A REVIEW PAPER ON MULTIDIMENTIONAL DATA STRUCTURES

Kujani. T*, Dhanalakshmi. T+, Pradha. P#
Asst. Professor, Department of Computer Science and Engineering,
SKR Engineering College,
Chennai, TamilNadu, India
kujanit@gmail.com*, dhanavt@gmail.com+, hanupradha@yahoo.com#

Abstract
Multimedia Data Structures is needful for storing multimedia data such as Text, Images, Audio, Video, and Graphical Objects. The growth of digital content gave rise to the revolution of very large multimedia Data Structures, in which the need of efficient storage, organization and retrieval of multimedia contents came into a big issue. This paper presents a review of two such multimedia Data Structures with its applications. The main motive is to provide a perspective on Spatial Multimedia Data Structures such as k-d tree and Quad tree for academicians.

Keywords: Multimedia Data, information retrieval, spatial databases, kd tree, quad tree.

1. INTRODUCTION

Data Structures are used to organize data. They are used for storing information as well for searching, retrieving and filtering information. Conventional data Structures do not provide sufficient flexibility in managing media data, because of the inability to manage spatio-temporal relations, to recognize contents in multimedia data which relates to recognition of semantics of the contents of media data, and to allow various types of query representation[18]. Since the computational world is full of multimedia information, it is highly essential to know about the multimedia data structures which are existing. Multimedia information is very expressive, self explanatory, narrative, etc[19]. Organizing video data and provide the visual content in compact forms becomes important in multimedia applications [3]. Diverse application areas are increasingly relying on multimedia understanding systems. Advances in multimedia understanding are related directly to advances in various disciplines including signal processing, computer vision, pattern recognition, multimedia databases and smart sensors[14]. To address the emerging needs of applications, that require access to and retrieval of multimedia data, this paper provides a deep understanding in such aspects.

Applications of multimedia databases using multimedia data structures are as follows: [17]

- Computer graphics, games, movies.
- Computer vision, CAD.
- For tracking of locations using street maps.
- Government and commercial uses of remote sensing pictures, satellite images, etc.
- Digital libraries, including digital catalogs, product brochures, training and education.
  Broadcast and entertainment.
- Medical databases, such as X-rays, MRI scans CT scans and etc.
- Special-purpose databases e.g. face/fingerprint databases for security, business directories, maps, etc.

1.1. Multimedia Data[13]

Media is divided into two classes: continuous and discrete. Continuous media such as audio and video, change with time. Discrete media are time independent. Common examples of discrete media are text (formatted and unformatted), still images and graphics.

*1.2. Multimedia data characteristics [*13*]*

Characteristics of multimedia data are summarized here:

Lack of structure: Multimedia data tend to be unstructured.

Temporality: Some multimedia data types such as video, audio and animation sequences have temporal requirements that have implications on their storage, manipulation and presentation

Massive Volume: Multimedia data such as video and audio often require a large storage device.

1.3. Spatial Data

Spatial data consists of spatial objects made up of points, lines, regions, rectangles, surfaces, volumes, and even data of higher dimension which includes time. Examples of spatial data include cities, rivers, roads, counties, states, mountain ranges, parts in a CAD system, etc. Examples of spatial properties include the extent of a given river, or the boundary of a given county, select regions within in a boundary, etc. A common way to deal with spatial data is to store it explicitly by parameterizing it and thereby obtaining a reduction to a point in a possibly higher dimensional space. Conventional database system is easy to use for this purpose since the system is just a collection of records, where each record has many fields. Managing spatial relation is one of the mandatory feature in many multimedia applications. One of the straightforward way of managing the spatial relation between components of information is to represent it by rectangular coordinates[2].

2. DIFFERENT MULTIMEDIA DATA STRUCTURES

There are several multimedia data structures namely, k-d tree, point quad tree, MX quad tree, R-trees, TV trees. The main aim of these techniques and data structures is to divide the multidimensional feature space into many subspaces so that only one or a few subspaces need to be searched for each query[11]. In this section, two important multimedia data structures are reviewed.

