MULTIDIMENSIONAL INDEXING METHODS FOR STRUCTURING THE SPACE DESCRIPTION OF STILL IMAGES

Diana Sophia Codaț

Ph.D. Student, Politehnica University of Timişoara, Romania

Corresponding author: Diana Codaț, diana.codat@gmail.com

ABSTRACT: This article presents a state of the art on the main multidimensional indexing methods for structuring the space description of still images. We present some general principle, the strengths and weaknesses of these techniques.
There are two main families of multidimensional indexing methods: conventional indexing methods and indexing methods based on filtering also called approximation approach. In this article we present only conventional indexing methods.
KEYWORDS: Data Mining, multidimensional indexing, M-Tree, K-Tree.

1. INTRODUCTION

The emergence of digital technologies in the field of multimedia highlighted the importance of the problems of multidimensional indexing and search by content in large databases. Conventional methods of multidimensional indexing have been proposed to organize the descriptors or as the feature vectors from digital images to prevent exhaustive sequential scan of large databases allowing fast access and query when searching by content. Generally the descriptors of the images are represented by multi-dimensional vectors in a large space. Each component of vectors corresponds to a different size of the data space and represents a specific attribute of a descriptor. Then the data belonging to a multidimensional space in which each dimension represents a different axis of the data space. The representation format is often used by multidimensional indexing methods thanks to its flexibility and simplicity to represent many different types of information. In this context, the major problem of multidimensional indexing methods is how to effectively index a large multidimensional data collection to respond quickly and effectively to user requests.

2. MULTIDIMENSIONAL INDEXING METHODS

Multidimensional indexing methods must meet a number of requirements to regarding the following aspects:

- Problem of dimension effectively address: problems related to the curse of dimensionality.
- Scalability: allow large-scale deployment of the index structure.
- Performance Research: ensure good search precision in a reasonable and acceptable time by the user.
- Dynamic: Supporter insertions and deletion of data without affecting-cantly if the organization of the index structure.
- Adaptability: Take into account the spatial distribution of data, and consider when indexing and structuring process.

The answer to all these conditions is a big challenge for multidimensional indexing methods. In the next section we review the multidimensional index to organize still images. We present their basic principles and their strengths and weaknesses regarding the performance in large dimension.

2.1 Conventional techniques multidimensional indexing

Conventional techniques multidimensional indexing based on the principle of grouping based packet vectors, then encompass in simple geometric shapes to handle. The idea is to reduce the course when looking at a subset of packages by selecting the most relevant and not access that fewer vectors. The first research on the multimedia indexing algorithms in a multidimensional space has been made between 1979 and 1984. A first classification of multidimensional indexing methods was performed by Gaede Indeed, Gaede classified these methods according to the types of data they support. There are two families of methods:

- Point of Access Methods (Access Point Methods - WFP) where data is represented by points (vectors).
- Spatial Access Methods (Spatial Access Methods - SAM) where the data is represented by complex space objects (segment, right, polygons, etc.).

A second classification of conventional indexing methods was subsequently established based on the

partitioning and data organization in the space of multidimensional vectors. Under such a test, two large families of methods can be considered:

o Methods based on partitioning data which summarize the data according to their proximity in space. They are all derived from the R-Tree method [Gut84] where each vector group is included in a particular geometric shape (hypersphere, hyper-rectangle etc.). Everything is structured in a balanced tree.

o Methods of space partitioning. These methods are all derived from KD-Tree and Quad-Tree, they cut the multidimensional space into disjoint regions and then store the data in this division.

Data partitioning methods.

The multidimensional index based on data partitioning allows for the distribution of vectors in multidimensional space. These methods are all derived from the R-tree [Gut84] or the data is grouped into simple geometric shapes (rectangles hyper, hyper-spheres, etc. according to their proximity in space. These geometric shapes derived from the data partitioning are organized as a tree in which the vectors are stored in the leaves and geometric shapes are stored in the internal nodes of the tree. The best known methods in the literature are detailed below.

Family R-Tree.

