An Intuitive Way to Describe Our World: A Microblog LBS Visualization System

Jian-Min Wang[1,2], Lai Gan[2], Ri-Peng Zhang[2], and Fang You[1,2]

[1] School of Arts and Media, Tongji University, Shanghai, China
wangjianmin@tongji.edu.cn
[2] School of Information Science and Technology, Sun Yat-sen University, Guangzhou, China
{503187646,pk_mati}@qq.com, youfang@tongji.edu.cn

Abstract. With the help of LBS provided by some social network service provider, people can obtain knowledge and perceive the world around them more conveniently. However, since the contradiction of a large number of POI nodes and the limited displaying space, finding a better layout algorithm to visualize the POI nodes is important for better user experience. This paper presents a new layout algorithm called Virtual Layout algorithm, which can solve the problem of the uneven POI nodes distribution and make a better presentation of the geo-spatial information. We also deployed a microblog LBS visualization system which consists of Virtual Layout algorithm, stylized map and location reference frame to verify the performance of the proposed algorithm.

Keywords: LBS visualization, Microblog, Layout algorithm, POI.

1 Introduction

Weibo.com, the main microblog service provider in China, with its convenient and social characteristics, is widely used by large amount of users. However, just publishing the common information (text/image) could not satisfy the users' demand, so the LBS (Location Based Service) was applied as a value-added service of Weibo.com. LBS is combination of different location-acquisition technologies, like Global Positioning System (GPS) and Geographic Information System (GIS). In detail, LBS obtains precise locating data (e.g., longitude & latitude) from mobile terminals, and combines them with the location profiles stored in the GIS database, providing users a more intuitive way to organize the geographic and social information of our real world [1].

With development of LBS and social network, people make much extra information for each GPS coordinates, which we called POI (Point Of Interest) [2]. Through POI data and LBS visualization, people can acquire and understand the contextual information around them more conveniently. For instance, we can present relationships between user and location, location and location, etc. Such kinds of information are bridges connecting the virtual world and the real world which can be inferred of some useful patterns.

S. Yamamoto (Ed.): HIMI 2014, Part I, LNCS 8521, pp. 108–119, 2014.

However, microblog LBS visualization is not just a way for demonstrating information [3]. In the process of creating an effective visualization, we should consider many factors such as scene, perception and cognition. An effective visualization should have a clear purpose and convey information directly. Unfortunately, there are still some problems on the visualization of microblog LBS.

- Most of the visualization systems are over dependent on the real map and weaken the relationship between different POI nodes. For instance, when we visualize dense or sparse POI nodes, it will lead to the result that some nodes are overlapped or some isolated nodes are too far away from the major region, which will make visualization illegible and waste of displaying space.
- As the displaying problem described above, owing to the limited displaying space of the overlapped POI nodes, we are not able to display enough valuable information on each POI node, so in some cases, it will lead to useless visualization.

In summary, developing an efficient model and a better visualization method for geospatial data to achieve more satisfactory visualization is worth detailed study [2]. Thus, in this paper, a geospatial model Weigeo and a layout algorithm Virtual Layout algorithm are developed for microblog LBS visualization. At the last of our research, we developed a microblog LBS visualization system based on the dataset obtained from Weibo.com to illustrate the Weigeo model and Virtual Layout algorithm. The microblog LBS visualization system, differing from the previous map visualization system, it provides a more even distribution of nodes for visualization, which can offer users a more clear way to understand the structural relationships between different POI nodes.

The paper is organized as follows: First, we describe some previous work on geospatial visualization; second, we describe the Weigeo model based on the POI data obtained from Weibo.com; third, we propose a new layout algorithm called Virtual Layout algorithm to rearrange the distribution of the POI nodes; fourth, we illustrate the performance of our visualization system; fifth, we present the implementation of our system; and last, we draw our conclusions and propose the future work we attempt to conduct.

