A Hybrid Index Structure for Geo-Textual Searches

Richard Göbel
richard.goebel@fh-hof.de

Andreas Henrich
andreas.henrich@uni-bamberg.de

Raik Niemann
raik.niemann@fh-hof.de

Daniel Blank
daniel.blank@uni-bamberg.de

ABSTRACT

The efficient execution of multi-criteria queries has gained increasing interest over the last years. In the present paper we propose an R-tree based approach for queries addressing textual as well as geographic filter conditions. Whereas most previous approaches use an index structure optimised for a single criterion adding special treatment for the other criterion at the leaf nodes or end points of this index structure, our approach uses a deeper integration. In short, R-trees are maintained for certain subsets of the whole term set. Furthermore, in each of these R-trees bit sets are used within the nodes to indicate whether entries for the terms associated with the single bits can be found in the corresponding sub-tree. Our index structure aims to be both, time and space efficient. The paper investigates the efficiency and applicability of the proposed index structure via practical experiments based on real-world and synthetic data.

Categories and Subject Descriptors: H3.3 Information Storage and Retrieval: Information Search and Retrieval [search process]; H3.4 Information Storage and Retrieval: Systems and Software [Performance Evaluation]

General Terms: Algorithms, Performance

Keywords: R-Tree, Inverted Index, Geographical IR

1. INTRODUCTION

Information systems increasingly manage heterogeneous data types. Beside conventional object-relational data also documents with text and images as well as spatially referenced data needs to be handled. Although specialised solutions exist for all this data, seamless access to these different types of data is still not obvious and often not sufficiently supported by existing systems. A key challenge is the fast access with search conditions combining textual and spatial criteria. The major drivers for this domain are probably spatial search engines as the next generation Internet search engines (e.g. [1, 9, 10]). This paper proposes a hybrid index structure addressing this challenge. As a general purpose index structure seems to be still difficult, we assume two constraints motivated by the specific application domain of a search engine for news: (1) We consider only point coordinates as geographical positions since news articles usually provide the location by the name of a larger town or a city. As a consequence the geographic footprint of a document is stored as one or more point positions. (2) We assume a static set of positions. This constraint is motivated by the use of a given gazetteer for deducing the geographical positions from place names.

2. ANALYSIS OF EXISTING INDEX STRUCTURES

Index structures support quick navigation to search results avoiding the checking of every individual entry. A typical index structure is the B-tree (see [2]) which is available in every (O)RDBMS. Text databases use an inverted index which is more appropriate in this context. An inverted index manages a set of terms where every term points to all documents containing it. Spatial searches are usually supported by specialised index structures such as the R-tree and its variants (see [7, 3]). Although R-trees are probably the best indexing method for spatial search at present, it is difficult to ensure fast responses for very large databases in every case (e.g. [11]). For some special cases however this tree can be optimised to meet efficiency requirements (e.g. [6]).

A spatial search engine needs to support search conditions combining textual and spatial search criteria. This means that specific search indexes need to be available for this purpose. In general three different options are possible:

1. Independent indexes are built for the spatial and the textual criteria. Separate searches are executed for each criterion and the intersections of all intermediate results are computed. This approach is slow if large intermediate result sets need to be handled.

2. The index structures for spatial and text search are connected with each other. Here either the locations of an R-tree point to inverted indexes (2.1) or the terms of an inverted index point to R-trees (2.2).

3. A search could be supported by a single index structure managing both, positions and terms.

Option 2 is described in [13]. The authors compare option 2.1 and 2.2 by applying them to some test data. These first tests seem to indicate that searches can be processed faster for option 2.2. However, no worst case analysis for time

and space is provided and no hybrid solution in between 2.1 and 2.2 is considered. A worst case analysis is given in [5]. In option 2.1 the worst case occurs for a large geographical region and a rare term. In this case most or even all nodes of the R-tree need to be searched, without returning any search results. Although this case does not occur for option 2.2 this option uses significantly more space than option 2.1 because many terms will occur in a large number of documents. This may result in space requirements which grow significantly faster than linear.

