International Journal of Science and Engineering Research (IJ0SER),

Vol 6 Issue 4 April -2018
3221 5687, (P) 3221 568X

Efficient Clue-Based Route Search On Road Networks

[1] Harini K, [2] Indhumathi M , [3] Dr A.Grace Selvarani,
Computer Science and Engineering.[3] Head of the Department,
Tamilnadu,India

Abstract:With the advances in geo-positioning technologies and location-based services, it is nowadays quite common for road networks to have textual contents on the vertices. Previous work on identifying an optimal route that covers a sequence of query keywords has been studied in recent years. However, in many practical scenarios, an optimal route might not always be desirable. For example, a personalized route query is issued by providing some clues that describe the spatial context between PoIs along the route, where the result can be far from the optimal one. Therefore, in this paper, we investigate the problem of clue-based route search (CRS), which allows a user to provide clues on keywords and spatial relationships. First, we propose a greedy algorithm and a dynamic programming algorithm as baselines. To improve efficiency, we develop a branch-and-bound algorithm that prunes unnecessary vertices in query processing. In order to quickly locate candidate, we propose an AB-tree that stores both the distance and keyword information in tree structure. To further reduce the index size, we construct a PB-tree by utilizing the virtue of 2-hop label index to pinpoint the candidate. Extensive experiments are conducted and verify the superiority of our algorithms and index structures.

Objective:
To improve the efficiency of the route search through the optimal route using efficient clue based on query processing.

INTRODUCTION

With the rapid development of location-based services and geo-positioning technologies, there is a clear trend that an increasing amount of geo-textual objects are available in many applications. For example, the location information as well as concise textual descriptions of some businesses (e.g., restaurants, hotels) can be easily found in online local search services (e.g., yellow pages). To provide better user experience, various keywords related spatial query models and techniques have emerged such that the geo-textual objects can be efficiently retrieved. It is common to search a Point-of-Interest (PoI) by providing exact address or distinguishable keyword (i.e., only few PoIs contain the keyword) in a region which can uniquely pinpoint the location. For example, we type the address "73 Mary St, Brisbane" or the name "Kadoya" on Google Maps to find a Japanese restaurant in the CBD area. Some existing work extends such query to more sophisticated settings, such as retrieving a group of geo-textual objects (usually more than 2) or a trajectory covering multiple keywords. However, it is not uncommon that a user aims to find a PoI with less distinguishable keyword such as "restaurant", but she can only provide more or less spatio- textual context information around the PoI. Liu et al. formalize such context information as clues and use them to identify the most promising PoIs.

Different with their work, we aim to find a feasible route on road networks by using clues. Particularly, in this paper, we investigate a novel query type, namely clue-based route search (CRS), which allows a user to provide clues on textual and spatial context along the route such that a best matching route w.r.t. the clues is returned. More specifically, a CRS query is defined over a road network G, and the input of the query consists of a source vertex vq and a sequence of clues, where each clue contains a query keyword and a user expected network distance. A vertex contains a clue keyword is considered as a match vertex. The query returns a path P in G starting at vq, such that (i.) P passes through a sequence of match vertices (PoIs) w.r.t. the clues and (ii.) the network distances between two contagious matched vertices are close to the corresponding user specified distance such that the user's search intention is satisfied.

Modeling Imprecise User Intention. The clue is typically based on observations that 1) the keywords of PoIs in the clue may be interchangeable or inexact terms (e.g., a user may think of a PoI being canteen whereas it may be referred to as a restaurant); 2) the spatial relationships between PoIs are approximate, which is a natural phenomenon for human to estimate distance. For example, if the distance between two PoIs in the clue is about 100 meters, the actual distance may be noticeably greater than or less than 100 meters. Consider a scenario in our daily life: a user wants to find a restaurant in a city visited many years ago. She cannot remember the exact name and address but she still recalls that on the way driving to the restaurant from her home, she passed a cafe at about 1 km away, and

drove about another 2 km to reach the restaurant. The information given above usually cannot precisely locate PoI, but intuitively it provides clues to identify the most likely PoIs along the route.

Increased Flexibility in Trip Planning. As mentioned before, most existing work aims to find an optimal route with minimum travel distance. However, in many real scenarios, such an optimal route might not always be desirable. For instance, a user may have some personalized requirements on the distances between PoIs when planning a trip. Consider such a scenario, a user wants to find a buffet restaurant and a nearby cinema only in walking distance, say 3km, thus he can watch a movie after dinner. Therefore, after having delicious food, he can walk to the cinema in order to maintain a healthy lifestyle. These personalized requirements make the route search become distance-sensitive and more flexible such that the distance between PoIs along the route should be as close as possible to the user specified distance.

