THE EFFICIENT GEOSPATIAL INDEX STRUCTURE FOR SPATIAL KEYWORD QUERY WITH MYANMAR LANGUAGE ON MOBILE DEVICES

MYAT THIRI KHINE

UNIVERSITY OF COMPUTER STUDIES, YANGON

NOVEMBER, 2019

The Efficient Geospatial Index Structure for Spatial Keyword Query with Myanmar Language on Mobile Devices

Myat Thiri Khine

University of Computer Studies, Yangon

A thesis submitted to the University of Computer Studies, Yangon in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

November, 2019

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research and has not been submitted for a higher degree to any other University or Institution.

1.11.2019	
…………………………………	……………………………………………
Date	Myat Thiri Khine

ACKNOWLEDGEMENTS

First of all, I would like to thank the Minister, the Ministry of Education for full facilities support during the Ph.D Course at the University of Computer Studies, Yangon.

Second, I would like to express very special thanks to Dr. Mie Mie Thet Thwin, the Rector, the University of Computer Studies, Yangon, for allowing me to develop this thesis and giving me general guidance during a period of my study.

I would like to say a very big thanks to my supervisor, Dr. Myint Myint Sein, Professor, University of Computer Studies, Yangon, for her excellent guidance, caring, patience, and providing me with excellent ideas for doing research. I have been extremely lucky to have a supervisor who cared so much about my work. Without her guidance and constant feedback, this dissertation would not have been achievable. I will always remember her for being a mentor to me.

I would like to extend my special appreciation to Dr. Mi Mi Nge, Managing Director, Realistic Infotech Group Co.Ltd. for the useful comments, sharing knowledge, giving advice, and insight which are invaluable to me.

I would also like to extend my special appreciation and thanks to Dr. Khine Moe Nwe, Course coordinator of the Ph.D. 9th batch, University of Computer Studies, Yangon, for her useful comments, advice, and insight which are invaluable to me.

I would like to express my respectful gratitude to Daw Aye Aye Khine, Associate Professor, Head of English Department, for her valuable supports from the language point of view and pointed out the correct usage in my dissertation.

I also thank my friends from Ph.D. 9th batch for providing support and friendship that I needed.

I am very much indebted to my family for always believing in me, for their endless love and support**.** They are always supporting and encouraging me during the years of my Ph.D study.

ABSTRACT

Nowadays, Location-based services (LBS) are widely used in numerous areas such as positioning systems, tourist information, social and business areas. Many researchers try to develop the LBS application to fulfill the user requirements. The usage of location-based services systems is increasing around the world including Myanmar. Tourists in Myanmar are now increasing day by day. Location-based services systems in Myanmar still have the requirements compared with the other developed countries to provide both local users and foreign users with the geo-information of Myanmar like nearest geo-information, geo-information around the user's desired point. Moreover, most of the location-based services applications are processed in online. Sometimes, it cannot access the information when the internet connection is weak or cannot be available.

This research is implemented for retrieving the geo-information on the mobile devices using both Myanmar and English language to provide both local and foreign visitors although it is considered for mainly focus on the Myanmar language. Therefore, it can support not only foreign users but also local users who are difficult for using the English version of the location-based services applications. For fetching the geo-information, the nearest keyword query and range keyword query can be operated not only from the user's current location but also from the user's desired place. These queries are processed based on the given location and given keyword. In this research, user can pre-find the desired geo-location due to be able to process these queries from the user's desired location. In addition, the geo-information can be obtained by using the keyword search. In this keyword query, the geo-information is not only the location information with their related text information but also the distance information from this geo-information to the user's current location. Moreover, it also gives the geo-information which has the minimum distance to current location among the resulted geo-information. These spatial keyword queries can be processed whether the internet connection is available or not.

The geo-information contains not only the geo-location (latitude and longitude) but also the related text information. The data of the geo-information can be huge so that it can be time-consuming for searching the geo-information. The geo-spatial index structure is one of the key factors to perform the fast spatial keyword query processing because the mobile devices possess the limited memory and a low computational

capacity than the personal computer. Thus, it is required to build the geospatial index structure to obtain the efficient spatial keyword query processing.

In this research, a new index structure is constructed to get the fast access the geo-data for Myanmar keyword query. In addition, this index structure is also considered for extracting the geo-information with English keyword Query. Hilbert curve, B-tree and inverted file are utilized for constructing this index structure to get the efficient searching on both geo-location data and textual data. Due to B-tree works on the one-dimensional data, Hilbert curve is used to map the two-dimensional data (geo-location data) to one-dimensional data. Then, B-tree is built according to the Hilbert value and inverted file is embedded to the B-tree node. The proposed index structures are separately constructed depend on the services such as school, hotel and so on. Geo-spatial keyword queries processing in this research are operated with this proposed index structure. It can reduce the searching time by using this proposed index structure. This research focuses on the Yangon Region. Mapbox vector map is used to display the geo-information in online and offline. Myanmar 3 font is used in this research. This location-based service application can be used in android mobile version 4.4 and above.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF EQUATIONS

CHAPTER 1
INTRODUCTION

The applications of location-based services (LBS) provide the user with day-to-day useful information like city routes, point of interest (POI), public transport schedules, etc. The varieties of LBS systems are evolved for navigation, positioning, social, commercial and location-based games. Researchers have been implementing the LBS application for the users to obtain the geo-location information of the specific region. Therefore, Geospatial information is essential not only for the business sector but also in our daily lives.

Geographical Information system (GIS) plays a vital role in the success of location-based services. It is a set of tools for collecting, storing, retrieving, transforming and displaying geospatial data from the real world [46]. It also supports the tools to supply and manage the base map data such as buildings, streets and also used to manage the point of interest data such as locations of the restaurant, bank, etc [73]. It can be generally defined to be an information system which operates the geographic data. Geographical data is the most vital component of location-based service. Geographic data possesses the specialty of having the component of location on the earth, specified either explicitly by coordinates or implicitly by another georeferenced object. This component is called as spatial data [75]. Geospatial information systems are the specialized software systems which can analyze and manage spatial data [1]. The functionality of the location-based services improves as the outcome of achieving the information by geographical information system [73].

Geo-spatial information in the geospatial database can be the geo-location and the text description of the location object. To obtain the information from the geospatial database, efficient spatial keyword search is required to implement. Interests of spatial keyword queries are rising in the research community to obtain the efficient and effective query processing on both spatial and textual description. Various kinds of Spatial keyword processing [5,9,29,71,81] have been proposed to retrieve the required geo-information based on the location and the given keyword. There are several spatial keyword searches such as nearest keyword search [70], range keyword search, approximate keyword search [84] [17] and so on. To obtain a faster access in spatial keyword query processing, geospatial index structure is one of the solutions.

1.1 Location-based services (LBS)

In a general form, the location-based services can be defined as services that use the capability to dynamically decide and transmit the user's location within a mobile network by using their mobile devices. From the mobile users' point of view, the location-based services are normally services that access with or offered by their mobile devices [75].

Location-based services use the information about users' location to determine the services like nearest interested point from a location [56]. It indicates the user's position based on a device and has the ability for providing the specific and relevant information according to a given spatial location that associated with a physical point or region relative to the earth's surface [46].

Central component of the location-based services is the content. Content is the data which are utilized by the location-based service to support location-based information or functionality for the users. In the context of location-based services, the content can be separated into two categories: geographic-based data and location-based information. Geographic-based data consists of the digital map data (eg. Street network) and location-based information is any information that can be related to a specific location (eg. restaurant information). Geographic data in a form of digital map is necessary for information presentation as a reference material. Geographic data is needed by location-based services for spatial queries and analysis functions [75].

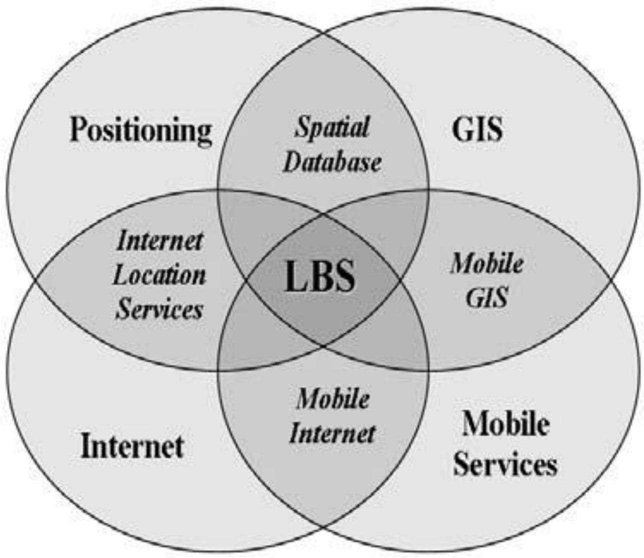


Figure 1.1 Technological Convergence to Generate Location-based Service (LBS)

The integration of the location-based services can be seen in Figure 1.1 [56]. It is created from the integration of mobile services, location aware technologies with Internet and GIS.

1.2 Geospatial Data in Geographical Information System

Geographic information system is an information system which is utilized to capture, collect, retrieve, manipulate, analyze and display the geo-related data or geospatial data. Geo-reference data is comprised of spatial (eg. geometry, coordinates) and non-spatial data (eg. name of city) [39]. Spatial data is represented in multidimensional space. It includes simple points, lines, polygons and complex objects composed of sub-objects. For example, the region of Myanmar contains towns (i.e. points), cities (i.e. regions), and roads (i.e. multiple connected lines, or line strings). Each feature has a location in space. In addition, all features are spatially related to each other. Spatial data supports the location information of the objects and non-spatial data which also known as attribute data represents the characteristics of the objects. Integration of these two data is known as geo-spatial data. Both these two data are necessary for GIS processing [8].

Geographical data can represent in two primary types of data – raster data and vector data. The raster and vector data can be seen in Figure 1.2.

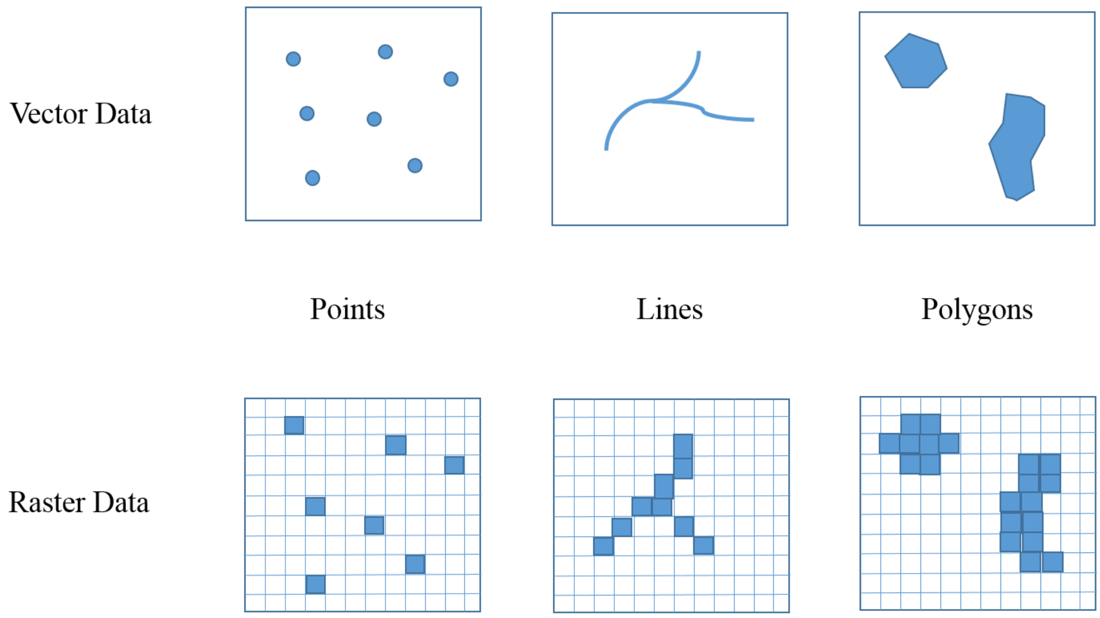


Figure 1.2 Example of Vector and Raster Data for Points, Lines and Polygons

Raster data: Raster data is represented as a matrix of grid cells or pixels. It stores the spatial data in discrete cells arranged into rows and columns. The resolution of the raster data corresponds to the grid or pixel size. Data capture in Raster can be imageries from an aerial photo or satellite. Raster data takes more storage space than the vector data.

Vector data: Vector format in geographical information is generally represented the geo-spatial data as a set of coordinates (geographical coordinates-latitude and longitude). It generally displays the geo-spatial information as three features – points, lines, and polygons. Point is used to represent a single location such as cites location, restaurant location. Line contains a series of coordinate points which is used to describe the road, river and so on. An area features like countries and cities is represented by the polygon which is determined by a set of connected lines enclosing the region [16].

1.3 Important of Spatial Indexing

The intention of the index is to obtain a quick access to the spatial data that matches the spatial and thematic components of a query. Indexing structures which support to retrieve the geo-data containing specific terms are called inverted lists, also referred to as an inverted file or inverted index [45]. Geographic Information retrieval systems often use inverted index techniques, but they are combined with spatial indexing methods to determine which terms relate to specific regions of space. Because speed of access is a major concern, it is important that the index provides access to all the main data items that are required to retrieve geo-information.

Spatial Information retrieval is generally related to spatial queries which refer to geographic location, feature of objects, interesting object. The location-based information in GIS database can contain spatial information and text descriptions. The queries that request the location-based information also include spatial and text component [29]. In the searching process, it is needed to index both the spatial and textual information. The data types of spatial and text are different and location index rely on the two-dimensional points. Therefore, the efficient geo-spatial index structure is essential for handling both spatial and text data at the same time [37].

Spatial index is the common approach to efficiently access the spatial object in spatial database based on the given keyword and location. Integrating the spatial and textual indexing methods is known as spatial keyword or spatio-textual indexing to process on both textual and geo-spatial data. It can provide the various spatial operation like range query, nearest query to obtain the efficient processing. The need for spatial indexing is one of the main problems in geographic data management.

1.4 Introduction to Myanmar Language

The nature of Myanmar language is complex. A Myanmar word is a sequence of characters. It contains various types of characters such as consonants, medials, vowels, tones, etc. Myanmar sentences do not have white space to specify words boundaries and are written from left to right.

Myanmar characters can be categorized into three groups, consonants, medials and vowels. The characters of Myanmar contain 33 basic consonants, four basic medials, and 12 basic vowels. Example of consonants are 'က' (k), 'ခ' (kh), 'ဂ' (g) and 'င'(ng). The consonants are used as the base characters in Myanmar word. Example of vowels are 'အ' (a.), 'အာ' (a), 'အိ' (i.), 'အီ' (i), 'အု' (u.) and 'အူ' (u). Myanmar words consist of one or more syllables which can have one or more characters. Syllables of Myanmar are fundamentally formed by combining the consonant and vowel. For example, the syllable 'ခို' (pigeon) is formed by combining the consonant 'ခ'(kh) and the vowels 'အို'(ou). So, the combination of 'ခို'(pigeon) is ခ + အို = ခို. The four basic medials are 'ျ' (y), 'ွ' (v), 'ြ' (r) and 'ှ'(h). The medials are written in the order y or r – v – h. For example, 'ကြွ'(krva), 'လွှ'(lvha) and 'မျှ'(myha). Myanmar text can contain other character like numerals, special characters.

The typing order of the Myanmar characters may vary (pigeon, ခို = ခ + ိ + ု (or) ခ + ု + ိ). The sequence of characters must have the same order to match the word. In Myanmar 3 Unicode, the sequence of the character needs to be the correct order and the typing order of the character cannot be changed. For example, the sequences of လှေ (boat) syllable in Myanmar 3 Unicode is လ + ှ + ေ (101C 103E 1031) [30]. The sequence of the Myanmar characters is also important for matching Myanmar word.

1.5 Motivation of Research

Nowadays, travelling is increasing for various reasons such as for relaxing, for business, for education. The usage of location-based service system is also increasing to search the location information about the unfamiliar region. There are several location-based services applications in the market like Google Maps and Maps.me. Google Map is one of the most popular location-based applications.

Most of the location-based applications can be utilized in online. Sometimes, the traveler faces the difficulties in searching the desired location while the internet connection is weak or not available. Many location-based services applications using in offline can be seen in commercial areas. But, it still has the requirements to fulfill the user's need in offline. Most of the applications can only show the offline map and cannot support the searching process. If it supports for searching process, it mostly supports the near-by query. Thus, it is required to implement more spatial query to provide the users in addition to the near-by query.

Moreover, it is difficult for some of the Myanmar citizens who are poor in using the application with English version because most of the location-based services are implemented with English language. Therefore, it is necessary to develop the location-based application for providing not only the foreign people but also the local people.

For the users to obtain the geo-information anywhere and anytime, location-based applications are more suitable to implement on the mobile devices than the web services. The geo-objects produce a huge amount of data and this data cannot easily be stored on the mobile devices. Moreover, the data amount becomes huge, and a sequential scan of every record is too time-consuming because of I/O operations. Spatial index is one of the problems in the mobile devices because the mobile devices own a limited memory and a low computational capacity than the Personal Computer (PC). In addition, the geo-information includes geolocation data (latitude and longitude coordinates) and their respective text description. Therefore, the construction of a new geospatial index structure is required to handle the geo-information for processing the given spatial keyword query.

1.6 Objective of Research

The main purpose of this research is building the geo-spatial index structure in order to retrieve the geo-information using both Myanmar and English keywords on the mobile devices. The next purpose is to efficiently process the spatial keyword queries for both two languages. The last one is to provide the mobile user for obtaining the functionalities of geographical information system not only in online mode but also in offline mode.

Other objectives of this research are:

- To implement the location-based services using both Myanmar and English keyword for the tourists as well as the local people
- To obtain the geo-information of the nearest keyword query from the current location or user's desired location
- To retrieve the geo-information of the range keyword query from the current location or user's desired location
- To fetch the geo-information based on the keyword query and to give the geo-information which is nearest to current location among the resulted geo-objects
- To perform the spatial keyword query processing by using the proposed index structure
- To display the resulted geo-information in online and offline
- To provide the user with the geo-information anywhere and anytime
- To forecast the travel location by pre-searching the geo-information of the nearest and range query from the user's desired location
- To reduce the spatial query processing time

1.7 Contribution of the Research

This research is to develop the location-based service application for obtaining the geo-information with two languages, Myanmar and English, on the mobile devices in offline and online. The main contribution of this research is to construct a new geo-spatial index structure for quickly accessing the geo-information based on the given geo-spatial keyword queries on the mobile devices. It is intended to process the spatial keyword query on both Myanmar and English language. The proposed index structure is constructed using the Hilbert curve, B-tree and inverted file.

Before constructing the proposed index structure, one of the important processes is the preparation of the geospatial database. To obtain the required geo-spatial database, the geo-information is collected with their coordinates and their related information. The collected geo-information is obtained with English version so that these collected geo-information are translated into Myanmar language.

Then, the spatial keyword query processing is taken place by using the proposed index structure to reduce the searching time. The spatial keyword queries in this research are nearest keyword query, range keyword query, and keyword query. For nearest keyword query and range keyword query, the geo-information is retrieved from the user's current location or the user's desired location. In keyword query, it not only retrieves the geo-information depend on the given keyword but also gives the nearest geo-information from these results geo-object to user's current location. The last process is to place the retrieving geo-information on the offline map or online map. The vector map is used to represent the resulted geo-information on mobile devices instead of raster map because raster data is not only based on the specific projection but also roomy.

1.8 Organization of the Research

This research is comprised of six chapters. Chapter 1 introduces LBS and GIS. Then, it also presents the important of Geo-spatial data in LBS and GIS. The geo-spatial data types in GIS are also discussed. Besides, it discusses Why geo-spatial indexing is necessary and then expresses the motivation, objective and contribution of the research. In addition, it introduces Myanmar language.

Chapter 2 reviews the literature to the spatial index structure and its indexing scheme for the spatial and textual. Then, it describes the related work of the combine index structure.

Chapter 3 mentions the types of index structure and background theory of the proposed index structure. Then, it presents the construction of the proposed geo-spatial index structure. This proposed index structure is explained in detail with its architecture and algorithm.

Chapter 4 discusses the proposed system architecture and the process of this system step by step. Moreover, it also presents the procedure of spatial keyword query using the proposed index structure.

Chapter 5 represents the detailed explanation of the system implementation and experimental results according to the spatial keyword query processing and construction of the proposed index structure.

Chapter 6 concludes with the research work by expressing the outcomes. Then, the limitation of the system and future work are also discussed in this chapter.

CHAPTER 2
LITERATURE REVIEW

Nowadays, the GIS applications are an extension to the Mobile GIS applications. Both Mobile GIS and LBS are important parts of the geospatial information and communication technology. In addition, spatial databases are becoming popular and play a vital role in many application areas like geographical information systems. Many researchers proposed various kinds of index structures to access the spatial database for processing the spatial keyword query. This chapter introduces and discusses Mobile GIS and its architectures. Then, the spatial database in geographical information system, spatial query processing, spatial indexing and related work of researchers are presented.

2.1 Mobile GIS

The mobile GIS contains a wide range of different applications such as proximity analysis, spatial query, navigation, and many others. Its applications are related to geography and it is necessary to access the geo-databases remotely or locally. For communications features, the mobile GIS can be offline (standalone mode) or online (connected mode). It has the ability for delivering the geo-spatial data to the mobile user anywhere and anytime. The mobile user can access its location and may connect to the internet or other devices/networks or in offline mode.

Mobile GIS is the accessing, utilizing, and storing of geographic data through mobile devices [85]. Mobile GIS application has two major areas: Field Based GIS and Location Based Services. Field Based GIS focuses on the collection of GIS data, update and validation in the field, such as a new point data is inserted or the attribute tables is changed on an existing GIS dataset. Location Based Services focuses on business-oriented location management functions, such as finding a specific location, street routing, navigation, tracking a vehicle, etc [34].

The criteria of the mobile GIS problem are:

(1) geodatabases are fully-fledged and available (remotely or locally).

(2) the mobile device is aware of its position.

(3) geographical functions are obtainable on mobile devices [72].

Both Mobile GIS and location-based services support a powerful interaction with other geospatial and Information technology to provide up-to-date technology solutions for the public [1].

2.2 Architectures of Mobile GIS

Mobile GIS is an expansion of Web GIS. It has various restrictions because of the limited ability of both hand-held hardware and the network data transfer speeds. The variety of mobile system platforms and the limitation in processing power are some examples of hand-held challenges. Several architectures have been proposed for the implementation of Mobile GIS [56] [57].

Some of the main architectures of Mobile GIS are as follows:

(1) **Stand-alone client architecture**: In this architecture, the application and geo-spatial data are entirely existed on the mobile devices (see in Figure 2.1). It has some limitations in this architecture. The amount of geo-spatial data that application can support depends on the hardware resources of the mobile devices and the communication with other systems is lack.

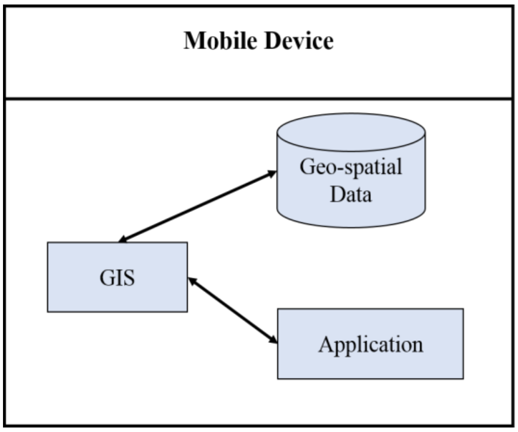


Figure 2.1 Stand-alone Client Architecture

(2) **Client-Server architecture**: In this architecture, geo-spatial data is stored in the separate computer and it provides the client by GIS server software. The GIS applications are still in the mobile devices. The client-server architecture is shown in Figure 2.2. The geo-spatial data is restricted by the resources of a server. In addition, numerous applications can concurrently access the server. If the connection to the server cannot be established, the GIS applications are useless.

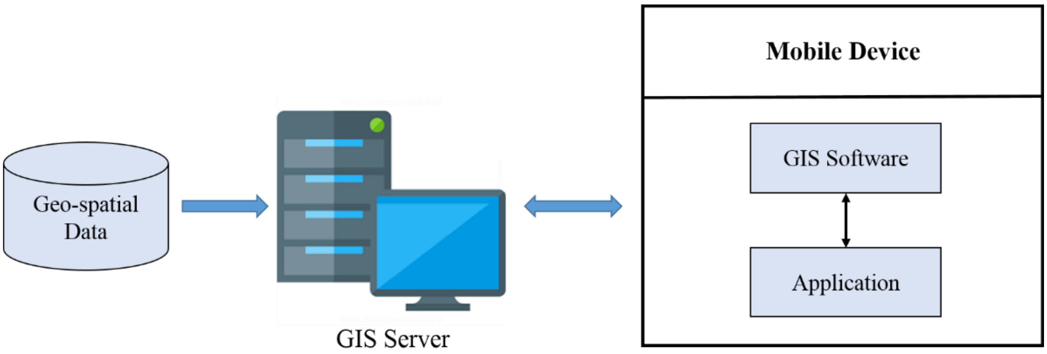


Figure 2.2 Client-server Architecture

(3) **Services architecture**: GIS server can be viewed as a web services and other web services are allowed to be part of the application. The mobile devices can communicate with each other because the web services utilize similar communications protocol. Moreover, the web services can also communicate with each other. This type of architecture is unsuitable for cooperation in remote area where the connectivity to servers is not available. Figure 2.3 shows the service architecture.

