Academic Science

International Journal of Innovations & Advancement in Computer Science
IJIACS
ISSN 2347 – 8616
Volume 3, Issue 1
April 2014

# Multi-Dimensional Image Indexing with R+-Tree

Megha Mihir Shah
M.Tech , I.E.T College of Engineering & Tech.
Alwar, Rajasthan
megha.shah2008@gmail.com

Pratap Singh Patwal
Prof. I.E.T College of Engineering & Tech.
Alwar, Rajasthan
pratappatwal@gmail.com

## ABSTRACT

In traditional approach used for image indexing is typically compute a single signature for each image based on color histograms, texture, wavelet transforms, region based etc, and return as the query result, image whose coordinates match closet to the inserted query image. Therefore, sometime it becomes very hard to match the similar coordinates when the objects that scaled differently or at different locations or only certain regions of the image match.

The research aspects of image feature representation and extraction, multi-dimensional indexing and based on the Content-Based Image Retrieval. Second, a variation to Guttman's R-trees (R+-trees) that avoids overlapping rectangles in intermediate nodes of the tree is intro - duced. Algorithms for searching, updating, initial packing and reorganization of the structure are discussed in detail. Finally, we provide analytical results indicating that R+-trees achieve up to 50% savings in disk accesses compared to an R-tree when searching files of thousands of rectangles.

## Keywords

Texture analysis, feature extraction, Multidimensional image, R+- tree

## INTRODUCTION

The existing system to store image data is not handling the image data in such a way, as it a rapid increase ot the size of digital image collection. The text-based and visual-based data needed to store in large amount. Many advances, Data Modeling, Multi- Dimensional Indexing, Query Evaluation, etc. have been major difficulties, in the size of image collections is large[1,2,7]. The main operations that have been pointing before are:

- *Point Queries*: given a point in the space of image, fine all images that contain it
- *Region queries*: Given region, find all shape boundaries that passed through it.

Finding index structures which allow efficient searching of an image database is still an unsolved problem [Faloutsos et al, 1994]. None of the index structures proposed for text retrieval has proved applicable to the problem.

The most promising approach so far has been multidimensional indexing, using structures such as the R*-tree [Beckmann et al, 1990], the TV-tree [Lin et al, 1994] and the SS+-tree [Kurniawati et al, 1997], but the overheads of using these complex index structures are considerable. A

more recent approach, which seems to offer better prospects of success, is the use of similarity clustering of images, allowing hierarchical access for retrieval and providing a way of browsing the database as a bonus.

The purpose of this paper is to describe new structure of Feature extraction based on texture for multi-dimensional images using R+ tree.

## 2. FEATURE EXTRACTION BY TEXTURE

Feature extraction is the basis of content-based Image Retrieval. These features includes both text-based( keywords, annotations, etc) and visual-based ( color, texture, shape, faces, etc.)

The former includes color, texture and shape features.. the domain-specific features are better covered in pattern recognition.

### 2.1 Texture

In the image the texture is the visual pattern having property of homogeneity. This includes surface, clouds, tree, background of image, etc. it contain important information about the structural arrangement of surface.

Feature extraction can be done by:

- ***Spectral Approach***: The spectral approach to texture analysis deals with images in the frequency domain. Therefore, this approach requires[1,22] Fourier transform to be carried out on the original images to acquire their corresponding representations in the frequency space. The two-dimensional power spectrum of an image reveals much about the periodicity and directionality of its texture[4,21,22]. For instance, an image of coarse texture would have a tendency towards low frequency components in its power spectrum, whereas another image with finer

Academic Science

**International Journal of Innovations & Advancement in Computer Science**
**IJIACS**
**ISSN 2347 – 8616**
**Volume 3, Issue 1**
**April 2014**

texture would have higher frequency components. Stripes in one direction would cause the power spectrum to concentrate near the line through the origin and perpendicular to the direction.

- ***Structural methods****:* In structural methods, texture is viewed as consisting of many textural elements (called texel) arranged according to some placement rules. From Fig. 2.1, the structural properties of some textures can be observed by human. This leads to many structural analysis methods.

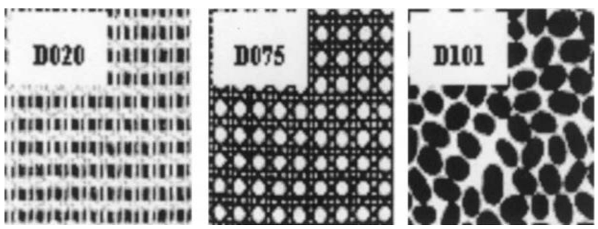


**FIGURE 2.1. Examples of structural textures (selected from the Brodatz album).**

The structural properties of these elements have been successfully applied to characterize the textures by many authors[4,24,30]. Commonly used element properties are average element intensity, area, perimeter, eccentricity, orientation, elongation, magnitude, Compactness[26], euler number, moments, etc. A structural method suits better for a description of a macro-texture.

