IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

IMPLEMENTATION AND ANALYSIS OF SEARCHING NEAREST KEYWORD SET USING PROJECTION AND MULTISCALE HASHING

[1]Miss.Patil Trupti Satish, [2]Prof.Mr.Mophare A.V.

[1]Student of M.E.CSE, [2] Professor in Computer Science & Engineering

[1] Department of Computer Science & Engineering ,

N. B. Navale Sinhgad College of Engineering, Pune Road, Kegaon, Solapur, India

Abstract: The existence of keywords allows us for the development of new tools to query and explore these multidimensional datasets. We are going to implement a method called Searching nearest Keyword Set using Projection and Multi Scale Hashing that uses hash-based index structures, and achieves high scalability and speedup. In multidimensional datasets, it is difficult for users to provide meaningful keys, and our work deals with another type of queries where users can only provide keywords as input. results are represented using text information (e.g., tags or keywords) associated with them. Our system is based on real datasets shows that we can show the efficient searching of keywords in multidimensional dataset.

Index Terms **- Searching, Hashing, Datasets, Projection, NKS, Color, Vector, Tags, Keywords**

I. INTRODUCTION

The Objects we can consider text, documents, classified or organized by a collection of relevant features, and are commonly represented as points in a multi-dimensional feature space. For example, images are represented using **color** feature vectors, and generally have descriptive textual contents (e.g., tags or keywords) associated with them. In this article, we implement searching multi-dimensional datasets where each data point has a set of keywords. The presence of keywords in feature space allows for the development of new tools to query and explore these multi-dimensional datasets. In this article, we have implemented and proved **nearest keyword set known as NKS** queries on text-rich multi-dimensional datasets. An **NKS query** is a set of user-provided keywords, and the result of the query may include k sets of data points each of which contains all the query keywords and forms one of the top-k tightest cluster in the multi-dimensional space. To present a Nearest Keyword Set search in Multi-Dimensional Dataset Using Projection Multi Scale Hashing & Ranking Function in which user searches top k- nearest keyword set in multidimensional dataset & find optimal results .

Main contributions are summarized as follows.

1. Propose a novel multi-scale index for exact and approximate NKS query processing.
2. Develop efficient search algorithms that work with the multi-scale indexes for fast query processing.
3. Conduct extensive experimental studies to demonstrate the performance of the proposed techniques.

II. COMPARISON WITH EXISTING SEARCH ENGINES AND PROPOSED SEARCH SYSTEM

S.N.	Parameter	Existing System	Proposed System
1.	Internet /Web	Necessary	Not Required
2.	Memory	Temporary[RAM]	Permanent[RAM/HDD]
3.	Database	ONLINE	OFFLINE
4.	Data Security	No	Yes
5.	Storage Cost	Comparatively High	Comparatively Low
6.	Computing Time	Slow	Fast
7.	New Data Addition Facility	Automatic	Manual
8.	Login Authentication	Not Required	Required
9.	Search Method	Keyword Specific	Nearest Keyword Specific