Clue-Based Route Navigation. Given a description including textual and distance information on an expected route, it is still not direct-viewing enough for users to obtain the exact route. This is usually the case when a user wants to know the way for a specific place and asks the others for help, she may still not be able to exactly figure out the route after obtaining the answers from them, where the answer usually comes in the form, for example, "go straight on the way for about 100 meters, you will see a cafe, and turn right, you will arrive the Japanese restaurant after about 150 meters walk". Therefore, a novel type of route search which automatically interprets the clues contained in such answers becomes necessary. By augmenting it on current navigation services, a better user experience can be provided.

Existing system:

Geo-positioning technologies and location-based services, it is nowadays quite common for road networks to have textual contents on the vertices. On identifying an **optimal route** that covers a sequence of query keywords has been studied in recent years. However, in many practical scenarios, an optimal route might not always be desirable. A personalized route query is issued by providing some clues that describe the spatial context between PoIs along the route, where the result can be far from the optimal one. We investigate the problem of **clue-based route search** (CRS), which allows a user to provide clues on keywords and spatial relationships. **Clue-based route search** (CRS), which allows a user to provide clues on textual and spatial context along the route such that a best matching route the clues is returned. To provide better user experience, various keywords related spatial query models and techniques have emerged such that the geo-textual objects can be efficiently retrieved.

- An optimal route might not always be desirable.
- less distinguishable keyword
- It is still not direct-viewing enough for users to obtain the exact route.
- It requires quadratic time and is not scalable especially for more frequent keywords.
- Short response time, the accuracy of the answer cannot be guaranteed.

Proposed system:

- We propose a **greedy algorithm** and a **dynamic programming algorithm** as baselines.
- To improve efficiency, we develop a **branch-and-bound algorithm** that prunes unnecessary vertices in query processing. In order to quickly locate candidate, we propose an **AB-tree** that stores both the distance and keyword information in tree structure.
- To further reduce the index size, we construct a **PB-tree** by utilizing the virtue of 2-hop label index to pinpoint the candidate.
- PB-tree to further improve the performance.
- A greedy algorithm as a baseline for answering the CRS query, which is called Greedy Clue Search algorithm.
- A **branch-and-bound algorithm** (BAB) by applying filter-and-refine paradigm such that only a small portion of vertices are visited, hence improves the search efficiency.
- A **greedy algorithm** to find a route whose length is smaller than a specified threshold while the total text relevance of this route is maximized.
- It is more efficient if we adopt CDP with all-pair, and for high frequency keywords, BAB with PB-tree has a much better performance on both response time and index size.

The enhancement techniques outperform the IR2-tree in query response time significantly. an effective method for keyword- based document retrieval. By doing so they develop a structure called the IR2-tree , which has the strengths of both R-trees and signature files. Like R-trees, the IR2- tree preserves objects' spatial proximity, which is the key to solving spatial queries efficiently.

- Index to improve the search efficiency.
- To improve efficiency.
- Improve the performance in route search.
- To further reduce the index size.
- To support the index updating.

- Runs much faster, and the index size of PB-tree is much smaller than AB-tree.

FAST SHORTEST-PATH DISTANCE QUERIES ON ROAD NETWORKS

We propose a new labeling method for shortest-path and distance queries on road networks. We present a new framework (i.e. data structure and query algorithm) referred to as highway-based labeling and a preprocessing algorithm for it named pruned highway labeling. Our proposed method has several appealing features from different aspects in the literature. Indeed, we take advantages of theoretical analysis of the seminal result by Thorup for distance oracles, more detailed structures of real road networks, and the pruned labeling algorithm that conducts pruned Dijkstra's algorithm. The experimental results show that the proposed method is comparable to the previous state-of-the-art labeling method in both query time and in data size, while our main improvement is that the preprocessing time is much faster.

