EFFICIENTLY PROCESSING SPATIAL AND KEYWORD QUERIES IN INDOOR VENUES

Nagamohan Thonta, Miss G.Keerthana, Sri.V.Bhaskara Murthy,
MCA Student, Assistant Professor, Associate Professor
Dept Of MCA
B.V.Raju College, Bhimavaram

ABSTRACT
Due to the growing popularity of indoor location-based services, indoor data management has received significant research attention in the past few years. However, we observe that the existing indexing and query processing techniques for the indoor space do not fully exploit the properties of the indoor space. Consequently, they provide below par performance which makes them unsuitable for large indoor venues with high query workloads. In this paper, we first propose two novel indexes called Indoor Partitioning Tree (IP-Tree) and Vivid IP-Tree (VIP-Tree) that are carefully designed by utilizing the properties of indoor venues. The proposed indexes are lightweight, have small pre-processing cost and provide near-optimal performance for shortest distance and shortest path queries. We are also the first to study spatial keyword queries in indoor venues. We propose a novel data structure called Keyword Partitioning Tree (KP-Tree) that indexes objects in an indoor partition. We propose an efficient algorithm based on VIP-Tree and KP-Trees to efficiently answer spatial keyword queries. Our extensive experimental study on real and synthetic data sets demonstrates that our proposed indexes outperform the existing solutions by several orders of magnitude.

I. INTRODUCTION

1.1 Motivation

Due to the recent breakthroughs in indoor positioning technologies (see [17], and its references), and the widespread use of smart phones, indoor location-based services (LBSs) are becoming increasingly popular [5]. Indoor LBSs can be very valuable in many different domains such as emergency services, health care, location-based marketing, asset management, and in-store navigation, to name a few. In such indoor LBSs and many others, indoor distances play a critical role in improving the service quality. For example, in an emergency, an indoor LBS can guide people to the nearby exit doors. Similarly, a passenger may want to find the shortest path to the boarding gate in an airport, a disabled person may issue a query to find accessible toilets within 100 meters in a shopping mall, or a student may issue a query to find the nearest photocopier in a university campus.

There is a huge demand for efficient and scalable spatial query processing systems for indoor location data. Unfortunately, as we explain next, the outdoor techniques provide below par performance for indoor spaces and the existing indoor techniques fail to fully utilize the unique properties of indoor venues resulting in poor performance

1.2 Limitations of Existing Techniques

1.2.1 Outdoor techniques

Techniques for outdoor LBSs cannot be directly applied for indoor LBSs due to the specific characteristics in indoor settings.

Referring to the aforementioned examples, briefly speaking, we need to not only represent the spaces (airport, shopping center) in proper data model but also manage all the indoor features (lifts, escalators, stairs) and locations of interest (boarding gates, exit doors, and shops) such that search can be conducted efficiently. Indoor spaces are characterized by indoor entities such as walls, doors, rooms, hallways, etc. Such entities constrain as well as enable indoor movements, resulting in unique indoor topologies

One possible approach for indoor data management is to first model the indoor space to a graph using existing indoor data modelling techniques [2], [19] and then applying existing graph algorithms to process spatial queries on the indoor graph. However, as we demonstrate in our experimental study, this approach lacks efficiency and scalability – the state-of-the-art outdoor techniques ROAD [16] and G-tree [37] may take more than one second to answer a single shortest distance query. This is mainly because the existing outdoor techniques rely on the properties of road networks and fail to exploit the properties specific to indoor space. For example, the indoor graphs have a much higher average out degree (up to 400) as compared to the road networks that have average out-degree of 2 to 4. Consequently, the size of the indoor graphs is much larger relative to the actual area it covers. For example, we use the buildings in Clayton campus of Monash University as a data set in our experiments and the corresponding indoor graph has around 6.7 million edges and around 41, 000 vertices. Compared to this, the road network corresponding to California and Nevada states consists of around 4.6 million edges and 1.9 million vertices [4]. Our experimental study shows that these outdoor techniques take around 1 second to answer a single indoor shortest distance query, in contrast, our specialized techniques that carefully exploit the properties of indoor space can process the same query in around 10 microseconds.

