THE CLASSIFICATION OF HIGH DIMENSIONAL INDICES FOR SPATIAL DATA SIMILARITY SEARCH

Yu XIA[a, *], Xinyan ZHU [b], Chang LI [a]

[a]School of Remote Sensing and Information Engineering, Wuhan University - geoxy@126.com
[b]State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University

WG IV/1

KEY WORDS: Spatial Database; Data Management; Digital Photogrammetry; Query Processing; GIS; Similarity Search; Metric Space

ABSTRACT:

The applications of spatial data similarity search are increasingly needed nowadays, and accordingly high dimensional index becomes one key technology to solve the problem of spatial data similarity search. Firstly, the distribution of high dimensional data is in-depth analyzed, and then high dimensional data retrieval for spatial data similarity search is also discussed. Secondly, based on the research, the classification of high dimensional indices for spatial data similarity search is presented, which initially makes a clear distinction of the relationship between the high dimensional index and the application of spatial data similarity search. Finally, the principle of high dimensional indices and the state of the applications in spatial data similarity search are analyzed with an example of typical index structure respectively, which lays a foundation for the research on index technology in spatial data similarity search.

1. INTRODUCTION

How to search similar objects of a given object from spatial database efficiently, called spatial data similarity search, has a wide application needs nowadays, and becomes an increasingly important problem. One of the key technologies to solve the similarity search problem is the high dimensional index. Accordingly, high dimensional index technology has been a very active research area over the past decades, and attracts many scholars' attentions. For example, (Antomn Guttman, 1984) introduced a dynamic index structure called R-tree, which was well suited to data objects of non-zero size located in multidimensional spaces; In the later years, the classic improved index structure R+-tree (Timos Sellis, 1987) and R* -tree (Norbert Beckmann, 1990) were presented respectively; The TV-tree (King-lp Lin, 1994) was designed for indexing high dimensional objects which used a variable number of dimensions to get a larger fanouts and fewer disk accesses; In (Stefan Berchtold, 1996), the X-tree was proposed which used a split algorithm minimizing overlap and additionally utilized the concept of supernodes to keep the directory as hierarchy as possible and avoid splits in the directory; (David A. White, 1996) proposed the similarity search tree, called SS-tree, which used spheres as page regions; The SR-tree (Norio Katayama, 1997) integrated bounding spheres and bounding rectangles, and used the intersection as the region, and enhanced the performance especially for high dimensional and non-uniform data; In (Roger Weber, 1998), a vector approximation schema, called VA-file, was presented to give the approximation of vectors and make the unavoidable sequence scan as fast as possible; (Christians Bohm, 2001) gave a general overview of index structures and algorithms in high dimensional space; (Christians M. Garcia-Arellano, 2002) compared the index structures among VA-File, IQ and A-tree, and then proposed a new static similarity search strategy called Quantized Clustering tree or QC-tree, which integrated the best characteristics observed in the IQ-tree and A-tree; (Ye Hangjun, 2003) presented the vector quantization (VQ) index schema, which assumed a Gaussian mixture distribution of real-world data, and trained optimized vector quantizer to partition data space; (Cui Jiangtao, 2005) proposed PCR-tree index based on R-tree and VA-File, which employed R-tree to manage the approximate vectors at principal components; In addition, many scholars paid much attention to high dimensional index in metric space, and proposed many distance based index structures where object proximity was only defined by a distance function satisfying the positivity, symmetry and triangle inequality; (Paolo Ciaccia,1997) introduced the M-tree index to organize and search large data sets, which was a dynamic balanced distance based index structure; (Tolga Bozkaya, 1999) proposed a distance based index structure called multi-vantage point (mvp) tree for similarity search, which used more than one vantage point and utilized pre-computed distances between the data points and the vantage points; (Gonzalo Navarro, 2002) presented SA-tree index which was based on approaching the searched objects spatially. (Benjamin Bustos, 2006) presented the Multi-Metric M-tree (M3-tree), which was the extension of the M-tree, stored partial distances to better estimate the weighted distance between routing entries and each search, where a single distance function was used to build the whole index. Though the researches have attributes to the high dimensional index technology, there isn't systematic research on high dimensional index classification. Furthermore, the relationship between high dimensional index structure and the application of spatial data similarity search is still not clear. Note that the deficiency limits further research on high dimensional index technology in spatial data similarity search. So the research on the classification of high dimensional index technology for feature-

* geoxy@126.com

based spatial data similarity search has the theoretical value and practical significance.