2 Relative Work

Currently, the researchers mainly focus on the domains about LBS modeling and geospatial data modeling. For example, Xiaoyan Chen et al. [4] built an effective system for geospatial information publication and subscription, but this is a theoretical model and not directly manifest the relationship between different locations. Moreover, Christian S. Jensen et al. [5] proposed a multidimensional data model which well matching the geospatial data in order to reveal multidimensional information. Nevertheless, the data model is difficult to be applied in current geospatial data system. Thus, this model is not being implemented in practical use. Joao

Mourinho et al. [6] utilized the geographical draft of the spider map to upgrade the LBS experience. The spider map puts the contextual information related to users such as central location of the user, related stay points and end points on the geographical draft which could help users eliminate the mass of irrelevant information to improve the user experience. Through these significant researches, we figure out that it is important to develop an appropriate model and a more intuitive method for LBS based on the features of the POI data.

3 Modeling

According to the POI data we obtained from Weibo.com, we construct a geospatial model called Weigeo. This model consists of four components, POI node, POI zone, POI links and POI virtual distance. Following we will describe notation of each component.

3.1 POI Node

POI node is the basic data obtained from Weibo.com, each POI node contains 27 attributes, such as title, longitude, latitude, check-in number, check-in user number, category, address, etc. So our goal is to rearrange the distribution of these POI nodes for presenting maximum of the attributes.

3.2 POI Zone

POI zone consists of POI nodes. We define a POI node is a part of a POI zone if this POI node is in the distance of zoneRange (analogous to radius of a circle). Moreover, if a POI node belongs to other POI zone in the distance of zoneRange to current POI zone, we can merge these two POI zones into a larger POI zone. In Section 4, we will detailed introduce the Divide Zone algorithm.

3.3 POI Link

POI link is depicted by connecting line, which represent the transformational distance between each POI nodes or POI zones.

3.4 POI Virtual Distance

POI virtual distance is proposed to solve the uneven distributing problem of the POI nodes. Virtual distance represents the final transformational position of a POI node, as shown in Fig. 1. The process of Virtual Layout algorithm will be detailed described in Section 4.

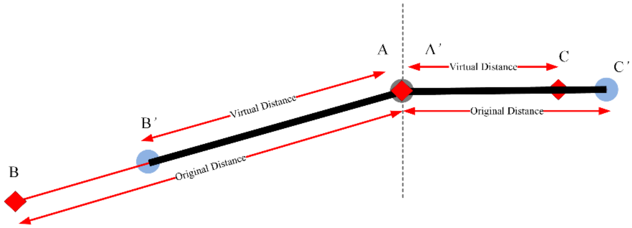

Fig. 1. Original Distance and Virtual Distance

4 Method

Currently, there're several main visualization layout algorithms, such as force-directed algorithm [7], multidimensional scaling algorithm [8] and relaxation algorithm [9], etc. Although these algorithms can ensure the information distributed evenly by moving the positions of points, but they have a serious deficiency that the original direction between point and point would be neglected, so these algorithms can not meet our requirements.

In this section, we proposed a novel graphic layout algorithm called Virtual Layout algorithm. The Virtual Layout algorithm mainly consists of three sub algorithms—Divide Zone algorithm, POI nodes virtual tree algorithm and POI zones virtual tree algorithm. These three sub algorithms cannot be separated from each other, the input of later sub algorithm depends on the output of previous sub algorithm. Following we will detailed describe these three sub algorithms.

4.1 Divide Zone Algorithm

Divide Zone is an algorithm which mainly applies DBSCAN [10] to cluster the POI nodes into a POI zone in accordance with the size of zoneRange (zoneRange is a parameter determining the size of clustering range). We cluster two POI nodes into one POI zone if their distance is less than the zoneRange, and if two POI zones are overlapped, we also cluster them together until all the nodes are divided into POI zones. In practical use, we use the ratio of the screen width as the zoneRange (e.g., a 1366×768 resolution screen, 0.01 zoneRange means 13.66 pixel units. And we transform pixel value to definitive GPS degree to cluster the POI nodes). The value of zoneRange can be customized by users for their convenient.

