Geo-textual Index Structure for Spatial Keyword Query with Myanmar Language

Myat Thiri Khine, Su Nandar Aung and Myint Myint Sein
University of Computer Studies, Yangon Myanmar
myatthirikhine@ucsy.edu.mm, sunandaraung@ucsy.edu.mm, myintucsy@gmail.com

Abstract

Geographic query is composed of query keywords and a location. Spatial keyword queries which are queries on spatial objects associated with textual attributes have received significant attention in geographic information system (GIS) recently. A spatial keyword query takes a user location and user-supplied keywords as arguments and returns objects that are spatially and textually relevant to these arguments. Geo-textual index play an important role in spatial keyword querying. There are a number of geo-textual indices have been proposed in recent years. Mostly, the R-tree and its variants and the inverted file are combined. Most of the keywords are considered for English language and keyword with Myanmar native language is also necessary for the users who are not familiar with the English language. This paper proposes an index structure that combines K-d tree and inverted file which is considered on the Myanmar keywords queries to find the desired location with Myanmar language efficiently. Myanmar 3 Unicode is used for keyword search.

Keywords: *Spatial Keyword Queries, Hybrid Index Structure, Proposed Index, Myanmar language.*

1. Introduction

Spatial database systems manage large collections of spatial data, which apart from spatial attributes contain non spatial information. Spatial data are data that have a location (spatial) and mainly required for Geographic Information Systems (GIS) whose information is related to geographic locations. Geographical information system stores spatial data and retrieves the geo-information from existing spatial data. Given a location and a set of keywords, spatial keyword query returns the objects that are relevant to the text describing the objects to the query keywords. Due to the popularity of keyword search, particularly on the Internet, many of these applications allow the user to provide a list of keywords that the spatial objects should contain, in their name or description or categories. Spatial Keyword search is an important tool in exploring useful information from spatial database and has been studied for years.

Many index structures have been proposed in recent years. R-tree is mainly used and that combine with inverted file, namely the families of IR-tree [13, 12, 6, 7, 9, 8, 10]. All use R-tree for spatial (latitude/longitude) index and inverted file for textual index. The construction of an efficient index structure should take into account overlaps between nodes and coverage of a node. Minimization of a node coverage leads to more precise searching within the tree and minimization of the overlap between nodes reduces the number of paths tested in the tree during a search that can reduce search time. As the data objects in the R-tree can be overlapping and covering each other, the search process in the R-tree might suffer from unnecessary node visits and higher IO cost [14]. Moreover, the IR-trees suffer from high update cost. Each node has to maintain an inverted index for all the keywords of documents associated with this node's MBR. When a node is full and split into two new nodes, all the textual information in the node has to be re-organized [16]. As the R-tree

need to reorganized, it suffers from higher CUP costs.

Hybrid index structure that combines the K-d tree and inverted file for spatial keyword search with minimum IO costs and CPU costs has been proposed [1][2][3]. This hybrid index structure is considered for English keyword queries. This paper presented an index structure that combines K-d tree and inverted file to process spatial keyword queries with Myanmar language within minimum time.

The remainder of the paper is organized as follows. Section 2 reviews the related work. Hybrid index structure is explained in section 3. Section 4 describes the proposed system. In section 5, Myanmar language is introduced. The experimental results is showed in section 6. Section 7 concludes the paper with directions for future work.

2. Related Works

Spatial Keyword search has been well studied for years due to its importance to commercial search engines. Various types of spatial keyword queries have been proposed. For spatial keyword search, the index structure is created for both spatial and textual relevance. Most index structures [10, 12, 6, 9, 8] use R-tree and its variants as spatial index and inverted file for text index. They all combine both indices depending on the combination schemes [5]. Among them [9] integrates signature file instead of inverted file into each node of the R-tree. Inverted file-R*tree (IF-R*) and R*-tree-inverted file (R*-IF) [10] are two geo-textual indices that loosely combine the R*-tree and inverted file. Hariharan et al. R. G¨obel, A. Henrich, R. Niemann, and D. Blank [9] presented the KR*-tree. This paper proposed a framework for GIR systems and focus on indexing strategies. I. D. Felipe, V. Hristidis, and N. Rishe [8] uses R*-tree for spatial index and inverted file for text index. Cary et al, [12] proposed SKI that combines and R-tree with an inverted index by the inclusion of spatial references in posting lists. In [12] the posting list of term contains all its term bitmaps rather than documents. The IR tree [6] creates each nodes of the R-tree with a summary of the text content of the objects in the corresponding subtree. Li et al. proposed an index structure, which is also called IR tree that stores one integrated inverted file for all the nodes. X. Cao, L. Chen, G. Cong, C. S. Jensen, Q. Qu, A. Skovsgaard, D. Wu, and M. L. Yiu [4] proposed S2I index structure based on R-tree and inverted file. The objects in [4] are stored differently according to the document frequency and infrequency of the term. T.Wang, G. Li, J. Feng [15] proposed a new index structure, spatial keyword R-tree, called SKR-Tree which extended from the R-tree with an R-tree node storing both spatial and keyword information.