Trees R-Tree Family [Gut84] index a multi-dimensional space of points with a balanced hierarchical division into hyper-rectangles. The R-Tree is a balanced tree in which each node is associated with a minimum Rectan-encompassing rule (REM), the latter is the REM of all rectangles of his son. Tree leaves contain a list of type inputs (REM, oid) where REM is the minimum bounding rectangle of the object identified by its oid. The size of the nodes and leaves is limited and fixed a priori. It corresponds to the size of a disk page. Figure 2.1 illustrates a simplified example in which we represented the data by roundabouts and regions by rectangles.

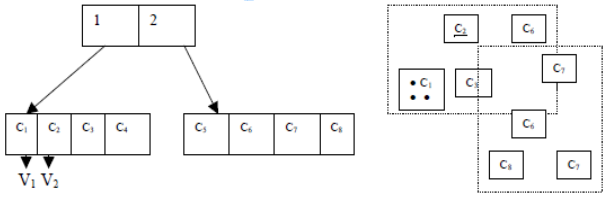

Fig. 2.1: Structure of R-Tree

*The R * -Tree.*

The R * -tree is a variant of R-Tree. To improve the performance of R-Tree, Beckmann et al. [B+90] proposed the reinsertion of data stored in R-Tree. Instead of cutting an overloaded node, the additional entries are deleted and re-insert into the tree at the same level. In many cases, this avoids the proposed subdivision, and therefore increases the rate of exploitation of the allocated memory space. The insertion of a new vector in the tree R * -tree is based on two criteria: minimize the volumes of REMs during an insert into a knot and minimize duplication rate when inserting a sheet. Partitioning of REMs is based on a criteria which takes into account the minimization of the overlap rate and the scope of hyper rectangle.

M-Tree.

M-Tree [CPZ97] is a tree structure based on the data partitioning. Data vectors are grouped in a hierarchical manner from their proximity in the data space and based on a distance metric. This is one of the first indexing methods aimed at reducing, in addition to the number of input / output, the CPU cost calculations of distances. Based on the distances recalculated and the triangle inequality, this technique avoids some calculations unnecessary distances when searching by eliminating sub irrelevant to the query trees. In an M-Tree, each node of the tree consists of a database object and object called redirection called a radius of coverage radius. M-Tree is constructed by successive insertions of objects (the basis vectors). The criterion for selection of the insertion path is to minimize coverage rays redirection objects. Partitioning saturated nodes selects two objects of the node and then distributes the remaining items by minimizing the volume of two regions obtained and their rate of overlap.

M-Tree suffers from overlap of the problem broadens the number of paths to go for a search, overlap also increases the number of distance calculations to respond to a user request which greatly reduces performance of the method. The Slim shaft [Tra00] is a method that has been proposed to improve the M-Tree reducing overlap between the REMs.

Slim-Tree.

The Slim [T+00] tree is a dynamic and balanced tree which groups the data into fixed size disk pages where each page corresponds to a node of the tree, and objects (vectors) are stored in the leaves of tree. It is based on a simple technique to quantify the degree of overlap between the nodes of the tree. It is well known that the degree of overlap directly affects the performance of multidimensional index, thus the Slim tree is one of the first methods that has been specifically designed to reduce duplication rates. This technique organizes objects in a hierarchical structure through a representative data which is the center of the smallest region that covers the objects in a subtree. Like the M-Tree method, distance from the

representative and the triangle inequality are used to eliminate the irrelevant search sub trees.
The construction of the index structure of the slim shaft by successive insertions of the different vectors of the database. From the root node, the insertion algorithm looks for the tree node corresponding to a region of space that can contain the vector to insert. If no node is found, the node whose center is closest to the vector is chosen. If the insertion algorithm is more than a node which may contain the vector, the insertion algorithm chooses the node is in a random manner, or it selects the node which has a minimum distance between the vector and the central node, or the node that owns the minimum vectors. Overlapping is a major problem affecting the majority of indexing techniques, and it is generally difficult to quantify due to the impossibility of calculating the volume of the intersection regions. To solve this problem, the slim shaft overlap rate estimated by the relative number of objects covered by two (or more) regions and then applies an algorithm called "slim down" to decrease the rate of overlap between regions. This algorithm consists of three steps. In the first step, the algorithm calculates the vector further of its representative vector (central node) to each node. In the second, the algorithm identifies the nodes that span the vector, moves it to the nearest node and corrects the radius of the resulting node.
Figure 2.2 illustrates the operation of the algorithm "slim down":

