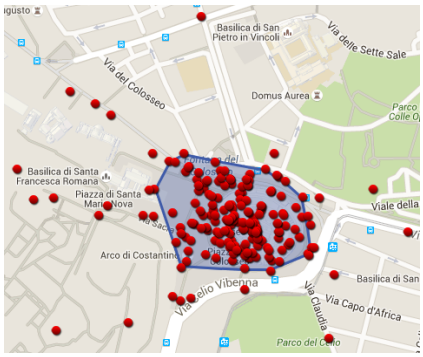
[2]http://geonames.org/



(a) Colosseum's points $0 \leq p_i \leq n$.



(b) Colosseum's points $cut \leq p_i \leq n$.



(c) Convex polygon corresponding to p_{cut}), chosen as RoI $\mathcal{R}$.

Fig. 3. G-RoI selection from Colosseum's convex polygons.

G-RoI selection procedure iterates by finding a new cut-off point from the set of points on the right of p_{cut}, as detailed in the next section.

4.2. Algorithmic details

Algorithm 1 shows the pseudo-code of the *G-RoI reduction* procedure. The input is a set of coordinates C and the output is a set of convex polygons CP. Starting from C, the procedure calculates the initial convex polygon cp_0 (line 1). Then, cp_0 is added to CP and is taken as current convex polygon cp (lines 2-3). A do-while block performs the area reduction steps (lines 4-22). At each step, the area of the current convex polygon cp is reduced by deleting one of its vertices. This implies that the area of cp_{i+1} is always lower than the area of cp_i. The algorithm ends when it cannot further reduce cp.

At the beginning of each reduction step, the current maximum density ρ^{max} is set to zero (line 5), while the convex polygon with maximum density cp^{max} and the vertex to be deleted v^{del} are initialized to null (lines 6-7). At each reduction step, for choosing the vertex to be deleted from cp, the algorithm iterates (lines 8-17) on each vertex $v \in cp$ performing the following operations:

- creates a temporary set of coordinates C^{tmp} obtained by deleting v from C (line 9);
- calculates the convex polygon cp^{tmp} from C^{tmp} (line 10);

ALGORITHM 1: G-RoI reduction.

Input : Set of coordinates C
Output: Set of convex polygons CP

```
1  cp_0 ← convexHull(C);                                  /* Initial convex polygon */
2  CP ← {cp_0};                                           /* Set of convex polygons */
3  cp ← cp_0;                                             /* Current convex polygon */
4  do
5      ρ^max ← 0;                                         /* Current maximum density */
6      cp^max ← ⊢;                                        /* Convex polygon with density = ρ^max */
7      v^del ← ⊢;                                         /* Vertex to be deleted */
8      for v ∈ cp do
9          C^tmp ← C − v;
10         cp^tmp ← convexHull(C^tmp);
11         A^tmp ← Area(cp^tmp);
12         if A^tmp > 0 then
13             ρ^tmp ← |C^tmp| / A^tmp;
14             if ρ^tmp > ρ^max then
15                 ρ^max ← ρ^tmp;
16                 cp^max ← cp^tmp;
17                 v^del ← v;
18     if ρ^max > 0 then
19         CP ← CP ∪ {cp^max};
20         cp ← cp^max;
21         C ← C − v^del;
22 while ρ^max > 0;
23 return CP
```

- calculates the area A^{tmp} of cp^{tmp} (line 11);
- if A^{tmp} is greater than zero (line 12), the density ρ^{tmp} of cp^{tmp} is calculated as the number of coordinates in C^{tmp} divided by A^{tmp} (line 13);
- if ρ^{tmp} is greater than ρ^{max} (line 14), ρ^{tmp} is assigned to ρ^{max} (line 15), cp^{tmp} is assigned to cp^{max} (line 16), and v is assigned to the vertex to be deleted v^{del} (line 17).

After having iterated on all vertices, if ρ^{max} is greater than zero (i.e., at least one polygon was found) (line 18), the algorithm adds cp^{max} to CP (line 19), assigns cp^{max} to cp (line 20), and deletes v^{del} from C (line 21). Finally, when the current reduction step does not change ρ^{max}, and so it remains equal to zero, which means that the current convex polygon cannot be further reduced (line 22), the algorithm returns the set of convex polygons CP generated (line 23).

Algorithm 2 shows the pseudo-code of the *G-RoI selection* procedure. The input is a set of convex polygons CP (i.e., output of *G-RoI reduction*) and a threshold $th \in (0, 1)$. Given CP, the algorithm creates a set of Cartesian points P, where each point p_i is a pair (i, A_i), with i identifying the step in which cp_i has been generated (by *G-RoI reduction*) and A_i representing the area of cp_i (lines 1-4). Given two adjacent points $p_i = (i, A_i)$ and $p_{i+1} = (i + 1, A_{i+1})$, A_i is strictly greater than A_{i+1}, because the area of cp_{i+1} is always lower than the area of cp_i (see Algorithm 1).

Then, the index of the cut-off point cut is set to zero (line 5). At each iteration (lines 6-19) the algorithm tries to find a cut-off point p_{cut} that is at the maximum distance from the line $y = 1 - x$ (which links the first and last normalized points in CP), and which is located below the line $y = 1 - th - x$ (i.e., within a threshold distance th from

the line $y = 1 - x$). Thus, at the beginning of each iteration, the maximum distance $dist^{max}$ is set to zero (line 7), and the index of the point with maximum distance i^{max} is set to cut (line 8).

ALGORITHM 2: G-RoI selection.

Input : Set of convex polygons CP; Threshold $th \in (0, 1)$
Output: Region of Interest $\mathcal{R}$.

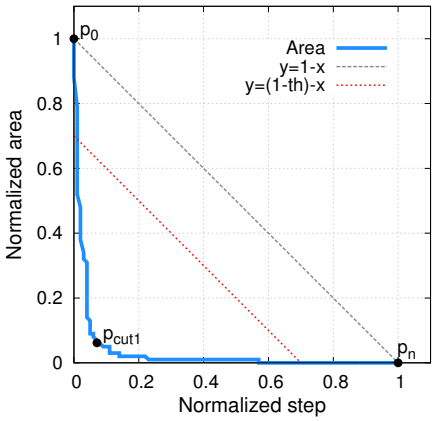
```
1  P ← ∅;                                                    /* Set of Cartesian points */
2  for cp_i ∈ CP do
3  |   A_i ← Area(cp_i);
4  |   P ← P ∪ {(i, A_i)};
5  cut ← 0;                                                  /* Index of the cut-off point */
6  do
7  |   dist^max ← 0;                                        /* Current maximum distance from y=1-x */
8  |   i^max ← cut;                                         /* Index of the point with dist^max */
9  |   for i ← cut + 1 to n − 1 do                           /* Where n = |CP| − 1 */
10 |   |   x^norm = (P_i.x − P_cut.x)/(P_n.x − P_cut.x);
11 |   |   y^norm = (P_i.y − P_n.y)/(P_cut.y − P_n.y);
12 |   |   if y^norm < 1 − th − x^norm then
13 |   |   |   dist^tmp = (1 − y^norm − x^norm) · √2/2;
14 |   |   |   if dist^tmp ≥ dist^max then
15 |   |   |   |   dist^max ← dist^tmp;
16 |   |   |   |   i^max ← i;
17 |   if dist^max > 0 then
18 |   |   cut ← i^max;
19 while dist^max > 0;
20 return cp_cut
```

