

## **Content-based Multimedia Data Retrieval on Heterogeneous System Environment**

Sanan Srakaew, Nikitas A. Alexandridis, Punpiti Piamsa-nga, George Blankenship

Department of Electrical Engineering and Computer Science,

George Washington University,

Washington D.C. 20052.

{srakaew, alexan, punpiti, blankeng}@seas.gwu.edu

### **Abstract**

In this paper, we propose a static data partitioning scheme for content-based multimedia data retrieval using a heterogeneous cluster system. Multimedia data is represented by unified k-tree structure of k-dimensional(k-d) signals proposed in [6]. Each dimension of k-d data is separated into small blocks and then formed into a hierarchical multidimensional tree structure, called a k-tree. The parallel version of k-tree model was introduced in [7]. The previous experimental results show the huge reduction of retrieval time on a cluster of homogeneous workstations. In this paper, we extend our parallel model to a heterogeneous cluster system environment, by taking into consideration the system characteristics such as computational time, input/output time, available storage, and communication latency. The experiments of the model with load balancing shows a significant reduction of retrieval time while maintaining the quality of perceptual results.

*Keywords:* Data partitioning, Image databases, Content-based retrievals.

## **1 Introduction**

Multimedia databases have become more important since the demand for multimedia information (such as text, audio, image and video) has increased. Currently content-based retrieval of multimedia data is being actively researched. However, content-based retrieval of multimedia data encounters three major difficulties. First, the content is subjective; this needs a powerful set of search facilities including keywords, sounds, color, texture, spatial information and motion. Second, if a method or processing technique is designed and developed for one type of data or feature, it's usually not appropriate for others. For instance, a technique designed for indexing audio data may not be usable for image data; or, a technique developed for a color feature may not be useful for a texture feature in image and video data. Third, the usual huge size of multimedia data requires an exhaustive search. A similarity search is desirable for a multimedia database since exactly matched retrieval cannot be applied. For example, if a picture of a house is

used as a query to an image database, we expect to retrieve pictures that contain similar houses in them. The comparison is not pixel by pixel between a query and the records in a database; but rather, closeness to the query. Similarity matching needs the computation of the distance between a query and each record in the database; the best match is chosen from the data set with the smallest distances. To solve these three problems, we use a mathematical model to represent the features; a k-tree model to represent the data structures of the multimedia data; and exploit parallelism to reduce the retrieval time.

In this paper, color and texture are the features of interest; they represent the subjective information of the multimedia data. We use a normalization technique to generate the indices. The domain of a feature is reduced to a set of selected values from a universe of potential values for the feature. We use an identification number for each element in the reduced set [7]. When data is inserted into the system, it is converted to the selected domain. The feature is represented by a histogram. For color feature, a few colors are picked from the whole infinite universe of colors. A finite number indexes each color. The color feature of an image or a video is represented by a histogram using the indexed color. For texture feature, we selected a set of textures and assigned an identification number to each texture. The feature of a texture is represented by the histogram of texture identification, which is the same method that was used for the color feature. The comparison of two features is based upon the distance between the histograms that define the features.

To reduce the response time, one may use a parallel model of a homogeneous system to perform a content-based multimedia retrieval. The experimental results were very positive in both qualitative and quantitative metrics. However, in practical, we do not have dedicated machines that always have the same configurations. The homogeneous model may be not used efficiently enough in the real-life heterogeneous environment. In this paper, we investigate a data partitioning scheme for multimedia database retrieval on a heterogeneous cluster system. We use system characteristics, such as processor speed, input/output time, and available storage, to partition data among the processors in the systems. Our computer system environment is composed of Sun Sparc and

Pentium-Linux machines, which are connected via a 10Mbit local area network. We evaluate the model by comparing the retrieval results with the previous homogeneous parallel retrieval. The experimental results show the heterogeneous model produces a significant reduction of the retrieval time of an image from a 30,000-record image database.

This paper is organized as follow. Section 2 describes our k-tree parallel model. Section 3 has the details of our cluster system environment and its heterogeneity. The experiment and its results are described in Section 4 The last section concludes our works and proposes future directions.