2. HIGH DIMENSIONAL DATA RETRIEVAL

Spatial data similarity search is generally based on spatial data features, such as image features, including color features, texture features, shape features, etc. , and vector features, including topological features, direction features, measure features, etc., and so on. The kind of similarity search are called feature-based spatial data similarity search. These features are usually represented as high dimensional vectors. For example, the homogenous texture descriptors recommended by MPEG-7 are represented as a vector of 62 dimensions, and the edge histogram descriptors are denoted as a vector of 150 dimensions. Therefore, one key problem of feature-based spatial data similarity search is high dimensional data retrieval efficiently.

2.1 High dimensional data distribution characteristics

To explore the nature of high dimensional data retrieval, high-dimensional data distribution is firstly discussed. In 2 dimension space, the difference between the area of the minimum boundary rectangle and that of the minimum boundary circle is relatively small, and the ratio is $(2\times\sqrt{2}/2\times r)^{2}/(\pi r^{2})=0.64$; However, in 64 dimension space, the ratio becomes $(2\times\sqrt{64}/64\times r)^{64}/(\pi^{32} r^{64}/32!)=9.54\times10^{-20}$; In 256 dimension space, the ratio evenly decrease to $(2\times\sqrt{256}/256\times r)^{256}/(\pi^{128} r^{256}/128!)=5.76\times10^{-80}$, where r denotes the radius of the hypersphere.

As figure 1 shows, with the increase of the dimension, the ratio decreases sharply, and the difference between the volume of the minimum boundary hypercube and that of the minimum boundary hypersphere becomes significant. Therefore, in the case of high dimension, the minimum boundary hypersphere doesn't reflect fittingly the distribution of high dimensional data.

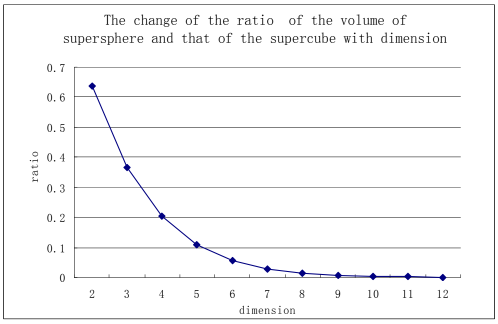


Figure 1. The change of the ratio of the volume of the hypercube and that of the hypersphere with dimension

Further, supposing 1,000,000 points distributes in a 256-dimension hypercube and each dimension is divided into 2 units, $2^{256}=1.16\times10^{77}$ units cubes are generated which exceed extremely the number of data points. For uniform datasets, there aren't points in the most of the units, and the minimum boundary hyperscube doesn't reflect the distribution of high dimensional data as well as the minimum boundary hypersphere.

From the two examples, we can see that high dimension data are very sparse in high dimensional space, and the minimum boundary hypercube and minimum boundary hypersphere don't reflect the distribution of high dimensional data well. Provided that the volume of over 75 percent of radius of the hypersphere is defined interior surface, the probability of a data point located in the interior surface in 2 dimension space is $(\pi r^{2}-\pi(3r/4)^{2})/(\pi r^{2})=0.438$, and in 6 dimension space the probability is $(\pi^{3} r^{6}/6!-\pi^{3}(3r/4)^{6}/6!)/(\pi^{3} r^{6}/6!)=0.822$. The rest may be deduced by analogy, and the change of the probability with the dimension is shown as figures 2.

