Automating the Indexing of Images Using Tree Tessellation Algorithm v.10

D. Tsishkou[1] L. Chen[1]

[1] LIM (Laboratory of Informatics and Multimedia)

Ecole Centrale de Lyon
36 Avenue Guy de Collongue, 69131 Ecully Cedex – France

{dzmitry.tsishkou, liming.chen}@ec-lyon.fr

Abstract

This paper describes an experimental study of TTA10 algorithm for indexing and retrieval of large image databases. Our approach consist of primary and secondary tessellation stages, therefore it can be used in computer cluster with parallel multiprocessor architecture. We propose results on compact index structure, which increases systems scalability and achieves logarithmic complexity. Tree structured index in Tree Tessellation Algorithm v.10 (TTA10) is optimized for balance monitoring and real-time sub-tree reorganization, avoiding of changes in the structure of the entire tree. Finally we will describe a practical implementation of our indexing/retrieval system on a database of 122000 images and post results on tree balance, topology effectiveness, complexity and performance.

Keywords

Indexing, retrieval, complexity, images, database, multimedia, MPEG-7, video.

1 Introduction

In recent years, there has been a growing interest in developing effective methods for searching large image databases based on image content. The interest in image search algorithms has grown out of the necessity of managing large image databases the are now commonly available on web and wide area networks [1,2]. Visual information is rich in content [6]. The same picture may invoke different responses from different user, at the different time, and in different contexts. A document may have different meanings at different levels, e.g., description, analysis, and interpretation [1]. With MPEG-7, an audio/video document is represented by a hierarchical structure both syntactically and semantically [2]. With syntactic decomposition, a document is divided into a hierarchy of segments, known as segment tree. A segment is further divided into video segment and audio segment, corresponding to the video frames (images) and the audio waveform, respectively. A frame can be recursively divided into subregions to form a region tree. Each subregion is described by a set of descriptors. The visual descriptors can be categorized into four groups: color, shape, motion, and texture. This paper concentrates on the images indexing/retrieval algorithm that use MPEG-7 visual descriptors as objects features. Recent research has produced much progress in visual information retrieval. Several systems, such as Virage, QBIC, Photobook, VisualSEEk, VideoQ [3] provide efficient tools for users to specify visual queries using image examples or sketches. Multiple visual features are used in combination. In order to work with large number of widely varying real-word objects taken in natural settings, some of them use a hierarchical structure for index storage. This paper we don't consider hierarchical region segmentation, comparison of visual descriptors and construction of multiple feature based search system. We evaluate TTA10 based solution to index/retrieve images using one visual descriptor represented as a feature vector. The most widely studied methods for speeding up search solution are k trees, k-d trees [5], R trees, tsvq, and other tree structured methods. Hierarchical discriminate analysis for image retrieval is also studied [4]. TTA10 algorithm uses primary and secondary tessellation stages in order to make an efficient tree-structured index. The major characteristics of the effective hierarchical (tree-based) image indexing/retrieving system are following:

- Logarithmic complexity, i.e. tree structure is highly balanced in general
- Compact index structure, which makes possible to use search system on the mid-end hardware
- Paralleled indexing/retrieval algorithm architecture in order to make computer cluster for high-end image databases with billions of samples
- Dynamic sub-tree monitoring and reorganization, which allows to add the new images into database in real-time without rebuild of entire tree-structured index
- Effective tree-topology structure organization, that is necessary to speed-up the process of nearest neighbors search

Current paper is organized as follows :

Section 2 presents a general overview of the TTA10 algorithm, and its parallel architecture is discussed. Section 3 presents variety of experiments designed to analyze system's behavior on several videos decomposed into frames (still images) and the set of web images. Section 4 concludes by presenting promising directions for future research.

2 Tree Tessellation Algorithm v. 10

Standard search methods for tree-based indexing transverse directly from the root to a leaf of the tree choosing the branch which minimizes distance between the query descriptor and the centroid at each point [4]. However, this strategy is not optimum since it does not allow dividing tessellation process into coarse/fine stages.

