ISSN:0975-9646
P. Sunil Kumar Reddy et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 7 (1) , 2016, 312-317

Adjacent Search Outcomes with Keywords

P. Sunil Kumar Reddy[1], Prof. P. Govindarajulu [2]

[1]Research Scholar, [2]Professor

[1, 2]Department of MCA,

[1, 2]S.V. University,

***Abstract*- spatial databases manage three-dimensional objects (such as points, rectangles, etc.), and provides quick access to those objects supported completely different selection criteria. The importance of spacial databases is mirrored by the convenience of modelling entities of reality in a very geometric manner. for instance, locations of restaurants, hotels, hospitals and then on ar typically diagrammatic as points in a very map, whereas larger extents like parks, lakes, and landscapes typically as a mix of rectangles. several functionalities of a spacial information ar helpful in varied ways in which in specific contexts. as an example, in a geographics system, range search may be deployed to search out all restaurants in a bound space, whereas adjacent search retrieval will discover the eating place nearest to a given address. standard spacial queries, like range search and adjacent search retrieval, involve solely conditions on objects geometric properties. Today, several fashionable applications incorporate novel sorts of queries that aim to search out objects satisfying each a spacial predicate, and a predicate on their associated texts.**

We design a variant of inverted index that's optimized for dimensional points, and is therefore named the spatial inverted index (SI-index). This access methodology with success incorporates purpose coordinates into a standard inverted index with tiny further area, attributable to a fragile compact storage theme. Meanwhile, an SI-index preserves the spatial neighbourhood of information points, Associate in Nursingd comes with an R-tree designed on each inverted list at very little area overhead. As a result, it offers 2 competitive ways in which for question process. we will (sequentially) merge multiple lists considerably like merging ancient inverted lists by ids. instead, we will conjointly leverage the R-trees to browse the purposes of all relevant lists in ascending order of their distances to the question point

Keywords: Spatial databases, Spatial quiries,SI-index

I. INTRODUCTION

Spatial info manages three-dimensional objects (such as points, rectangles, etc.), and provides quick access to those objects supported totally different choice criteria. The importance of abstraction databases is mirrored by the convenience of modeling entities of reality during a geometric manner. for instance, locations of restaurants, hotels, hospitals then on square measure usually pictured as points during a map, whereas larger extents like parks, lakes, and landscapes usually as a mixture of rectangles. several functionalities of a abstraction info square measure helpful in varied ways in which in specific contexts. for example, during a earth science data system, vary search may be deployed to search out all restaurants during a bound space, whereas adjacent search retrieval will discover the building nearest to a given address.

Today, the widespread use of search engines has created it realistic to put in writing abstraction queries during a spic-and-span method. Conventionally, queries specialise in objects' geometric properties solely, like whether or not some extent is during a parallelogram, or however shut 2 points square measure from one another. we've got seen some fashionable applications that decision for the power to pick out objects supported each of their geometric coordinates and their associated texts. for instance, it'd be fairly helpful if a quest engine may be accustomed realize the adjacent building that gives "steak, spaghetti, and sprite" all at identical time. Note that this can be not the "globally" adjacent building (which would are came back by a conventional adjacent search query), however the adjacent building among solely those providing all the demanded foods and drinks.

There square measure simple ways in which to support queries that mix abstraction and text options. for instance, for the higher than question, we have a tendency to might initial fetch all the restaurants whose menus contain the set of keywords , and so from the retrieved restaurants, realize the adjacent one. Similarly, one might conjointly have it off reversely by targeting initial the abstraction conditions—browse all the restaurants in ascending order of their distances to the question purpose till encountering one whose menu has all the keywords. the most important disadvantage of those easy approaches is that they'll fail to produce real time answers on troublesome inputs. A typical example is that the $64000 adjacent search lies quite far-flung from the question purpose, whereas all the nearer searchs square measure missing a minimum of one amongst the question keywords.

In this paper, we have a tendency to style a variant of inverted index that's optimized for three-dimensional points, and is therefore named the abstraction inverted index (SI-index). This access technique with success incorporates purpose coordinates into a standard inverted index with little additional house, thanks to a fragile compact storage theme. Meanwhile, Associate in Nursing SI-index preserves the abstraction neighbourhood of information points, Associate in nursing comes with an R-tree engineered on each inverted list at very little house overhead. As a result, it offers 2 competitory ways in which for question process. we will (sequentially) merge multiple lists much like merging ancient inverted lists by ids. instead, we will conjointly leverage the R-trees to browse the purposes of all relevant lists in ascending order of their distances to the question point. As incontestible by experiments, the SI-index considerably outperforms the IR2-tree in question potency, usually by an element of orders of magnitude.