Computing shortest path or distance between two points is one of the most fundamental and important problems on road networks with many applications. In most of cases, this computation problem can be described in the following algorithmic problem: Given a (almost) planar graph, we preprocess it subject to the restriction that the space for the preprocessing results is limited (e.g., a small constant times the space used to store the graph). Then we would like to quickly answer queries on single-pair (or multipair) shortest paths. This is a natural variant of the problem as in many applications, such as that of computing driving directions (e.g., Google Maps, Bing Maps, Yahoo! Maps). Therefore theoretical and practical approaches involving precomputation on input graphs have been particularly a central topic in the literature for more than two decades and continuing. One of the most notable recent developments is the emergence of practical labeling methods. Labeling methods precompute label L(v) for each vertex v and answer distance between two points s and t using labels L(s) and L(t) without looking at the input graph. These practical methods adopt the most standard labeling framework (i.e., data structure and query algorithm) called hub-based labelings . The label L(v) is a set of pairs (u; uv), where u is a vertex and uv is distance between v and u. Since the labeling method can answer distance by only looking at two consecutive segments of precomputed data, their query time is known to be the fastest among all other proposed methods.

In this paper, we proposed a new labeling framework referred to as the "highway-based labeling framework". Highway-based labeling exploits highways and decomposes an input graph into shortest paths. Then, it computes labels that contain the distance to the shortest paths. In addition, we proposed a preprocessing algorithm for the framework named "pruned highway labeling". This algorithm is based on the pruned landmark labeling and can compute labels efficiently. We only conduct a pruned Dijkstra search from each shortest path. Our experimental results show that our proposed method is much faster than previous (and state-of-theart) labeling methods in the preprocessing time. Moreover, the query time is sufficiently fast as it is also a labeling method.

COLLECTIVE SPATIAL KEYWORD QUERYING

With the proliferation of geo-positioning and geo-tagging, spatial web objects that possess both a geographical location and a textual description are gaining in prevalence, and spatial keyword queries that exploit both location and textual description are gaining in prominence. However, the queries studied so far generally focus on finding individual objects that each satisfy a query rather than finding groups of objects where the objects in a group collectively satisfy a query. We define the problem of retrieving a group of spatial web objects such that the group's keywords cover the query's keywords and such that objects are nearest to the query location and have the lowest inter-object distances. Specifically, we study two variants of this problem, both of which are NP-complete. We devise exact solutions as well as approximate solutions with provable approximation bounds to the problems. We present empirical studies that offer insight into the efficiency and accuracy of the solutions.

With the proliferation of geo-positioning, e.g., by means of GPS or systems that exploit the wireless communication infrastructure, accurate user location is increasingly available. Similarly, increasing numbers of objects are available on the web that has an associated geographical location and textual description. Such spatial web objects include stores, tourist attractions, restaurants, hotels, and businesses. This development gives prominence to spatial keyword queries. A typical such query takes a location and a set of keywords as arguments and returns the single spatial web object that best matches these arguments. We observe that user needs may exist that are not easily satisfied by a single object, but where groups of objects may combine to meet the user needs. Put differently, the objects in a group collectively meet the user needs. For example, a tourist may have particular shopping, dining, and accommodation needs that may best be met by several spatial web objects. As another example, a user may wish to set up a project consortium of partners within a certain spatial proximity that combine to offer the

capabilities required for the successful execution of the project. To address the need for such collective answers to spatial keyword queries, we assume a database of spatial web objects and then consider the problem of how to retrieve a group of spatial objects that collectively meet the user's needs given as location and a set of keywords: 1) the textual description of the group of objects must cover the query keywords, 2) the objects are close to the query point, and 3) the objects in the group are close to each other.

We present the new problem of retrieving a group of spatial objects, each associated with a set of keywords, such that the group covers the query's keywords and has the lowest cost measured by their distance to the query point, and the distances between the objects in the group. We study two particular instances of the problem, both of which are NP-complete. We develop approximation algorithms with provable approximation bounds and exact algorithms to solve the two problems. Results of experimental evaluation offer insight into the efficiency and the accuracy of the approximation algorithms, and the efficiency of the exact algorithms.

FAST EXACT SHORTEST-PATH DISTANCE QUERIES ON LARGE NETWORKS BY PRUNED LANDMARK LABELING

We propose a new exact method for shortest-path distance queries on large-scale networks. Our method precomputes distance labels for vertices by performing a breadth-first search from every vertex. Seemingly too obvious and too inefficient at first glance, the key ingredient introduced here is pruning during breadth-first searches. While we can still answer the correct distance for any pair of vertices from the labels, it surprisingly reduces the search space and sizes of labels. Moreover, we show that we can perform 32 or 64 breadth-first searches simultaneously exploiting bitwise operations. We experimentally demonstrate that the combination of these two techniques is efficient and robust on various kinds of large-scale real-world networks. In particular, our method can handle social networks and web graphs with hundreds of millions of edges, which are two orders of magnitude larger than the limits of previous exact methods, with comparable query time to those of previous methods.

