GROUPING METHOD FOR NEIGHBOR OBJECTS OF MOVING OBJECT USING HASH INDEX

Baesung Lee, Jiyoung Kim, Kiyun Yu*

Dept. of Civil and Environmental Engineering, Seoul National University, 599 Gwanak-ro, Gwanak-gu, Seoul, Korea – (aimhigh1101, soodaq, kiyun)@snu.ac.kr

KEYWORDS: Grouping method, moving object, hash index

ABSTRACT:

A location-based social network (LBSNS) is a social network service (SNS) that is based on a user's location, unlike SNS, and facilitates additional activities like neighbour-user search and group formation in real time. This research examined the efficiency of a real-time grouping method for extracted neighbour objects on a WMTS (Web Map Tile Service) for which a hash index was used. As a result, the suggested method is not different from buffering a neighbour area according to the average number of returned data per 1 km^2, and the required time is shorter than that of buffering.

1. INTRODUCTION

Currently, the popularity of the LBSNS (location-based social network) model, a social network service (SNS) that is based on users' location data, is increasing. Users of these services benefit from the activity between existing friend relationships, whereby new relationships or new groups can be established after a user-neighbor search. Accordingly, the number of research studies concerning this subject are steadily increasing.

Song et al. (2010) proposed an efficient obtaining method, whereby a moving pattern is used in the event of a large amount of location data and the occurrence of unnecessary obtaining procedures is reduced; however, as a single server is limited in its capacity to serve many users, it is difficult to actually apply this proposed method.

Ahn et al. (2011) researched a community-forming model that can adaptively change based on a smartphone's location. The study proposes an LBSNS model that provides proper management of the formed community and automatic member updating. Since community relationships between smartphones can only be formed according to social links using previously searched POI (Point of Interest), however, the proposed method cannot include a large number of users and unspecified neighbor users.

To solve these problems, this research study proposed a neighbor-object grouping algorithm that can provide smooth service with a single server to numerous users in an actual capacity, and can extract neighbor users from many unspecified persons. Accordingly, this study's proposed neighbor-object grouping algorithm is based around moving objects in a WMTS (Web Map Tile Service), which is the standard protocol of a mobile-LBS base map.

2. GROUPING ALGORITHM OF NEIGHBOR OBJECTS AROUND MOVING OBJECTS BASED ON HASH INDEX

2.1 HASH INDEX

As a moving object increases its quantity and moves continuously, it generates large amounts of location-movement calculations in the database. For reducing the index-restructuring load, a hash-index method is proposed (Song, 2001). This method dissects the entire area into grids and then stores the number of grids in the database. When a moving object belongs to a certain grid, a location change is not enacted, and if it moves to another grid, this is reflected in the index; therefore the index-restructuring load is reduced (Kim, 2006). This hash-index method makes the speed of grouping neighbor objects faster, as the index remains unchanged when a moving object moves within the same grid.

2.2 PROPOSED ALGORITHM

The proposed method consists of the following four steps:

Step 1. Measure longitude and latitude of specific moving object, and calculate the map-tile index in which it is located.

Step 2. Extract neighbor objects from 9 tiles that consist of a step-1 tile and around 8 tiles.

Step 3. Grouping of extracted data as a unit

Step 4. If moving object randomly moves to another area, perform tiling (step 1 to step 3) if it moves within the same tile, but do not perform grouping (use previous grouping).

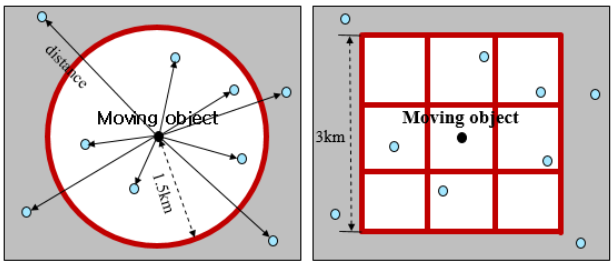
* Corresponding author

3. EXPERIMENT AND RESULT

3.1 DESIGNING EXPERIMENT

"Neighbor" commonly represents the circumference of a certain object. In this research, however, neighbor refers to the circle area according to a user's coordinates.

Commonly, neighbor-search on a commercial portal website uses an object-extraction method that is in the buffer area, which is referred to as "buffering method" in the remainder of this paper (Figure 1(a)). This research study therefore compared the efficiency of a proposed method, referred to as "tiling method" in the remainder of this paper (Figure 1(b)), with that of the buffing method, which is considered a comparable model.



(a) (b)

Figure 1. Example of buffering and tiling method

For this purpose, we measure the distance between the center object and neighbour objects using a haversine formula, whereby the average number of returned data per 1 km^2 and the average required time per second are compared.

3.2 EXPERIMENT ENVIRONMENT AND DATA

The experiment wherein we extracted neighbor objects by tiling and buffering uses a cloud-computing environment provided by AWS (Amazon Web Services) and an Amazon Linux AMI with a CentOS 6 base.