III. System Model of Proposed Scheme and Algortihm of Projection and Hashing

The below diagram shows the proposed model of our scheme

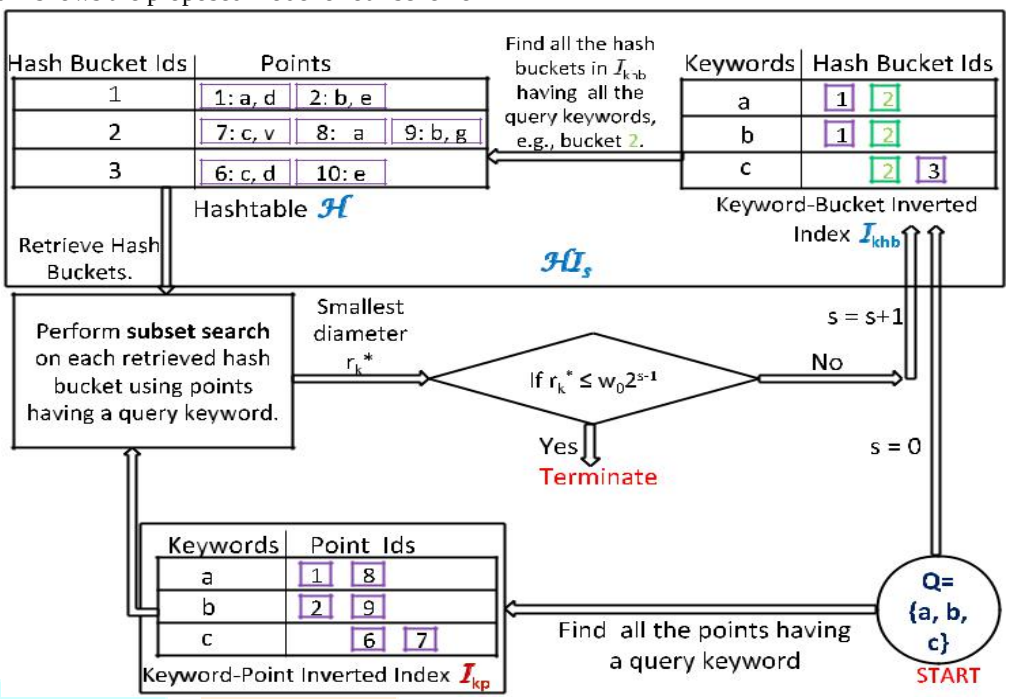


Fig. Index structure and flow of execution of Projection & Multi-Scale Hashing

In the Figure, Hash bucket ids are given for each set of keywords in a particular file and these are retrieved to perform subset search on each obtained hash buckets using points with the keywords. The function Q accepts all the values by assigning initial value to S and it forms hash bucket Id's .If the diameter is small then cluster is formed with the help of this equation (if r*K<=w*2^(s-1)).It is an increment procedure if the hash bucket are not formed before but it is terminated if it is formed already. A hash table is a data structure used to complete an agreeable show, a structure that can depict to values. A hash table uses a hash ability which enrolls a document into an assortment of jars or openings. In this table, all the keywords are matched with the hash bucket ids and then the keywords are joined with these ids and clusters are formed. In the clusters formed we have these keywords grouped and these together form a cluster of essential keywords are in the cluster with Id's in hash table. The user interface is created in such a way that every user has a login id and password and through which the files are being uploaded and then the files are subjected to hashing function in which the proposed system architecture takes place using Projection & Multi-Scale Hashing algorithm and the admin is also provided with a id and password and all the user activity is monitored in here. The admin using ranking algorithm he forms bar graphs for each and every file uploaded the data is then displayed statistically in which the user and the can witness all the information about the file being uploaded and its nature. Ranking is done by three categories where third category is based on the mean of first two categories. It has defined formula to calculate the distance between two keywords. If K is 1, then it selects the nearest keyword and if k>1 then it has two cases:

1. Mean of all nearest keyword values are taken for regression.
2. Nearest neighbor is selected to classify.

ALGORITHM – PROJECTION AND MULTI SCALE HASHING

In: *Q:* query keywords; *k*: number of top results
In: w_0: initial bin-width
1: *PQ* ←[e([],+∞)]: priority queue of top-k results
2: *HC:* hashtable to check duplicate candidates
3: *BS:* bitset to track points having a query keyword
4: for all $o \in U_{\forall vQ \in Q} I_{kp}[v_Q]$ **do**
5: *BS[o]* ←true /* Find points having query keyword*/
6: **end for**
7: **for all** $s \in \{0, \ldots, L-1\}$do
8: Get HI at s
9: *E[]*←0/* List of hash buckets*/
10: **for all** $v_Q \in Q$ **do**
11: **for all** $bId \in I_{khb}[v_Q]$ **do**
12: *E[bId]* ←*E[bID]*+1
13: **end for**
14: **end for**
15: **for all** $i \in (0, \ldots, Size\ Of\ (E))$ **do**
16: **if** *E(i)= SizeOf(Q)* **then**
17: *F'* ←*Ø* /* Obtain a subset of points*/
18: **for all** $o \in H[i]$ **do**
19: **if** *BS[o]*= true **then**
20: *F'*← *F' U o*
21: **end if**
22: **end for**
23: **if** checkDuplicateCand(*F', HC*)=false **then**
24: searchInSubset(*F', PQ*)
25: **end if**
26: **end if**
27: **end for**
28: /* check termination condition */
29: **if** $PQ[k].r <= w_0\ 2^{s-1}$ **then**
30: Return *PQ*
31: **end if**
32: **end for**
33: /* Perform search on D If algorithm has not terminated */
34: **for all** o ∈ *D* **do**
35: **if** *BS[o]*=true **then**
36: *F'* ← *F' U o*
37: **end if**
38: **end for**
39: searchInSubset(*F',PQ*)
40: Return *PQ*

Algorithm details step in Projection and multi scale hashing.

It maintains a bitset BS. For each $v_Q \in Q$, it retrieves the list of points corresponding to v_Q from I_{kp} in step 4. For each point o in the retrieved list, algorithm marks the bit corresponding to o's identifier in BS as true in step 5. Thus, it finds all the points in D which are tagged with at least one query keyword. Next, the search continues in the HI structures, beginning at s = 0. For any given index level s, algorithm works with $H^{(s)}$and $I^{(s)}_{khb}$ in HI at step 8. ProMiSH-E retrieves all the lists of hash bucket ids corresponding to keywords in Q from the inverted index $I^{(o)}_{khb}$ at steps (10-11). An intersection of these lists yields a set of hash buckets each of which contains all the query keywords in steps (12-16) . For each selected hash bucket, Algorithm retrieves all the points. the bucket from the hashtable H, and filters these points using bitset BS to get a subset of points F' in steps (17-22).

Subset F' contains only those points which are tagged with at least one query keyword and is explored further. Subset F' is checked whether it has been explored earlier or not using check Duplicate C and in step 23. Since each point is hashed using 2^m signatures, duplicate subsets may be generated. If F' has not been explored earlier, then Pr performs a search on it using search In Subset at step 24 . Results are inserted into a priority queue PQ of size k. Each entry of PQ is a tuple containing a set of points and their diameter. PQ is initialized with k entries, each of whose set is empty and the diameter is +∞. Entries of PQ are ordered by their diameters, and entries with equal diameters are further ordered by their set sizes. A new result is inserted into PQ only if its diameter is smaller than the k-th smallest diameter in PQ. If Projection and multiscale hashing does not terminate after exploring the HI structure at index level s, then the search proceeds to HI at index level (s+1). Terminates when the k-th smallest diameter

rk has been found during steps (29-31). If Projection and multiscale hashing fails to terminate after exploring HI at all the index levels s ϵ (0,…,L-1)then it performs a search in the complete dataset D during steps (34-39).

IV. IMPLEMENTATION

MODULES DESCRIPTION

Modules Description:

- Multi-dimensional data
- Nearest Keyword
- Indexing
- Hashing

Multi-dimensional Data: Keyword-based search in text-rich multi-dimensional datasets facilitates many novel applications and tools. multi-dimensional datasets where each data point has a set of keywords. The presence of keywords in feature space allows for the development of new tools to query and explore these multi-dimensional datasets. these algorithms may take hours to terminate for a multidimensional dataset of millions of points. Therefore, there is a need for an efficient algorithm that scales with dataset dimension, and yields practical query efficiency on large datasets. multi-dimensional spaces, it is difficult for users to provide meaningful coordinates, and our work deals with another type of queries where users can only provide keywords as input.

Nearest Keyword: We consider multi-dimensional datasets where each data point has a set of keywords. The presence of keywords in feature space allows for the development of new tools to query and explore these multidimensional datasets. An NKS query is a set of user provided keywords, and the result of the query may include k sets of data points each of which contains all the query keywords and forms one of the top-k tightest cluster in the multi-dimensional space.