Hybrid index structures provide information in every node on both, its spatial extend and the terms contained in documents below it. In the geospatial case the spatial region is usually provided by upper and lower bounds for longitude and latitude (bounding box). The set of terms available below a node may be represented either directly by these terms (e.g. coded by integers) or by bit lists where every position represents the presence or absence of a term.

So far only few papers address concepts combining spatial and textual criteria in one node. [8] proposes the KR*-tree assigning a list of terms to every node. Bit list approaches are proposed by [4] for answering geographic nearest neighbour queries including search terms and [12] considering the problem of retrieving a set of k coordinates with the least spatial extension for k entries with alternative coordinates. For these papers space requirements are not yet sufficiently addressed. It is obvious that the bit list will probably require less space. But even bit lists may become too large for a spatial search engine.

The present paper proposes an approach which excludes terms occurring only in few documents from these lists. Queries containing these rare terms are treated separately from other queries. The few documents for a rare term are searched sequentially. Such a sequential search can be performed fast due to the small number of documents.

3. A HYBRID INDEX STRUCTURE

With the aforementioned ideas in mind we propose an index structure made up of the following three components:

1. On the top level there is a global inverted index. This index maintains document lists and document counts for all terms.
2. In addition for the more frequent terms (occurring in more than *RLimit* documents) zero, one, or more extended R-trees are maintained. These R-trees are copies of a precomputed R-tree storing all point positions from the gazetteer. In addition to the usual bounding boxes representing the geographical extend of a subtree these R-trees maintain bit lists indicating whether a certain term occurs in the documents maintained in a subtree. The size of the bit lists is limited by the parameter *BLength*. If the space of the bit lists in an R-tree is exhausted, a new copy of the precomputed R-tree is generated and terms becoming more frequent than *RLimit* in the future are associated with this new extended R-tree.
3. Finally, for each point position maintained in the leaf nodes of the R-trees a local inverted index is maintained managing all documents for this position.

Figure 1 gives an overview of the hybrid index structure which will be explained in more depth in the following.

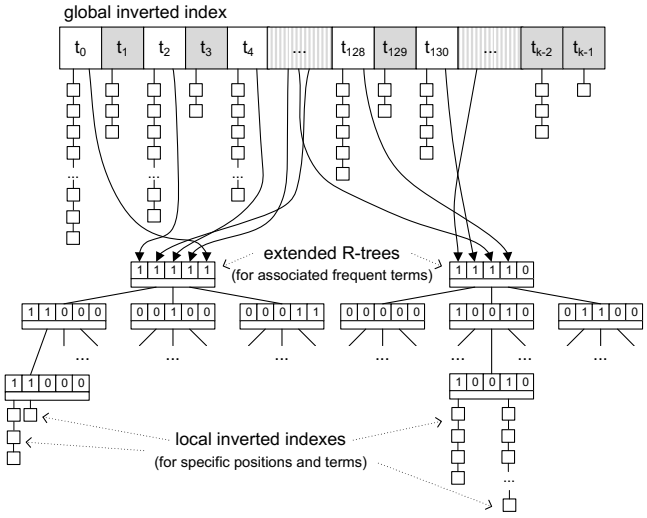

Figure 1: Overview of the proposed hybrid index structure, assuming a value of $RLimit = 3$ and $BLength = 5$; rare terms for which no tree structure is maintained are shaded grey

Documents are at the beginning only added to the global inverted index. This index provides also information about the frequency of terms (how many documents contain this term). If this frequency exceeds a certain limit (*RLimit*) for a term t_i, then the information for term t_i is also stored in an extended R-tree. For this purpose the term in the global inverted index additionally points to the corresponding tree structure.

One important characteristic of our hybrid index structure is that up to *BLength* terms (which occur in more than *RLimit* documents) are maintained in one extended R-tree. Each reference to a node in the extended R-tree is accompanied by a bounding box (maximum and minimum latitude and longitude) enclosing all positions associated with documents in that subtree and a bit list of length *BLength* in which one bit is associated with each term maintained in the extended R-tree. This bit is set to 1 if the term occurs in at least one document in the subtree, and to 0 otherwise.

