ISSN (Print) : 0974-6846
Indian Journal of Science and Technology, *Vol 8(33), DOI: 10.17485/ijst/2015/v8i33/72962, December 2015* ISSN (Online) : 0974-5645

Wavelet Tree based Hybrid Geo-Textual Indexing Technique for Geographical Search

Arun Yadav[1*] and Divakar Yadav[2]

[1]Department of Computer Science and Engineering, Ajay Kumar Garg Engineering College, Ghaziabad – 201009, Uttar Pradesh, India; ak_ydv@yahoo.com
[2]Department of Computer Science, Jaypee Institute of Information Technology, Noida – 201309, Uttar Pradesh, India; dsy99@rediffmail.com

Abstract

Background/Objectives: There is significant commercial and research interest in location based search for search engines. Searching of keywords belonging to one or more locations (geographic references) requires geographical web search and ranking on the basis of spatial and textual relevancy. This type of search sets the requirement of spatial and textual indexing. **Methods/Statistical Analysis:** This paper uses a new spatial-textual hybrid indexing technique, based on Wavelet Tree (WT) to handle point and region queries for Geographical Information Retrieval. Here, WT data structure is used for both textual and spatial indexing. Minimum Bounding Rectangles (MBRs) of different geographical points (latitude, longitude) is created for designing hybrid index. For searching textual keywords, we need to design inverted index. It is created using wavelet tree. Also, a spatial-textual relevancy scheme is used for relevant document retrieval to the end users. **Findings:** The algorithm has been implemented in order to measure the performance in terms of search time. Approximately 40,000 Wikipedia pages have been crawled and stored in database along with geographical coordinates (latitude, longitude) of locations in India to design MBRs of these locations. The results show that wavelet tree based hybrid index algorithm performance increase with the increase in query length. For small query length, B/R* tree performs better but for larger query lengths, wavelet tree based hybrid index outperforms other techniques. Precision and recall of web documents have also been calculated using hybrid index. For varying query lengths, the precision and recalls are varying which shows that by reducing the time in search time precision and recall are preserve. **Applications/Improvement:** Our algorithm outperforms the existing algorithms both in terms of simplicity in implementation and searching time. In future we will propose a compression technique on hybrid index to minimize the space taken by hybrid index that will further improve the searching time in case of single as well as multiple geographical references of documents.

Keywords: Hybrid-indexing, Indexing, Information Retrieval, Wavelet Tree

1. Introduction

Search engines are very much similar to database systems where documents are stored in a repository and an index is maintained. Queries are evaluated by processing the index to identify matches which are then returned to the users. This sets the requirement of developing an indexing structure suitable for thematic as well as spatial document retrieval in efficient manner. This paper presents a wavelet tree based spatial-textual hybrid index for document retrieval. This data structure (wavelet tree) is used to design hybrid index which perform a good trade-off between search efficiency and the storage requirement as well.

In designing Geographical Information Retrieval (GIR) Systems[1,2,3] efficient indices should be applied to both, the keyword-based thematic information and the geographic references to the same document.

Paper[1] have proposed a hybrid index structure which integrated inverted files and R*-trees to handle both textual and location based queries. They concluded that the structure in which first inverted file is used and then R* tree, is more efficient in terms of query time.

* *Author for correspondence*

With this advancement, paper[4] proposed a wavelet tree based structure for representing the geographical data. They represented the Minimum Bounding Rectangles (MBR) for solving the spatial queries with the help of wavelet tree efficiently as compared to other traditional spatial indexes structure.

In this paper, we extend this approach for spatial as well as textual indexes and propose a spatial-textual hybrid index for Geographical Information Retrieval based on the same data structure i.e. wavelet tree. Using wavelet tree approach, we design MBR of all locations (latitude, longitude) and search location at leaf of wavelet tree. In the second phase, we create dynamic inverted file of searched documents of location using wavelet tree for textual keyword. A wavelet tree-based structure allows us to represent minimum bounding rectangles solving geographic range queries in logarithmic time. It has rank and select operations which can be used for document retrieval and traversal of the tree in almost constant time. For a given bitmap R of size k, rank (R, k) returns the frequency of 0/1 till position k and select (R, l) returns the position of l^{th} bit set to 0/1 in B.