The algorithm iterates (lines 9-16) on each point p_i between p_{cut} and p_n (i.e., $p_i \in (p_{cut}, p_n)$) and performs the following operations:

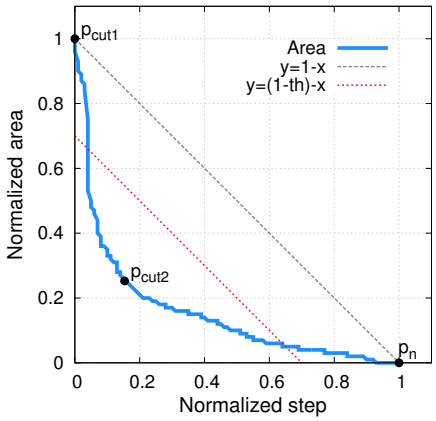
- normalizes $p_i.x$ with respect to $[p_{cut}.x, p_n.x]$ and stores such value in x_{norm} (line 10);
- normalizes $p_i.y$ with respect to $[p_n.y, p_{cut}.y]$ and stores such value in y_{norm} (line 11);
- if the normalized point (x_{norm}, y_{norm}) is below the line $y = 1 - th - x$ (line 12), $dist^{tmp}$ is calculated as the distance of that point from $y = 1 - x$ (line 13).
- if $dist^{tmp}$ is greater than $dist^{max}$ (line 14), $dist^{max}$ is updated to $dist^{tmp}$ (line 15) and i^{max} is updated to i (line 16).

After having iterated on all points in $\{p_{cut}, ..., p_n\}$, if $dist^{max}$ is greater than zero (i.e. a new cut-off point was found) (line 17), cut is updated to i^{max} (line 18). Finally, when $dist^{max}$ is equal to zero (i.e., there are no points below $y = 1 - th - x$) (line 19), the algorithm returns the convex polygons cp_{cut} as RoI $\mathcal{R}$ (line 20).

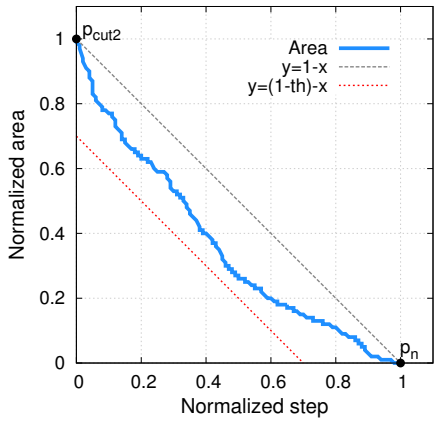
Figure 4 shows an example in which *G-RoI selection* procedure iterates three times to find the cut-off point. At the first iteration, the algorithm analyses the points in $\{p_0, ..., p_n\}$ and finds the first cut-off point p_{cut1} (see Figure 4(a)). At the second iteration, the algorithm analyses the points in $\{p_{cut1}, ..., p_n\}$ and finds a new cut-off point p_{cut2} (see Figure 4(b)). At the third iteration, the algorithm analyses the points in $\{p_{cut2}, ..., p_n\}$ but it does not find any cut-off point (see Figure 4(c)). Therefore, the algorithm returns as RoI $\mathcal{R}$ the convex polygon corresponding to p_{cut2}.



(a) Iteration 1: Found cut-off point p_{cut1}.



(b) Iteration 2: Found cut-off point p_{cut2}.



(c) Iteration 3: No cut-off point found.

Fig. 4. G-RoI selection procedure: An example with three iterations.

4.3. Complexity analysis

In the following we show that the time complexity of G-RoI is $\mathcal{O}(n^3 \log n)$, where n is the number of input coordinates (i.e., the size of C). To this end, we analyze separately the complexity of the *G-RoI reduction* and *G-RoI selection* procedures.

The complexity of *G-RoI reduction* is $\mathcal{O}(n^3 \log n)$. In fact, the first part of the procedure (lines 1-3 of Algorithm 1) has the complexity of calculating the initial convex hull polygon, which is equal to $\mathcal{O}(n \log n)$ (as proven in [Graham 1972]). The second part of the procedure (lines 4-22) has a complexity of $\mathcal{O}(n^3 \log n)$, as detailed in the following:

(1) The *do-while* block performs n iterations, because at each iteration the procedure deletes one of the coordinates in C.
(2) The *for* block performs at most n iterations, because the number of vertices in cp is at most n.
(3) Each *for* iteration has the complexity of calculating a convex hull polygon, which is $\mathcal{O}(n \log n)$.

The complexity of *G-RoI selection* is $\mathcal{O}(n^2)$. In fact, the first part of the procedure (lines 1-4 of Algorithm 2) has the complexity of calculating the area of each convex polygon generated by the *G-RoI reduction* procedure. Since the complexity of calculating the area is $\mathcal{O}(n)$ (as proven in [Braden 1986]), and the number of convex polygons is $\mathcal{O}(n)$, the complexity of this part of the procedure is $\mathcal{O}(n^2)$. Also the second part of the procedure (lines 5-19) has a complexity of $\mathcal{O}(n^2)$, as detailed in the following:

(1) The *do-while* block performs at most n iterations, because at each iteration the procedure deletes at least one of the points in P.
(2) The *for* block performs n iterations, because it checks all the points in P.

Considering the whole of the two procedures, it can be concluded that the time complexity of G-RoI is $\mathcal{O}(n^3 \log n)$.

5. EVALUATION

We experimentally evaluated the accuracy of G-RoI in detecting the RoIs associated to a set of PoIs, comparing it with three existing techniques: *Circle* [Spyrou and Mylonas 2016] (representative of the predefined-shapes approach), *DBSCAN* [Zheng et al. 2012] (density-based clustering), and *Slope* [Cai et al. 2014] (grid-based aggregation). The analysis was carried out on 24 PoIs located in the center of Rome (St. Peter's Basilica, Colosseum, Circus Maximus, etc.) and 24 PoIs located in the center of Paris (Louvre Museum, Eiffel Tower, etc.) using about 2.3 millions geotagged items published in Flickr from January 2006 to May 2016 in the areas under analysis.