A distance query asks the distance between two vertices in a graph. Without doubt, answering distance queries is one of the most fundamental operations on graphs, and it has wide range of applications. For example, on social networks, distance between two users is considered to indicate the closeness, and used in socially-sensitive search to help users to find more related users or contents, or to analyze influential people and communities. On web graphs, distance between web pages is one of indicators of relevance, and used in context-aware search to give higher ranks to web pages more related to the currently visiting web page. Other applications of distance queries include top-k keyword queries on linked data, discovery of optimal pathways between compounds in metabolic networks, and management of resources in computer networks. Of course, we can compute the distance for each query by using a breadth first search (BFS) or Dijkstra's algorithm. However, they take more than a second for large graphs, which is too slow to use as a building block of these applications. In particular, applications such as socially-sensitive search or context-aware search should have low latency since they involve real-time interactions between users, while they need distances between a number of pairs of vertices to rank items for each search query. Therefore, distance queries should be answered much more quickly, say, microseconds. The other extreme approach is to compute distances between all pairs of vertices beforehand and store them in an index. Though we can answer distance queries instantly, this approach is also unacceptable since preprocessing time and index size are quadratic and unrealistically large. Due to the emergence of huge graph data, design of more moderate and practical methods between these two extreme approaches has been attracting strong interest in the database.

In this paper, we proposed a novel and efficient method for exact shortest-path distance queries on large graphs. Our method is based on distance labeling to vertices, which is common to the existing exact distance querying methods, but our labeling algorithm stands on a totally new idea. Our algorithm conducts breadth-first search (BFS) from all the vertices with pruning. Though the algorithm is simple, our pruning surprisingly reduce the search space and the labels, resulting in fast preprocessing time, small index size and fast query time. Moreover, we also proposed another labeling scheme exploiting bit-level parallelism, which can be easily combined with the pruned labeling method to further improve the performance. Extensive experimental results on large-scale real-world networks of various types demonstrated the efficiency and robustness of our methods. In particular, our method can handle networks with hundreds of millions of vertices, which are two orders of magnitude larger than the limits of the previous methods, with comparable index size and query time.

KEYWORD SEARCH IN SPATIAL DATABASES: TOWARDS SEARCHING BY DOCUMENT

This work addresses a novel spatial keyword query called the m-closest keywords (mCK) query. Given a database of spatial objects, each tuple is associated with some

descriptive information represented in the form of keywords. The mCK query aims to find the spatially closest tuples which match m user-specified keywords. Given a set of keywords from a document, mCK query can be very useful in geotagging the document by comparing the keywords to other geotagged documents in a database. To answer mCK queries efficiently, we introduce a new index called the bR*-tree, which is an extension of the R*-tree. Based on bR*-tree, we exploit a priori-based search strategies to effectively reduce the search space. We also propose two monotone constraints, namely the distance mutex and keyword mutex, as our a priori properties to facilitate effective pruning. Our performance study demonstrates that our search strategy is indeed efficient in reducing query response time and demonstrates remarkable scalability in terms of the number of query keywords which is essential for our main application of searching by document.

With the ever-increasing popularity of services such as Google Earth and Yahoo Maps, as well as other geographic applications, queries in spatial databases have become increasingly important in recent years. Current research on queries goes well beyond pure spatial queries such as nearest neighbor queries, range queries, and spatial joins. Queries on spatial objects associated with textual information represented by sets of keywords are beginning to receive significant attention from the spatial database research community and the industry. This paper focuses on a novel type of query called the m**closest keywords** (mCK) query: given m keywords provided by the user, the mCK query aims at finding the closest tuples (in space) that match these keywords. While such a query has various applications, our main interest lies in that of a **search by document**. As an example, Fig. 1 shows the spatial distribution of three keywords that are obtained from place marks in some mapping application. Given a blog that contains these three keywords, the user is interested to find a spatial location that the blog is likely to be relevant to.

