IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 17 June 2022, accepted 29 July 2022, date of publication 8 August 2022, date of current version 12 August 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3197169

RESEARCH ARTICLE

RealROI: Discovering Real Regions of Interest From Geotagged Photos

KWANG WOO NAM[1], (Member, IEEE), AND KWANGSOO YANG[2], (Member, IEEE)
[1]School of Computer, Information, and Communication Engineering, Kunsan National University, Gunsan 54150, Republic of Korea
[2]Department of Computer Science, Florida Atlantic University, Boca Raton, FL 33431, USA

Corresponding author: Kwang Woo Nam (kwnam@kunsan.ac.kr)

This work was supported in part by the National Research Foundation of Korea (NRF) Grant by the Korean Government through the Ministry of Science and ICT (MSIT) under Grant 2020R1F1A1048432; in part by the Korea Agency for Infrastructure Technology Advancement (KAIA) Grant by the Ministry of Land, Infrastructure and Transport under Grant RS-2022-00143336; and in part by the U.S. National Science Foundation under Grant 1844565.

ABSTRACT Given a set of geotagged photos with a view direction, the discovering real regions of interest (DRRI) problem identifies real regions of Interest (RealROIs) in the photos. The problem is important in many societal applications, including tourist route recommendations and travel advertisements. Much work utilized geotagged photos to discover regions of interest or routes and to provide recommendations according to similar preferences. However, these approaches are not ideal for identifying RealROIs because the usage of geotagged photos is limited to the GPS coordinates of the photo. To remedy this issue, we propose the DRRI problem that can use geometric and directional information to identify the interesting regions and develop novel algorithms to discover RealROIs in photos based on the direction in which the photos were taken. Our experimental results and case studies show that our approach outperforms the related work.

INDEX TERMS Geotagged photos, region of interest, social data mining, recommendations.

I. INTRODUCTION

A. BACKGROUND

With recent advancements in mobile technology, the integration of sensor devices such as GPSs in smartphones and cameras has become extremely common. Images or videos captured through these devices include various spatial information, such as GPS, time, and directional information. We refer to these images as geotagged photos [1]. Recently, millions of users have shared geotagged photos through social platforms such as Instagram, Facebook, Twitter, and Flickr to represent their interests and experiences. Much work has been conducted to utilize these geotagged photos to discover regions of interest (ROI) or routes [2], [3] and to provide recommendations according to similar preferences [4], [5], [19], [20], [21].

Most smartphones have built-in magnetometers, making them ideal for collecting photos with directional information. For example, 15 percent of the geotagged dataset from Flickr has camera direction information. Photos from social media services such as Facebook are often taken directly from smartphones, and the corresponding geotagging rate is expected to be much higher than that for Flickr photos [6].

However, despite these technological advancements, traditional approaches have focused only on GPS locations. In this paper, we propose a novel problem and algorithm that can use geotagged photos with camera directions and identify real regions of interest.

B. PROBLEM DEFINITION

Given a set of geotagged photos with a known view direction, the discovering real regions of interest (DRRI) problem identifies the real regions of interest (RealROIs) in geotagged photos. The problem is important in many societal applications, such as tourist route recommendations and travel advertisements. Recently, much work has focused on the

The associate editor coordinating the review of this manuscript and approving it for publication was Xianzhi Wang.

locations of photos and identified social groups based on these locations [4], [7], [8], [9], [10].

Traditionally, geotagged photos have only been associated with GPS information. In contrast, geotagged photos include additional information such as the direction in which the photo was taken and the angle of the camera at the time of capture. For example, spatially tagged media content provides directional and topological information to identify interesting spatial regions or points of interests (POIs). In this paper, we utilize geometric, directional and topological information, and identify RealROIs.

C. CONTRIBUTIONS

The traditional approach to POI identification is not ideal to identify RealROIs in several ways due to the nature of photos. First, although the GPS location of a vehicle or person is the same as that of the object of interest in a given case, the location of the photo may differ from that of the photographer or object of interest.

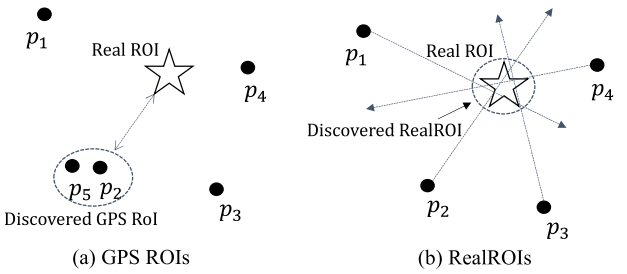


FIGURE 1. GPS ROIs and RealROIs.

Consider the example in Figure 1(a). The Statue of Liberty in New York is located on an island; as many photos are taken from boats, the ROI generated by existing GPS-based algorithms locates on the river which is some distance from the Statue of Liberty. Second, the location of the photographer can widely vary based on the distance between the object of interest and the location from which the image was captured. As shown in Figure 1(b), this issue may result in a situation in which the RealROIs cannot be generated. Third, a point of interest (POI) in traditional photos has a circular shape based on (x, y) coordinates, but the target of a photo is linearly aligned to the viewpoint of a photographer. This results that regions not in the photo are excluded from the ROI. Therefore, ROIs with GPS coordinates (GPS ROIs) cannot represent the RealROIs.

To remedy this issue, we propose a novel approach that can use the directions of camera views and identify ROIs based on view directions. Our new approach identifies cointeresting points in different view directions and groups these cointeresting points to find RealROIs.

The remainder of this paper is structured as follows. Section 2 reviews the related work. Section 3 introduces the concept of RealROIs. Section 4 defines extroverted/introverted RealROIs and describes the corresponding algorithms. In Section 5, we will evaluate our algorithms and show that our approach outperforms the related work. Section 6 shows cast studies and discusses our algorithms.

II. RELATED WORK

A. MINING GEOTAGGED PHOTOS

The discovery of ROIs provides a basis for product recommendations, advertisements, and tourist destination recommendations. POIs are conventionally determined based on a user's location. In [11], a user's location and time information were used to extract and analyze travel routes. Each user was identified by the ID of the shared photo, and the travel route for each user was extracted. In [2], the authors focused on both the identification of POIs from Flickr photos and association mining among the identified POIs. For certain points, the data were preprocessed such that only one photo was used.