Structural texture methods have been extended to invariant texture classification. A difficulty of these methods is how to extract texels of a texture.

# 3.MULTI-DIMENSION INDEXING TECHNIQUE

we have to select appropriate multidimensional indexing algorithms to index the reduced but still high dimensional feature vectors. There are three major research communities contributing in this area, i.e. computational geometry, database management, and pattern recognition. The existing algorithm, k-d tree, priority k-d tree [30], quad-tree, K-D-B tree, hB-tree, R-tree and its variants *R*C-tree and *R*+-tree [11, 19, 10, 6, 15]. Still their performances were far from satisfactory. Pushed by then urgent demand of spatial indexing from GIS and CAD systems, Guttman proposed the R-tree indexing structure in 1984 [17]. Based on his work, many other variants of R-tree were developed. Sellis *et al.* proposed *R*C tree in [19]. Greene proposed her variant of R-tree in [10]. In 1990, Beckman and Kriegel proposed the best dynamic R-tree variant, *R**-tree [6]. However, even for *R**-tree, it was not scalable to dimensions higher than 20 [18]. Very good reviews and comparisons of various indexing techniques in image retrieval can be found in [30]. The research goal of White and Jain in [30] was to provide general purpose and domain-independent indexing algorithms. Motivated by k-d tree and R-tree, they proposed VAM k-d tree and VAMSplit R-tree.

After dimension reduction using the image approach, the following three characteristics of the dimension-reduced data can be used to select good existing indexing algorithms: $^2$ the new dimension components are ranked by decreasing variance,

$^2$ the dynamic ranges of the dimensions are known,

$^2$ the dimensionality is still fairly high.

On their test data sets, they found that the BA-KD-tree gave the best performance. Considering that most of the tree indexing techniques were designed for traditional database queries (point queries and range queries) but not for the similarity queries used in image retrieval, there was a need to explore the new characteristics and requirements for indexing structures in image retrieval.

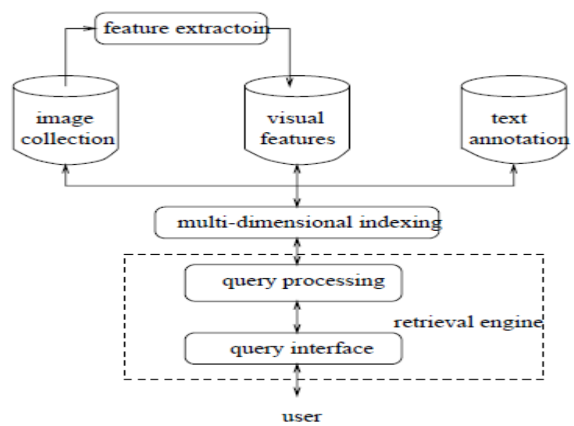


**Figure 3.1. Multi-Dimensional feature extraction**

Such a technique was explored in [31], where Tagare developed a tree adaptation approach which refined the tree structure by eliminating inefficient tree nodes for similarity queries. So far, the above approaches only concentrated on how to identify and improve indexing
techniques which are scalable to high dimensional feature vectors in image retrieval. The other nature of feature vectors in image retrieval, i.e. non-Euclidean similarity measure not been deeply explored.

# 4. R$^+$ - Tree

In this section we introduce the R+-tree and discuss the algorithms for searching and updating the data structure.

## 4.1. Description

As mentioned above, R-trees are a direct extension of B-trees in *k*-dimensions. The data structure is a height-balanced tree which consists of intermediate and leaf nodes. Data objects are stored in leaf nodes and intermediate nodes are built by grouping rectangles at the lower level. Each intermediate node is associated with some rectangle which *completely* encloses all rectangles that correspond to lower level nodes. Figure 4.1 shows an example set of data rectangles and Figure 4.2 the corresponding R-tree built on these rectangles (assuming a branching factor of 4). Considering the performance of R-tree searching, the concepts of *coverage* and *overlap* [19] are important. Coverage of a level of an R-tree is defined as the total area of all the rectangles associated with the nodes of that level. Overlap of a level of an R-tree is defined as the total area contained within two or more nodes. Obviously, efficient R-tree searching demands that both overlap and coverage be minimized. Minimal coverage reduces the amount of *dead space* (i.e. empty space) covered by the nodes. Minimal overlap is even more critical than minimal coverage. For a search window falling in the area of *k* overlapping nodes at level *h −l*, with *h* being the height of the tree, in the worst case, *k* paths to the leaf nodes have to be

followed (i.e. one from each of the overlapping nodes), therefore slowing down the search from *l* to *l k* page accesses.