In this paper, we take a step towards searching by document by addressing the mCK query. We use the bR*-tree to effectively summarize keyword locations, thereby facilitating pruning. We propose effective a priori-based search strategies for mCK query processing. Two monotone constraints and their efficient implementations are also discussed. Our performance study on both synthetic and real data sets demonstrates that the proposed bR*-tree answers mCK queries efficiently within relatively short query response time. Furthermore, it demonstrates remarkable scalability in terms of the number of query keywords and significantly outperforms the existing MWSJ approach when m is large. While handling large number of keywords is an important step towards searching by document, there is still much room for future research.

EFFICIENT RETRIEVAL OF THE TOP-K MOST RELEVANT SPATIAL WEB OBJECTS

The conventional Internet is acquiring a geo-spatial dimension. Web documents are being geo-tagged, and geo-referenced objects such as points of interest are being associated with descriptive text documents. The resulting fusion of geo-location and documents enables a new kind of top-k query that takes into account both location proximity and text relevancy. To our knowledge, only naive techniques exist that are capable of computing a general web information retrieval query while also taking location into account. This paper proposes a new indexing framework for location aware top-k text retrieval. The framework leverages the inverted file for text retrieval and the R-tree for spatial proximity querying. Several indexing approaches are explored within the framework. The framework encompasses algorithms that utilize the proposed indexes for computing the top-k query, thus taking into accounts both text relevancy and location proximity to prune the search space. Results of empirical studies with an implementation of the framework demonstrate that the paper's proposal offers scalability and is capable of excellent performance.

Driven in part by the emergence of the mobile Internet, the conventional Internet is acquiring a geo-spatial dimension. On the one hand, many (geo-referenced) points of interest—e.g., stores, tourist attractions, hotels, entertainment services, public transport, and public services—are being associated with descriptive text documents. On the other hand, web documents are increasingly being geo-tagged. This fusion of geo-location and documents enables queries that take into account both location proximity and text relevancy. One study has found that about one fifth of web search queries are geographical and have local intent, as determined by the presence of geographical terms such as place names and postal codes . Indeed commercial search engines have started to provide location based services, such as map services, local search, and local advertisements. For example, Google Maps supports location-aware text retrieval queries. Additional examples of location-based services include online yellow pages. This paper considers a new kind of top-k query that takes into account both location proximity and text relevancy for points of interest with associated text. An example query may request a "good micro-brewery that serves pizza" and that is close to the user's hotel. We call this type of query a location-aware top-k text retrieval (LkT) query. The answer to such a top-k query is a list of k objects ranked according to a

ranking function that combines their distances to the query location and the relevance of their textual descriptions to the query phrase. The LkT query is different from the query that retrieves relevant documents within a geographic range. In the paper, we compute the text relevancy of a query result by means of language models and a probabilistic ranking function that have sound foundations in statistical theory and have performed well empirically in many information retrieval tasks . The LkT query poses new challenges for both existing spatial database and existing information retrieval techniques that have been developed separately. The research in spatial databases mainly focuses on highly structured, map-based geometric data and their attributes. In contrast, information retrieval research often treats location information as common keywords.

This paper proposes a new indexing framework for location aware top-k text retrieval. The framework integrates the inverted file for text retrieval and the R-tree for spatial proximity querying in a novel manner. Several hybrid indexing approaches are explored within the framework. The framework encompasses algorithms that utilize the proposed indexes for computing the top-k query, and it is capable of taking into account both text relevancy and location proximity to prune the search space at query time. Results of empirical studies with an implementation of the framework demonstrate that the paper's proposal offers scalability and is capable of excellent performance. This work opens to a number of promising directions for future work. First, it is worth adapting existing optimization techniques developed for the inverted file (e.g., compression) and R-trees to the paper's setting. Second, it is of interest to develop algorithms for other type of queries, e.g., range queries, based on the hybrid index. Third, it would be interesting to understand how the top-k queries considered can best be processed if the spatial objects are constrained to a road network

SPATIAL KEYWORD QUERY PROCESSING: AN EXPERIMENTAL EVALUATION

Geo-textual indices play an important role in spatial keyword querying. The existing geo-textual indices have not been compared systematically under the same experimental framework. This makes it difficult to determine which indexing technique best supports specific functionality. We provide an all-around survey of 12 state of-the-art geo-textual indices. We propose a benchmark that enables the comparison of the spatial keyword query performance. We also report on the findings obtained when applying the benchmark to the indices, thus uncovering new insights that may guide index selection as well as further research.

