ISSN: 2277-9655
(I2OR), Publication Impact Factor: 3.785
(ISRA), Impact Factor: 2.114

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

SPATIAL INVERTED INDEX FOR SEARCHING MUILTIDIMENSIONAL DATA

Syeda Farheen Fatima*, Raafiya Gulmeher
* Dept. Computer Science Engineering K.B.N Engg College Gulbarga, indian.
Dept. Computer Science Engineering K.B.N Engg College Gulbarga, indian.

ABSTRACT

Conventional spatial queries, such as range search and nearest neighbor retrieval, involve only conditions on objects' geometric properties. Today, many modern applications call for novel forms of queries that aim to find objects satisfying both a spatial predicate, and a predicate on their associated texts. For example, instead of considering all the restaurants, a nearest neighbor query would instead ask for the restaurant that is the closest among those whose menus contain "steak, spaghetti, brandy" all at the same time. Currently, the best solution to such queries is based on the IR2-tree, which, as shown in this paper, has a few deficiencies that seriously impact its efficiency. Motivated by this, we develop a new access method called the spatial inverted index that extends the conventional inverted index to cope with multidimensional data, and comes with algorithms that can answer nearest neighbor queries with keywords in real time. As verified by experiments, the proposed techniques outperform the IR2-tree in query response time significantly, often by a factor of orders of magnitude.

KEYWORDS: Nearest neighbor search, keyword search, spatial index.

INTRODUCTION

Aspatial database manages multidimensional objects (such as points, rectangles, etc.), and provides fast access to those objects based on different selection criteria. The importance of spatial databases is reflected by the convenience of modeling entities of reality in a geometric manner. For example, locations of restaurants, hotels, hospitals and so on are often represented as points in a map, while larger extents such as parks, lakes, and landscapes often as a combination of rectangles. Many functionalities of a spatial database are useful in various ways in specific contexts. For instance, in a geography information system, range search can be deployed to find all restaurants in a certain area, while nearest neighbor retrieval can discover the restaurant closest to a given address. Today, the widespread use of search engines has made it realistic to write spatial queries in a brandnew way. Conventionally, queries focus on objects' geometric properties only, such as whether a point is in a rectangle, or how close two points are from each other. We have seen some modern applications that call for the ability to select objects based on both of their geometric coordinates and their associated texts. For example, it would be fairly useful if a search engine can be used to find the nearest restaurant that offers "steak, spaghetti, and brandy" all at the same time. Note that this is not the "globally" nearest restaurant (which would have been returned by a traditional nearest neighbor query), but the nearest restaurant among only those providing all the demanded foods and drinks.There are easy ways to support queries that combine spatial and text features. For example, for the above query, we could first fetch all the restaurants whose menus contain the set of keywords {steak, spaghetti, brandy}, and then from the retrieved restaurants, find the nearest one. Similarly, one could also do it reversely by targeting first the spatial conditions—browse all the restaurants in ascending order of their distances to the query point until encountering one whose menu has all the keywords. The major drawback of these straightforward approaches is that they will fail to provide real timeanswers on difficult inputs. A typical example is that the real nearest neighbor lies quite faraway from the query point, while all the closer neighbors are missing at least one of the query keywords.

Spatial queries with keywords have not been extensively explored. In the past years, the community has sparked enthusiasm in studying keyword search in relational databases. It is until recently that attention was diverted to multidimensional data. The best method to date for nearest neighbor search with keywords is due to Felipe et al.. They nicely integrate two well-known concepts: R-tree, a popular spatial index, and signature file, an effective method for

ISSN: 2277-9655
(I2OR), Publication Impact Factor: 3.785
(ISRA), Impact Factor: 2.114

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. On the other hand, like signature files, the IR2-tree is able to filter a considerable portion of the objects that do not contain all the query keywords, thus significantly reducing the number of objects to be examined.