To start with, the hybrid index structure is initialised as an empty global inverted index. Furthermore, a static tree structure containing all point positions from the Gazetteer[1] is created. We use a static R-tree containing all geographical positions from the gazetteer as a blueprint, since R-trees containing all documents for *BLength* terms will have to maintain a huge subset of these positions anyway for realistic values of *BLength*. Furthermore, this allows for the efficient application of tree packing algorithms. Although this blueprint contains already a bit list of the specified length (*BLength*), all positions are unassigned. A first copy of this tree structure becomes the "current tree structure" for the global inverted index.

The insertion of a new document works as follows: For every contained term the algorithm adds a reference into the global inverted index. A term will also be added to the associated tree if the number of references exceeds *RLimit*. This

[1]In our experiments Geonames (http://www.geonames.org, last visit: 8.6.09) is used with approximately 150,000 entries, i.e. point positions, after some data cleansing.

may require that a reference to the current tree is assigned to the considered term, if *RLimit* has not been exceeded before the insertion of the new document.

A reorganisation of the extended R-tree is required if the bit list has reached its capacity *BLength*. In this case the blueprint is used to generate a new R-tree. Afterwards two options exist:

- All further terms are integrated into the new tree. To this end, the new tree becomes the "current tree". This approach has the property that terms in the new tree tend to have a lower frequency than terms in the previous trees.
- The assignment of terms to trees is completely reorganised. Although this approach requires a significant effort it supports the optimisation of the distribution of terms in the available trees. As an example the frequency of terms could follow the same distribution for every tree. This would mean that all trees contain frequent and rare terms. As an alternative, a query log could be analysed and the distribution of the terms over the R-trees could be optimised in a way that the probability of terms usually occurring in one query to be maintained in one R-tree is maximized. However, such optimisations are future work.

It remains to mention that for each position in an extended R-tree a local inverted index is maintained together with a bit list indicating whether the associated terms occur in documents associated with that position or not. Consequently, lists with document references exist only for terms with a 1 at the associated position of the bit list (cf. figure 1).

A search request with a bounding box and a single term is processed as follows:

The algorithm uses the frequency of the search term to decide whether the search is performed by the global inverted index or by the related tree structure. In the case of a search in the global inverted index (search term occurs in at most *RLimit* documents) all document descriptions referenced by the term are searched sequentially to identify all documents meeting the spatial criterion. Otherwise (search term occurs in more than *RLimit* documents) the extended R-tree associated with the search term is traversed using the bounding rectangles and the bit lists for pruning and finally the encountered local inverted indexes are used to calculate partial results which are merged subsequently.

A search with multiple terms (connected with a logical AND) is also well supported by this index structure:

If one of the conjunctive terms is a rare term, then the corresponding documents in the global inverted index are sequentially searched to check the spatial search condition and the presence of the other terms. If none of the search terms is a rare term, then every term is assigned to an R-tree. In the optimal case all search terms refer to a single tree and the search can be performed using a bit list as a search pattern. With this approach the search procedure identifies a number of point positions each referencing a set of documents via an inverted index. At this point the inverted index of a point position is used to identify all result sets for every search term. The intersection of these sets provides the documents for a point position containing all search terms.

It may also happen that the search terms reference different tree structures. Here it is necessary to generate the result sets for every tree structure separately and compute the intersection of these sets. This computation may be costly if these result sets are large. In this case it could help if frequent terms are stored in a single tree and the other trees contain less frequent terms. Then, no intersections would need to be generated between large result sets of very frequent terms, since these terms are maintained in one tree structure.

4. EVALUATION

A theoretical analysis of hybrid index structures is given in [5]. This analysis shows, that these hybrid index structures have both moderate space requirements and acceptable search efficiency. Under certain constraints these index structures can even ensure a logarithmic search time complexity in the average case.