## 2 The K-Tree Parallel Model

A *k*-tree is a directed graph; each node has  $2^k$  incoming edges and one outgoing edge with a balanced structure [6]. A *k*-tree is a *binary tree* for 1-dimensional data and a *quadtree* for 2-dimensional data. Exploiting a *k*-tree brings three main benefits. First, the *k*-tree holds the information of spatio-temporal data on the tree structure itself. It reduces distance computation time to a comparison between two tree nodes. Second, a *k*-tree can accelerate multiresolution processing by calculating small, global information first and then large, local information when precise resolution is needed. Third, the data on a *k*-tree is unified since only the degree of the tree changes, while the processing algorithm and data structure remain invariant. Therefore, an algorithm for a particular type of feature can be reused for a feature of another media type.

Content-based retrieval of multidimensional signals is done by comparing features extracted from the input query with features extracted from every record in the database. The features of a multidimensional signal are subjective information. They are characteristics that are used to distinguish one signal from others. A 2-dimensional signal, such as an image, is characterized by features such as color, texture, and intensity. The basic algorithms for the searching of data in each of the different domains are quite similar. A matching search requires that the index key (defining feature) be unique and matched to the query. Exactly matched searching requires exhaustive comparisons that are inefficient and unsuitable for multidimensional signals; similarity searching is more appropriate. A similarity-search reorders the database by distance between each record and the query; the result is selected from the ranking.

Multimedia data retrieval requires similarity searching; exact matching, which is used in conventional database, is not appropriate for this type of application. Similarity searching generally can be done in two steps; 1) finding distances between a query and all records in the database

and then 2) sorting the distances and returning the results – the set of data items that have shortest distances. We also call this process “ranking.” The details of regular weighted ranking are discussed in [7]. In Figure 1, we show a parallel searching using multi-feature scheme. Prior to the search the database is distributed among processors. Each processor performs the comparison between the query and its portion of the database. The search results based on those features are sorted in parallel to create the final ranking.

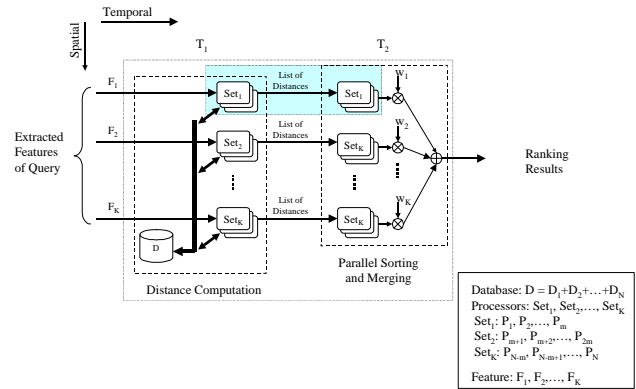


Figure 1: Parallel model for multi-feature searching

## 3 Cluster System Environment

In this section, we give the details of our experimental platforms, evaluated multimedia database systems, and our proposed load-balancing algorithm for the *k*-tree model.

### 3.1 Computational platform

A cluster system environment plays an important role in a distributed multimedia database. In this paper, we establish a uniform framework for content-based multimedia data retrieval on a heterogeneous cluster system environment. In our experiment, one of the workstation serves as a host while others are assigned to be slave nodes. The system is composed of seven machines, which include two models of Sun workstations and Pentium/Linux PCs connected via a 10Mbps hub and the Message Passing Interface (MPI) library is used as a message-passing interconnection mechanism [8].

### 3.2 Multimedia database system

In this paper, we use an image database using histogram-based features as indices. Two types of histogram-based feature (colors and textures) have been examined. Before beginning the extraction of features, all

images are normalized, scaled down to 128x128 pixels. The color feature extraction is performed in two steps. The first step transforms the number of colors of the scaled images to a pre-selected 166-color set [5]. The second step stores the transformed image in a quad tree structure. Texture feature extraction requires three steps. The first step transforms the 64 blocks of 16x16 pixels in to 64 sets of wavelet data using a Quadratic Mirror Filter (QMF) (2 iterations, 7 sub-bands) [7]. Each wavelet data produces seven subbands of means and variances; i.e. a 14-element vector. In the second step, the texture vectors are then compared to 162-reference textures from VisTex [7] in the known texture table to generate 64 texture indices representing textures for blocked data. The third step constructs and stores the texture features in a quad tree structure.

The steps of the quad tree generation are the same for both features. The transformed color images and texture-identification (texture-id) matrices are mapped onto the leaves of a quad tree structure. The leaves represent a single pixel of the normalized image. Histograms of the leaves, which share the same parent nodes, are summed and the results are stored at their parent nodes. The process continues iteratively for each level until the root has been reached.