Geotagged photos have been extensively used in existing studies for estimating ROIs. Kisilevich *et al.* [1] proposed an algorithm that considers user information through Photo-DBSCAN, thus addressing the issues caused by high-frequency uploaders. DBSCAN is useful for clustering points of interest. In particular, the proposed algorithm based on geotagged photos directly searches the user ROI to obtain clustering results. Moreover, Zeng *et al.* [12] discovered the movement patterns of users and clustered the locations shared by geo-photo users to obtain POIs for different cities. In [13], the authors analyzed user-generated travel routes from Flickr photo datasets to recommend tourist destinations for specific regions. In [4], the authors applied sequential trajectory pattern mining to Flickr photos. In addition to the research described above, various studies have considered only GPS data coordinates for applications such as POI extraction, route extraction, and travel route recommendations. While this approach is suitable for locating user ROIs, it is not sufficient for extracting objects of particular interest to the user, even within an ROI. To explore a large region in a short amount of time, it is necessary to find the regions or objects of greatest interest among existing POIs.

B. MINING PHOTOS WITH VIEW DIRECTIONS

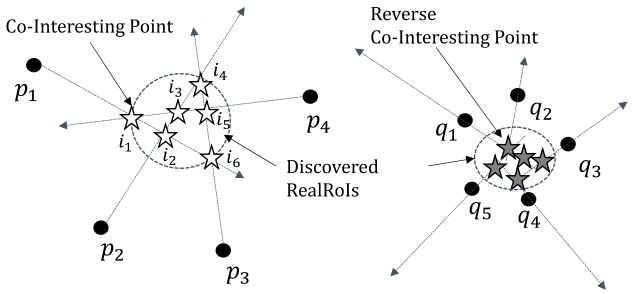
GPS information can only be used to identify the location at which a geotagged photo was taken, and the location can be far from real objects in an arbitrary view direction [6]. Park *et al.* [6] developed a method to estimate the exact direction in which a photo is taken by comparing the GPS coordinates and 3D model information contained in a geotagged photo. Additionally, this approach was used to find objects of interest. The concepts of *inward* and *outward* photos proposed in [14] and [15] are very similar to the view direction concept we introduce. Notably, novel methods were developed to detect and classify photos posted on social media sites (e.g., Flickr) using *inward* and *outward*

photos with view directions. Hotspots were detected using DBSCAN and a grid-based approach [16]. Three types of POIs were classified based on the direction from which the photo was taken. First, the points at which photos were taken both inside and outside a POI were designated as hotspots within the ROI. Second, points with many photos taken in one direction were also classified. Third, points with photos taken in all directions were classified. Although the hotspots were classified using the direction information from the photos, the location and direction information of the user was also considered. This approach is useful for locations where user interest does not vary, such as tourist destinations; however, it is insufficient for locations where user interest varies, such as shopping malls.

Hirota *et al.* [14], [15], [16] obtained spatial properties such as shooting locations or ROIs through clustering and classification based on grids. Methods based on grids have a common problem; notably, the identification of ROIs depends on the size of the predefined grid. Instead, our paper presents a novel concept called cointeresting points among various views directions. Cointeresting points are not limited by the predetermined cell size of a grid and can be used to detect ROIs of both small and large sizes.

III. PRELIMINARIES

First, we consider the example in Figure 2. Four photos (i.e., p_1 to p_4) of an attraction were taken. The algorithms presented in previous studies performed clustering based on the photograph locations to discover ROIs and assess similarities. Accordingly, as shown in Figure 2, p_1, p_2, p_3, and p_4 do not form a meaningful ROI cluster because the number of photos cannot meet the minimum density threshold for density clustering.



(a) Introvert RealROI (b) Extroverted RealRoI

FIGURE 2. Examples of introverted and extroverted ROIs.

The objective of this paper is to use the view direction of each photo to find ROIs that cannot be found using existing algorithms. The arrows starting from the locations of users (i.e., p_1, p_2, p_3, and p_4) represent the view directions and distances of the photos in Figure 2(a). Stars (i.e., i_1, i_2, i_3, i_4, i_5, *and* i_6) are the intersection points between two view directions. In this example, we can identify four view directions (i.e., p_1, p_2, p_3, and p_4) and six intersections (i.e., i_1, i_2, i_3, i_4, i_5, and i_6) among these view directions. We refer to the intersection of two view directions as cointeresting points (*coIP*). Intuitively, the clustering of *coIP*s can produce the RealROI, which we call the introverted RealROI. Figure 2(a) is also an example of a RealROI that cannot be identified in the traditional approach.

In addition, we can identify some semantic regions by discovering photos taken from the ROI facing outwards. Figure 2(b) depicts a case in which photos of the surrounding landscape are taken from an observatory. For example, many photos of the New York cityscape are taken from the Empire State Building observatory. In this case, the algorithm for finding intersections does not perform well. However, by drawing a straight line in the opposite direction from which the photo was taken, the intersections can be found. This method also works well in indoor environments in which GPS signals cannot be received. For example, the user may also want to take photos of art hanging from a wall in a museum. Accordingly, the intersections among the reverse view directions of photos can be found by considering the forward and reverse directions together; this reverse co-origin point of interest is called *orgIP*. A region that forms a cluster when the reverse coIP reaches a certain threshold can be defined as an extroverted RealROI for common user interest. The following sections describe the algorithms used to establish introverted and extroverted RealROIs.

TABLE 1. Notations and descriptions.

Notation	Description
p	A geotagged photo with a view direction
P	A set of photos with view directions
d_{min},d_{max}	Min and max view distances from $p.loc$
$\overrightarrow{vs}_{p_a}$	A view segment for a photo p_a
$\overleftarrow{vs}_{p_a}$	A reverse view segment for a photo p_a
coIP	A cointeresting point
orgIP	A co-origin interesting point
iRealROI	An introverted *RealROI*
eRealROI	An extroverted *RealROI*

IV. DISCOVERING RealROIs

The proposed approach follows three main steps: 1) construction of view segments from geotagged photos, 2) identification of cointeresting points, and 3) clustering of cointeresting points to find the RealROI.