5.1. Performance metrics

To measure the accuracy of the algorithms in detecting RoIs, we use *precision* and *recall* metrics. As in [de Graaff et al. 2013], let roi_{real} be the real RoI for a PoI, and let roi_{found} be the RoI found by an algorithm. Let us define the true positive area roi_{TP} as the intersection of roi_{found} and roi_{real}. Precision $Prec$ and recall Rec are defined as:

$$Prec = \frac{Area(roi_{TP})}{Area(roi_{found})} \qquad Rec = \frac{Area(roi_{TP})}{Area(roi_{real})} \tag{1}$$

A roi_{found} larger than roi_{real} produces a high recall and a low precision, whereas roi_{found} smaller than roi_{real} produces a low recall and a high precision. If $roi_{real} \subseteq roi_{found}$ then $roi_{TP} = roi_{real}$ and therefore the recall is 1 but the precision is lower than 1. On the other hand, if $roi_{found} \subseteq roi_{real}$ the precision is 1 but the recall is lower than 1.

To rank the results, we combine precision and recall using the F_1 score:

$$F_1 = \frac{2 \cdot Prec \cdot Rec}{Prec + Rec} \tag{2}$$

5.2. Data source

The evaluation has been performed on geotagged data collected from Flickr[3], which is one of the most used social networks for photo sharing. Flickr shares more than one billion of photos that can be gathered using public APIs, which allow to retrieve metadata about all the photos matching the provided search criteria, e.g. the photos taken in a radius from a given geographical point.

Using the APIs, we collected metadata about 2.3 millions geotagged items published in Flickr from January 2006 to May 2016 in the central areas of Rome and Paris. For

[3]http://flickr.com

each photo matching the search criteria, the Flickr APIs returned a metadata element such as the one shown in Figure 5.


```
{ "id":"987654321",
  "owner":{"id":"123456789@N00","username":"FlickrUser"},
  "dateTaken":"May 3, 2015 4:39:24 PM",
  "tags":[
    {"value":"italy"},{"value":"rome"},{"value":"piazzadispagna"},
    {"value":"itali"},{"value":"spanishteps"}
  ],
  "title":"Night at Piazza di Spagna",
  "description": "In the Piazza di Spagna, just below the Spanish Steps",
  "geoData":{ "longitude":12.482045, "latitude":41.905888}
  ...
}
```


Fig. 5. An example of metadata element returned by the Flickr APIs.

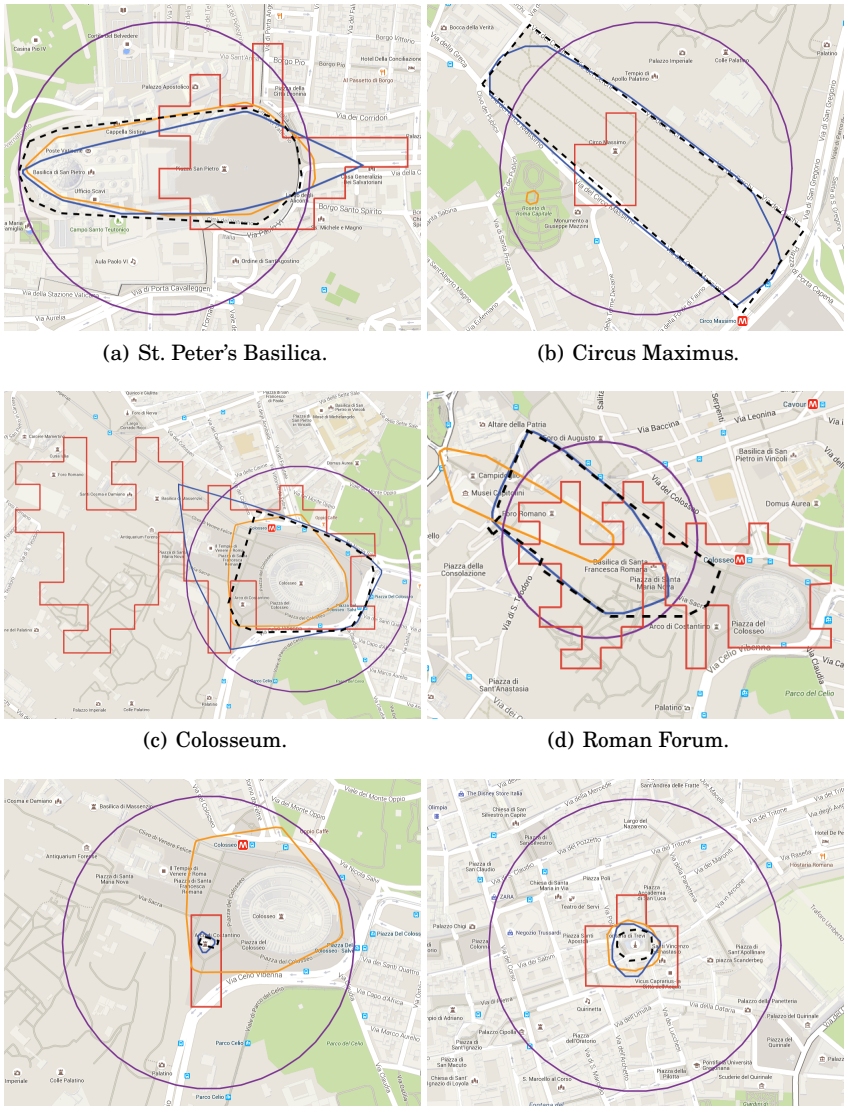
Each metadata element was parsed to extract the relevant features associated to geotagged items introduced in Section 2 (*text*, *tags*, *coordinates*, *userId*, *timestamp*).

5.3. Experimental results

The techniques under analysis need some parameters to work. We made several preliminary tests to find parameter values that perform effectively in all the scenarios, taking into account that the various PoIs are characterized by significant variability of shape, area and density (number of Flickr photos divided by area). For the Circle technique, the radius was set to 260 meters. With DBSCAN, the maximum distance between points is 10 meters and the minimum number of cluster points is 150. For the Slope technique, the square cell side is 55 meters and the minimum cell support is 150. For G-RoI, the threshold th was set to 0.27. The next two sections present the results obtained on 24 representative PoIs in Rome and 24 PoIs in Paris, respectively.

5.3.1. Rome. Figure 6 reports a graphical view of six (out of the 24 analyzed) representative PoIs in Rome (St. Peter's Basilica, Circus Maximus, Colosseum, Roman Forum, Arch of Constantine and Trevi Fountain): $i)$ purple lines represent the RoIs found by Circle; $ii)$ orange lines represent the RoIs identified by DBSCAN; $ii)$ red lines the RoIs found by Slope RoI; $iii)$ blue lines those found using G-RoI; $iv)$ black dotted lines the real RoIs.

As shown in the figure, the RoIs identified by the Circle technique are very approximative compared to the real ones. This is due to two reasons: $i)$ circles cannot be used to represent elongated shapes (e.g. Circus Maximus); $ii)$ with a given radius it is difficult to represent well places with very different areas (e.g., Colosseum vs Trevi Fountain). DBSCAN produced accurate results with St. Peter's Basilica and Colosseum, but failed in finding RoIs from two adjacent places (e.g., Colosseum and Arch of Constantine) or of places with low density. The low accuracy of DBSCAN with low density places is particularly evident in the case of Circus Maximus, where the RoI identified is very small compared to the real one. This is due to the fact that, when the points are few and distant each other, DBSCAN does not recognize them as part of the same cluster. Also Slope failed in distinguishing RoIs from two adjacent places (e.g., Colosseum and Roman Forum) that do not present significant density variations. Moreover, Slope fails



(a) St. Peter's Basilica. (b) Circus Maximus.