### 3.3 Data Partitioning and Load Balancing

To exploit heterogeneous parallelism, we use task and machine characteristics to decide which processors the tasks should be allocated to. The heterogeneity in the task level is the differences of the searching into the feature indices and the heterogeneity of the machines includes processor speed, input/output time, available storage, and communication latency. The general idea is to give more data to a machine whose estimated execution time is smallest. Data given to a particular machine can be locally stored or remotely accessed via a faster network link. For example, suppose a system consists of two machines; one is twice faster than another. Thus, the faster workstation should have two-third of database and the slower one should have the one-third portion. However, if the faster machine doesn't have enough disk space to store all two-third portion, parts of database needed on this workstation can be remotely kept on another one. The size of remotely kept data is dependent of average link latency between these two machines. We can generalize the load distribution in a mathematical model as follows.

Let  $S$  be a set of heterogeneous machines;  $S = \{M_1, M_2, \dots, M_m\}$ ;  $m$  be the number of machines;  $M_i(C_i, S_i, L_i)$  be a machine described by computational power  $C_i$ , an available storage  $S_i$ , and estimated network bandwidths

$L_i$ ;  $L_i = \{\lambda_{ij}, \forall i \neq j\}$ .  $\lambda_{ij}$  is an average network latency between  $M_i$  and  $M_j$  for all  $i \neq j$ .

Let  $B$  a data base size and  $E_i$  be an estimated execution time of task  $T$  on  $M_i$

$$E_i = C_i + R_i$$

where  $C_i$  is an *estimated computational time* and

$$R_i \text{ is an } \textit{estimated IO time}$$

Let  $S_{ij}$  be a portion of  $B$  on  $M_j$  that  $E_i$  needs to access, therefore

$$E_i = C_i * \sum [p_{ij}] + (R_i / \lambda_{ii}) * \sum [p_{ij} * \lambda_{ij}]$$

Our goal is to minimize  $E_i$  subject to the following constraints:

$$\begin{aligned} \sum [p_{ij}] &= 1 \\ \sum [p_{ij}] &\leq S_i / B, \quad 1 \leq j \leq m \\ E_i &= E_j, \quad \forall i \neq j \end{aligned}$$

## 4 Experiments and Results

The cluster environment consists of seven Unix-based machines; two Pentium/Linux-based PCs, two Sun Sparc-20 workstations, and three Sun Ultrasparc I workstations. One of Sun Ultrasparc is designated as the host. The MPI library is used as the interconnection mechanism. We use comparative computational power to classify workstation types. The computational power depends on processor speed, input/output time, cache size, and so on. We benchmark a workstation by running several similarity searchings on a database of 5,000 images using single- and multi- feature algorithms. The average values of computational time, I/O time, and network latency are used to rank the computational power of workstations. The computational power ratio of a Sun Ultrasparc I to a Pentium II PC to a Sun Sparc-20 is 3:2:1.8.

In this experiment, image retrieval is performed using two features; color and texture. The extracted features are derived from database images of 128x128 pixels; each is evenly divided into 64 blocks. The *quadtree* of the histograms for each image is made up of 3 levels; 64 leaf nodes. We perform two data partitioning schemes based on the database of 30,000 images. In the first scheme, the database of color and texture histograms is evenly divided among workstations, without considering the heterogeneity of cluster environment. In the second scheme, data partitioning is based on the ratio of computational power of the workstations and available storage.