A. DISCOVERY OF INTROVERTED RealROIs

Database of geotagged photos P consists of view direction and distances taken by various users, $P = \{p_1, p_2, \ldots, p_n$. The geotagged photo is defined as follows.

Definition 1 (Geotagged Photo, p): A geotagged photo p with a view direction is a tuple of

$$(pid, uid, uri, loc, d, t)$$

where *pid* is the photo identifier, *uid* is the identifier of the owner of the photo, *uri* is the identifier of the location and filename of the actual photo file, *loc* represents the spatial coordinates of the photo, d is the photographing azimuth, and t is the time at which the photo was taken.

A geotagged photo has unique *pid* and *uid* information. The *uri* can be used to identify the geographic location of a user. In addition, a camera with a built-in GPS can produce the coordinates (i.e., latitude and longitude) of a photo location (i.e., *loc*). $\vec{d}$ is the view direction of a geotagged photo (i.e., the azimuth of the lens center). The geomagnetic azimuth information of the camera can be obtained with a geomagnetic sensor. t is the time at which the photo was taken, as represented by a long-type timestamp, such as Unix epoch time.

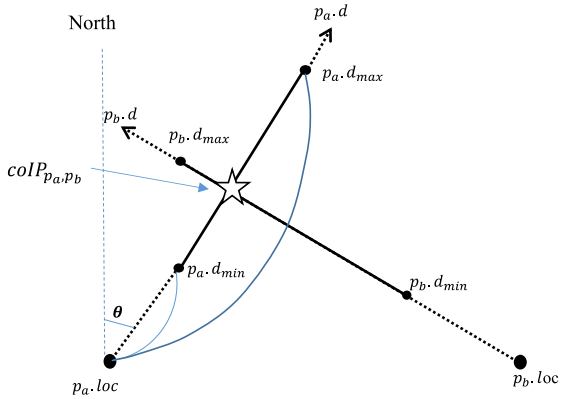


FIGURE 3. A cointeresting point between p_a and p_b.

Figure 3 shows an example of geotagged photos with view directions. Two photos (i.e., p_a and p_b) were taken toward the same target. $p_a.d$ is expressed as an angle θ in the clockwise direction with respect to the north and has a real value ranging from 0 to 360.

Definition 2 (Minimum and Maximum View Distances, d_{min} and d_{max}): The minimum view distance is d_{min}, and the maximum view distance is d_{max} for geotagged photo p.

In our approach, we use the minimum view distance d_{min} and the maximum view distance d_{max} to identify *coIP*s based on two geotagged photos. The view distance (i.e., d_{min} and d_{max}) is important for identifying a RealROI. For example, for a photo taken at home or in an art gallery, the photo view will be restricted, ranging from 30 cm to 10 m. Sometimes, the target-point far from the location where a photo was taken (e.g., the Eiffel Tower or a natural landscape) may have a large photo view, ranging from 30 m to 100 m. This view distance can be used to remove unimportant *coIP*s and identify the RealROI. If d_{min} and d_{max} are too small or large, then the algorithm may create many intersections among view directions, and most of them will be irrelevant. We can use d_{min} and d_{max} to construct view segments from geotagged photos.

Definition 3 (View Segment, $\overrightarrow{vs}$): Given the minimum and maximum view distances d_{min} and d_{max} and a geotagged photo p_a, a view segment is a directed line segment

$$\overrightarrow{vs}_{p_a} = (p_a.d_{min}, p_a.d_{max})$$

where $p_a.d_{min}$ is the starting point of the segment and $p_a.d_{max}$ is the end point toward $p_a.d$ from $p_a.loc$.

Figure 3 illustrates a view segment for p_a: (i.e., $\overrightarrow{vs}_{p_a}$ $(p_a.d_{min}, p_a.d_{max})$) and a view segment for p_b: (i.e., $\overrightarrow{vs}_{p_b} = (p_b.d_{min}, p_b.d_{max})$).

Definition 4 (Cointeresting Point, coIP): Given two geotagged photos p_a and p_b, and an intersection point between $\overrightarrow{vs}_{p_a}$ and $\overrightarrow{vs}_{p_b}$ with d_{min} and d_{max} (i.e., lx), a cointeresting point is defined as

$$coIP_{p_a,p_b} = (lx, p_a, p_b)\ ,\ \text{where } p_a.uid \neq p_b.uid.$$

Assume that two different users have taken photos facing the same attraction. Then it is evident that there is a potential attraction at the point where the two view segments of the two photos intersect each other. We refer to this point as a cointeresting point *coIP* between the two geotagged photos. A *coIP* is an intersection point between two view segments and is located at l_{p_a,p_b} based on (lat, long). A *coIP* has the form of a tuple (l_{p_a,p_b}, p_a, p_b) for two photos p_a and p_b. Users may take many photos facing the same point of interest and share these photos to others. However, a certain target object may not be attractive since one user took many photos of the same object of interest (e.g., photos of a child in his or her home). To address this issue for high-frequency uploaders, the corresponding *coIP*s are excluded with the condition $p_a.uid \neq p_b.uid$.

Definition 5 (Neighborhood coIP): Let ϵ be the distance threshold. Two cointeresting points $coIP_{p_a,p_b}$ and $coIP_{p_q,p_r}$ are neighboring if the following condition is satisfied:

$$dist(coIP_{p_a,p_b}.lx - coIP_{p_q,p_r}.lx) < \epsilon$$

We use the density-based clustering method (e.g., DBSCAN) to group *coIP*s and discover RealROIs. The proposed density-based clustering method requires three user-specified parameters: (1) the distance threshold ϵ, (2) the density threshold k, (3) and the minimum user threshold m with a unique ID *uid*.

Definition 6 (Core coIP): A *core coIP* is a *coIP* if the following conditions are satisfied:

1) $|neighborhood\ coIP| \geq k$
2) |unique users in the *neighborhood coIP*| $\geq m$.

where k is the density threshold of a geotagged photo and m is the minimum user threshold for a given *uid*.