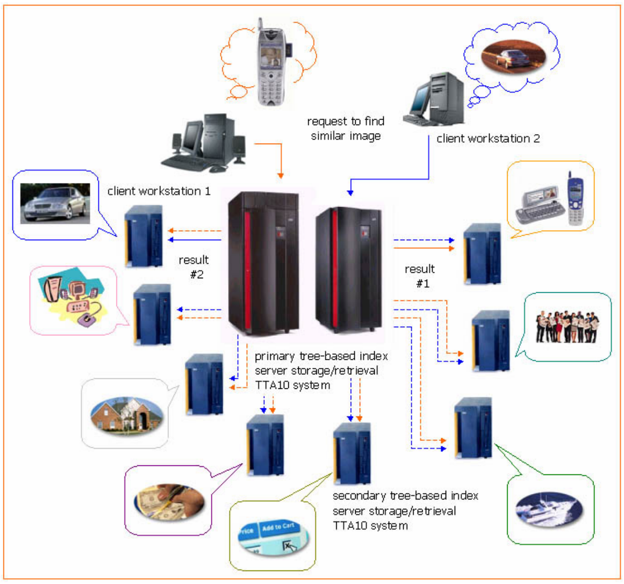

Figure 1 - *Parrallel architecture of Tree Tessellation Algorithms v.10*

We use primary and secondary tessellation during indexing and retrieval. Primary tessellation is able to decompose the database into smaller tractable problems. At the root node of the primary tree, where the entire database of samples is found, the classes may not be efficiently separable [4,5]. But because we do not attempt to completely separate the classes at a single level, we can successfully solve the problem in stages by breaking it down into simpler pieces. The primary tessellation stage (coarse stage) cannot completely separate the classes; it can make the first stage in separation, dividing the samples among its children into clusters. Each cluster contains fewer samples than its parent does, and therefore, be optimized to the coarse set of samples. In order to maintain good generalization and performance of the system, achieving perfect scalability we propose to use secondary tessellation. It makes possible to result in a successful separation of classes, meanwhile using inherited parallel architecture and small memory consumption it makes entire system highly scalable. It can be used on a wide range of servers, from a single workstation to multi-processor cluster. Retrieval process starts at primary tree and goes through the index until the primary final node [4,5]. Each primary final node is connected to the sub-tree, this is a part of secondary index, and according to the transaction map is send to the secondary server, which stores current index. Finally secondary server finds the most similar cluster to the request, or the top-N best match. In general we are not limited to the current parallel architecture only; therefore we can use topology without central node as well. The tree is built one level at a time. The collection of children nodes represent a tessellation of their parent. Suppose that we want to add one trainig sample to entire index. In order to satisfy dynamic monitoring and sub-tree reorganization possibility, we use balance marks on each nodes structure. Balance mark represents number of images tessellated on the right and left leafs. The optimal balance should contribute to equal number of images, tessellated on the right and left. We monitor balance marks on each node and reorganize a sub-tree if a balance mark ratio is lower than specified threshold.

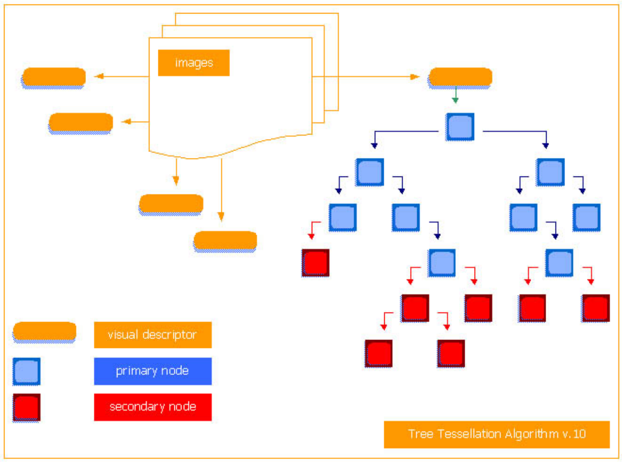

Figure 2 - *Primary (blue) / Secondary (red) indexes tree*

Operative reorganization is assigned to secondary tessellation index, meanwhile global reorganization should include primary index as well.