The rest of the paper is organized as follows. Section two Surveys the previous work associated with ours. Section three defines the matter studied during this paper formally Section four presents an answer supported si index and Section five presents distance browsing rule for activity keyword-based adjacent search search. Section vi proposes the SI-idnex, and establishes its theoretical properties. Section seven evaluates our techniques with intensive experiments. Section eight concludes the paper with a outline of our findings.

II. RELATED WORK

Spatial queries with keywords haven't been extensively explored. within the past years, the community has sparked enthusiasm in learning keyword search in relative databases. it's till recently that focus was entertained to dimensional information [1], [2],. the simplest technique so far for adjacent search search with keywords is owing to Felipe et al. [1]. They nicely integrate 2 well-known concepts: R-tree [4], a preferred abstraction index, and signature file [3], a good technique for keyword primarily based document retrieval. By doing so that they develop a structure known as the IR2-tree [1], that has the strengths of each R-trees and signature files. Like R-trees, the IR2- tree preserves objects' abstraction proximity, that is that the key to finding abstraction queries expeditiously. On the opposite hand, like signature files, the IR2-tree is in a position to filter a substantial portion of the objects that don't contain all the question keywords, so considerably reducing the quantity of objects to be examined.

The IR2-tree, however, conjointly inherits a downside of signature files: false hits. That is, a signature file, owing to its conservative nature, should still direct the search to some objects, although they are doing not have all the keywords. The penalty so caused is that the got to verify Associate in Nursing object whose satisfying a question or not can't be resolved mistreatment solely its signature, however needs loading its full text description, that is pricey owing to the ensuing random accesses. it's noteworthy that the false hit downside isn't specific solely to signature files, however conjointly exists in alternative strategies for approximate set membership tests with compact storage (see [5] and also the references therein).

Therefore, the matter can't be remedied by merely substitution signature file with any of these strategies. during this paper, we tend to style a variant of inverted index that's optimized for dimensional points, and is so named the abstraction inverted index (SI-index). This access technique with success incorporates purpose coordinates into a traditional inverted index with little additional house, because of a fragile compact storage theme. Meanwhile, Associate in Nursing SI-index preserves the abstraction vicinity of knowledge points, Associate in Nursingd comes with an R-tree designed on each inverted list at very little house overhead. As a result, it offers 2 competitory ways in which for question process. we are able to (sequentially) merge multiple lists significantly like merging ancient inverted lists by ids. or else, we are able to conjointly leverage the R-trees to browse the purposes of all relevant lists in ascending order of their distances to the question point. As incontestible by experiments, the SI-index considerably outperforms the IR2-tree in question potency, usually by an element of orders of magnitude.

III. PROBLEM DEFINITIONS

Let P be a collection of flat points. As our goal is to mix keyword search with the present location-finding services on facilities like hospitals, restaurants, hotels, etc., we are going to concentrate on spatiality two, however our technique are often extended to absolute dimensionalities with no technical obstacle. we are going to assume that the points in P have number coordinates, such every coordinate ranges in [0; t], wherever t could be a integer. this can be not as restrictive because it could seem, as a result of although one would really like to impose real valued coordinates, the set of various coordinates expressible underneath an area limit continues to be finite and enumerable; thus, we tend to may still convert everything to integers with correct scaling.

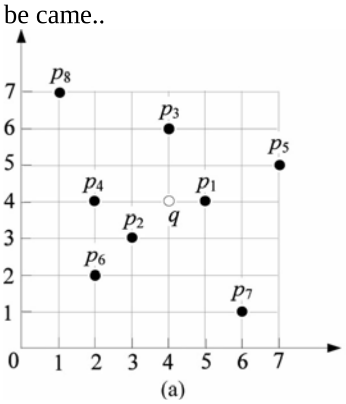
As with [1], every purpose p € P is related to a collection of words, that is denoted as Wp and termed the document of

p. for instance, if p stands for a edifice, Wp are often its menu, or if p could be a edifice, Wp are often the outline of its services and facilities, or if p could be a hospital, Wp are often the list of its out-patient specialities. it's clear that Wp might doubtless contain various words.