4.2 POI Nodes Virtual Tree Algorithm

In Section 3, we have presented the concept of POI virtual distance. POI virtual distance is the output of Virtual Layout algorithm, which is substitute of original distance. It can refine the distance of every two POI nodes, avoiding some nodes located too close or too loose. First of all, we will present some parameter of this algorithm:

$$meanLen = \frac{Total\ length\ of\ the\ virtual\ tree\ in\ a\ POI\ zone}{Number\ of\ the\ tree\ edges} \quad (1)$$

$$perLen = \frac{meanLen}{5} \quad (2)$$

$$maxLen = 2\ meanLen \quad (3)$$

$$minLen = perLen \quad (4)$$

MeanLen represents the average length of the virtual tree in a POI zone and different POI zone will come to different meanLen, so as the maxLen and minLen. This results a dynamic adaptation of different POI zones. The procedure of POI nodes virtual tree algorithm is described as follows:

1. If a POI node is in stable state, select another POI node which is closest to this node but not in the stable state, then connect these two nodes.
2. After obtaining the original distance of these two nodes, we can calculate the virtual distance according to Table 1.

Table 1. Output of Virtual Distance

Original Distance	**Virtual Distance**
0<Original Distance<minLen	minLen
minLen<Original Distance<maxLen	Original Distance/perLen
maxLen<Original Distance<zoneRange	maxLen

So we can see the schematic diagram shown in Fig. 2. The POI nodes virtual tree algorithm actually makes a centralized tendency of the edges of which distances are too short or too long. After this process, we can obtain an even distribution of nodes in each POI zone.

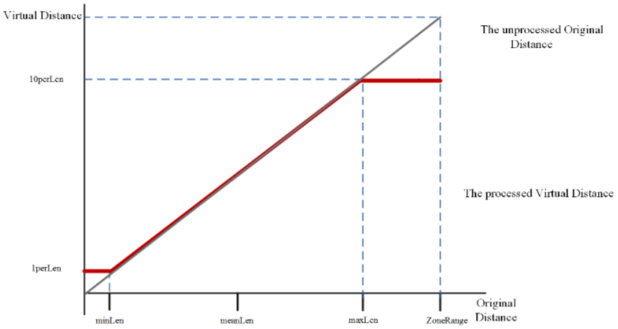

Fig. 2. Schematic diagram of POI nodes virtual tree algorithm

For example, as shown in Fig. 1, the three diamonds represent the original positions of three POI nodes (A, B and C), the three circulars represent the final positions (A’, B’ and C’). First, we can set A as a stable node (So A’ and A is the same), and then we choose node C which is closest to A, calculate their virtual distance. As edge <A’, C> is too short, we should stretch this edge to minLen as shown in Table 1. Finally, we can figure out the new position of C’. Similarly, edge <A’, B> is too long and we should shrink it to maxLen, and we can figure out the new position of B’.

4.3 POI Zones Virtual Tree Algorithm

Once all the POI nodes are all well rearranged in the POI zones, we can obtain the scope and central point of each POI zone. Then we adjust each POI zone to ensure the distance among all the POI zones is reasonable. The process is familiar with POI nodes virtual tree algorithm, finding the connection between every two POI zones is actually to find the minimum connection of two POI nodes respectively belong to different POI zones. In the process of moving the POI zones, all the POI nodes of this POI zone should also be moved.

As a whole, we can summarize the Virtual Layout algorithm into four states shown in Fig. 3. Figure (a) shows state of the original POI nodes. Figure (b) shows state of different POI zones clustered by Divide Zone algorithm. Figure (c) shows state of generating virtual tree among POI nodes in each POI zone and we can also see the status of POI zone such as scope and central point changing during this process. Figure (d) shows state of the final status of generating stable virtual tree among POI zones. It is obvious to see that the Virtual Layout algorithm is effective to make an even distribution and provide more displaying space for POI attribute information.

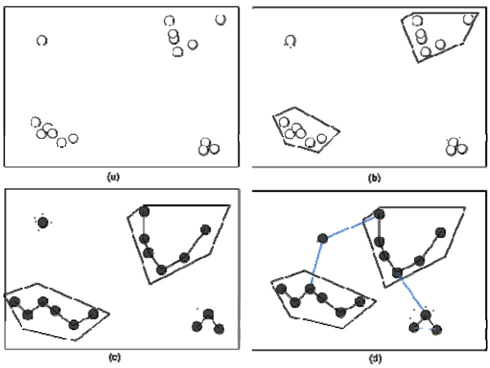

Fig. 3. Four States of the Virtual Layout Algorithm

4.4 The Features of VIRTUAL Layout Algorithm

Virtual Layout is a graph layout algorithm based on the graphic point distribution. It aims to distribute nodes on a plane more rational, and ensure the distance between each node is not too close nor too loose which lead to the result of losing some valid information or wasting some displaying space. Compare to other graphic layout algorithms, Virtual Layout algorithm has following features.

