A Spatial Keyword Evaluation Framework for Network-based Spatial Queries

Ji Zhang Wei-Shinn Ku Xunfei Jiang Xiao Qin
Dept. of Computer Science and Software Engineering, Auburn University, Auburn, AL 36849, USA
{jizhang,weishinn,xzj0009,xqin}@auburn.edu

ABSTRACT

An increasing number of spatial keyword query techniques that return qualified objects based on a comprehensive consideration of spatial and keyword constraints have been presented in the literature. Due to the complexity of the solutions to spatial keyword queries, systems that can effectively demonstrate the mechanisms will attract interest from the spatial database research community. However, very limited visualization systems have been developed for illustrating spatial keyword query evaluation. In this demonstration, we present a system that visualizes advanced solutions to efficiently answer the Spatial Keyword k Nearest Neighbor (SKkNN) query. With the two-level data management method and the friendly user interface implemented by the Standard Widget Toolkit (SWT) and Open Graphics Library (OpenGL), our system is able to not only interact with users in diverse manners, visualize datasets, and display the SKkNN query evaluation process, but it also helps users better understand the solutions in a more intuitive way.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Application—*spatial databases and GIS*

General Terms

Algorithms, Experimentation

Keywords

Spatial keyword query, Location-based services

1. INTRODUCTION

An increasing number of Spatial Keyword (SK) query techniques have been proposed in the literature [2, 3, 5, 6]. Spatial keyword queries take spatial and keyword requirements into account and return objects that satisfy all the conditions of the search. However, most existing solutions for SK queries are restricted to the Euclidean space, which is not realistic since most users move on spatial networks in real life. Moreover, only the objects that fully match the given keywords can be found by current approaches. Nevertheless, the retrieved objects could be *far away* from the query location. Due to the complexity of the solutions to SK queries, systems that can effectively demonstrate the mechanisms will attract interest from the spatial database research community.

In this demonstration, we present an advanced framework, which is able to visualize novel Spatial Keyword k Nearest Neighbor (SKkNN) solutions that can not only retrieve qualified objects on road networks, but also obtain both fully and partially keyword-matched objects [7, 8]. Under a two-level data management method, the data in visualization area is maintained independently from the input datasets, which significantly simplifies data retrieval and improves the performance of query evaluation. Because a friendly graphical interface is provided, users are able to easily interact with our system in diverse manners.

2. SYSTEM ARCHITECTURE

The framework of our demonstration system is illustrated in Figure 1. The data management component manages the configuration dataset input by users, such as the size and location of a visualization area. The evaluation component deals with SK query evaluation processing and produces query results based on query inputs. The visualization component interacts with users, including user input retrieval and result demonstration. In addition, two query

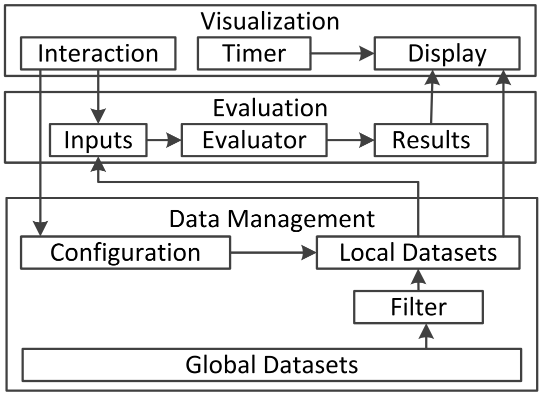


Figure 1: Framework of our demonstration system.

SIGSPATIAL'13, Nov 05-08 2013, Orlando, FL, USA ACM 978-1-4503-2521-9/13/11. http://dx.doi.org/10.1145/2525314.2525315

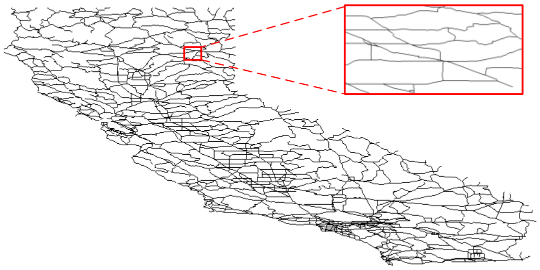


Figure 2: An example of the two-level data management method.

evaluation demonstration methods are supported in our system. The search process is proceeded step by step, which is triggered by either mouse clicks or events controlled by the timer component.