With the proliferation of online objects with both an associated geo-location and a text description, the web is acquiring a spatial dimension. Specifically, web users and content are increasingly being geo-positioned and geo-coded. At the same time, textual descriptions of points of interest, e.g., cafes and tourist attractions are increasingly becoming available on the web. This development calls for techniques that enable the indexing of data that contains both text descriptions and geo-locations in order to support the efficient processing of spatial keyword queries that take a geo-location and a set of keywords as arguments and return relevant content that matches the arguments. Spatial keyword queries are being supported in real-life applications, such as Google Maps where points of interest can be retrieved, Foursquare where geo-tagged documents can be retrieved, and Twitter where tweets can be retrieved. Spatial keyword querying is also receiving increasing interest in the research community where a range of techniques have been proposed for efficiently processing spatial keyword queries. Three types of spatial keyword queries are receiving particular attention, namely the Boolean kNN query, the top-k kNN query, and the Boolean range query. We proceed to illustrate them with examples.

By surveying and subjecting 12 geo-textual indexing techniques empirical study, the paper offers structure that may help the area of geo-textual indexing progress more effectively. The paper considers the support for three fundamental kinds of geo-textual queries.

OPTIMAL ROUTE QUERIES WITH ARBITRARY ORDER CONSTRAINTS

Given a set of spatial points DS, each of which is associated with categorical information, e.g., restaurant, pub, etc., the optimal route query finds the shortest path that starts from the query point (e.g., a home or hotel), and covers a user-specified set of categories (e.g., {pub, restaurant, museum}). The user may also specify partial order constraints between different categories, e.g., a restaurant must be visited before a pub. Previous work has focused on a special case where the query contains the total order of all categories to be visited (e.g., museum! restaurant! pub). For the general scenario without such a total order, the only known solution reduces the problem to multiple, total-order optimal route queries. As we show in this paper, this nai¨ve approach incurs a significant amount of repeated computations, and, thus, is not scalable to large data sets. Motivated by this, we propose novel solutions to the general optimal route query, based on two different methodologies, namely backward search and forward search. In addition, we discuss how the proposed methods can be adapted to answer a variant of the optimal route

queries, in which the route only needs to cover a subset of the given categories. Extensive experiments, using both real and synthetic data sets, confirm that the proposed solutions are efficient and practical, and outperform existing methods by large margins.

Consider a tourist who will have a free day to travel around Hong Kong. Without much knowledge about the city, she/he searches online maps to plan for a trip. Usually, she/he has a fixed starting point, e.g., her/his hotel, and certain objectives in mind, such as visiting a museum, dining at a fine restaurant, and enjoying a few drinks at a local pub. Meanwhile, some destinations may need to be visited in a certain order. For instance, the trip should have a pub after a restaurant. The ideal route should cover all the destinations, satisfy all order constraints, and minimize the total travel length. Searching for such a route is captured by the optimal route query , , which usually has a vast search space, and, consequently, is too tedious to be done manually. Currently, major online map providers have already shown interest in tools that assist such trip planning tasks. For example, Google City Tours (citytours.googlelabs.com) provides suggested tours for a given starting address. However, these tours are predefined, and cannot be customized according to the user's plans. Yahoo Travel (travel.yahoo.com) has a similar service that allows users to search and share trips, which, unfortunately, cannot answer optimal route queries either. Fig. 1 illustrates an example optimal route query on a data set DS with six locations p1-p6. Each location is associated with one category Cp, e.g., p1;p2 are museums;

This paper investigates the problem of optimal route query processing. Existing solutions are either limited to a specific setting of the problem, or incur expensive, redundant computations. Hence, we propose novel and efficient solutions, based on two methodologies: backward and forward search. The solution BFS that combines merits from both backward and forward search achieves the best performance. In the future, we plan to study alternative definitions of the optimal route query, that have temporal constraints (e.g., have lunch at a specified period) or maximize the number of categories to be visited given a total travel length budget.

SYSTEM IMPLEMENTATION

Implementation is the process that actually yields the lowest-level system elements in the system hierarchy (system breakdown structure). The system elements are made, bought, or reused. Production involves the hardware fabrication processes of forming, removing, joining, and finishing; or the software realization processes of coding and testing; or the operational procedures development processes for operators' roles. If implementation involves a production process, a manufacturing system which uses the established technical and management processes may be required.

The purpose of the implementation process is to design and create (or fabricate) a system element conforming to that element's design properties and/or requirements. The element is constructed employing appropriate technologies and industry practices. This process bridges the system definition processes and the integration process.

