Concise Query Processing in Uncertain Database

M. Sowmya
Dept of CSE,
KITS, Warangal
Andhra Pradesh, India.

V. Shankar
Associate Professor,
Dept of CSE,
KITS ,Warangal.

ABSTRACT

Wireless communication technology has been rapidly increasing, it became quite common for people to view maps or get related services from the handheld devices, such as mobile phones and PDAs. Spatial databases have witnessed an increasing number of applications recently, due to the fast advance in the fields of mobile computing and embedded systems and the spread of the Internet .Range queries are often posed by user to retrieve the useful information from a spatial database. We present a novel idea that a concise representation of a specified size for the range query results, while incurring minimal information loss, shall be computed and returned to the user. Such a concise range query not only reduces communication costs, but also offers better usability to the users, providing an opportunity for interactive exploration. The usefulness of the concise range queries is confirmed by comparing it with other possible alternatives, such as sampling and clustering. In this proposed system, we include the entities and associate the object attributes such as restaurants, shopping places etc which represents a point within a Hilbert curve which facilitates in reducing search space for spatial data, and to provide a range for attribute such that all the information is retrieved with minimal loss. The proposed system also includes peer to peer system through which multiple spatial databases can be accessed in efficient time.

Keywords

Spatial Database, Range Queries

1. INTRODUCTION

Spatial databases have witnessed an increasing number of applications recently, partially due to the fast advance in the fields of mobile computing and embedded systems and the spread of the Internet. For example, it is quite common these days that people want to figure out the driving or walking directions from their handheld devices (mobile phones or PDAs). However, facing the huge amount of spatial data collected by various devices, such as sensors and satellites, and limited bandwidth and/or computing power of handheld devices, how to deliver light but usable results to the clients is a very interesting, and of course, challenging task. Range queries, as one of the most commonly used tools, are often posed by the users to retrieve needful information from a spatial database. However, due to the limits of communication bandwidth and hardware power of handheld devices, displaying all the results of a range query on a handheld device is neither communication efficient nor informative to the users. This is simply because that there are often too many results returned from a range query.

2. PREVIOUS WORK

Continuing advances in consumer electronics, mobile communications, and positioning technologies combine to render it increasingly realistic to assume that entire populations of users of mobile services, termed moving objects, can be tracked accurately. These developments offer a foundation for the delivery of increasingly sophisticated location-enabled mobile services. Motivated by this scenario, one line of research aims to provide appropriate data management foundations that enable the provisioning of efficient services. Proposals exist for the efficient computation of, e.g., window queries and nearest neighbour queries on moving objects.

With the rapid growth in database, networking, and computing technologies, a large amount of personal data can be integrated and analyzed digitally, leading to an increased use of data-mining tools to infer trends and patterns. This has raised universal concerns about protecting the privacy of individuals. Combining data tables from multiple data sources allows us to draw inferences which are not possible from a single source. For example, combining patient data from multiple hospitals is useful to predict the outbreak of an epidemic. The traditional approach of releasing the data tables without breaching the privacy of individuals in the table is to de-identify records by removing the identifying fields such as name, address, and social security number. However, joining this de-identified table with a publicly available database (like the voters database) on columns like race, age, and zip code can be used to identify individuals. Recent research has shown that for 87% of the population in the United States, the combination of non-key fields like date of birth, gender, and zip code corresponds to a unique person. Such non-key fields are called *quasi-identifiers*. In what follows we assume that the identifying fields have been removed and that the table has two types of attributes: (1) the quasi-identifying attributes explained above and (2) the sensitive attributes that need to be protected.

3. PROPOSED SYSTEM

Many researchers have focused on the problem of K nearest neighbor (KNN) queries in spatial databases. This type of query is frequently used in Geographical Information Systems and is defined as: given a set of spatial objects (or points of interest), and a query point, find the K closest objects to the query. An ex- ample of KNN query is a query initiated by a GPS device in a vehicle to find the 5 closest restaurants to the vehicle. With spatial network databases (SNDB), objects are restricted to move on pre-defined paths (e.g., roads) that are specified by an underlying network. This means that the shortest network distance (e.g., shortest path, shortest time) between objects (e.g., the vehicle and the restaurants) depend on the connectivity of the network rather than the objects' locations. The majority of the existing works on KNN queries are based on either computing the distance between a query and the objects on-line, or utilizing index structures. The solution proposed by the first group is based on the fact that the current algorithms (e.g., Dijkstra) for computing the