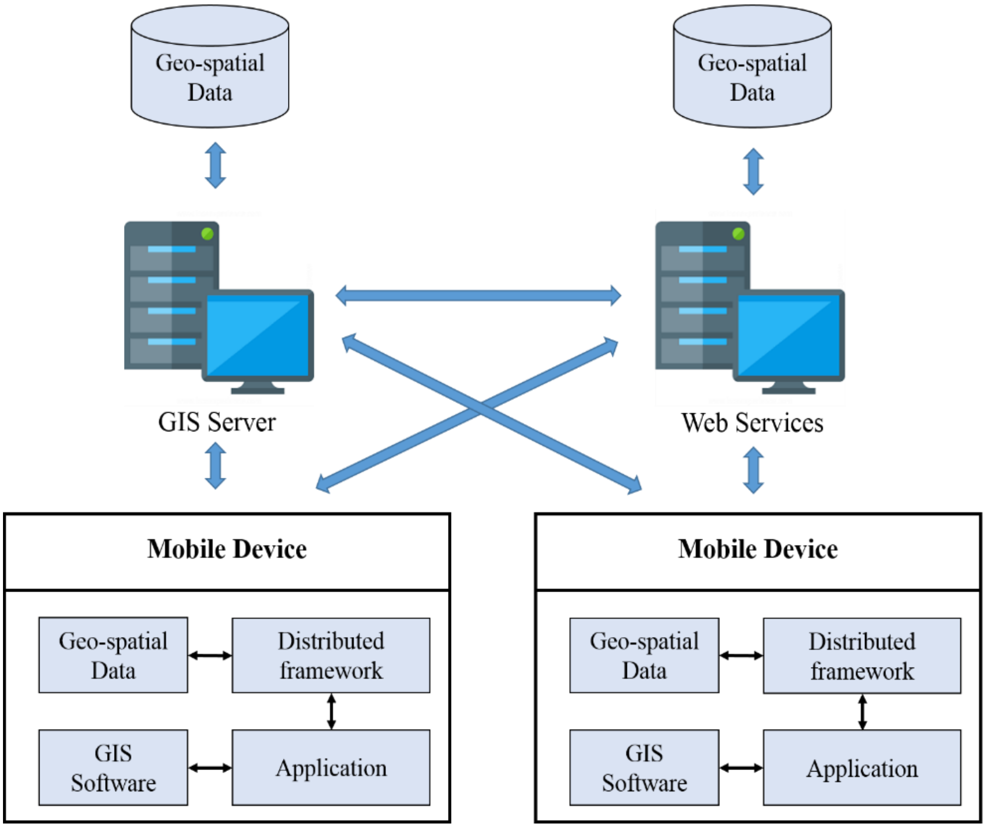


Figure 2.3 Services Architecture

(4) **Peer-to-Peer architecture:** A server is not necessary to store the geo-spatial data. Each mobile device must store the geo-data (see in Figure 2.4). So, it also has the restriction as same as the Stand-alone client architecture. Therefore, each mobile device stores a subset of the geo-data for permitting the more data storage throughout the application. If the mobile device requires the geo-data, it depends on the resource management of its distributed framework to realize that the geo-data exists locally or not. If it does not exist, the distributed framework must understand how to access that geo-data on the other mobile devices.

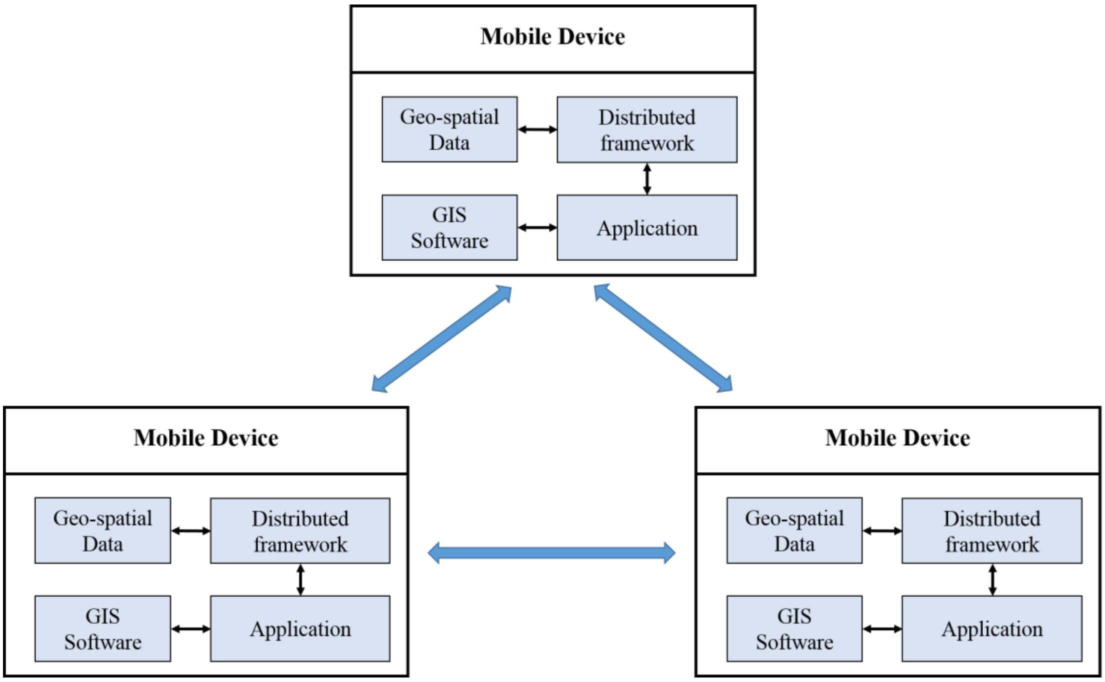


Figure 2.4 Peer-to-Peer Architecture

2.3 Spatial Database in GIS

Location-based services support the ability to obtain the geographical location of mobile devices and then services are provided based on that location. Location-based services are based on postal addresses, digital road maps and point of interest data sets. These maps play a vital role in using location-based which includes position or route based queries. Spatial databases are necessary for these location-based application services.

The Geographic Information System is the principal technology stimulating interest in Spatial Database Management Systems. In spatial databases, geo-data is related with spatial coordinates and extents. They are retrieved depending on spatial proximity. Various kinds of spatial indexes have been presented to assist spatial data retrieval.

Spatial databases have been defined as database systems with a model and query language which provide the spatial data types, spatial indexing and the efficient algorithms for spatial query processing. Several spatial databases in the real-world interest are extremely large. For example, the sizes may be ranged from tens of thousands to millions objects.

Spatial database systems comprise of data regarding objects and properties in the world with respect to their locations. The spatial objects may intersect, be adjacent to others, or contain other objects. Spatial operators are much more expensive to work out compared to the more conventional operators such as select and relational join. The efficiency of these operations is dictated by how the data is represented, and how the relevant data can be quickly retrieved.

The spatial database is a collection of data objects over a particular multi-dimensional space. The quantity of spatial data is large and the structure and relationships of the spatial data are also complex. Efficient processing of queries operating the spatial relationships depend on auxiliary indexing structures. Although there are some approaches operating for spatial relationships, it is extremely inefficient to pre-calculate and store spatial relationships between all of the data objects because of the size of spatial data objects. Alternatively, spatial relationships take place dynamically during query processing. It is necessary to have an index over spatial locations in order to discover the spatial objects efficiently depend on proximity. The basic data structure must provide the efficient spatial operations such as identifying objects in a defined query region and finding the neighbors of an object.

Most of the indexes depend on the principle of divide and conquer. Indexing structures of this approach are typically hierarchical. The approach is suitable for a database system with the limited memory space. An advantage of hierarchical structures is that they are efficient in range searching. Indexing in a spatial database (SD) is unlike the indexing in a traditional database. The data in an SDs are multi-dimensional objects and are related with the spatial coordinates. The search does not depend on the attribute values. It depends on the spatial properties of objects.

The basic idea of indexing spatial data in a space is to separate the space into a manageable sub-spaces. This process is operated until each inseparable sub-space contains a small number of objects that can be keep in a data page. Different space partitioning approach may be designed to reduce the storage space and the retrieval time. The suitable data structure is required to use for reducing the retrieval time. The two basic issues are necessary to design the index structure for the SD. These are the facility of information retrieval and the efficient use of storage.

The index structure which has the efficient locality feature can be used to improve the performance and in order to efficiently handle the queries that include the spatial objects consists of more complex data types. Multidimensional structure can be roughly classify into two different types, data partitioning and space partitioning. The bounding spheres or bounding rectangles is used as the index type previously. That is, the index is store information of the bounded intervals (minimum and maximum) in all dimensions. The most popular index for this type is R-tree [28] and its variants. Then, the space is decomposed recursively into disjoint partitions. The kd-tree [50] and quadtree [66] are this type of index structure.

The spatial indexes are used to support the retrieving and managing of geospatial data effectively and efficiently. The geo-spatial database plays a vital role for the geospatial query processing such as searching the nearest location [69]. The geo-spatial indexing is essential to process a large collection of geo-spatial data for answering such queries.

2.4 Indexing Scheme for Spatial and Textual Index

Indexing Scheme can be generally classified as three types for geo-textual indices. They are spatial indexing, text indexing, and hybrid indexing for spatial and textual index [10].

2.4.1 Spatial Indexing

There are many index structures that have been proposed for fetching the geo-information. The geo-spatial index structures are designed for indexing points data, region data and for handling both types of data. Examples of index structure to index point data are KD trees, Grid files, Point quadtrees and so on. Examples of index structure to handle the region data as well as point data are R-tree, Region quad trees and so on [58].

Spatial indexing scheme has mainly three categories: R-tree based indices, grid-based indices, and space filling curve-based indices [10].

2.4.2 Text Indexing (inverted file, signature file, bitmap)

Text indexing Scheme can be an inverted file, a signature file or a bitmap [10].

Inverted file [86] is widely utilized in information retrieval systems like search engine. It contains a distinct keyword in the collection of documents. Each keyword is corresponded with an inverted list where each list is a sequence of postings. Each posting consists of the identifier of document which contains the description term and frequency of term in document. In each inverted list, the postings are sorted by the id of document.

The two other text indexes are the signature file and bitmap [19]. These text indexes can be used for document indexing. The main idea of signature file is to construct a fast filter to get all documents matching with the query, although some additional documents which do not match may pass the filter also, as 'false hits'. A signature, typically a hash coded version, is generated for each document for this purpose. A bitmap in which each bit represents the occurrence of a keyword can be viewed as a special type of signature file.

As expressed by the experimental results in the work by Zobel et al. [87], signature files were not competitive with the inverted file in terms of efficiency and space, especially for large datasets. However, a recent work [26] on signature files, denoted as the Bit Funnel uses Bloom filter to represent the set of terms in each document as a fixed sequence of bits to address the limitations of signature files.

2.4.3 Geo-textual Indexing

The geo-textual indices integrate spatial and text indexing in one of the three combination which are text-first loose combination, spatial-first loose combination, and tight combination [10]. The indices are categorized according to how the spatial and text index are combined.

A text-first loose combination index usually utilizes the inverted file as the top-level index and then arrange the postings in each inverted list in a spatial structure, which can be an R-tree, a grid or a spatial filling curve.

In spatial-first index, the top level is a spatial structure, and its leaf nodes contain inverted files or bitmaps for the text information of objects in the nodes.

On the other hand, the tight combination index combines a spatial and a text index tightly such that both types of information can be used to prune the search space simultaneously during query processing. Two types of tight combinations have been used: One integrates a text summary into every node of a spatial index (e.g., [4]), and one integrates the spatial information into each inverted list.

Vail et al. [74] proposed the two spatial-textual indexing, text-primary index and spatial-primary index. These two indexes were based on the fixed grid scheme. Spatial-textual indexing scheme is spatial primary and constructs a set of text indexes that is related to spatial. Text-spatial indexing is text primary and spatial that associate with the order lists of document.

2.5 Spatial Keyword Query

Many mobile application comprises a large number of geo-spatial objects with the text description, the latitude and the longitude. The spatial keyword search is significantly interested in research area. In spatial keyword query, searching process is taken place with a given location and keywords and then returns the geo-objects that are not only spatially near but also textually relevant to the given query. There are two types of query (1) attributes query, which depends on non-spatial data and (2) proximity queries, which depend on spatial data such as point range queries and nearest query [69].

2.5.1 Point query

A point query takes a single coordinate and query for items that are related with this location. Point queries are not very typical as they refer to single (x,y) coordinates. GPS data is too fine for items to be precisely at given coordinates [53].

2.5.2 Nearest query

Given a collection of data points p_i and a query point in an m-dimensional metric space, search the data point that is closest to the query point. Nearest query is a query that retrieves closest points *nq* depend on a location *l* from the geo-dataset S. For the result points $nq = \{nq \in S : d(nq, l) \leq d(p_i, l)\}$, where d is the distance [54].

Nearest query is also known as top-k nearest query. Several nearest query has been presented. m-closest keywords (mCK query) [85] is proposed to search the closed tuple in spatial that relevant m query keywords. bR*tree which is R-tree based structure is presented to effectively summarize spatial information and keywords.

Locality search [61] searches top-k sets of spatial web objects that are near the query location and are relevant to the keywords query. Moving top-k spatial keyword query is considered [23] and [36]. A moving spatial point and a set of keyword are taken and the k-objects are continuously searched that match with the objects which is spatially closest and textually relevant.

2.5.3 Range Query

In range query, a set of points rg $(rg_1, rg_2, \ldots, rg_k)$ is retrieved based on the location l and given range (eg. 1km). Given the geo-dataset S, location query l , and range query m , For the set of result points rg, $rg \in S : d(rg_i, l) \leq m \ \forall \ 1 \leq i \leq k$, where d is distance. All the results in range query are within the given range. The range can be circular range and rectangular range in range query.

Two types of spatial keyword query on road network, range keyword query and group keyword query, are considered in [44]. Node-Partition-Distance (NPD) index and keyword coverage are presented to handle these queries in a distributed setting. Keyword coverage takes out the set of nodes within specific distance from certain keyword. NPD index is used to store the useful distances. So, each fragment on road network can independently calculate the exact distance between any node relevant with specific keyword and any node inside each fragment.

2.6 Related Works of Geospatial Index Structure

Various geospatial index structure has been proposed for the spatial keyword query processing. The problem of top-k keyword search on spatial databases was studied [76] and spatial keyword R-tree, called SKR-Tree, was proposed. SKR-tree that is extension of R-tree stores the spatial and text information in R-tree node.

Ap-tree+ [17] was introduced to provide the continuous spatial approximate keyword query by combining the min-wise signatures to AP-tree depend on one-permutation min-wise hashing method.

R*-IF and IF–R* tree [85] were proposed for Boolean range spatial query processing that is to access the web documents that relevant to a given keyword within the pre-defined spatial area. R*-IF and IF–R* are geo-textual indices that merge R-tree for spatial queries and inverted file for retrieving text information. In R*-IF that is spatial-first index, R*tree is firstly built to access all the objects in database without considering the text elements. For text indexing, the inverted file is built in each R*-tree leaf node but not inside the node. So, it first retrieves a set of nodes within the query region and then returns the objects whose documents contain the query keywords. In IF–R* that is text-first index, a separate R*-tree is created for the objects in database containing term t. all the R*-tree relevant to the keywords require to be accessed. In Boolean range query, IF-R* is better than R*-IF.

W-IR-tree [78] was proposed to process the Top-k nearest query. In this tree, the data objects are divided depending on their keywords and then that objects are partitions depend on their spatial locations.

In SKI (Hybrid Spatial-Keyword Indexing) [81], bitmaps and R-tree are used to store text and spatial information for efficiently process top-k spatial queries.

KR* tree [29], keyword R*-tree, was presented to process the spatial keyword query. Each KR* tree nodes hold a set of keyword that are distinct. Then, KR-tree list is created due to vary in the number of keyword appearing in each node. KR* tree list is like an inverted file index but it keeps the id of node instead of stores id of object.

IR2-tree [22] combines signature file and each R-tree node. The signature file is integrated in each IR2-tree node to indicate the text component of all the spatial objects in subtree rooted at that node. The expanding of the tree is based on the length of the signature file. This tree can be utilized for processing Boolean keyword query and Boolean range queries.

IUR-tree [43] was proposed for spatial-textual reverse kNN queries. Each node is with two text vectors (union vector and intersection vector. The weight of each term in union and intersection vectors is maximum and minimum weight of the terms in the document which are contained in the related subtree. The number of objects in subtree at that node associate with each non-leaf node. Branch-and-bound algorithm is presented over the index to process the reverse kNN queries.

IR-tree [13,79] was proposed to provide Boolean range queries, Boolean kNN queries and Top-k kNN queries. R-tree with the text component of the objects in the related subtree. Each node has a pointer to inverted file to describe the objects in the subtree embedded at the node. The inverted file of a node includes vocabulary of all distinct term in the text representation of the objects and a set of posting lists that relates to a term. Li et al. [42] proposed IR-tree with top-k document search algorithm for textual filtering, spatial filtering, relevance computation, and document ranking.

S2I (Spatial Inverted Index) [63] is presented for Top-k spatial keyword queries. It maps each frequent term to an aggregated R-tree (aR-tree). aR-tree stores coordinate of the objects and maintains an aggregated value that describes the maximum impact (normalize weight) of the term on the object under the node. It is possible to access the object ranked in terms of both keyword relevance and spatial relevance.

SKIF (Spatial-Keyword Inverted File) [36] is proposed to handle location-based web searches. It can process both spatial and textual data simultaneously. The location of each object assumes as region instead of a single point and spatial relevant is shown as the overlap between query region and region of an object. Inverted file is for the text component. For spatial, each distinct grid cell is described by inverted list. Each inverted list composes of a list of postings. Each contains an object id and its spatial value. SKIF intended to search *k* objects with the highest weighted scores in terms of both text relevancy and spatial relevancy.

2.7 Summary

This chapter discusses the overview of the Mobile GIS and its architecture. Then, it introduces the spatial database and reviews the indexing Scheme for spatial and textual index. This research emphasizes on the geospatial indexing. There are three indexing schemes for spatial and textual indexing. This chapter reviews these three indexing schemes: spatial indexing, text indexing, and geo-textual indexing. In addition, it describes the spatial keyword query and discusses the related work of the combined index structure for spatial keyword query processing.

CHAPTER 3
GEO-SPATIAL INDEX STRUCTURE

The efficient geo-spatial index structures are required to consider for answering the spatial keyword query that contains the spatial and textual query. Geo-spatial index structure that combines the spatial location and text data enable to index both geo-location and text data. It provides to obtain the efficient and effective spatial keyword queries processing. This research is considered for constructing the efficient geo-spatial index structure to process spatial keyword queries with two languages, Myanmar and English. In this chapter, the spatial index structure is briefly introduced at first. Then, the types of the spatial index are discussed. Next, the background theories of the proposed index structure are also mentioned. The proposed geo-spatial index structure is presented with the architecture and the construction of the proposed index structure is also explained.

3.1 Introduction of Spatial Index Structure

One of the most important applications of spatial database systems is Geographic database systems, also known as a geographic information system (GIS). It generally comprises two parts. The first one is to query the components and to manipulate the geographical data. The second one is to manage the components and to store the data. Nevertheless, the main purpose of the geographic information system is for analyzing the geographical data [38].

Spatial index plays a vital role to get the better performance of the geo-spatial database. Spatial data is used in several application areas such as social, geographic information systems and image databases. The spatial objects are complex because they contain not only spatial information (location data) but also their text descriptions. Each spatial object in databases can be as a point in a multidimensional space. The spatial indexing techniques enable to process the geo-spatial data efficiently and effectively. The purpose of spatial indexing is to provide the spatial selection of spatial objects such as nearest queries, range queries and so on [50]. Spatial indexes that have been utilized can be mainly categorized into three types: tree structure, space-filling curves, and grids. These index structures are discussed in the next sections.

3.2 Tree structure

There are several tree structures for spatial indexing. Among them, R-tree, Quad-tree, Kd-tree are popular tree structures and they are also presented in this section.

3.2.1 R-tree

R-tree is a tree-based data structure that is used for accessing multidimensional information (eg. Geographical points) [28]. R-tree, dynamic data structure, is height-balanced tree that is similar to B-tree.

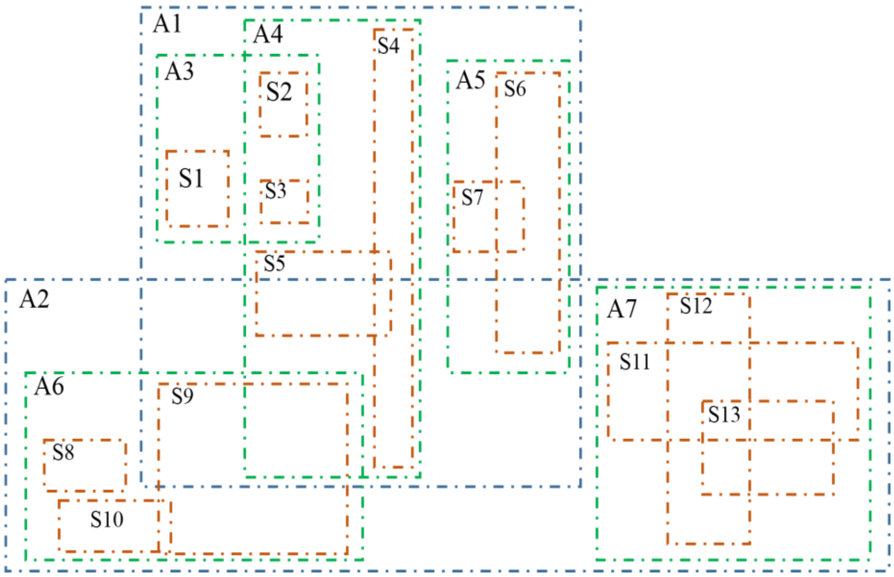


Figure 3.1 Objects with their Minimum Bounding Rectangle

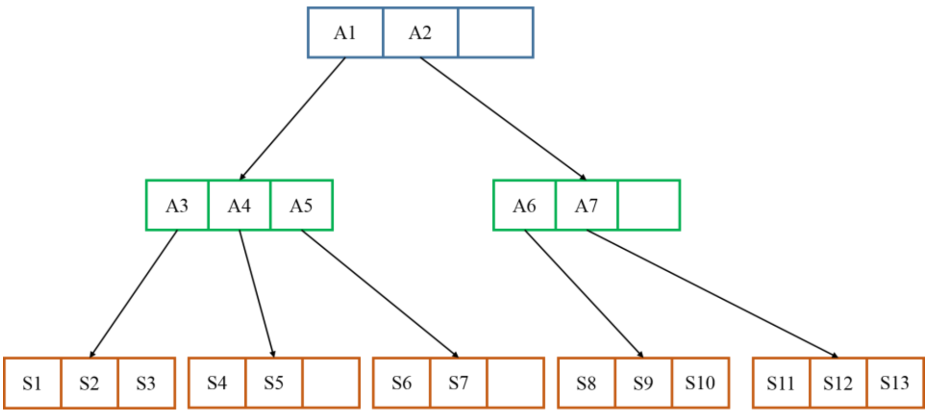


Figure 3.2 R-tree Structure

This index used the minimum bounding box whose shape is a rectangle that represents a set of geometric objects. It contains intermediate and leaf nodes. R-trees nodes may hold multiple items. In the intermediate node, each node contains MBR (minimum bounding box) which encloses all its children and pt (pointers) to the child nodes. Each leaf node has MBR of data objects and id (object identifier) that point to the objects in the database. The bounding box can overlap each other in the case of bounding different nodes. Because of overlapping rectangles, it needs to search multiple path while searching for a specific object. R-tree nodes are implemented as disk block so that the required size depends on the maximum number of items per node. In R-tree, each node in the leaf contains between n to N index records if it is not the root where N is the maximum number of items in a node and n is the minimum number of item in a node that is n ≤N/2. Each intermediate node contains between n to N children if it is not the root. The node in the root contains at least two items if it is not a leaf. All leaf nodes must occur on the same level [55]. Figure 3.1 shows the minimum bounding rectangles of spatial objects and R-tree structure is shown in Figure 3.2.

3.2.2 Quadtree

The quadtree is hierarchical data structure and its characteristic is depending on the recursive splitting space [66]. Quadtrees is a simple method to index the spatial data and recursively slipping the space into four equal sub-squares. It can be different based on the data type, the idea of the splitting process and the resolution [65]. There are several different types of quadtree and some useful quadtrees, which are point quadtree, point region quadtree, and region quadtree, are introduced in this section.

Point quadtree is split based on the data that store in the internal node of the quadtree. Each internal point node in quadtree contains four children that are NE (northeast), NW (northwest), SW (southwest), and SE (southeast) [50]. A point quadtree node has two portions that are 4 pointers for pointing 4 children and point data to describe its coordinates and its related value which may be the country name. The tree can be unbalanced or skewed because its shape is based on the order of the data point insertion. Example of point quadtree is shown in Figure 3.3.

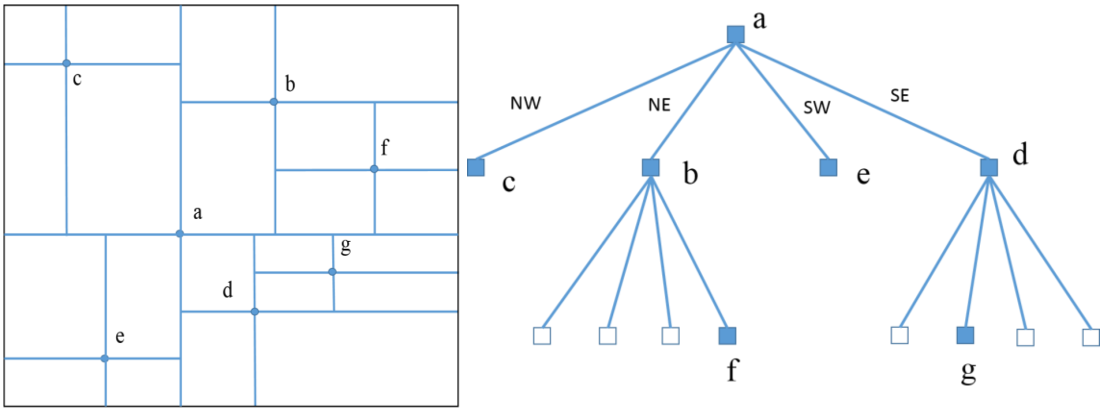


Figure 3.3 Point Quadtree Example

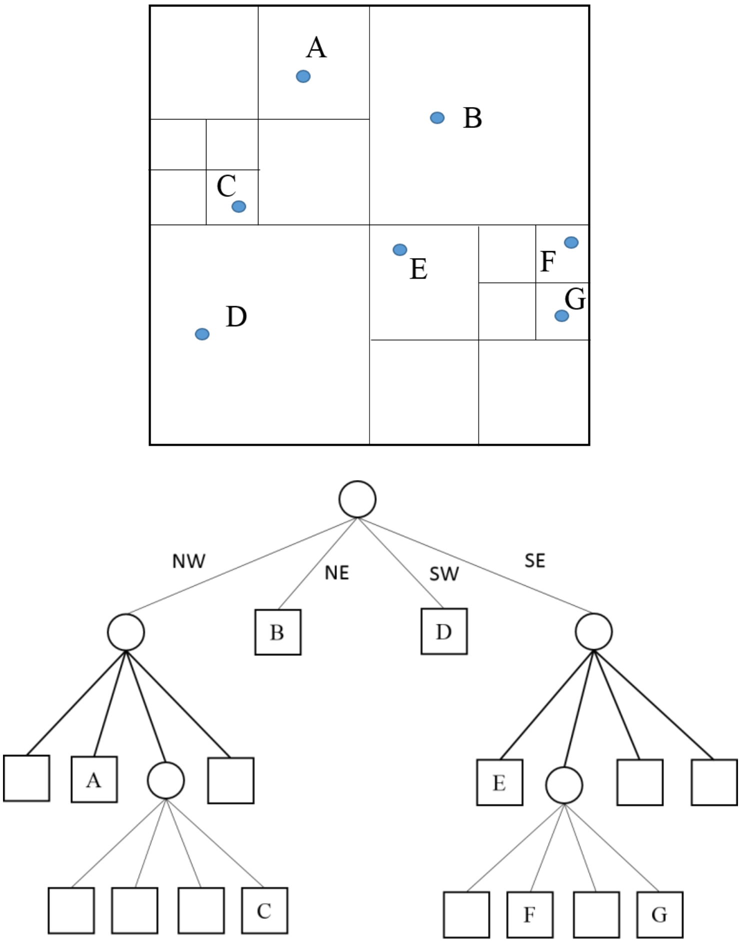


Figure 3.4 Point Region Quadtree Example

Point region quadtree, which can be seen in Figure 3.4, is a little different from the point quadtree. It also has four children. But it does not split depending on the data point. It split the region which holds the data points into four subsquares until zero or a single data point contained in the region. So, it will be recursively split to describe a set of point in two dimensional if every square has more than one point [68]. It can require many subdivisions to separate densely clumped points and may lead to a deep search paths.