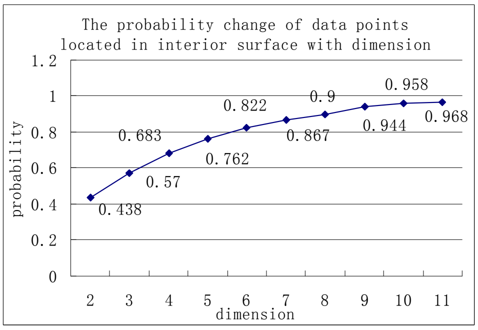


Figure 2. Probability change of data points located in interior surface with dimension in the case of the radius threshold is 75% of that of the hypersphere.

It is shown that from the figure 2, as the dimension grows, the probability of distribution in interior surface increases, and when the dimension is 8, the probability reaches 90 percent.

Note that if interior surface is defined as the volume of over different radius threshold, the probability of data distribution in interior surface is different in the case of the same dimension. In other words, the dimension is different in the case of same probability of data distribution in interior surface when the interior surface involves different radius.

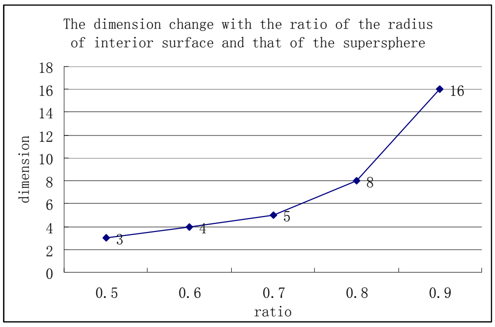


Figure 3. The dimension changes with the ratio of the radius threshold and the radius of the hypersphere when the probability of data distribution in interior surface reaches 80%

From the analysis, we conclude (1) high dimensional data point is very sparse in high dimension space, and traditional partition based indices don't reflect the distribution of high dimensional data well; (2) high dimensional data points incline to locate the interior surface, and with the increase of the dimension, the tendency is more notable. The related work includes (Yue Li, 2004), (Jinhua Li, 2001), (Tolga Bozkaya, 1997) and so on, but the researches don't involve the quantified analysis and give the curve reflecting the relationship among the dimension, the radius and the probability.

2.2 High dimension data retrieval in similarity search

Spatial data similarity search is to search the data points that are within the specific range of the given search point, or the K points that are nearest neighbours to the given search point. The two kinds of search refer to range search and K nearest neighbour search respectively. For the fact that many users don't concern about the distance threshold, similarity search often means K nearest neighbour search. In fact, K nearest neighbour search can be implemented by adjusting the threshold of range search.

From the analysis above, we can see that with the increase of dimension, the distribution of high dimension spatial data becomes very sparse, and inclines to locate in the interior surface of the hypersphere. The characteristics have a sound impact on the retrieval of high dimensional data in spatial data similarity search. (1) The sparse distribution means the poor property of clustering, which leads to the fact that traditional partition-based indices cannot reflect the data distribution in high dimension space well, so that the performance of partition-based indices decreases rapidly as the dimension grows. (2) The data distribution in high dimension space inclines to locate the interior surface of the hypersphere, ant that the ratio of the maximum and the minimum pairwise distances of a set of random points decreases and becomes close to 1 as the dimension increases (Jinhua Li, 2001), so similarity search or nearest neighbour search becomes more difficult in the case of high dimension.

In view of these, high dimensional data retrieval in spatial data similarity search is more difficult than the retrieval of traditional data. Consequently, there isn't high dimension index structure suitable for all applications of spatial data similarity search. So aiming to different applications, this paper has classified the high dimensional indices into three categories: partition-based indices, approximation based indices and distance-based indices.

3. PARTITION BASED INDICES

Partition based indices are usually tree-structure indices formed by partitioning vector space recursively, and data nodes are organized in a hierarchically structured directory. In general, the index structures can be classified into two categories: data-partitioning based indices and space-partitioning based indices (Beomseok, Nam, 2004; Gang Qian, 2006), which are also called other names, such as data dependent indices and data in dependent indices (Xiaodong Fu, 1997), data organizing indices and space organizing indices (Christian Bohm, 2001) and so on.