System Implementation is the stage in the project where the theoretical design is turned into a working system. The most critical stage is achieving a successful system and in giving confidence on the new system for the user that it will work efficiently and effectively. The existing system was long time process.

The proposed system was developed using .NET. The existing system caused long time transmission process but the system developed now has a very good user-friendly tool, which has a menu-based interface, graphical interface for the end user. After coding and testing, the project is to be installed on the necessary system. The executable file is to be created and loaded in the system. Again the code is tested in the installed system. Installing the developed code in system in the form of executable file is implementation.

FUTURE ENHANCEMENT

First, users may prefer a more generic preference model, which combines PoI rating, PoI average menu price, etc, in the query clue. Second, it is of interest to take temporal information into account and further extend the CRS query. Each PoI is assigned with a opening hours time interval [To, Tc], and each clue contains a visiting time t, where the resulting query aims to find a path such that the time interval of each matched PoI covers the visiting time. Third, requiring users to provide exact keyword match is difficult sometimes as they are just providing "clue", which may be inaccurate in nature. Thus, it is of interest to extend our model to support the approximate keyword match. Hence, the matching distance can be modified by incorporating both spatial distance and textual distance together through a linear combination.

CONCLUSION:

We study the problem of CRS on road networks, which aims to find an optimal route such that it covers a set of query keywords in a given specific order, and the matching distance is minimized. To answer the CRS query,

we first propose a greedy clue-based algorithm GCS with no index where the network expansion approach is adapted to greedily select the current best candidates to construct feasible paths. Then, we devise an exact algorithm, namely clue-based dynamic programming CDP, to answer the query that enumerates all feasible paths and finally returns the optimal result. To further reduce the computational overhead, we propose a branch and-bound algorithm BAB by applying filter-and-refine paradigm such that only a small portion of vertices are visited, thus improves the search efficiency. In order to quickly locate the candidate vertices, we develop AB-tree and PB-tree structures to speed up the tree traversal, as well as a semi-dynamic index updating mechanism. Results of empirical studies show that all the proposed algorithms are capable of answering CRS query efficiently, while the BAB algorithm runs much faster, and the index size of PB-tree is much smaller than AB-tree.

[1]. REFERENCES

[2]. B. Zheng, N. J. Yuan, K. Zheng, X. Xie, S. Sadiq, and X. Zhou,"Approximate keyword search in semantic trajectory database,"in Proc. IEEE Int. Conf. Data Eng., 2015, pp. 975–986.

[3]. K. Zheng, S. Shang, N. J. Yuan, and Y. Yang, "Towards efficient search for activity trajectories," in Proc. IEEE Int. Conf. Data Eng., 2013, pp. 230–241.

[4]. X. Cao, G. Cong, C. S. Jensen, and B. C. Ooi, "Collective spatial keyword querying," in Proc. ACM SIGMOD Int. Conf. Manage. Data, 2011, pp. 373–384.

[5]. C. Long, R. C.-W. Wong, K. Wang, and A. W.-C. Fu, "Collective spatial keyword queries: A distance owner-driven approach," in Proc. ACM SIGMOD Int. Conf. Manage. Data, 2013, pp. 689–700.

[6]. T. Guo, X. Cao, and G. Cong, "Efficient algorithms for answering the m-closest keywords query," in Proc. ACM SIGMOD Int. Conf. Manage. Data, 2015, pp. 405–418.

[7]. D. Zhang, Y. M. Chee, A. Mondal, A. K. Tung, and M. Kitsuregawa, "Keyword search in spatial databases: Towards searching by document," in Proc. IEEE Int. Conf. Data Eng., 2009, pp. 688– 699.

[8]. J. Liu, K. Deng, H. Sun, Y. Ge, X. Zhou, and C. S. Jensen, "Cluebased spatio-textual query," Proc. VLDB Endowment, vol. 10, no. 5, pp. 529–540, 2017.

[9]. F. Li, D. Cheng, M. Hadjieleftheriou, G. Kollios, and S.-H. Teng, "On trip planning queries in spatial databases," in Proc. 9th Int. Conf. Advances Spatial Temporal Databases, 2005, pp. 273–290.

[10]. X. Cao, L. Chen, G. Cong, and X. Xiao, "Keyword-aware optimal route search," Proc. VLDB Endowment, vol. 5, no.11, pp. 1136– 1147, 2012.