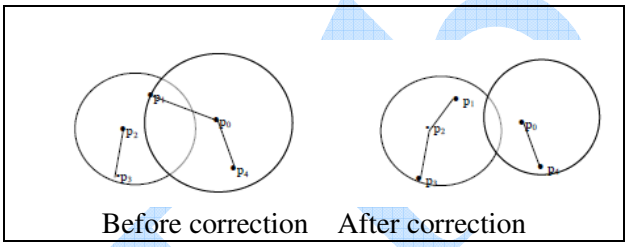

Fig. 2.2: The operation of the algorithm "slim down"

Note that the algorithm "Slim-Down" well reduces duplication rate in comparison with M-Tree, for against it produces trivial knots vectors containing little or empty nodes which greatly reduces the performance of the index.

PM-Tree.
PM-Tree (Pivot Metric Tree) is a method of multidimensional indexing approach that combines the pivot and M-Tree method in the objective of reducing the volume of the region defining the data vectors. Such a reduction increases the rate of discharges under trees irrelevant and subsequently increases the efficiency of the index. The index structure of the PM-Tree is constructed in the same manner as M-Tree, for against it introduces elements of the database called Pivot to identify smaller regions containing the vectors.
Note that the difference between PM and M-Tree Tree which lies the fact that PM-Tree uses regions "spherering" instead hyper spheres to delimit areas that contain sub group of objects stored in the leaves of tree, this reduces to a considerable extent the volume of regions and increases the performance of the index.

MH-Tree.
MH-Tree is an index proposed by Guoren et al. [G+07] based on data partitioning by hyper planes. MH tree is dynamic and balanced, it is designed to support the dynamic data and requires no periodic reorganizations of the structure, it is based on using a partitioning hyper plane instead of a key dimension for the data partitioning and filtering thereof. Note that the tree is only suitable for MH Euclidean spaces due to the characteristics of the hyper plane used for partitioning. The idea of using a partitioning hyper plane to split the data and not just one dimension of that done in most applications, there are several dimensions of data that can contain a large amount of information.

2.2. Partitioning method of the space

Unlike indexing methods based on partitioning data whose main disadvantage is the overlap between the geometric shapes chosen for grouping data, methods of partitionnent space partitionnent data space into geometric shapes (hyper plaine, hyper-rectangle etc.) disjoint. Several techniques have been proposed as Pyra-amide-Tree [BBK98] iMinMax [O+00], P + -tree [ZOT04] VITRI [SOZ05], Kpyr [UBD05] etc.

Pyramide.
The pyramid technique [BBK98] is one of the first methods proposed are not suffering the problems of the curse of dimensionality, it divides the space into pyramids, and then assigns the data a number pyramid and its height up at the top of the pyramid. Each pyramid has a base having a surface of 2 × dimensions, it is sliced parallel to its base. The slices near the top are smaller than those that are close to the base. This division of space has the property to create a number of cells that increases linearly with the size.
The Pyramid technique performance degrades slowly when the data size increases. However, this strongly depends on the distribution data and the position of the query in the space. Indeed, when data distribution is not uniform, choosing the top of the pyramids is not significant, which deteriorates the quality of indexing and the speed of the search. Similarly, when a query is close to the base of a pyramid can generate

unnecessary access to a set of data that has not necessarily similarity to the query.
This can significantly affect search performance. On the other hand, the performance of the pyramid depends on the distribution data. Indeed, if the data are grouped in a corner of the space, all requests should be in the same corner as the data distribution often follows those queries data, it is for this reason that the method of Pyramid is less efficient than a sequential for highly aggregated data.