Indexing: Indexing time as the metrics to evaluate the index size for Projection Multi scale Hashing and Projection Multi scale Hashing Indexing time indicates the amount of time used to build Projection Multi scale variants. The memory usage and indexing time of searching under different input real data. Memory usage grows slowly when the number of dimensions in data points increases

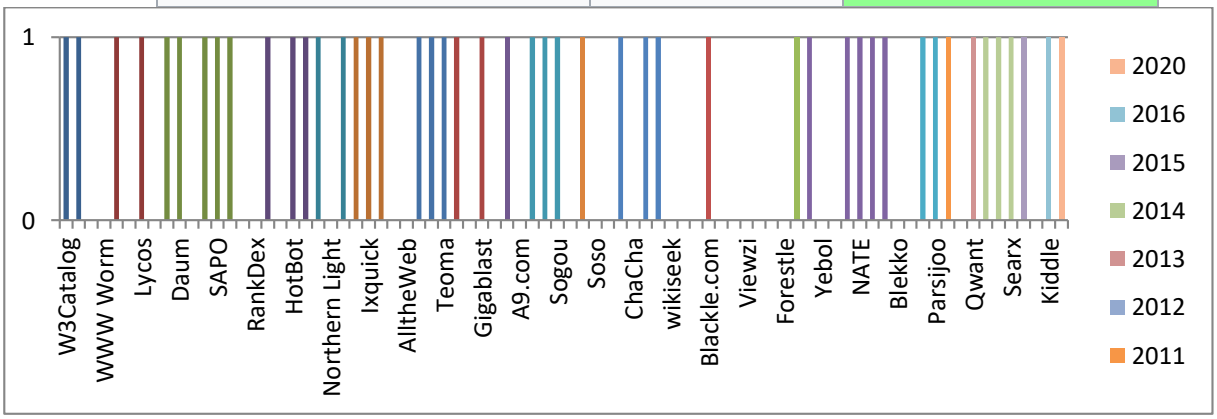
Hashing: Hashing is a technique or process of mapping keys, values into the hash table by using a hash function. It is done for faster access to elements. The efficiency of mapping depends on the efficiency of the hash function used. The hashing technique is inspired by Locality Sensitive Hashing (LSH), which is a state-of-the-art method for nearest neighbor search in high dimensional spaces. Unlike LSH-based methods that allow only approximate search with probabilistic guarantees, the index structure in Projection Multi scale Hashing supports accurate search. Random projection with hashing has come to be the state-of-the-art method for nearest neighbor search in high-dimensional datasets.

V. RESULTS AND ANALYSIS

Year	Engine	Current status
1993	W3Catalog	Active
	Aliweb	Active
	JumpStation	Inactive
	WWW Worm	Inactive
1994	WebCrawler	Active
	Go.com	Inactive, redirects to Disney
	Lycos	Active
	Infoseek	Inactive, redirects to Disney
1995	Yahoo! Search	Active, initially a search function for Yahoo! Directory
	Daum	Active
	Magellan	Inactive
	Excite	Active
	SAPO	Active
	MetaCrawler	Active

Year	Engine	Current status
	AltaVista	Inactive, acquired by Yahoo! in 2003, since 2013 redirects to Yahoo!
1996	RankDex	Inactive, incorporated into Baidu in 2000
	Dogpile	Active, Aggregator
	Inktomi	Inactive, acquired by Yahoo!
	HotBot	Active
	Ask Jeeves	Active (rebranded ask.com)
1997	*AOL NetFind*	Active (rebranded AOL Search since 1999)
	Northern Light	Inactive
	Yandex	Active
1998	Google	Active
	Ixquick	Active as Startpage.com
	MSN Search	Active as Bing
	empas	Inactive (merged with NATE)
1999	AlltheWeb	Inactive (URL redirected to Yahoo!)
	GenieKnows	Active, rebranded Yellow (redirection to justlocalbusiness.com)
	Naver	Active
	Teoma	Active (© APN, LLC)
2000	Baidu	Active
	Exalead	Inactive
	Gigablast	Active
2001	Kartoo	Inactive
2003	Info.com	Active
2004	A9.com	Inactive
	Clusty	Active (as Yippy)
	Mojeek	Active
	Sogou	Active
2005	SearchMe	Inactive
	KidzSearch	Active, Google Search
2006	Soso	Inactive, merged with Sogou
	Quaero	Inactive
	Search.com	Active
	ChaCha	Inactive
	Ask.com	Active
	Live Search	Active as Bing, rebranded MSN Search
2007	wikiseek	Inactive
	Sproose	Inactive