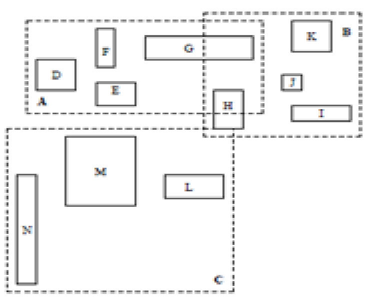


Figure 4.1: Some rectangles organized into R-tree

For example, for the search window **W** shown in Figure 4.3, both subtrees rooted

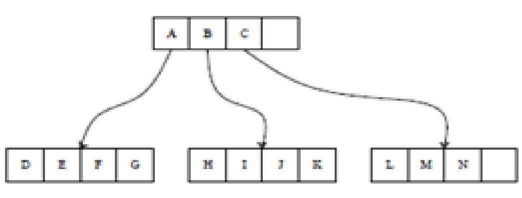


Figure 4.2: R-tree for rectangles of Figure 4.1

at nodes *A* and *B* must be searched although only the latter will return a qualifying rectangle. The cost of such an operation would be one page access for the root and two additional page accesses to check the rectangles stored in *A* and *B*. Clearly, since it is very hard to control the overlap during the dynamic splits of R-trees, efficient search degrades and it may even degenerate the search

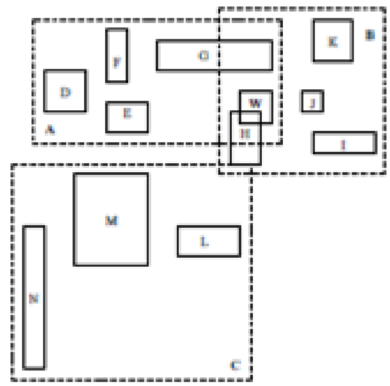


Figure 4.3: An example of a "bad" search window

from logarithmic to linear. It has been shown, that zero overlap and coverage is only achievable for data points that are known in
advance and, that using a packing technique for R-trees, search is dramatically improved [19]. In the same paper it is shown that zero overlap is not attainable for region data objects. However, if we allow partitions to *split* rectangles then zero overlap among intermediate node entries can be achieved. This is the main idea behind the R+-tree structure. Figure 3.4 indicates a different grouping of the rectangles of Figure 4.1 and Figure 4.5 shows the corresponding R+-tree. Notice that rectangle *G* has been split into two sub-rectangles the first contained in node A and the second in P. That is, whenever a data rectangle at a lower level overlaps with another rectangle, we decompose it into a collection of non-overlapping sub-rectangles whose union makes up the original rectangle. R+-trees can be thought as extension


Figure 4.4: The rectangles of Figure 4.1 grouped to from an R+-tree

of K-D-B trees to cover non-zero area objects (i.e. not only points but rectangles as well). An improvement over the KD-B-trees is the reduced coverage; the nodes of a given level do not necessarily cover the whole initial space. Moreover, compared to R-trees, R+-trees exhibit very good searching performance, especially for point queries, at the expense of some extra space.

# CONCLUSION

In this paper, This article mainly discussed various texture classification and feature extraction techniques. The Basic Statistical approaches yield characterizations of textures as smooth, coarse, grainy, and so on. One of the simplest approaches for describing texture is to use moments of the gray-level histogram of an image or region. Structural approaches deal with the arrangement of image primitives. They use a set of predefined texture primitives and a set of construction rules to define how a texture region is constructed with the primitives and the rules past and current technical achievements in visual feature extraction, multidimensional indexing, and system design are reviewed. Open research issues are identified and future research directions suggested. From the previous section, we can see that a successful image retrieval system requires the seamless integration of multiple research communities' efforts. With some indicative results of the search performance of both R- and R+-trees. First, we show the number of disk accesses required to search an R-tree or R+-tree in case of a point query. In the coming days it would become very easy to Content- Based Image Retrieval for multidimensional images

# REFERENCES

[1]A Photographic Album for Artists and Designers" by Brodatz, P., Textures: New York, Dover Publications

[2] "A Statistical Approach to Texture Classi_cation from Single Images " by Chen, C. C., *Markov Rando,* Manik Varma and Andrew Zisserman International Journal of Control

[3] Mandis Beigi, Ana Benitez, and S.F. Chang. In Proc. SPIE storage and Retrievel for Image and Video Databases, San Jose. http://www.ctr.columbia.edu/metaseek.