2. Indexing Techniques

Internet and information on web is growing rapidly, which require efficient searching strategy to search optimal and relevant data. Data on the web is not only related to text but also many of web users require web pages related to geographical reference. Most of the pages are generally referenced as single geographical location. But, some of the pages belong to more than one geographical reference. The searching of geographically referenced pages or to find most relevant geographical web page among multiple geographical referenced pages, we require an index structure which would be better to search document in terms of spatial as well as textual index in optimal time and with comparatively less memory space.

During last two decades, lot of research has been done on optimization of search results related to textual as well as spatial search. Researchers proposed and implemented many indexes for searching textual and spatial queries. Many data structures have been proposed to design index for textual as well as spatial search.

Spatial indexes are major part for searching spatial documents. Spatial indexes can be used to solve both range queries and point queries of which range queries have proven to be better than Point Access Methods (PAMs). Paper[5] classified these index structures into two categories: Point Access Methods (PAMs) and Spatial Access Methods (SAMs) as shown in Figure 1. PAMs are used to improve the access time of spatial points. SAMs are more general and are used to improve the access time in collections of geographical objects like polygons, lines etc. As per above discussion, we can classify access methods in Figure 1.

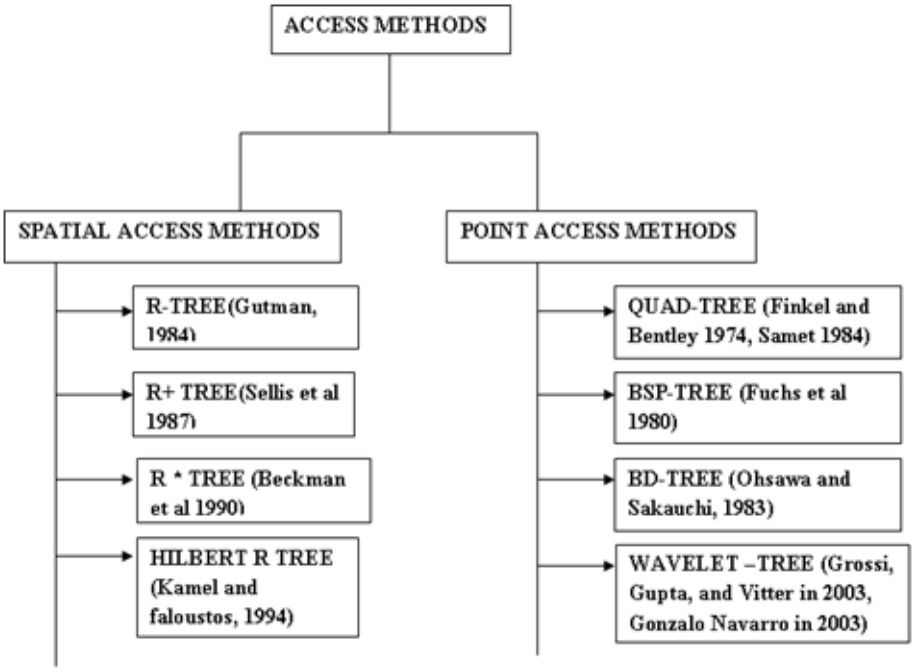

Figure 1. Classification of spatial access methods.

Existing index structures for handling the spatial-keyword queries can be divided into two parts:

- Index using parallel processing.
- Hybrid indexing.

The design of spatial index for spatial data are generally done using R*-tree and inverted file. Inverted file is used for textual search and R*-tree (member of r-tree family) is used for spatial search. Using individual index for textual and spatial search, results of both searches is merged to receive final results. This type of index is useful by using parallel computing[6], also known as dual indexing. Major concern in this index is that it returns large data since it takes large time in sending results. Searching time increases on increasing the query length.

Hybrid index structure has been proposed for textual as well as spatial search for textual followed by spatial or spatial followed by textual using R*-tree (spatial index) and inverted file (textual index). Hybrid index[6] is very useful for small query length, but if query length increases then document search time also increases. For more details about working structure, we can refer to[6].