The IR2-tree, however, also inherits a drawback of signature
files: false hits. That is, a signature file, due to its conservative nature, may still direct the search to some objects, even though they do not have all the keywords. The penalty thus caused is the need to verify an object whose satisfying a query or not cannot be resolved using only its signature, but requires loading its full text. description, which is expensive due to the resulting random accesses. It is noteworthy that the false hit problem is not specific only to signature files, but also exists in other methods for approximate set membership tests with compact storage and the references therein). Therefore, the problem cannot be remedied by simply replacing signature file with any of those methods. In this paper, we design a variant of inverted index that is optimized for multidimensional points, and is thusnamed the spatial inverted index (SI-index). This access method successfully incorporates point coordinates into a conventional inverted index with small extra space, owing to a delicate compact storage scheme. Meanwhile, an SI-index preserves the spatial locality of data points, and comes with an R-tree built on every inverted list at little space overhead. As a result, it offers two competing ways for query processing. We can (sequentially) merge multiple lists very much like merging traditional inverted lists by ids. Alternatively, we can also leverage the R-trees to browse the points of all relevant lists in ascending order of their distances to the query point. As demonstrated by experiments, the SI-index significantly outperforms the IR2-tree in query efficiency, often by a factor of orders of magnitude. The rest of the paper is organized as follows. Section 2 defines the problem studied in this paper formally. Section 3 surveys the previous work related to ours. Section 4 gives an analysis that reveals the drawbacks of the IRtree. Section 5 presents a distance browsing algorithm for performing keyword-based nearest neighbor search. Section 6 proposes the SI-idnex, and establishes its theoretical properties. Section 7 evaluates our techniques with extensive experiments. Section 8 concludes the paper with a summary of our findings.

LITERATURE REVIEW

W.Arif , d. barbara, et. all 1995

Emerging multimedia applications require database systems to provide support for new types of objects and to process queries that may have no parallel in traditional database applications. One such important class of queries are the proximity queries that aims to retrieve objects in the database that are related by a distance metric in a way that is specified by the query. The importance of proximity queries has earlier been realized in developing constructs for visual languages. In this paper, we present algorithms for answering a class of proximity queries-fixed-radius nearest-neighbor queries over point object. Processing proximity queries using existing query processing techniques results in high CPU and I/O costs. We develop new algorithms to answer proximity queries over objects that lie in the one-dimensional space (e.g., words in a document). The algorithms exploit query semantics to reduce the CPU and I/O costs, and hence improve performance. We also show how our algorithms can be generalized to handle d-dimensional objects

G. cong, b. ooi, k. et. all 128-139

In constrained data mining, users can specify constraints to prune the search space to avoid mining uninteresting knowledge. This is typically done by specifying some initial values of the constraints that are subsequently refined iteratively until satisfactory results are obtained. Existing mining schemes treat each iteration as a distinct mining process, and fail to exploit the information generated between iterations. We propose to salvage knowledge that is discovered from an earlier iteration of mining to enhance subsequent rounds of mining. In particular, we look at how frequent patterns can be recycled. Our proposed strategy operates in two phases. In the first phase, frequent patterns obtained from an early iteration are used to compress a database. In the second phase, subsequent mining processes operate on the compressed database. We propose two compression strategies and adapt three existing frequent pattern mining techniques to exploit the compressed database. Results from our extensive experimental study show that our proposed recycling algorithms outperform their nonrecycling counterpart by an order of magnitude.

ISSN: 2277-9655
(I2OR), Publication Impact Factor: 3.785
(ISRA), Impact Factor: 2.114

R. hariharan, et. all 16, 2007.
¢Location-based information contained in publicly available GIS databases is invaluable for many applications such as disaster response, national infrastructure protection, crime analysis, and numerous others. The information entities of such databases have both spatial and textual descriptions. Likewise, queries issued to the databases also contain spatial and textual components, for example, "Find shelters with emergency medical facilities in Orange County," or "Find earthquake-prone zones in Southern California." We refer to such queries as spatial-keyword queries or SK queries for short. In recent times, a lot of interest has been generated in efficient processing of SK queries for a variety of applications from Web-search to GIS decision support systems. We refer to systems built for enabling such applications as Geographic Information Retrieval (GIR) Systems. An example GIR system that we address in this paper is a search engine built on top of hundreds of thousands of publicly available GIS databases. Building a search engine over such large repositories is a challenge. One of the key aspects of such a search engine is the performance. In this paper, we propose a framework for GIR systems and focus on indexing strategies that can process SK queries efficiently. We show through experiments that our indexing strategies lead to significant improvement in efficiency of answering SK queries over existing techniques.