1. No iteration, only one time can reach equilibrium state

Force-directed algorithm, multidimensional scaling algorithm and relaxation algorithm are algorithms simulating physical models, so they need a series of iteration to reach the equilibrium state and obtain the location coordinates. Due to repeated iteration, if the number of points is large, it will be bound to cause a long time to reach equilibrium state. However, Virtual Layout algorithm can reduce time complexity and we do not need constant iteration to keep the equilibrium state. To sum up, firstly, we use a clustering algorithm (Divide Zone algorithm) to generate POI zones, of which time complexity is O(NlogN); second, we generate virtual trees for the POI nodes, of which time complexity is O(N*N*M) and M is the number of clustering number. Finally, we generate virtual trees among POI zones, of which time complexity is O(N*N). The final location of each node will be obtained by only one process, thus it is a more efficient algorithm compared to others.

2. Clear view of distribution

Virtual Layout algorithm adopted virtual distance to transform continuous distance to discrete length, which can do good effort to avoid visual illusion and provide a clear view of all the POI nodes.

3. Strong direction information

Although force-directed algorithm, multidimensional scaling algorithm and relaxation algorithm can provide an even distribution layout, but it is obvious that if we over emphasize the equilibrium of the point distribution, we will be likely to cause significant information loss (direction or distance deviation). So if these algorithms are applied on the map, they will cause serious visual illusion which may mislead the users. Accordingly, Virtual Layout algorithm makes a compromise of making distribution equilibrium and keeping information integrity, adjusting parts of the POI nodes and keeping most of the location information.

5 Microblog LBS Visualization System

In this section, we will develop a microblog LBS visualization system to illustrate the performance of the Weigeo model and Virtual Layout algorithm. Fig. 4 shows the framework of microblog LBS visualization system. The system can be divided into two modules, data collection module and visualization module. Data collection module takes charge of crawling data from Weibo.com and OpenStreetMap, processing the raw data and storing the processed data in database. These data includes POI information and stylized map data, which compose the foundation of the whole system. The second module is visualization module, including stylized map, Virtual Layout algorithm and location reference frame. In Section 4, we have detailed introduced the Virtual Layout algorithm, following we will introduce the stylized map and location reference frame.

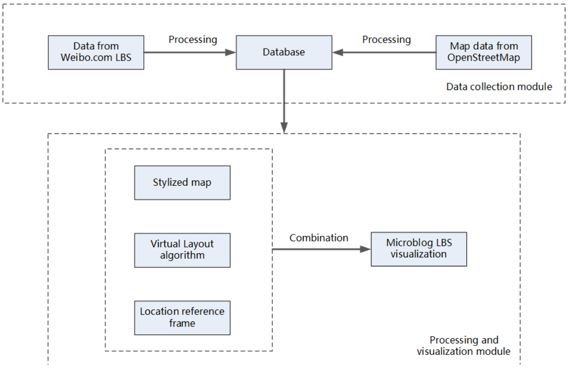

Fig. 4. Microblog LBS visualization system framework

5.1 Stylized Map

The general map contains various kinds of geographic information, in some cases, it will puzzle users to get a clear view of their concerning POI nodes. So in order to show the most valuable information on a limited map and eliminate the redundant factors, we apply stylized map. We utilize OpenStreetMap geographic data as our basic geographic data and use TileMill to edit the OpenStreetMap data. The OpenStreetMap data includes attributes of map such as background, labels, roads and rivers, etc. We use Carto Css to program the stylization files. Fig. 5 shows the final output of our stylized map, we can see that this customized map eliminates much of the redundant labels, but keeps the key geographic information.