Region quadtree (in Figure 3.5) is used to represent the binary image. 2nx2n pixels array is subdivided into quadrants until a single value (color) is contained in the square. It is one of the popular quadtrees and is used to hold the raster data [24] [67].

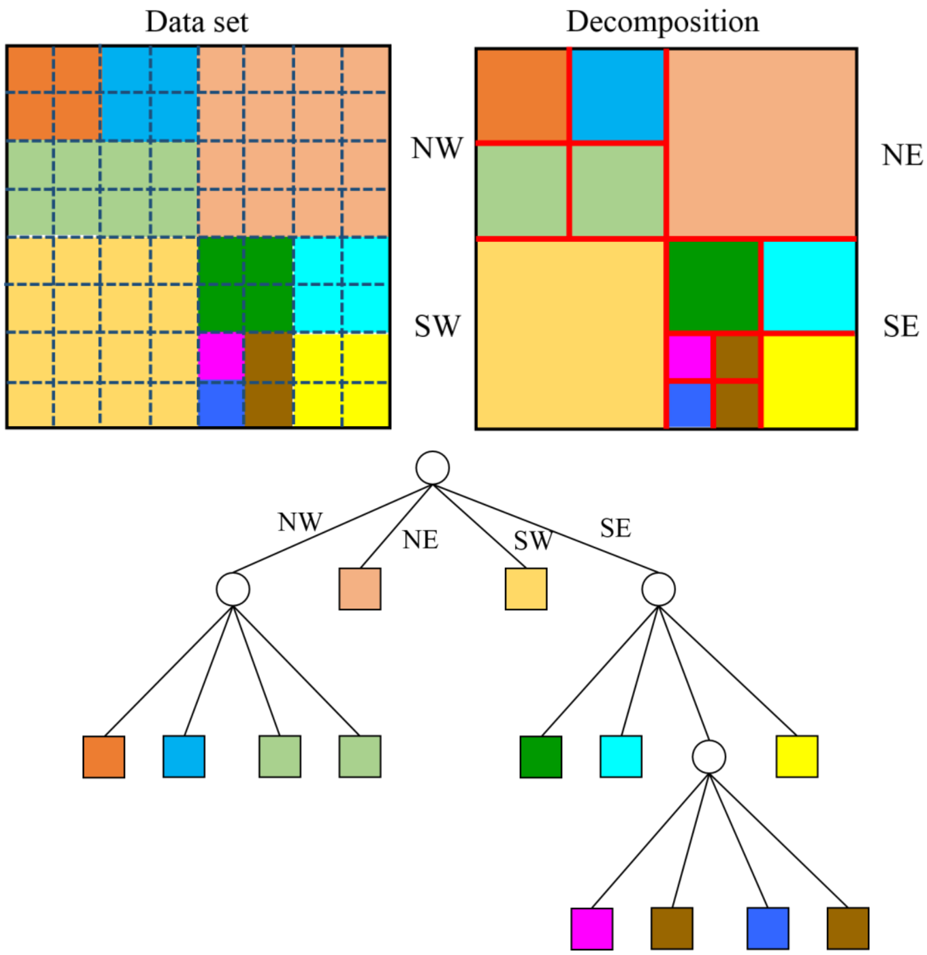


Figure 3.5 Example of Region Quadtree with its Dataset and Decomposition

3.2.3 Kd-tree

Kd-tree is like a space partitioning tree that recursively split the space. It is a binary search tree that can store the k-dimensional points [50]. In two-dimensional space, the rectangular area is corresponded to each kd-tree internal node that has a single point. The entire region corresponds to the root node. The rectangle area is split into two portions (left sub-tree and right sub-tree) that roughly has equal size. The value of point in left sub-tree is smaller than or equal to the decomposition value. The value of point in the right sub-tree is larger than the decomposition value. The decomposition point is kept at the root and a set of points in the two-portion are stored recursively in two sub-tree. Every point contains the x coordinate value and y coordinate value. At first, the x coordinate is split and then y coordinate is split. A set of points S is alternatingly split by x-coordinate and y-coordinate until S only contains a single point. In the splitting by x-coordinates, S is split with the vertical line into two parts throughout the median x-coordinate of the points if the depth of the node is even. The slipping point is stored in the root node. Then a set of points in the left side is stored in the left sub-tree and a set of points in the right side is stored in the right sub-tree. In the splitting by y-coordinate, S is split with the horizontal line into two parts throughout the median y-coordinated of the points if the depth of the node is odd. The points which are below or on the splitting node are stored in the left sub-tree and the points which are above the splitting node are stored in the right-subtree [15]. The tree structure of the kd-tree and its decomposition are shown in Figure 3.6. In K-d tree, the data points are scattered all over the tree. So, it requires more computation and intensive search.

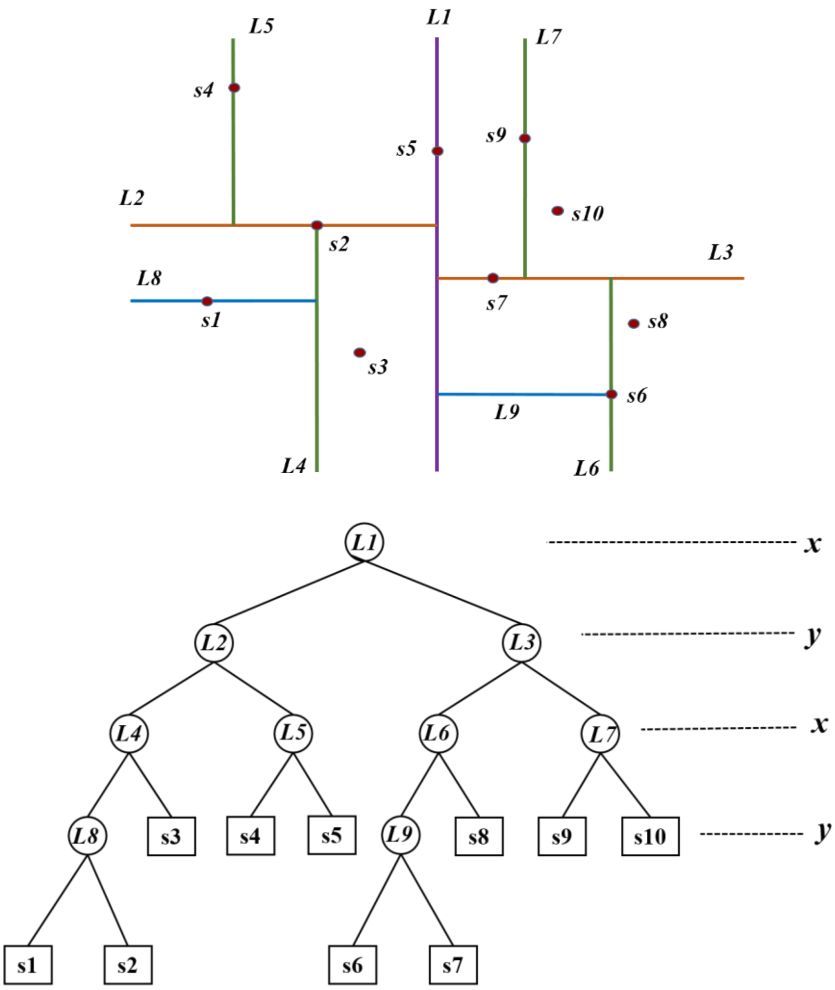


Figure 3.6 Kd-Tree Structure with its Decomposition

3.3 Space Filling Curve

Space-filling curve (SFC) is utilized for mapping from the multidimensional space to one-dimensional space. It passes through each point in the entire space only one time without overlapping. It arranges the points as a linear order in the multidimensional space [47]. In the space-filling curve, a basic path is started from the grid of side 2 on the k-dimensional square. The path goes through each point in the grid only once without across itself. It contains two free ends that may be connected with other paths. The basic curve is order 1. Each vertex of the basic curve is changed with the curve of order i-1 to derive a curve of order i. That may be rotated and/or reflected to fit the new curve [18].

Some popular space-filling curve is Z-order curve, RBG curve, and Hilbert curve. Z-order mapping is used as a linear index for two-dimensional space. The property of mapping from two-dimensional to one-dimensional value in z-order curve is bit shuffling property that interleaving the bits of two coordinates points to obtain the one-dimensional number. The basic order 1 in z-order curve is a 2x2 grid shown in Figure 3.7 (a). For higher order of z-order curve, each vertex of the basic curve is replaced with the previous order curve. The order of 1 and 3 of the z-order curve is shown in Figure 3.7 (b) and 8 (c) respectively [18,33,49,51,52].

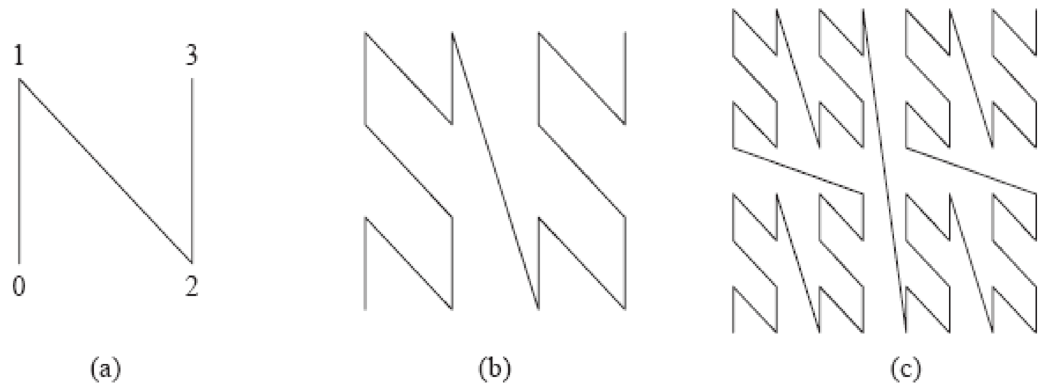


Figure 3.7 Structure of Z-order Curve (a) Order 1 (b) Order 2 (c) Order 3

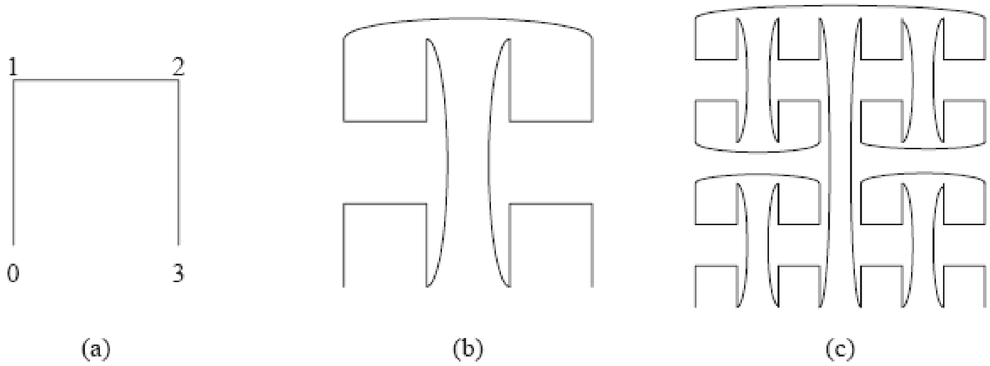


Figure 3.8 Structure of RGB Curve (a) Order 1 (b) Order 2 (c) Order 3

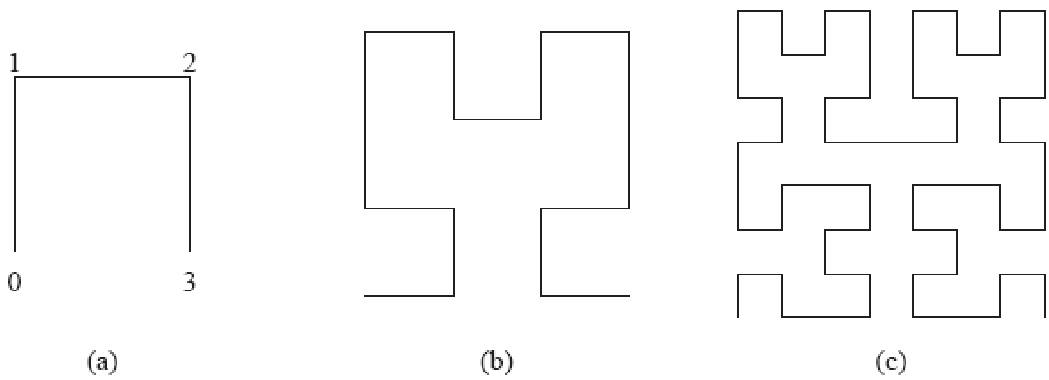


Figure 3.9 Structure of Hilbert Curve (a) Order 1 (b) Order 2 (c) Order 3

In the Gray-code curve (RBG curve), the numbers are coded into binary gray code in which successive numbers differ in value in one-bit position only. Order 1 in the basic RGB curve is also the 2x2 grid. It is shown in Figure 3.8 (a). The previous order curve is reflected over the x-axis and then over the y-axis in the higher order of RGB curve. The order of 2 and 3 is shown in Figure 3.8 (b) and 3.8 (c) [18,33,20,21]. Hilbert curve preserves the objects in the multi-dimensional space in the linear space [32,41,48]. It is used in the proposed index structure of this research and the detailed discussion of Hilbert curve can be seen in section 3.5.1. The structure of the Hilbert curve can be seen in Figure 3.9.

3.4 Grid file

Grid file is one of the data structure that outperforms the one-dimensional index for queries containing the multidimensional data. In the grid file, the data space is partitioned into an irregular grid into cells that may have different shape and size. The split lines pass through the whole space and its positions are stored in scales utilizing one scale per dimension. The grid file comprises two separate parts: the directory and the linear scale. The director is a k-dimensional array whose information is logical pointers to buckets. Each cell in data space relates to one data element of the directory array. All data elements within a cell are stored in the bucket pointed to its corresponding directory information. Two or more cells can share the same bucket in an order containing more than one cell in the bucket regions. The linear scale stores the information of dividing each dimension. It is one linear scale per dimension. Every linear scale is one-dimensional array which split the values on a specific dimension. The example of a grid file is shown in Figure 3.10 [25].

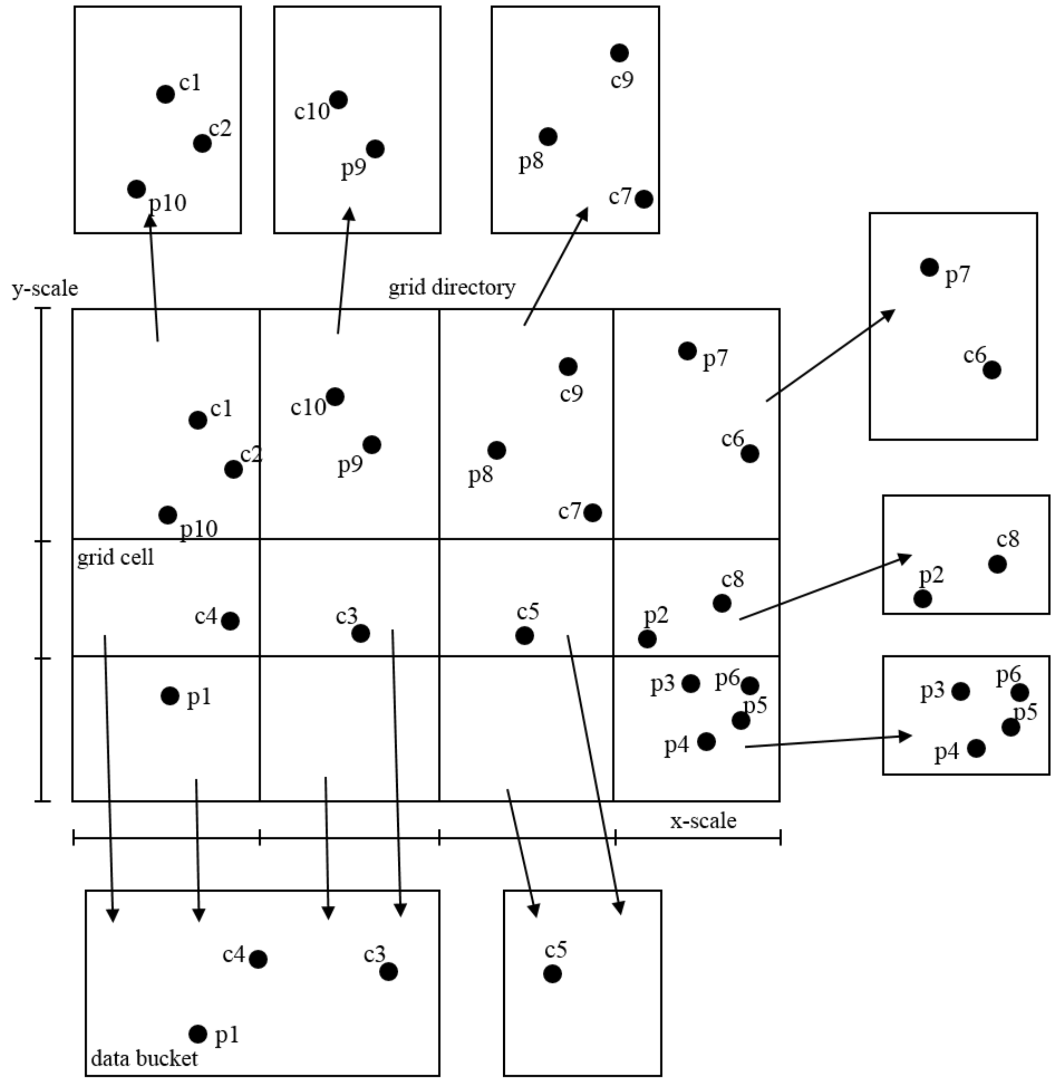


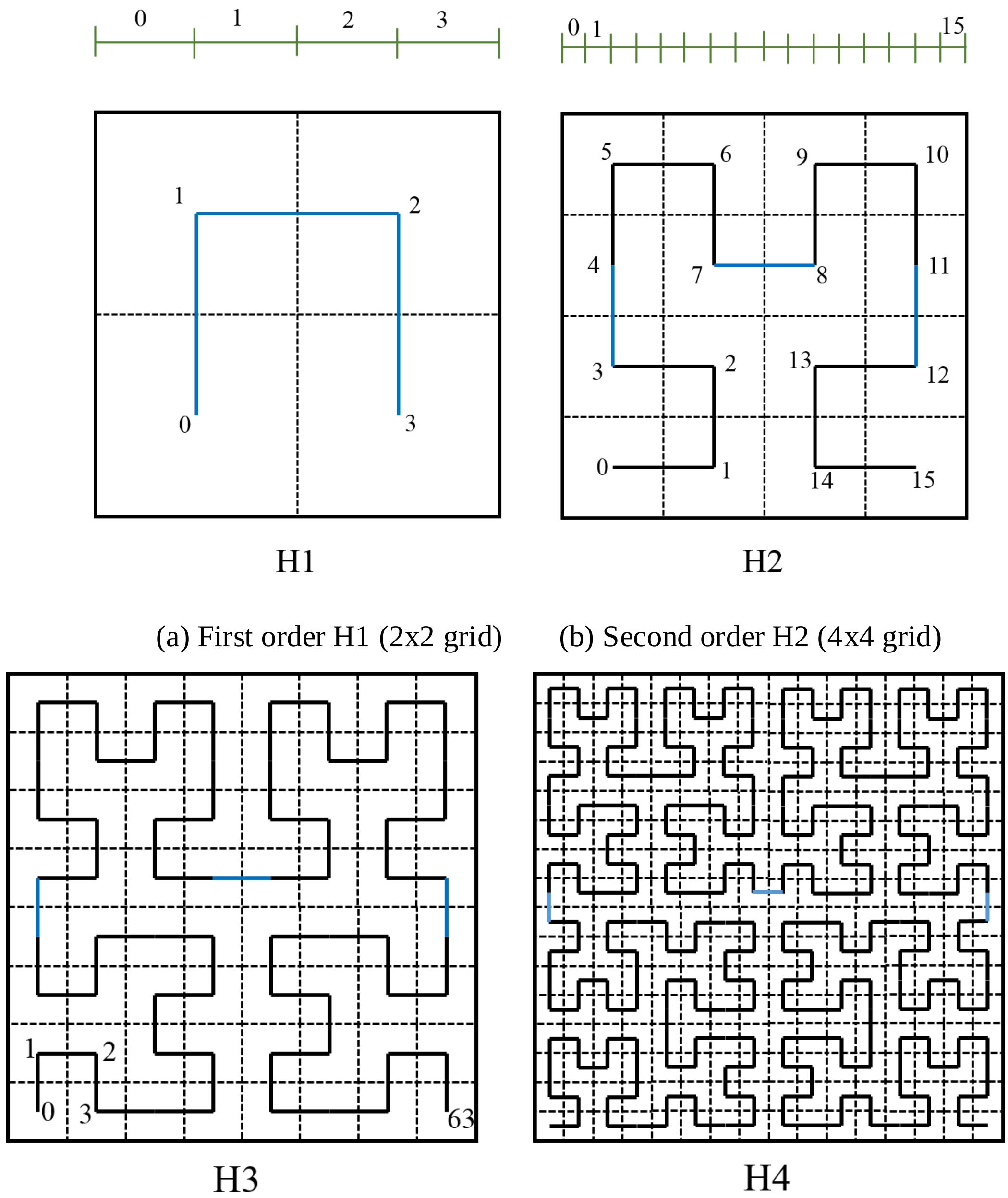
Figure 3.10 Example of Grid File

3.5 Background Theory of Proposed Geo-Spatial Index Structure

In this research, a new geo-spatial index structure is proposed for processing the geo-spatial keyword query. The languages of the query keyword in this research are Myanmar and English languages. This geo-spatial index structure provides to retrieve the nearest keyword query and rang keyword query from the current location or the user's desired location. In addition, it can also process keyword query which explores the geo-information that relevant to the given keyword. Hilbert curve, B-tree and inverted file are used for constructing the proposed geo-spatial index structure. The inverted file is mainly used for text indexing. At first, the Hilbert curve and B-tree are discussed and then the proposed index structure is explained in detail with the architecture.

3.5.1 Hilbert curve

Hilbert curve is one of the popular curves in space-filling curve. It stores the multidimensional space data into one-dimensional data [11]. The interval I can be mapped continuously onto the square Q. After partitioning I into four equal subintervals and Q into four equal sub-squares, each subinterval can be mapped continuously onto one of the sub-squares. This step is repeated for the next step. The sub-squares can be ordered because of adjacent subintervals correspond to adjacent sub-squares with an edge in common. I and Q are partitioned into $2^h x 2^h$ equal sub-square for h=1,2,3... if this operation is continued and infinitum [64]. The partitioning operation is terminated after h steps to given an approximation of a space-filling curve of order h. The Hilbert space-filling curve goes through every point only once in n-dimensional space without crossing the path. If D is a set of points in space, the points can be stored in the sequence order according to the curve go through them [3]. Two-dimensional space of Hilbert curve is shown in Figure 3.11. Figure 3.11 (a) shows the basic Hilbert curve (order 1, H1) of the 2x2 grid. In the process for deriving the higher order of the Hilbert curve, the bottom left vertex and the bottom right vertex is rotated 90 degrees clockwise and 90 degrees counter-clockwise respectively. Each pair of consecutive sub-squares share a common edge. The curve starts from the lower-left corner and ends at the lower right corner and the points. The order 2 (H2), the order 3 (H3) and the order 4 (H4) of Hilbert curve are shown in Figure 3.11 (b), (c) and (d).



(a) First order H1 (2x2 grid) (b) Second order H2 (4x4 grid)

(c) First order H1 (8x8 grid) (d) Second order H2 (16x16 grid)

Figure 3.11 Hilbert Space Filling Curve of Order 1, 2, 3, and 4

3.5.2 B-tree

B-tree is a balanced search tree which is a generalization of a binary search tree and stores the data in sorted order. Unlike a binary tree, each b-tree node may contain keys and children. B-tree holds the values in each node of the tree. Each node may contain more than two children and all the leaf nodes must have at the same level. In b-tree of order d, every node contains at most d children and at least d/2 children [35].

R.root is the root in B-tree R. B-tree R is the rooted tree containing the following properties :

1. each nodes x has three attributes: x.n, x.n keys and x.leaf. The number of keys, x.n, is stored in the node x. The x.n keys ($x.key_1$, $x.key_2$, $x.key_3$, …, $x.key_{x.n}$) are stored in nondecreasing order. Thus, the stored keys are $x.key_1 \leq x.key_2 \leq x.key_3 \leq \ldots \leq x.key_{x.n}$. In x.leaf, a Boolean value is false if x is an internal node and a Boolean is true if x is a leaf node.
2. If an internal node of b-tree has x.n keys, then x contains x.n+1 children. Therefore, each internal node x has x.n+1 pointers ($x.p_1$, $x.p_2$, …, $x.p_{x.n+1}$) to points its children. Leaf nodes do not have children so that their p_i attributes are undefined.
3. The key $x.key_i$ split the ranges of keys stored in each subtree. If k_i is any key stored in the subtree with root $x.p_i$, then $k_1 \leq x.key_1 \leq k_2 \leq x.key_2 \leq \ldots \leq x.key_{x.n} \leq k_{x.n+1}$.
4. All the leaves have the same depth which is the height h of the tree.
5. Nodes contain the lower and upper bounds based on the number of keys of the nodes. These bounds have a fixed parameter $t \geq 2$ which called the minimum degree of the B-tree. Each node except the root must have at least t-1 keys. Every internal node expects the root has at least t-1 children. The root must have at least one key if the tree is nonempty. Every node may contain at most 2t-1 keys. So, an internal node may have at most 2t children. If it contains exactly 2t-1 keys, a node is full [14].