2.1. The k-d Tree[6]

The "k-dimensional tree" which is usually abbreviated as "k-d tree." This structure is a natural generalization of the standard one-dimensional binary search tree [1]. kD-tree is the most important multidimensional structure. Decomposes a multidimensional space into hyperrectangles. A binary tree with both a dimension number and splitting value at each node [16].A recursive space partitioning tree. Partition along x and y axis in an alternating fashion. Each internal node stores the splitting node along x (or y). Used for point location and multiple database queries, k –number of the attributes are used to perform the search. Geometric interpretation is used to perform search in 2D space thus the name 2-d tree, generally used.

2.1.1. Construction of k-d tree:

Constructing a k-d tree requires, selecting the splitting value and choosing the dimension to split[15]. A 2-d tree is a binary search tree which satisfies the conditions

a) If N is a node in the tree such that level (N) is even, then every node M in the subtree rooted at N. LeftLink has the property that M. XVal<N. XVal, and every node P in the subtree rooted at N. RightLink has the property that P. XVal≥N. XVal.

b) If N is a node in the tree such that level(N) is odd, then every node M in the subtree rooted at N. LeftLink has the property that M. YVal<N. YVal, and every node Q in the subtree rooted at N. RightLink has the property that Q. YVal≥N. YVal.

2.1.2. Insertion in k-d tree:

For better understanding, the number of dimension considered is two. Inserting a node Q into the tree referred by T, need to branch let if Q.XVal<T.XVal and branch to right side if Q.XVal>T.XVal.

Let us for example consider a grid of size 8, the origin of the grid is (0, 0) is at the bottom-left corner of the grid. The following information is used to understand the insertion in a 2d tree.

Table 1. Sample values for insertion in 2d tree

Places	(XVal , YVal)
LM	16,22
PT	45,32
CT	55,30
TX	3,4
DW	60,20

For insertion let us consider each point one by one in order. Initially the 2-d tree is empty. The tree is constructed as follows and the illustration is given in Figure:1

1. Inserting the first node LM(16,22) results with one node.
2. When inserting PT, the XValues are compared and PT is inserted to the left of LM as LM’s XVal is lesser than PT.
3. Now, while inserting CT, the XVal of CT and LM are compared. Since CT’s XVal is greater the right side is chosen and now compared with PT’s XVal, since CT’s XVal is again greater than PT, the right side is chosen.
4. When inserting TX, the XVal of TX and LM are compared and leftside is chosen as TX XVal is lesser than LM.

5. At last, when inserting DW, the XVAL of DW is compared with LM , right side is chosen and now XVAL of PT is checked, so again rightside is chosen and finally the XVal of DW is compared with CT, also here right side is chosen because DW's XVal is greater than CT.

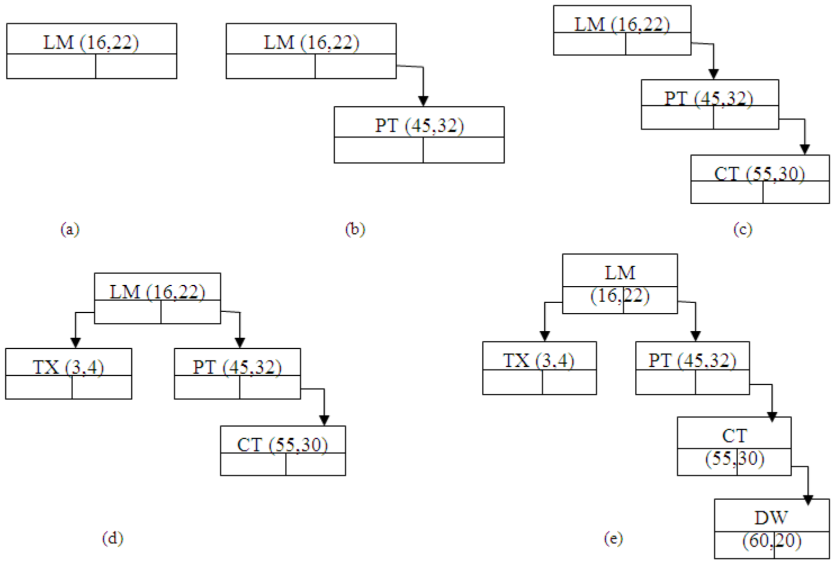

Figure 1: 2-d tree representing the example above (a), (b), (c), (d) and (e) represents the steps 1-5.