The proposed concept was evaluated by a prototype supporting the generation of the hybrid index as well as the execution of search requests. The generation is controlled by several parameters including the capacity of inner nodes and leaf nodes and the target ratio between width and height of bounding rectangles for tree nodes.

The index structure was tested with synthetic data and real data. Tests with synthetic data facilitate a detailed analysis of strength and weaknesses of such an index structure because the distribution and size of these data sets can be controlled completely.

Figure 2 summarises the result of an experiment with synthetic data. This figure shows the search effort relative to a growing number of points (positions in the gazetteer) and documents. This experiment uses a Gaussian distribution for the points which are randomly assigned to documents (5 positions per documents). Every term is randomly assigned to $\frac{1}{i}$ documents where i is the position of the term in an ordering of the terms simulating a Zipf distribution. Queries consist of a rectangle and a search term occurring in one out of 500 documents. The experiment starts with 100 points and 1000 documents. Both values are doubled for every step. At the same time the area of the spatial search rectangle is decreased by a factor of 2. This figure seems to confirm that the search effort does not grow faster than logarithmic.

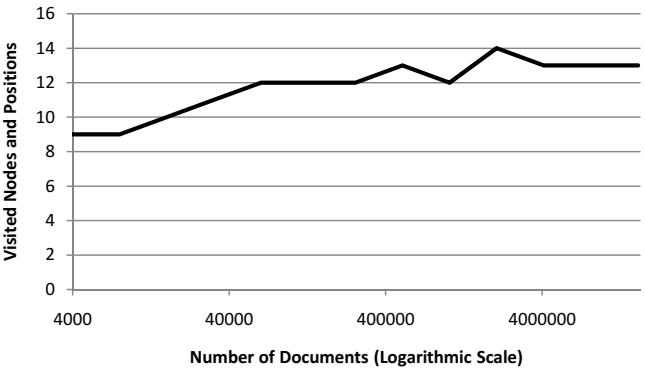

Figure 2: Search effort for synthetic data

Figure 3 summarises the results of an experiment with real world data. The selected data set consists of geotagged Wikipedia articles from the English Wikipedia. Every ar-

ticle is annotated with a single lat/long-pair[2]. For this experiment about 100,000 English articles were used. The experiment was performed for subsets with different sizes. For every size 100 searches were performed. The centre coordinates of a search region were randomly generated assuming a uniform distribution. The size of the search rectangle was manually adjusted to ensure the same number of results for each subset. Search conditions include a single frequent term occurring in 16,180 documents (solid line) or a single rare term occurring in 260 documents (dashed line). As expected the search effort for a rare term is significantly below the effort for a frequent term. Also here both lines seem to indicate that the search complexity does not grow faster than logarithmic.

Several other experiments were performed. All of these experiments seem to confirm that the search time does not grow faster than logarithmic for this index structure. This is somehow surprising since the theoretical analysis does not strictly require this behaviour in every case. In further experiments we plan to identify critical cases for this index structure.

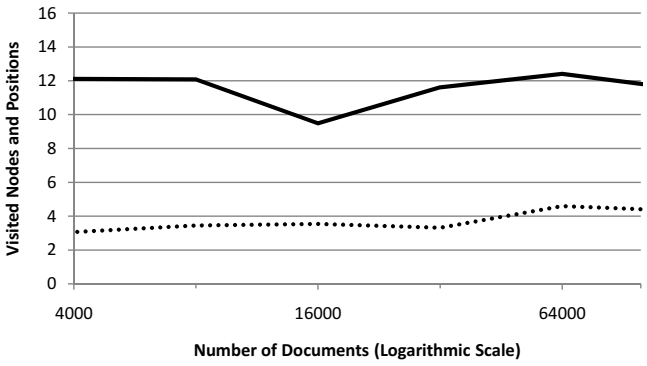

Figure 3: Search effort for Wikipedia articles

5. CONCLUSION

In the present paper we have described a novel hybrid index structure designed for the efficient execution of queries using textual and geographical selection conditions at the same time. The structure maintains classical inverted lists for rare terms—occurring in no more than *RLimit* documents—and additional extended R-trees for more frequent terms. To keep the overhead small up to *BLength* terms are mapped to one R-tree. *RLimit* and *BLength* are useful design parameters for tuning the index structure. To improve the selectivity of the structure bit lists are maintained in each node to indicate whether documents containing the corresponding terms can be found in the respective subtree.