3.1 Data-partitioning based indices

Data-partitioning based indices partition data space according to the distribution of the data, including R-tree, R+-tree, R*-tree, SS-tree and X-tree, etc.. An R-tree is a height-balanced tree similar to a B-tree, and is built by Minimum Boundary Rectangle (MBR). In that the MBRs can be overlapped, the number of paths to be travelled is more than one. So the efficiency decreases when serious overlap appears, and efficient R-tree searching demands that both overlap and coverage should be minimized; R+-tree restricts the overlap of the MBRs to improve search performance, but the cost of building the tree unavoidably increases; R*-tree uses the forced reinsert technology to improve the performance, however the cost of building the tree is more expensive compared with R-tree; SS-tree divides the space by enclosing hypersphere, and improves the spanouts of the nodes to enhance the performance. The weak point is that the volume of the hypersphere is usually larger than the hypercube, so that the overlap is more serious; X-tree introduces the hyper-node, and decreases the number of node splits to improve the performance.

3.2 Space-partitioning based indices

Space-partitioning based indices partition the space in a specific algorithm, including kdb-tree(Bernhard Seeger,1990), hB-tree and so on. The kdb-tree is a dynamic index structure combining the kd-tree and the b-tree, and has a good performance for indexing the high dimensional point data; The hB-tree is an index structure based on the kdb-tree, and the node uses the k-dimension tree. The common characters of partition based index structures are as follows: (1) it is necessary to partition the data space whether the data distribution is depended on or not; (2) usually the index is a tree structure because of the recursive partition to the data space. The difference between the two kinds of partition based index structures is that (1) data partition based indices are usually formed from bottom to top; (2) space partition based indices structures are usually formed from top to bottom.

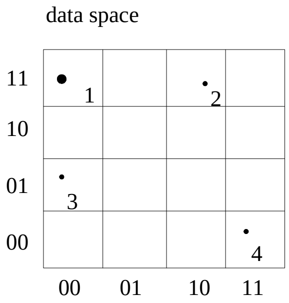
Partition based index structures locate quickly the specific data according to the coordinates, and they are efficient when the dimension is not high. However, when the dimension becomes high, the performance decreases sharply. That can be shown with following example of R-tree. As we know, R-tree performs well when indexing two dimension data. But as the dimension increases, on the one hand, the overlap of the MBRs of the directory becomes more and more serious; on the other hand, the radius of the nearest neighbour search becomes larger and larger. From the characteristics of the high dimension data, we can see the ratio of the maximum distance and the minimum distance between the any points pare decreases as the dimension increases, and becomes close to 1 when the dimension is large enough. As a result, when the dimension becomes enough large, the overlap between the MBRs becomes sharply serious, and so the radius of hyper-sphere with the search point as the center almost intersects with all the MBRs. All nodes have to be searched and the performance deteriorated rapidly, and is worse than the sequence scan. The phenomenon is called "dimension crisis". The main causation is that the partition to data space can not reflect the rule of the data distribution in space when the dimension becomes very large. However when the dimension is not very large, the kind of index structures performs well, such as R-tree performs well when the dimension is 3, and the X-tree performs well even when the dimension reaches 8. So, the kind of indices is suitable for the application of similarity search when the dimension is not high, usually from several to about 10. In fact, the kind of indices plays a very import role on spatial data similarity search.

4. APPROXIMATION BASED INDICES

From above analysis, we can see that when the dimension is very high, all the nodes in the kind of index structures will be travelled, and the performance is unavoidably poorer than the primary sequence scan, so the crisis of dimension is not avoided in the kind of index structures. Then, when the dimension is very high, can we give up the idea of partition based index, and implement the quick retrieval by reducing the cost of sequence scan?