Query processing and execution are done in two stages which take more time to provide result. In paper[7], proposed hybrid index for large dataset and paper[8] propose hybrid index in the combination of textual-spatial query execution. Our approach is totally different than these existing approaches. We are using same data structure for textual as well as spatial search. We designed bit map of MBRs which takes very less space for designing hybrid index.

In[9] the more refined version of R*-tree hybrid index is given. This hybrid structure is called KR*-tree. Only difference is that it also searches the keyword along with spatial object overlapping the query window and intersects the document of results. KR*-tree is very useful for small keyword and single geographical reference of document. Increasing the query length increases the time to produce the result. In[10, 11] a new hybrid index using R-tree and signature file is proposed. In this hybrid index, each node of R-tree is combined with signature file, during searching all keyword of object is defined as node and other nodes are removed for searching which do not belongs to the search keyword. IR^2 tree is more efficient that R*-tree but signature file provides redundant result during keyword search. That's why IR^2 tree does not perform meaningful result.

In 2009, another hybrid index[12] is presented using R-tree and inverted file which is called as IR-tree. In this structure, each node of R-tee is associated with related inverted file of sub rooted node. Major drawback of this structure is duplication of inverted file and storing of large data on every node of R-tree. It takes more time to search document and more space to store inverted file.

Paper[13] had proposed an index structure which integrated R-tree and inverted file to handle spatial-textual queries. They explained that spatial-textual index is much better than other indices and proposed single as well as double scoring scheme for spatial document relevancy.

In paper[14], proposed an efficient wavelet tree namely wavelet matrix for large alphabets. It is simply an alternative representation for large alphabets which retains all the properties of wavelet trees. It is significantly faster but, bit complex to construct.

3. Proposed Algorithms

This section describes the design of new spatial index (hybrid) to solve geographical queries. Minimum Bounding Rectangles (MBRs) of different geographical points (latitude, longitude) have been created for designing hybrid index. We have used wavelet tree data structure for designing it. The base of index structure is same as R-tree but it increases the response time and reduces the space complexity during execution. At the leaf of index, there is inverted file of geographical documents. For searching textual keywords, we again need to design inverted index. It is also created using wavelet tree. So, this index is very useful for spatial search especially. In both index creation, geographical as well as textual, we use same data structure which can be implemented simultaneously for textual and spatial searching, Other existing indexing techniques uses R-tree family for geographical index and B-tree for inverted index. Through theoretical as well as experimental evaluation we have shown that our proposed hybrid index technique outperforms the existing techniques. In this paper, we have not discussed about, how to find geographic locations from user query because our main focus is to optimize spatial indexes. We propose pseudo code for index construction and searching geographical document using hybrid index technique.

3.1 Algorithm: WGeo Search()

1. Initiate two 2D Array,Array X and Array Y
   //Step1 to Step7 are used for creation of root node

//for array X
2. if upper vertex belongs to MBR
3. then X[i][j]=1
4. else X[i][j]=1
//For array Y
5. if lower right belongs to MBR
6. then Y[i][j]=1
7. else Y[i][j]=0
8. Select array X or array Y as root
9. Create a node with B[n],A[n] //Array B is used storing bitmap associated with node
10. for i=0 to n/2
B[i]=0, A[i]=id
for i= n/2+1 to n
B[i] = 1,A[i]= id //id is considered as identifier
11. for i=0 to n/2
create a left node
for n/2+1 to n
create a right node
12. if (n>1)
then n=n/2,repeat step 10 to 12
13. else stop.

3.2 Search Document from Spatial Index

W Search_document()
1. j←extract_overlap_id() //this function will return the id of overlapped MBR.
2. While(tree!=NULL)
do
3. if(B[j]==0)
4. tree=tree→left
5. j=Rank(B,tree)
6. else
7. tree=tree→right
8. j=rank(B,tree)
9. Search_invertedfile()

3.3 Time Complexity of Algorithm

If we have n number of node then

$2+2^2+\ldots\ldots\ldots\ldots\ldots n$

$2(2^{\log 2n}-1)/2-1=2^{\log 2n+1}-2$

We will search element in this tree. Because it is a binary process with x nodes,we will search in left half or right half ,complexity will be $\log 2^x$.

