An Algorithm for Multi-Source Geographic Data System

Chiang-Sheng Lee, Hsine-Jen Tsai, and Yin-Yih Chang

1 Introduction

Geographic data is very diverse and dynamic. It is becoming a critical part of computing applications. Traditionally, geographic data is captured, stored and displayed in a single data source. Over the last decades, there has been an exponential increase in the amount of geographic data stored and available from multiple data sources. System designers need to develop integrated systems that allow users to access and manage information from several scattered geographic data sources at the same time to come up with a satisfactory answer. One reason for such need has been that environments for data access have changed from centralized data systems into multiple, distributed data sources. Another more recent cause for the attention of integration technologies is the emergence of Geographical Information System and its needs for accessing data repositories, application and legacy source that located across the organization intranet or on the Internet [Smart et al 2010].

The geographic data may be unstructured or semi-structured, and usually there is no a regular schema to describe them. As the amount of geographic data grows, the problem of integration among multiple data sources becomes a critical issue in developing distributed geographic systems. Many researchers have been interested in resolving the heterogeneity between geographic data sources, and different solutions have been proposed [Janowicz 2008, Ghulam 2010, Tsai 2011].

While interoperability of distributed information system gains much attention and studies, there is comparison little written about to gain a better performance of data integration of multiple data sources and it has being studied by many researchers in different applications. In [Song 2007], they focus on box covering algorithms that cover a computer network with the minimum possible number of boxes. They demonstrated that such covering problems can be mapped to the well-known graph coloring problem and argued that the algorithms presented provide a solution close to optimal. [Hert et al 1996] presented an online terrain-covering algorithm for a robot moving in an unknown three-dimensional underwater environment. They showed that the path length of their algorithm is shorter and to be linear in the size of the description of the boundary of the area. The focus of our study is to develop an algorithm that covers a region with the minimum possible number of geographic data, i.e. maps.

2 The System

This study is based on a map integration system which embeds in a large distributed data environment. A data server resides on top of scattered database systems and allows applications to access data from remote databases. In order to test our algorithm, we implement a system that would mimic that of a distributed data environment. Figure 1 provides a simple illustration of the map integration system in a distributed data environment.

While there are different format of a geographic data, the focus in our study is in the form of maps. The basic structure of the data in our system is a bounding box which is based on a R-tree structure [Guttman, 1984].

3 The Process

The process starts with the server receiving the request from users/applications. The request, again, is in the form of a bounding box. We assume that there is no data source in the

C.-S. Lee (✉)
Department of Industrial Management, National Taiwan University of Science and Technology, Taipei City, Taiwan, R.O.C
e-mail: cslee@mail.ntust.edu.tw

H.-J. Tsai • Y.-Y. Chang
Department of Information Management, Fu-Jen Catholic University, New Taipei City, Taiwan, R.O.C.
e-mail: tsai.fju@gmail.com; 042833@mail.fju.edu.tw

H. Selvaraj et al. (eds.), *Progress in Systems Engineering: Proceedings of the Twenty-Third International Conference on Systems Engineering*, Advances in Intelligent Systems and Computing 1089,
DOI 10.1007/978-3-319-08422-0_55,

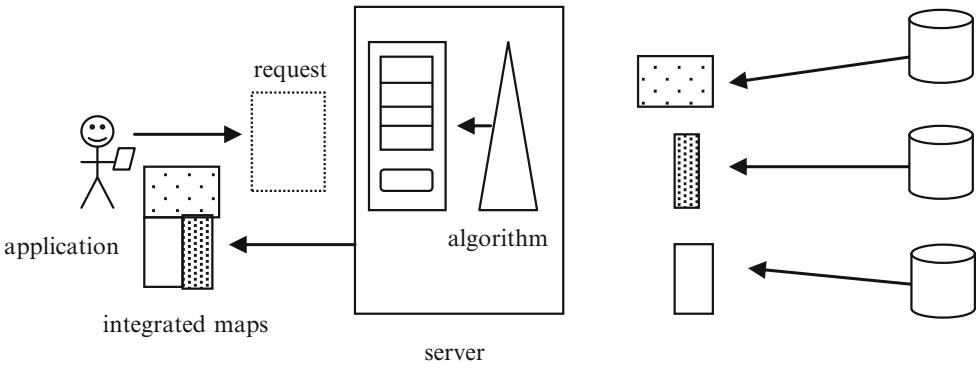

Fig. 1 The Map Integration System

distributed environment which can provide with a map that can cover the entire bounding box of the request. In another word, to response to the request, multiple geographic data need to be retrieve from the scattered data sources. The server of the system makes use of its database to locate data sources that can provide maps overlapping part of the requested map area.