4.1 VA-File index

Approximation based indices are implemented through this idea, and can receive good performance in the case of high dimension. The representative is the VA-File (Roger Weber, 1998). The main idea is to divide every dimension of data space into some segments, form some subspaces or cells, and code these cells through bit strings. These cells are represented approximately by corresponding bit strings or vector approximation.



vector data

1	0.1	0.9
2	0.6	0.8
3	0.1	0.4
4	0.9	0.1

approximation

1	00	11
2	10	11
3	00	01
4	11	00

Figure 4. VA-File (Roger Weber, 1998)

The creation of the VA-File is considered as the process of building the representation of the approximations. If the number of bits to allocate to each dimension is b_j, and the total number of bits in vector approximation is $\sum_{j=1}^{n} b_j = b$, the per-dimension partition position can be denoted as ($p_j[0]$, $p_j[1]$, ……, $p_j[2^{b_j}]$), and the data space is divided into 2^b hyper-rectangular cells, each of which can be represented as a unique bit-string of length b. Each data point is approximated by the bit-string of the cell into which it falls. Figure 4 illustrates this for five sample points.

Similarity search can be performed by scanning the entire approximation file, and by excluding the vast majority of vectors from the search based only on these approximations (Roger Weber, 1998).

4.2 Improved VA-File index

In addition, the kind of indices includes still the improved VA-File indices. Note that VA-File index all the same uses the strategy of two levels “filtering-refine” to complete the high dimensional retrieval. Consequently, the improvement of VA-File can be done from two aspects.

In view of this, we classify the improved VA-File into two categories according to the strategy. (1) From the perspective of the organization method of approximate vector, the first category reorganizes approximate vectors according to some specific rule and makes the cost to scan the approximate vector decrease. The primary filtering performance of the index improves for it is easier to find the approximate vector. The kind of indices involves sorting approximate vectors based on principle component (Cui Jiangtao, 2005) and using R-tree to index approximate vector (Cui Jiangtao, 2005) and so on; (2) From the point of view of approximation, the other category try to achieve a more accurate approximate vector and to improve the secondary retrieval performance. The kind of indices include VA+-File which uses the KL transformation to eliminate the correlation of every components, and allocate different number of bits to every dimension according to the different energy of every dimension after KL transformation (Hakan Ferhatosmanoglu, 2000); and IQ index which introduces minimum bounding rectangles to VA-File, and obtains the data point density in MBRs, and then allocates different number of bits to the dimensions of different point density (Stefan Berchtold , 2000); and the local polar coordinate file (LPC-File) which approximates vector by polar coordinates on the partitioned local cells and enhances significantly discriminatory power of the approximation (Guang-Ho Cha, 2000). From the point of view of the strategy of bits distribution and organization of approximate vector respectively, these methods improve the primary filtering and secondary retrieval efficiency and can achieve better performance that VA-File.

From the analysis, we can find the kind of indices complete the quick retrieval by making the cost of sequence scan decrease. For the storage space of approximation vector is smaller than the primary vector, and then the I/O times needed to access decreases sharply. The kind of indices avoid successfully the “dimension crisis” phenomenon, and is suitable for the application of similarity search where the dimension is very large, usually from over dozens dimension to more than one hundred dimension.

5. DISTANCE BASED INDICES

Spatial object in spatial data similarity search is usually represented as high dimension feature vector, and one of important trait is that distance calculation cost or CPU cost is expensive. Partition based indices divide the data space according to coordinate information without consideration of distance cost, and the performance cannot be guaranteed in the case of high dimension. And then, is it available to decrease the cost of distance calculation by reducing the number of feature vectors which are necessary to similarity match calculation, and accordingly complete the efficient retrieval of high dimension data?

Distance-based indices utilize the distance properties or measure properties in metric space to index high dimension data. The formal description of metric space is can be given as follows: The set X denotes the universe of valid objects, and U is a subset of it, a given set of data objects where we search. There is a function d : X × X $\longrightarrow$ R, which satisfies following three axioms.