3 Experimental Setup

In this section, we present the results of testing TTA10 based system on a database of 122000 images. In order to test effectiveness of the tessellation and its topology we choose several different movies and set of web images. Each movie was decomposed into single frames (images), therefore we suppose that the two nearest frames are the most similar visually in general. Hence we could use current assumption as a one of criteria of topology effectiveness for some experimental research. So if the most similar image to the original one is a nearest-neighbor - the topology structure is near optimal. We use 64 bin quantized color histogram as a visual descriptor. Because we don't consider accuracy of classification for different visual descriptor or/and its combination, but we investigate, during current test, general performance of TTA10 indexing/retrieval approach (independently to the visual descriptors set). Each image was resized to 64x48 pixels in size in order to decrease computational costs on visual descriptor. The TTA10 image indexing/retrieval system was implemented on java 2.0, and all test were run on Pentium-450 based workstation. We used 2 parts of experimental database, where the first part includes 17785 frames from "5th Element", 38408 frames from "Charlie's Angels", 21844 frames from "Lock, Stock and Two Smoking Barrels", 17815 frames from "Shrek", and 18811 frames from "Six Sense" movie. Finally we add 9000 web images from Cambridge Research Lab. All original movies were stored in MPEG-4 format with standard quality.

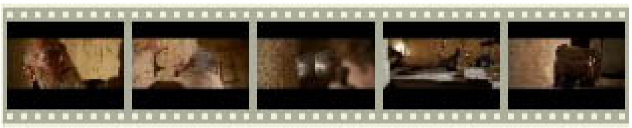

Figure - 3 *Sample frames from the "5th Element" video*

Figure - 4 *Sample frames from the "Charlie's Angels" video*

Figure - 5 *Sample frames from the "Lock, Stock and Two Smoking Barrels" video*

Figure - 6 *Sample frames from the "Cambridge Research Lab" web images*

Figure - 7 *Sample frames from the "Shrek" video*

Figure - 8 *Sample frames from the "Six Sense" video*

We have computed the tree balance parameter as a function that is equal to one in the case if all the nodes in the tree are completely balanced, i.e. the number of images tessellated thought each node to the left leaf of it and to the right leaf of it are the same. Other case the tree is not completely balanced. In order to be correct we also include information on the nodes level that is why the balance of the node, which is close to the root has higher weight in the total balance function. Results on the session one showed that the tree balance is highly stable in the most of cases. That has happened because of the tessellation procedure. The numbers near 0.8 demonstrated the effectiveness of current approach; however some progress could be made in the future. According to the summary of results obtained both for primary and total index time complexity is nonlinear. This could be explained by the nonlinearity of the primary tessellation algorithm and linearity of the secondary tessellation algorithm. Meanwhile the degree of nonlinearity is low, because of the usage of thresholds for accuracy/balance of the tree during tessellation process. On the session two, we used larger databases, containing up to 31278 images, however results on the tree balance, time complexity were the same quality as for session one. Results on tree balance showed high scalability of a TTA10 based system, because the tessellation algorithm is able to efficiently proceed large amount of a data. We suppose that the increase of a size of the database on more than several degrees will maintain current effectiveness, according to the results we have got. Final retrieval time complexity is compounded by descriptors time complexity and the search time complexity. According to the exprimental results the descriptors time comlexity is independent to the size of a tree. However absolute time comlexity to compute one descriptor is ten time higher than to search one image using this descriptor, therfore this time is a kind of a bottle-neck. Meanwhile this paper we introduced Tree Tessellation Algorithm v.10 and didn't compare visual descriptors.