The server then makes use of its algorithm to decide the covering sequence of those geographic data. By processing those geographic data one at a time, the region of the request area is partitioned into uncovered portion and covered portion. The process continues until either the collection of overlapped geographic data is empty or the requested map area is totally covered.

Not only accessing correct geographic data, but also performing the integration within limited time needs to be considered in a distributed geographic environment. The motivation of our algorithm is to increase the performance of the integration system by finding a set of possible minimum number of geographic data (i.e. maps). The algorithm not only finds the possible minimum set of geographic data but decides the covering sequence [Tsai et al 2013]. The following section gives more detailed descriptions of our algorithm.

4 The Algorithm and an Example

The objective of our algorithm is to find the minimum number of geographic data to cover the bounding box of the incoming request. A bounding box is an area defined by two longitudes and two latitudes and specified by the set of coordinates which represents the right-bottom and left-top of the bounding box. The right-bottom point is the minimum longitude and maximum latitude. The left-top point is specified by the maximum longitude and minimum latitude of the bounding box. Given a set of geographic data that partially overlap the request bounding box and a list of geographic data whose bounding boxes overlap part of the request bounding box, the algorithm starts the search by attempting to cover from the left-top corner of the request bounding box to the right-bottom corner of the bounding box.

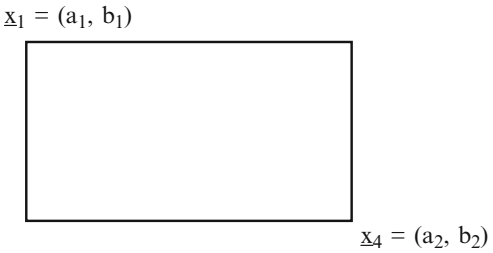

Fig. 2 A request bounding box with (a_1, b_1) and (a_2, b_2)

To achieve above goal, our algorithm is to decide the next adopted graphical data that will cover the current corner points as many as possible and repeat the same scheme until all corner points covered. Before introducing the main algorithm, we define the following notations.

1. Let P be the set of corner points of the region that has not be covered by the located geographic data. Initial values for P is the set of four corner points of the request bounding box. For example, $P = \{x_1 = (a_1, b_1), x_2 = (a_2, b_1), x_3 = (a_1, b_2), x_4 = (a_2, b_2)\}$ where x_1 and x_4 are the left-top and right-bottom corner points of the request bounding box, respectively. That is, a_1 is the minimum latitude and b_1 is the maximum longitude of the bounding box, a_2 and b_2 are the maximum latitude and minimum longitude, respectively. (see Fig.2)
2. Let $S_i = (s_{i1}, s_{i2}, s_{i3}, s_{i4})$ be the set of four longitude/latitude coordinates of the bounding box with (s_{i1}, s_{i2}) for the left-top corner point and (s_{i3}, s_{i4}) for the right-bottom corner point.

The algorithm includes the following steps:

Step1: Find the corner-point set P for the request area and set k=0 at the first time

Step2: Let ni be the number of corner points covered by the i'th geographic data from the database. Under the same maximum value of ni, put the last geographic data found into the list.

Step3: Set k=k+1 and execute the following two steps.

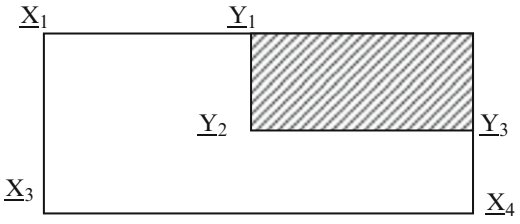

Fig. 3 The shaded graphical data after the first loop of the algorithm

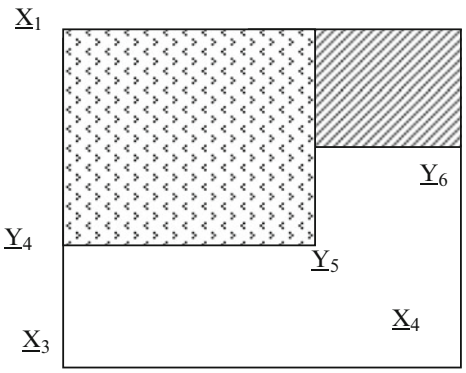

Fig. 4 The shaded graphical data after the second loop of the algorithm

(i) delete the corner points covered by the previous geographic data from the set P, and
(ii) add the new corner points to P if they are generated by the same geographic data.