(c) Colosseum. (d) Roman Forum.

(e) Arch of Constantine. (f) Trevi Fountain.

Fig. 6. RoIs identified by different techniques: Circle (purple lines), DBSCAN (orange), Slope (red), G-RoI (blue). Real RoIs shown as black dotted lines.

in finding good RoIs for places with low density (e.g., with Circus Maximus it found a very small RoI compared to the real one).

Differently from the previous techniques, G-RoI is able to represent PoIs characterized by different shapes, areas and densities. In fact, G-RoI works well with both compact and elongated shapes (e.g., Trevi Fountain and Circus Maximus), with both small and large areas (e.g., Arch of Constantine and Roman Forum), and with various densities (from Circus Maximus to Colosseum). In addition, G-RoI accurately distinguishes RoIs of adjacent PoIs (e.g., Arch of Constantine and Colosseum).

Table II. Precision, Recall, and F_1 score of Circle, DBSCAN, Slope and G-RoI over 24 PoIs in Rome. For each row, the best F_1 score is indicated in bold.

PoI	*Circle*			*DBSCAN*			*Slope*			*G-RoI*		
	Prec	*Rec*	F_1	*Prec*	*Rec*	F_1	*Prec*	*Rec*	F_1	*Prec*	*Rec*	F_1
St. Peter's Basilica	0.39	1.00	0.56	0.96	0.86	**0.91**	0.56	0.50	0.53	0.92	0.78	0.84
Circus Maximus	0.39	0.84	0.53	0.00	0.00	0.00	0.81	0.13	0.22	0.95	0.94	**0.94**
Colosseum	0.33	1.00	0.50	0.90	0.75	**0.82**	0.27	0.83	0.40	0.61	1.00	0.76
Roman Forum	0.62	0.85	0.71	0.61	0.25	0.00	0.44	0.62	0.51	0.95	0.80	**0.87**
Arch of Constantine	0.00	1.00	0.01	0.01	1.00	0.02	0.06	1.00	0.11	0.53	0.85	**0.65**
Trevi Fountain	0.01	1.00	0.03	0.42	1.00	0.59	0.14	1.00	0.24	0.49	1.00	**0.66**
Piazza Colonna	0.02	1.00	0.05	0.93	0.52	0.67	0.18	1.00	0.31	0.92	0.82	**0.87**
Tiber Island	0.14	1.00	0.24	1.00	0.02	0.03	0.40	0.26	0.31	0.72	0.81	**0.76**
Mausoleum of Hadrian	0.11	1.00	0.20	0.86	0.65	**0.74**	0.63	0.59	0.61	0.77	0.59	0.67
Piazza del Popolo	0.11	1.00	0.20	0.98	0.58	0.73	0.60	0.88	0.71	0.60	0.98	**0.74**
Villa Borghese	1.00	0.24	0.38	1.00	0.00	0.00	1.00	0.00	0.01	1.00	0.44	**0.61**
Piazza di Spagna	0.11	1.00	0.20	0.72	0.65	0.68	0.41	0.77	0.54	0.87	0.84	**0.86**
Piazza Venezia	0.09	1.00	0.17	0.57	0.78	0.66	0.13	0.99	0.22	0.52	0.96	**0.68**
Piazza Navona	0.06	1.00	0.11	0.71	0.96	**0.81**	0.23	1.00	0.38	0.49	0.99	0.66
Trastevere	1.00	0.36	0.53	1.00	0.01	0.02	1.00	0.04	0.08	1.00	0.55	**0.71**
Our Lady in Trastev.	0.02	1.00	0.03	0.62	0.98	0.76	0.14	1.00	0.25	0.83	0.94	**0.88**
Capitoline Hill	0.09	1.00	0.17	0.31	1.00	0.47	0.45	0.43	0.44	0.94	0.93	**0.94**
Vatican Museums	0.41	1.00	0.58	0.75	0.51	0.00	0.55	0.78	0.65	0.65	0.87	**0.75**
Pantheon	0.04	1.00	0.09	0.58	0.93	0.72	0.17	1.00	0.29	0.71	0.98	**0.82**
The Mouth of Truth	0.03	1.00	0.06	0.98	0.24	0.38	0.38	0.90	0.54	0.75	0.88	**0.81**
Palazzo Montecitorio	0.04	1.00	0.08	1.00	0.15	0.26	0.79	0.58	**0.67**	0.98	0.42	0.59
Campo de' Fiori	0.02	1.00	0.04	0.56	1.00	0.72	0.24	0.98	0.39	0.77	0.96	**0.85**
St Mary Major	0.12	1.00	0.22	1.00	0.21	0.35	0.88	0.53	0.66	0.86	0.65	**0.74**
Janiculum	0.59	0.70	0.64	0.00	0.00	0.00	1.00	0.03	0.07	0.94	0.78	**0.85**
Mean values	*0.24*	*0.92*	*0.26*	*0.69*	*0.54*	*0.43*	*0.48*	*0.66*	*0.38*	*0.78*	*0.82*	***0.77***

Table II illustrates the performance (Precision, Recall, F_1 score) of the four techniques, for all the 24 PoIs that have been considered. The last row of the table reports mean values computed over the 24 PoIs.

The results reported in the table confirm that using a predefined shape (the Circle) does not bring to accurate results. In fact, Circle produces a very high recall with a low precision (which result in a mean F_1 score of 0.26), which means that the RoI identified by the technique is too large compared to the real one. In most cases, the recall is equal to 1 because the RoIs found contain the real ones (see also Section 5.1).

DBSCAN achieves the best results (F_1 score ranging from 0.74 to 0.91) with four PoIs - St. Peter's Basilica, Colosseum, Piazza Navona and Mausoleum of Hadrian - which are characterized by a similar density. On average, the precision of DBSCAN was 0.69 and the recall was 0.54, which leads to a mean F_1 score of 0.43. The fact that the precision is higher than the recall, means that the RoIs identified by DBSCAN are too small compared to the real ones.

Slope identifies the best RoI only with one PoI, Palazzo Montecitorio, with an F_1 score of 0.67. On the mean, the precision of Slope was 0.48 and the recall was 0.66, with a mean F_1 score of 0.38. In this case, the precision is lower than the recall, which means that the RoIs identified by this techniques are on average larger than the real ones.