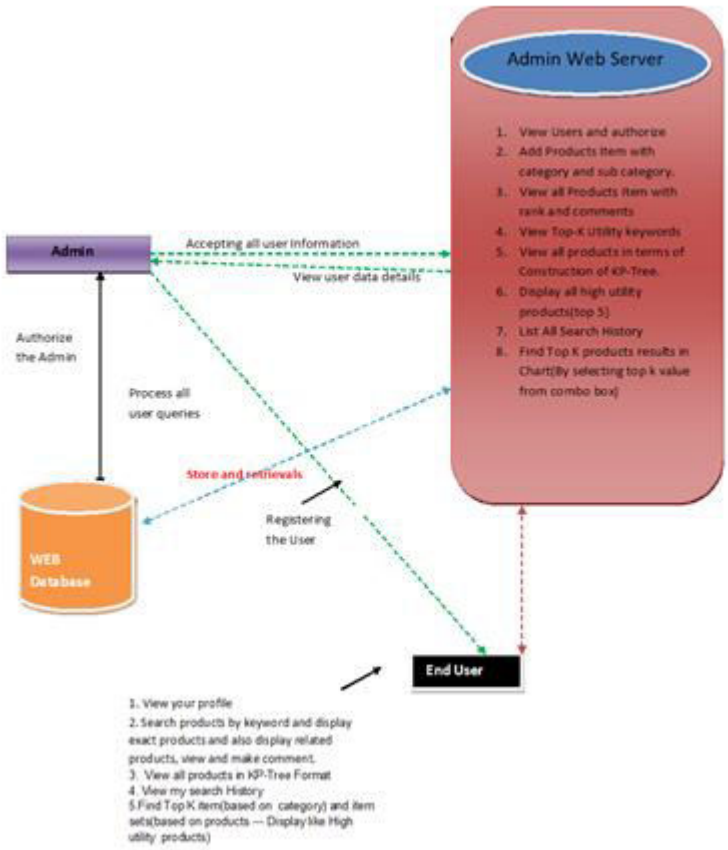
1.2.2 Indoor techniques

Adopting the idea of mapping the indoor space to a graph and applying graph algorithms, existing techniques use door-to-door graph [33] and/or accessibility base graph [19] to process various indoor spatial queries. Door-to door (D2D) graph [33]. In a D2D graph, each door in the indoor space is represented as a graph vertex. A weighted edge is created between two doors di and dj if they are connected to the same indoor partition (e.g., room, hallway), where the edge weight is the indoor distance between the two doors. Fig. 1 shows an example of an indoor space that contains 17 indoor partitions(P1 to P17) and 20 doors (d1 to d20). The corresponding D2D graph is shown in Fig. 2a where edge weights are not displayed for simplicity. The doors d1 to d5 are all connected to each other by edges because they are associated to the same partition P1. Accessibility base (AB) graph [19]. In an AB graph, each indoor partition is mapped to a graph vertex, and each door is represented as an edge between the two partitions it connects. Fig. 2b shows the AB graph for the indoor space shown in Fig. 1. Since partitions P1 and P2 are connected by door d4, an edge labeled d4 is created between P1 and P2 in the AB graph. Partitions P1 and P3 are connected by two doors d2 and d3, and thus two labeled edges are created between P1 and P3. Although an AB graph captures the connectivity information, it does not support indoor distances. Distance matrix (DM) [19]. A distance matrix can also be used to facilitate shortest distance/path queries. A distance matrix stores the distances between all pairs of doors in the indoor space. Although this allows optimally retrieving the distance

between any two doors (i.e., in O(1)), it requires huge pre processing cost and quadratic storage which makes it unattractive for large indoor venues. Furthermore, the distance matrix cannot be used to answer k nearest neighbors (kNN) and range queries without utilizing other structures such as AB graph.

The existing techniques apply graph algorithms on a D2D graph and/or AB graph to answer spatial queries. For instance, the state-of-the-art indoor spatial query processing technique [19] computes the shortest distance between a source point *s* and a target point *t* (shown as stars in Fig. 1) using Dijkstra's like expansion on a D2D graph or AB-graph. Although severa optimizations are employed in [19], these techniques essentially rely on a Dijsktra's like expansion over the entire graph which is computationally quite expensive. Consequently, the state-of-the art indoor query processing takes more than 100 seconds to answer a single shortest path query on the Clayton campus data set used in our experiments (whereas our proposed technique processes the same query in around 10 microseconds).

II. SYSTEM ARCHITECTURE



III. EXISTING SYSTEM

Indoor Data Modelling Data modelling for indoor space is fundamental for querying indoor space. In [15], a 3D model is proposed for indoor space but it fails to support indoor distance computations. CityGML [1] and IndoorGML [2] are XML based methods to model and exchange the indoor space data. As stated in Section 1, the distance-aware model [19] introduces an extended graph based on an accessibility base graph and D2D graph that enables

indoor distance computations between two indoor positions.

Spatial Queries in Indoor Space. Several indoor spatial queries such as shortest distance queries, shortest path queries, kNN queries and range queries have been studied under various settings [20], [31], [32], [35]. The most notable techniques [19], [33] have already been discussed in Section 1. Existing research also studies various other indoor queries such as trip planning queries and multi-criteria route planning queries [23], [24]. Indexing and querying moving indoor objects have also been studied in the past [11], [30].