Now $x = 2^{\log_2 n+1} = 2^{\log_2 n} * 2$

$= \log_2 (2^{\log 2^n} * 2)$

$= \log_2 2^{\log 2^n} + \log_2 2^2$

$= \log_2 n + 1$

Because $2^{\log 2^n} = n$ and $\log_2 2^2 = 1$.

Now complexity of searching of geographical document = lg(n)----------(1)//lgn means log n base2.

For 'm' number of MBRs we will have to traverse the complete wavelet tree from root to the leaf node which will lead to the complexity of m * O (lg(n)).

Therefore, total time complexity = m * O (lg(n))] = ------(2)

For example, to find the documents containing the textual keyword "*colleges*". A wavelet tree of the inverted lists attached to the intersecting MBRs is constructed. Binary search for the root of this wavelet tree require q * O (log p) time for p keywords which are contained in q documents and traversal of the complete tree using rank operation take O(1) time.

Therefore, total complexity of hybrid indexing = [(m* O(lg(n))] + [q * O(lg(p))].----(3)

Complexity given in equation 3 is retrieval of spatial followed by textual data using hybrid indexing and wavelet tree data structure. We compare the complexity of this hybrid index with the complexity of existing indexing structure. We found that our hybrid index structure out perform for increasing the query length. It has been also proved in section 4 by experimenting evaluation of data in Table 1.

Table 1. Comparison of existing algorithm with proposed algorithm

Query Length	B/R	B/R*	Hy B/R	Hy B/R*	Dual W/W	Hy W/W
2	155	139	151	196	161	153
3	229	184	192	215	182	172
4	311	278	287	296	193	181
5	399	362	345	332	221	196
6	526	443	469	415	234	205
7	645	548	537	492	279	223
8	877	712	614	546	318	254

4. Comparison with other Algorithm

The algorithm designed in section 3 is implemented on experimental data. We collected geographical coordinates of 5 states of India. The MBRs have been created and apply algorithm to design spatial hybrid index and searching of web documents. Figure 5 shows the creation of MBRs for designing spatial index. These MBR's are stored in 2D array and design hybrid index using wavelet tree and search web documents.

We implemented the proposed algorithm in order

to measure the performance of preposed techniques in term of search time. We crawled approximately 40,000 Wikipedia pages and stored in database. We also stored geographical coordinates (latitude, longitude) of locations in India to design MBRs of these locations. The algorithm has been implemented in PHP having MYSQL database. The server used was Apache 2.4.4 and the execution platform was windows 8-64 bit having 4GB RAM. In this paper we have compared the proposed wavelet tree based hybrid indexing technique with different existing indexing techniques like B-tree/R-tree (B/R), B-tree/R*-tree (B/R*), hybrid B-tree/R-tree (HyB/R), hybrid -tree/R*-tree (HyB/R*) and wavelet tree based dual index . The results are shown in Table 1 and Figure 2 in tabular as well as graphical form respectively. Results show that wavelet based hybrid index performances increase with the increase in query length. For small query length, B/R* tree performs better but for larger query lengths, wavelet tree based hybrid index outperforms other techniques.

We also computed precision and recall of web documents using hybrid index. For varying query lengths in our experimental setup (in words) the precision and recalls are varying, which shows that by reducing the time in search time precision and recall are preserve. One example is given to find precision and recall from our experimental data.

Query: Schools in India.

Total no of records retrieved: 35.

Total no of records available in DB on that topic: 31.

Total no of relevant records in retrieved records: 27.

Now, to calculate precision and recall we need to check:

A = The number of relevant records retrieved = 27.

B = The number of relevant records not retrieved = 31-27=4.

C = The number of irrelevant records retrieved = 35-27= 8.