distance between a query object *q* and an object *O* in a network will automatically result in the computation of the distance between *q* and the objects that are (relatively) closer to *q* than *O*. These approaches apply an optimized network expansion algorithm with the advantage that the network expansion only explores the objects that are closer to *q* and computes their distances to *q* during expansion. However, the main disadvantage of these approaches is that they perform poorly when the objects are not densely distributed in the network because then they require retrieving a large portion of the network for distance computation. The second group of approaches is designed and optimized for metric or vector spatial index structures (e.g., m-tree and r-tree, respectively). These approaches require pre- computations of the distances between objects and object groups based on their distances to some reference nodes (this is more intelligent as compared to a naïve approach that pre-computes and stores distances between all the node-pairs in the network). These solutions filter a small subset of possibly large number of objects as the candidates for the closest neighbors of *q*, and require a refinement step to compute the actual distances between *q* and the candidates to find the actual nearest neighbors of *q*. The main drawback of applying these approaches on SNDB is that they do not offer any solution as how to efficiently compute the distances between *q* and the candidates.

Moreover, applying an approach similar to the first group to perform the refinement step in order to compute the distances between *q* and the candidates will render these approaches, which traverse index structures to provide a candidate set, redundant since the network expansion approach does not require any candidate set to start with. In addition to this drawback, approaches that are based on vector index structures are only appropriate for spaces where the distance between objects is only a function of their spatial attributes (e.g., Euclidean distance) and cannot properly approximate the distances in a network. [4]

Location-based services (LBS) combine the functionality of location-aware devices (e.g., GPS-like devices), wireless and cellular phone technologies, and information management to provide personalized services to users based on their current locations. Examples of LBS include location-aware emergency services (Dispatch the nearest ambulance"), location-based advertisement (Send e-coupons to all cars that are within two miles of my gas station"), live traffic reports (Let me know if there is congestion within ten minutes of my route"), and location-based store finders (Where is my nearest restaurant"). Users registered with LBS continuously send their locations to a location-based database server. Upon requesting a service, a registered user issues a location-based query that is executed at the server based on the knowledge of the user's location. Location-based queries are either snapshot or continuous queries. Examples of snapshot queries include Where is my nearest gas station" and what are the restaurants within one mile of my location", while examples of continuous queries include continuously report my nearest police car" and continuously report the gas stations within one mile of my car".

Although LBS promise safety and convenience, they threaten the privacy and security of their users. The privacy threat comes from the fact that LBS providers rely mainly on an implicit assumption that users agree to reveal their private locations to get services. In other words, a user trades her privacy with the service. If a user wants to keep her private location information, she has to turn of her location aware device and (temporarily) unsubscribe from the service. With untrustworthy servers, such a model poses several privacy threats. For example, an employer may check on her employee's behavior by knowing the places where she visits and the time of each visit, the personal medical records can be inferred by knowing which clinic a person visits, or someone can track the locations of his ex-friends. In fact, in many cases, GPS devices have been used in stalking personal locations. Unfortunately, the traditional approach of pseudonymity (i.e., using a fake identity) is not applicable to LBS where the location information of a person can directly lead to the true identity.

For example, asking about the nearest Pizza restaurant to my house using a fake identity will reveal my true identity, as a resident of the house. In an attempt to preserve the privacy of LBS users, several research groups have presented the concept of a location anonymizer that is responsible for blurring actual users' locations into cloaked areas. Upon registration with the location anonymizer, mobile users specify their own desired level of privacy through a user-specified privacy profile that may contain one or more of the following parameters: k-anonymity, minimum area Amin, and maximum area Amax. K-anonymity indicates that the user wants to be k-anonymous, i.e., not distinguishable among k users, while Amin and Amax indicate that the user wants to hide her location information within an area of at least Amin and at most Amax, respectively. The location anonymizer is basically a trusted third party that acts as a middle layer between mobile users and the location-based database server in order to: (1) receive the exact location information from mobile users along with a privacy profile of each user, (2) employ an existing location anonymization technique to blur users' exact locations into cloaked areas that satisfy each user's privacy profile, (3) send the cloaked areas to the database server, and (4) compute the exact answer from a candidate list of answers returned by the database server and send the exact answer to the user.

Therefore, the goal of a concise range query is to find a concise representation, with the user-specified size, for all the points inside the query range. Ideally, one would like to have a concise representation of minimum information loss. We first give a dynamic programming algorithm that finds the optimal solution in one dimension in Section 3.1. Unfortunately, this optimization problem in two or more dimensions is NP-hard. In Section 3.2, we present a nontrivial reduction from PLANAR 3-SAT to the concise representation problem and prove its NP-hardness. Nevertheless, in our applications, the optimal solution is often unnecessary while efficiency is typically important.

