Multimedia Analysis and Retrieval System (MARS) Project *

Tom Huang † Sharad Mehrotra ‡ Kannan Ramchandran §

Abstract

To address the emerging needs of applications that require access to and retrieval of multimedia objects, we have started a *Multimedia Analysis and Retrieval Systems* (MARS) project at the University of Illinois. The project brings together researchers interested in the fields of computer vision, compression, information management and database systems with the singular goal of developing an effective multimedia database management system. As a first step towards the project, we have designed and implemented an image retrieval system. This paper describes the novel approaches towards image segmentation, representation, browsing, and retrieval supported by the developed system. Also described are the directions of future research we are pursuing as part of the MARS project.

Keywords: multimedia systems, content-based retrieval, image database, segmentation, information retrieval, wavelet based compression.

*This work was supported in part by the NSF/DARPA/NASA Digital Library Initiative Program under Cooperative Agreement 94-11318, in part by the U.S. Army Research Laboratory under Cooperative Agreement No. DAAL01-96-2-0003, in part by NASA under the Cooperative Agreement No. NASANAG 1-613, and in part by the University of Illinois Research Board.

†Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign

‡Department of Computer Science, University of Illinois at Urbana-Champaign

§Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign

1 Introduction

Advances in high performance computing, communication, and storage technologies as well as emerging large scale multimedia applications has made multimedia data management one of the most challenging and important directions of research in computer science. Such systems will support visual data as "first-class" objects that are capable of being stored and retrieved based on their rich internal contents. Applications of multimedia databases include among others:

- Government and commercial uses of remote sensing images, satellite images, air photos,etc;
- Digital libraries, including digital catalogs, product brochures, training and education, broadcast and entertainment, etc;
- Medical databases, such as X-rays, MRI,etc;
- Special-purpose databases, e.g. face/fingerprint databases for security, business directories, maps, etc.

While current technology allows generation, scanning, transmission, and storage of large numbers of digital images, video and audio, existing practice of indexing, access and retrieval of visual data are still very primitive. Most current systems rely on manual extraction of content information from images. Such information is stored using text annotations and indexing and retrieval is then performed using these annotations. Although useful in some domains, such techniques are severely limited since manual indexing is inherently not scalable and, furthermore, textual descriptors are inadequate for describing many important features based on which users wish to retrieve visual data (e.g., color, texture, shape, and layout). Also, textual description are ineffective in supporting unanticipated user queries.

Development of multimedia database management systems requires an integrated research effort in the fields of image analysis, computer vision, information retrieval and database management. Traditionally, these research areas have been studied in isolation with little or no interaction among the respective research communities. Image analysis and computer vision researchers have developed effective algorithms for image representation and segmentation. However, incorporation of these algorithms into the data management system in order to support effective retrieval is largely an open problem. On the other hand, research on information retrieval has focussed on developing effective retrieval techniques to search for information relevant to the users' queries. Effectiveness is measured using the *precision* of the information retrieved (i.e., how relevant is the retrieved information to the user), and the *recall* (i.e., how much of the relevant information present in the database was retrieved) (Salton and McGill, 1983) Efficient processing of user queries, as well as, support for concurrent operations which are important for scalability have been relatively ignored. Furthermore, research has primarily focussed on textual data. Finally, database management research has concentrated on efficiency of storage and retrieval, as well as, support for concurrent users and distributed processing. However, the techniques have been developed in the context of simple record-oriented data and little has been done to extend the techniques to either textual, image or multimedia data.

To address the challenges in building an effective multimedia database system, we have started the *Multimedia Analysis and Retrieval System* (MARS) project. MARS brings together a research team with interest in image analysis, coding, information retrieval and database management. As part of the MARS project, we are addressing many research challenges including automatic segmentation and feature extraction, image representation and compression techniques suitable for browsing and retrieval, indexing and content-based retrieval, efficient query processing, support for concurrent operations, and techniques for seamless integration of the multimedia databases into the organizations' information infrastructure. As a first step, we have developed a prototype image retrieval system (referred to as MARS/IRS) that supports

content-based retrieval over a testbed consisting of a set of images of paintings and photographs provided by the Getty foundation. This paper describes the design and implementation of MARS/IRS including novel techniques for segmentation, representation, browsing and retrieval developed. We also discuss directions of future research we are pursuing as part of the MARS project.