Spatial Queries in Outdoor Space. Query processing in Euclidean space and road networks [6], [10], [26], [34] is very well studied. Since an indoor space can be converted into a D2D graph, various techniques [27], [36], [37] in spatial road networks can also be applied. G-tree [36], [37] is the state-of-the art technique for processing a variety of spatial queries on road networks. Although our proposed indexes, IP-Tree and VIP-Tree, are inspired by G-tree, there are some fundamental differences.

Specifically, G-tree uses an existing multilevel graph partitioning algorithm [14] for graph decomposition whereas we design a new algorithm that carefully exploits the properties of the indoor space to minimize the total number of access doors. Also, the smaller number of access doors in our nodes allows us to use materialization in the VIP-tree which proves to be a much more efficient strategy but is not feasible for G-tree. Furthermore, our algorithms to process shortest path queries, range queries, kNN queries and spatial keyword queries are also entirely different.

Spatial Keyword Queries in Outdoor Space. Spatial keyword queries have been extensively studied in Euclidean space [18], [22], [29]. For example, the Inverted R-tree [38] is proposed that creates, for each keyword t, an R-tree based on the objects that contains the keyword t. Inverted R-trees are efficient when the number of query keywords is small because the query needs to process only a few R-trees. Information retrieval R-tree (IR2-tree) [9] aims to address this problem by utilizing signatures that summarize the keywords contained in the descendent entries of a node. Information R-tree (IR-tree) [8], [28] utilizes inverted files for each node that maintains the keywords information in the node. WIR-tree [29] is similar to IR-tree, but it partitions the objects according to keyword frequencies instead of spatial locations. A detailed experimental evaluation comparing different spatial keyword approaches is presented in [7].

Disadvantages

- In the existing work, the system is very less effective due to lack of Construction of Spatial Query processing tress.
- The system is not minimizing the total amount of misinformation between users and spatial queries.

IV. PROPOSED SYSTEM

To handle fundamental spatial queries, we propose two novel indoor indexes called Indoor Partitioning tree (IP-Tree) and Vivid IP-Tree (VIP-Tree) that optimize the indexing by exploiting the properties of indoor spaces. The basic observation is that the shortest path from a point in one indoor region to a point in another region passes through a small subset of doors (called access doors). For example, the shortest path between two points located on different

floors of a building must pass one of the stairs/lifts connecting the two floors. The proposed indexes take into account this observation in their design and have the following attractive features. Near-optimal efficiency. Our experimental study on real and synthetic data sets demonstrates that IP-Tree and VIP-Tree outperform the state-of-the-art techniques for indoor space [19] and road networks [16], [37] by several orders of magnitude. In comparison
with the distance matrix, that allows constant time retrieval of distance between any two doors at the cost of expensive pre computing and quadratic storage, our VIP-Tree also achieves comparable (and near-optimal) performance for shortest distance
and path queries.
Low indexing cost. VIP-Tree and IP-Tree have small construction cost and low storage requirement. For example, for the largest data set used in our experiments that consists of around 83, 000 rooms (around 13.4 million edges), VIP-Tree and IP-Tree consume around 600 MB and can be constructed in less than 2 minutes. In contrast, it took almost 14 hours to construct the distance matrix for a much smaller building consisting of around 2, 700 rooms (around 110, 000 edges).
Low theoretical complexities. Our proposed indexes do not only provide practical efficiency but also have low storage and computational complexities. Table 1 compares the storage complexity and shortest distance/path computation cost of our proposed approach with the distance matrix which has near-optimal computational complexity. For the data sets used in our experiments, the average values of _ and *f* are less than 4. For our proposed trees, *M* is the number of leaf nodes which is bounded by the number of doors *D*. Note that VIP-Tree has a significantly low storage cost compared to the distance matrix but has the same computational complexity. A detailed theoretical analysis is provided in the conference version [25] of this paper.

Advantages

- The system is more effective in Spatial Query Processing by Keyword-Partitioning Tree (KP-Tree).
- The system is more effective due to presence of Shortest Distance Queries and k Nearest Neighbor (kNN) techniques.

V. IMPLEMENTATION

- **Admin**

In this module, the Admin has to login by using valid user name and password. After login successful he can perform some operations such as view and authorize users, Adding Categories Sub-Categories, Adding Product Posts for by Selecting Category and Sub-Categories, Viewing Top- K Utility Keywords, Viewing all Products in terms of Construction of KP-Tree, Viewing all High Utility Products, Viewing All User Search History and Finding Top K Products Results in Chart.

Viewing and Authorizing Users

In this module, the admin views all users details and authorize them for login permission. User Details such as User Name, Address, Email Id and Mobile Number.

Add Categories, Sub-Categories and Product Posts

In this module, the admin adds Categories, Sub-Categories and Product Posts. The Product Posts are added by selecting particular category and Sub-Category and Product Details such as, Product Title, Price, Description and Image of that Product.