3.1 Optimal Data Initialization

In this module we present a dynamic programming algorithm for computing the optimal concise representation for a set of points P lying on a line. We will extend this concept to higher dimensions, leading to an efficient heuristic. Let p1; . . . ; pn be the points of P in sorted order. Let Pi;j represent optimal partitioning underlying the best concise representation, i.e., with the minimum information loss, for the first i points of size j. The optimal solution is simply the concise representation for different points which could be found using a dynamic programming approach. The key observation is that in one dimension, the optimal partitioning always contains segments that do not overlap, i.e., we should always

create a group with consecutive points without any point from another group in-between. Since our problem is also a clustering problem, it is tempting to use some popular clustering heuristic, such as the well-known k-means algorithm, for our problem as well. However, since the object function makes a big difference in different clustering problems. The clustering constraint there is that each cluster has at least k points, while we require that the number of clusters is k.

3.2 Hilbert Group

Given the optimal algorithm in one dimension, a straightforward idea is to use a function IR to map the points of P from higher dimensions down to one dimension. We propose a space-filling curve traverses the space in a predetermined order. The most widely used space-filling curve is the Hilbert curve. The hth-order Hilbert curve has 2hd cells in d dimensions and visits all the cells. Each cell will be assigned a Hilbert value in sequence starting from 0, and all points falling inside the cell will get the same Hilbert value. The basic idea is to first compute the Hilbert value for each point in P, and sort all points by this value, mapping them to one dimension. Then, we simply group these points using our 1D dynamic programming. The optimality of the algorithm will be lost, and the quality of the result will depend on how well the Hilbert curve preserves the neighborhood information among points in the original, higher dimensional space.

3.3 Iterative Group

In this module we present direct algorithm in two or more dimensions. It is an iterative algorithm that finds the k groups P1; . . . ; Pk, one at a time .In each iteration, we start with a seed, randomly chosen from the remaining points, and greedily add points into the group one by one. In the ith iteration, we first initialize the group Pi to include only the seed. Let U be the set of remaining points. Let the extent and area of the minimum bounding box of the set of points X, respectively. As we add points into Pi, we keep an estimated total information loss. When deciding which point p should be added to Pi, we first compute the minimizing values. The intuition is that, if we have k bounding boxes left to group the points in U, then one partitioning that is always doable is to draw k squares, to enclose all the points, which results in an information loss equal to the second term cannot be further reduced by adding more points, we should probably stop and start a new group. Finally, for the last group Pk, we simply put all the remaining points into it.

3.4 Query Processing with R-Trees

In order to use the algorithms above a concise range query Q with budget k from the client, the database server would first need to evaluate the query as if it were a standard range query using some spatial index built on the point set P, typically an R-tree. After obtaining the complete query results, the server then partitions P into k groups and returns the concise representation. However, as the main motivation to obtain a concise answer is exactly because P is too large, finding the entire P and running the algorithms are often too expensive for the database server. In this module, we present algorithms that process the concise range query without computing P in its entirety. The idea is to first find k bounding boxes that collectively contain all the points in P by using the existing spatial index structure on P. Each of these bounding boxes is also associated with the count of points inside. Therefore, adopting concise range queries instead of the traditional exact range queries not only solves the bandwidth and usability problems, but also leads to substantial efficiency improvements. We first group B points in proximity area into a minimum bounding rectangle (MBR); these points will be stored at a leaf on the R-tree. These MBRs are then further grouped together level by level until there is only one left. Each node in the R-tree is associated with the MBR enclosing all the points stored below, denoted by MBR. Each internal node also stores the MBRs of all its children. The standard range query Q can be processed using an R-tree as follows: We start from the root of the R-tree, and check the MBR of each of its children. Then, we recursively visit any node u whose MBR intersects or falls inside Q. When we reach a leaf, we simply return all the points stored there that are inside Q

4. SIMULATION RESLUTS

The concept of this paper is implemented in .Net technology (Visual studio c#)and with Oracle database. The proposed paper's concepts shows efficient results and has been efficiently tested on different Datasets. The figures below shows the real time results compared.