2.1 Data Management

Due to high density of spatial data (i.e., road networks and points of interest (POIs)), and limited visualization space on screen, a two-level data management method is designed for better demonstrations. Rather than retrieving all the spatial data from the global datasets, we create a logically limited area for visualization. All data located in the visualization area is cached in the local datasets. The size and location of the visualization area can be customized through the user interface and recorded in the configuration module. In addition, POI datasets may be maintained independently from road network datasets. The filter module is used to efficiently retrieve visualization subsets from both datasets. Figure 2 displays a concrete example of road networks in the state of California[1]. The roads in the visualization area highlighted in a red box are enlarged and displayed in a separate page of our demonstration system.

2.2 Query Evaluation

Since both keyword and spatial constraints are considered in SK queries, the query evaluation component receives inputs from the user interface and POIs and road network data from the local datasets. The evaluator evaluates queries by using corresponding query solutions. The results are produced for visualization. Figure 3 illustrates a general design that applies a filter-and-refine strategy to answer SK queries. Keyword constraint filter calculates the keyword relevancy of each POI based on the given keywords. Then, keyword and spatial refine comprehensively considers both keyword and spatial constraints. All POI candidates are ordered by their ranking scores calculated by the ranker. The method of ranking score calculation can be customized by users. In addition to the query solution, an internal state machine is created for demonstration purpose. The entire evaluation process is divided into steps, each of which is tagged as a state. Instead of internally producing results based on inputs, the evaluator conducts the process step by step, which allows users to better understand the two novel SKkNN solutions, Network Expansion-based SKkNN (NEkNN) and Voronoi Diagram-based SKkNN (VDkNN) [7, 8].

2.3 Visualization

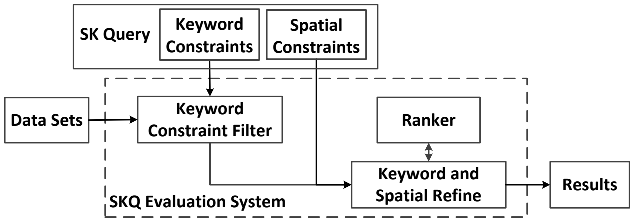


Figure 3: The evaluation module.

The visualization component mainly works on user interaction, which receives user inputs and displays query results. Through the graphical user interface, users can not only set the configuration of the demonstration system and query inputs, but also interact with the query evaluation framework, such as proceeding forward or backward in the query evaluation process.

3. SKkNN QUERY EVALUATION

Given a query point, a set of keywords, and a positive integer k, SKkNN query returns the best k POIs ordered by their ranking scores. Based on the query type definition of SKkNN, we develop two novel solutions that utilize the Dijkstra's shortest path algorithm and network Voronoi diagram (NVD) techniques to efficiently answer the SKkNN query.

3.1 Network Expansion-Based Algorithm

Dijkstra's algorithm-based approaches have been widely applied for searching the shortest path between objects on spatial networks in various applications [1]. The algorithm recursively expands the unvisited paths from a given query point to objects. During the search, the shortest distance of each object from the starting point is recorded. The search is continued until all the objects have arrived and no possible shorter path exists in the network. Based on Dijkstra's algorithm, Incremental Network Expansion (INE) is proposed to answer k nearest neighbor queries on spatial networks [4]. In particular, INE starts with the road segment e_i, which covers the query point q and retrieves all objects on e_i. If any object is found on e_i, the object will be inserted into the result set. Furthermore, the endpoint of e_i, which is closer to q, will be expanded while the second endpoint of e_i will be placed in a priority queue Q_p. INE repeats the process by iteratively expanding the first node in Q_p and inserting newly discovered nodes into Q_p until k objects are retrieved.

We develop a Network Expansion-based SKkNN (NEkNN) algorithm based on INE. NEkNN first filters out objects that do not match any keywords. Those unqualified objects are not visible to the next step. Then, we search candidate objects by expanding the network from the query point and calculating the ranking score of each visited object. Meanwhile, NEkNN keeps a result set, in which candidate objects are sorted in descending order based on their ranking scores. The expansion continues until the result set contains k objects and the stop condition is met.