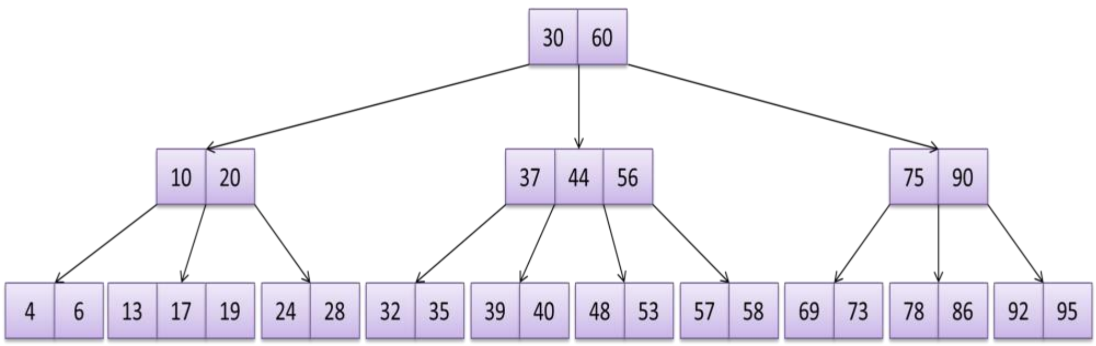


Figure 3.12 The Structure of the B-tree Order 2

The structure of the b-tree is shown in Figure 3.12. if $n \geq 2$, the height h of B-tree with minimum degree t for any n key is $h \leq log_t \frac{n+1}{2}$.

3.5.3 Inverted File

Inverted file which also known as the inverted index is used as the most common data structure in both information retrieval system and database management. It is a word-oriented mechanism to index a text collection in order to speed up the searching process. In tradition, the inverted file structure is comprised of two components: the vocabulary and the occurrences. Vocabulary consists of all distinct words in the text. Occurrences contain the list that consists of all necessary information for each word of the vocabulary. It may be frequency, text position, document where the word appears and so on [60]. Inverted index is also used in spatial query processing for text indexing. In this techniques, it keeps the inverted list for each keyword. The inverted list principally contains the ID of all the location consisting of the specific keyword. It is an effective way of searching on the keyword matching for answering the location-based query.

3.6 Constructing Proposed Geo-Spatial Index Structure

In this section, the proposed geo-spatial index structure is discussed detail with the architecture. This proposed index structure is constructed by using the Hilbert curve, B-tree and inverted file. B-tree with inverted file is constructed based on the Hilbert value (hvalue). The inverted file contains Id, keywords with Myanmar and English language, and location. This index structure is supported for processing the spatial keyword query with both Myanmar and English keywords. The proposed geo-spatial index structure is built according to the services such as restaurant, bus stop. The architecture of the proposed index structure is shown in Figure 3.13.

B-tree that is balanced search tree is better at minimizing disk I/O operations. B-tree can efficiently handle one-dimensional queries. It is less efficient in retrieving the two-dimensional data. Thus, Hilbert curve is used to map the two-dimensional coordinate point to one-dimensional point for building B-tree with inverted file in this proposed index structure. Hilbert Curves express a mapping between the coordinate of points and one-dimensional values. It goes through every point in n-dimensional space once and the points are placed in a sequence according to the order in which the curve passes across them [40].

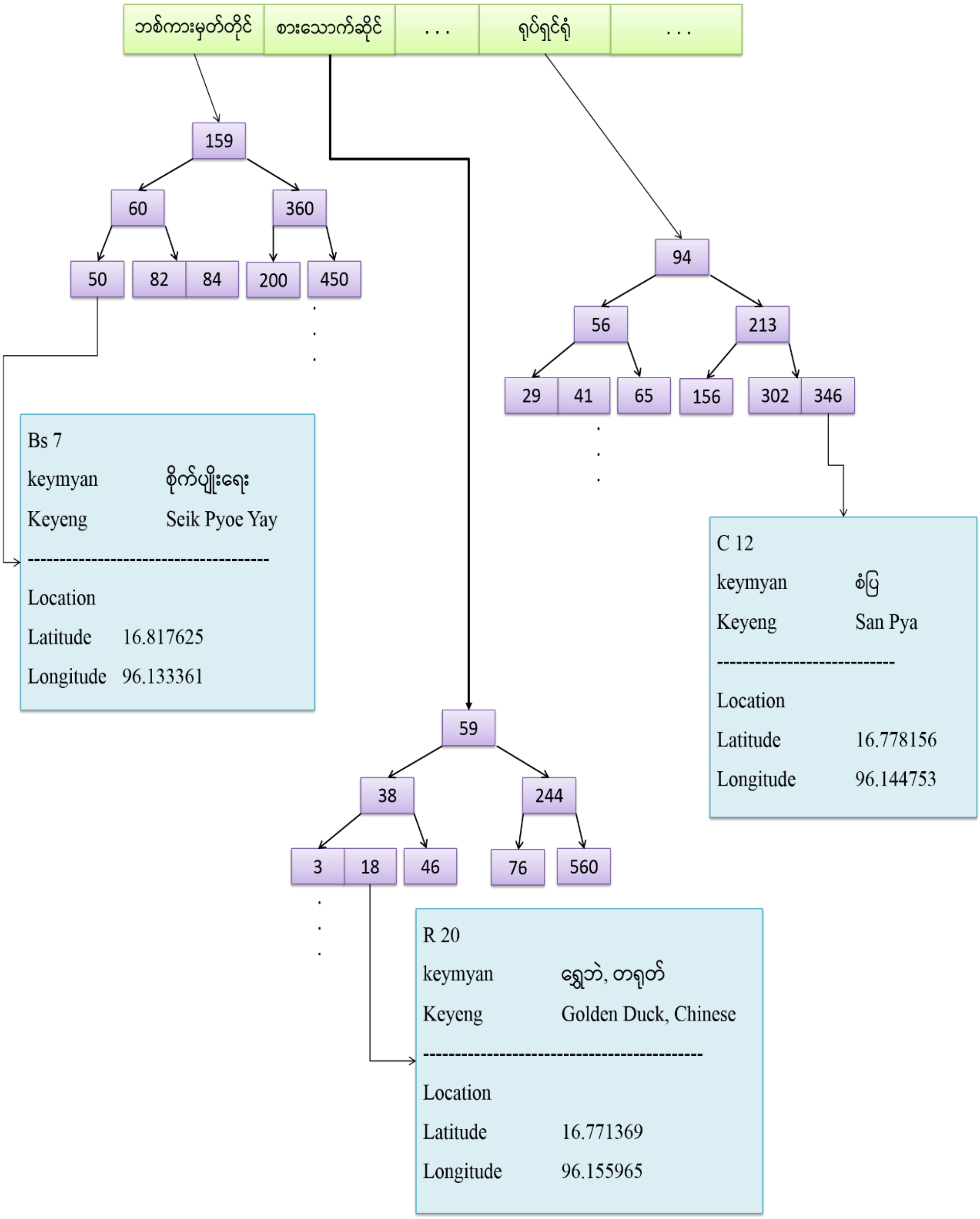


Figure 3.13 Architecture of Proposed Index Structure

3.6.1 Hilbert Curve's Mapping

The mapping example of a Hilbert curve can be seen in Figure 3.14. The Hilbert value is calculated before constructing B-tree with the inverted file. The calculation steps of the Hilbert value are described in the following. In this calculation, x and y coordinate takes as input and the output is hvalue (decimal value).

Step 1: Convert x and y coordinate to binary representation

Step 2: Interleaving the binary number of x and y coordinate and assign to the string S.

Step 3: Split the string S from left to right into the two-bits strings (si where i=1,2,…,n)

Step 4: Change for each si with the decimal value (di) with the following rules:

If si is ‘00’ change to ‘0’

If si is ‘01’ change to ‘1’

If si is ‘10’ change to ‘3’

If si is ‘11’ change to ‘2’

(where i=1,2,…,n)

and put the value into the array in the same order as si

Step 5: For each number m in the array

If m=0, change the every following occurrence of the value in the array ‘1’ to ‘3’ and ‘3’ to ‘1’

If m=3, change the every following occurrence of the value in the array ‘0’ to ‘2’ and ‘2’ to ‘0’

Step 6: Convert each number in the array to two-bit binary string and concatenate all string from left to right.

Step 7: Convert the binary string to the decimal value [18].

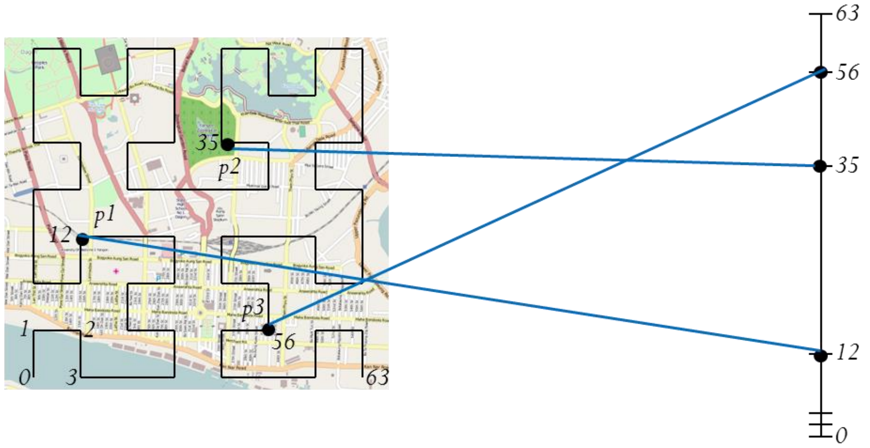


Figure 3.14 Mapping Example of Hilbert Curve

For example, let x coordinate value be 4 and y coordinate value be 5. The binary representation of x is ‘1001’ and the binary representation of y is ‘101’. So, the string S is ‘110001’ after interleaving the binary string of x and y. Then, the binary string S is divide into two-bits strings: s1=11, s2=00 and s3=01. In the next step, these two-bits string are converted to the decimal value according to the rule shown in step 4. Therefore, s1 value ‘11’ is converted to ‘2’ and assign it to d1. s2 value ‘00’ is converted to ‘0’ and assign it to d2. s3 value ‘01’ is converted to ‘1’ and assign it to d3. These decimal value are put into the array as the same order of si. According to the step 5, the result of the d1, d2 and d3 are 2,0 and 3 respectively. The value of the binary string is 100011. So, the hvalue is 35. This Hilbert value is used in constructing the B-tree.

3.6.2 Procedure for Constructing B-tree with Inverted file

The procedure of B-tree in proposed geo-spatial index structure is described in BTreeInsert, SplitChild and nonfullInsert algorithm. BTreeInsert algorithm is shown in Figure 3.15.

Algorithm: BTreeInsert

```
Input : t: B-tree with empty root node
        key: hvalue
        p: a set of points with coordinates and correponding value set.
Ouptut: T: B-tree

 r ← t.root
 If r.count is equal to 2m-1
       create a new node s
       t.root ← s;
       s.leaf ← false;
       s.count ← 0;
       s.c1 ← r;
       call SplitChild(s,1,r);
       call nonfullInsert(s,key,p)
 Else nonfullInsert(r,key,p)
 End If
```

Figure 3.15 Algorithm of B-tree Insertion

BTreeInsert takes B-tree with empty node (t), Hilbert value (key) and a set of coordinate points with their corresponding value (p) as inputs. Therefore, B-tree with empty node is nedded to create before using BTreeInsert algorithm. Then, BTreeInsert algorithm is used to insert a new key. If the root node is not empty, it inserts the value into the node. Whether the root node is full or not is checked in Line 2. Line 3 to 9 takes place if the root node is full. If the root is full, it is split and a new node s is created and it becomes the root. SplitChild algorithm is called for splitting the root and nonfullInsert algorithm is called for to add the key and data into the node.

Algorithm: SplitChild(x,i,y)

```
create a new node z
z.leaf ← y.leaf;
z.count ← m-1;
For j=1 to m-1
        z.key_j ← y.key_{j+m}
End For
If not y.leaf
        For j=1 to m
                z.c_j ← y.c_{j+m}
        End For
End If
y.count ← m-1
For j=x.count+1 downto i+1
        x.c_{j+1} ← x.c_j
End For
x.c_{i+1} ← z
For j=x.count downto i
        x.key_{j+1} ← x.key_j
End For
x.key_i ← y.key_m
x.count ← x.count+1
```

Figure 3.16 Algorithm of Splitting Child

The procedure SplitChild takes three inputs: x (nonfull internal node), i (index such that $x.c_i$) and y (node to split). The full node y is split into two nodes, y and z, having only t-1 keys each. The keys that are larger than the median key of y is taken into the node z and z become a new child of x. The median key of y is moved up into the parent node x which split z and y. Step 1 to 11 create a new node z and insert the largest m-1 key and corresponding m children of y into z. Then, the key count for y is adjusted in the step 12. In steps 13 to 21, z is added as the child of x and the median key from y is moved up to x in order to divide y from z. Then, the key count of x is adjusted. Figure 3.16 shows the SplitChild Algorithm.

Algorithm: nonfullInsert(x,k,p)

```
i ← x.count
if x.leaf
        while i is greater than or equal to 1 and k is less than x.key_i.k
                x.key_{i+1} ← x.key_i
                i ← i-1
        End while
        insert k and p to the x.key_{i+1}
        x.count ← x.count+1
else
        while i is greater than or equal to 1 and k less than x.key_i.k
                i ← i-1
        End while
        i ← i+1
        If x.c_i.count is equal to 2order-1
                call SplitChild(x,i,x.c_i)
                if  k is greater than x.key_i.k
                        i ← i+1
                End If
        End If
        call nonfullInsert(x.c_i,k,p)
End If
```

Figure 3.17 Algorithm of Inserting the Value into the Node

The procedure of nonfullInsert is for inserting the value into the node. nonfullInsert algorithm is shown in Figure 3.17. It takes the node x which is nonfull node, the key k and p that is the corresponding data of the two-dimensional geo-point. The step 3 to 8 work for inserting k and p into x which is the leaf. If x is not a leaf node, k is inserted into the appropriate leaf node in the subtree rooted at internal node x. In this case, step 10 to 13 finds the spot to recurse on the correct node. Step 14 discover whether the child of x is full. If it is full, that child node is split into two nonfull children by using the SplitChild in step 15. Step 16 and 17 determine which of the children is now the correct one to descend to. Then, the value is inserted into the appropriate subtree in step 20.

3.7 Summary

The spatial index structure is described in this chapter. It has three main categories, tree structures, space-filling curves, and grid. These three types of index structures are presented in this chapter. Then, it discusses the theories of Hilbert curve, B-tree and inverted file that are used for building the proposed index structure. Besides, the proposed geospatial index structure is discussed in detail. It explains the creation of the proposed index structure with the use of architecture. It discusses the step by step procedure of creating this proposed index structure, (1) the operation of Hilbert curve that maps between two dimensional and one dimensional, (2) the construction of the B-tree with the inverted file.

CHAPTER 4
THE PROPOSED SYSTEM ARCHITECTURE

Location-based search helps the user visiting with mobile devices to access the location information. The usage of location based system on web has been emerged and interested in past decade. Because of the popularization of the mobile devices, location based system using on mobile devices is increasing to access the location information. The information of a place contains a coordinate point (x,y) and the textual information. The location information is accessed by using the spatial query such as range query, nearest query. So, the geo-information is answered based on the given query location and given textual information.

This chapter presents the system architecture of the proposed system that provides the user to access the geo-information. Then, the processing of the geospatial keyword queries using the proposed index structure is also discussed. Geo-spatial keyword query processing in this research are nearest neighbor query, range query, and keyword query.

4.1 Proposed System Architecture

As the use of mobile devices is rising, the use of location-based services applications is also increasing. Users utilize location-based applications to know their location and to find their desired location.

The main idea of this proposed system is to provide all the visitors to obtain their required geo-information anywhere and anytime. The language of the user can vary, depending on their countries. So, the keyword search in this proposed system mainly relies on two languages, Myanmar and English, in order to provide both foreign people and local people. In addition, this proposed system also provides the user to obtain the geo-information whether internet access is available or not. The architecture of the proposed system can be seen in Figure 4.1. The geo-information with their coordinate points and their respective information is stored in the spatial database. Collecting the geo-data for the spatial database is explained in section 5.4. There are three main search types (1) nearest search, (2) range search and (3) keyword search.

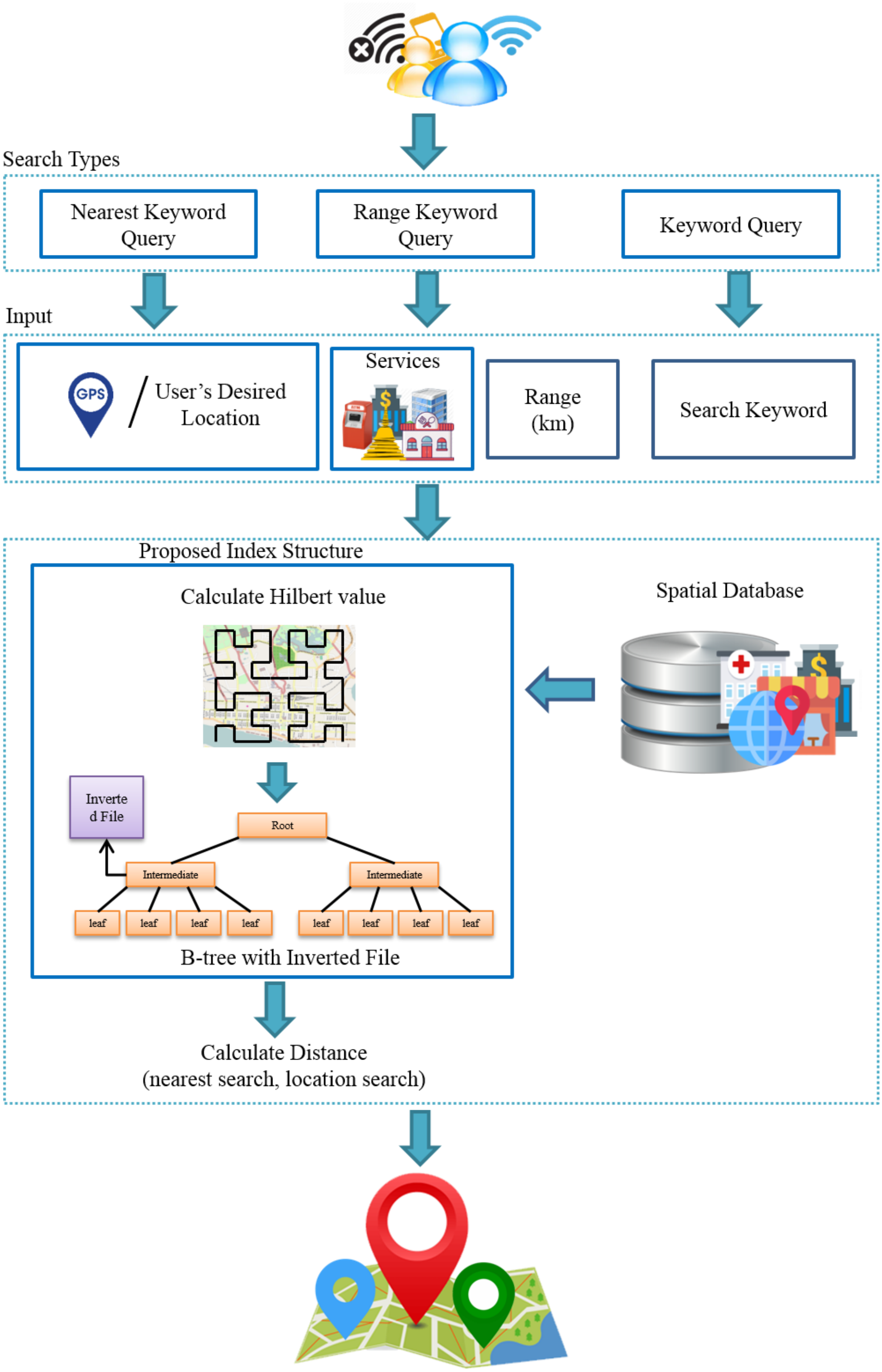


Figure 4.1 Proposed System Architecture

It takes four inputs to retrieve the desired geo-information in this proposed system. The four inputs are user location, services, range, and the keyword. The searching process is taken place by using the proposed index structure depending on the selected search type and given input. The desired geo-information is obtained with their coordinate location and relevant text description. Then, the result is displayed on the map. The system flow diagram is shown in Figure 4.2.

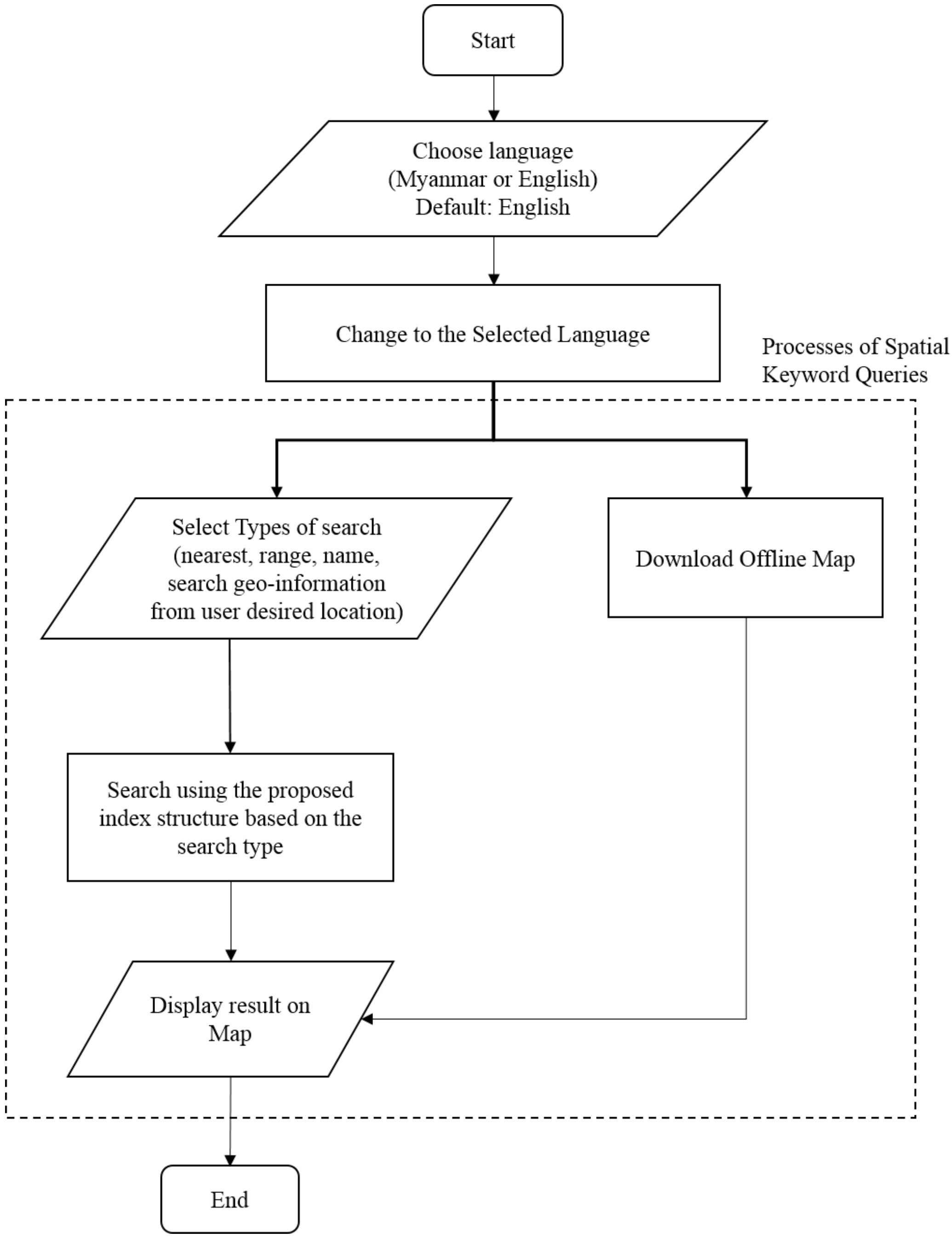


Figure 4.2 Flow Chart of the Proposed System

It generally contains five steps. The first step is to choose the desired language from two types of languages (Myanmar and English). The system is operated based on the language which is the user-selected language. If the user does not choose the language while using the system, it is operated with the English language which is default language of the system. The next step is downloading the offline map. The map of this system can be used in both offline and online versions. To use the map in an offline, the map is needed to download. If it is not downloaded, the map cannot be seen in an offline condition. It can only work on the online. The third step is choosing the search types. The searching process types are (1) nearest query from current location, (2) range query from current location, (3) keyword query, and (4) query for geo-information from user's desired location. In this searching types, user location is required to obtain for processing the query. The user location of this proposed system can have two types. The first one is current location which can obtain by using the Global Positioning System (GPS). The second one is user's desired location which can obtain from the user input. The coordinate of the desired start location is retrieved by searching in the proposed index structure based on the user input. The nearest query and range query are searched from the current location and from the user desired start location. According to the input query, the searching process is performed in the fourth step. The detail of the searching process for the query is explained in the next sections. The result obtaining from the searching process is displayed on the map with their respective geo-information.

4.2 Geospatial Keyword Query

Spatial keyword query plays an important role in location-based search. The geographical search based on the spatial and textual is required in searching of keyword that belongs to one or more locations. For example, find the bank within 2 kilometers from the user location or find the nearest ATM from the user location. The spatial and textual indexing is necessary for processing geo-spatial keyword query efficiently.

A place P that belongs to a set of spatial object S has the geo-location l and textual information t. The spatial keyword query $Q=<l, t>$ generally takes both the user location and keyword information as arguments, and retrieves the place that is spatially and textually relevant to these arguments. The spatial keyword query in this proposed system is performed for keyword query, nearest query and range query from current location or user desired start location.

4.2.1 Problem Statement of Geospatial Keyword Queries

The spatial Database *D* that contains a set of points ($p_1, p_2, \ldots, p_i$). Each point p in *D* has many attributes <p_{id}, p_{loc}, p_{et}, p_{mt}>. p_{id} is the identifier of the point, p_{loc} is spatial location of that point, p_{et} is the text information of that point with English language and p_{mt} is the text information of that point with Myanmar languages.

Nearest Geospatial Keyword Search problem involves determining the points in the spatial database which point is the nearest place to a given query point and given respective keyword. Nearest Spatial Keyword Query takes two arguments that defined as Q = (q_{uloc}, q_t). q_{uloc} is user location and q_t is a set of keywords that can be English or Myanmar language. The output result gives the nearest place Rn with respective q_t from q_{uloc} that is satisfied $Rn = \min\{\, d(p_i, q_{uloc}) \mid pi \in D\}$ where d is the distance. The distance of the points from the user location is analyzed based on the given query point and data point in spatial database to obtain the point which has the minimum distance from the user location. The nearest query can be seen in Figure 4.3.

Range Geospatial Keyword Query Q has three arguments and it is defined as Q = (q_{uloc}, q_t, q_r), where q_{uloc} is the user location, q_t is a set of keyword, q_r is the range. The result list *Rl* of range query is fetched if the points in the spatial Database include within the given range q_r and service of these points are matched q_t. The rule of range query is $d(p_{loc_i}, q_{uloc}) \leq q_r \; and \; (\,(p_{et} = q_t) or (p_{mt} = q_t))$. Figure 4.4 shows the range query from user's current location and user's desired location.