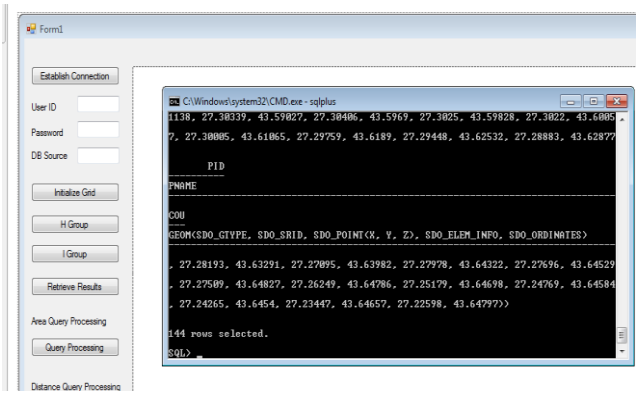

Fig .1 Necessary datasets stored in database

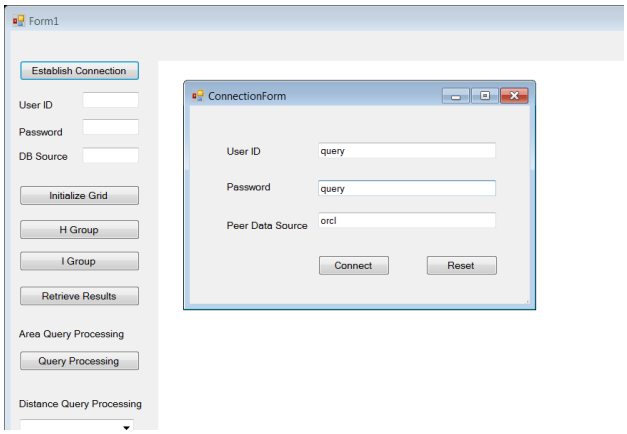

Fig. 1 Data base connection form.

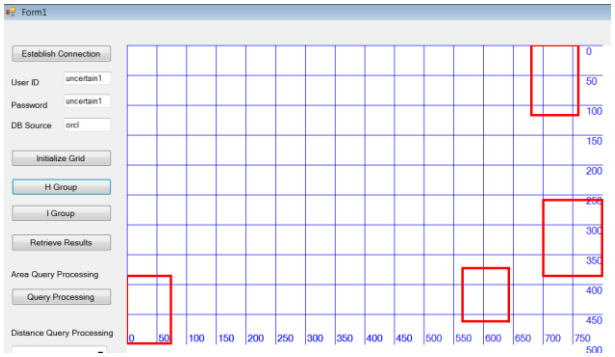

Fig. 1 Proposed system performing Hilbert Group

Hilbert curve calculates the range of all the points (query) posed by user and tries to map the points from higher dimension to one dimension.

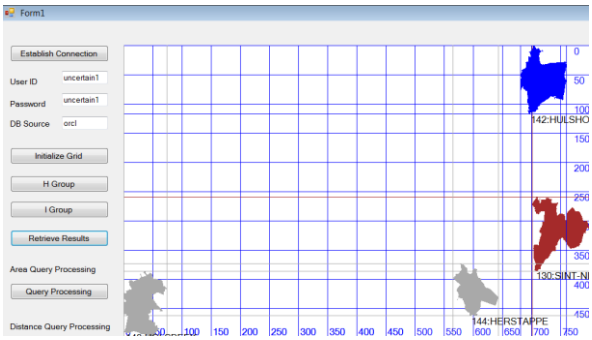

Fig. 3 Proposed system retrieving regional data

The above screen shows only the required and need full regions requested by user rather than all the unnecessary information which reduces query result size which saves communication band width, client memory .Usability of all query results can also be increased.

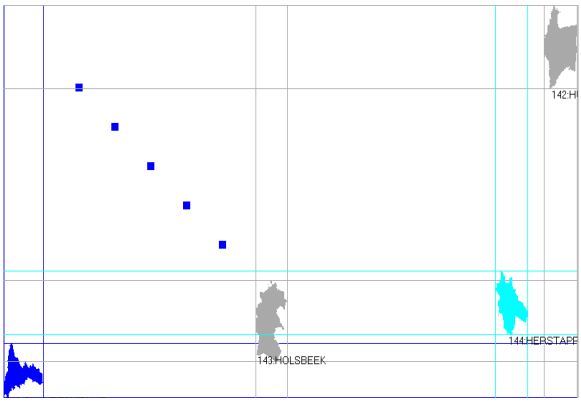

Fig. 4 Proposed system Hilbert Group Moving Objects

Here we can know the path of the object moving from one location to other location.