A *core coIP* and its neighborhood are considered as one cluster. A *coIP* is a core *coIP* if the number of *neighborhood coIPs* within a given radius ϵ from the *coIP* is at least k and if the number of neighborhood users (i.e., *uid*s) is at least m.

Definition 7 (Noise and Border coIPs): A *coIP* is a *noise coIP* if the *coIP* is not a *core coIP* and none of its neighbors

are *core coIPs*. A *coIP* is a *border coIP* if the *coIP* is not a *core coIP* and at least one of its neighbors is a *core coIP*.

Given the definitions of *core*, *border*, and *noise coIPs*, our proposed approach incrementally groups two *core coIP*s to construct a cluster. If the *core coIP* is close enough to another *core coIP*, we group these two *coIP*s to construct one cluster. *Noise coIP*s are discarded.

Definition 8 (Introverted RealROIs, iRealROIs): Given the distance threshold ε, minimum number of *coIP*s k and minimum number of distinct *uid*s m within ε,

$$iRealRoI = \{colP_l \in colP \wedge colP_m \in colP| \\ densityConnected(coIP_l, coIP_m \varepsilon, k, m)\}.$$

When $coIP_l$ and $coIP_m$ are *core coIP*s or *border coIP*s with the minimum number of *neighborhood coIP*s, the two *coIP*s are reachable within a given distance threshold ε. We can define this as $densityConnected(coIP_l, coIP_m \varepsilon, k, m)$.

Algorithm 1 Discovering *IRealROIs*

Input: $\boldsymbol{P}$, d_{min}, d_{max}, ε, k, m
Output: A set of *iRealROI*s, *IRealROI*
1: *IRealROI* $\leftarrow \emptyset$// A set of *iRealROIs*
2: *CoIP* $\leftarrow \emptyset$// A set of *coIP*s
3: // Construct a set of view segments based on d_{min} and d_{max}
4: **foreach** $p \in P$ **do**
5: $\overrightarrow{vs} \leftarrow$ extract a view segment from p using d_{min}, d_{max}
6: $P_v \leftarrow P_v \cup (\overrightarrow{vs}, p)$// add $\overrightarrow{vs}$ to P_v with p
7: *CoIP* $\leftarrow$ find *coIP*s among view segments in P_v
8: *IRealROIs* $\leftarrow$ CoIP_Clustering(*CoIP*, ε, k, m)
9: **return** *IRealROI*

Algorithm 1 shows the procedure of finding a set of *iRealROI*s. The first step is constructing a set of view segments with all photos based on d_{min} and d_{max} and camera view directions (lines 4-6). Then, the *coIP*s are identified from the set of view vectors P_v. A plane sweep algorithm is used to identify line segment intersections (line 7). In this step, a line intersection algorithm can be used. Next, clusters are formed with a density-based algorithm and the additional condition m (line 8). Our approach uses a modified density clustering algorithm for c*oIP*s. Finally, the function returns a set of *iRealROI*s which are produced from density-based clustering (line 9).

Algorithm 2 shows the modified DBSCAN algorithm with condition m. First, neighbors within a certain radius ε are identified (line 4). If the number of neighboring points is less than k or the number of users is less than m, a *coIP* is classified as noise (lines 5). If the *core coIP* condition is satisfied, all neighbors are designated as $iRealROI_i$ with core *coIPs* (line 8). Additionally, the core *coIP* conditions can be checked for all neighbors and will be expanded to the satisfied neighbors in the same $iRealROI_i$ (lines 10-17). Then, neighbors at neighboring points are identified (lines 11-12). If $coIP_{cur}$ meets the *core coIP* condition, the remaining $coIP_{cur}$ neighbors, excluding noise *COIPs*, are added to the current $iRealROI_i$ (lines 15-16). The added neighbors are then checked repeatedly again (line 10). After the while loop finished, the $iRealROI_i$ is added to *IRealROI* (line 18). Finally, *IRealROI* is returned when the algorithm is finished (line 20).

Algorithm 2 CoIP_Clustering

Input: *CoIP*, ε, k, m
Output: A set of *iRealROI*s, *IRealROI*
1. $i \leftarrow 1$
2. **foreach** $coIP_a \in CoIP$ **do**
3. **if** ! IsROI($coIP_a$) **then**
4. $X \leftarrow$ NeighborhoodCoIPs(*CoIP*, $coIP_a$, ε)
5. **if** $|X| < k \vee |owners\,(X)| < m$ **then**
6. Mark_AsNoise($coIP_a$)
7. **else**
8. Mark_AsRoI($all(X)$, $iRealROI_i$)
9. Push($all(X)$)
10. **While** $X \neq \emptyset$ **do**
11. $coIP_{cur} \leftarrow$ Pop(X)
12. $Y \leftarrow$ NeighborhoodCoIPs(*CoIP*, $coIP_{cur}$, ε)
13. **if** $|Y| \geq k \wedge |owners\,(Y)| \geq m$ **then**
14. **foreach** $coIP_b \in Y$ **do**
15. **if** ! IsNoise ($coIP_b$) $\wedge$! IsRoI($coIP_b$) **then**
16. Mark_AsRoI($coIP_b$, $iRealROI_i$)
17. Push($coIP_b$)
18. *IRealROI* $\leftarrow$ *IRealROI* $\cup$ $iRealROI_i$
19. $i \leftarrow i + 1$
20. **return** *IRealROI*

Many existing studies have sought to find spatially cointerested user groups. A spatially cointerested user group is the basic unit for recommendation in location-based services. In the traditional approach, density-based clustering is the most popular algorithm for discovering spatially cointerested groups. However, this approach can be limited when users are dispersed, as shown in Figure 4. Although users are strongly interested in the same spatial feature, traditional methods cannot discover these user groups.

Definition 9 (Cointerested Real Users: CRUs): A CRU is a set of cointerested users who own the geotagged photos of a *coIP* in an *iRealROI*.