View all Products with Ranks and Comments

In this module, the admin can see all the uploaded products with product ranks and

comments. The Product details contain Product title, description, price, and image.
The Comment details include commented user, their comment and the date of comment.

View Top-K Utility Keywords

In this module, the all keywords which are all used very frequently and less frequently will be displayed in a Rank (No. of times used) in a Top-K Order.

View all Products in terms of Construction of KP-Tree

In this, the admin can see all the products in a Tree Format. In this Tree, Firstly (On Top) Category then Sub-Category and lastly (at Bottom) Product Posts will be displayed.

View all high Utility Products

In this, the top 5 Mining products will be displayed along with their details based on ranks. The Product details contain Product title, description, price, and image.

Find Top K Products Results in Chart

In this, the top K number of products will be displayed based on top rank of products in a chart based on the value selected from the combo box.

- **User**

In this module, there are n numbers of users are present. User should register before performing any operations. Once user registers, their details will be stored to the database. After registration successful, he has to login by using authorized user name and password. Once Login is successful user can perform some operations like viewing their profile details, searching for products based on product description, searching products and viewing them in a KP-Tree Format, Viewing Own Search History and Finding Top K Product Item Sets by selecting category and Top K Value.

Viewing Profile Details

In this module, the user can see their own profile details, such as their address, email, mobile number, profile Image.

Search Products

In this, the user search for products based on product description. The matched results will be displayed in two ways: Exact Matched and Related Products. Related Products are the products which are not exactly matched for user entered keyword and they are belong to the same category of exactly matched products category.

Search and View Products in KP-Tree Format

In this, the user search for products based on product description and the matched products will display in a KP-Tree Format. In a Tree there would be three layers. In a first top layer the Category name and in a second layer the Sub-Category Name and in a last layer the Product Title would be shown and user can see the product details by clicking on product name.

Finding Top K Item Sets

In this, the user finds Product Items Sets based on Category and Top k Value. The Result is the top K number of products from the Selected Category.

VI. CONCLUSION

In this paper, we propose two novel indoor indexes, IP-Tree and VIP-Tree, for efficiently processing shortest distance queries. IP-Tree and VIP-Tree have low storage requirement, small preprocessing cost and are highly efficient. Our extensive experimental study on real and synthetic data sets demonstrates that the proposed indexes outperform the existing techniques by several orders of magnitude. For spatial keyword queries, we extended VIP Tree by embedding keyword information on each node. We also proposed a partition-specific index called KP-Tree that indexes the objects for each indoor partition. The experimental studies demonstrate that our proposed indexes significantly outperform the competitors. An important direction for future work is to study spatial keyword queries considering similarity to the query

keywords as well as synonyms and product categories**.**

REFERENCES

[1] http://www.citygml.org/.
[2]http://www.opengeospatial.org/projects/groups/indoorgmlswg.
[3] T. Abeywickrama, M. A. Cheema, and A. Khan. K-SPIN: Efficiently processing spatial keyword queries on road networks. IEEE Transactions on Knowledge and Data Engineering, 2019.
[4] T. Abeywickrama, M. A. Cheema, and D. Taniar. k-nearest neighbors on road networks: a journey in experimentation and in-memory implementation. PVLDB, 2016.
[5] M. A. Cheema. Indoor location-based services: challenges and opportunities. SIGSPATIAL Special, 10(2):10–17, 2018.
[6] M. A. Cheema, W. Zhang, X. Lin, Y. Zhang, and X. Li. Continuous reverse k nearest neighbors queries in euclidean space and in spatial networks. VLDB J., 21(1):69–95, 2012.
[7] L. Chen, G. Cong, C. S. Jensen, and D. Wu. Spatial keyword query processing: An experimental evaluation. PVLDB.
[8] G. Cong, C. S. Jensen, and D. Wu. Efficient retrieval of the top-k most relevant spatial web objects. PVLDB, 2(1):337–348, 2009.
[9] I. D. Felipe, V. Hristidis, and N. Rishe. Keyword search on spatial databases. In ICDE, pages 656–665, 2008.
[10] A. Hidayat, S. Yang, M. A. Cheema, and D. Taniar. Reverse approximate nearest neighbor queries. IEEE Trans. Knowl. Data Eng., 30(2):339–352, 2018.
[11] C. S. Jensen, H. Lu, and B. Yang. Indexing the trajectories of moving objects in symbolic indoorspace. In SSTD, 2009.
[12] M. Jiang, A. W. Fu, and R. C. Wong. Exact top-k nearest keyword search in large networks. In ACM SIGMOD, pages 393–404, 2015.
[13] G. Karypis and V. Kumar. Analysis of multilevel graph partitioning. In Supercomputing, 1995.
[14] G. Karypis and V. Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. SIAM J. Sci. Comput., 1998.
[15] J. Lee. A spatial access-oriented implementation of a 3-d gis topological data model for urban entities. GeoInformatica, 2004.