Traditional adjacent search search returns the info purpose nearest to a question purpose. Following [1], we tend to extend the matter to incorporate predicates on objects' texts. Formally, in our context, AN adjacent search (NN) question specifies some extent Q and a collection Wq of keywords (we discuss with Wq because the document of the query). It returns the purpose in Pq that's the adjacent to Q, wherever Pq is outlined as

$$P_q = \{p \in P \mid W_q \subseteq W_p\}.$$

In alternative words, Pq is that the set of objects in P whose documents contain all the keywords in Wq. within the case wherever Pq is empty, the question returns nothing. the matter definition can be generalized to k adjacent search (kNN) search, that finds the k points in Pq nighest to q; if Pq has but k points, the whole Pq ought to be came..



(a)

p	W_p
p_1	$\{a, b\}$
p_2	$\{b, d\}$
p_3	$\{d\}$
p_4	$\{a, e\}$
p_5	$\{c, e\}$
p_6	$\{c, d, e\}$
p_7	$\{b, e\}$
p_8	$\{c, d\}$

(b)

Fig. 1. (a) Shows the locations of points (b) gives their associated texts.

For example, assume that P consists of eight points whose locations area unit as shown in Fig. 1a (the black dots), and their documents area unit given in Fig. 1b. contemplate letteruery|a question |a question} purpose q at the white dot of Fig. 1a with the set of keywords Wq = c,d. Adjacent search search finds p6, noticing that each one points nearer to letter than p6 area unit missing either the question keyword c or d. If k = a pair of adjacent searches area unit needed, p8 is additionally came back additionally. The result's still notwithstanding k will increase to three or higher, as a result of solely 2 objects have the keywords c and d at an equivalent time. we have a tendency to contemplate that the info set doesn't slot in memory, and desires to be indexed by economical access strategies so as to reduce the amount of I/Os in responsive a question.

IV. SOLUTION BASED ON INVERTED INDEX

Inverted indexes (I-index) have established to be a good access technique for keyword-based document retrieval. within the spacial context, nothing prevents America from treating the text description Wp of some extent p as a document, and then, building AN I-index. Figure a pair of illustrates the index for the dataset of Figure one. every word within the vocabulary has AN inverted list, enumerating the ids of the points that have the word in their documents. Note that the list of every word maintains a sorted order of purpose ids, that provides extended convenience in question process by permitting AN economical merge step. as an example, assume that we wish to seek out the points that remonstrate c and d. this is often basically to figure the intersection of the 2 words' inverted lists. As each lists area unit sorted within the same order, we are able to do therefore by merging them, whose I/O and hardware times area unit each linear to the entire length of the lists. Recall that, in NN process with IR2-tree, some extent retrieved from the index should be verified (i.e., having its text description loaded and checked). Verification is additionally necessary with I-index, except for precisely the opposite reason. For IR2-tree, verification is as a result of we have a tendency to don't have the elaborated texts of some extent, whereas for I-index, it's as a result of we have a tendency to don't have the coordinates. Specifically, given AN NN question letter with keyword set Wq, the question algorithmic program of I-index 1st retrieves (by merging) the set Pq of all points that have all the keywords of Wq, and then, performs |Pq| random I/Os to urge the coordinates of every purpose in Pq so as to judge its distance to letter.

word	*inverted list*
a	p_1 p_4
b	p_1 p_2 p_7
c	p_5 p_6 p_8
d	p_2 p_3 p_6 p_8
e	p_4 p_5 p_6 p_7

Fig .2 Example of inverted index

According to the experiments of [12], once Wq has solely one word, the performance of I-index is extremely unhealthy, that is anticipated as a result of everything within the inverted list of that word should be verified. apparently, because the size of Wq will increase, the performance gap between Iindex and IR2-tree keeps narrowing such I-index even starts to outgo IR2-tree at |Wq|=4. this is often not as stunning because it could appear. As |Wq| grows giant, not several objects ought to be verified as a result of the amount of objects carrying all the question keywords drops speedily. On the opposite hand, at now a plus of Iindex starts to pay off. That is, scanning Associate in Nursing inverted list is comparatively low-cost as a result of it involves solely consecutive I/Os1, as against the random nature of accessing the nodes of Associate in Nursing IR2-tree.