2.1.3. Deletion in k-d Tree:

Deletion in k-d tree is little complex. If the node to delete is leaf, the deletion is simple. If the node is interior node the process is little complex. The deletion of interior node Q consists of the following steps:

a) Need to find a replacement code U that occurs in the Tree.
b) Replace all of Q's nonlink fields by those of U
c) Recursively delete U from T

2.1.4. Applications of k-d Trees[8]:

- Query processing in sensor networks
- Nearest-neighbor searchers
- Optimization
- Ray tracing

2.1.5. Advantages:

An efficient data structure for finding nearest neighbors in multidimensional data. The kd tree is the only data structure that allows easy multi-key search.

2.2. Quad Tree

Development of quadtree as hierarchical data structuring technique for representing spatial data (like points, regions, surfaces, lines, curves, volumes, etc.) has been motivated to a large extent by storage requirements of images, maps, and other multidimensional data. Quad-tree structures are often used to model motion between frames of a video sequence[12]. The quad tree is a data structure appropriate for storing information to be retrieved on composite keys[4]. Briefly, the quadtree is a class of hierarchical data structures which is based on the recursive partition of a square region into quadrants and sub-quadrants until a predefined limit[6]. A quad tree for representing a picture is a tree in which successively deeper levels represent successively finer subdivisions of picture area[7]. Most of the applications involve geometric data. They doesn't possess natural ordering[18]. The Geometric Objects are classified as follows: **Scalars**: 1-d point. **Point**: location in d-dimensional space. Arrays: double p[d]; structures: struct { double x, y, z; } **Vectors**: direction and magnitude (length) in that direction[18].

2.2.1. Point Quad Trees

Recursively subdivide cells into 4 equal-sized sub cells until a cell has only one point in it[18]. Each division results in a single node with 4 child pointers called NorthWest(NW), SouthWest(SW), NorthEast(NE), SouthEast(SE). When cell contains no points, add special "no-point" node. Figure 2. Shows the graphical represention. Figure 3. Shows the Point Quad tree for the entries specified in Table1.

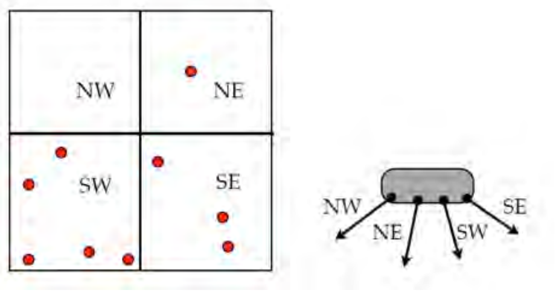

Figure 2: Partitioning in Quad Tree[17]

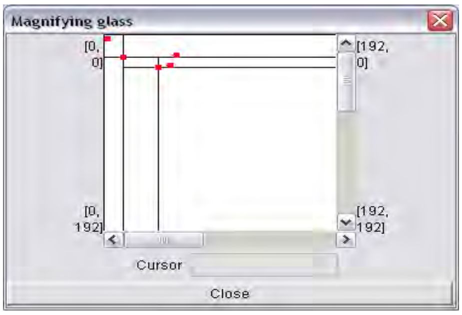

Figure 3: Represents the position of Point Quad Tree For entries in Table 1.

2.2.2. MX Quad Tree (MatriX Quad Tree)

MX Quad Tree is useful for image data, smallest element Space is recursively subdivided until smallest unit is reached[18]. Can be used to represent a matrix (especially 0/1 matrix) - recursive decomposition of matrix (given by the MX tree) can be used for faster matrix transposition and multiplication. The aim behind MX-quad trees was to ensure that the shape of the tree was independent of the number of nodes present in the tree, as well as the order of insertion of the nodes. The Figure 4, shows the MX Quad Tree for the entries specified in Table1.