Recall = A/(A+B)*100% =27/(27+4)*100% = 87.09%

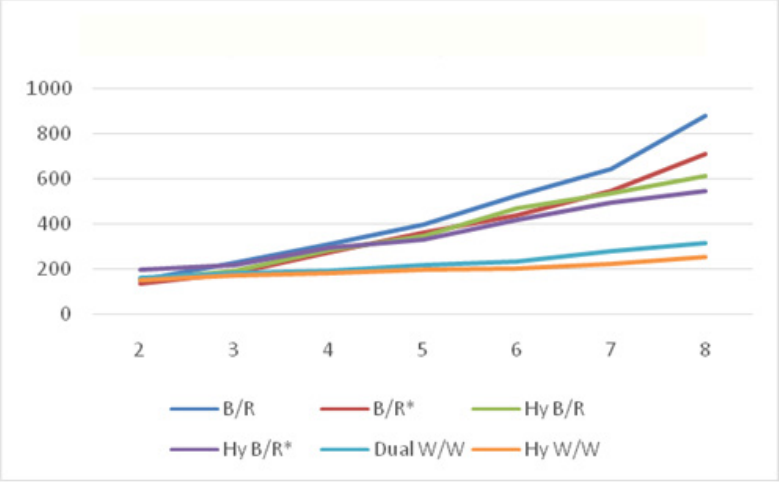

Figure 2. Comparison of wavelet-tree hybrid index with other algorithms.

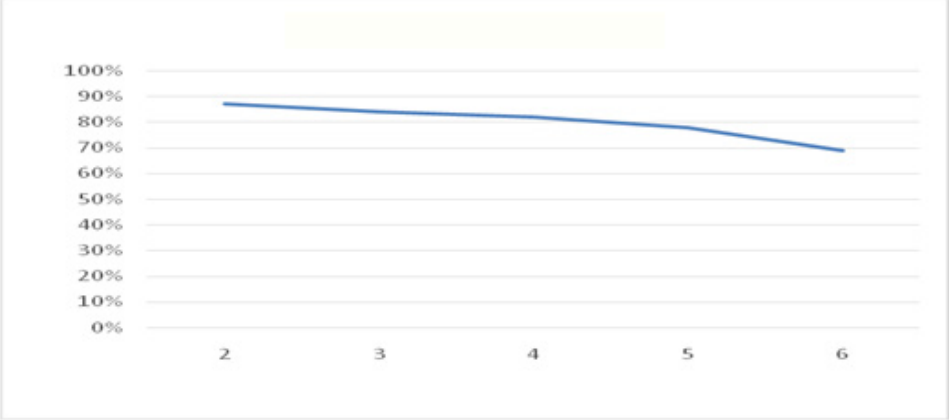

Figure 3. Comparison of wavelet-tree hybrid index with other algorithms.

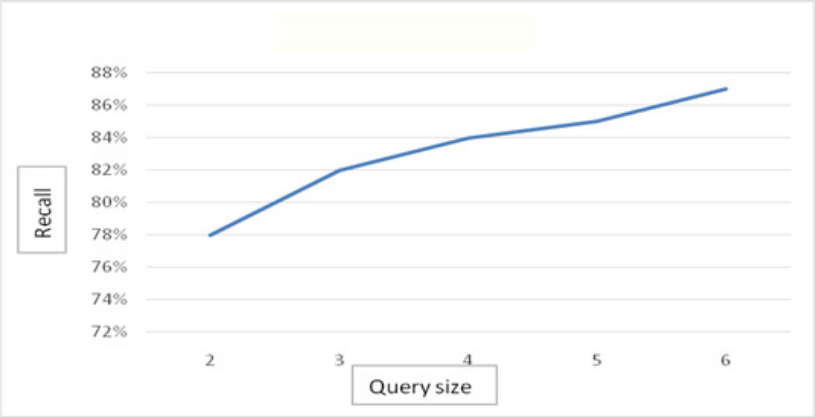

Figure 4. Graph showing the recall w.r.t. query length.