The problem of Keyword Search determines the points which match the given keyword. Keyword Query takes one argument q_w that is a set of keyword. The keyword can be English and Myanmar keyword. The result output q_{ke} can be a point or a set of points which match the q_w. In addition to in this keyword query, it also calculates the distance, and determines the nearest point to q_{uloc} among q_{ke}

The last query of the proposed system is to retrieve the nearest geospatial keyword query and range geospatial keyword query from the user desired location instead of user's current location. The problem of these queries is the same as nearest spatial keyword query and range keyword query which is discussed in above. In this query, the user location q_{uloc} is the user desired location which is given by user.

The nearest query and the range query from user desired location can be seen in Figure 4.3 and Figure 4.4.

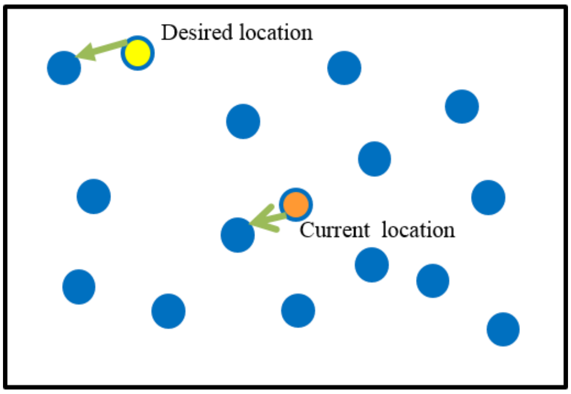


Figure 4.3 Nearest Query from User's Current Location or User's Desired Location

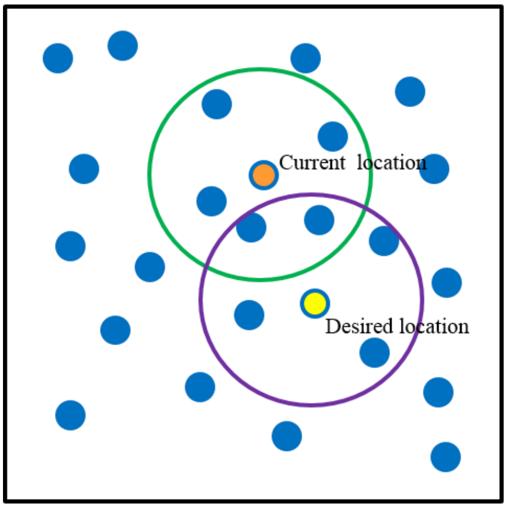


Figure 4.4 Range Query from User's Current Location or User's Desired Location

4.2.2 Nearest Geospatial Keyword Query

The nearest query search is the proximity search that retrieves the nearest object from the current location. The system of diagram of the nearest query is shown in Figure 4.5.

The input of the nearest query Q of the proposed system takes two arguments: $<q_{uloc}, q_t>$ where q_{uloc} is the user location obtaining from GPS and q_t is the user's input text information, that is, services such as restaurant, bank. The proposed index structure I that contains the service q_t is searched to decide which index structure is used for query processing the nearest query because the proposed index structure is constructed according to the services which already discussed in the previous chapter. Then, the distance between q_{uloc} and p_{loc_i} in the proposed index structure I is calculated and the distance is checked for searching the point which has the minimum distance to given q_{uloc}. Then, the nearest location Rn is displayed on the map with its coordinate and its text information.

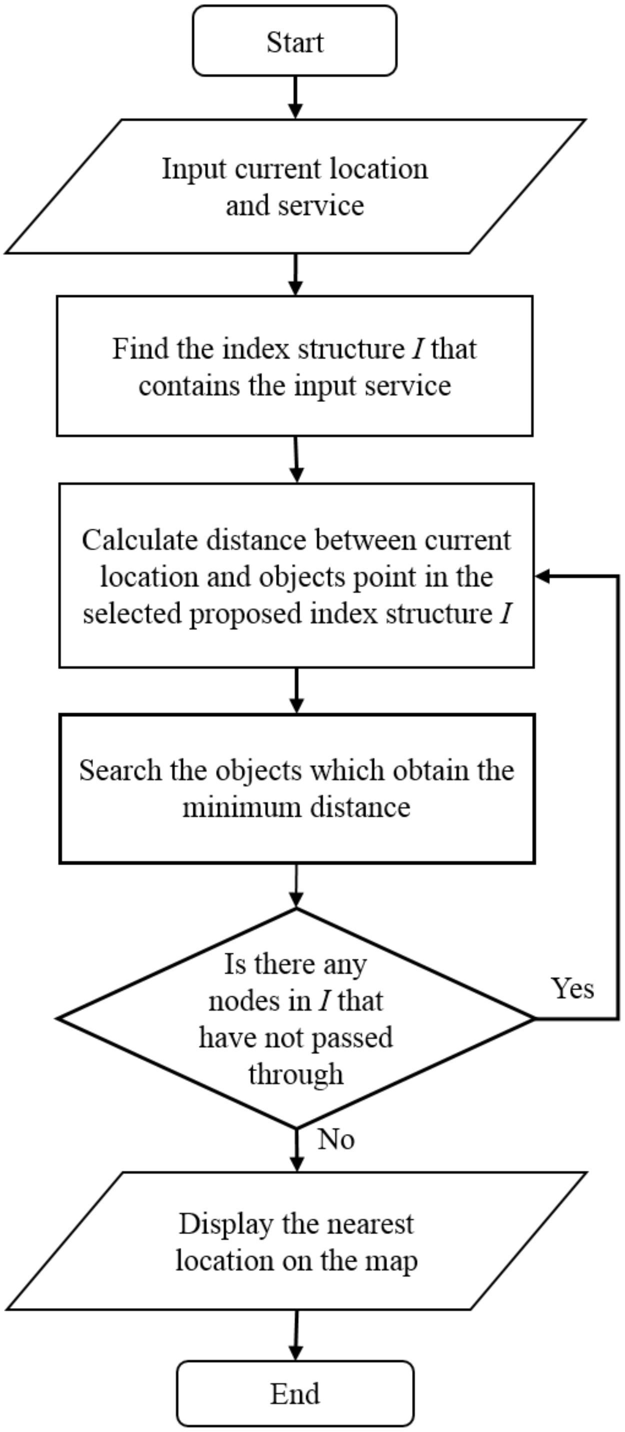


Figure 4.5 System Flow Diagram of the Nearest Query

4.2.3 Range Geospatial Keyword Query

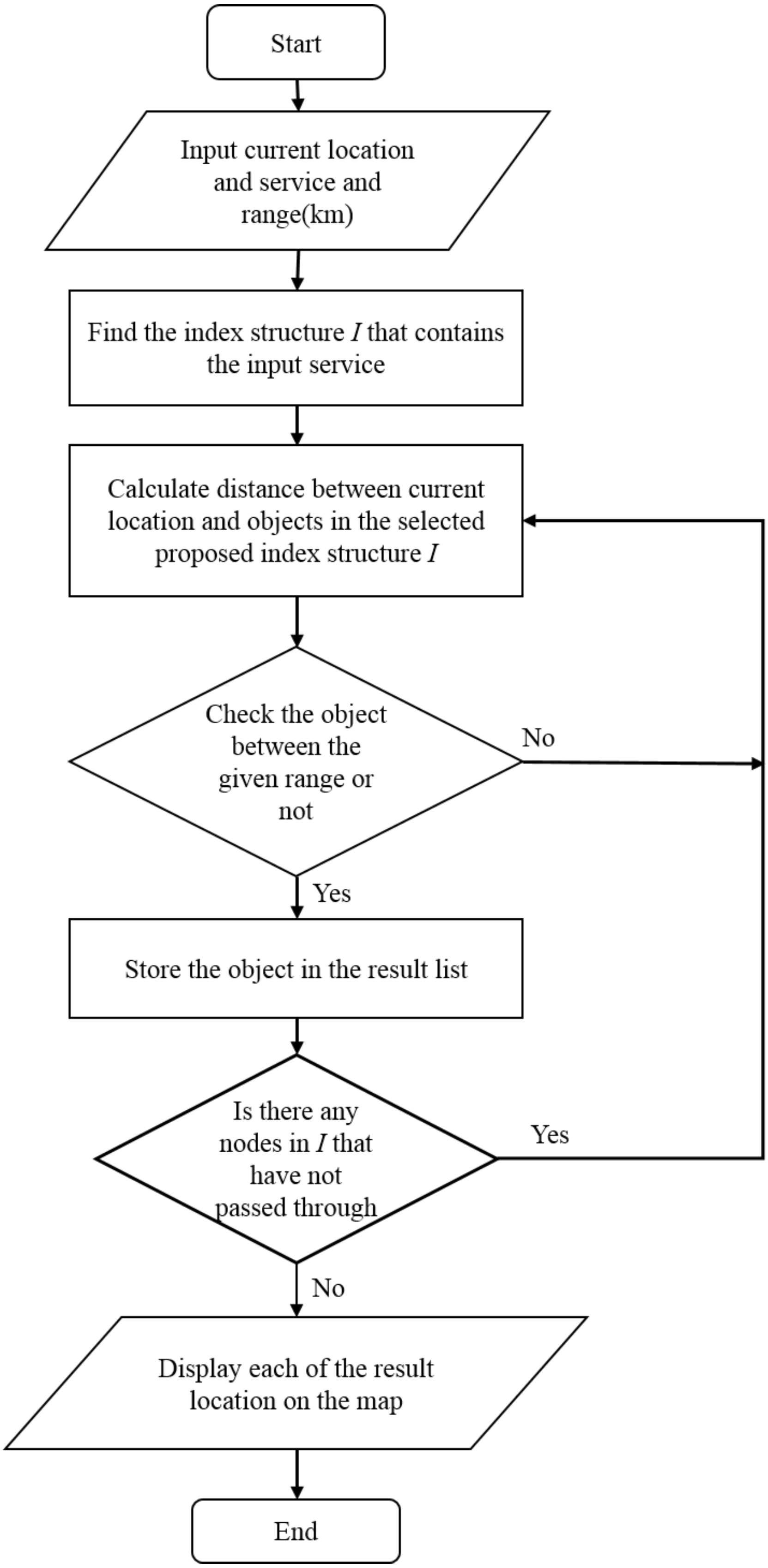


Figure 4.6 System Flow Diagram of Range Query

Range query of the proposed system finds the geo-points within the given range. The system flow diagram of Range Search is shown in Figure 4.6.

It takes the three inputs <q_{uloc}, q_t, q_r> where q_{uloc} is the current location, q_t is the services and q_r is the range. As the nearest query search, it finds the proposed index structure I based on the q_t for processing the range query. The proposed index structure I is already mentioned in the previous chapter. Then, it calculates the distance between each point p_{loc} in the proposed index structure I and the user's current location q_{uloc}. The range query retrieved the result list RL from D based on $d\left(p_{loc_i}, q_{uloc}\right) \leq q_r$ where d is the distance. Then, the distance is checked whether the point is contained within the given range or not. All the points within the given range are continuously retrieved. The information of the results is displayed on the map with their coordinate.

4.2.4 Keyword Query

The Keyword Query retrieved the geo-location that matches the user's input keyword. The system flow diagram of Keyword Query is shown in Figure 4.7. In this search, it needs to process three main steps: (1) find the geo-object that relevant to the input keyword, (2) calculate the distance (3) decide the nearest geo-object.

Firstly, it takes user's input keywords. The input keyword can be Myanmar and English keywords and it may be the exact location name or name of the service. Then, the input keywords are searched in the keyword of service list to decide that the keyword of services is contained in the input keywords or not. If it contains in the input string, it determines that which proposed index structure I is used during the searching process. Then, the geo-information is searched in the proposed index structure I that relevant to the service keyword. If the service keyword does not contain in the input keywords, it requires searching in all of the proposed index structure to retrieve the objects that is related to the input keyword query. Therefore, the worst case in this condition can occur if it is needed to search in all of the proposed index structure.

Next, the distance is calculated between the result geo-objects and the user's current location. This query gives the result geo-objects with the distance so that users can know the distance between their desired search place and their current location.

Then, it determines which objects are the nearest to the user's current location among the result objects. Finally, the resulted geo-objects are displayed on the map with the distance and the nearest objects to user's current location.

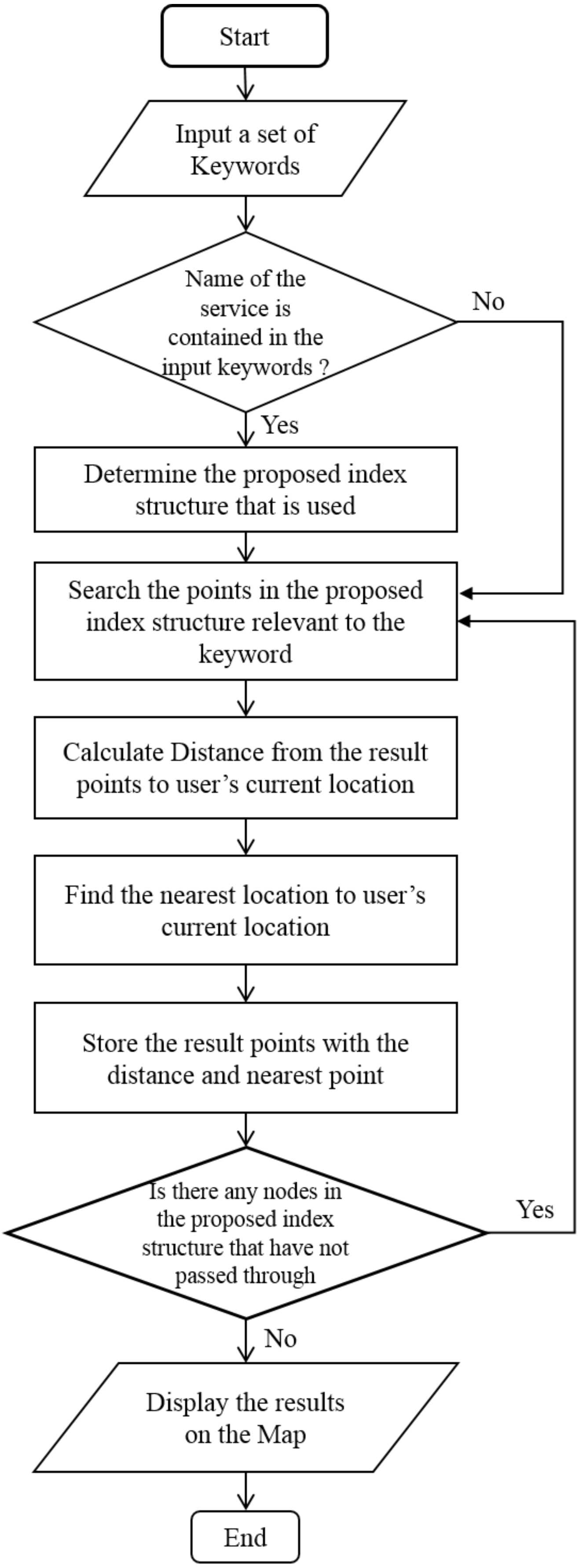


Figure 4.7 System Flow Diagram of the Keyword Search

4.2.5 Query for Geo-Information from User's Desired Location

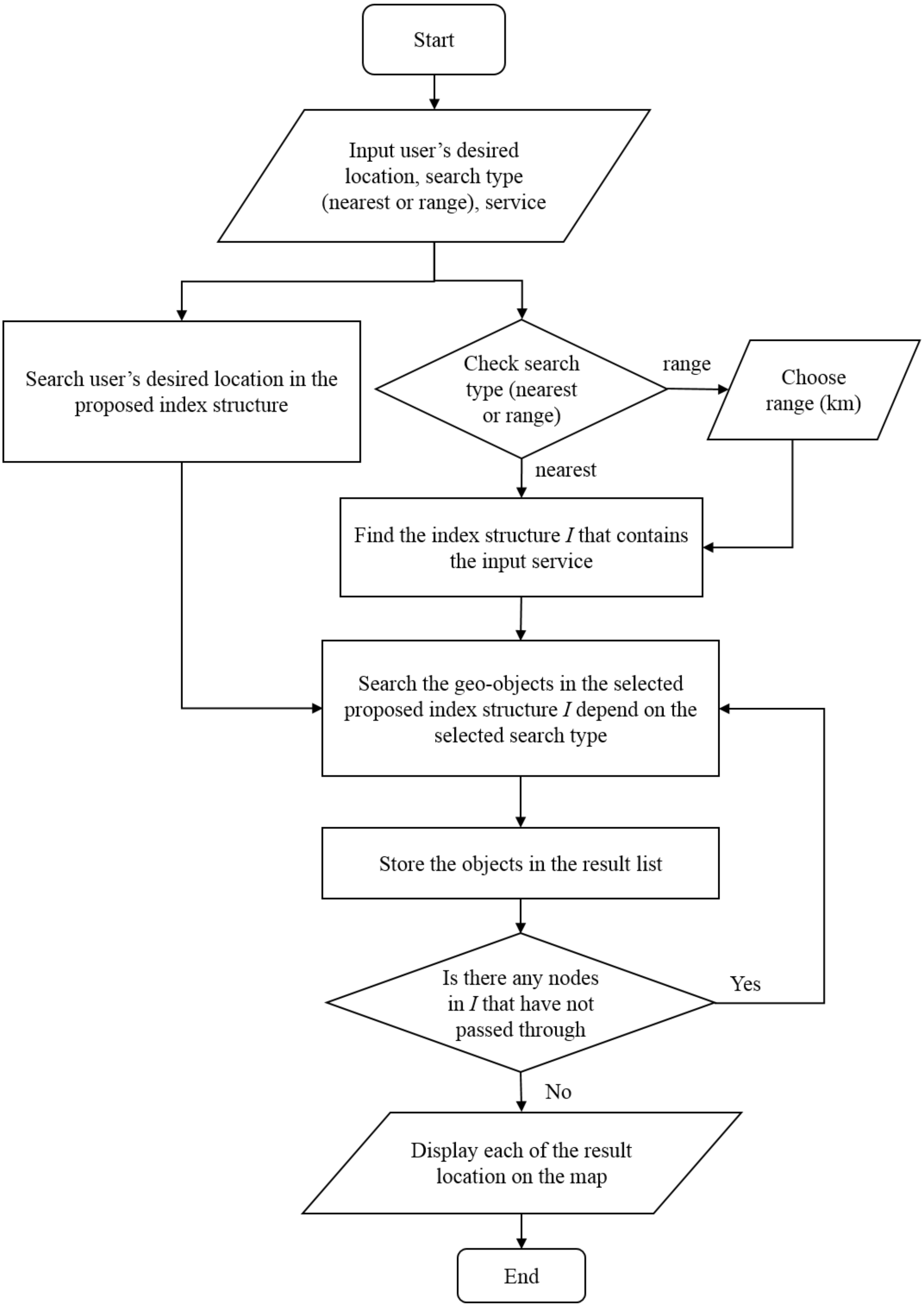


Figure 4.8 Retrieving the Nearest Query and Rang Query from the User's Desired Location

This section discusses the retrieving the geo-information from the user's desired location. Normally, most of the geo-location is retrieved from user's current location. It is intended to implement for users to obtain in advance the geo-information around the desired location for predicting the travel location. In this query, Nearest Keyword Query and Range Keyword Query can be processed from user's desired location. The system flow diagram is shown in Figure 4.8.

It takes three arguments$(q_{dloc}, q_{ser}, q_{stype})$ where q_{dloc} is the user's desired place, q_{ser} is the service that searches for nearest or range query and q_{stype} is the search type (nearest query or range query).

At first, it retrieves the location of the desired place d_{loc} from the spatial database D. Then, it searches the geo-location based on the search types. If the selected search type is range keyword query, the range q_r is required for searching process. The proposed index structure, which is used in searching process, is selected based on the input service q_{ser}. the distance between the user's desired location d_{loc} and the location of the points p_{loc} is calculated for processing the nearest query or range query. The procedure of the nearest query and range query is the same as the discussion above. The resulted locations are displayed on the Map.

4.3 Algorithms of Geospatial Keyword Query

In this section, the algorithms of geo-spatial keyword queries are discussed. For nearest query and range query, the distance calculation is essential to fetch the geo-information. The distance calculation in this research is introduced. Then, Distance calculation algorithm, nearest keyword query algorithm and rang keyword query algorithm are discussed in this section.

4.3.1 Distance Calculation

Geodesic calculation plays an important role in many geospatial applications. For example, finding the distance between two places. The Pythagorean theorem (in Figure 4.9) is used to obtain the distance between two locations on a flat plane.

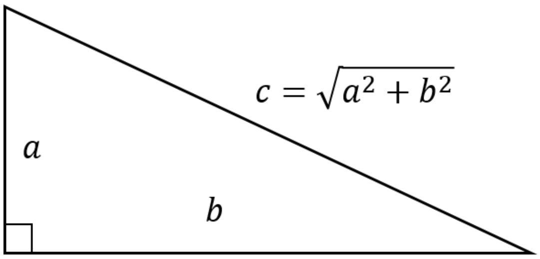


Figure 4.9 Pythagorean Theorem for Measuring Distance

But, it does not work well for the locations on the earth which is not a flat plane. The earth's shape like a sphere and the coordinates value of the point do not directly map to the units of distance. The latitude and longitude line on the earth can be seen in Figure 4.10 and Figure 4.11. There are several formulas that are used for calculating the distances between the two points on the sphere such as spherical law of cosines, Haversine formula.

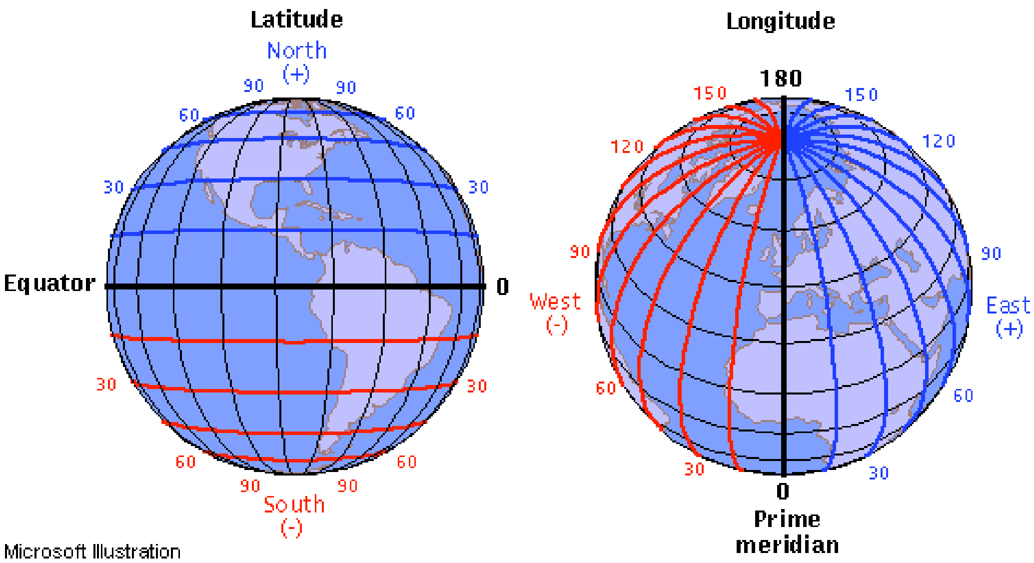


Figure 4.10 Left: Meridian (Line of Longitude), Right: Parallels (Circle of Latitude)

Euclidean geodesic approximation [2] built by V. Agafonkin is used in this proposed system. It is faster than the Haversine formula. This formula combines the two formulas with Pythagorean theorem to perform a Euclidean approximation of the distance between two points by using latitude and longitude. The equation of Euclidean geodesic approximation is shown in Equation 4.1.

$$d = \sqrt{\partial x^2 + \partial y^2} \qquad (4.1)$$

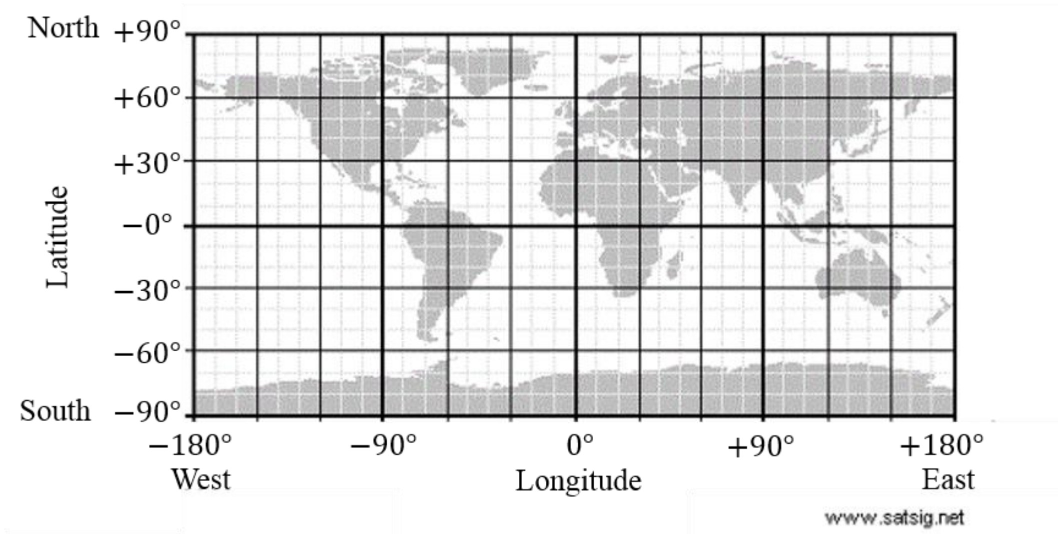


Figure 4.11 Latitude and Longitude with their Degree

The variable ∂x (see equation 4.2) is to approximate the distance between two longitudes. For the differences of the longitudes, the length of the parallel with 360-degree circle based on its latitude, starting with the length of the equator which is 24901 miles, and scaling with the latitude's cosine.

$$\partial x = 24901 \frac{|lng_1 - lng_2|}{360} cos(\frac{lat1 + lat2}{2}) \quad (4.2)$$

The variable ∂y (in equation 4.3) is to approximate the distance between two latitude. It measures with the length of the meridian (about 12430 miles).

$$\partial y = 12430 \frac{|lat_1 - lat_2|}{180} \quad (4.3)$$

4.3.2 Nearest Keyword Query Algorithm

The procedure of Nearest Keyword Query Search is described in NearestKQSearch and NNsearch Algorithm which are shown in Figure 4.12. NearestKQSearch Algorithm takes three parameters nnser (services), currlat (current latitude), currlng (current longitude). The output returns the nearest point Rnn. Step 1 finds the B-tree, whose service is matched the input service. To search the nearest point, NNsearch algorithm is called with three parameters in Step 2. The first one is the root node of the selected B-tree. The second and third parameters are latitude and longitude values of the current location.