D. papadias, q. shen et. all 2004.
¢Given two sets of points P and Q, a group nearest neighbor (GNN) query retrieves the point(s) of P with the smallest sum of distances to all points in Q. Consider, for instance, three users at locations q1 q2and q3 that want to find a meeting point (e.g., a restaurant); the corresponding query returns the data point p that minimizes the sum of Euclidean distances |pqi| for 1≤i≤3. Assuming that Q fits in memory and P is indexed by an R-tree, we propose several algorithms for finding the group nearest neighbors efficiently. As a second step, we extend our techniques for situations where Q cannot fit in memory, covering both indexed and nonindexed query points. An experimental evaluation identifies the best alternative based on the data and query properties.

H. shin, b. moon. et. all 2000
A spatial distance join is a relatively new type of operation introduced for spatial and multimedia database applications. Additional requirements for ranking and stopping cardinality are often combined with the spatial distance join in on-line query processing or internet search environments. These requirements pose new challenges as well as opportunities for more efficient processing of spatial distance join queries. In this paper, we first present an efficient k-distance join algorithm that uses spatial indexes such as R-trees. Bi-directional node expansion and plane-sweeping techniques are used for fast pruning of distant pairs, and the plane-sweeping is further optimized by novel strategies for selecting a sweeping axis and direction. Furthermore, we propose adaptive multi-stage algorithms for k-distance join and incremental distance join operations. Our performance study shows that the proposed adaptive multi-stage algorithms outperform previous work by up to an order of magnitude for both k-distance join and incremental distance join queries, under various operational conditions.

D. papadias et. all 2002.
¢Existing work on multiway spatial joins focuses on the retrieval of all exact solutions with no time limit for query processing. Depending on the query and data properties, however, exhaustive processing of multiway spatial joins can be prohibitively expensive due to the exponential nature of the problem. Furthermore, if there do not exist any exact solutions, the result will be empty even though there may exist solutions that match the query very closely. These shortcomings motivate the current work, which aims at the retrieval of the best possible (exact or approximate) solutions within a time threshold, since fast retrieval of approximate matches is the only way to deal with the ever increasing amounts of multimedia information in several real time systems. We propose various techniques that combine local and evolutionary search with underlying indexes to prune the search space. In addition to their usefulness as standalone methods for approximate query processing, the techniques can be combined with systematic search to enhance performance when the goal is retrieval of the best solutions.

H. shin, b. moon, , et. all 2000
¢A spatial distance join is a relatively new type of operation introduced for spatial and multimedia database applications. Additional requirements for ranking and stopping cardinality are often combined with the spatial distance join in on-line query processing or internet search environments. These requirements pose new challenges as well as opportunities for more efficient processing of spatial distance join queries.

ISSN: 2277-9655
(I2OR), Publication Impact Factor: 3.785
(ISRA), Impact Factor: 2.114

In this paper, we first present an efficient k-distance join algorithm that uses spatial indexes such as R-trees. Bi-directional node expansion and plane-sweeping techniques are used for fast pruning of distant pairs, and the plane-sweeping is further optimized by novel strategies for selecting a sweeping axis and direction. Furthermore, we propose adaptive multi-stage algorithms for k-distance join and incremental distance join operations. Our performance study shows that the proposed adaptive multi-stage algorithms outperform previous work by up to an order of magnitude for both k-distance join and incremental distance join queries, under various operational conditions.