Fig. 5. Output of stylized map

5.2 Location Reference Frame

Location reference frame is a measurement of the POI nodes of a region. Since a POI node contains up to 27 attributes, too much information would be difficult for displaying in limited displaying space, so we design a location reference frame for our visualization system. We will describe our POI node designed for microblog LBS visualization system as follows:

1. The size of the circular represents the check-in users of a POI node, more will be bigger.
2. The color of the outer ring of the circular represents the number of check-ins, hotter will be redder.
3. The category will be shown as icon below the POI node, along with the name of this POI node.
4. Timestamp will be displayed outer the POI node.

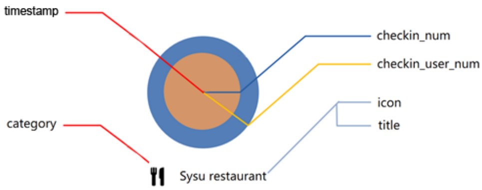

Fig. 6. Design of the location reference frame

6 Implementation

6.1 Data Preparation

Microblog LBS visualization system stored two kinds of data, one is geospatial data, and the other is stylized map data, both of which are obtained by a specific web crawler. Finally, we obtained 147 POI categories, 8437 POI nodes, 519,687 users and 1,007,702 check-ins of Guangzhou city for our illustration. In order to illustrate the performance of our visualization system, three locations (Higher Education Mega Center, Xinguang Highway and Taian Garden), 9 POI nodes of each location, 27 POI nodes in total are chosen as the dataset. The POI nodes (Take prior four POI nodes of Higher Education Mega Center as an example) are shown in Table 2.

6.2 Divide POI Zones

First we used Virtual Layout sub algorithm Divide Zone algorithm to cluster the POI nodes. We set zoneRange = 0.03, and we get 3 POI zones in the end.

Table 2. POI nodes of Higher Education Mega Center

Poiid	Title	Longitude, Latitude	Check-ins	Users	Category
B2094752D46A A2FE429E	Agricultural bank of China Panyu city branch	(23.047653, 113.378603)	1	1	Bank
B2094757D06A AAFA4399	GOGO new plaza	(23.062707, 113.391915)	6073	4412	Shopping Mall
B2094757D06F A6FE439A	Higher Education Mega Center North	(23.058074, 113.385632)	1811	1575	Subway Station
B2094757D06E AAFE429A	Sun Yat-sen University (East Campus)	(23.066774, 113.391935)	3821	2414	University
……	……	……	……	……	……

6.3 Generate Virtual Tree Among POI Nodes

By using POI nodes virtual tree algorithm presented in Section 4, we can acquire new position of each POI node. Take the POI nodes around Higher Education Mega Center as an example, we build a virtual tree in this POI zone, some POI links of the virtual tree are shown in Table 3.

Table 3. POI links around Higher Education Mega Center

POI link	From	To	Original distance (perLen)	Virtual distance (perLen)
(B2094752D46AA2FE429E, B2094757D06FA6FE439)	(23.048, 113.379)	(23.058, 113.386)	11.622881	10.0
(B2094757D06AAAFA4399, B2094654D468A0FB459)	(23.063, 113.392)	(23.063, 113.390)	2.0022447	2.0022447
(B2094757D06FA6FE439A, B2094757D065A4F8429)	(23.058, 113.386)	(23.056, 113.392)	5.697238	5.697238
(B2094757D06EAAFE429A, B2094654D468A0FB459)	(23.063, 113.390)	(23.067, 113.392)	3.6607623	3.6607623
……	……	……	……	

Once all the virtual trees are established, we can draw the virtual trees on the map. As Fig. 7 shows, Figure (a) is the result of virtual tree, Figure (b) is the comparison of original POI nodes and processed POI nodes. The virtual distance is set the range from minLen = 1perLen to maxLen = 10perLen. According to the transformation shown in Table 1, most of the links are within the limits of the range except for the first link, which is too long and need to be shrunk. On the whole, the POI nodes are mostly parallel to the original nodes. Although the algorithm makes some excursion and shrink, it keeps the relative direction between each node.