3.2 Voronoi Diagram-Based Algorithm

To make the network expansion process more efficient, network Voronoi diagram technique is applied in our second solution and is named Voronoi Diagram-based k Nearest Neighbor (VDkNN). In order to be independent of the density and distribution of candidate objects, a network Voronoi diagram is first created based on the road network and POI

---

[1] http://www.census.gov/geo/www/tiger/

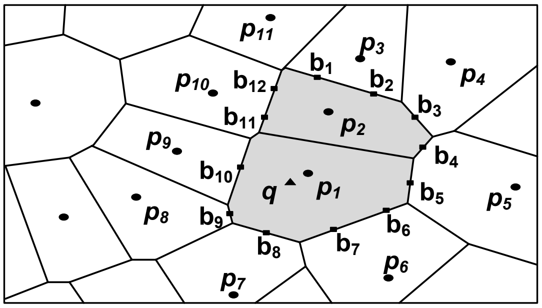


Figure 4: A VDkNN query example.

locations. In addition, for each NVD cell, we pre-compute the distances between all the border nodes of the cell to its center as well as the node-to-node distances of all the border nodes. Consequently, for a new cell, we can quickly extend the searched region to the border nodes without expanding all the internal network segments.

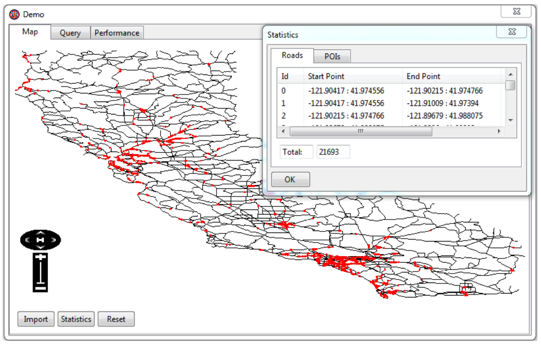
VDkNN first calculates the keyword relevancy of each POI. The ones that do not match any keyword will not be considered in the following process. Then, the search starts from the Voronoi cell VC_i that the query point resides. The ranking score of the POI associated with VC_i is calculated and added to the candidate list. After VC_i is visited, the search expands to the neighboring cells of VC_i. The process continues until there is no better candidate existing in the unvisited NVD cells. Differing from the traditional shortest path-based approaches that are sensitive to the density of road networks, Voronoi diagram-based approaches can avoid unnecessary data retrieval by jumping from one Voronoi cell to another.

Figure 4 illustrates an example of the solution, in which the network Voronoi diagram has been generated and the query point locates in the Voronoi cell of p_1. After keyword relevance calculation, the solution starts the search from p_1. If p_1 is relevant to the given keywords, its ranking score is calculated. Otherwise, the cell is skipped. After p_1 is visited, the search is expanded to p_2, one of its unvisited neighboring cells. The stopping condition of the search is checked before each expansion. We calculate the ranking scores of the border nodes which are on the boundary of the visited area (the shaded region in Figure 4). We assume that the border nodes match all the given keywords. If the border node with the highest ranking score is not better than the k^{th} POI in the candidate list, the POIs in the unvisited area cannot have a higher ranking score than objects in the candidate list. Then, the expansion terminates.

4. DEMONSTRATION

We implement NEkNN and VDkNN solutions in our demonstration system. The graphical framework and user interface are created by using the Standard Widget Toolkit (SWT) and OpenGL library for visualization. Global datasets, query evaluation processing, and system performance are visualized in three tab pages, respectively. We highlight VDkNN solution in this demonstration paper due to space limit.

4.1 Global Dataset Display



Id	Start Point	End Point
0	-121.90417 : 41.974556	-121.90215 : 41.974766
1	-121.90417 : 41.974556	-121.91009 : 41.97394
2	-121.90215 : 41.974766	-121.89679 : 41.988075

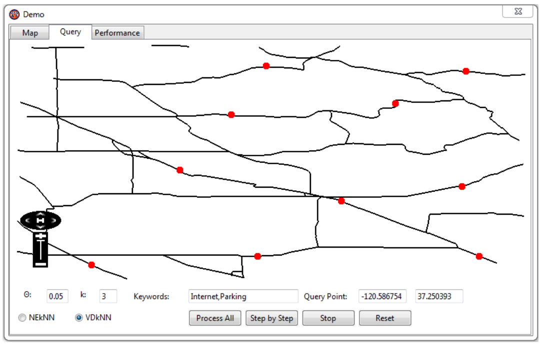
Total: 21693

Figure 5: Global dataset.