S. agrawal et. all 2002
Internet search engines have popularized the keyword-based search paradigm. While traditional database management systems offer powerful query languages, they do not allow keyword-based search. In this paper, we discuss DBXplorer, a system that enables keyword-based searches in relational databases. DBXplorer has been implemented using a commercial relational database and Web server and allows users to interact via a browser front-end. We outline the challenges and discuss the implementation of our system, including results of extensive experimental evaluation.

N. beckmann et all 1990
The R-tree, one of the most popular access methods for rectangles, is based on the heuristic optimization of the area of the enclosing rectangle in each inner node. By running numerous experiments in a standardized testbed under highly varying data, queries and operations, we were able to design the R*-tree which incorporates a combined optimization of area, margin and overlap of each enclosing rectangle in the directory. Using our standardized testbed in an exhaustive performance comparison, it turned out that the R*-tree clearly outperforms the existing R-tree variants. Guttman's linear and quadratic R-tree and Greene's variant of the R-tree. This superiority of the R*-tree holds for different types of queries and operations, such as map overlay, for both rectangles and multidimensional points in all experiments. From a practical point of view the R*-tree is very attractive because of the following two reasons 1 it efficiently supports point and spatial data at the same time and 2 its implementation cost is only slightly higher than that of other R-trees.

G. bhalotia ,et. all 2002
¢With the growth of the Web, there has been a rapid increase in the number of users who need to access online databases without having a detailed knowledge of the schema or of query languages; even relatively simple query languages designed for non-experts are too complicated for them. We describe BANKS, a system which enables keyword-based search on relational databases, together with data and schema browsing. BANKS enables users to extract information in a simple manner without any knowledge of the schema or any need for writing complex queries. A user can get information by typing a few keywords, following hyperlinks, and interacting with controls on the displayed results. BANKS models tuples as nodes in a graph, connected by links induced by foreign key and other relationships. Answers to a query are modeled as rooted trees connecting tuples that match individual keywords in the query. Answers are ranked using a notion of proximity coupled with a notion of prestige of nodes based on inlinks, similar to techniques developed for Web search. We present an efficient heuristic algorithm for finding and ranking query results.

X. cao , l. chen , et. all 2012
The web is increasingly being used by mobile users. In addition, it is increasingly becoming possible to accurately geo-position mobile users and web content. This development gives prominence to spatial web data management. Specifically, a spatial keyword query takes a user location and user-supplied keywords as arguments and returns web objects that are spatially and textually relevant to these arguments. This paper reviews recent results by the authors that aim to achieve spatial keyword querying functionality that is easy to use, relevant to users, and can be supported efficiently. The paper covers different kinds of functionality as well as the ideas underlying their definition.

X. cao , et. all 2011
¢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-

ISSN: 2277-9655
(I2OR), Publication Impact Factor: 3.785
(ISRA), Impact Factor: 2.114

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.

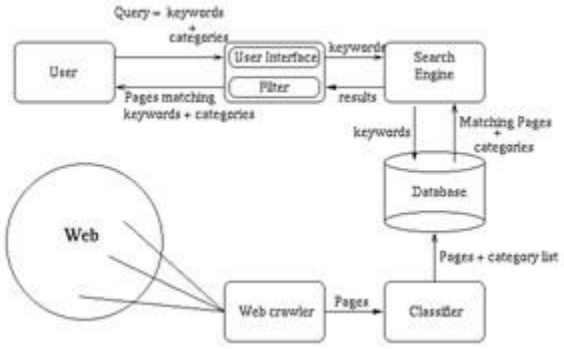
EXISTING SYSTEM

¢Spatial queries with keywords have not been extensively explored. In the past years, the community has sparked enthusiasm in studying keyword search in relational databases. ¢It is until recently that attention was diverted to multidimensional data. The best method to date for nearest neighbor search with keywords is due to Felipe et al.. They nicely integrate two well-known concepts: R-tree, a popular spatial index, and signature file, 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. On the other hand, like signature files, the IR2 -tree is able to filter a considerable portion of the objects that do not contain all the query keywords, thus significantly reducing the number of objects to be examined.