The hardware that was used is comprised of 1 RDS m3.large (vCPU 2, Memory 7.5 GB), 1 EC2 t2.medium (vCPU 2, Memory 4 GB), and 1 EC2 c3.xlarge (vCPU 4, Memory 7.5 GB).

For the software environment, MySQL version 5.6.19a is used, and PL/SQL and PHP are used for the algorithm. Both methods are compared using multiple threads.

To obtain data for the experiment, 5,000 location-information items were randomly extracted from the LBS application "Neighbor Shops," which was developed by Ministry of Land, Infrastructure and Transport, and they were then displayed on the Google-Map-facilitated WMTS. At the time of the data display, the buffer size of the buffer method was a 1.5 km radius that is applied for neighbor searches involving commercial portal services. The tiling method is performed at a Google-Map zoom level of 15 (the side length of a tile is 3 km, which is similar to the buffer size of 1.5 km (Figure 2)). Also, the thread of moving objects is increased to 140 at an interval of 20, based on a presumption that each moving object moves 100 times; that is, 20 moving objects move around 100 times, grouping neighbor objects each time, and the experiment is then conducted until the moving object is increased up to 140.

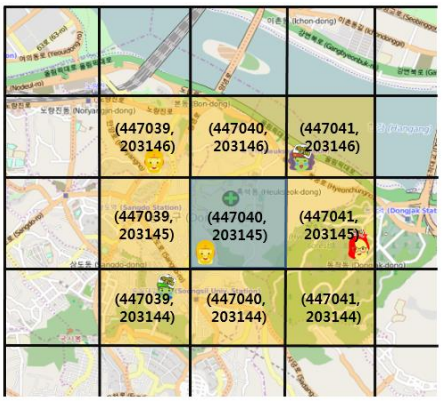


Figure 2. Tiles of Google map (zoom level = 15)

3.3 EVALUATION AND ANALYSIS

Regarding each case, the average number of returned data per area is shown in Table 1. Each data equals quotient of returned data divided by area (the area size of buffering method is $1.5^2\pi$ km^2, while the area size of tiling method is 3^2 km^2)

The average required time is shown in Table 2, as follows.

Table 1. Average number of returned data per area (# / km^2)

# of thread	Average number of returned data per area	
	Tiling method	Buffering method
20	5.3925	5.8383
40	5.1770	5.5541
60	5.2973	5.6609
80	5.1664	5.5064
100	5.1737	5.5074
120	5.1635	5.5435
140	5.2297	5.5636

Table 2. Average required time (secs)

# of thread	Average required time	
	Tiling method	Buffering method
20	0.0292	0.0565
40	0.0630	0.1330
60	0.0259	0.2405
80	0.0858	0.3177
100	0.2224	0.4274
120	0.3463	0.4881
140	0.1166	0.7198

The average number of returned data per area is almost the same for both the tiling and buffering methods; although, as the thread becomes larger, the average number of the returned data is constant. In the case of the average required time, however, the tiling method reduces the time at a considerable rate compared to the buffering method, proving that the tiling method is more efficient than the buffering method.

Additionally, when the thread is over 120, the average required time becomes abnormally longer, and similarly, the return of data starts to become abnormal. The following reason explains these abnormal changes: The proposed tiling method is stable until 120 moving objects move 100 times; however, in excess of 120, the capability to stably group neighbor objects is limited. Further study is therefore required if an actual service model is ever applied.

4. CONCLUSION

This research proposes a grouping method that extracts neighbor objects on the basis of a moving object's location according to the map tile; moreover, this method does not perform grouping when a moving object moves within the same tile. A buffering method was compared using 5,000 location-data items, whereby an actual map portal site that is currently popular was used, and the required time of the proposed method to group neighbor objects around a moving object in real time was verified as shorter than that of the buffering method. To actually apply the proposed method in the design of a stable service model, however, an additional study of a more diverse neighbor-object grouping method is needed.

REFERENCES

Song et al. (2010), Improved algorithm for location based alarming service, Database of Information Science Journal 37(1), pp 22–32

Ahn et al. (2011), Design of location based social network service model centering around smartphone, Journal of the korea industrial information system society 16(5), pp 55–62

Z. Song, N. Roussopoulos (2001), Hashing moving objects, Proc. of Mobile data management, pp. 161–172.

Kim (2006), The high-performance real-time storage management system for moving object trajectory, doctoral thesis, Inha University, Korea

ACKNOWLEDGEMENTS

This research was supported by a grant(15CHUD-C061156-05) from National Spatial Information Research Program funded by Ministry of Land, Infrastructure and Transport of Korean government.

This work was supported by the ICT R&D program of MSIP/IITP. [B0101-15-1349, Development of Volunteered Geospatial Information Platform Technology and Application for the Elderly and Disabled]