Algorithm : Nearest Keyword Search

```
Input : nnser : Service to search
        currlat : current latitude
        currlng : current longitude
Output : Rnn : nearest point

NearestKQSearch (nnser, currlat, currlng)
    sbtree ← Btree that service match the nnser
    Rnn ← Call NNsearch (sbtree.root, currlat, currlng)
    return Rnn

-------------------------------------------------------------------------------

NNsearch (root, ulat,ulng)
    initialize mindis to10000
    for i=0 to root.count
        disnn ← Call calDis ( root.key_i.plat, root.key_i.plng, ulat, ulng)
        if disnn is less than mindis
            mindis ← disnn
            resultnn ← root.key_i
        End If
    End For
    If root is not leaf
        For j=0 to root.count
            If  root.child_j is not null
                call NNsearch (root.child_j, ulat, ulng)
            End If
        End For
    End If
    return resultnn
```

Figure 4.12 Nearest Keyword Search Algorithm

NNsearch Algorithm operates for searching the nearest location. From step 2 to step 8, the distance between the current location and the elements of the root is searched and then it determines which point has the minimum distance from the current location. For calculating the distance, it calls the calDis algorithm which takes the coordinate of the current location and the coordinates of the element. The Distance Calculation Algorithm is shown in Figure 4.13. Then, the distance is checked whether it is the minimum distance or not. If the point has the minimum distance from current location, it is stored in the result variable. Determining the minimum distance is operated from step 4 to step 6. Step 9 to step 15 checks the node has child or not. If the node has the child, the child node is picked and the point, which obtains the minimum distance, is continuously searched for all the nodes.

Algorithm : calDis (lat1, lng1, lat2, lng2)

```
x ← 12430 * (abs(lat1-lat2)/180)
 y ← 24901 * (abs(lng1-lng2)/360) * cos( (lat1+lat2)/2)
dist ← sqrt( pow(x,2), pow(y,2)
return dist
```

Figure 4.13 Distance Calculation Algorithm

4.3.3 Range Keyword Query Algorithm

The Range Keyword Search Algorithm is shown in Figure 4.14. Algorithms of RangeKQSearch and RangeSearch retrieve the geo-information based on the user given range. RangeKQSearch Algorithm takes surser (services), currlat (current latitude), currlng (current longitude) and surrange (range) as arguments. The index b-tree for using during the searching process is determined based on the user's input services. Then, RangeSearch Algorithm is called with four parameters for searching the list of geo-locations from the current location based on the given range. The four parameters are the root node of the index b-tree, latitude of the current location, longitude of the current location, and range. This algorithm returns the list of point that contains within the given range.

Algorithm : Range Keyword Search

```
Input : surser : Service for searching
        currlat : current latitude
        currlng : current longitude
        surrange : range
Output : Rsur : All points within given range
```

RangeKQSearch (surser, currlat, currlng, surrange)

```
sbtree ← Btree that service match the surser
Rsur ← Call RangeSearch (sbtree.root, currlat, currlng, surrange)
return Rsur
```

RangeSearch (root, ulat,ulng, r)

```
For i=0 to root.count
    dissur ← Call calDis ( root.key_i.plat, root.key_i.plng, ulat, ulng)
    if dissur is less than r
        resultsurr ← root.key_i
    End If
End For
If root is not leaf
    For j=0 to root.count
        If  root.child_j is not null
            call RangeSearch (root.child_j, ulat, ulng, r)
        End If
    End For
End If
return resultsurr
```

Figure 4.14 Range Keyword Search Algorithm

In Range Search algorithm, retrieving the list of point within the user's given range is processed from step 1 to step 8. To check the point is within the given range or not, the distance between the current location and the points in root node is calculated by using the calDis Algorithm. Then, the distance is checked within the given range or not. If it is within the given range, it is added to the result list. This process is continuously operated until it reaches the leaf node. Thus, if it is not the leaf node, it still searches the point in the elements of the child node that is contained within the range or not.

4.3.4 Keyword Query Algorithm

In keyword query, it required input keyword. The output of keyword query in this proposed system gives not only the geo-objects with distance but also the nearest object among them. The distance is calculated between the geo-objects that match the input keyword and user's current location. KQSearch Algorithm is used to perform these operations. Figure 4.15 shows KQSearch Algorithm.

Algorithm : Keyword Search

Input : keyword

Output: geo-objects that relevant to input keyword

KQSearch (kwname)

1. Take input keyword (*kwname*) as argument.
2. Search the service name which is contained in the *kwname* or not.
3. If it contains, *kwname* is searched in the proposed index structure of that service.
4. If not, *kwname* is searched in the proposed index structure for each service.
5. If search item is found,
    (a) calculate the distance from that item to user's current location
    (b) store the geo-object with the distance into the result list (*kelist*).
6. Find the nearest location (*nl*) among *kelist*.
7. Give *kelist* and *nl* to display on map.

Figure 4.15 Keyword Search Algorithm

4.4 Summary

This chapter presents the performance of the proposed system on spatial keyword query processing. The overview of the system is described in detail. Then, it explains the problem statement of the spatial query processing in proposed system. It also describes the flow of the spatial keyword query. In this spatial keyword query, the nearest keyword query, the range keyword query, and keyword query are performed. The flow of these queries are explained. In addition, it also discusses the operation of the nearest query and range query from user’s current location and user’s desired location. Euclidean geodesic approximation is presented in this chapter to calculate the distance. Moreover, it expresses the algorithm of nearest keyword query, range keyword query and the procedure of the keyword query in this chapter.

CHAPTER 5
IMPLEMENTATION AND EVALUATION

This proposed system is developed for the mobile users to search efficiently and effectively the geo-information anywhere and anytime. Location information is important for the visitors to know where they are and where their desired places are. This proposed system provides the user to know their location and to search the nearest geo-information and geo-information of the surrounding area from the user current location. It can also find the location with precise location name. In addition, it is implemented for searching the location not only from user current location but also from the user desired start location. Therefore, the user can forecast the geo-information about the nearest and surrounding area from the desired location. So, it can provide the traveler to plan the trip during the travelling by pre-search the geo-information which they want to go. It is implemented on the android mobile application. This application can also operate both online and offline condition. So, it can avoid the location problem during the weakness of the internet connection or not available. The focus area of this application is Yangon Region which is a famous commercial city in Myanmar and has many colonial buildings. Many people travel to Yangon for business or for visit. It provides for both tourists and local people to avoid the problems such as missing the location.

This chapter represented the detailed system implementation of the proposed system. In system implementation, it is discussed about the implementation of the system on the Android mobile system; the access to the current location and the offline map of the proposed system; the collection of the geo-data and the database design of the proposed index structure; the retrieving the geo-data. Moreover, the processing of the spatial queries on the collected geo-data and the system evaluation are also described in this chapter.

5.1 Requirements of System Implementation

This proposed system is developed on the android mobile system. In this implementation, it needs to access the current location of the user for operating the nearest and range query. Although this system mainly focuses on the applying offline android mobile devices, it can also apply on the online android mobile devices. The requirements of the system implementation for the proposed system are:

(1) Android version 4.4 or above
(2) Google Play service 16.0.0
(3) Android SDK (API 19 or higher)
(4) Mapbox map SDK 7.2.0

5.2 Accessing Current Location in Android Mobile Devices

Android application provides the user for solving the daily problem like obtaining the location. Location Providers are used to access the location of the mobile devices. There are two location providers for obtaining the user's current location. These location providers are GPS Location Provider and Network Location Provider.

5.2.1 GPS Location Provider

GPS location provider uses the satellites to decide the location of the android devices. The information is transmitted from the satellites to the receiver. The transmitted information is used for measuring the distance to each satellite by timing how long it takes to obtain a transmitted signal. GPS location provider produces the precise geo-location data. One of the drawbacks is consuming more battery power than other location providers. This can cause the long time during the receiving and processing the location data. The time to obtain the first location data may take over a minute and it can differ from device to device or between different versions of Android.

5.2.2 Network Location Provider

Network location provider utilizes Wi-Fi network location and cell tower location to access a location. It gives the value from the nearest tower location. The time of network provider for obtaining the first location data is less than the GPS provider.

5.2.3 The reason for using GPS Location Provider

This proposed system is mainly intended to apply in offline version. Thus, the current location of the device is required to work in an offline. In this proposed system, GPS location provider is used to access the location of the devices although Network location provider is faster than GPS location provider in accessing the geo-location. Because Network location provider will not obtain the location when the connectivity of network is poor. But it has some weaknesses when using a GPS location Provider. It can take time for obtaining the geo-location in an indoor region. Thus, it sometimes takes over a minute to acquire the current location data in the first time. The advantage is that GPS location provider can obtain more accurate location data than the network location provider because network location provider produce the location based on the cell tower and it will give the nearest tower location.

5.3 Offline Map

The Map to display on both online and offline versions use the Mapbox Map in this proposed system. Offline maps are becoming popular and Google Maps is also permitting the user to save the map for offline access. Mapbox effectively provides the offline map that has raster map and vector map.

Raster map splits into square tiles. The images for each tile and each zoom level within that tile have to be stored in the map. The pre-generated image is required to download from the database to fetch an image of a certain tile. Several images are required to store in raster map so that its causes memory consumption.

Another kind of offline map is the vector map which is different from the raster map and shows a map in vector format. Vector format can save the memory consumption than raster format. For example, if a map may consume 1GB of memory in raster format, it may only consume 100MB in a vector map.

For offline map in this proposed system, a vector offline map is used. The area of the region which can be downloaded is limited because the Mapbox vector map for android has some limitations in offline use.

5.3.1 Limitations of Offline Mapbox

Mapbox map offers multiple regions to download and uses in offline but the total download in offline is limited with a maximum tile count "ceiling" over all downloaded regions [88]. If the tile ceiling is up to 6000 tiles, it needs to pay the downloaded region. The tiles size on disk will differ based on the location which is downloaded and the selected style. The Tile Count Estimator[1] can be used for calculating the number of tiles that required for offline use.

5.3.2 Tile in Offline Map

An image of the offline map is sliced into tiles that are organized into numerous zoom levels. Zoom levels start from 0. The dimensions of the map image are 1x1 pixels at zoom level 0. The size of the image increases twice at each next zoom level and it can be seen in Figure 5.1 [89].

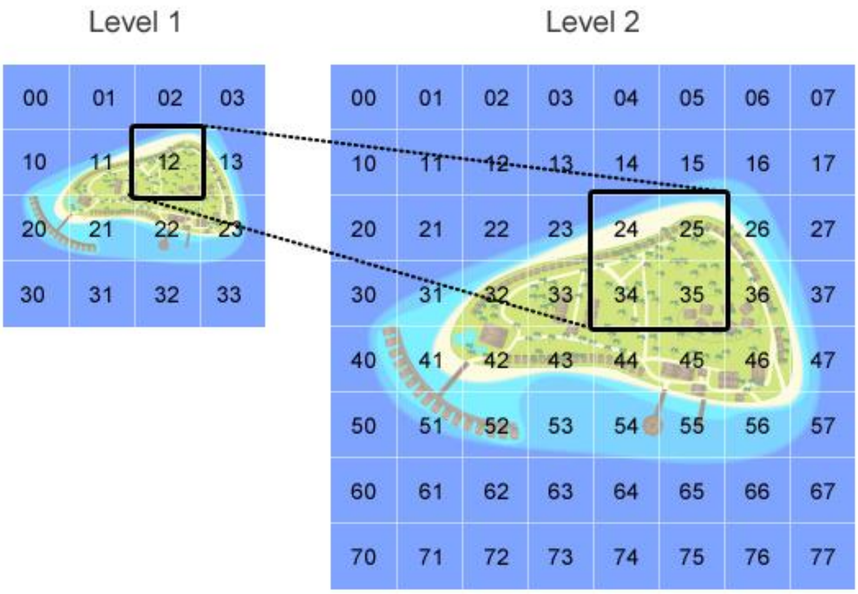


Figure 5.1 Image Size at Zoom Level 1 and 2

A map can contain layers that hold the map objects. Layers can be visible or invisible. The object of the layer will not show on the map if the layer is invisible. Map objects can be shown on the map and can be added to any layer. The coordinates of the object are set in pixels. For defining the location of object (in Figure 5.2) [89], coordinates utilize the image which originally used for tile generation.

[1]https://docs.mapbox.com/help/interactive-tools/offline-estimator/

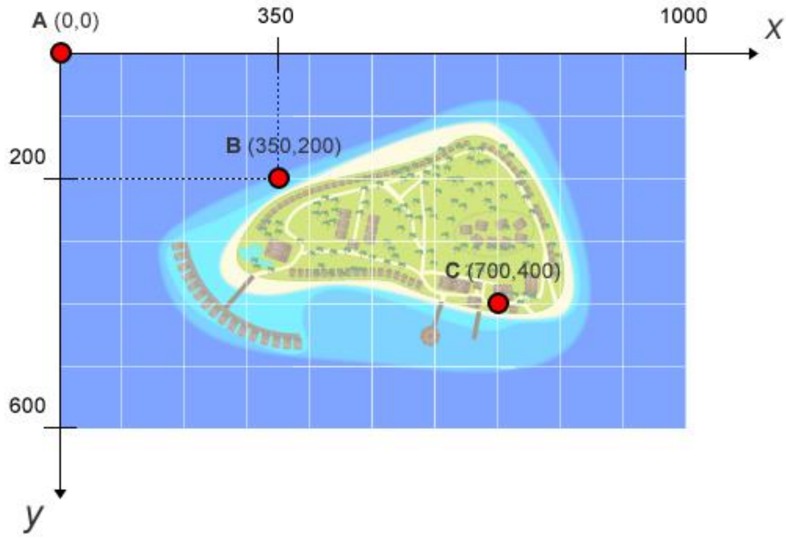


Figure 5.2 Defining the Object Location

5.3.3 Offline Map Download Area of the Proposed System

The area of this proposed system mainly focuses on the Yangon Region. The download area of Yangon offline map in this system is from coordinate of the top left location (17.067287, 95.995789) to coordinate of the bottom right location (16.628297, 96.341858). The zoom level which can apply on this Yangon offline map is level 5 to level 15. Because the tile for Mapbox offline map cannot exceed 6000 tiles for free version. For example, if zoom level 5 to 16 is used for this download offline map, the tile of this download area is 6434 tiles. So, it exceeds the limitation of tile in offline map. So, zoom level 5 to 15 is chosen with the total tile 1674 tiles for download area of Yangon offline map region.

5.4 Preparing the geo-data for Spatial Database

Geo-data of Yangon region is collected from the google earth and OpenStreetMap using QGIS. Then, the collecting geo-information is checked and prepared for achieving more accurate location of the geo-data. In addition, geo-data of this proposed system have two languages, Myanmar and English. The collecting geo-data are mostly with English. Thus, the geo-data are translated to Myanmar languages to use in this system. There are 72 categories of services and nearly 10000 geo-data in the spatial database. The available services can be seen in Table 5.1.

Table 5.1 Categories of Available Services

Advertising Agency	Monastery	Car parts & Accessories
Airline	Museum	Car Wash
Airport	Pagoda	Car Workshop
Air Ticket	Parks	Copy Shop
ATM	Pharmacy	Cosmetics Shop
Bag Shop	Photo Studio	Florist Shop
Bakery & Cafe	Restaurant & bar	Furniture
Bank	School	Market
Beauty Salon	Shoe Shop	Mobile Sales & Services
Book Store	Shopping Center	Veterinary clinic
Bus Stop	Stadium	Printing Service
Car sales & services	Training School	Pet Shop
Church	Travel & Tour	Temple Chinese
Cinema	University	Delivery Service
Clinic	Language School	Electronic Store
Driving School	Cloth Shop	Exchange Center
Embassy	Computer Sales & Services	Government
Fitness Center	Private School	Optical clinic
Gas Station	Kinder Garten	library
Gold and Jewellery	Bridge	Musical Instrument
Hospital	Convenience Store	Pool
Hotel	Police Station	Music Studio
Hostel	Art Gallery	Sign Shop
Guest House	Bus Ticket Agency	Yoga

5.5 Geo-data Dataset of Yangon Region

The collected geo-data is stored in the spatial database. Normally, SQLite database is widely used in developing the android application. In this proposed system, the database that contains all the collected geo-data is stored in the mobile devices. In addition, the download offline Yangon map is also maintained in the mobile devices. The memory of the mobile devices is limited and it has a low computational capacity than the personal computer. Thus, utilizing the smaller file size is better in order to obtain a good performance on the mobile devices. In this proposed system, the csv file is used to store the geo-data on the mobile devices. The file size of SQLite database is larger than the file size of CSV file. For example, CSV file size is 19MB when SQLite database is 23MB. The sample geo-dataset of the Yangon region is shown in Table 5.2 with their coordinates and text information. Table 5.3 shows the sample data with the number of nodes in each service.

Table 5.2 Sample Geo-Dataset

PID	Latitude	Longitude	Name (Eng)	Name (Myan)	Service (Eng)	Service (Myan)
ad1	16.8498	96.12841	Ad Mar Digital Marketing Agency	အဲဒ်မာဒစ်ဂျစ်တယ်မားကတ်တင်း အေဂျင်စီ	Advertising Agency, Advertising Services	ကြော်ငြာအေဂျင်စီ, ကြော်ငြာဝန်ဆောင်မှု
ad2	16.83435	96.16752	ADK Myanmar Advertising Agency	အေဒီကေမြန်မာကြော်ငြာလုပ်ငန်း	Advertising Agency, Advertising Services	ကြော်ငြာအေဂျင်စီ, ကြော်ငြာဝန်ဆောင်မှု
ad3	16.7785	96.17133	Amara Digital Marketing Agency	အမရာဒစ်ဂျစ်တယ်မာကတ်တင်း အေဂျင်စီ	Advertising Agency, Advertising Services	ကြော်ငြာအေဂျင်စီ, ကြော်ငြာဝန်ဆောင်မှု
atm5	16.77925	96.1581	AGD ATM	အေဂျီဒီအေတီအမ်	ATM	အေတီအမ်, ငွေထုတ်စက်
bank195	16.8625	96.06601	Global Treasure Bank	ကမ္ဘာ့ရတနာဘဏ်	Bank	ဘဏ်
rest103	16.77697	96.14622	Fuji Japanese Restaurant	ဖူဂျီဂျပန်စားသောက်ဆိုင်	Restaurant, Japanese Restaurant	စားသောက်ဆိုင်,ဂျပန်စားသောက်ဆိုင်
rest153	16.84462	96.15238	Inya Bar	အင်းယားဘား	Restaurant, Bar	စားသောက်ဆိုင်,ဘားဆိုင်

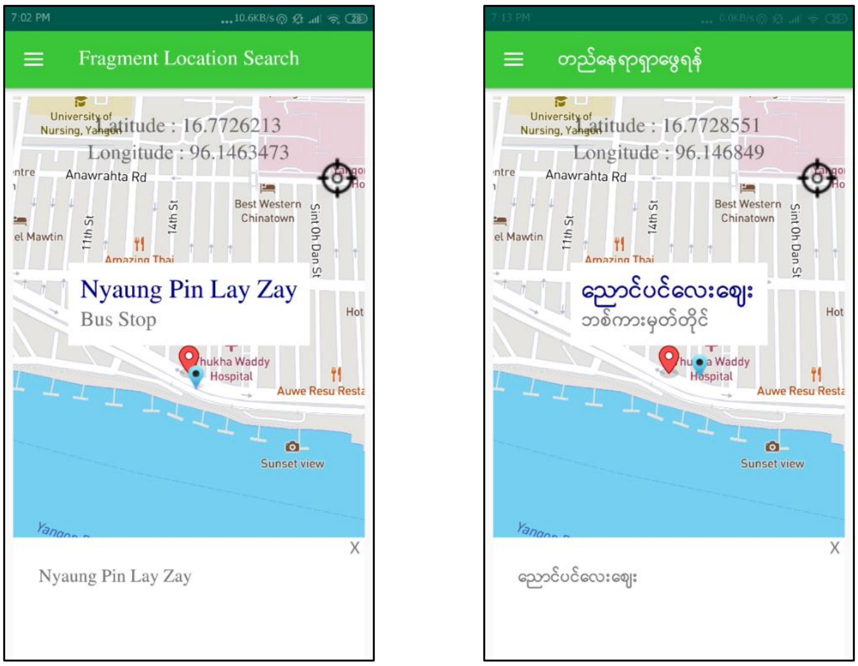
Table 5.3 Sample Data with the Number of Nodes in Each Service

No.	Services	Number of Nodes
1	ATM	188
2	Bakery & Cafe	234
3	Bank	409
4	Bag Shop	92
5	Beauty Salon	211
6	Book Store	81
7	Bus Stop	1942
8	Bus Ticket Agency	110
9	Car sales & services	119
10	Church	46
11	Cinema	42
12	Clinic	110
13	Computer Sales & Services	174
14	Convenience Store	163
15	Cosmetics Shop	105
16	Embassy	51
17	Exchange Center	94
18	Fitness Center	62
19	Gas Station	78
20	Gold and Jewellery	56
21	Hospital	75
22	Hotel	203
23	Language School	105
24	Market	163
25	Mobile Sales & Services	79
26	Monastery	235
27	Museum	16
28	Musical Instrument	46
29	Optical clinic	54
30	Pagoda	67
31	Parks	44
32	Pet Shop	91
33	Pharmacy	129
34	Restaurant & bar	505
35	School	1090
36	Shopping Center	112
37	Travel & Tour	192
38	University	64

5.6 Implementation for User Interaction

The design of the UI in this system has mainly two parts. The first part is to change the languages. In this application, it can use two languages, Myanmar and English, for processing the queries. The UI design for changing languages is simple. Users can easily select the languages they want to use by choosing from the two radio buttons. After selecting the language, the system sends it to all the pages to change the languages with the selected one. Then, the application can apply with the desired selected language to operate the searching process.

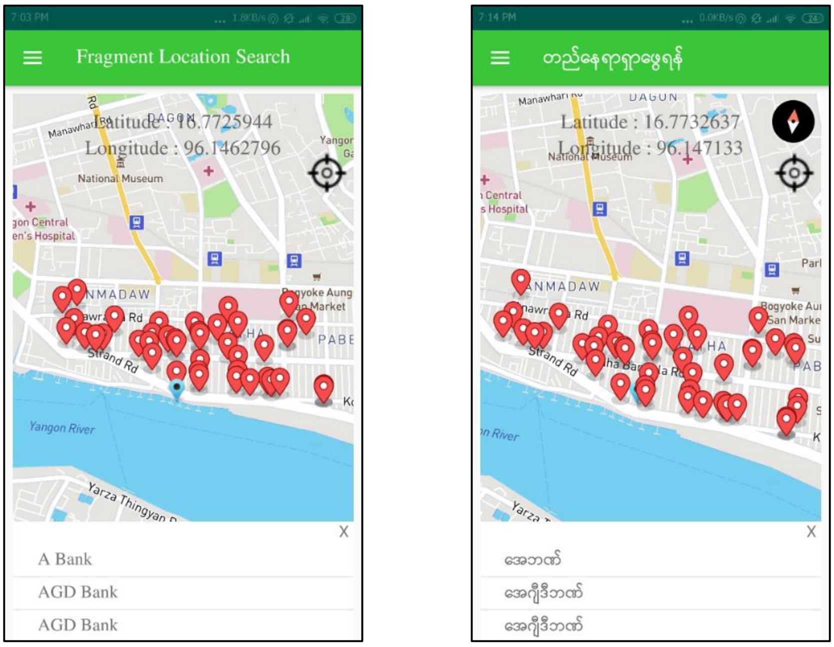
The second part of UI design supports the users to process the available queries and to obtain the offline map. The user can choose the actions which they want to perform. The available queries have nearest query, range query, name query and extracting the geo-information from desired location. If the user chooses the nearest query and range query, these query processing is operated from the user's current location. UI design for these two queries are user-friendly design. In both nearest query and range query, the user location is accessed from GPS on the mobile devices and then gives to the latitudes and longitudes textboxes. The user location is accessed in every two seconds. Then, the name of the service is required to fill for performing these queries. The difference in these two queries is that rang query is essential to give the range to operate.



(a) with English languages (b) with Myanmar languages

Figure 5.3 Nearest Keyword Query from User's Current Location

After inserting the required data, the searching process for these queries is operated in the proposed index structure. Then, the results of the nearest query and range query are retrieved and displayed on the map with the geo-location and respective text information. The result of nearest query can be seen in Figure 5.3 with both Myanmar and English languages. The range query's result is shown in two languages in Figure 5.4. In these two figures, user's current location can be seen with the blue location icon and the result output are shown with the red location icon.



(a) with English languages (b) with Myanmar languages

Figure 5.4 Range Keyword Query from User's Current Location

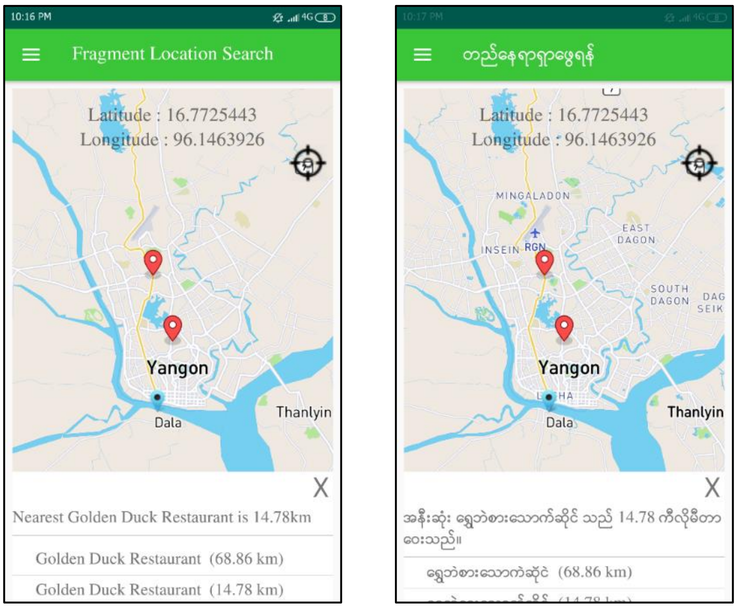
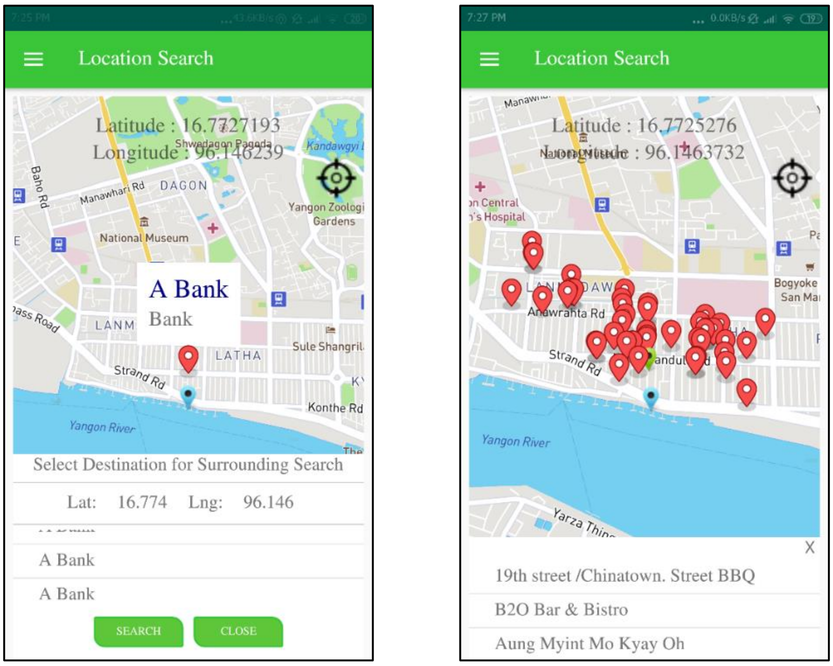


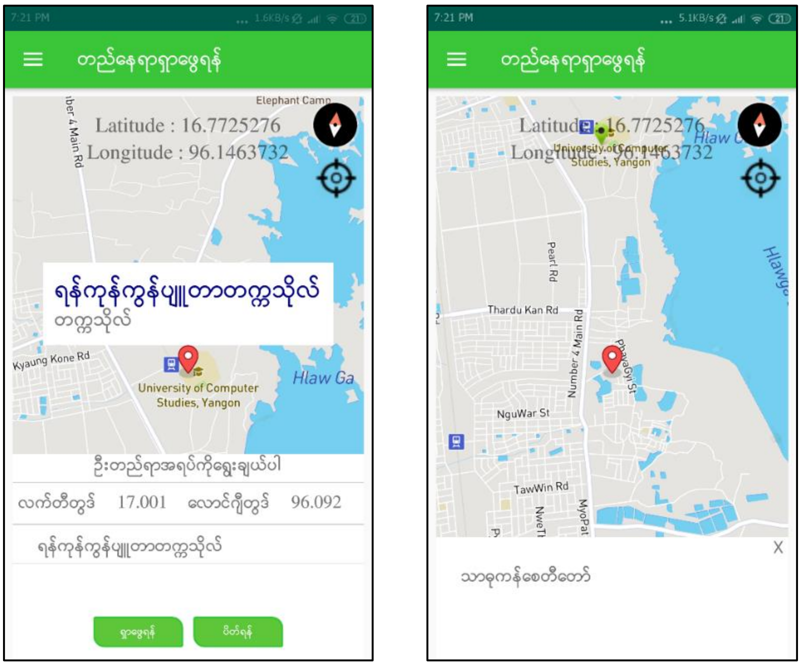
Figure 5.5 Result of Keyword Query

The next query is the keyword query. It only needs to give one input that is the name of the location to search. The result of the keyword query is shown in Figure 5.5. It retrieved the geo-information with the distance that far from the current location. The user can also obtain the nearest location among these result geo-location.