One of the notable advantages of the proposed view direction-based algorithm is that it can discover a spatially cointerested group to make recommendations from dispersed users and photos. Additionally, we can obtain explanatory knowledge from the directionality information among various *IRealROIs* and *CRUs* that are distributed in a geographical space, as shown in Figure 4.

Algorithm 3 shows a generalized algorithm that can discover *CRUs* for an *iRealROI*. An *iRealROI* is a set of *coIP*s that are of consistent interest to the same target object. In definition 4, we define each *coIP* to contain information about cointeresting geotagged photos. For each *coIP* in the

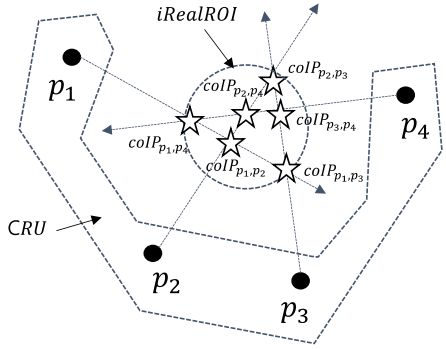


FIGURE 4. Example of an *iRealROI* and a *CRU*.

iRealROI, the algorithm can find a *CRU* by collecting the *uids* of the users in $coIP_x.p_a$ and $coIP_x.p_b$ (line 3).

Algorithm 3 Discovering *CRUs* From an *iRealROI*

Input: *iRealROI*
Output: *CRU*
1. ***CRU*** $\leftarrow \emptyset$
2. **foreach** $coIP_x \in iRealROI$ **do**
3. $\quad CRU \leftarrow CRU \cup coIP_x.p_a.uid \cup coIP_x.p_b.uid$
4. **return** *CRU*

B. DISCOVERY OF EXTROVERTED RealROIs

A RealROI can be obtained from an *iRealROI* based on the view directions of photos, and an extroverted RealROI can be used to find RealROIs based on the opposite extensions of view directions. This method uses not only the photos taken facing the target object but also photos taken of the surrounding scenery from a RealROI. This is useful, for example, where the user wants to capture the scenery of Manhattan from the Empire State Building observatory or the surrounding natural scenery from the top of a tall mountain. The extroverted RealROI method can be used to supplement the introverted RealROI method. This is useful for finding RealROIs when the introverted RealROI method may miss.

Definition 10 (Reverse View Segment, $\overleftarrow{vs}$): Given a reverse view direction $\overleftarrow{d}$ and a maximum view distance $\overleftarrow{d}_{max}$, a reverse view segment for a geotagged photo p_a is a directed line segment

$$\overleftarrow{vs}_{p_a} = (p_a.loc, p_a.\overleftarrow{d}_{max})$$

where $p_a.loc$ is the starting point of the reverse view segment and $p_a.\overleftarrow{d}_{max}$ is the end point toward $p_a.\overleftarrow{d}$ from $p_a.loc$.

When a user takes a photo in the view direction d, the reverse direction of the photo is defined as $\overleftarrow{d}$. Figure 5 shows two reverse view directions $p_a.\overleftarrow{d}$ and $p_b.\overleftarrow{d}$, which originate from $p_a.loc$ and $p_b.loc$, respectively. Assume that the reverse view directions are infinitely long. This will lead to many unnecessary operations and make the computation slower. To remedy this issue, we use $\overleftarrow{d}_{max}$ to limit the effective part

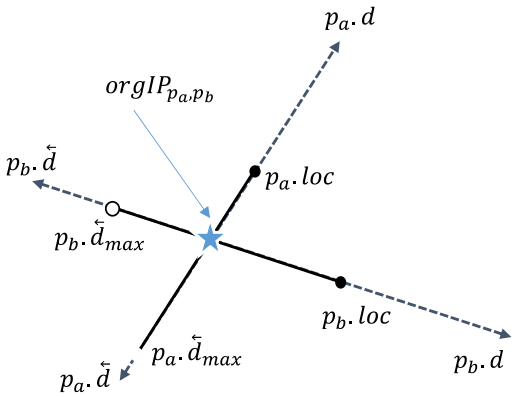


FIGURE 5. An interesting origin point between p_a and p_b.

of the view direction. A reverse view segment $\overleftarrow{vs}_{p_a}$ is a valid segment from $p_a.loc$ to $p_a.\overleftarrow{d}_{max}$ that can start from the same effective origin point.

Definition 11 (Co-Origin Point of Interest, orgIP): Given two geotagged photos p_a and p_b, a co-origin point of interest can be defined as

$$orgIP_{p_a,p_b} = (\overleftarrow{lx}, p_a, p_b) \text{ where } p_a.uid \neq p_b.uid$$

where $\overleftarrow{lx}$ is an intersection point between two reverse view segments $\overleftarrow{vs}_{p_a}$ and $\overleftarrow{vs}_{p_b}$ within $\overleftarrow{d}_{max}$.

An *orgIP* is a point at which reverse view directions intersect, denoted as location *lx*. An *orgIP* can be represented as a tuple $(\overleftarrow{lx}, p_a, p_b)$ for two photos p_a and p_b, and this representation is similar to that for a *coIP*. Given geotagged photos taken facing outward from nearby points by two different users, we can see that there is a candidate origin point for the two photos when two user view segments $\overleftarrow{vs}$ intersect each other. We refer to this intersection as the location of a co-origin point of interest *orgIP* between the two photos.

The extroverted RealROI algorithm uses the same density-based clustering approach as the one used for the introverted RealROI algorithm. Therefore, we skip the details because Definitions 5 to 7 can be extended to deal with extroverted RealROI.

Definition 12 (Extroverted RealROIs, eRealROIs): Given distance threshold ε, the minimum number of *orgIP*s k and the minimum number of distinct *uids* m within ε,

$$eRealRoI = \{orgIP_l \in orgIP \wedge orgIP_m \in orgIP | \\ densityConnected(orgIP_l, orgIP_m \varepsilon, k, m)\}.$$

The algorithm for finding *eRealROIs* is the same as that for finding *iRealROIs* after replacing *coIP* with *orgIP*. Therefore, a detailed algorithm is not presented.