PROPOSED SYSTEM

In this paper, we design a variant of inverted index that is optimized for multidimensional points, and is thus named the spatial inverted index (SI-index). This access method successfully incorporates point coordinates into a conventional inverted index with small extra space, owing to a delicate compact storage scheme. ¢Meanwhile, an SI-index preserves the spatial locality of data points, and comes with an R-tree built on every inverted list at little space overhead. As a result, it offers two competing ways for query processing. ¢We can (sequentially) merge multiple lists very much like merging traditional inverted lists by ids. Alternatively, we can also leverage the R-trees to browse the points of all relevant lists in ascending order of their distances to the query point. As demonstrated by experiments, the SI-index significantly outperforms the IR2 -tree in query efficiency, often by a factor of orders of magnitude.

SYSTEM ARCHITECTURE



under the system architecture, the web collects information and send the pages to classifier through web crawler and this classifier classifies the pages according to different categories and send to database, this database manages multidimensional objects and provide fast access to those objects based on different selection criteria, here the user enters a keywords to search engine through the user interface and search engine gives keywords to database, look for the keywords which may be of different categories, if it finds the particular keywords then returns the results to user through filter.

ALGORITHM

Best-first search is a search algorithm which explores a graph by expanding the most promising node chosen according to a specified rule. Judea Pearl described best-first search as estimating the promise of node n by a "heuristic evaluation function which, in general, may depend on the description of n, the description of the goal, the information gathered by the search up to that point, and most important, on any extra knowledge about the problem domain."Some authors have used "best-first search" to refer specifically to a search with a heuristic that attempts to predict how close the end of a path is to a solution, so that paths which are judged to be closer to a solution are extended first.

ISSN: 2277-9655
(I2OR), Publication Impact Factor: 3.785
(ISRA), Impact Factor: 2.114

This specific type of search is called greedy best-first search.Efficient selection of the current best candidate for extension is typically implemented using a priority queue. The A* search algorithm is an example of best-first search, as is B*. Best-first algorithms are often used for path finding in combinatorial search. (Note that neither A* nor B* is a greedy best-first search as they incorporate the distance from start in addition to estimated distances to the goal.

OPEN = [initial state]
while OPEN is not empty or until a goal is found do
1. Remove the best node from OPEN, call it n.
2. If n is the goal state, backtrace path to n (through recorded parents) and return path.
3. Create n's successors.
4. Evaluate each successor, add it to OPEN, and record its parent. Done

CONCLUSIONS

We have seen plenty of applications calling for a search engine that is able to efficiently support novel forms of spatial queries that are integrated with keyword search. The existing solutions to such queries either incur prohibitive space consumption or are unable to give real time answers. In this paper, we have remedied the situation by developing an access method called the spatial inverted index (SI-index). Not only that the SI-index is fairly space economical, but also it has the ability to perform keyword-augmented nearest neighbor search in time that is at the order of dozens of milli-seconds. Furthermore, as the SI-index is based on the conventional technology of inverted index, it is readily incorporable in a commercial search engine that applies massive parallelism, implying its immediate industrial merits.