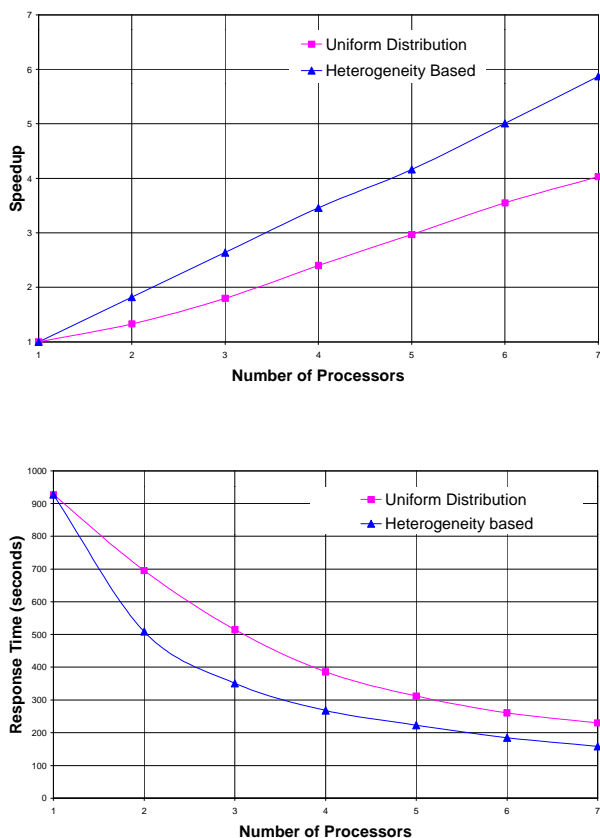


Figure 2: (a) Response time; (b) Speedups.

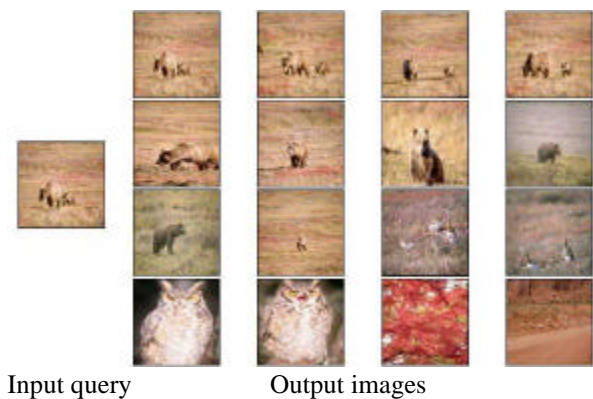


Figure 3: Perception results.

The results shown in Figure 2(a) depicts the response time of the system as a function of the number of processors used to perform the ranking; the selection features are both color and texture. Figure 2(b) shows speedup as a function of the number of processors. As the number of processors used to perform the computation

increases, the computation time decreases significantly. Moreover, the data partitioning based on heterogeneity information achieves a higher speedup than an even distribution approach. Figure 3 depicts the top-twenty output images on the sorted list when both color and texture are used as selection features.

## 5 Summary

We introduced a parallel model for multimedia database content-based retrievals on a cluster of heterogeneous workstations. The model allows the extension of the system for the new types of data, new techniques, and new types of interest contents with less effort. The experimental results show that heterogeneous processing with load balancing can reduce retrieval time over a homogeneous approach. Data partitioning based on system heterogeneity achieves a better response time in comparison with uniform distribution of database over a cluster of workstations. Our future work will focus mainly on classification technique based on multi-resolution structure of the k-tree at different levels. Some load balancing and process migration techniques are also in our future work.

## References

- [1] P. Chalermwat, N. Alexandridis, P. Piamsa-nga, and M. O'Connell, Parallel image processing on heterogeneous computing network systems, *International Conference on Image Processing*, 1996.
- [2] T. El-Ghazawi, P. Chalermwat, P. Piamsa-nga, A. Ozkaya, N. Speciale, and D. Wilson, PACET: PC-parallel architecture for cost-efficient telemetry processing, *IEEE Aerospace Conference*, 1998.
- [3] V. Gudivada and V. Raghavan, "Special issue on content-based image retrieval systems," in *IEEE Computers*, Vol. 28, No. 9, September 1995.
- [4] Z. Kemp, "Multimedia and spatial information systems," *IEEE Multimedia*, 2(4), 1995.
- [5] J. R. Smith and S.-F. Chang, SaFe: "A General Framework for Integrated Spatial and Feature Image Search," *IEEE Workshop on Multimedia Signal Processing*, 1997.
- [6] P. Piamsa-nga, N. Alexandridis, G. Blankenship, G. Papakonstantinou, P. Tsanakas, and S. Tzafestas, "A Unified Model for Multimedia Retrieval by Content," *International Conference on Computer and Their Application (CATA98)*, 1998.
- [7] P. Piamsa-nga, N. Alexandridis, S. Srakaew, and G. Blankenship, "A parallel algorithm for multi-feature content-based multimedia retrieval," *Seventh International Conference on Intelligent Systems (ICIS98)*, Paris, France, July 1-3, 1998.
- [8] S. Srakaew, N. A. Alexandridis, P. Piamsa-nga, and G. Blankenship, "A parallel model for multimedia retrieval based on multidimensional signal structure," in *International workshop on systems, signal and image processing (IWSSIP98)*, Zagreb, Croatia, June 3-5, 1998.