Finally, G-RoI outperformed the other RoI mining techniques in 19 out of 24 PoIs, with a mean precision of 0.78, a mean recall of 0.82, and a mean F_1 score of 0.77 (0.34 higher than the F_1 score of DBSCAN). These results confirm the ability of G-RoI to accurately identify RoIs regardless of shapes, areas and densities of PoIs, and without being influenced by the proximity of different PoIs. In the few cases in which G-RoI does not result the best technique (5 out 24 PoIs in the case of Rome), its accuracy is very close to the best technique, i.e., it gets a F1 score that is lower than the F1 obtained with the best technique by just 0.08, on average. We noticed that in these cases G-RoI is able to return the correct shape of the place, but the area is either larger

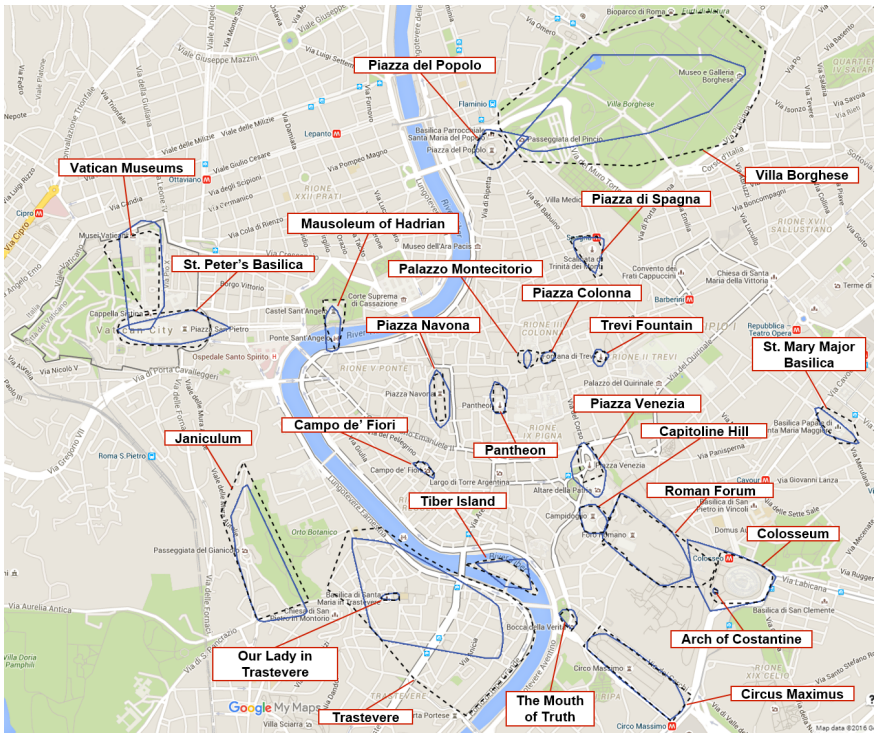


Fig. 7. City of Rome: RoIs identified by G-RoI (blue lines) compared with real ones (black dotted lines).

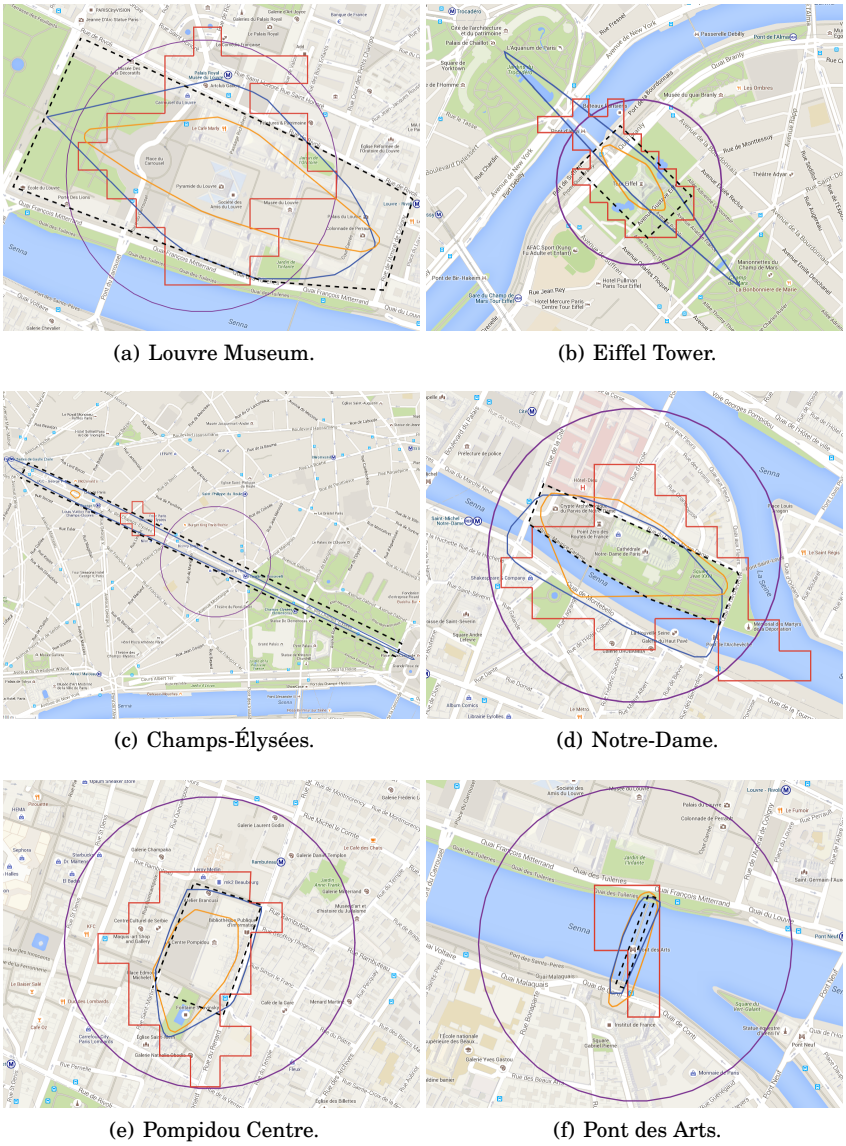
or smaller than the actual one. In the first case the recall is high but the precision is low (see, for example, Piazza Navona), in the second case is the opposite (e.g., Mausoleum of Hadrian). In both cases, this results in a relatively low F1 score compared to that of the traditional techniques. For a complete view of the results produced by G-RoI, Figure 7 shows all the 24 RoIs of Rome found by G-RoI, compared with the real ones.

5.3.2. Paris. Figure 8 presents a graphical view of six (out of the 24 analyzed) representative PoIs in Paris (Louvre Museum, Eiffel Tower, Champs-Élysées, Notre-Dame, Pompidou Centre, Pont des Arts), while Table III presents the performance of the four techniques (Circle, DBSCAN, Slope and G-RoI), for all the 24 PoIs that have been considered in Paris.

The experimental results confirm the behavior observed in Rome RoIs. Also in this case, Circle does not compute accurate results, producing a very high recall with a low precision (which results in a mean F_1 score of 0.23).