C. ALGEBRAIC COST MODEL

We developed a cost model for the proposed algorithm to estimate the computational cost. Let n be the number of line segments. The cost of the proposed algorithm consists of

two major parts. The first part involves finding all intersections to create *coIP*s among $\overrightarrow{vs}$ segments. The corresponding cost is $O(n^2)$. Let k be the number of intersections. Then, the cost can be reduced to $O((n+k)\log n)$ by using the Bentley–Ottmann algorithm [17]. The second part involves establishing a set of clusters using the density-based clustering method at a cost of $O(k \log k)$ [18]. Therefore, the total cost of the proposed approach is $\mathrm{O}\left((n+k)\log n + k \log k\right)$.

V. EXPERIMENTAL EVALUATION

We conducted experiments to evaluate the performance of the proposed algorithms. We wanted to answer four questions: (1) What is the effect of the distance threshold ε, (2) What is the effect of the density threshold k, (3) What is the effect of the minimum user threshold m, and (4) What is the effect of the view distances d_{min} and d_{max}?

A. EXPERIMENTAL SETUP

We used a real-world geotagged photos dataset from Flickr in the experiments. We used photos from around New York within a 100 km radius of 40.730610 degrees latitude and −73.935242 degrees longitude. We collected 578,155 geotagged photos using the Flicker API and chose 87,442 geotagged photos with view directions. We conducted experiments using two methods: (1) the GPS ROI method and (2) the RealROI method. The algorithms were implemented in Python run in Windows 10. All experiments were performed on an Intel i7-4770 3.40 GHz 4-Core CPU with 16 GB RAM.

B. EFFECT OF THE DISTANCE THRESHOLD

The first experiment evaluated the effect of the distance threshold ε to validate the efficiency and stability of the algorithm. The average size of the discovered area and distance threshold ε were used to construct a neighborhood. Figure 6 shows the average size of the discovered area (m^2). The distance threshold was varied from 100 m to 500 m. The average size of the RealROI increases linearly, whereas the average size of the GPS ROI increases quadratically.

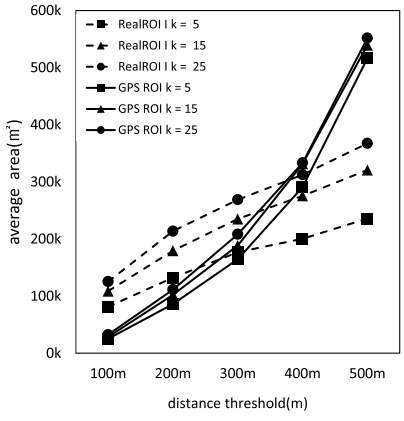


FIGURE 6. Effect of the distance threshold (ε).

We obtained similar results when we varied k from 5 to 25. Notably, when the size of an attraction increases linearly, photographers will be distributed in two main portions of the surrounding area.

It is important to note that when the distance threshold decreases (e.g., 100 m and 200 m), the average size of the RealROI area becomes larger than that of the GPS ROI. This is because photographers located in very small areas when the size of the ROI is larger than the photo area.

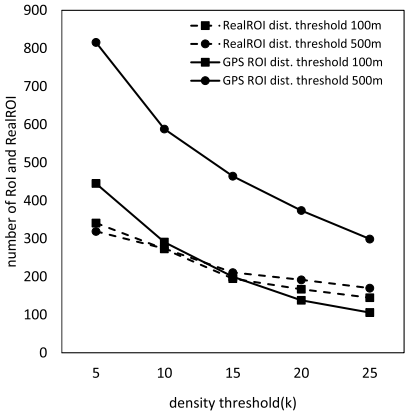


FIGURE 7. Effect of the density threshold (k).

C. EFFECT OF THE DENSITY THRESHOLD

The second set of experiments evaluated the effect of the density threshold on the robustness of the algorithm. We varied the value of k and compared the performance of the GPS ROI and RealROI methods (see Figure 7). Robustness was measured based on the number of discovered areas. We varied the density threshold k from 5 to 25. As k increases, the number of RealROIs decreases. However, the number of GPS ROIs significantly decreases compared to number of RealROIs. We obtain similar results when we increase the maximum distance threshold d_{max} from 100 m to 500 m because a limited number of RealROIs are available in real-world datasets. Thus, the proposed RealROI algorithm is robust at different density thresholds.

D. EFFECT OF THE MINIMUM USER THRESHOLD

The third set of experiments evaluated the effect of the minimum user threshold on the performance of the algorithm. We varied the value of the minimum user threshold m. Performance was assessed based on the number of discovered RealROIs. We evaluated the minimum user threshold m at values of 3 and 4. The distance threshold d_{max} was varied from 100 m to 500 m when k varied from 5 to 25.

Figure 8 shows that the number of RealROIs for a minimum user threshold of 3 significantly decreases compared with that for a threshold of 4. The minimum user threshold m is the number of unique users within a distance threshold ε. Thus, a high minimum user threshold value leads to more robust results than those for a low value. However,

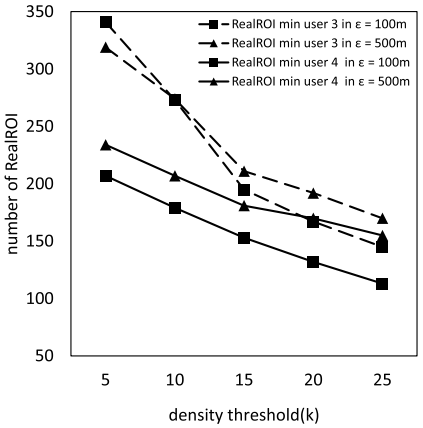


FIGURE 8. Effect of the minimum user threshold (*m*).

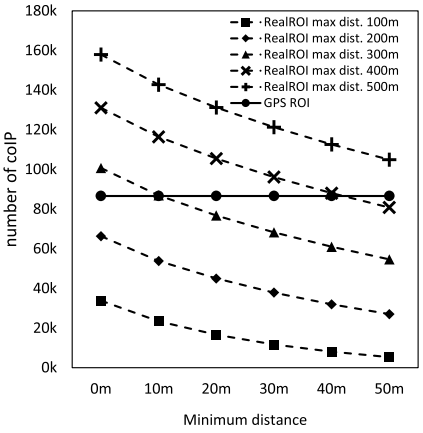


FIGURE 9. Effect of the view distance (d_{min} and d_{max}).

as k increases, the difference between the number of discovered RealROIs for each minimum user threshold decreases.