The comparative Study of the Hilbert Grouping and the Iterative Grouping with the R-Trees are shown below:

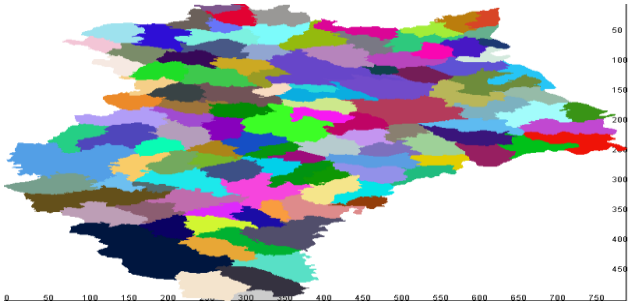

Fig. 5 Hilbert Grouping

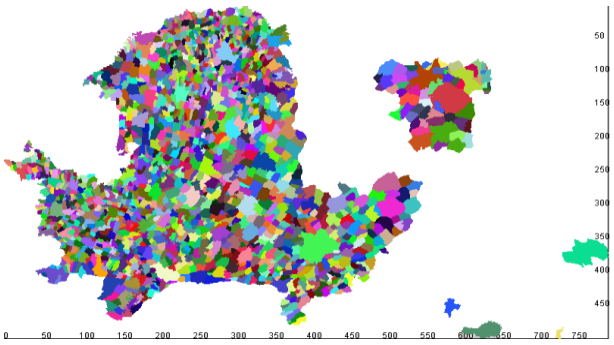

Fig. 6 Comparative Iterative Grouping

In Hilbert Grouping , space-filling curve traverses the space in a predetermined order where as in iterative grouping ,k groups are formed one at a time and for each iteration we start with a seed, randomly chosen from the remaining points, and greedily add points into the group one by one.

5. CONCLUSION

A new concept that of concise range queries has been proposed in this paper, which simultaneously addresses the following three problems of traditional range queries. First, it reduces the query result size significantly as required by the user. The reduced size saves communication bandwidth an also the client's memory and computational resources, which are of highest importance for mobile devices. Second, although the query size has been reduced, the usability of the query results has been actually improved. The concise representation of the results often gives the user more intuitive ideas and enables interactive exploration of the spatial database. Finally, we have designed R-tree-based algorithms so that a concise range query can be processed much more efficiently than evaluating the query exactly, especially in terms of I/O cost. This concept, together with its associated techniques presented here, could greatly enhance user experience of spatial databases, especially on mobile devices.
The proposed system can be enhanced by using the techniques based on the idea that one dimensional index can be reused in order to manage multidimensional data, if the dimensionality is reduced to one. This idea was the first to be explored. Techniques in the second category are based on the idea that centralized hierarchical indexes can be reused to manage dispersed multidimensional data, if they are properly distributed. More elaborated solutions have been proposed in this category. However, the reusage of existing techniques in the approaches in both categories leads to the maintenance of

some fundamental features that oppose to the nature of either the distributedness or the multidimensionality. Our intention is to overcome these shortcomings by creating a technique that manages disperse multidimensional data in an inherently distributed way without altering the dimensionality.

6. REFERENCES

[1] X. Lin, Y. Yuan, Q. Zhang, and Y. Zhang, "Selecting Stars: The k Most Representative Skyline Operator," Proc. Int'l Conf. Data Eng. (ICDE), 2007.

[2] C. Jermaine, S. Arumugam, A. Pol, and A. Dobra, "Scalable Approximate Query Processing with the dbo Engine," Proc. ACM SIGMOD, 2007.

[3] J. Xu, W. Wang, J. Pei, X. Wang, B. Shi, and A.W.-C. Fu, "Utility- Based Anonymization Using Local Recoding," Proc. ACMSIGKDD , 2006.

[4] H.V. Jagadish, B.C. Ooi, K.-L. Tan, C. Yu, and R. Zhang, "iDistance: An Adaptive B+-Tree Based Indexing Method for Nearest Neighbor Search," ACM Trans. Database Systems, vol. 30, no. 2, pp. 364-397, 2005.

[5] K. Mouratidis, D. Papadias, and S. Papadimitriou, "Tree-Based Partition Querying: A Methodology for Computing Medoids in Large Spatial Datasets," VLDB J., vol. 17, no. 4, pp. 923-945, 2008.

[6] C.S. Jensen, D. Lin, B.C. Ooi, and R. Zhang, "Effective Density Queries on Continuously Moving Objects," Proc. Int'l Conf. Data Eng. (ICDE), 2006.

[7] G. Ghinita, P. Karras, P. Kalnis, and N. Mamoulis, "Fast Data Anonymization with Low Information Loss," Proc. Int'l Conf. Very Large Data Bases (VLDB), 2007

[8] G. Aggarwal, T. Feder, K. Kenthapadi, S. Khuller, R. Panigrahy, D. Thomas, and A. Zhu, "Achieving Anonymity via Clustering," Proc. Symp. Principles of Database Systems (PODS), 2006.