Many of the research topics being pursued in the MARS projects are also being addressed by other research teams both in the industry and the academia. One project related in scope is the *Query by Image Content* (QBIC) system being developed at IBM Almaden Research Center (Faloutsos etal. 1993; Flickner etal 1995). The QBIC system supports queries based on color, texture, sketch, and layout of images. Another important related project is the ADVENT system developed at the Columbia University (Smith and Chang, 1994, 1995, 1996; Wang 1995). Their main research focus is color/texture region extraction in both the uncompressed domain and the compressed domain. The color set concept is used in their color region extraction approach to make it faster and more robust. Its texture region extraction is based on the features (means and variances) extracted from Wavelet sub-bands[8]. Instead of decompressing the existing compressed images to obtain the texture features, they perform texture feature extraction in the compressed domain, such as Discrete Cosine Transformation (DCT), Discrete Wavelet Transformation (DWT). Other projects related to ours include **Photobook** in MIT (Pentland, Picard, and Sclaroff 1995), **Alexandria** in UCSB (Manjunath and Ma, 1995), as well as, **DLI** projects at Stanford, Berkeley, CMU, and MU (Schartz and Chen, 1996) which are working on low level feature extraction (image and video), feature representation, concept mapping, and database architecture.

2 MARS Image Retrieval System

MARS/IRS is a simple prototype image retrieval system that supports similarity and content-based retrieval of images based on a combination of their color, texture, shape and layout properties. The distinguishing feature of the current implementation includes novel approach towards segmentation, shape representation, support for complex content-based queries, as well as compression techniques to support effective browsing of images. In this section, we describe the current implementation of MARS/IRS.

2.1 System Architecture

The major components of MARS/IRS are shown in Figure 1 and are discussed below.

- **User interface:** written using Java applets and accessible over the World Wide Web using the Netscape browser. The user interface allows users to graphically pose content-based and similarity queries over images. Using the interface, a user can specify queries to retrieve images based on a single property or a combination of properties. For example, a user can retrieve images similar in color to an input query image. A more complex query is to retrieve images that are similar in color to an input image I_1 and contain a shape similar to a specified shape in image I_2. The interface also allows users can combine image properties as well as text annotations (e.g., name of the creator, title of a painting, etc.) in specifying queries. The user interface is accessible over the WWW at PURL http://quirk.ifp.uiuc.edu/mars/mars.html.
- **Image Indexer:** The image indexer takes as input an image as well as its text annotation. With the help of the image analyzer it extracts image properties (e.g., color, texture, shape). Furthermore, it extracts certain salient textual properties (e.g., name of the artist, subject of the painting, etc.) and stores these properties into the feature database.