(1) Positiveness,

$\forall x, y \in X$, $d(x, y) \geq 0$, when and only when $x = y$, $d(x, y) = 0$; when $x \neq y$, $d(x, y) > 0$

(2) Symmetry,

$$\forall x, y \in X \ , d(x, y) = d(y, x) ,$$

(3) Triangular inequality,

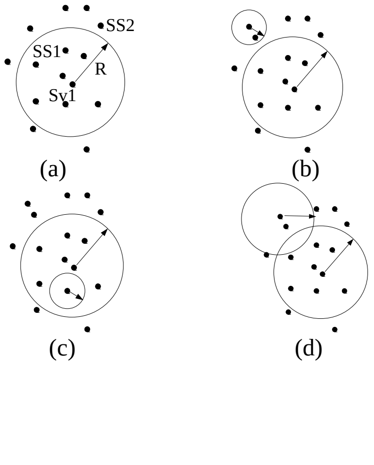
$$\forall x, y, z \in X \ , \ d(x, y) \leq d(x, z) + d(y, z) .$$

Then the pair (X, d) is called metric space.

From different points of view, distance based indices can be divided into different types (Dong Daoguo, 2005). (1) From the perspective of data organization, distance based indices can be divided into static indices and dynamic indices, where the static indices including VP-tree (Gisli R Hjaltason, 2003) and SA-tree (Gonzalo Navarro, 2002), etc., don't support the dynamic operation of insert, delete and so on, and the dynamic indices are the index structures, whose creation is the process of inserting every data point orderly, and usually from bottom to top, including M-tree (Paolo Ciaccia, 1997), Slim-tree (C. Train Jr., 2000) and so on; (2) From the angle of distance measure, distance based indices can be divided into discrete distance indices and continuous distance indices. The former means the distance function is discrete, including BK-tree (Edgar Chavez, 2001) and FQ-tree (Edgar Chavez, 2001). The latter refers to the indices whose distance function is continuous, including VP-tree (Gisli R Hjaltason, 2003) and M-tree (Paolo Ciaccia, 1997); (3) From the point of view of index structure, distance based indices can be divided into tree type indices and non-tree type indices. Just as its names implies, tree type indices are organized as tree structures including BK-tree, VP-tree, BS-tree (Edgar Chavez, 2001). Non-tree type indices include FQA, AESA (Edgar Chavez, 2001) and so on.

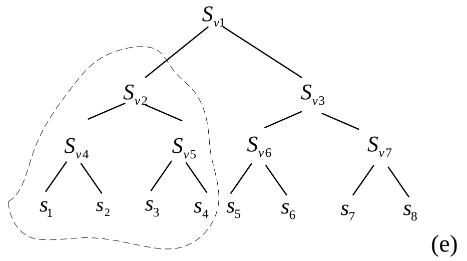
With an example of classic index structure VP-tree (Gisli R Hjaltason, 2003), the principal of distance based index structures is analyzed. The main idea of VP-tree is that data space is partitioned into two subspaces in term of the distance from vantage point, as shown in figure 5(a). The algorithm recursively repeats the partitioning process and forms a hierarchical binary tree. As figure 5(e) shows, the tree is generated from data space SS where left branch is built from SS1 and right branch is generated from SS2. Each internal node of the binary vp-tree is of the form (S_v, M, R_{ptr},L_{ptr}), where Sv is the vantage point, M is the median distance among the distances of all the points from S_v indexed below that node, and R_{ptr} and L_{ptr} are points to the left and right branches (Tolga Bozkaya, 2000).

For a given search point q, if $d(q,Sv1)$ - r $\geq R$, as figure 5 (b) shows, where $d(q,Sv1)$ can be calculated in real time, and r is from the search condition, and R is the median distance M of the vantage point Sv1, then left branch can be pruned which are marked with dashed in figure 5(e); if $d(q,Sv1)+r$ $\leq R$, as figure 5 (c) shows, the right branch can be pruned, and so the distance from the search point to the vantage point is exempted from calculation; if the two conditions of $d(q,Sv1)$ - r $\leq$ R and $d(q,Sv1)$ + r $\geq R$ are satisfied at the same time, as figure 5 (d) shows, then both branches cannot be pruned and further travel in both branches is needed.