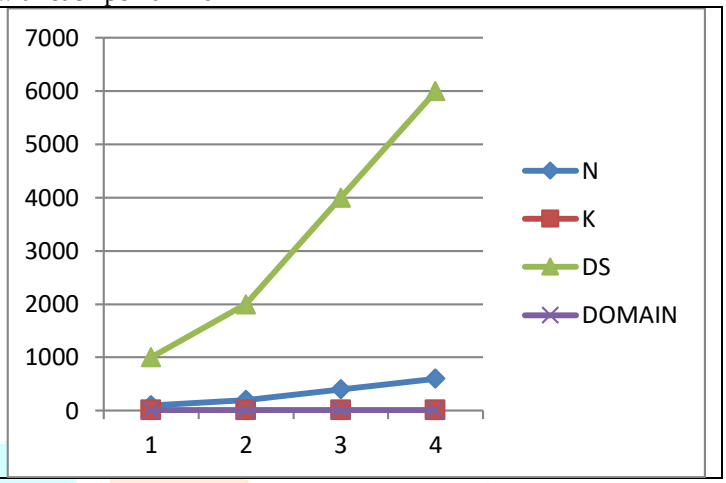
Year	Engine	Current status
	Wikia Search	Inactive
	Blackle.com	Active, Google Search
2008	Powerset	Inactive (redirects to Bing)
	Picollator	Inactive
	Viewzi	Inactive
	Boogami	Inactive
	LeapFish	Inactive
	Forestle	Inactive (redirects to Ecosia)
	DuckDuckGo	Active
2009	Bing	Active, rebranded Live Search
	Yebol	Inactive
	Mugurdy	Inactive due to a lack of funding
	Scout (Goby)	Active
	NATE	Active
	Ecosia	Active
	Startpage.com	Active, sister engine of Ixquick
2010	Blekko	Inactive, sold to IBM
	Cuil	Inactive
	Yandex (English)	Active
	Parsijoo	Active
2011	YaCy	Active, P2P
2012	Volunia	Inactive
2013	Qwant	Active
2014	Egerin	Active, Kurdish / Sorani
	Swisscows	Active
	Searx	Active
2015	Yooz	Active
	Cliqz	Inactive
2016	Kiddle	Active, Google Search
2020	Petal Search	Active



1. Datasets: Data was randomly generated. Each component was chosen uniformly from [0-6000]. Each point was randomly tagged with keywords. A dataset is characterized by its (1) size, N; (2) domain, D; (3) dictionary size, U; and (4) the number of keywords ,K associated with each point. We created datasets by varying these parameters for our empirical studies.