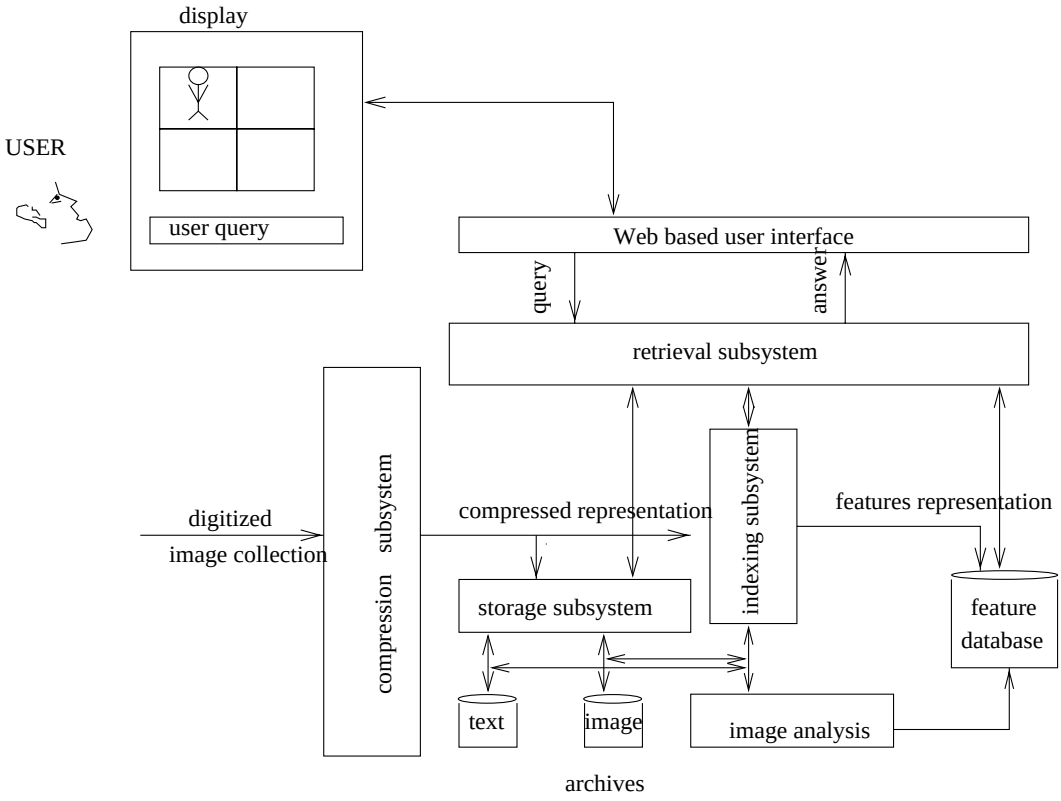

Figure 1: The MARS system Components

- **Image Analyzer:** The image analyzer extracts salient image properties like the global color and texture as well as the shape. The global color is represented using a color histogram over the HSV space. At each image pixel three texture features– coarseness, contrast, and directionality — are computed and the set of feature vectors forms a 3-D global texture histogram. Furthermore, images are segmented and the shape features of the objects in the image is represented using a modified Fourier Descriptor of the object boundary.
- **Feature Database:** An image in the feature database is represented using its image as well as textual properties. An image consists of global color histogram, a texture histogram, shape features, textual features like name of artist, subject of painting, etc., as well as, color and texture layout properties. The feature database is currently implemented using POSTGRES (Stonebraker and Kemnitz, 1991). Furthermore, users can associate a full-text description with the images.
- **Query Processor:** The query processor is written on top of POSTGRES in C. It takes the query specified at the user interface, evaluates the query using the feature database, and returns to the user images that are best matches to the input query. The query language supported allows users to pose complex queries that are composed using image as well as textual properties. First, the query processor ranks the images based on individual properties. It then combines the ranking on individual properties to determine the overall ranking of the images based on the complex query. Techniques are developed to efficiently identify the best N matches without requiring that every image be ranked based on each

property.

As mentioned previously, currently MARS/IRS uses as a testbed a set of images of paintings and photographs of artifacts made available to us by the Getty Museum Foundation.

2.2 Image Representation

In MARS/IRS, an image I consists of a set of *global* properties as well as a set of objects $\{O_1^I, O_2^I, \ldots, O_n^I\}$. Global properties are either:

- fixed descriptors like the artist's name, title, museum to which the image belongs.
- free-text description of the image.
- low-level image properties like color, texture, and layout.

Objects within an image are identified using automatic segmentation (described later in the section), and associated with each objects are *local* properties which could include description of shape, average color, texture, centroid, area, as well as textual annotations. This section describes the representation of the low-level image features used in modeling an image.

Color: While color features could be represented in many color spaces, we use the hue, saturation, value (HSV) color space since it approximates a perceptually uniform color space, making it easier for the user to specify colors. The global color histogram of an image is computed and stored. *Histogram intersection* method is used to compare the overall color content of an image with the colors specified in the user query. With respect to changes in image background colors, the histogram intersection similarity measure is more robust than the Euclidean histogram distance or matrix-weighted histogram distance. Using the histogram intersection method, a user may retrieve images in a database that contain a specific color or set of colors. For example, a user may retrieve all images that contain red and green, but no blue.

Color Layout: While the color histogram is useful for queries on the relative amount of each color in an image; it is not useful for queries on the spatial location of colors. For example, it is not possible to retrieve all images that contain a red region above and to the right of a large blue region based solely on the color histogram. Such queries can be answered correctly only if an image can be accurately segmented into regions of different color which is difficult to achieve. But for queries relating to simple spatial relationships between colors, a relatively non-ideal segmentation may still be sufficient.

To represent spatial arrangement of colors in an image, we do a simple k-means clustering on the HSV histogram of an image to produce a rough segmentation. For each region in the segmentation, we store the following information for color indexing: centroid, area, eccentricity, average color, and maximum bounding rectangle. Images will be searched by comparing the relative locations and colors of the indexed regions to see if they matched the color layout query.

Texture: Texture is another important feature of images, and researchers have done a great deal of work in this area. We have implemented texture measures based on *coarseness*, *contrast*, and *directionality*, which are generally considered to be fairly good measures for texture[1]. At each image pixel we compute these three texture features from the pixel's local neighborhood. The set of feature vectors from all image pixels forms the 3-D global texture histogram. We compute these measures for each image and use a weighted Euclidean distance function as

[1] Psychophysical studies suggest that the human visual system uses these three measures as primary features for texture discrimination

the matching criteria. The method described by Hideyuki Tamura (used by QBIC) uses 3 scalar measures, and does not consider the relationship between texture components. Our method includes this texture information and thus returns better matches (i.e. the textures are perceptually more similar).