If the set P is empty after Step3, then the algorithm stops. Otherwise, it goes to Step2.

The following example illustrates our algorithm's scheme. Let the bounding box of the request is specified by (a1, b1) and (a2, b2), the initial values are defined as follows:

$P = \{\underline{x}_1 = (a_1, b_1), \underline{x}_2 = (a_2, b_1), \underline{x}_3 = (a_1, b_2), \underline{x}_4 = (a_2, b_2)\}$ and $k = 0$.

Figure 3 shows the covered status of the request bounding box after algorithm finds geographic data that covers the request bounding box with maximum value of $n = 1$ and is shaded under the diagonal line. The set P has new corner points which are $P = \{ \underline{Y}_1, \underline{Y}_2, \underline{Y}_3, \underline{X}_1, \underline{X}_3, \underline{X}_4\}$

For one more example for this algorithm, Figure 4 shows that the algorithm finds the maximum value of n=3 and the second graphical data is also shaded. The new corner-point set P would be P = { Y3, Y4, Y5, X6, X3, X4} and K = 2. Since set P is not empty, the algorithm will continue until P becomes vacant.

5 Conclusion

In this work, we have demonstrated an approach to search mechanism in the context of geographic data integration. Our contribution has been the proposal of an algorithm to retrieve multiple geographic data to response a user's request of a geographic map covering a certain region. This approach leads to a process that ensures the possible minimum geographic data are located. The three steps of the approach allow such a property to be satisfied. The first step keeps the corner points of the bounding box of the request to ensure the total area of the request will be covered. By checking the corner points set the second step locates the geographic map that overlaps uncovered area and updates the corner points of the uncovered area. Since the geographic map located covers the maximum number of corner points the algorithm ensures a possible minimum number of geographic maps are located. The third step then updates the parameters of the algorithm and ensures the algorithm halts. As future work, we will consider a more complex algorithm that utilizes an optimization function.

Reference

1. P.Smart, C.Jones, and F.Twaroch, (2010) "Multi-source toponym data integration and mediation for a meta-gazetteer services", GIScience, LNCS 6292, 234-248.
2. K.Janowicz, M.Wilkes and M.AndlutzN, (2008) "Similarity-Based Information Retrieval and Its Role within Spatial Data Infrastructures", Geographic Information Science, pp. 151-167.
3. Ali Mohammad Ghulam, (2010) "A Framework for Creating Global Schema Using Global Views from Distributed Heterogeneous Relational Databases in Multi-database System", Global Journal of Computer Science and Technology Vol. 10 Issue 1 (Ver 1.0), pp. 31-35
4. A. Guttman, (1984) "A dynamic index structure for spatial searching", SIGMOD '84 Proceedings of the ACM SIGMOD international conference on Management of data, pp. 47 - 57
5. S. Hert, S. Tiwari, V. Lumelsky, (1996) "A terrain-covering algorithm for an AUV", Autonomous Robots 3, pp. 91-119.
6. Chaoming Song[1], Lazaros K Gallos[1], Shlomo Havlin[2], and Hernán A Makse, (2007) "How to calculate the fractal dimension of a complex network: the box covering algorithm", Journal of Statistical Mechanics: Theory and Experiment, pp.03006
7. Hsine-Jen Tsai, (2011) "A spatial mediator model for integrating heterogeneous spatial data", Dissertation, Iowa State University under the guidance of Dr. Les Miller
8. H-J Tsai, C-S. Lee and L. Miller. (2013) "A Search Mechanism for Geographic Information Processing System", In Proceedings of the Institute of Industrial Engineers Asian (IIE Asian) Conference, pp. 945 – 952.