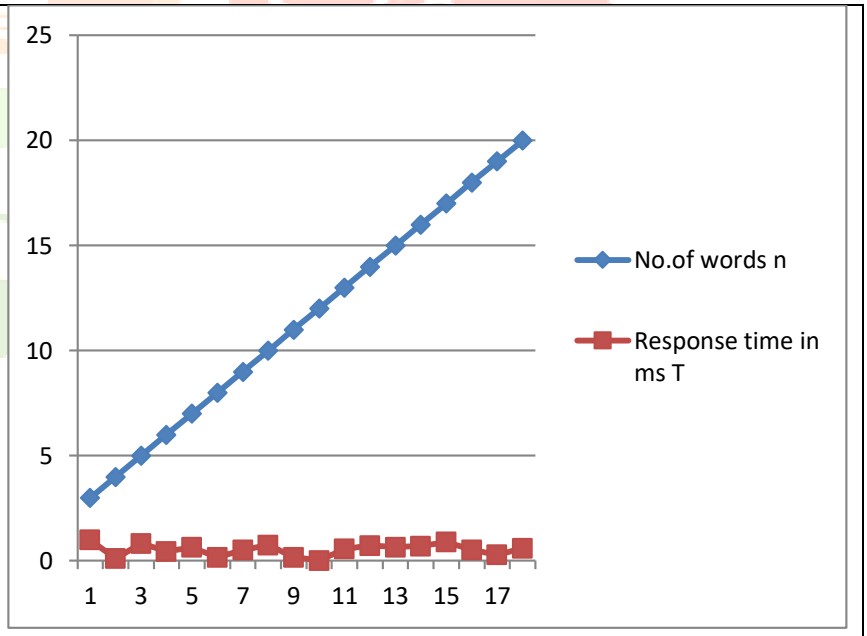
Size N = 600
Key K= 10
Domain D= 10
Dictionary Size = N x K = 600 x 10 = 6000
No of Keywords associated with each point = 10

DATA SET			
N	K	DS	DOMAIN
100	10	1000	10
200	10	2000	10
400	10	4000	10
600	10	6000	10



2. Response time : Graph represents Response time of retrieving top-1 results. By using below dimensions 'n' is number of words in keywords, 'T' is response time used for retrieving each document. Response Time measures the server response of every single transaction or query. Response time starts when a user sends a request and ends at the time that the application states that the request has completed.

No.of words	Response time in ms
n	T
3	0.9936
4	0.1064
5	0.8316
6	0.4494
7	0.6337
8	0.1742
9	0.5135
10	0.7497
11	0.1737
12	0.0149
13	0.5591
14	0.7307
15	0.6538
16	0.7035
17	0.8941
18	0.5229
19	0.2872
20	0.5983



VI. CONCLUSION AND FUTURE WORK

This topic proposes solution to the problem of top-k nearest keyword set search in multi-dimensional datasets and proposes a novel index called Projection & Multi-Scale Hashing based on random projections and hashing.

Based on this index, develop Projection & Multi-Scale Hashing that finds an optimal subset of points and utilizes less indexing time.

Ranking functions: In the future, proposed system can explore other scoring schemes for ranking the result sets. In one scheme, it may assign numbers to the keywords of a point. Then, each group of points can be scored based how many times that keyword find.

Disk extension: Proposed system may explore the extension of disk that sequentially reads only required buckets to find points containing at least one query keyword. Therefore, it can stored on disk using a directory file structure and create a directory for each bucket which stored in a separate file named after its key in the directory.

VII. ACKNOWLEDGMENT

I profoundly grateful to Prof. A.V. Mophare for his expert guidance and continuous encouragement throughout to see that this project rights its target since its commencement to its completion. I would like to express my deepest appreciation towards Principal Dr. S.D. Navale, Prof. B.R.Solunke, HOD and PG coordinator department of computer science & engineering. I must express my sincere heartfelt gratitude to all staff members of computer science & engineering department who helped me directly or indirectly during this course of work. Finally, I would like to thank my family and friends, for their precious support.

REFERENCES

[1] W. Li and C. X. Chen, "Efficient data modeling and querying system for multi-dimensional spatial data," in Proc. 16th ACMSIGSPATIAL Int. Conf. Adv. Geographic Inf. Syst., 2008, pp. 58:1–58:4

[2] D. Zhang, B. C. Ooi, and A. K. H. Tung, "Locating mapped resources in web 2.0," in Proc. IEEE 26th Int. Conf. Data Eng., 2010,pp. 521–532

[3] V. Singh, S. Venkatesha, and A. K. Singh, "Geo-clustering of images with missing geo tags," in Proc. IEEE Int. Conf. Granular Compute., 2010, pp. 420–425.