DBSCAN achieves the best results only with four PoIs (i.e., Notre-Dame, Moulin Rouge, Paris Opera, and Arc de Triomphe). On the mean, the precision of DBSCAN was 0.85 and the recall was 0.42, which means that the RoIs identified by this techniques are on average smaller than the real ones. Furthermore, Slope identifies the best RoI only for two PoIs (i.e. Eiffel Tower and Place de la Concorde). On average, the precision of Slope was 0.45 and the recall was 0.64, with an average F_1 score of 0.44. In this case, the precision is lower than the recall, which means that the RoIs identified by this techniques are on average larger than the real ones.

Finally, G-RoI outperformed the other RoI mining techniques in 18 out of 24 PoIs, with a mean precision of 0.81, a mean recall of 0.66, and a mean F_1 score of 0.70 (0.23



(a) Louvre Museum. (b) Eiffel Tower.

(c) Champs-Élysées. (d) Notre-Dame.

(e) Pompidou Centre. (f) Pont des Arts.

Fig. 8. RoIs identified by different techniques in Paris: Circle (purple lines), DBSCAN (orange), Slope (red), G-RoI (blue). Real RoIs shown as black dotted lines.

higher than the mean F_1 score of DBSCAN). In particular, G-RoI results to be the only technique able to identify an accurate RoI for the Champs-Élysées that are characterized by a very elongated shape, achieving a very high F_1 score (0.77). The behavior of G-RoI for the Eiffel Tower deserves to be discussed: differently from the other techniques, G-RoI produces a larger RoI with an elongated shape. This is due to the fact that anyone who wants to take a picture of the Eiffel Tower does not come strictly under it, but at some distance in front of it or behind it. Specifically, most geotagged items on this subject are located at Trocadéro, commonly considered the best place to

Table III. Precision, Recall, and F_1 score of Circle, DBSCAN, Slope and G-RoI over 24 PoIs in Paris. For each row, the best F_1 score is indicated in bold.

PoI	*Circle*			*DBSCAN*			*Slope*			*G-RoI*		
	Prec	*Rec*	F_1	*Prec*	*Rec*	F_1	*Prec*	*Rec*	F_1	*Prec*	*Rec*	F_1
Louvre Museum	0.66	0.72	0.69	1.00	0.36	0.53	0.74	0.49	0.59	0.94	0.69	**0.79**
Tour Eiffel	0.28	1.00	0.44	1.00	0.38	0.55	0.56	0.98	**0.72**	0.46	0.57	0.51
Champs-Élysées	0.18	0.26	0.22	1.00	0.01	0.02	0.65	0.08	0.14	0.95	0.64	**0.77**
Notre-Dame	0.18	1.00	0.30	0.76	0.84	**0.79**	0.32	0.84	0.46	0.53	0.90	0.67
Pompidou Centre	0.13	1.00	0.23	0.82	0.66	0.73	0.37	0.98	0.54	0.78	0.98	**0.87**
Pont des Arts	0.01	1.00	0.02	0.31	1.00	0.48	0.11	0.75	0.19	0.42	1.00	**0.59**
Place de la Concorde	0.26	1.00	0.41	1.00	0.15	0.26	0.74	0.79	**0.77**	0.99	0.43	0.60
Moulin Rouge	0.02	1.00	0.04	0.81	0.86	**0.84**	0.00	0.00	0.00	0.72	0.62	0.67
Place de la Bastille	0.07	1.00	0.13	1.00	0.24	0.39	0.62	0.87	0.73	0.95	0.69	**0.80**
Sacré-Cœur Basilica	0.05	1.00	0.09	0.48	0.90	0.63	0.02	0.01	0.01	0.81	0.63	**0.71**
Jardin des Plantes	0.77	0.79	0.78	1.00	0.00	0.01	1.00	0.09	0.16	0.97	0.84	**0.90**
Saint-Sulpice	0.06	1.00	0.11	1.00	0.09	0.17	0.59	0.57	0.58	0.96	0.48	**0.64**
Pantheon	0.11	1.00	0.19	1.00	0.29	0.45	0.62	0.82	0.70	0.74	0.78	**0.76**
Trocadéro	0.20	1.00	0.34	1.00	0.28	0.43	0.83	0.52	0.64	0.89	0.70	**0.78**
Place de la République	0.08	1.00	0.14	0.97	0.46	0.62	0.58	0.77	0.66	0.98	0.59	**0.74**
Musée de l'Orangerie	0.02	1.00	0.05	1.00	0.52	0.68	0.24	0.88	0.38	0.91	0.70	**0.79**
Galeries Lafayette	0.07	1.00	0.12	0.92	0.26	0.41	0.36	0.83	0.50	0.87	0.76	**0.81**
Arab World Institute	0.04	1.00	0.07	0.96	0.49	0.65	0.28	0.99	0.44	0.96	0.55	**0.70**
Grand Palais	0.17	1.00	0.30	1.00	0.38	0.55	0.61	0.94	0.74	0.83	0.85	**0.84**
Petit Palais	0.05	1.00	0.10	1.00	0.36	0.53	0.07	0.33	0.11	0.78	0.59	**0.67**
Paris Opera	0.07	1.00	0.13	0.90	0.56	**0.69**	0.37	0.84	0.52	0.93	0.49	0.64
Pont Neuf	0.04	1.00	0.08	0.83	0.18	0.30	0.16	0.74	0.27	0.55	0.59	**0.57**
Arc de Triomphe	0.05	1.00	0.10	0.55	0.77	**0.64**	0.30	1.00	0.46	0.50	0.35	0.41
Sorbonne	0.20	1.00	0.33	0.00	0.00	0.00	0.75	0.21	0.33	0.99	0.47	**0.64**
Mean values	*0.16*	*0.95*	*0.23*	*0.85*	*0.42*	*0.47*	*0.45*	*0.64*	*0.44*	*0.81*	*0.66*	***0.70***

take picture with Eiffel Tower in background. Overall, also the results on Paris confirm the ability of G-RoI in identifying RoIs characterized by a variety of shapes, areas and densities of PoIs.

5.3.3. Comparison with other techniques using preprocessed data. To further evaluate the accuracy of G-RoI compared to that achieved by the other techniques, in this section we present the results obtained by DBSCAN and Slope on all the cases of study presented above (places of interest in Rome and Paris) by using the same preprocessed data used by G-RoI. We recall that G-RoI has been designed to find the RoI of a place of interest, given a set of geotagged data referring to that place. For this reason, G-RoI preprocessing is a preliminary step in which the geotagged data referring to a place are selected for subsequent analysis.

Table IV reports the F1 score achieved by DBSCAN and Slope with and without preprocessing compared to that of G-RoI over the 24 PoIs in Rome. On average, using preprocessed data, DBSCAN and Slope improve their accuracy by 18% and 26% respectively. However, even using preprocessing, in most cases the accuracy of both techniques is lower than that achieved by G-RoI. Specifically, G-RoI is still the most accurate technique in 18 out of 24 PoIs.