V. MERGING AND DISTANCE BROWSING

Since verification is that the performance bottleneck, we should always attempt to avoid it. there's a straightforward thanks to do therefore in associate degree I-index: one solely has to store the coordinates every|of every} purpose beside each of its appearances within the inverted lists. The presence of coordinates within the inverted lists naturally motivates the creation of associate degree R-tree on every list categorisation the points in that (a structure harking back to the one). Next, we have a tendency to discuss the way to perform keyword-based adjacent search search with such a combined structure.

The R-trees permit United States to remedy associate degree awkwardness within the means NN queries ar processed with associate degree I-index. Recall that, to answer a question, presently we've got to initial get all the points carrying all the question words in Wq by merging many lists (one for every word in Wq). This seems to be unreasonable if the purpose, say p, of the ultimate result lies fairly near the question purpose Q. it might be nice if we have a tendency to may discover p terribly presently all told the relevant lists so the algorithmic rule will terminate directly. {this would|this is able to|this may|this might|this may} become a reality if we have a tendency to could browse the lists synchronously by distances as against by ids. particularly, as long as we have a tendency to may access the points of all lists in ascending order of their distances to Q (breaking ties by ids), such a p would be simply discovered as its copies all told the lists would positively emerge consecutively in our access order. therefore all we've got to try to to is to stay enumeration what number copies of a similar purpose have popped up endlessly, and terminate by news the purpose once the count reaches |Wq| At any moment, it's enough to recollect just one count, as a result of whenever a brand new purpose emerges, it's safe to chuck the previous one.

As associate degree example, assume that we wish to perform NN search whose question purpose Q is as shown in Fig. 1, and whose Wq equals fc; decigram. Hence, we are going to be victimisation the lists of words c and d in Fig. 2. rather than increasing these lists by ids, the new access order is by distance to Q, namely, p2; p3; p6; p6; p5; p8; p8. The process finishes as presently because the second p6 comes out, while not reading the remaining points. Apparently, if k adjacent searches ar wished, termination happens when having reported k points within the same fashion. Distance browsing is straightforward with R-trees. In fact, the best-first algorithmic rule is strictly designed to output knowledge points in ascending order of their distances to Q. However, we have a tendency to should coordinate the execution of best-first on |Wq| R-trees to get a world access order. this could be simply achieved by, as an example, at every step taking a "peek" at consequent purpose to be came from every tree, and output the one that ought to come back next globally. This algorithmic rule is predicted to figure well if the question keyword set Wq is tiny. For sizable Wq, the big variety of random accesses it performs could overwhelm all the gains over the consecutive algorithmic rule with merging.

A serious disadvantage of the R-tree approach is its house price. Notice that some extent has to be duplicated once for each word in its text description, leading to terribly dearly-won house consumption. within the next section, we are going to overcome the matter by coming up with a variant of the inverted index that supports compressed coordinate embedding.

VI. SPATIAL INVERTED LIST

The spatial inverted list (SI-index) is basically a compressed version of associate degree I-index with embedded coordinates as delineated in Section four. question process with associate degree SI-index may be done either by merging, or at the side of R-trees during a distance browsing manner. what is more, the compression eliminates the defect of a standard I-index specified associate degree SI-index consumes a lot of less area.

A. The Compression Scheme

Compression is already wide accustomed cut back the scale of associate degree inverted index within the standard context wherever every inverted list contains solely ids. therein case, a good approach is to record the gaps between consecutive ids, as critical the precise ids. for instance, given a collection S of integers f2; 3; 6; 8g, the gap-keeping approach can store f2; 1; 3; 2g instead, wherever the ith worth (i-2) is that the distinction between the ith and (i-1)Þth values within the original S. because the original S are often exactly reconstructed, no data is lost. the sole overhead is that decompression incurs additional computation price, however such price is negligible compared to the overhead of I/Os. Note that gap-keeping are abundant less helpful if the integers of S aren't during a sorted order. this is often as a result of the area saving comes from the hope that gaps would be abundant smaller (than the first values) and therefore may be portrayed with fewer bits. this might not be true had S not been sorted. pressing associate degree SI-index is a smaller amount simple. The distinction here is that every component of a listing, a.k.a. a point p, could be a triplet (idp; xp; yp) as well as each the id and coordinates of p. As gap-keeping needs a sorted order, it are often applied on only 1 attribute of the triplet. for instance, if we have a tendency to attempt to kind the list by ids, gap-keeping on ids might result in smart area saving, however its application on the x- and y-coordinates wouldn't have abundant result.