(a)User's desired start location (b) the result of geo-location

Figure 5.6 Retrieving Geo-location from User's Desired Location in English



(a)User's desired location (b) the result of geo-location

Figure 5.7 Retrieving Geo-location from User's Desired Location with Myanmar language

The last query is retrieving the geo-location of nearest and range query from the user's desired location instead of user's current location. Thus, it has the textbox for acquiring the user's desired location, the two radio button for choosing query (nearest or range) and the textbox to request the service which is searched for the selected query.

Based on the input data, the system first shows the user's desired location which is inserted by user. It is shown in Figure 5.6 (a) and Figure 5.7 (a). The Figure 5.6 (a) shows the query which is processed with English language. Figure 5.7 (a) shows the query which is processed with Myanmar language. The result of the user's desired location can be one or more that can be seen in figure 5.6. The user can select the place which is used as the desired start location. Then, the output result is retrieved based on the selected desired location and input service. The result of the query can be seen in Figure 5.6 (b) and Figure 5.7 (b). The user's desired location is shown with the green location icon.

These results are displayed on the online map if the offline map has not download. The last activity in this application is to download the offline map. To show the result of the quires on the offline map, the offline map is first required to download. This system can download the Yangon map for offline use.

5.7 Evaluation of the system

This system is tested on the Redmi Note 4 with MIUI version 10.2.2.0 (Android version: 6.0). The evaluation of the proposed index structure is compared with R-tree index. In this evaluation, there are two parts in experimental discussion. In the first part, the construction time of the proposed index structure is discussed. In the second part, the evaluation of the searching time is presented based on result that obtain from the spatial keyword queries processing, nearest keyword query, range keyword query, and keyword query. The searching time complexity is O(log n). Moreover, it also explained how difference in searching time between searching from the user's current location and user's desired location.

5.7.1 Experimental Result for Constructing the proposed index structure

The construction of index structure is compared with R-tree's construction time. The construction time is measured with the milliseconds. In this experiment, over 8000 geo-data is executed 10 times. Table 5.4 shows the experimental result of tree construction time. In every execution, the construction time of the proposed index structure is more than the construction time of R-tree. The proposed index structure is constructed based on the service. In addition, it also built the inverted file in each node for indexing the geo-information. R-tree is built only on the geo-location. So, the construction time of the proposed index structure takes much longer than R-tree.

Table 5.4 Comparison of Tree Construction Time

No. of Execution	R-tree (ms)	Proposed Index Structure (ms)
1	1809	4853
2	1791	4683
3	1789	4707
4	1788	4520
5	1792	4624
6	1813	4675
7	1736	4594
8	1794	4690
9	1798	4819
10	1759	4713

Figure 5.8 Processing Time for Constructing Index Structure

Then, the construction time is also evaluated based on the number of objects. The processing time of constructing the proposed index structure can be seen in Figure 5.8. Its processing time is higher than that of the R-tree. Based on the number of geo-data, the construction time of proposed index structure is increased.

5.7.2 Experimental Result of Spatial Keyword Query

The experimental result of spatial keyword query is measured based on the searching time using the proposed index structure compared with the R-tree. The searching time of nearest keyword query, range keyword query and keyword query is discussed in detail according to the processing.

5.7.2.1 Evaluation of the Nearest Keyword Query

The evaluation of the nearest keyword query is considered as three parts. In this processing, nearest keyword query can be obtained from user's current location or from user's desired location. At first, the evaluation of nearest keyword query is from the current location. Then, the processing of the nearest keyword query from user's desired location is presented. Lastly, the processing between the user's current location and the user's desired location is compared as the searching time.

Table 5.5 Searching Time of Nearest Keyword Query from Current Location

No. of Execution	R-tree (ms)	Proposed Index Structure (ms)	Services
1	170	30	Bank
2	163	6	University
3	186	40	Restaurant
4	183	5	Pagoda
5	171	14	ATM
6	179	14	Monastery
7	186	15	Hotel
8	202	65	School
9	184	10	Clinic
10	175	7	Bag shop

The searching time of the nearest keyword query that is evaluated over 8000 geo-data can be seen in Table 5.5. It is processed from user's current location. The number of execution is 10 times. The searching time of the proposed index structure is faster than that of R-tree as shown in Table 5.5. But, the searching time in the proposed index structure is sometimes high and sometimes low. The searching time for 'University' keyword is 6ms and the searching time for 'Pagoda' keyword is 5ms. But, the searching time of 'Restaurant' keyword and 'School' keyword are 40ms and 65ms, respectively. The reason is that the proposed index structure is constructed based on the services. Therefore, the searching time is based on the number of items in each service.

Table 5.6 Processing Time of Nearest Keyword Query from User' Desired Location

No. of Execution	R-tree (ms)	Proposed Index Structure (ms)	Input for User's Desired Location	Services
1	346	143	Sule Pagoda	Bank
2	336	158	A Bank	University
3	348	164	Golden Duck Restaurant	ATM
4	359	184	Sakura Hospital	Restaurant
5	346	163	KMD Computer Center	Clinic
6	353	161	AK Bag	Beauty Salon
7	363	27	Computer University	ATM
8	377	46	Pre-school	Bank
9	346	164	Sedona Hotel	Shopping Mall
10	352	164	Junction City	Gas Station

Table 5.6 shows the searching time of nearest keyword query from user desired location. In this processing, user's desired location is required to search firstly and then the nearest geo-object is retrieved based on the input service. The searching time is

calculated based on these two processing. This evaluation is also tested with over 8000 geo-data. It can be seen that the searching time is quicker than that of the R-tree according to the 10 times execution as shown in Table 5.6.

The text information of the geo-objects is stored in each node of the proposed index structure. For R-tree, the geo-location and text information of the geo-objects is situated separately. So, it takes more time to retrieve the geo-information. As the discussion mentioned above, the searching time is based on the number of items of each service. In execution 7, it only takes 27ms for searching the user's desired location (Computer University) and the nearest service (ATM). In execution 4, it takes 184 ms for searching the user's input desired location (Sakura Hospital) and the nearest service (Restaurant). The searching time in execution 4 and the execution 7 are obviously different because the number of items in 'University' and 'ATM' services is less than the number of items in 'Hospital' and 'Restaurant' services.

In addition to the comparison of Table 5.5 and Table 5.6, the searching time in Table 5.6 takes more time than the searching time in Table 5.5 because the searching from user's desired location is needed to take more process than the searching from the current location. The comparison of the nearest keyword query between user's current location and user's desired location based on the total number of objects is shown in Figure 5.9.

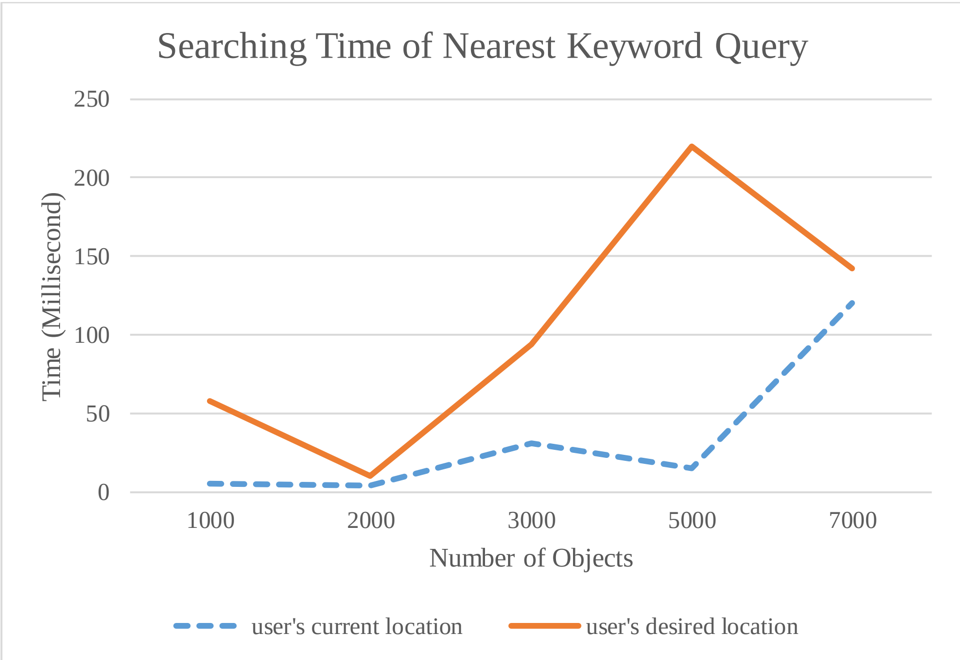


Figure 5.9 Comparison of Nearest Keyword Query between from User's Current Location and from User' Desired Location

Figure 5.10 shows the searching time of the nearest keyword query from the current location based on the total number of geo-objects. The nearest keyword query from the user' desired location is shown in Figure 5.11.

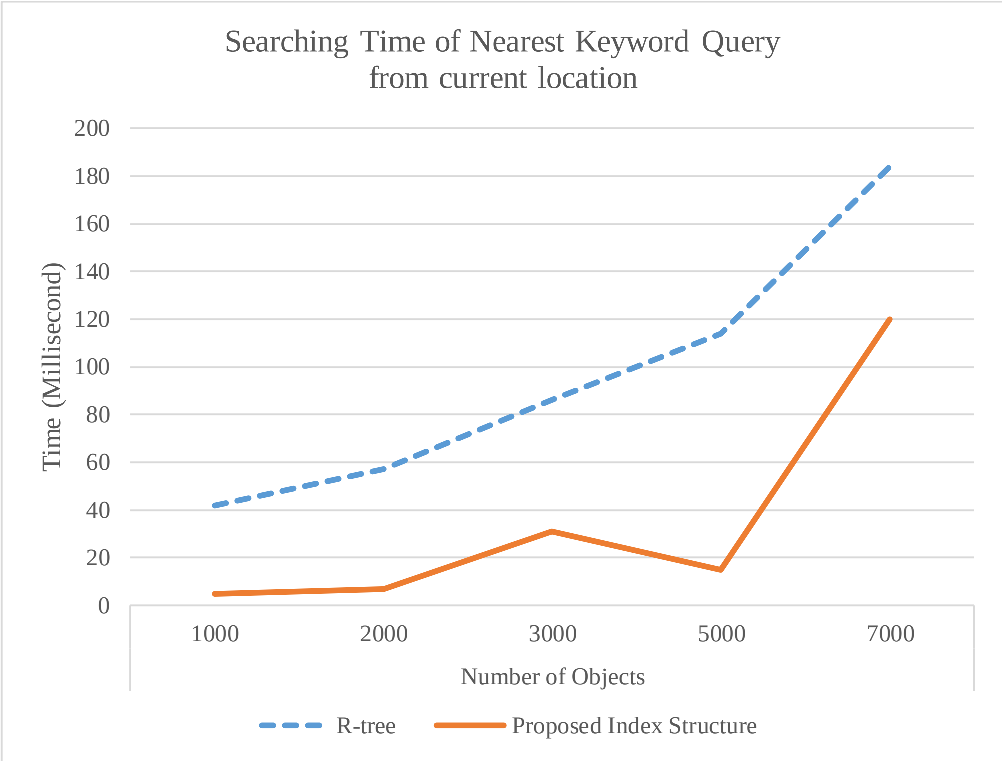


Figure 5.10 Comparison of Nearest Keyword Query from Current location

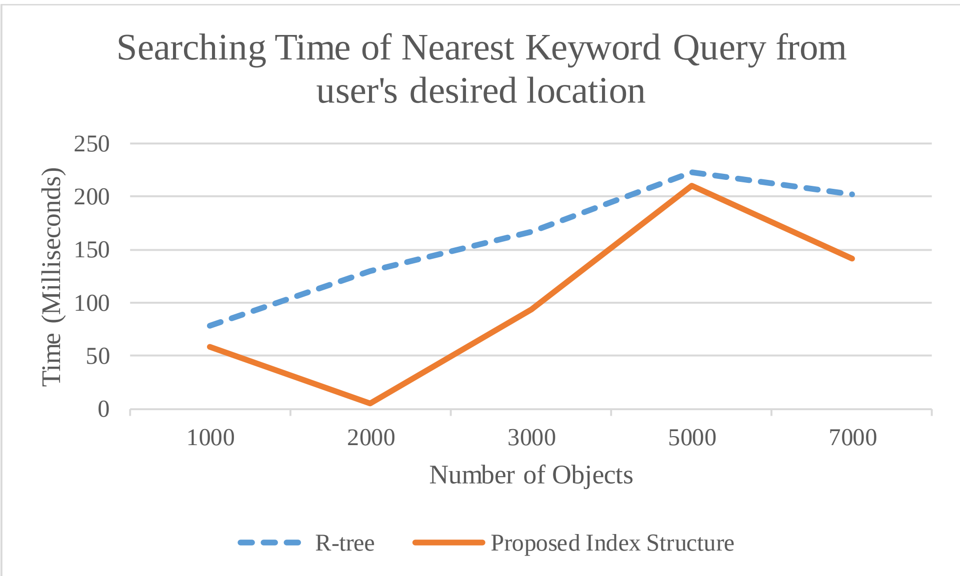


Figure 5.11 Comparison of Nearest Keyword Query from User's Desired Location

The searching time of the proposed index structure based on the total number of objects is less than that of the R-tree. In these two figures, the searching time of the proposed index structure can be less although the total number of objects is high. In Figure 5.10, the searching time of 3000 objects is more than that of 5000 objects. Also in Figure 5.11, the searching time of 5000 objects is obviously more than that of 7000 objects.

5.7.2.2 Evaluation of Range Keyword Query

In this system, range keyword query can be found from the current location or from the user's desired location. The evaluation of the Range Keyword Query is discussed for the searching time of these two locations. The number of processing geo-objects is used over 8000 for comparison in Table 5.7 and Table 5.8. Comparison of the range query keyword searching from the current location is shown in Table 5.7. In this comparison, the searching of the proposed index structure is visibly faster than R-tree's searching. Obviously, the searching time can be faster if the number of items in the service is few. It can be seen that it only takes 1ms in some of the services as shown in Table 5.7.

Table 5.7 Searching Time of Range Keyword Query from Current Location

No. of Execution	R-tree (ms)	Proposed Index Structure (ms)	Services
1	185	5	Bank
2	183	1	Pagoda
3	177	1	University
4	171	5	Restaurant
5	167	2	ATM
6	185	3	Monastery
7	172	3	Hotel
8	177	12	School
9	197	1	clinic
10	174	2	bag shop
11	173	2	Pharmacy
12	200	20	Bus Stop

Then, the comparison of the searching time of the range query from the desired location is shown in Table 5.8. The searching time of R-tree is slower than the proposed index structure. Moreover, the searching time in proposed index structure is also obviously different based on the user's desired location and services. The reason is same as above in the searching of the nearest keyword query. The searching time in execution 7 and 8 is evidently less than the searching with other services.

Table 5.8 Searching Time of Range Keyword Query from User's Desired Location

No. of Execution	R-tree (ms)	Proposed Index Structure (ms)	Input for User's Desired Location	Services
1	361	147	Sule Pagoda	Bank
2	356	154	A Bank	University
3	339	148	Golden Duck Restaurant	ATM
4	347	158	Sakura Hospital	Restaurant
5	342	169	KMD Computer Center	Clinic
6	359	153	AK Bag	Beauty Salon
7	349	17	Computer University	ATM
8	352	19	Pre-school	Bank
9	359	145	Sedona Hotel	Shopping Mall
10	351	152	Junction City	Gas Station

The comparison of the range keyword query between the current location and the desired location is shown in Figure 5.12. The comparison is based on the total number of objects and is operated on the proposed index structure. In this comparison, the searching from current location only takes a few milliseconds. The searching from the desired location is clearly high as shown in Figure 5.12 because it requires to operate two actions: desired location search and range search for input service.

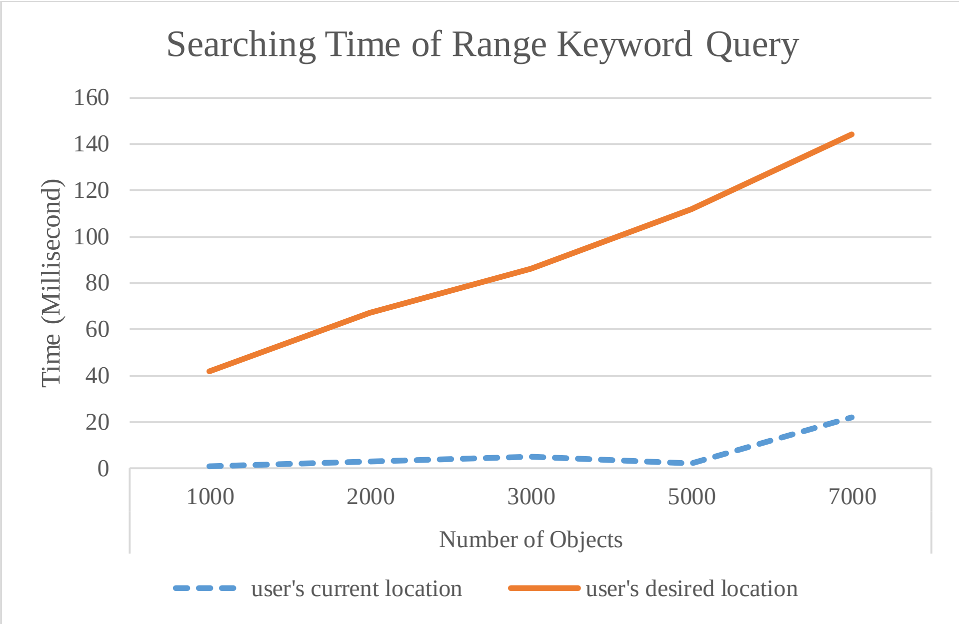


Figure 5.12 Comparison of Range Keyword Query between User's Current Location and User's Desired Location

The comparison of range keyword query between the proposed index structure and R-tree is shown in Figure 5.13 and Figure 5.14. Figure 5.13 also shows the experimental result of the range query from the current location. The evaluation of the rang query from the desired location can be seen in Figure 5.14.

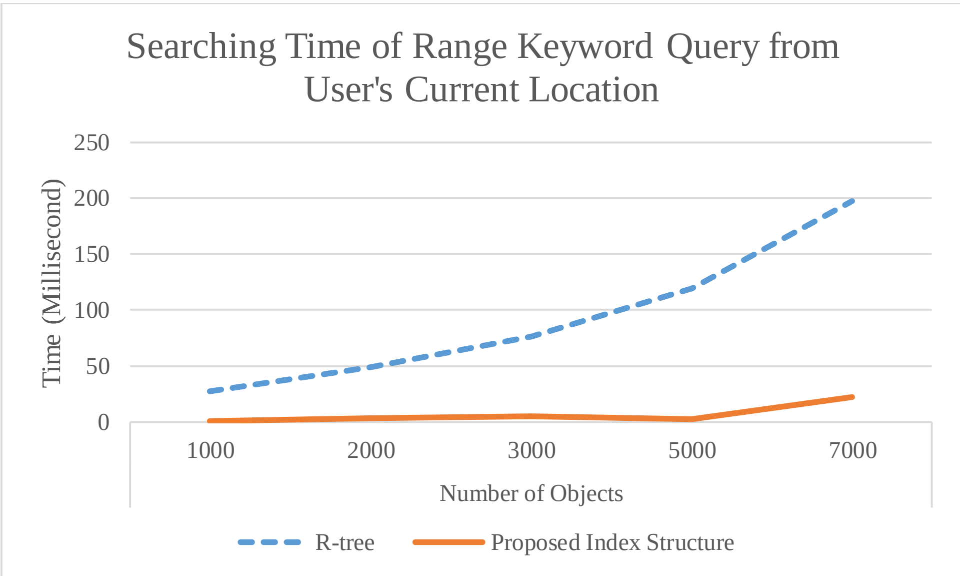


Figure 5.13 Comparison of Range Keyword Query from Current Location

The searching time using the proposed index structure takes less time than using R-tree. Therefore, the searching time in range keyword query also depends on the number of objects in each service as already explained in nearest keyword query. The searching of 5000 objects in Figure 14 is obviously less than the searching of 3000 objects.

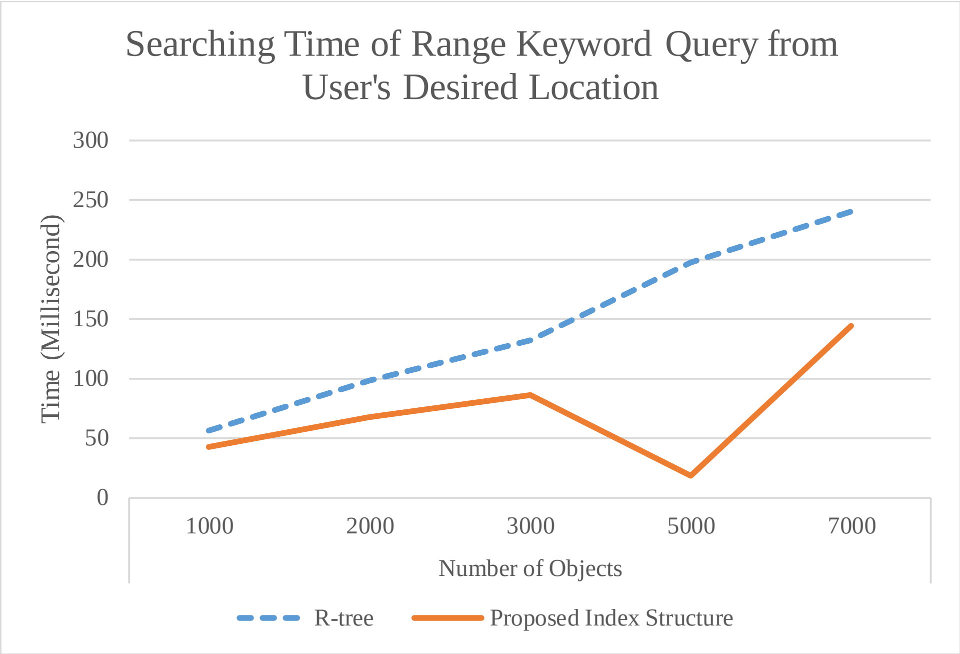


Figure 5.14 Comparison of Range Keyword Query from User's Desired Location

5.7.2.3 Evaluation of Keyword Query

In Keyword Query, there are three operations to retrieve the geo-information. The proposed index structure is built based on the service. So, the first operation is that it needs to operate for determining the relevant proposed index structure based on the input query. If it can determine the proposed index structure for processing the queries, the objects are searched in this proposed index structure. If the service name is not included in the input query and the proposed index structure cannot be determined for searching case, the worst case occurs that it requires to search all of the items in each proposed index structure. In this case, the searching time can be high. The second operation is depending on the calculation of the distance between the objects that are relevant to the input keyword and user's current location. This calculation step is taken place to show the distance from the retrieved objects to current location. The third operation is for finding the nearest location to user's current location among the objects that are relevant to the input keyword. So, the searching time is calculated according to Equation 5.1.

$$Tkq = Ts + Td + Tn \tag{5.1}$$

where Tkq = Time of Keyword Query

Ts = Time of determine the proposed index structure

Td = Time of distance calculation

Tn = Time of searching the nearest location

The searching time of keyword query is expressed in Table 5.9 with the input keyword. This evaluation is operated using over 8000 geo-objects. It executes 10 times. In this comparison, the searching time of R-tree is higher than that of the proposed index structure. In this keyword query, the searching time is based on how many objects are found. Moreover, it also depends on the input query that contains the service name or not.

Table 5.9 Searching Time of Keyword Query

No. of Execution	R-tree (ms)	Proposed Index Structure (ms)	Services
1	176	165	A Bank
2	173	144	Sule Pagoda
3	162	155	Golden Duck
4	176	130	Sule Shangri-la Hotel
5	175	146	UCSY
6	174	113	Japanese Restaurant
7	177	149	Shwedagon Pagoda
8	175	145	AYA ATM
9	162	135	Thidaron Monestary
10	177	14	Skin Care Clinic

The comparison of the keyword query based on the total number of objects is shown in Figure 5.15. the searching time of the proposed index structure is faster than that of R-tree according to Table 5.8 and Figure 5.15. But, the searching time of the keyword query is sometimes slightly faster than that of R-tree. In this case, R-tree does not take a long time to find the data if the amount of total data is few. For the proposed index structure, the searching time does not depend on the total number of object and it can take more time if the amount of data in each service is huge. It can be seen in Figure 5.15 that the searching time in 3000 objects of proposed index structure is nearly the same with the searching time of R-tree because the amount of total data is not much and the data amount in the searching service is large.