[4] V. Singh, A. Bhattacharya, and A. K. Singh, "Querying spatial patterns," in Proc. 13th Int. Conf. Extending Database Technol.: Adv.Database Technol., 2010, pp. 418–429.

[5] J. Bourgain, "On lip schitz embedding of finite metric spaces in Hilbert space," Israel J. Math., vol. 52, pp. 46–52, 1985.

[6] H. He and A. K. Singh, "Graph Rank: Statistical modeling and mining of significant sub graphs in the feature space," in Proc. 6thInt. Conf. Data Mining, 2006, pp. 885–890.

[7] X. Cao, G. Cong, C. S. Jensen, and B. C. Ooi, "Collective spatial keyword querying," in Proc. ACM SIGMOD Int. Conf. Manage. Data, 2011, pp. 373–384.

[8] C. Long, R. C.-W. Wong, K. Wang, and A. W.-C. Fu, "Collectives patial keyword queries: A distance owner-driven approach," In Proc. ACM SIGMOD Int. Conf. Manage. Data, 2013, pp. 689–700.

[9] D. Zhang, Y. M. Chee, A. Mondal, A. K. H. Tung, and M. Kitsure gawa, "Keyword search in spatial databases: Towards searching by document," in Proc. IEEE 25th Int. Conf. Data Eng., 2009,pp. 688–699.

[10] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni, "Locality sensitive hashing scheme based on p-stable distributions," In Proc. 20th Annu. Symp. Comput. Geometry, 2004, pp. 253–262.

[11] Y. Zhou, X. Xie, C. Wang, Y. Gong, and W.-Y. Ma, "Hybrid indexstructures for location-based web search," in Proc. 14th ACM Int.Conf. Inf. Knowl. Manage., 2005, pp. 155–162.

[12] R. Hariharan, B. Hore, C. Li, and S. Mehrotra, "Processing spatialkeyword(SK) queries in geographic information retrieval (GIR)systems," in Proc. 19th Int. Conf. Sci. Statistical Database Manage.,2007, p. 16.

[13] S. Vaid, C. B. Jones, H. Joho, and M. Sanderson, "Spatio-textualindexing for geographical search on the web," in Proc. 9th Int. Conf. Adv. Spatial Temporal Databases, 2005, pp. 218–235.

[14] A. Khodaei, C. Shahabi, and C. Li, "Hybrid indexing and seamlessranking of spatial and textual features of web documents," in Proc.21st Int. Conf. Database Expert Syst. Appl., 2010, pp. 450–466

[15] A. Guttman, "R-trees: A dynamic index structure for spatialsearching," in Proc. ACM SIGMOD Int. Conf. Manage. Data, 1984,pp. 47–57.

[16] I. De Felipe, V. Hristidis, and N. Rishe, "Keyword search on spatialdatabases," in Proc. IEEE 24th Int. Conf. Data Eng., 2008,pp. 656–665.

[17] G. Cong, C. S. Jensen, and D. Wu, "Efficient retrieval of the top-kmost relevant spatial web objects," Proc. VLDB Endowment, vol. 2,pp. 337–348, 2009.

[18] B. Martins, M. J. Silva, and L. Andrade, "Indexing and ranking inGeo-IR systems," in Proc. Workshop Geographic Inf., 2005, pp. 31–34.

[19] Z. Li, H. Xu, Y. Lu, and A. Qian, "Aggregate nearest keywordsearch in spatial databases," in Proc. 12th Int. Asia-Pacific WebConf., 2010, pp. 15–21.

[20] M. L. Yiu, X. Dai, N. Mamoulis, and M. Vaitis, "Top-k spatial preferencequeries," in Proc. IEEE 23rd Int. Conf. Data Eng., 2007,pp. 1076–1085.

[21] T. Xia, D. Zhang, E. Kanoulas, and Y. Du, "On computing top-tmost influential spatial sites," in Proc. 31st I nt. Conf. Very Large Databases, 2005, pp. 946–957.