D. Zhang, K.L. Tan, Anthony K.H. Tung [16] proposed I3 (Integrated Inverted Index), which adopts the Quad tree structures to hierarchically partition the data space into cells. The basis unit of I3 is the keyword cell, which captures the spatial locality of a keyword. X.Cao, G.Cong, Christian S. Jensen, Jun.J. Ng, BengC.Ooi, N.T. Phan, D. Wu [5] proposes a Web Object Retrieval System (SWORS) that is capable of efficiently retrieving spatial web objects that satisfy spatial keyword queries. This system use IR tree and inverted file for index. It supports two types of queries that are location aware top-k text retrieval (Lkt) query and spatial keyword group (SKG) query.

3. Hybrid Index Structure (K-d Tree and inverted file)

Hybrid geo-textual index structure [1][2][3] that integrates location index and text index to efficiently process spatial keyword queries. In this structure, K-d tree is combined with inverted file. K-d tree is used for spatial queries and inverted file is used for keywords information that is the most efficient index for text information retrieval. For each node of K-d tree, an inverted file is created for indexing the text components of objects contained in the node. As K-d trees represent a disjoint partition, this index

structure can't cause more IO costs and also K-d trees don't need to rebalance the textual information so it can reduce update cost (CPU costs).

Most geo-textual indices use the inverted file for text indexing. Inverted file can be used to check the query keywords contain or not. K-d tree structure is known as point indexing structures as it is designed to index data objects which are points in a multi-dimensional space.

4. Proposed System

The proposed system considers for processing the spatial keyword queries with Myanmar language. The hybrid index structure that combines the K-d tree and inverted file is used to efficiently retrieve the user desired information. In this index structure, K-d tree is used for spatial queries and inverted file is used for Myanmar keywords information. The proposed index structure with Myanmar language is shown in Figure 1 and the example Dataset is shown in Table 1.

Table 1. Example Dataset

id	Latitude	Longitude	Keywords
Obj1	16.808336	96.162481	ငါးထပ်ကြီး, ဘုရား
Obj2	16.800442	96.162225	ဗိုလ်ချုပ်, ပန်းခြံ, ဥယျာဉ်
Obj3	16.829281	96.155644	ဆီဒိုးနား, ဟိုတယ်
Obj4	16.779581	96.169647	ဂမုန်းပွင့်, ရှော့ပင်း, ဈေးဝယ်, စင်တာ
Obj5	16.837589	96.130664	ဆင်ဖြူတော်, ဟိုတယ်
Obj6	16.816497	96.127464	ဖူဂျီ, ဂျပန်, စားသောက်ဆိုင်
Obj7	16.810881	96.176419	ရွှေဗဟို, ရုပ်ရှင်ရုံ
Obj8	16.820872	96.129361	ရွှေကောင်း, ဟော့ပေါ့, စားသောက်ဆိုင်

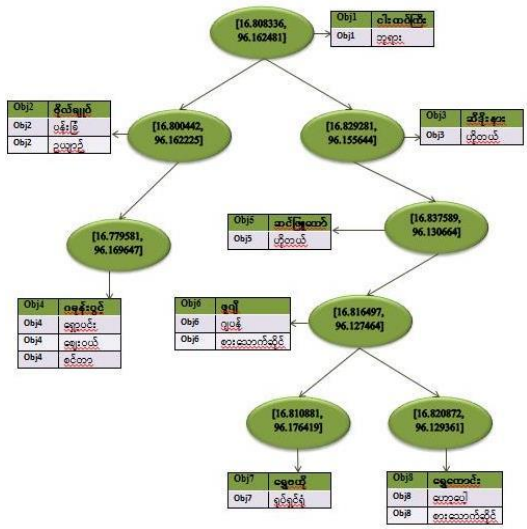

Figure 1. Proposed Index Structure for Dataset of Table 1

Algorithm 1 is porposed range keyword search procedure. The procedure RANGEKEYWORDSEARCH returns all points 'p' such that $d(q,p) \leq r$ and $\sum_{j=1}^{qk.count} keyword \in \sum_{i-1}^{n} p.word_i$. Token each word from user input keywords and then saved in array qk. Use Boolean OR semantics model to check at least one required keywords contain or not in inverted file of each point such that,

$$result = \begin{cases} 1, if\ one\ or\ more\ keywords\ conatin\ in\ p.word \\ 0, otherwise \end{cases}$$