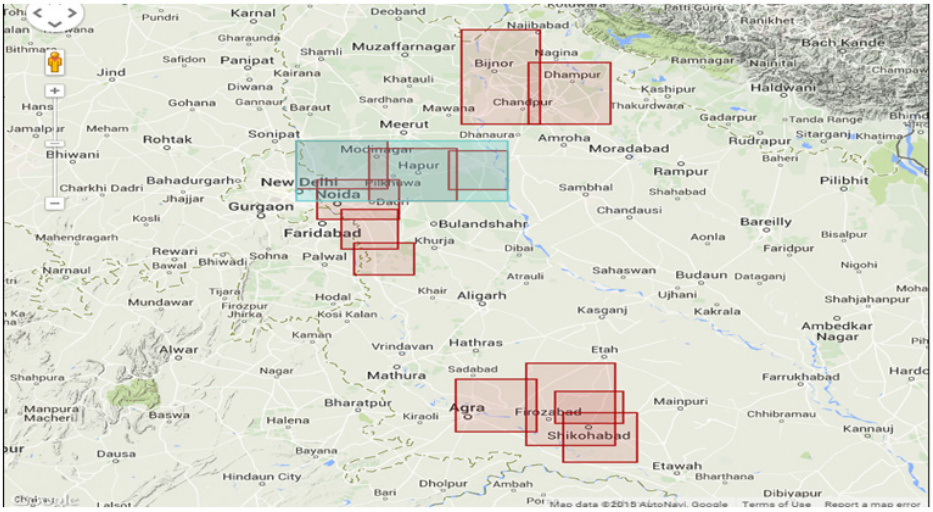

Figure 5. Design of MBRs and overlapped query window.

Precision = A/(A+C)*100% = 27/(27+8)*100% = 77.14%

The precision and recall of 5 different query lengths is given in Table 2 and Table 3. Graph of precision and recall are given in Figure 3 and Figure 4.

Table 2. Precision at different query length

Query size	2	3	4	5	6
Precision	87%	84%	82%	78%	69%

Table 3. Recall at different query length

Query Size	2	3	4	5	6
Recall	78%	82%	84%	85%	87%

5. Conclusion

In this paper we proposed and implemented spatial hybrid indexing technique using single data structure (i.e. wavelets tree) for both, textual as well as for spatial indexing. We proposed two algorithms, one for designing spatial hybrid index and second for searching documents using wavelet tree from hybrid index. The experimental results presented in section 4 shows that our algorithm outperforms the existing algorithms both in terms of simplicity in implementation and searching time.

6. References

1. Yinghua Z, et al. Hybrid index structures for location-based web search. Proceedings of the 14th ACM International Conference on Information and Knowledge Management; 2005. p. 155–62.
2. Sreejith K, Sebastian CD. A novel architecture of perception oriented web search engine based on decision theory. Indian Journal of Science and Technology. 2014, 7(10).
3. Vigneshwari S, Aramudhan M. Social information retrieval

based on semantic annotation and hashing upon the multiple ontologies. Indian Journal of Science and Technology. 2015 Jan; 8(2):103–7.
4. Nieves RB, et al. A fun application of compact data structures to indexing geographic data. Fun with Algorithms. Springer Berlin Heidelberg. 2010; 6099:77–88.
5. Gaede V, Gjnther O. Multidimensional access methods. ACM Computing Surveys (CSUR). 1998 Jun; 30(2):170–231.
6. Zhou Y, Xie X, Wang C, Gong Y, Ma WY. Hybrid index structure for location-based web search. CIKM; 2005. p. 155–62.
7. Chen Y, Markowetz ST. An efficient query processing in geographic web search engines. SIGMOD; 2006. p. 277–88.
8. Vaid S, Jones CB, Joho H, Sanderson M. Spatio-textual indexing for geographical search on the web. SSTD; 2005; 3633. p. 218–35.
9. Hariharan R, Hore B, Li C, Mehrotra S. Processing Spatial-Keyword (SK) queries in Geographic Information Retrieval (GIR) systems. SSDBM; 2007 Jul 9-11. p. 1–16.
10. Felipe DI, Hristidis V, Rishe N. Keyword search on spatial databases. ICDE; 2008 Apr 7-12. p. 656–65.
11. Larson RR. Geographic information retrieval and spatial browsing. Proceedings of the Data Processing Clinic - Geographic Information Systems and Libraries: Patrons, Maps and Spatial Information; 1995 Apr 10-12. p. 81–124.
12. Cong G, Jensen CS, Wu D. Efficient retrieval of the top-k most relevant spatial web objects. Proc VLDB endow. 2009 Aug; 2(1):337–48.
13. Khodaei A. SKIF-P: A point-based indexing and ranking of web documents for spatial-keyword search. In Proceedings of Geoinformatica. 2011 Oct 22. DOI: 10.1007/s10707-011-0142-7.
14. Francisco C, Navarro G, Ordonez A. The Wavelet Matrix: An efficient wavelet tree for large alphabets. Information Systems. 2015 Jan; 47:15–32.