E. EFFECT OF THE VIEW DISTANCE

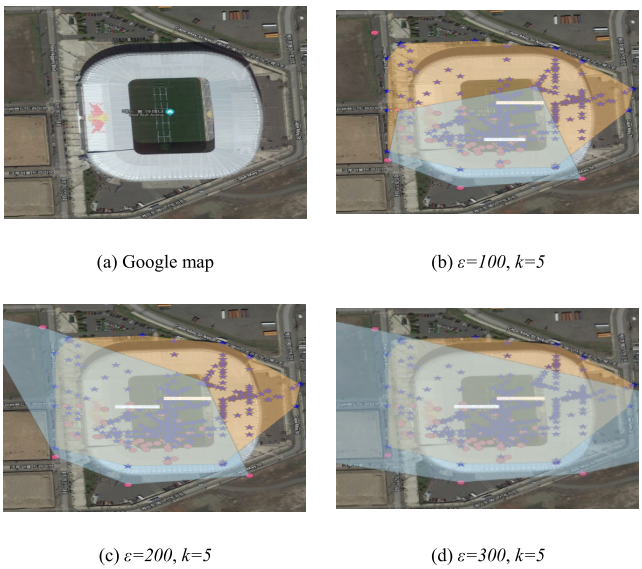
In the final experiments, we evaluated the effect of the minimum view distance d_{min} and maximum view distance d_{max} on the robustness and performance of the algorithm. We varied the value of the minimum view distance d_{min} from 10 m to 50 m. Performance was based on the number of discovered *coIP*s. As shown in Figure 9, we evaluated the algorithm for maximum view distances d_{max} of 100 m, 200 m, 300 m, 400 m, and 500 m. The GPS ROI method was evaluated for reference. The number of *coIPs* decreases linearly with increasing maximum view distance d_{max}. Additionally, the differences among results in each case were relatively constant.

VI. CASE STUDY AND DISCUSSIONS

A. CASE STUDY

In this section, we show an example case study comparing RealROIs and GPS ROIs around Redbull Arena. Red Bull Arena is a soccer-specific stadium in Harrison, New Jersey, USA, that is home to the New York Red Bulls of Major League Soccer, as shown in Figure 5(a). During the season, many spectators take pictures and share them through social networking services during matches at the stadium. As described in Section 4, our experimental datasets were collected from Flickr. More than a thousand photos were taken in the area around the stadium. RealROIs and GPS ROIs were simultaneously identified around the stadium. When changing the parameters in the experiment, the variations in GPS ROIs and RealROIs around the stadium clearly demonstrate the strength of the proposed RealROI algorithm.

In this case, we evaluated the effect of the distance threshold ε to validate the efficiency and stability of the algorithm. The distance threshold ε was varied from 100 m to 300 m, and the minimum user threshold k was fixed at five. Figure 10(b) shows a RealROI and a GPS ROI when $\varepsilon = 100$ m and k=5. In the figure, the red dots are the GPS locations of the photos. Blue stars are the *coIPs* among the view directions of the photos. Additionally, a yellow polygon is a RealROI. The yellow translucent polygon is the RealROI found with the proposed algorithm, and the blue translucent polygon is the GPS ROI found with the traditional algorithm. In this figure, we found that the RealROI covers a much larger area than the GPS ROI, and this area includes the entire stadium. Additionally, the GPS ROI area is smaller and skewed to one side of the stadium.



(a) Google map (b) ε=*100*, k=*5*

(c) ε=*200*, k=*5* (d) ε=*300*, k=*5*

FIGURE 10. Case study: RealROI vs. GPS ROI around Redbull Arena.

Figure 10(c) and Figure 10(d) show the RealROI and the GPS ROI when ε was changed to 200 m and 300 m under the same conditions. In these figures, we can see interesting changes in the RealROI and GPS ROI. In contrast to the growing GPS ROI when ε increases, the RealROI remains almost invariably the same. The change in the GPS ROI is a common result of traditional density-based clustering. However, the size of the RealROI slightly changes and the RealROI covers the actual stadium ROI. This result was consistent with the expected result. Similar results to those

in this case were obtained for other famous ROIs, such as the Statue of Liberty in New York. The results of the experiments in Section 4.b support the robustness and effectiveness of the proposed approach in real-world cases.

B. DISCUSSIONS

Our proposed approach achieves a significant performance improvement over the related work. This improvement was obtained by using three key components: 1) construction of view segments from geotagged photos, 2) identification of cointeresting points, and 3) clustering of cointeresting points to find the RealROI. The core idea of our algorithm is to use geometric and directional information to identify the real regions of Interest (RealROIs). The existing method with only GPS datasets fails to find RealROIs due to a lack of directional information. The case study shows that our approach correctly identifies RealROIs whereas the related work cannot find RealROIs. We also see that our approach produces more stable results over the changes of parameters compared to the related work.

VII. CONCLUSION

In this paper, we propose a novel algorithm that discovers the RealROI from geotagged photos from social media with camera view directions. The proposed RealROI algorithm was verified to be superior to the traditional GPS ROI method based only on GPS coordinates because it can find RealROIs that cannot be found using GPS coordinates alone and identify ROIs in high-density POI areas. In this paper, algorithms for discovering RealROIs require significant computing times. In future studies, we will propose methods based on indexing to optimize the execution time of this algorithm.

REFERENCES

[1] S. Kisilevich, F. Mansmann, and D. Keim, "P-DBSCAN: A density based clustering algorithm for exploration and analysis of attractive areas using collections of geo-tagged photos," in *Proc. 1st Int. Conf. Exhib. Comput. Geospatial Res. Appl. (COM)*, 2010, p. 38.

[2] T. Kurashima, T. Iwata, G. Irie, and K. Fujimura, "Travel route recommendation using geotags in photo sharing sites," in *Proc. 19th ACM Int. Conf. Inf. Knowl. Manage. (CIKM)*, 2010, pp. 579–588.