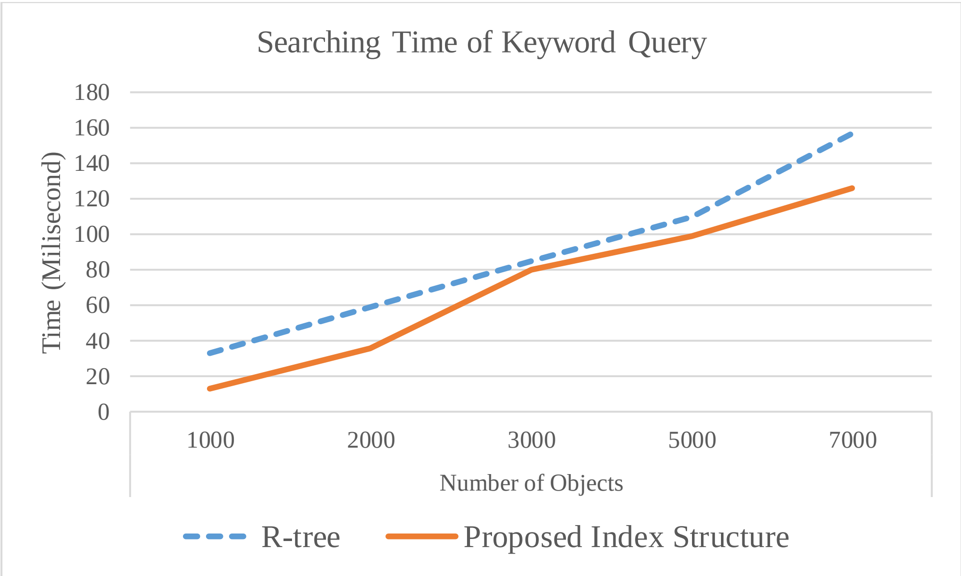


Figure 5.15 Comparison of Keyword Query

5.8 Summary

This chapter describes the implementation of the proposed system. It explains the implementation of the proposed index structure on android mobile devices and the access to the current location. Then, it describes the map of the proposed system. The proposed system can display the geo-data in offline or online. This chapter explains the offline map of the proposed system and its limitation for offline use. Then, it presents the available Yangon Map area for the offline map. Besides, it describes the geo-data collection for the geo-spatial database and the available categories of services in the proposed system. The implementation of the user interaction is discussed in this chapter. It also explains the detailed process of the spatial keyword queries in the proposed system with their input and output.

The evaluation of the system is presented in this chapter. First, it discusses the construction time of the proposed index structure. Then, the experimental result of the spatial keyword query is explained. The evaluation of the nearest keyword query, range keyword query, and keyword query are described in detail with their searching time. Due to the experimental result, the proposed index structure can perform the spatial keyword query with the efficient searching time.

CHAPTER 6

CONCLUSION AND FURTHER EXTENSION

This research is applied for processing the spatial keyword query by using the proposed index structure. The main objective is to provide the visitor for obtaining the desired geo-location efficiently and effectively. The limitation of the system and further extension are discussed in this chapter.

This research is intended to provide the users to obtain the geo-information anywhere and anytime. To support not only foreign users but also local users, it is implemented two languages, Myanmar and English. It is easy for the local people who are difficult to use with the English version. The geo-information contains geo-location and text information. The coordinates of the spatial data are two dimensional data. So, a new index structure is built in order to extracting the geo-location data with their respective text information effectively and efficiently. Hilbert curve, b-tree and inverted file are used to constructing this proposed index structure. B-tree is efficient for one-dimensional query. To build the b-tree for two-dimensional query, Hilbert cure is applied for mapping two dimensional to one-dimensional data. Then, the inverted file is embedded in the b-tree node in order to mainly support for searching the text information. This proposed index structure can reduce the searching time.

The geo-information in this proposed system focus on the Yangon region. It can retrieve the geo-data based on the given spatial keyword query. The processing of the spatial keyword query can be performed by using the proposed index structure. The major available spatial keyword queries are nearest keyword query and range keyword query. These queries are fetched from user's current location or user's desired location. In addition, the geo-information can also be retrieved for keyword query with their distance. So, this proposed system provides the user to know the geo-information not only around the current location but also around the desired location. Thus, users can find in advance the places around the desired location which they want to know and pre-plan the places they want to go.

This system is developed on the android mobile devices in order to access anywhere. For anytime, this system can be used whether the internet connection is available or not. So, the offline map is required to apply for offline use. The Mapbox map is used to display the geo-location in offline and online. Therefore, this proposed system helps the user to search the desired geo-information from current location or the desired places anywhere and anytime.

The evaluation of this proposed system is operated using over 8000 geo-objects and also tested on the total number of objects, 1000, 2000, 3000, 5000 and 7000. The proposed index structure is compared with R-tree. The comparison is executed for the spatial keyword query processing by performing 10 times execution. The nearest keyword query, range keyword query and keyword query are evaluated in this spatial query processing. For the nearest keyword query and range keyword query, the geo-information can be searched from the current location or desired location. Therefore, the searching time of these two queries is executed and compared with the searching time of the R-tree. It also compares the searching time from the current location and from desired location.

In the experimental result of these spatial keyword queries, the searching proposed index structure is faster than R-tree. But, the searching of the proposed index structure can be high although the total number of objects is less because the searching time is based on the number of objects in the service instead of total number of objects in the proposed index structure. The reason is that the proposed index structure is constructed based on the services. Therefore, it needs to determine that which the proposed index structure is used for processing the spatial keyword queries. In keyword query, it has the worst case. If the service name does not contain in the input query and the proposed index structure cannot know for processing the queries, it takes more time in order to process the keyword query. So, the worst case can occur if it requires to search the necessary geo-information by going through all of the items in the proposed index structure.

Another worst case is the construction of the proposed index structure. In the constructing step, the geo-information of the objects is also stored in the node of the proposed index structure. So, it takes more time than construction of R-tree. Although the construction time is high, the proposed index structure can reduce the searching time and can retrieve the geo-information effectively and efficiently.

6.1 The Limitation of the System

Although Yangon map can be used in an offline, this map is needed to have on the mobile devices. So, the offline map is required to download to the android mobile devices if it does not store in the mobile devices. To download the map, it needs the online. After finishing the download, the map can be used in offline to display the geo-location. The map of the area and zoom level is limited in offline. It can only use within the determining area for Yangon offline map which explained in the previous chapter because the tile of offline map cannot exceed 6000 tiles for free version.

6.2 Further Extension

This system does not consider on the road network and only applies on the Yangon Region. For the languages, it can only operate for two languages: English and Myanmar. Nowadays, there are several tourists come to visit Myanmar and the number of tourists is increasing day by day. The tourists come from different countries and use different languages. For further extension, the proposed index structure will be constructed to retrieve the geo-information with different languages. Moreover, it will apply for the real road network to give the user more precise location for the query like nearest location. The area of the region will be expanded to obtain the geo-information of the whole country.

AUTHOR’S PUBLICATIONS

[P1] Myat Thiri Khine, Su Nandar Aung and Myint Myint Sein, “Geo-textual Index Structure for Spatial Keyword Query with Myanmar Language", in Proceedings of the 14th International Conference on Computer Applications (ICCA2016), Yangon, Myanmar, February 2016.

[P2] Myat Thiri Khine, Myint Myint Sein, “Poster: Index Structure for Spatial Keyword Query with Myanmar Language on the Mobile Devices”, in Processing MobiSys '16 Companion Proceedings of the 14th Annual International Conference on Mobile System, Application, and Services Companion, Singapore, 2016.

[P3] Myat Thiri Khine, Myint Myint Sein, “Location and Transportation Services for Yangon Region”, The Seventh International Conference on Science and Engineering 2016 (ICSE 2016), Yangon, Myanmar.

[P4] Myat Thiri Khine, Myint Myint Sein, "Geo-Spatial Index Structure for Myanmar Keyword Query", in Proceedings of the 15th International Conference on Computer Applications (ICCA2017), Yangon, Myanmar, February 2017.

[P5] Myat Thiri Khine, Myint Myint Sein, “Location Based Trip Planning Mobile Application for Acient Bagan Region”, 2017 IEEE 6th Global Conference on Consumer Electronics (IEEE GCCE 2017), Nagoya, Japan, Oct 2017.

[P6] Myat Thiri Khine, Myint Myint Sein, “Location-based Services for Surrounding Area with Myanmar Language on Mobile Devices”, 11th Multi-disciplinary International Workshop on Artificial Intelligence (MIWAI 2017), Brunei, Nov 2017.

[P7] Myat Thiri Khine, Myint Myint Sein, “Location-based Services for Surrounding Area with Myanmar Language on Mobile Devices”, Lecture Notes in Computer Science (LNCS), Volume 10607.

[P8] Myat Thiri Khine, Myint Myint Sein, “Continuous Location based Query for Bagan's Trip Planning Mobile Application”, 2018 IEEE 7th Global Conference on Consumer Electronics (IEEE GCCE 2018), Nara, Japan, Oct 2018.

BIBLIOGRAPHY

[1] R. Abdalla, "Introduction to Geospatial Information and Communication Technology (GeoICT)", Springer International Publishing Switzerland, 2016. [DOI 10.1007/978-3-319-33603-9]

[2] V. Agafonkin, "Euclidean geodesic approximation", [Online]. Available: https://blog.mapbox.com/fast-geodesic-approximations-with-cheap-ruler-106f229ad016

[3] F. Angiulli, C Pizzuti, "Approximate k -Closest-Pairs with Space Filling Curves", in Proceedings of the 4th International Conference on Data Warehousing and Knowledge Discovery, pp. 124-134, 2002.

[4] L. Balasubramanian, M. Sugumaran, "A State-of-Art in R-Tree Variants for Spatial Indexing", International Journal of Computer Applications, Volume 42, No.20, pp. 35-41, March 2012.

[5] X. Cao, G. Cong, C.S. Jensen, and B.C. Ooi, "Collective spatial keyword querying", in Proceedings of the 2011 International Conference on Management of Data, ACM, Athens, Greece, pp. 373-384, 2011.

[6] X. Cao, L. Chen, G. Cong, C. S. Jensen, Q. Qu, A. Skovsgaard, D. Wu and M. L. Yiu, "Spatial keyword querying", in International Conference on Conceptual Modeling, Verlag Berlin Heidelberg, 2012.

[7] Cary, O. Wolfson, and N. Rishe. "Efficient and scalable method for processing top-k spatial boolean queries", In SSDBM, pp. 87–95, 2010.

[8] K.T. Chang, "Geographic Information System", Tata McGraw-Hill, New Delhi, 2010.

[9] Y. Y. Chen, T. Suel and A. Markowetz, "Efficient query processing in geographic web search engines", in Proceedings of the 2006 ACM SIGMOD International Conference on Management of Data, ACM, Chicago, IL, USA, pp. 277-288, 2006.

[10] L. Chen, G. Cong, C. S. Jensen and D. Wu, "Spatial Keyword Query Processing: An Experimental Evaluation", in VLDB, 2013.

[11] S. Chen, J.Y. Liang, Y. K. Lee, M. H. Fan, C. H. Huang, "Large Efficient Searching Algorithms for Multi-dimensional Space Data Using Hilbert Space-filling Curves", Journal of Information & Computational Science, vol 8, no. 1, pp. 41–50, 2011.

[12] M. Christoforaki, J. He, C. Dimopoulos, A. Markowetz, and T. Suel. “Text vs. space: efficient geo-search query processing”, In CIKM, pp. 423–432, 2011.

[13] 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.

[14] T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein, “Introduction To Algroithm”, PHI Learning, Third Edition, 2013.

[15] M. M. P. Crespo, “Design, Analysis and Implementation of New Variants of Kd-trees”, 2010.

[16] S. Davis, “GIS for Web Developers”, The Pragmatic Programmers LLC., Raleigh, North Carolina, 2007.

[17] Z. Deng, M. Wang, L. Wang, X. Huang, W. Han, J. Chu, and A. Zomaya, “An Efficient Indexing Approach for Continuous Spatial Approximate Keyword Queries over Geo-Textual Streaming Data”, ISPRS International Journal of Geo-Information, vol. 8, no. 2, pp. 57, Jan. 2019 [Online]. Available: http://dx.doi.org/10.3390/ijgi8020057

[18] Faloutsos, S. Roseman, “Fractals for Secondary Key Retrieval”, In Proceedings of the Eighth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, PODS '89, pp. 247-252.

[19] Faloutsos and S. Christodoulakis, “Signature files: An access method for documents and its analytical performance evaluation”, ACM Transactions on Information Systems, vol 2, no. 4, pp. 267–288, 1984.

[20] C. Faloutsos, “Multiattribute Hashing Using Gray Codes”, in Proceeding of ACM SIGMOD International Conference on Management of Data, pp. 227–238, Aug. 1986.

[21] C. Faloutsos, “Gray Codes for Partial Match and Range Queries”, IEEE Transactions On Software Engineering, Vol. 14, No. 10, pp. 1381–1393, Aug. 1988.

[22] D. Felipe, V. Hristidis, and N. Rishe, “Keyword search on spatial databases”, In ICDE, pp. 656–665, 2008.

[23] G. Li, J. Feng, and J. Xu. Desks, “Direction-aware spatial keyword search”, In ICDE, pp. 474–485, 2012.

[24] R. A. Finkel, J. L. Bentley, “Quad Trees: a data structure for retrieval on composite keys”, Acta Informatica, vol 4, no. 1, pp. 1–9, 1974.

[25] V. Gaede, O. Gu ̈nther, “Multidimensional Access Methods”, ACM Computing Surveys, Vol. 30, No. 2, June 1998.

[26] B. Goodwin, M. Hopcroft, D. Luu, A. Clemmer, M. Curmei, S. Elnikety, and Y. He. Bitfunnel, “Revisiting signatures for search”, In SIGIR, pp 605–614, 2017.

[27] R. H. Güting, “An Introduction to Spatial Database Systems”, Spatial Database Systems of the VLDB Journal, Vol. 3, No. 4, October 1994.

[28] Guttman, “R-trees: a dynamic index structure for spatial searching”, In Proceedings of ACM SIGMOD International Conference on Management of Data, pp. 47–57, 1984.

[29] R. Hariharan, B. Hore, C. Li, S. Mehrotra, “Processing Spatial-Keyword (SK) Queries in Geographic Information Retrieval (GIR) Systems”, 19th International Conference on Scientific and Statistical Database Management (SSDBM 2007), pp 16-25, 2007.

[30] M. Hosken, “Representing Myanmar in Unicode Detail and Example Version 4”, SIL International and Payap University Linguistics Institute, Chiang Mai, THAILAND.

[31] W. Huang, G. Li, K.-L. Tan, and J. Feng, “Efficient safe-region construction for moving top-k spatial keyword queries”, In CIKM, pp 932–941, 2012.

[32] H. V. Jagadish, “Analysis of the Hilbert curve for representing two-dimensional space”, Information Processing Letters, Vol. 62, No. 1, pp. 17–22, April 1997.

[33] H. V. Jagadish, “Linear Clustering of Objects with Multiple Attributes”, in Proceeding of ACM SIGMOD International Conference on Management of Data, pp. 332–342, 1990.

[34] Jagoe, “Mobile location services: The definitive guide”, Upper Saddle River, New Jersey: Prentice Hall, 2002.

[35] M. R. Kabat, “DESIGN AND ANALYSIS OF ALGORITHMS”, PHI Learning, 2013

[36] Khodaei, C. Shahabi, and C. Li, “Hybrid indexing and seamless ranking of spatial and textual features of web documents”, In DEXA, pp. 450–466, 2010.

[37] Khodaei, C. Shahabi, C. Li, “SKIF-P: a point-based indexing and ranking of web documents for spatial-keyword search”, Geoinformatica archive, Vol. 16 Issue 3, pp. 563-596, July 2012.

[38] H.P. Kriegel, T. Brinkhoff, R. Schneider, “The Combination of Spatial Access Methods and Computational Geometry in Geographic Database Systems”, in Proceedings 2nd Symp. on the Design of Large Spatial Databases, Zurich, Switzerland, 1991, in Lecture Notes in Computer Science, Vol. 525, Springer, pp 5-22, 1991.

[39] P. Kuba, “Data Structures for Spatial Data Mining”, FI MU Report Series, 2001.

[40] K Lawder, J. “Calculation of Mappings Between One and n-dimensional Values Using the Hilbert Space-filling Curve”, Technical Report, 2000.

[41] J. K. Lawder and P. J. H. King, “Querying Multi-Dimensional Data Indexed Using the Hilbert Space-Filling Curve”, ACM SIGMOD Record, Vol. 30, No. 1, pp. 19–24, Mar. 2001.

[42] Z. Li, K. C. K. Lee, B. Zheng, W.-C. Lee, D. L. Lee, and X. Wang, “Ir-tree: An efficient index for geographic document search”, IEEE TKDE, Vol. 23, No. 4, pp. 585–599, 2011.

[43] J. Lu, Y. Lu, and G. Cong, “Reverse spatial and textual k nearest neighbor search”, In SIGMOD, pp. 349–360, 2011.

[44] S. Luo, Y. Luo, S. Zhou, G. Cong, J. Guan, Z. Yong, “Distributed Spatial Keyword Querying on Road Networks”, in Proceedings 17th International Conference on Extending Database Technology (EDBT), Athens, Greece, pp 235-246, March 24-28, 2014.

[45] D. Manning, P. Raghavan, and H. Schütze, “Introduction to Information Retrieval”, New York, NY, USA: Cambridge University Press, 2008.

[46] K. Michael, K.G. Michael, “Innovative Automatic Identification and Location-Based Services: From Bar Codes to Chip Implants”, New York, Information Science Reference, 2009.

[47] M. F. Mokbel, W. G. Aref, I. Kamel, “Performance of MultiDimensional SpaceFilling Curves”. GIS’02, McLean, Virginia, USA, November 8–9, 2002.

[48] Moon, H. V. Jagadish, C. Faloutsos, and J. H. Saltz, “Analysis of the Clustering Properties of the Hilbert Space-Filling Curve,” IEEE Trans. on Knowledge and Data Eng., Vol. 13, No. 1, pp. 124–141, Jan. 2001.

[49] G.M. Morton, “A computer oriented geodetic data base and a new technique in file sequencing”, Technical report, IBM, Ottawa, Canada, 1966.

[50] P. V. Oosterom, “Spatial Access Methods”, in Geographical Information Systems Principles, Technical Issues, Management Issues, and Applications (edited by ongley, Goodchild, Maguire en Rhind), Wiley, vol. 1, pp. 385-400, 1999.

[51] J. A. Orenstein and T. H. Merrett, “A Class of Data Structures for Associative Searching”, in Proceedings of Symposium on Principles of Database Systems (PODS), pp. 181–190, 1984.

[52] J. A. Orenstein, “Spatial Query Processing in an Object-Oriented Database System”, in Proceeding of ACM SIGMOD International Conference on Management of Data, pp. 326–336, 1986.

[53] W. Osborn, A. Hinze, “Issues in Location-Based Indexing for Co-operating Mobile Information Systems”, LNCS 4805, pp. 226–235, 2007.

[54] Papadias, “Nearest Neighbor Query”, In: LIU L., ÖZSU M.T. (eds) Encyclopedia of Database Systems. Springer, Boston, MA, 2009.

[55] P. Patel, D. Garg, “Comparison of Advance Tree Data Structures”, International Journal of Computer Applications (0975 – 8887) Vol. 41, No.2, March 2012

[56] C. Pontikakos, M. Sambrakos, T. Glezakos and T. Tsiligiridis, “Location-based services: A framework for an architecture design”, 2010.

[57] Poorazizi, A. A. Alesheikh, “Developing a Mobile GIS for Field Geospatial Data Acquisition”, WSEAS International Conference on URBAN PLANNING and TRANSPORTATION (UPT'07), Heraklion, Crete Island, Greece, July 22-24, 2008.

[58] S. K. Powar, G. M. Magar, “SPATIAL KEYWORD QUERY PROCESSING: R*-IF TREE IMPLEMENTATION”, International Journal of Computer Engineering and Applications, Vol. 7, Issue 2, pg 65-72, 2018.

[59] R. S. Purves, P. Clough, C. B. Jones, “Geographic Information Retrieval: Progress and Challenges in Spatial Search of Text”, Foundations and Trends® in Information Retrieval: Vol. 12: No. 2-3, pp 164-318, 2018. [Online]. Available: http://dx.doi.org/10.1561/1500000034

[60] Z. Qin, E. Zhou, Y. Ding, Y. Zhao, F. Deng, H. Xiong, “Data Service Outsourcing and Privacy Protection in Mobile Internet”, 2018. [DOI: 10.5772/intechopen.79903]

[61] Q. Qu, S. Liu, B. Yang,C. S. Jensen, “Efficient Top-k Spatial Locality Search for Co-located Spatial Web Objects”, IEEE, 2014

[62] P. Rigaux, M. Scholl, A. Voisard, “Spatial Databases with Application to GIS”, Morgan Kaufmann Publishers, Elsevier Science (USA), 2002.

[63] J. B. Rocha-Junior, O. Gkorgkas, S. Jonassen, and K. Nørv°ag, “Efficient processing of top-k spatial keyword queries”, In SSTD, pp. 205–222, 2011.

[64] H. Sagan, “Space Filling Curve”, Springe- Verlag, 1994

[65] H. Samet. “An Overview of Hierarchical Spatial Data Structure”, In Proceedings of the 5th Israeli Symposium on Artificial Intelligence, Vision, and Pattern Recognition, Tel Aviv, Israel, pages 331-351, December 1988.

[66] H. Samet, “The Quadtree and Related Hierarchical Data Structures”, ACM Computing Surveys (CSUR), vol.16, no. 2, pp. 187-260, June 1984. [doi>10.1145/356924.356930]

[67] H. Samet, A. Rosenfeld, C. A. Shaefer and R. E. Webber, “A Geographic Informaiton System Using Quadtrees”, Pattern Recognition Vol. 17, No. 6, pp. 647-656, 1984.

[68] C. A. Shaffer, “Data Structures and Algorithm Analysis in Java”, Third Edition, 2011.

[69] S. Shekhar, R. R. Vatsavai, X. Ma, and J. S. Yoo, “Navigation Systems: A Spatial Database Perspective”, Location-Based Services, Agnes Voisard and Jochen Schiller (Eds.), Morgan Kaufmann, New York, 2004.

[70] V. Singh, B. Zong, A. K. Singh, “Nearest Keyword Set Search in Multi-Dimensional Datasets”, IEEE Transactions on Knowledge and Data Engineering, Vol 28, Issue: 3, March 1, pp. 741-755, 2016. [DOI: 10.1109/TKDE.2015.2492549]

[71] Taniar, M. Safar, Q. T. Tran, J. W. Rahayu and J. H. Park, “Spatial network RNN queries in GIS”, Computer Journal, Vol. 54, No. 4, pp. 617-627, 2011.

[72] M. H. Tsou, “Integrated Mobile GIS and Wireless Internet Map Servers for Environmental Monitoring and Management”, (the Special Issue on Mobile Mapping and Geographic Information Systems) in Cartography and Geographic Information Science. Vol. 31, No. 3, pp. 153-165, 2004.

[73] T. Turk, “Location Based Services (LBS) and Related Standards”, Commission IV, WG IV/6, 2006.

[74] S. Vaid, C. B. Jones, H. Joho, and M. Sanderson, "Spatio-textual indexing for geographical search on the web", In SSTD, pp. 218–235, 2005.

[75] K. Virrantaus, J. Veijalainen, J. Markkula, A. Garmash, A. Katasonov, V. Terziyan, H. Tirri, "Developing GIS-supported location-based services", in Proceedings of the Second International Conference on Web Information Systems Engineering, Kyoto, Japan, Dec 2001. [DOI: 10.1109/WISE.2001.996708]

[76] T. Wang, G. Li, J. Feng, "Efficient Algorithms for Top-k Keyword Queries on Spatial Databases", 12th IEEE International Conference on Mobile Data Management, 2011.

[77] D. Wu, M. L. Yiu, C. S. Jensen, and G. Cong, "Efficient continuously moving top-k spatial keyword query processing", In ICDE, pp. 541–552, 2011.

[78] D. Wu, M. L. Yiu, G. Cong, and C. S. Jensen, "Joint top-k spatial keyword query processing", IEEE TKDE, Vol. 24, No. 10, pp.1889–1903, 2012.

[79] D. Wu, G. Cong, and C. S. Jensen, "A framework for efficient spatial web object retrieval", VLDBJ, Vol. 21, No.6, pp.797–822, 2012

[80] H. Yan, S. Ding, and T. Suel, "Inverted index compression and query processing with optimized document ordering", In WWW, pp. 401–410, 2009.

[81] J. Zhang, N. Mamoulis, D. Papadias and Y. Tao, "All-nearest-neighbors queries in spatial databases", in Proceedings of the 16th International Conference on Scientific and Statistical Database Management (SSDBM), pp. 297-306, 2004.

[82] D. Zhang, Y. M. Chee, A. Mondal, A. K. H. Tung, and M. Kitsuregawa, "Keyword search in spatial databases: Towards searching by document", In ICDE, pp. 688–699, 2009.

[83] D. Zhang, B. C. Ooi, and A. K. H, "Tung. Locating mapped resources in web 2.0", In Proceedings of IEEE International Conference on Data Engineering (ICDE), pp. 521–532, 2010.

[84] B. Zheng, N. Jing Yuan, K. Zheng, X. Xie, S. Sadiq, X. Zhou, "Approximate Keyword Search in Semantic Trajectory Database", In Proceedings of IEEE International Conference on Data Engineering (ICDE), Seoul, pp. 975-986, 2015.

[85] Y. Zhou, X. Xie, C. Wang, Y. Gong, and W.-Y. Ma, "Hybrid index structures for location-based web search", CIKM '05 Proceedings of the 14th ACM international conference on Information and knowledge management, Germany, pp. 155–162, 2005.

[86] J. Zobel and A. Moffat, "Inverted files for text search engines", ACM Computing Surveys, Vol. 38, No. 2, pp. 6, 2006.

[87] J. Zobel, A. Moffat, and K. Ramamohanarao, "Inverted files versus signature files for text indexing", ACM Transactions on Database Systems, Vol. 23, No. 4, pp. 453–490, 1998.

[88] "Offline maps", https://docs.mapbox.com/help/troubleshooting/mobile-offline/

[89] "mAppWidget", https://lemberg.github.io/mappwidget/user_guide.html

ACRONYMS

AP	Adaptive spatial-textual Partition
CSV	Comma-Separated Values
GIS	Geographical Information System
GPS	Global Positioning System
IR2	Information Retrieval R
IUR	Union-R
KD	K-Dimensional
kNN	k Nearest Neighbors
LBS	Location Based Services
MIUI	Mi User Interface
NE	Northeast
NPD	Node-Partition-Distance
NW	Northwest
PC	Personal Computer
POI	Point of Interest
QGIS	Quantum Geographic Information System
S2I	Spatial Inverted Index
SDK	Software Development Kit
SE	Southeast
SFC	Space Filling Curve
SKI	Hybrid Spatial-Keyword Indexing
SKIF	Spatial Keyword Inverted File
SKR	Spatial Keyword R
SW	Southwest