REFERENCES

[1] S. Agrawal, S. Chaudhuri, and G. Das, “Dbxplorer: A System forKeyword-Based Search over Relational Databases,” Proc. Int’l Conf. Data Eng. (ICDE), pp. 5-16, 2002.
[2] N. Beckmann, H. Kriegel, R. Schneider, and B. Seeger, “The R- tree: An Efficient and Robust Access Method for Points and Rectangles,” Proc. ACM SIGMOD Int’l Conf. Management of Data, pp. 322-331, 1990.
[3] G. Bhalotia, A. Hulgeri, C. Nakhe, S. Chakrabarti, and S.Sudarshan, “Keyword Searching and Browsing in Databases Using Banks,” Proc. Int’l Conf. Data Eng. (ICDE), pp. 431-440, 2002.
[4] X. Cao, L. Chen, G. Cong, C.S. Jensen, Q. Qu, A. Skovsgaard, D. Wu, and M.L. Yiu, “Spatial Keyword Querying,” Proc. 31st Int’l Conf. Conceptual Modeling (ER), pp. 16-29, 2012.
[5] X. Cao, G. Cong, and C.S. Jensen, “Retrieving Top-k Prestige- Based Relevant Spatial Web Objects,” Proc. VLDB Endowment, vol. 3, no. 1, pp. 373-384, 2010.
[6] X. Cao, G. Cong, C.S. Jensen, and B.C. Ooi, “Collective Spatial Keyword Querying,” Proc. ACM SIGMOD Int’l Conf. Management of Data, pp. 373-384, 2011.
[7] B. Chazelle, J. Kilian, R. Rubinfeld, and A. Tal, “The Bloomier Filter: An Efficient Data Structure for Static Support Lookup Tables,” Proc. Ann. ACM-SIAM Symp. Discrete Algorithms (SODA), pp. 30- 39, 2004.
[8] Y.-Y. Chen, T. Suel, and A. Markowetz, “Efficient Query Processing in Geographic Web Search Engines,” Proc. ACM SIGMOD Int’l Conf. Management of Data, pp. 277-288, 2006.
[9] E. Chu, A. Baid, X. Chai, A. Doan, and J. Naughton, “Combining Keyword Search and Forms for Ad Hoc Querying of Databases,” Proc. ACM SIGMOD Int’l Conf. Management of Data, 2009.
[10] G. Cong, C.S. Jensen, and D. Wu, “Efficient Retrieval of the Top-k Most Relevant Spatial Web Objects,” PVLDB, vol. 2, no. 1, pp. 337- 348, 2009.
[11] C. Faloutsos and S. Christodoulakis, “Signature Files: An Access Method for Documents and Its Analytical Performance Evaluation,” ACM Trans. Information Systems, vol. 2, no. 4, pp. 267-288, 1984.
[12] I.D. Felipe, V. Hristidis, and N. Rishe, “Keyword Search on Spatial Databases,” Proc. Int’l Conf. Data Eng. (ICDE), pp. 656-665, 2008.
[13] R. Hariharan, B. Hore, C. Li, and S. Mehrotra, “Processing Spatial- Keyword (SK) Queries in Geographic Information Retrieval (GIR) Systems,” Proc. Scientific and Statistical Database Management (SSDBM), 2007.

ISSN: 2277-9655
(I2OR), Publication Impact Factor: 3.785
(ISRA), Impact Factor: 2.114

[14] G.R. Hjaltason and H. Samet, "Distance Browsing in Spatial Databases," ACM Trans. Database Systems, vol. 24, no. 2, pp. 265-318, 1999.
[15] V. Hristidis and Y. Papakonstantinou, "Discover: Keyword Search in Relational Databases," Proc. Very Large Data Bases (VLDB), pp. 670-681, 2002.
[16] I. Kamel and C. Faloutsos, "Hilbert R-Tree: An Improved R-Tree Using Fractals," Proc. Very Large Data Bases (VLDB), pp. 500-509, 1994.
[17] J. Lu, Y. Lu, and G. Cong, "Reverse Spatial and Textual k Nearest Neighbor Search," Proc. ACM SIGMOD Int'l Conf. Management of Data, pp. 349-360, 2011.
[18] S. Stiassny, "Mathematical Analysis of Various Superimposed Coding Methods," Am. Doc., vol. 11, no. 2, pp. 155-169, 1960.
[19] J.S. Vitter, "Algorithms and Data Structures for External Memory," Foundation and Trends in Theoretical Computer Science, vol. 2, no. 4, pp. 305-474, 2006.
[20] D. Zhang, Y.M. Chee, A. Mondal, A.K.H. Tung, and M. Kitsuregawa, "Keyword Search in Spatial Databases: Towards Searching by Document," Proc. Int'l Conf. Data Eng. (ICDE), pp. 688-699, 2009
[21] Y. Zhou, X. Xie, C. Wang, Y. Gong, and W.-Y. Ma, "Hybrid Index Structures for Location-Based Web Search," Proc. Conf. Information and Knowledge Management (CIKM), pp. 155-162, 200