[3] Y. Zheng, Y. Li, Z. Zha, and T. Chua, "Mining travel patterns from GPS-tagged photos," *ACM Trans. Intell. Syst. Technol.*, vol. 3, no. 3, p. 56, 2012.

[4] S. Kisilevich, D. Keim, N. Andrienko, and G. Andrienko, "Towards acquisition of semantics of places and events by multi-perspective analysis of geotagged photo collections," in *Geospatial Visualisation*. Berlin, Germany: Springer, 2013, pp. 211–233.

[5] I. Memon, L. Chen, A. Majid, M. Lv, I. Hussain, and G. Chen, "Travel recommendation using Geo-tagged photos in social media for tourist," *Wireless Pers. Commun.*, vol. 80, no. 4, pp. 1347–1362, 2014.

[6] M. Park, J. Luo, R. T. Collins, and Y. Liu, "Beyond GPS: Determining the camera viewing direction of a geotagged image," in *Proc. Int. Conf. Multimedia (MM)*, 2010, pp. 631–634.

[7] Y. Cai, Y. Lu, S. H. Kim, L. Nocera, and C. Shahabi, "Gift: A geospatial image and video filtering tool for computer vision applications with geo-tagged mobile videos," in *Proc. IEEE Int. Conf. Multimedia Expo Workshops (ICMEW)*, Jun. 2015, pp. 1–6.

[8] Y. Chen, A. Cheng, and W. H. Hsu, "Travel recommendation by mining people attributes and travel group types from community-contributed photos," *IEEE Trans. Multimedia*, vol. 15, no. 6, pp. 1283–1295, 2013.

[9] W. Ding, K. Yang, and K. W. Nam, "Measuring similarity between geo-tagged videos using largest common view," *Electron. Lett.*, vol. 55, no. 8, pp. 450–452, 2019.

[10] S. Kim, Y. Lu, and G. Constantinou, "MediaQ: Mobile multimedia management system," in *Proc. ACM Multimedia Syst. Conf.*, 2014, pp. 224–235.

[11] Y. Zheng, "Mining travel patterns from geotagged photos," *ACM Trans. Intell. Syst. Technol.*, vol. 3, no. 3, pp. 1–18, 2012.

[12] Z. Zeng, R. Zhang, X. Liu, X. Guo, and H. Sun, "Generating tourism path from trajectories and geo-photos," in *Proc. Int. Conf. Web Inf. Syst. Eng.*, 2012, pp. 199–212.

[13] E. Spyrou, I. Sofianos, and P. Mylonas, "Mining tourist routes from Flickr photos," in *Proc. 10th Int. Workshop Semantic Social Media Adaptation Personalization (SMAP)*, Nov. 2015, pp. 1–5.

[14] M. Shirai, M. Hirota, H. Ishikawa, and S. Yokoyama, "A method of area of interest and shooting spot detection using geo-tagged photographs," in *Proc. 1st ACM SIGSPATIAL Int. Workshop Comput. Models Place*, 2013, pp. 34–41.

[15] M. Shirai, M. Hirota, S. Yokoyama, N. Fukuta, and H. Ishikawa, "Discovering multiple hotspots using geo-tagged photographs," in *Proc. 20th Int. Conf. Adv. Geographic Inf. Syst.*, 2012, pp. 490–493.

[16] M. Hirota, M. Endo, D. Kato, and H. Ishikawa, "Discovering hotspots using photographic orientation and angle of view from social media site," *Int. J. Inf. Soc.*, vol. 10, no. 3, pp. 109–117, 2019.

[17] J. L. Bentley and T. A. Ottmann, "Algorithms for reporting and counting geometric intersections," *IEEE Trans. Comput.*, vol. C-28, no. 9, pp. 643–647, 1979.

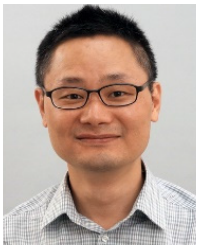
[18] H. Song and J.-G. Lee, "RP-DBSCAN: A superfast parallel DBSCAN algorithm based on random partitioning," in *Proc. Int. Conf. Manage. Data*, Houston, TX, USA, May 2018, pp. 1173–1187.

[19] D. Kim, Y. Kang, Y. Park, and J. Lee, "Understanding tourists' urban images with geotagged photos using convolutional neural networks," *Spatial Inf. Res.*, vol. 28, pp. 241–255, 2020, doi: 10.1007/s41324-019-00285-x.

[20] H. Lee and Y. Kang, "Mining tourists' destinations and preferences through LSTM-based text classification and spatial clustering using Flickr data," *Spatial Inf. Res.*, vol. 29, pp. 825–839, 2021, doi: 10.1007/s41324-021-00397-3.

[21] J. Y. Park, D. J. Ryu, K. W. Nam, I. Jang, M. Jang, and Y. Lee, "DeepDBSCAN: Deep density-based clustering for geo-tagged photos," *ISPRS Int. J. Geo-Inf.*, vol. 10, no. 8, p. 548, Aug. 2021.

KWANG WOO NAM (Member, IEEE) received the Ph.D. degree in computer science from Chungbuk National University. After Ph.D. degree, he studied location-based services and spatial database systems at the Electronics and Telecommunications Research Institute, South Korea. He is currently a Professor with the School of Computer, Information, and Communication Engineering, Kunsan National University, Republic of Korea. His research interests include spatial and moving objects database, spatial big data, and GeoAI.



KWANGSOO YANG (Member, IEEE) received the B.S. degree in electrical engineering from Yonsei University, in 1998, and the M.S. and Ph.D. degrees in computer science from the University of Minnesota, in 2010 and 2015, respectively. He was a Software Engineer for LG CNS, Seoul, South Korea, from 2001 to 2008. He is currently an Associate Professor with the Department of Computer and Electrical Engineering and Computer Science, Florida Atlantic University. His research interests include spatio-temporal network databases, spatio-temporal networks, and evacuation routing problems. He has received the NSF CAREER Award, in 2019.