The procedure COMPUTEBBOXES( ) calculates the bounding boxes lBB and rBB for the left and for the right sub tree, respectively. The procedure INTERSECTS(…) tells if the bounding box BB intersects with the region that satisfies the distance constraints. If the intersection is non-empty, the sub tree to be explored. The DISTANCE(…) procedure calculates the distance between two points using Euclidean distance $d(q,p) = \sqrt{(q_{lat} - p_{lat})^2 + (q_{lon} - p_{lon})^2}$.

Algorithm 1. Range Keyword Search Algorithm Using Proposed Index Structure

Input: user's required keyword, K-d tree, query point, range, Max/Min BB
pq: priority queue
qk : array
RANGEKESEARCH (keyword,T,BB,q,r)
if T=leaf then return
p<−T.key; i<−T.discr;
distance<− DISTANCE(q,p);
    if distance ≤ r and $\sum_{j=1}^{qk.count} keyword \in \sum_{i-1}^{n} p.word_i$ then
pq.PUSH (p,distance);
COMPUTEBGBOXES (lBB,rBB,p[i],i)
if INTERSECTS (lBB,q,r) then
    RANGEKESEARCH (keyword, T.left, lBB, c, radius)
if INTERSECTS (rBB,q,r) then
    RANGEKESEARCH(keyword, T.right, rBB, c, radius)

5. Myanmar language

The nature of Myanmar language is complex rather than the English language. Myanmar language has various types of characters such as consonants, medials, vowels, tones, etc. Myanmar word consists of one or more syllables that can contain one or more characters. Myanmar sentences do not have white space to specify words boundaries. Moreover, the sequence of the Myanmar characters is also important for matching the Myanmar word. The typing order of the Myanmar characters may be vary (pigeon, ခိုး → ခ + ◌ိ + ◌ု or ခ + ◌ု + ◌ိ). The sequence of characters must have the same order to match the word. For example, the sequences of လှေ(boat) syllable in Myanmar 3 Unicode is လ + ◌ှ + ◌ေ (101C 103E 1031) [11]. Therefore, the input sequence of characters will be reordered to obtain the correct sequence before the matching process.

6. Experimental Results

The propose system adopts the browser-server model for desktop and laptop computer. Figure 2 shows input required queries for searching. Users can specify the current's location by clicking a location in Google Map to get the latitude and longitude of that location and can type the required keywords with Myanmar language. Then, user can choose the desired range. The query is sent to the server and then and then are displayed on Google Maps in the browser. Figure 3 shows the results after searching. The browser side use Google Map API to provide interfaces to users for generating queries and viewing the returned spatial web objects.

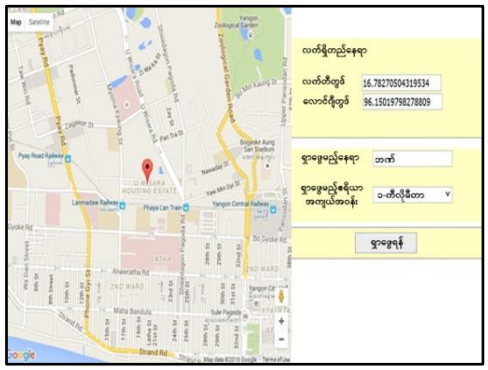

Figure. 2 Input Required Queries for Searching

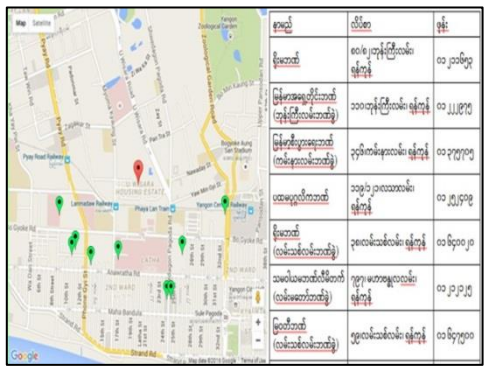

Figure 3. The Results After Searching

Figure 4 compare the searching time (second) between using proposed index structure and without using proposed index. Depending on the desired range (km), searching time is varied. Searching time using index structure is faster than without using index about 100-times in

second. Figure 5 shows the index construction time (second) depending on the size of datasets.