(a) (b)

(c) (d)



(e)

Figure 5. The idea of distance-based indices

From the analysis, we can see that data filtering in VP-tree is completed by triangular properties in metric space. In fact, the triangle inequality property is applied for pruning the search space in all distance based indices. Note that the cost of distance calculation is related with not only the definition of similarity measure, but also the number of dimension. Usually the distance calculation cost grows with the number of dimensions. For example, it is needed to calculate the distance of two feature vectors with n dimensions. When we take Euclid distance as the similarity measure method without consideration of the evolution operation, n-1 times addition and n times multiplication are needed. If the number of feature vector pairs is m, then n^m times multiplication operation is needed. For high dimension data retrieval, n reaches hundreds, even thousands, and m depends on the dataset and the specific index structure. Therefore, with the increase of the dimension, the distance calculation becomes significantly more expensive. Distance based indices are suitable for the application of the spatial data similarity search with expensive distance calculation, and usually including the case of high dimension. It is true that they have been used in many applications of similarity search, and are a prosperous research area.

6. CONCLUSION

High dimensional data index is one key technology to solve the problem of spatial data similarity search. In this paper, the quantified analysis of the distribution of high dimensional data and the curve reflecting the relationship among the dimension, the radius and the probability are given, which show the sparsity and distribution tendency of high dimensional data. Because of the properties in high dimensional space, traditional partition based index can not reflect data distribution well and can also not avoid "dimension crisis", which leads to poor performance. For the fact that the case of high dimension does often appear in spatial data similarity, aiming to the application of spatial data similarity search, we present the classification of high dimensional indices for spatial data similarity search: partition based indices, approximation based indices and distance based indices. Partition based indices locate the specific data according to the coordinates, which are suitable for the application of spatial data similarity search in the case of low dimension, such as less than 10 dimension; Approximation based indices implement the retrieval by reducing the cost of sequence scan, which are suitable for the high dimension similarity search such as over ten dimension to more than 100 dimension; Distance based indices employ triangle inequity to complete the retrieval which are suitable for the similarity search of expensive distance calculation, usually including the case of high dimension, such as dozens to hundreds of dimension.

REFERENCES

Antomn Guttman, 1984. R-trees: a dynamic index structure for spatial searching. *Proc. ACM SIGMOD Int. Conf. on Management of Data*, Boston,MA, pp.47-57

Benjamin Bustos, 2006. Dynamic Similarity Search in Multi-Metric Spaces. *MIR'06*. Santa Barbara, California, USA. pp. 137-146.

Beomseok Nam, 2004. A Comparative Study of Spatial Indexing Techniques for Multidimensional Scientific Datasets. *Proc. of the 16th Int. Conf. on Scientific and Statistical Database Management (SSDBM'04)*. pp.171-180

Christian Bohm, 2001. Searching in high-dimensional spaces: Index structures for improving the performance of multimedia databases. *ACM Computing Surveys*, 33(3), pp. 322-373

Christian M.Garcia-Arellano, 2002. Quantization Techniques for Similarity Search in High-Dimensional Data Spaces. *Ph.D thesis*. University of Toronto, Toronto, Japan.

C. Train Jr.,2000. Slim-Trees: High Performance Metric Trees Minimizing Overlap Between Nodes. *International Conference on Extending Database Technologh(EDBT) 2000*. Konstanz, Germany, pp. 51-65.

Cui Jiangtao, 2005. Research on Vector Approximation Method in High-dimensional Index Technology. *Ph.D thesis*. Xidian University, Xi'an, China.

David A. White, 1996. Similarity Indexing with the SS-tree. *Proc. 12th Int. Conf. on Data Engineering*, New Orleans, LA. pp. 516-523.

Dong Daoguo, 2005. Research on High-dimensional data index structures. *Ph.D thesis*. Fudan University, Shanghai, China.

Edgar Chavez, 2001. Searching in Metric Spaces. *ACM Computing Surveys*. 33(3), pp. 273-321.