Although the present paper describes only a first version of this new hybrid index structure the preliminary experimental results are encouraging. Nevertheless, various open issues and optimisation opportunities exist. This includes a more detailed investigation of the design parameters *RLimit* and *BLength* and its impact in real world scenarios. This also includes the analysis of allocation techniques associating terms to extended R-trees considering objectives like load balancing or performance for multi term queries.

[2]http://de.wikipedia.org/wiki/Wikipedia:WikiProjekt_Georeferenzierung/Wikipedia-World, last visit: 2.6.09

6. REFERENCES

[1] E. Amitay, N. Har'El, R. Sivan, and A. Soffer. Web-a-where: Geotagging web content. *Proceedings of 27th International ACM SIGIR Conference, pages 273 - 280, Sheffield, UK, 2004.*

[2] R. Bayer and C. McCreight. Organization and maintenance of large ordered indexes. *Acta Informatica 1(3), pages 173 - 189, 1972.*

[3] N. Beckmann, H.-P. Kriegel, R. Schneider, and B. Seeger. The R*-tree: an efficient and robust access method for points and rectangles. *SIGMOD Rec.*, 19(2):322–331, 1990.

[4] I. Felipe, V. Hristidis, and N. Rishe. Keyword search on spatial databases. *Proceedings of 24th International Conference on Data Engineering, pages 656 - 665, Cancun, Mexico, 2008.*

[5] R. Göbel and A. de la Cruz. Computer science challenges for retrieving security related information from the internet. *In: G. Zeug and M. Pesaresi, M. (Eds.), 2007. Global Monitoring for Security and Stability (GMOSS) - Integrated Scientific and Technological Research Supporting Security Aspects of the European Union, EU-report EUR 23033 EN.*

[6] R. Göbel. Towards logarithmic search time complexity for two-dimensional R-trees. *Proceedings of Second International Conference on Systems, Computing Sciences and Software Engineering, Springer, ISBN: 978-1-4020-6267-4, 2007.*

[7] A. Guttman. R-trees: A dynamic index structure for spatial searching. *Proceedings of ACM SIGMOD Conference, Boston, pages 47 - 57, 1984.*

[8] R. Hariharan, B. Hore, C. Li, and S. Mehrotra. Processing spatial-keyword (SK) queries in geographic information retrieval (GIR) systems. *Proceedings of 19th International Conference on Scientific and Statistical Database Management, Banff, Canada, 2007.*

[9] J. T. Heuer and S. Dupke. Towards a spatial search engine using geotags. *Proceedings of GI-Days 2007 - Young Researchers Forum, Florian Probst and Carsten Keßler (Eds.), IfGIprints 30. ISBN: 978-3-936616-48-4, 2007.*

[10] C. B. Jones, A. A. I, D. Finch, G. Fu, and S. Vaid. The spirit spatial search engine: Architecture, ontologies and spatial indexing. *Proceedings of 3rd International Conference on Geographic Information Sciences, pages 125 - 139, Adelphi, MD, USA, 2004.*

[11] K. R. Kanth and A. Singh. Optimal dynamic range searching in non-replicating index structures. *Proceedings of International Conference on Database Theory, pages 257 - 276, Springer Verlag Berlin Heidelberg, 1999.*

[12] D. Zhang, Y. Chee, A. Mondal, A. Tung, and M. Kitsuregawa. Keyword search in spatial databases: Towards searching by document. *Proceedings of IEEE International Conference on Data Engineering, pages 688 - 699, Shanghai, China, 2009.*

[13] Y. Zhou, X. Xie, C. Wang, Y. Gong, and W. Ma. Hybrid index structures for location-based web search. *Proceedings of 14th ACM International Conference on Information and Knowledge Management, pages 155 - 162, Bremen, Germany, 2005.*