Table V reports the F1 score achieved by DBSCAN and Slope with and without preprocessing compared to that of G-RoI over the 24 PoIs in Paris. DBSCAN and Slope improve their accuracy using preprocessed data, with an average increase of 27% for the former and 11% for the latter. Also in this case, in most cases the accuracy of both techniques is lower than that achieved by G-RoI, even using preprocessing. In fact, G-RoI remains the most accurate technique in 15 out of 24 PoIs.

It is worth noticing that, even using preprocessed data, DBSCAN and Slope are still unable to cope with low-density places (e.g., Circus Maximus and Villa Borghese in Rome, Champs-Élysées and Jardin des Plantes in Paris).

Table IV. F_1 score achieved by DBSCAN and Slope with and without preprocessing compared to that of G-RoI over the 24 PoIs in Rome. For each row, the best F_1 score is indicated in bold.

PoI	*DBSCAN*		*Slope*		*G-RoI*
	No preproc.	*Preproc.*	*No preproc.*	*Preproc.*	
St. Peter's Basilica	0.91	**0.92**	0.53	0.58	0.84
Circus Maximus	0.00	0.00	0.22	0.07	**0.94**
Colosseum	0.82	**0.86**	0.40	0.82	0.76
Roman Forum	0.00	0.50	0.51	0.38	**0.87**
Arch of Constantine	0.02	0.37	0.11	0.16	**0.65**
Trevi Fountain	0.59	0.43	0.24	0.42	**0.66**
Piazza Colonna	0.67	**0.87**	0.31	0.60	**0.87**
Tiber Island	0.03	0.07	0.31	0.26	**0.76**
Mausoleum of Hadrian	0.74	**0.75**	0.61	0.38	0.67
Piazza del Popolo	0.73	**0.92**	0.71	0.74	0.74
Villa Borghese	0.00	0.00	0.01	0.01	**0.61**
Piazza di Spagna	0.68	0.81	0.54	0.71	**0.86**
Piazza Venezia	0.66	0.61	0.22	0.48	**0.68**
Piazza Navona	**0.81**	0.65	0.38	0.58	0.66
Trastevere	0.02	0.03	0.08	0.02	**0.71**
Our Lady in Trastev.	0.76	0.66	0.25	0.81	**0.88**
Capitoline Hill	0.47	0.78	0.44	0.75	**0.94**
Vatican Museums	0.00	0.65	0.65	0.60	**0.75**
Pantheon	0.72	0.52	0.29	0.60	**0.82**
The Mouth of Truth	0.38	0.57	0.54	0.62	**0.81**
Palazzo Montecitorio	0.26	0.61	0.67	**0.83**	0.59
Campo de' Fiori	0.72	0.00	0.39	0.64	**0.85**
St Mary Major	0.35	0.62	0.66	0.50	**0.74**
Janiculum	0.00	0.06	0.07	0.04	**0.85**
Mean values	*0.43*	*0.51*	*0.38*	*0.48*	***0.77***

Table V. F_1 score achieved by DBSCAN and Slope with and without preprocessing compared to that of G-RoI over the 24 PoIs in Paris. For each row, the best F_1 score is indicated in bold.

PoI	*DBSCAN*		*Slope*		*G-RoI*
	No preproc.	*Preproc.*	*No preproc.*	*Preproc.*	
Louvre Museum	0.53	0.78	0.59	0.59	**0.79**
Tour Eiffel	0.55	0.79	0.72	**0.81**	0.51
Champs-Élysées	0.02	0.00	0.14	0.04	**0.77**
Notre Dame	**0.79**	0.67	0.46	0.56	0.67
Pompidou Centre	0.73	0.75	0.54	0.76	**0.87**
Pont des Arts	0.48	0.36	0.19	0.29	**0.59**
Place de la Concorde	0.26	0.00	**0.77**	0.00	0.60
Moulin Rouge	**0.84**	0.66	0.00	0.00	0.67
Place de la Bastille	0.39	0.67	0.73	0.78	**0.80**
Sacré-Cœur Basilica	0.63	0.69	0.01	0.63	**0.71**
Jardin des Plantes	0.01	0.04	0.16	0.26	**0.90**
Saint-Sulpice	0.17	**0.82**	0.58	0.41	0.64
Pantheon	0.45	0.75	0.70	0.74	**0.76**
Trocadéro	0.43	0.49	0.64	0.66	**0.78**
Place de la République	0.62	0.73	0.66	0.53	**0.74**
Musée de l'Orangerie	0.68	0.76	0.38	0.67	**0.79**
Galeries Lafayette	0.41	0.64	0.50	0.65	**0.81**
Arab World Institute	0.65	**0.81**	0.44	0.74	0.70
Grand Palais	0.55	0.83	0.74	0.83	**0.84**
Petit Palais	0.53	**0.91**	0.11	0.00	0.67
Paris Opera	0.69	**0.89**	0.52	0.60	0.64
Pont Neuf	0.30	0.49	0.27	0.37	**0.57**
Arc de Triomphe	**0.64**	0.61	0.46	0.47	0.41
Sorbonne	0.00	0.18	0.33	0.34	**0.64**
Mean values	*0.47*	*0.60*	*0.44*	*0.49*	***0.70***

6. CONCLUSION

RoI mining techniques are aimed at discovering Regions-of-Interest (RoIs) from Places-of-Interest (PoIs) and other data. Existing RoI mining techniques are based on the use of *predefined shapes*, *density-based clustering* or *grid-based aggregation*. In this paper we presented *G-RoI*, a novel RoI mining technique that exploits the indications contained in geotagged social media items to discover the RoI of a PoI with a high accuracy.

We experimentally evaluated the accuracy of G-RoI in detecting the RoIs associated to a set of PoIs, comparing it with three existing techniques: *Circle* (predefined-shapes approach), *DBSCAN* (density-based clustering), and *Slope* (grid-based aggregation). The analysis was carried out on a set of PoIs located in the center of Rome, characterized by different shapes, areas and densities, using a large set of geotagged photos published in Flickr over six years. The experimental results show that G-RoI is able to detect more accurate RoIs than existing techniques. Over a set of 24 PoIs in Rome, G-RoI achieved better results than related techniques based on the three classes of existing algorithms in 19 cases, with a mean precision of 0.78, a mean recall of 0.82, and a mean F_1 score of 0.77. In particular, the F_1 score of G-RoI is 0.34 higher than that obtained with the well-known DBSCAN algorithm.

To better assess the accuracy of G-RoI, further experiments have been run over an additional set of 24 PoIs in Paris. Also in this case, G-RoI achieved best results in 18 cases, with a mean precision of 0.81, a mean recall of 0.66, and a mean F_1 score of 0.70 (0.23 higher than that obtained with DBSCAN). These results confirm the ability of G-RoI to accurately identify RoIs regardless of shapes, areas and densities of PoIs, and without being influenced by the proximity of different PoIs. For the purpose of reproducibility, an open-source version of G-RoI and all the input data used in the experiments are available at https://github.com/scalabunical/G-RoI.