Gisli R Hjaltason, 2003. Index-Driven Similarity Search in Metric Spaces. *ACM Transactions on Database Systems*. 28(4), 517-580.

Gonzalo Navarro, 2002. Searching in metric spaces by spatial approximation.*The VLDB Journal* 11, pp. 28-46

Gang Qian, 2006. Dynamic Indexing for Multidimensional Non-ordered Discrete Data Spaces Using a Data-Partitioning Approach. *ACM Transactions on Database Systems*, 32(2), pp.439-484

Guang-Ho Cha, 2002. An efficient indexing method for nearest neighbor searches in high-dimensional image databases. *IEEE Transactions on multimedia*, 4(1), pp. 76-87

Hakan Ferhatosmanoglu, 2000. Vector Approximation based Indexing for Non-uniform High Dimensional Data Sets. *ACM Int. Conf. on Information and Know-ledge Management*. New York: ACM Press. New York, NY USA, pp. 202-209

Jinhua Li, 2001. Efficient similarity search based on data distribution properties in high dimensions. *Ph.D thesis*. Michigan: Michigan State University.

King-lp Lin, 1994. The TV-tree: An Index Structure for High-Dimensional Data. *VLDB Journal*, 3, pp. 517-542.

Norio Katayama, 1997. The SR-tree: An Index Structure For High-Dimensional Nearest Neighbor Queries. Proc. *ACM SIGMOD Int. Conf. on Management of Data*, pp. 369-280.

Norbert Beckmann, 1990. The R*-tree: An Efficient and Robust Access Method fro Points and Rectangles. *Proc. ACM SIGMOD Int. Conf. on Management of Data*, Atlantic City,NJ, pp.322-331.

Paolo Ciaccia, 1997. M-Tree: An Efficient Access Method for Similarity Search in Metric Spaces. *Proceedings of the 23rd International Conference on Very Large Databases(VLDB'97)*. San Francisco, USA, pp. 426-435.

Roger Weber, 1998. A quantitative analysis and performance study for similarity search in high dimensional spaces. *Proceedings of the 24th International Conference on Very Large Data Bases (VLDB'98)*. NewYork,USA, pp. 194-205.

Stefan Berchtold, 1996. The X-tree: An Index Structure For High-Dimensional Data. *VLDB'96*. Mumbai, India. pp. 28-39.

Stefan Berchtold, 2000. H V Jagadish.Independent Quantization:An Index Compression Technique for High-Dimensional Data Spaces.San Diego: *Proc. of the 16th Int.Conf. on Data Engineering (ICDE'00)*. Ca California: IEEE Computer Science Society Press. California, USA pp.577-588.

Timos Sellis, 1987. The R+-tree: a dynamic index for multi-dimensional objects. *VLDB'87*. Brighton, England, pp.507-518.

Tolga Bozkaya, 1997. Distance Based Indexing for High Dimensional Metric Spaces. *ACM SIGMOD Record*. 26(2), pp. 357-368

Tolga Bozkaya, 1999. Indexing Large Metric Spaces for Similarity Search Queries. *ACM Transactions on Database Systems*. 24(3), pp. 361-404.

Xiaodong Fu, 1997. GPR-tree: A Global Parallel Index Structure for Multiattribute Declustering on Cluster of Workstations. *Proc. of the 1997 Advances in Parallel and Distributed Computing Conference (APDC'97)*. pp. 300-306

Ye Hangjun, 2003. Research on Indexing and Retrieval Techniques in Large-Scale Image Database. *Ph.D thesis*. Beijing: Tsinghua University.

Yue Li, 2004. Efficient similarity in high-dimensional data spaces. *Ph.D thesis*. New Jersey: New Jersey Institute of Technology.

ACKNOWLEDGEMENTS

This work was supported by a grant from the National High Technology Research and Development Program of China (863 Program) (No.2007AA12Z201) and the National Key Technologies Research and Development Program of China (No.2006BAB10B03).