Figure 5 shows an example of California road network and restaurants[2]. The restaurants are plotted by red points. From the figure, we observe that restaurants are not uniformly distributed over the entire road network. Instead, most of them are clustered in a couple of metropolitan areas, such as the Greater Los Angeles area. The density of roads and restaurants are high. Demonstrating all of them in a limited screen is not user-friendly. Therefore, a zooming user interface is provided for global data visualization. In addition, their detailed information, such as coordinates, keywords associated with restaurants, and the statistical data, can be found in a separate window, which is shown on the top right corner of Figure 5. Both keyboard and mouse inputs are accepted by our system. For example, the map can be zoom-in/zoom-out by either scrolling up/down with a mouse, or clicking "+" and "-" symbols in the navigation bar on the bottom left corner of the map. Users can also change the visualization area by either dragging the map to all directions, or clicking navigation arrows. "Import", "Statistics", and "Reset" buttons lead users to load global datasets, display the statistics of the datasets, and clear the datasets.

4.2 Query Evaluation Display

The evaluation process that employs SKkNN algorithms is displayed in the query tab, as Figure 6 shows. Our system supports both an automatic and a manual method for query evaluation process demonstration. Initially, the system switches to a local network with POIs (e.g., restaurants) and waits for query inputs from users. The parameter θ is used as a weight in ranking score calculation. The value k indicates the number of results that will be returned by our system. The following two parameters are the keywords

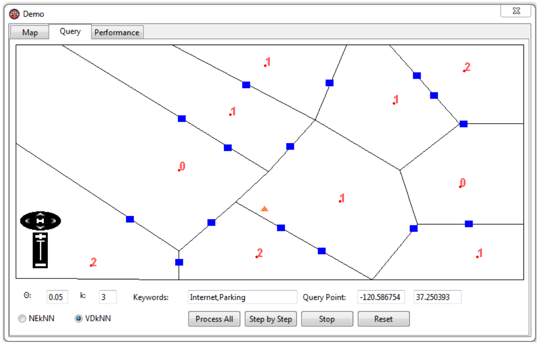


Θ: 0.05
k: 3
Keywords: Internet,Parking
Query Point: -120.586754 37.250393
○ NEkNN
● VDkNN

Figure 6: The datasets and user inputs of a SKkNN query.

---

[2] http://www.edigitalz.com/



Θ: 0.05 k: 3 Keywords: Internet,Parking Query Point: -120.586754 37.250393

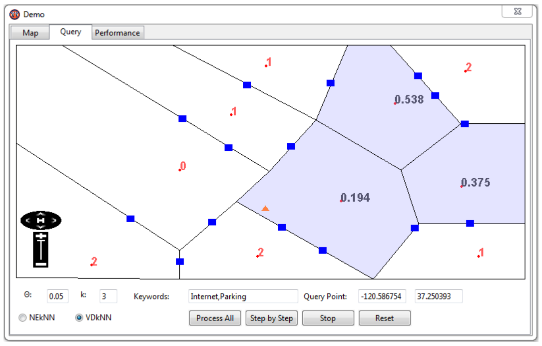
○ NEkNN ● VDkNN

Figure 7: Network Voronoi diagram generation.

and the query location of the SKkNN query. The query location can be input by either selecting a location on the map or filling in the corresponding coordinates in the text boxes. If the "Process All" button is clicked, the query evaluation starts in an automatic manner. The status of the query evaluation process will be updated and displayed in a particular interval (2 seconds by default). If "Step by Step" button is clicked, the query evaluation process will be controlled by users. "Stop" and "Reset" buttons stop and reset the process, respectively.