To attack this drawback, allow us to 1st pass over the ids and concentrate on the coordinates. even supposing every purpose has 2 coordinates, we will convert them into only 1 in order that gap keeping are often applied effectively. The tool required could be a area filling curve (SFC) like Hilbert- or Z-curve. SFC converts a three-d purpose to a 1D worth such if 2 points square measure march on the first area, their 1D values additionally tend to be similar. As spatiality has been delivered to one, gap-keeping works nicely when sorting the (converted) 1D worth.

p_6	p_2	p_8	p_4	p_7	p_1	p_3	p_5
12	15	23	24	41	50	52	59

Fig. 3. Converted values of the points in Fig. 1a based on Z-curve.

For example, supported the Z-curve, the ensuing values, known as Z-values, of the points in Fig. 1a square measure incontestible in Fig. a pair of in ascending order. With gap-keeping, we are going to store these eight points because the sequence twelve, 3, 8, 1, 7, 9, 2, 7. Note that because the Z-values of all points will be accurately reconditioned, the precise coordinates will be reconditioned moreover. allow us to place the ids into thought. currently that we've got with success forbidden the 2 coordinates with a 2nd SFC, it might be natural to have confidence employing a 3D SFC to deal with ids too. As way as house reduction thinks about, this 3D approach might not a foul resolution.

The problem is that it'll destroy the section of the points in their original house. Specifically, the reborn values would not preserve the spatial proximity of the points, as a result of ids normally don't have anything to try to to with coordinates. If one is concerned the needs of getting Associate in Nursing id, it'll be clear that it primarily provides a token for us to retrieve (typically, from a hash table) the small print of Associate in Nursing object, e.g., the text description and/or different attribute values. moreover, in responsive a question , the ids additionally give the bottom for merging. Therefore, nothing prevents us from employing a pseudo-id internally.

For example, consistent with Fig. 2, p6 gets a pseudo-id zero, p2 gets a one, and so on. Obviously, these pseudo-ids will co-exist with the "real" ids, which may still be unbroken at the side of objects' details. As Associate in Nursing example that offers the total image, contemplate the inverted list of word d in Fig. a pair of that contains p2, p3, p6, p8, whose Z-values square measure fifteen, 52, 12, twenty three severally, with pseudo-ids being one, 6, 0, 2, severally. Sorting the Z-values mechanically additionally puts the pseudo-ids in ascending order. With gap-keeping, the Z-values square measure recorded as twelve, 3, 8, twenty nine and also the pseudo-ids as zero, 1, 1, 4. thus we will exactly capture the four points with four pairs :. Since SFC applies to any spatiality, it's easy to increase our compression theme to any dimensional house.

Let us assume that the total information set has n>1 points and r of them seem in L. to create our analysis general, we tend to additionally take the spatiality d into consideration. Also, recall that every coordinate ranges from zero to t, wherever t may be a whole number. Naively, every pseudo-id will be drawn with O(log n) bits, and every coordinate with O(log t) bits. Therefore, with none compression, we tend to will represent the whole L in O(r(log n + d log t)) bits.

B. Building R-Trees

Our goal is to let every block of an inverted list be directly a leaf node within the R-tree. this is often in distinction to the choice approach of building an R-tree that shares nothing with the inverted list, that wastes house by duplicating every purpose within the inverted list. moreover, our goal is to supply 2 search methods simultaneously: merging and distance browsing (Section 4).

As before, merging demands that points of all lists ought to be ordered following identical principle. this is often not a retardant as a result of our style within the previous section has set down such a principle: ascending order of Z-values. Moreover, this ordering encompasses a crucial property that standard id-based ordering lacks: preservation of abstraction proximity.