IMinMax.
Just like the Pyramid-Tree technique, iMinMax [O+00] decomposes the multidimensional space in 2 xd pyramids. The difference between the two methods lies in the strategy with which the data is represented. Indeed, every vector in multidimensional iMinMax is represented by a key consisting of his closest surface coordinate on the dimension d-1 and its magnitude. This method uses a single transformation to project multidimensional vectors in a one-dimensional space (key vectors).

P+-Tree.
P + -tree [ZOT04] is an improvement in the Pyramid Technique. This approach combines a method of division of space-based Bissecting K-means and the Pyramid technique. Indeed, P + -tree divide the space into hyper rectangles, then applies the pyramid technique to each of these subspaces. Each vertex of the pyramid represents the center of a data group. If the data are around the top, and despite the existence of some queries on the edge of the pyramid produces naturally in the process of seeking access to a wide area. The data covered are not many that most of them are located around the top of the pyramid.

ViTri (Video Triplet).
Shen et al. [SOZ05] proposed a new video indexing method VITRI form of B + tree. A video sequence is first divided into a number of groups containing "frames" like. Each group is then modeled by an hypersphere in a d-dimensional space which also represents the dimension of the multidimensional vector on each "frame." Each hypersphere is represented by the triplet (position, radius, density) who respectively represent the position of the group center, radius and the number of "frames" in the group. The grouping of the radius (where hyper-sphere) is calculated by the average of standard deviations between different "frames" and the grouping of center. The similarity between two groups was assessed by estimating the number of "frames" similar between the two groups, this is calculated by the intersection between the two hyper-spheres multiplied by the smallest density. Thus, the local information in each group is identified in an efficient manner. The exhaustive search at the VITRI structure for a large amount of data is very costly. For this, Shen et al. propose to transform the multidimensional data in a one-dimensional space, and then to apply a principal component analysis (PCA) on the one-dimensional data to be designated by the following optimum reference point to which the original distance between the multidimensional vectors is maximal after processing. The resulting data is then indexed by the B + tree.

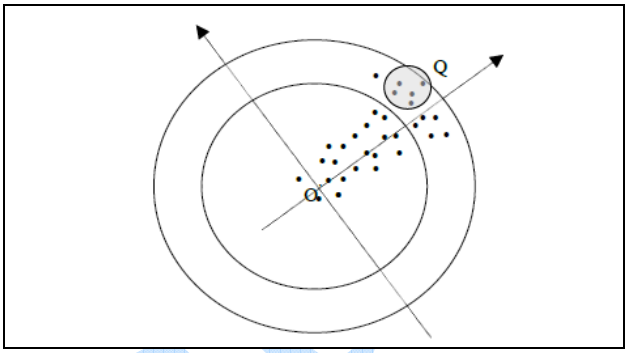

Fig. 2.3: Geometric representation of the indexing method according to ViTri

Kpyr.
Urruty et al. [UBD05] proposed an indexing method that combines classification method to multidimensional indexing technique. Kpyr proceeds first by a classification step through the K-Means algorithm wherein the data is partitioned into a number of homogeneous groups. Then, each group is made a base change in the corresponding space in a hyper unit cube to which is applied the pyramid technique. Data for each group are indexed by a B + tree.
Kpyr structure is a binary tree structure called "shaft space" as the tree nodes store the boundaries calculated from the inter-group distances while the leaves of the tree are represented by B + tree groups.