During the VDkNN query evaluation process, our system first generates a network Voronoi diagram as well as border nodes of each cell as shown in Figure 7. The border nodes are highlighted by blue boxes in the map. The search can expand from one Voronoi cell to another through the border nodes. We do not display connections or distance between restaurants and their border nodes due to the limited visualization space. In addition, the keyword relevancy is calculated. For simplicity, we use the number of keywords matched with given keywords as the keyword relevancy in ranking score calculation. The numbers are displayed next to each POI and can help users understand how the two parameters, keyword relevancy and weighted distance, work in the result selection.

Then, our system starts searching from the Voronoi cell where the query point locates. The query point is indicated by a small orange triangle in the map. We calculate the ranking score of each restaurant that we have visited so far. For example, three Voronoi cells have been visited in Figure 8. Their keyword relevancy have been replaced with their ranking scores shown by black floating point numbers. The higher the ranking score is, the better the restaurant is. After adding the three restaurants into the candidate list, the search will expand to the next neighboring Voronoi cell.



Θ: 0.05 k: 3 Keywords: Internet,Parking Query Point: -120.586754 37.250393

○ NEkNN ● VDkNN

Figure 8: VDkNN query evaluation.

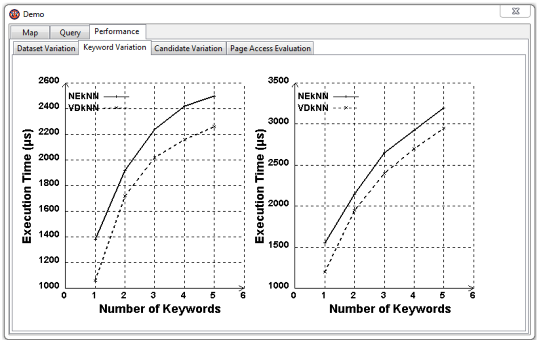


Figure 9: System performance of NEkNN and VDkNN queries.

4.3 Performance

To quantitatively demonstrate the performance of NEkNN and VDkNN. The numerical results are integrated and presented in the "Performance" tab as shown in Figure 9. Four groups of performance evaluation metrics are supported; users can vary/investigate: the size of input datasets, the number of keywords, the number of candidates produced by the solutions, and the number of page access.

5. CONCLUSION

In this paper, we present a spatial keyword query evaluation framework and implement a SKkNN evaluation system that can efficiently return both fully and partially matched POIs by expanding the search from the query location to all directions. With the help of the two-level data management, our system focuses on a small portion of data for visualization, which reduces the difficulty in displaying large datasets on a limited screen. In addition, both keyboard and mouse inputs are allowed by the friendly user interface.

6. REFERENCES

[1] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959.

[2] I. D. Felipe, V. Hristidis, and N. Rishe. Keyword Search on Spatial Databases. In *ICDE*, pages 656–665, 2008.

[3] R. Hariharan, B. Hore, C. Li, and S. Mehrotra. Processing Spatial-Keyword (SK) Queries in Geographic Information Retrieval (GIR) Systems. In *SSDBM*, page 16, 2007.

[4] D. Papadias, J. Zhang, N. Mamoulis, and Y. Tao. Query Processing in Spatial Network Databases. In *VLDB*, pages 802–813, 2003.

[5] J. B. Rocha-Junior and K. Nørvåg. Top-k spatial keyword queries on road networks. In *EDBT*, pages 168–179, 2012.

[6] D. Zhang, Y. M. Chee, A. Mondal, A. K. H. Tung, and M. Kitsuregawa. Keyword Search in Spatial Databases: Towards Searching by Document. In *ICDE*, pages 688–699, 2009.

[7] J. Zhang, W.-S. Ku, X. Jiang, X. Qin, and Y.-L. Hsueh. Evaluation of Spatial Keyword Queries with Partial Result Support on Spatial Networks. In *In Proceedings of the 14th IEEE International Conference on Mobile Data Management (MDM)*, Milan, Italy, 2013.

[8] J. Zhang, W.-S. Ku, and X. Qin. Evaluation of Spatial Keyword Queries with Partial Result Support on Spatial Networks. Technical Report CSSE13-01, Auburn University, 2013. `http://www.eng.auburn.edu/files/acad_depts/csse/csse_technical_reports/csse13-01.pdf`.