REFERENCES

Albino Altomare, Eugenio Cesario, Carmela Comito, Fabrizio Marozzo, and Domenico Talia. 2016. Trajectory Pattern Mining for Urban Computing in the Cloud. *Transactions on Parallel and Distributed Systems (IEEE TPDS)* (2016).

C. Bradford Barber, David P. Dobkin, and Hannu Huhdanpaa. 1996. The Quickhull Algorithm for Convex Hulls. *ACM Trans. Math. Softw.* 22, 4 (Dec. 1996), 469–483.

Luke Bermingham and Ickjai Lee. 2014. Spatio-temporal Sequential Pattern Mining for Tourism Sciences. *Procedia Computer Science* 29, 0 (2014), 379 – 389. 2014 International Conference on Computational Science.

Bart Braden. 1986. The surveyors area formula. *The College Mathematics Journal* 17, 4 (1986), 326–337.

Guochen Cai, Chihiro Hio, Luke Bermingham, Kyungmi Lee, and Ickjai Lee. 2014. Sequential pattern mining of geo-tagged photos with an arbitrary regions-of-interest detection method. *Expert Systems with Applications* 41, 7 (2014), 3514 – 3526.

Eugenio Cesario, Chiara Congedo, Fabrizio Marozzo, Gianni Riotta, Alessandra Spada, Domenico Talia, Paolo Trunfio, and Carlo Turri. 2015. Following Soccer Fans from Geotagged Tweets at FIFA World Cup 2014. In *Proc. of the 2nd IEEE Conference on Spatial Data Mining and Geographical Knowledge Services*. Fuzhou, China, 33–38. ISBN 978-1- 4799-7748-2.

Eugenio Cesario, Andrea Raffaele Iannazzo, Fabrizio Marozzo, Fabrizio Morello, Gianni Riotta, Alessandra Spada, Domenico Talia, and Paolo Trunfio. 2016. Analyzing Social Media Data to Discover Mobility Patterns at EXPO 2015: Methodology and Results. In *The 2016 International Conference on High Performance Computing & Simulation (HPCS 2016)*. Innsbruck, Austria. To appear.

E. Chaniotakis and C. Antoniou. 2015. Use of Geotagged Social Media in Urban Settings: Empirical Evidence on Its Potential from Twitter. In *2015 IEEE 18th International Conference on Intelligent Transportation Systems*. 214–219.

Yizong Cheng. 1995. Mean shift, mode seeking, and clustering. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 17, 8 (Aug 1995), 790–799.

David J. Crandall, Lars Backstrom, Daniel Huttenlocher, and Jon Kleinberg. 2009. Mapping the World's Photos. In *Proceedings of the 18th International Conference on World Wide Web (WWW '09)*. ACM, New York, NY, USA, 761–770.

Victor de Graaff, Rolf A. de By, Maurice van Keulen, and Jan Flokstra. 2013. Point of Interest to Region of Interest Conversion. In *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (SIGSPATIAL'13)*. ACM, New York, NY, USA, 388–391.

Martin Ester, Hans peter Kriegel, Jrg S, and Xiaowei Xu. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*. AAAI Press, 226–231.

Laura Ferrari, Alberto Rosi, Marco Mamei, and Franco Zambonelli. 2011a. Extracting Urban Patterns from Location-based Social Networks. In *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Location-Based Social Networks (LBSN '11)*. ACM, New York, NY, USA, 9–16.

Laura Ferrari, Alberto Rosi, Marco Mamei, and Franco Zambonelli. 2011b. Extracting Urban Patterns from Location-based Social Networks. In *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Location-Based Social Networks (LBSN '11)*. ACM, New York, NY, USA, 9–16. DOI:http://dx.doi.org/10.1145/2063212.2063226

Fosca Giannotti, Mirco Nanni, Fabio Pinelli, and Dino Pedreschi. 2007. Trajectory Pattern Mining. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '07)*. ACM, New York, NY, USA, 330–339.

Ronald L. Graham. 1972. An efficient algorith for determining the convex hull of a finite planar set. *Information processing letters* 1, 4 (1972), 132–133.

Per Christian Hansen. 1992. Analysis of Discrete Ill-Posed Problems by Means of the L-Curve. *SIAM Rev.* 34, 4 (1992), 561–580.

Slava Kisilevich, Daniel Keim, and Lior Rokach. 2010a. A Novel Approach to Mining Travel Sequences Using Collections of Geotagged Photos. In *Geospatial Thinking*, Marco Painho, Maribel Yasmina Santos, and Hardy Pundt (Eds.). Lecture Notes in Geoinformation and Cartography, Vol. 0. Springer Berlin Heidelberg, 163–182.

Slava Kisilevich, Florian Mansmann, and Daniel Keim. 2010b. P-DBSCAN: A Density Based Clustering Algorithm for Exploration and Analysis of Attractive Areas Using Collections of Geo-tagged Photos. In *Proceedings of the 1st International Conference and Exhibition on Computing for Geospatial Research & Application (COM.Geo '10)*. ACM, New York, NY, USA, Article 38, 4 pages.

Takeshi Kurashima, Tomoharu Iwata, Go Irie, and Ko Fujimura. 2010. Travel Route Recommendation Using Geotags in Photo Sharing Sites. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management (CIKM '10)*. ACM, New York, NY, USA, 579–588.

Jieming Shi, Nikos Mamoulis, Dingming Wu, and David W. Cheung. 2014. Density-based Place Clustering in Geo-social Networks. In *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data (SIGMOD '14)*. ACM, New York, NY, USA, 99–110.

Evaggelos Spyrou and Phivos Mylonas. 2016. Analyzing Flickr metadata to extract location-based information and semantically organize its photo content. *Neurocomputing* 172 (2016), 114 – 133.

Georges Voronoi. 1908. Nouvelles applications des paramètres continus à la théorie des formes quadratiques. Deuxième mémoire. Recherches sur les parallélloèdres primitifs. *Journal für die reine und angewandte Mathematik* 134 (1908), 198–287.

Zhijun Yin, Liangliang Cao, Jiawei Han, Jiebo Luo, and Thomas S Huang. 2011. Diversified Trajectory Pattern Ranking in Geo-tagged Social Media. In *SDM*. SIAM, 980–991.

Linlin You, G. Motta, D. Sacco, and Tiany Ma. 2014. Social data analysis framework in cloud and Mobility Analyzer for Smarter Cities. In *Service Operations and Logistics, and Informatics (SOLI), 2014 IEEE International Conference on*. 96–101.

Jing Yuan, Yu Zheng, Liuhang Zhang, XIng Xie, and Guangzhong Sun. 2011. Where to Find My Next Passenger. In *Proceedings of the 13th International Conference on Ubiquitous Computing (UbiComp '11)*. ACM, New York, NY, USA, 109–118.

Yan-Tao Zheng, Zheng-Jun Zha, and Tat-Seng Chua. 2012. Mining Travel Patterns from Geotagged Photos. *ACM Trans. Intell. Syst. Technol.* 3, 3, Article 56 (May 2012), 18 pages.