[4] MPEG-7 applications document. ISO/IEC JTC1/SC29/WG 11 N1911, MPEG97, Oct1997.

[5] Third draft of MPEG-7 requirements . ISO/IEC JTC1/SC29/WG 11 N1921, MPEG97, Oct1997.

[6] James Allen. Relevance feedback with too much data. In Proc. Of SIGIR'95, 1995

[7] Richard O. Dud and Peter E. Hart. Pattern Classification and cene Analyis, Chapter 6.

[8] C. Faloutsos, M. Flickner, w. Niblack, D. Petkovic, Efficient and effective quering by image content. IBM research Report, Aug, 1993.

[9] Chistos Faloutso and King-Ip Lin. Fastmap: A fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets In Proc. Of SIGMOD.

[10] Diane Greene. An implementation and performance analysis of spatial data access. In proc. ACM, 1989.

[11] M. H. Gross, R. Koch, Multiscale image texture analysis in wavelet paces. IEEE Int. Conf. Image Proc

[12] A Guttman. R-Tree: a dymanic index structure for spatial searching. In proc. ACM SIGMOD, 1984

[13] Texture Analysis and Indexing Using Gabor-like Hermite Filters Carlos Joel Rivero-Moreno Stéphane Bres

[14] Y. Rui, K. Chakrabarti, S. Mehrotra, Y. Zhao, and T. S. Huang, Dynamic clustering for optimal retrieval in high dimensional multimedia databases, in *TR-MARS-10-97, 1997.*

[15] Y. Rui, T. S. Huang, and S. Mehrotra, Content-based image retrieval with relevance feedback in MARS, in *Proc. IEEE Int. Conf. on Image Proc., 1997.*

[16] L. Yang and F. Algregtsen, Fast computation of invariant geometric moments: A new method giving correct results, in *Proc. IEEE Int. Conf. on Image Proc., 1994*.

[17] N. Roussopoulos and D. Leifker, ''Direct Spatial Search on Pictorial Databases Using Packed RTrees,'' *Proc. ACM SIGMOD* , May 1985.

[18] H. Samet, ''Quadtrees and Related Hierarchical Data Structures for Computer Graphics and Image Processing,'' 1986. (under preparation)

[19] M. Stonebraker, B. Rubenstein, and A. Guttman, ''Application of Abstract Data Types and Abstract Indices to CAD Data Bases,'' Tech. Report UCB/ERL M83/3, Electronics Research Laboratory, University of California, Berkeley, January 1983.

[20] M. Stonebraker, T. Sellis, and E. Hanson, ‘‘Rule Indexing Implementations in Database Systems,’’ *Proceedings of the First International Conference on Expert Database Systems* , April 1986

[21].“Textural Properties for Pattern Recognition,” by Hawkins, J. K In *Picture Processing and Psychopictorics,*

[22]. “*A Photographic Album for Artists and Designers” by* Brodatz, P., *Textures:* New York, Dover Publications

[23]. “A Statistical Approach to Texture Classi_cation from Single Images “ by Chen, C. C., *Markov Rando,* Manik Varma and Andrew Zisserman International Journal of Control

[24]. “A New Method for Classification of Structural Textures” by Bongkyu Lee

[25]. “Statistical Texture Analysis” by G. N. Srinivasan, and Shobha G.

[26]. “Texture Classification. Machine Vision and Media Processing “

[27]. “Computer Analysis of visual textures” by F Tomita and S Tsuji

[28]. “Edge Detection in textures, Computer Graphics and image processing “ by L.S Davis and A.Mitiche

[29].” Texture measures for automatic classification of pulmonary diseases” by R.Sutton and E.Hall, IEEE Transactions on computers.

[30]. D. White and R. Jain, Similarity indexing: Algorithms and performance, in *Proc. SPIE Storage and Retrieval for Image and Video Databases, 1996*.

[31] D. L. Swets and J. J. Weng, Efficient content-based image retrieval using automatic feature selection, in *Proc. IEEE, 1995*.

[32]”Texture Analysis Methods – A Review” by Andrzej Materka and Michal Strzelecki

[33]” Contextual classification and segmentation of textured images” by P W Fung, G Grebbin, and Y.Attikiouze

[34] “Image Segmentation and Feature Extraction,” by Sklansky, J *IEEE Transactions on Systems,*

[35]“Statistical and Structural Approaches to Texture,” by Haralick, R.M., *Proceedings of the IEEE*, 1979.

[36] “Image Segmentation and Feature Extraction,” by Sklansky, J., *IEEE Transactions on System.*