The property makes it doable to make sensible R-trees while not destroying the Z-value ordering of any list. Specifically, we are able to (carefully) cluster consecutive points of a listing into MBRs, and incorporate all MBRs into an R-tree. The proximity-preserving nature of the Z-curve can make sure that the MBRs square measure moderately tiny once the spatiality is low. as an example, assume that an inverted list includes all the points in Fig. 3, sorted within the order shown. to make an R-tree, we have a tendency to could cut the list into four blocks , , and . Treating every block as a leaf node ends up in an R-tree a dead ringer for the one in Fig. 3a. Linking all blocks from left to right preserves the ascending order of the points' Z-values.

Creating an R-tree from an area filling curve has been thought of by Kamel and Faloutsos. totally different from their work, we'll examine the matter during a additional rigorous manner, and arrange to get the optimum answer. Formally, the underlying drawback is as follows. there's AN inverted list L with, say, r points p1, p2; . . . ; pr, sorted in ascending order of Z-values. we would like to divide L into variety of disjoint blocks specified (i) the amount of points in every block is between B and 2B-1, wherever B is that the block size, and (ii) the points of a block should be consecutive within the original ordering of L. The goal is to form the ensuing MBRs of the blocks as tiny as doable. the full range of decisions could also be but B-1 as a result of care should be taken to form positive that the amount of these remaining points is a minimum of B. In any case, C[i; j] equals the bottom value of all the permissible decisions, or formally:

$$C[i,j] = \min_{k=i+B-1}^{\min\{i+2B-2,j+1-B\}} (A[i,k] + C[k+1,j]).$$

The equation indicates the existence of solutions supported dynamic programming. One will simply style associate degree rule that runs in O(Br2)time: it suffices to derive C[I, j] in ascending order of the worth of j- i, namely, beginning with those with j- i=2B, followed by those with j-i=2B-1, then on till finishing at j - i = r -1. we are able to considerably improve the computation time to O(Br), by the observation that j will be fastened to r throughout the computation so as to get C[1, r] eventually.

We have finished explaining the way to build the leaf nodes of associate degree R-tree on associate degree inverted list. As every leaf could be a block, all the leaves will be keep in an exceedingly blocked SI-index as delineated in Section six A. Building the nonleaf levels is trivial, as a result of they're invisible to the merging-based question algorithms, and hence, don't ought to preserve any common ordering. we tend to ar absolve to apply any of the prevailing R-tree construction algorithms. it's noteworthy that the nonleaf levels add solely atiny low quantity to the area overhead as a result of, in associate degree R-tree, the quantity of nonleaf nodes is out and away under that of leaf nodes.

VII. EXPERIMENTS

Data. Our experiments area unit supported each artificial and real information. The spatiality is often a pair of, with every axis consisting of integers from zero to 16; 383. The artificial class has 2 information sets: Uniform and Skew, that dissent within the distribution of knowledge points, and in whether or not there's a correlation between the abstraction distribution and objects' text documents. Specifically, every information set has one million points.

We use the name of the subdivision to rummage around for its page at Wikipedia, and collect the words there because the text description of the corresponding datum. All the points, still as their text documents, represent the info set Census. the most statistics of all of our information sets area unit summarized in Table one.

Results on area consumption. we are going to complete our experiments by reportage the area price {of every|of every} technique on each information set. whereas four ways area unit examined within the experiments on question time, there area unit solely 3 as way as area is bothered. keep in mind that SI-m and SI-b really deploy an equivalent SI-index and therefore, have an equivalent area price. within the following, we are going to talk over with them conjointly as SI-index. Fig.3 provides the area consumption of IR2-tree, SI-index, and IFR on information sets Uniform, Skew, and Census, severally.

	number of points	vocabulary size	average number of objects per word	average number of words per object
Uniform	1 million	200	50k	10
Skew	1 million	200	50k	10
Census	20847	292255	33	461