3. CONCLUSION

Conventional methods of multidimensional indexing based on data partitioning are based on the use of geometric shapes that allow encompassing refine filtering regions that can contain all the results. Unfortunately these methods suffer from the problems of the curse of dimensionality; their performance degrades when the size increases. The main disadvantage of these methods is the overlap between the geometric shapes encompassing vectors. Indeed, the overlap makes filtering rules unable to eliminate irrelevant geometric shapes. Therefore, a large number of regions (rectangles, circles etc.) are visited during the research leading to an increase in response time. For that the main objective of most methods based on partitionnent is to minimize the

overlap rate between the geometric shapes inclusive. Unlike conventional indexing methods based on the data partitioning, the main drawback is the collection of geometric shapes inclusive, methods partitionnent of the data space partitionnent data space in geometric shapes (hypersphere, hyper-rectangle etc.) disjointed without any overlap. Partitioning is done without considering the distribution of data, which allows the generation of regular geometric shapes simple to manage. Most of these methods do not support the non-uniform distribution of actual data or highly aggregated. Like the conventional methods based on data partitioning, these methods suffer from the problems of the curse of dimensionality, their performance is significantly degrade when the size increases.

REFERENCES

[Bed88] **M. A. Bednarczyk** - *Categories of asynchronous systems*, PhD thesis, University of Sussex, 1988.

[BBK98] **S. Berchtold, C. Bohm, H. P. Kriegel -** *The pyramid technique: Towards breaking the curse of dimensionality*, in Proceedings of ACM SIGMOD Int. Conf on Management of Data, pages 142–153, Seatle, Washington, USA, 1998.

[B+90] **N. Beckmann, H. P. Kriegel, R. Schneider, B. Seeger -** *The r*-tree: An efficient and robust access method for points and rectangles,* in Proceedings of ACM SIGMOD International conference on Management of Data, pages 322–331, Atlantic City, NJ, USA, 1990. ACM Press.

[CPZ97] **P. Ciaccia, M. Patella, P. Zezula -** *M-tree: an efficient access method for similaritysearch in metric spaces,* in Proceedings of the 23rd International Conference on VeryLarge Data Bases, VLDB'97, Morgan Kaufmann, Greece, pp. 426–439, 1997.

[Gut84] **A. Guttman -** *R-trees: A dynamic index structure for spatial searching,* in Proc. of the ACM SIGMOD Int. Conf. on Management of Data, pages 47-57, Boston, MA, June 1984.

[G+07] **W. Guoren X. Zhou, W. Bin, B. Qiao, D. Han -** *A hyperplane based indexing technique for high dimensional data*, in Proceeding of Information Sciences 177 (2007) 2255-2268.

[O+00] **B. C. Ooi., K. L. Tan, C. Yu, S. Bressan -** *Indexing the Edges - A Simple and Yet Efficient Approach to High-Dimensional*, in 19th ACM SIGMOD SIGACT SIGART Symposiumon Principles of Database Systems, Dallas, USA, May,p.166-174, 2000.

[SNW94] **V. Sassone, M. Nielsen, G. Winskel** - *Relationships between models of concurrency*, in Proceedings of the REX 3 school and symposium, 1994.

[SOZ05] **H. T. Shen, B. C. Ooi, X. Zhou -** *Towards effective indexing for very large video sequence database,* in Proceedings of the ACM SIGMOD International Conference on Management of Data, pages 730–741, Baltimore, Maryland, USA, 2005.

[S+94] **V. Sassone, M. Nielsen, G. Winskel, J. Doe** - *Relationships between models of concurrency*, in Proceedings of IEEE, 1994.

[T+00] **C. Traina Jr., A. Traina, B. Seeger, C. Faloutsos** - *Slim-trees: high performance metric trees minimizing overlap between nodes*, in: C. Zaniolo, P.C. Lockemann, M.H. Scholl, T. Grust (Eds.), Proceedings of the 7th International Conference on Extending Database Technology, EDBT'00, Springer, Konstanz, Germany, 2000, pp. 51–65.

[UBD05] **T. Urruty, F. Belkouch, C. Djeraba -** *KPYR: An Efficient Indexing Method*, in Multimedia and Expo, 2005. ICME 2005. IEEE International Conference on volume, Issue, 6-6 July 2005 Page(s): 1448 – 1451.

[ZOT04] **R. Zhang, B. C. Ooi, K. L. Tan -** *Making the pyramid technique robust to query types and workloads*, in Proceedings of 20-th Int. Conf. on Data Engineering, IEEE ICDE, pages 313–324, Boston, USA, 2004.