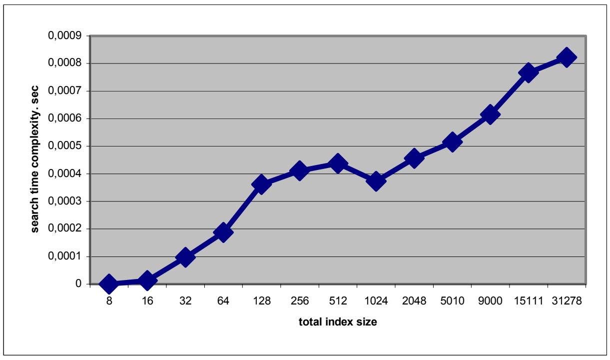

Figure – 9 *Session two. A comprasion of a search time complexity for various size of total index (primary index is 1/10 of total)*

Current results showed that TTA10 algorithm achieves near logarithmic comlexity on a wide range of database sizes. The same time we demonstarated retrieval(pure search) time complexity less that 1 millisecond for a database of 31278 images.

All results for the best match were using precomputed visual descriptors for each image; therefore all scores are the pure tests on TTA10 engines performance. Each result is a mean value, computed according average search (pure retrieval) time on the whole database of images. According to the results presented in the experimental part of a paper we made some extrapolations for the large databases (100 000 000 – 1 000 000 000 of images) in order to predict possible search time and memory size for one workstation solution. We suppose that the search time would be less than 5 milliseconds per one image (time to compute visual descriptors is not included), and memory size would be near 3Gb. Using parallel architecture cluster it's possible to decrease memory consumption proportionally to the number of clusters. The comparison of retrieval results for different sets reveals that the primary/secondary ratio equal to 4/96 is competitive for the retrieval accuracy; meanwhile it's the most preferable configuration due to low memory consumption.

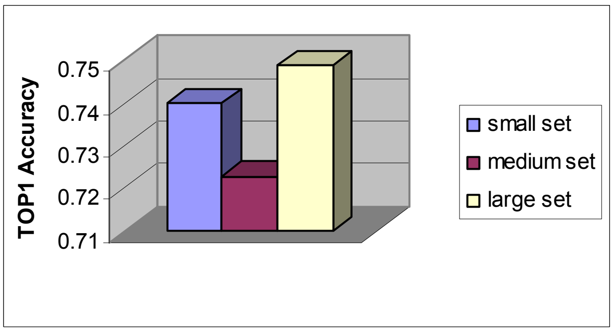

Figure – 10 *TOP1 accuracy: various test sets, pri/sec is 8/92 with totally 74610 images in the large set*

The figure 10 demonstrates that performance in terms of an accuracy of TTA10 is highly stable in general for variety of database sizes. The circumstances under which the evidence combination strategies provide the most benefit is in the case of primary/secondary ratio (4/96;8/92).

4 Conclusions

We have evaluated the TTA10 algorithm for indexing/retrieval of images in large databases. Algorithm shows logarithmic complexity, i.e. tree structure is highly balanced in general. The results on retrieval accuracy are highly stable for different size of images database, therefore the TTA10 method is reliable and accurate. Inherited paralleled indexing/retrieval algorithm architecture makes possible to utilize computer cluster for high-end databases with billions of images. We presented a dynamic sub-tree monitoring and reorganization, which allows to add the new images into database in real-time without rebuild of entire tree-structured index. The usage of optimized distances on secondary tessellation can further improve classification accuracy. N-stage indexing will improve systems generalization.

References

[1] Y. Wang, Z. Liu, and J-C. Huang, "Multimedia Content Analysis Using both Audio and Visual Clues", IEEE SPM, pp. 12-36, November 2000.

[2] Y. Chahir, Liming Chen, "Searching Images on the Basis of Color Homogeneous Objects and their Spatial Relationship", pp. 302-326, JVCIR 2000.

[3] S-F. Chang, J. R. Smith, M. Beigi, and A. B. Benitez, "Visual information retrieval from large Distributed on-line Repositories", Communications of the ACM, Vol.40, no.12, pp. 63-71, Dec. 1997.

[4] D. Swets, J. Weng, "Hierarchical Discriminant Analysis for Image Retrieval", IEEE Transactions on Pattern Analysis And Machine intelligence, vol. 21, no.5, pp. 386-401, May 1999.

[5] R. F. Sproull, "Refinements to nearest-neighbor searching in k-dimensional trees", Algorithmica, vol. 6, pp. 579-589, 1991.

[6] M. Jones, J. Rehg, "Statistical Color Models with Applications to Skin Detection.", CRL TR98/11.