Chapter 3

R*-Tree Based Similarity and Clustering Analysis for Images

Jiaxiong Pi
University of Nebraska at Omaha, USA

Yong Shi
University of Nebraska at Omaha, USA

Zhengxin Chen
University of Nebraska at Omaha, USA

ABSTRACT

Image content analysis plays an important role for adaptive multimedia retrieval. In this chapter, the authors present their work on using a useful spatial data structure, R*-tree, for similarity analysis and cluster analysis of image contents. First, they describe an R*-tree based similarity analysis tool for similarity retrieval of images. They then move on to discuss R*-tree based clustering methods for images, which has been a tricky issue: although objects stored in the same R* tree leaf node enjoys spatial proximity, it is well-known that R* trees cannot be used directly for cluster analysis. Nevertheless, R* tree’s indexing feature can be used to assist existing cluster analysis methods, thus enhancing their performance of cluster quality. In this chapter, the authors report their progress of using R* trees to improve well-known K-means and hierarchical clustering methods. Based on R*-Tree’s feature of indexing Minimum Bounding Box (MBB) according to spatial proximity, the authors extend R*-Tree’s application to cluster analysis containing image data. Two improved algorithms, KMeans-R and Hierarchy-R, are proposed. Experiments have shown that KMeans-R and Hierarchy-R have achieved better clustering quality.

INTRODUCTION

Image content analysis has found many applications in various domains (such as in biomedical science) and plays an important role for adaptive multimedia retrieval. In order to effectively analyze the image contents, first we should be able to access and manipulate the images themselves (rather than staying with the metadata description of the images). Research effort has been made in this regard. For example,
aimed at supporting a better coupling between enhanced image processing technique and an advanced image management for storage and retrieval, Li and Chen (2005) developed a system which supports the coupling of effective image management with enhanced image processing techniques, where image retrieval is conducted by using an image algebra. This approach is illustrated by a prototype medical image retrieval system on patient chromosome images, where similarity-based retrieval is conducted: when a query chromosome is provided, patient chromosomes in the database that are similar to the query chromosome are retrieved.

Upon manipulating images, the next important task for image content analysis is to effectively index the images. Various spatial data structures developed in the past can be useful for dealing with the spatial data store in images, including KD-trees (Bentley 1975), octrees (Aronov et al., 2003), TV-tree (Lin et al., 1994), X-tree (Berchtold et al. 1996), SR-tree (Katayama and Satoh, 1997), M-tree (Ciaoccia et al. 1997), etc. In this chapter, we report our research of similarity search and clustering methods on images using on more traditional R*-tree data structure, because R*-tree (and its “relative” R-tree) is a popular and relative mature technique. This research is also a continuation of our own related research in the past (e.g., Schreck and Chen, 2000; Durby, Shi and Chen, 2004). After a brief review on the basics of R*-trees, the presentation of our current work consists of two related topics: First, we describe an R*-tree based similarity analysis tool for similarity retrieval of images. We then move on to discuss R*-tree based clustering methods for images, presenting R*-tree based algorithms of KMean-R and Hiearchy-R. Experiments have shown that KMeans-R and Hierarchy-R have achieved better clustering quality. The contribution of R*-tree based clustering to image content analysis leads us to further consider ontology-related issues, as to be briefly addressed in a discussion section.

**BASICS OF R* TREES**

Just like a B-Tree, an R-Tree (Guttman 1984) relies on a balanced hierarchical structure, in which each tree node is mapped to a disk page. However, whereas B-Trees are built on single-value keys and rely on a total order on these keys, R-Trees organize rectangles according to a containment relationship. Each object to be indexed will be represented by Minimum Bounding Box (MBB) in the index structure except point for which an MBB simply degrades to a point. All indexed objects will eventually be put in leaf nodes. A leaf node contains an array of leaf entries. A leaf entry is a pair (mbb, oid), where mbb is the Minimum Bounding Box (MBB) and oid is the object ID. Each internal node is associated with a rectangle, referred to as the directory rectangle (dr), which is the minimal bounding box of the rectangle of its child nodes. The structure of R-Tree satisfies the following properties:

1. For all nodes in the tree (except for the root), the number of entries is between $m$ and $M$, where $0 \leq m \leq M/2$.
2. For each entry (dr, node-id) in a non-leaf node $N$, dr is the directory rectangle of a child node of $N$, whose page address is node-id.
3. For each leaf entry (mbb, oid), mbb is the minimal bounding box of spatial component of the object stored at address oid.
4. The root has at least two entries (unless it is a leaf).
5. All leaves are at the same level.