[11]. M. Sharifzadeh, M. Kolahdouzan, and C. Shahabi, "The optimal sequenced route query," VLDB J., vol. 17, no. 4, pp. 765–787, 2008.

I. Abraham, D. Delling, A. V. Goldberg, and R. F. Werneck, "Hierarchical hub labelings for shortest paths," in Proc. 20th Annu. Eur. Conf. Algorithms, 2012, pp. 24–35.

[12]. T. Akiba, Y. Iwata, and Y. Yoshida, "Fast exact shortest-path distance queries on large networks by pruned landmark labeling," in Proc. ACM SIGMOD Int. Conf. Manage. Data, 2013, pp. 349–360.

[13]. E. W. Dijkstra, "A note on two problems in connexion with graphs," Numerische Mathematik, vol. 1, no. 1, pp. 269–271, 1959.

[14]. M. Jiang, A. W.-C. Fu, R. C.-W. Wong, and Y. Xu, "Hop doubling label indexing for point-to-point distance querying on scale-free networks," Proc. VLDB Endowment, vol. 7, no. 12, pp. 1203–1214, 2014.

[15]. T. Akiba, Y. Iwata, K.-I. Kawarabayashi, and Y. Kawata, "Fast shortest-path distance queries on road networks by pruned highway labeling," in Proc. Meeting Algorithm Eng. Expermiments, 2014, pp. 147–154.

[16]. C. S. Jensen, J. Kol_a_rvr, T. B. Pedersen, and I. Timko, "Nearest neighbor queries in road networks," in Proc. 11th ACM Int. Symp. Advances Geographic Inf. Syst., 2003, pp. 1–8.

[17]. B. Zheng, et al., "Keyword-aware continuous kNN query on road networks," in Proc. IEEE Int. Conf. Data Eng., 2016, pp. 871–882.

[18]. T. Akiba, Y. Iwata, and Y. Yoshida, "Dynamic and historical shortest- path distance queries on large evolving networks by pruned landmark labeling," in Proc. 23rd Int. Conf. World Wide Web, 2014, pp. 237–248.

[19]. J. L. Bentley and J. B. Saxe, "Decomposable searching problems I. Static-to-dynamic transformation," J. Algorithms, vol. 1, no. 4, pp. 301–358, 1980.

I. De Felipe, V. Hristidis, and N. Rishe, "Keyword search on spatial databases," in Proc. IEEE Int. Conf. Data Eng., 2008, pp. 656–665.

[20]. G. Cong, C. S. Jensen, and D. Wu, "Efficient retrieval of the top-k most relevant spatial web objects," Proc. VLDB Endowment, vol. 2, pp. 337–348, 2009.

[21]. X. Cao, G. Cong, and C. S. Jensen, "Retrieving top-k prestigebased relevant spatial web objects," Proc. VLDB Endowment, vol. 3, pp. 373–384, 2010.

[22]. L. Chen, G. Cong, C. S. Jensen, and D. Wu, "Spatial keyword query processing: An experimental evaluation," Proc. VLDB Endowment, vol. 6, pp. 217–228, 2013.

[23]. D. Zhang, B. C. Ooi, and A. K. Tung, "Locating mapped resources in web 2.0," in Proc. IEEE Int. Conf. Data Eng., 2010, pp. 521–532.

[24]. G. Li, J. Feng, and J. Xu, "DESKS: Direction-aware spatial keyword search," in Proc. IEEE Int. Conf. Data Eng., 2012, pp. 474–485.

[25]. J. B. Rocha-Junior and K. Nørva g, "Top-k spatial keyword queries on road networks," in Proc. 15th Int. Conf. Extending Database Technol., 2012, pp. 168–179.

[26]. K. C. Lee, W.-C. Lee, and B. Zheng, "Fast object search on road networks," in Proc. 12th Int. Conf. Extending Database Technol., 2009, pp. 1018–1029.

[27]. R. Zhong, G. Li, K.-L. Tan, and L. Zhou, "G-tree: An efficient index for KNN search on road networks," in Proc. 22nd ACM Int. Conf. Inf. Knowl. Manage., 2013, pp. 39–48.

[28]. M. Jiang, A. W.-C. Fu, and R. C.-W. Wong, "Exact top-k nearest keyword search in large networks," in Proc. ACM SIGMOD Int. Conf. Manage. Data, 2015, pp. 393–404.