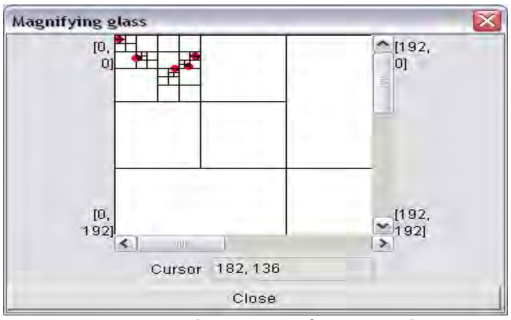

Figure 4: Represents the position of Point Quad Tree For entries in Table 1

2.2.3. Applications of Quad tree [5]

- Spatial indexing.
- Efficient collision detection in two dimensions.
- View frustum culling of terrain data.
- Storing sparse data, such as a formatting information for a spreadsheet or for some matrix calculations.
- Solution of multidimensional fields.
- Conway's Game of Life simulation program.
- State estimation.
- Quadtrees are also used in the area of fractal image analysis.

2.3. CONCLUSION

Thus in this paper, information specified by different researches are collectively documented. Also in this paper we have reviewed the two different multidimensional (multimedia) data structures. Understanding Multimedia systems stand to benefit for the advances in feature extraction, pattern classification, learning and human computer interaction. This paper will provide fundamental idea to understand the two spatial multidimensional DataStructures such as k-d tree and quad tree.

References

[1] Andrew W. Moore, Efficient Memory-based Learning for Robot Control, PhD Thesis: Technical Report No. 209, Computer Laboratory, University of Cambride, 1991

[2] Atsuo Yoshitaka, Member, IEEE, and Tadao Ichikawa, “A Survey on Content-Based Retrieval for Multimedia Databases”, IEEE Transactions On Knowledge And Data Engineering, VOL. 11, NO. 1 , pp.81 – 93, 1999.

[3] B-L Yeo and M.M. Yeung, “Retrieving and Visualization Video”," Comm. of the ACM, Vol. 40, No. 12, December 1997, pp. 43-52.

[4] Finkel, R., & Bentley, J. L. (1974). Quad Trees: A Data Structure for Retrieval on Composite Keys. Acta Informatica 4(1): 1-9.

[5] http://en.wikipedia.org/wiki/Quadtree

[6] http://www.docstoc.com/docs/112420020/Multidimensional-Data-Structures

[7] Hunter, Gregory.M.Steiglitz, Kenneth, “Operations on Images Using Quad Trees”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume:1 , Issue: 2 ,pp. 145 - 153 (1979)

[8] Jianping Fan , Department of Computer Science , UNC-Charlotte “Spatial data structures –kd-trees. ppt”

[9] Kuo F.- Effelsberg W.- Garcia-Luna-Aceves J.; “*Multimedia Communications*”, Prentice Hall, 1998

[10] Li, Zhu; Katsaggelos, Aggelosli.K, “A Kd-Tree Based Dynamic Indexing Scheme for Video Retrieval and Geometry Matching”, Proceedings of 17th International Conference on Computer Communications and Networks,pp.1-5, (2008).

[11] Lu, Guojun , “Techniques and data structures for efficient multimedia retrieval based on similarity”, IEEE Transactions on Multimedia, Volume: 4 , Issue: 3 , Page(s): 372 – 384, (2002)

[12] Mathew, Reji, Taubman, David S., “Joint scalable modeling of motion and boundary geometry with quad-tree node Merging ”, In 16th IEEE International Conference on Image Processing, pp. 3745 - 3748 (2009)

[13] Oya Kalipsiz, Yildiz, “Multimedia Databases” , Technical University, Computer Science Department Yildiz Istanbul, Turkey,2000 IEEE, 0-7695-0743-3/00

[14] Naphade, Milind Ramesh, “Multimedia understanding: challenges in the new millennium” In Proceedings of International Conference on Image Processing, Volume: 3 , Page(s): 33 – 37, 2000.

[15] Samet, Foundations of Multidimensional and Metric Data Structures, pp. 446-473 and 827-832.

[16] Steve Renals, “Nearest neighbors and kD-trees”, Informatics 2B— Learning and Data Lecture, 2007.

[17] Tom Huang, Sharad Mehrotra, Kannan Ramchandran, “Multimedia Analysis and Retrieval System (MARS) Project”

[18] Wikipedia on Kd Trees, Quad Trees.

[19] Yihun Alemu, Jong-bin Koh, Muhammed Ikram, Dong-Kyoo Kim, “Image Retrieval in Multimedia Databases: A Survey” International Conference on Intelligent Information Hiding and Multimedia Signal Processing,pp.681-689 (2009).