Table 1. Data set

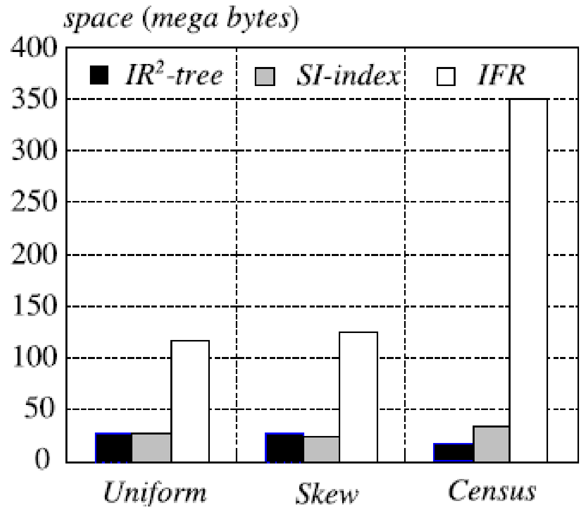


Fig 3 comparison of space consumption

As expected, IFR incurs prohibitively massive area price, as a result of it has to duplicate the coordinates of an information purpose p as repeatedly because the variety of distinct words within the text description of p. As for the opposite strategies, IR2-tree seems to be slightly extra space economical, though such a plus doesn't justify its costly question time, as shown within the earlier experiments. Summary. The SI-index, in the midst of the projected question algorithms, has bestowed itself as

a superb exchange between area and question potency. Compared to IFR, it consumes considerably less area, and yet, answers queries

VIII. Conclusion

In this paper, we have a tendency to style a variant of inverted index that's optimized for four-dimensional points, and is so named the abstraction inverted index (SI-index). This access methodology with success incorporates purpose coordinates into a traditional inverted index with little additional house, because of a fragile compact storage theme. Meanwhile, AN SI-index preserves the abstraction vicinity of knowledge points, ANd comes with an R-tree designed on each inverted list at very little house overhead. As a result, it offers 2 competitory ways that for question process. we will (sequentially) merge multiple lists a great deal like merging ancient inverted lists by ids. or else, we will conjointly leverage the R-trees to browse the purposes of all relevant lists in ascending order of their distances to the question point.

References

[1] I.D. Felipe, V. Hristidis, and N. Rishe, “Keyword Search on Spatial Databases,” Proc. Int’l Conf. Data Eng. (ICDE), pp. 656-665, 2008.

[2] R. Hariharan, B. Hore, C. Li, and S. Mehrotra, “Processing Spatial-Keyword (SK) Queries in Geographic Information Retrieval (GIR) Systems,” Proc. Scientific and Statistical Database Management (SSDBM), 2007.

[3] C. Faloutsos and S. Christodoulakis, “Signature Files: An Access Method for Documents and Its Analytical Performance Evaluation,” ACM Trans. Information Systems, vol. 2, no. 4, pp. 267-288 1984.

[4] N. Beckmann, H. Kriegel, R. Schneider, and B. Seeger, “The Rtree: An Efficient and Robust Access Method for Points and Rectangles,” Proc. ACM SIGMOD Int’l Conf. Management of Data, pp. 322-331, 1990.

[5] B. Chazelle, J. Kilian, R. Rubinfeld, and A. Tal, “The Bloomier Filter: An Efficient Data Structure for Static Support Lookup Tables,” Proc. Ann. ACM-SIAM Symp. Discrete Algorithms (SODA), pp. 30-39, 2004.

[6] [6] X. Cao, G. Cong, C. S. Jensen, and B. C. Ooi. Collective spatial keyword querying. In *Proc. of ACM Management of Data (SIG-MOD)*, pages 373–384, 2011.

[7] B. Chazelle, J. Kilian, R. Rubinfeld, and A. Tal. The bloomier filter: an efficient data structure for static support lookup tables.In *Proc. of the Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 30–39, 2004.

[8] Y.-Y. Chen, T. Suel, and A. Markowetz. Efficient query processing in geographic web search engines. In *Proc. of ACM Managementof Data (SIGMOD)*, pages 277–288, 2006.

[9] [9] E. Chu, A. Baid, X. Chai, A. Doan, and J. Naughton. Combining keyword search and forms for ad hoc querying of databases. In *Proc. of ACM Management of Data (SIGMOD)*, 2009.

[10] [10] G. Cong, C. S. Jensen, and D. Wu. Efficient retrieval of the top-k most relevant spatial web objects. *PVLDB*, 2(1):337–348, 2009.

[11] C. Faloutsos and S. Christodoulakis. Signature files: An access method for documents and its analytical performance evaluation. *ACM Transactions on Information Systems (TOIS)*, 2(4):267–288, 1984.

[12] I. D. Felipe, V. Hristidis, and N. Rishe. Keyword search on spatial databases. In *Proc. of International Conference on Data Engineering (ICDE)*, pages 656–665, 2008.