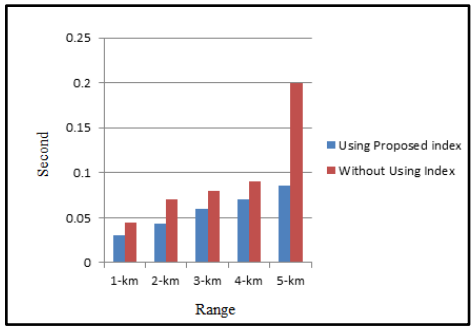

Figure 4. Searching Time in range keyword search

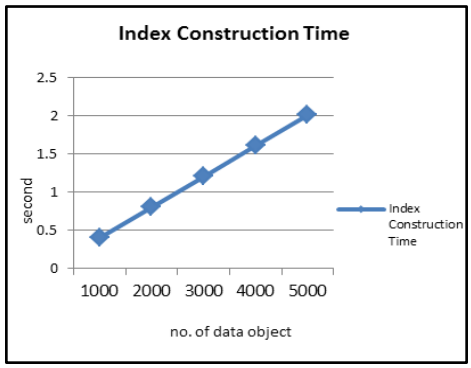

Figure 5. Index Construction Time

7. Conclusion

This paper presented the hybrid index structure that combines the K-d tree and inverted file with Myanmar keyword to efficiently retrieve the user desired information using Myanmar language with minimum IO costs and CPU costs. The result is retrieved from the exact match of the keyword. This proposed system is worked on the desktop version. As a further extension, we will consider the approximate keyword search. Moreover, we will find the desired location with Myanmar language on the mobile version.

References

[1] S. N. Aung, M. M. Sein, "Hybrid Geo-Textual Index Structure for Spatial Range Keyword Search", *Computer Science & Engineering: An International Journal (CSEIJ)*, Vol. 4, No.5/6, December 2014.

[2] S. N. Aung and M. M. Sein, "K-Nearest Neighbours Approximate Keyword Search for Spatial Database", in Proceedings of 9thInternational Conference on Technological Advances in Electrical, Electronics and Computer Engineering (ICTAEECE), Bangkok, Thailand, 7th February 2015, pp. 65-68.

[3] S. N. Aung and M. M. Sein, "Index Structure for Nearest Neighbors Search with Required Keywords on Spatial Database", the 9th International Conference on Genetic and Evolutionary Computing (ICGEC 2015), Yangon, Myanmar, 26-28 August 2015.

[4] X. Cao, L. Chen, G. Cong, C. S. Jensen, Q. Qu, A. Skovsgaard, D. Wu, and M. L. Yiu, "Spatial keyword querying", in*ER*, 2012, pages 16–29.

[5] X. Cao, G. Cong, Christian S. Jensen, Jun.J. Ng, BengC.Ooi, N.T. Phan, D. Wu, "SWROS: A System for the Efficient Retrieval of Relevant Spatial Web Objects".

[6] A. Cary, O. Wolfson, and N. Rishe, "Efficient and scalable method for processing top-k spatial Boolean queries", in*SSDBM*, 2010, pages 87–95.

[7] G. Cong, C. S. Jensen, and D. Wu, "Efficient retrieval of the top-k most relevant spatial web objects", *PVLDB*, 2009, 2(1):337–348.

[8] I. D. Felipe, V. Hristidis, and N. Rishe, "Keyword search on spatial databases", in *ICDE*, 2008, pages 656–665.

[9] R. G¨obel, A. Henrich, R. Niemann, and D. Blank, (2009), "A hybrid index structure for geo-textual searches", in *CIKM*, 2009, pages 1625–1628.

[10] R. Hariharan, B. Hore, C. Li, and S. Mehrotra, (2007), "Processing spatial-keyword (sk) queries in geographic

information retrieval (gir) systems", in *SSDBM*, 2007, page 16.

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

[12] 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*, 2011, 23(4):585–599.

[13] J. B. Rocha-Junior, O. Gkorgkas, S. Jonassen, and K. Nørv°ag, "Efficient processing of top-k spatial keyword queries", in *SSTD*, 2011, pages 205–222.

[14] Y. Theodoridis, T. Sellis, "Optimization Issues in R-tree Construction", Technical Report KDBSLAB-TR-93-08.

[15] 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.

[16] D. Zhang, K.L. Tan, Anthony K.H. Tung, "Scalable Top-K Spatial Keyword Search", *EDBT/ICDT*'13 March 18-22, 2013.