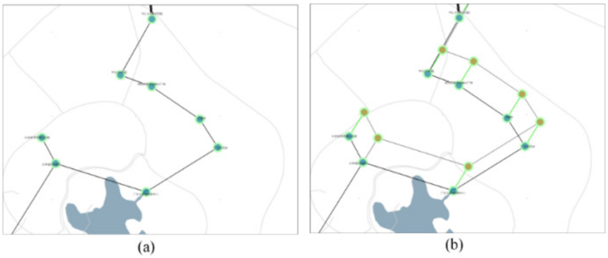

Fig. 7. Generate virtual tree among POI nodes

6.4 Generate Virtual Tree Among POI Zones

To find out the connection of POI zones is actually to find the shortest distance of POI nodes belong to different POI zones. As POI nodes have been divided into three POI zones, so we just need two POI links to connect the three POI zones as Table 4 shown.

Table 4. POI links of three POI zones

POI link	From	To	Original distance (perLen)	Virtual distance (perLen)
(B2094752D46AA2FE429E, B2094654D46FA0FD4998)	(23.048, 113.379)	(23.065, 113.345)	1.2626879	0.8
(B2094757D06EAAFE429A, B2094654D46CA6F9469B)	(23.063, 113.392)	(23.122, 113.386)	1.8961813	0.8

In Fig. 8, we can see that the new positions of POI nodes which keep the original directions but offer a more even distribution. And by using stylized map, we can reduce useless geographic redundancy and provide a more aesthetic visualization. We can infer that Virtual Layout algorithm is able to make an improvement for the LBS visualization and it can become an extra pattern of map information presentation.

Fig. 8. Generate virtual tree among POI zones

7 Conclusion

With the development of location technology, the combination of LBS and social network is enriching our lives. Microblog LBS visualization system, presented by this paper, is not only a kind of visualization, but also an important bridge connecting the virtual world and real world, offering a more convenient way for people to explore places they are interested in. Weibo.com as a primary social network service provider in China, it has a variety of geospatial and social information. How to effectively use these information to construct a better model to describe our world and detect valuable pattern is a promising direction for the future researching. In the end, we hope that in the future, we can improve the performance of Virtual Layout algorithm for three-dimensional visualization and apply it in other domains. And for our microblog LBS visualization system, we want to find a better interaction method to present more POI information in the limited displaying space.

Acknowledgements. This work was supported by the National Natural Science Foundation of China under Grant No.61073132 and 60776796; the Fundamental Research Funds for the Central Universities (101gpy33); Project 985 of Innovation Base for Journalism & Communication in the All-media Era, Sun Yat-sen University.

References

1. Chen, X., et al.: System architecture of LBS based on spatial information integration. In: Geoscience and Remote Sensing Symposium. IEEE (2004)
2. Contributors, W.: Point of interest, in Wikipedia, The Free Encyclopedia (2013)
3. Iliinsky, N.: On beauty. Beautiful Visualization: Looking at Data through the Eyes of Experts, 1–14 (2010)
4. Chen, X., Chen, Y., Rao, F.: An efficient spatial publish/subscribe system for intelligent location-based services. In: Proceedings of the 2nd International Workshop on Distributed Event-Based Systems 2003, pp. 1–6. ACM, San Diego (2003)
5. Jensen, C.S., et al.: Multidimensional data modeling for location-based services. In: Proceedings of the 10th ACM International Symposium on Advances in Geographic Information Systems 2002, pp. 55–61. ACM, McLean (2002)
6. Mourinho, J., Galvao, T., Falcão e Cunha, J.: Spider Maps for Location-Based Services Improvement. In: Snene, M., Ralyté, J., Morin, J.-H. (eds.) IESS 2011. LNBIP, vol. 82, pp. 16–29. Springer, Heidelberg (2011)
7. Chernobelskiy, R., Cunningham, K.I., Goodrich, M.T., Kobourov, S.G., Trott, L.: Force-Directed Lombardi-Style Graph Drawing. In: Speckmann, B., et al. (eds.) GD 2011. LNCS, vol. 7034, pp. 320–331. Springer, Heidelberg (2011)
8. Davison, M.L.: Multidimensional scaling. Wiley, New York (1983)
9. Fry, B.: Visualizing data: exploring and explaining data with the Processing environment. O'Reilly Media (2008)
10. Ester, M., et al.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: KDD (1996)