Shape: Although shape is a very important feature that a human can easily extract from an image, reliable automatic extraction and representation of shapes is a challenging open problem in computer vision.

Some simple shape features are the perimeter, area, number of holes, eccentricity, symmetry, etc. Although these features easy to compute, they usually return too many false positives to be useful for content-based retrieval, thus they are excluded from our discussion.

Advanced methods that can represent more complex shapes fall into two categories. Region-based methods are the first category. These methods are essentially the Moment-Invariants Methods (MIM). The disadvantage of the MIM is its high computational cost (features are computed using the *entire* region including interior pixels), and low discriminatory power. The descriptors are also tend to return too many false positives.

Boundary-based methods are the second category, which include the Turning Angle Method (TAM) and Fourier Descriptors (FD). These methods provide a much more complete description of shape than MIM; however, they suffer the disadvantage of being dependent on the starting point of the shape contour, and they can recover parameters (rotation, scale, starting point) only by solving a non-linear optimization problem, which is not feasible in a real-time content-based retrieval system. Furthermore, to the extent of our knowledge, no research has been done on how to deal with the *spatial discretization* problem when using these methods.

We proposed the Modified Fourier Descriptor (MFD) (Rui, She, and Huang, 1996b), which satisfies the following four conditions:

1. Robustness to Transformation – the representation must be invariant to translation, rotation, and scaling of shapes, as well as the starting point used in defining the boundary sequence.
2. Robustness to Noise – shape boundaries often contain local irregularities due to image noise. More importantly, spatial discretization introduces distortion along the entire boundary. The representation must be robust to these types of noise.
3. Feature extraction Efficiency – feature vectors should be computed efficiently.
4. Feature matching Efficiency – since matching is done on-line, the distance metric must require a *very* small computational cost.

2.3 Image Segmentation

Our image segmentation is based on clustering and grouping in Spatial-Color-Texture space. For a typical natural image, there is a high number of different colors and textures. C-means clustering is one way to reduce the complexity while retaining salient color and texture features.

1. Randomly pick c starting points in the Color-Texture space as the initial means.
2. Cluster each point as belonging to the nearest neighbor mean.
3. Compute the new mean for each cluster.
4. Repeat 2 and 3 until all the clusters converge (i.e. when the number of pixels and mean value of each cluster does not change).

After this procedure, we have c clusters, each of which may corresponds to a set of image pixels. We define *cluster* as a natural group which has similar features of interest. The image pixels corresponding to a particular cluster may or *may not* be spatially contiguous. We define a *region* as one of the spatially connected regions corresponding to a cluster.

The c-means clustering generally produces regions of various sizes; some of the regions are very small (containing only a few pixels). We consider these regions as speckle noise and set a minimum region size threshold to filter out these small regions. The deleted regions are merged with the largest neighboring region.

After c-means clustering we have c clusters, each corresponding to several spatial regions. The next step is to extract the desired object from the regions.

One way to do this is to define a threshold in Color-Texture space. If a region's Color-Texture feature is above the threshold, then this region is considered as the object; otherwise, considered as the background. One obvious disadvantage of this thresholding method is that the threshold is image-dependent. We propose an attraction based grouping method (ABGM) to overcome this disadvantage (Rui, She, and Huang 1996a). The method is motivated by the way the human visual system might do the grouping.

As defined in physics,

$$F_{12} = G\frac{M_1 M_2}{d^2}$$

reflects how large the attraction is between the two masses M_1 and M_2 when they are of distance d. In ABGM, we use the similar concept, but now M_1 and M_2 are the size of the two regions, and d is the Euclidean distance between the two regions in 6-D Spatial-Color-Texture space.

The ABGM method is described as follows:

1. Choose attractor region A_i's from the clustered regions according to the knowledge of the application at hand.
2. Randomly choose an unlabeled region R_j. Find the attractions F_{ij} between A_i and R_j.
3. Associate region R_j with the attractor A_i that has the largest attraction to R_j.
4. Repeat steps 2 and 3 until all the regions are labeled.
5. Form the output segmentation by choosing the attractor of interest and its associated regions.

Note that if the attractor is bigger or closer (in 6-D space) to a unlabeled region, its attraction will be larger and thus the unlabeled region will be labeled to this attractor with higher probability. This is what a human visual system might do in the labeling process.

2.4 User Interface and Query Language

The user interface of MARS/IRS allows users to browse images sequentially (or in a random order), as well as, to graphically pose content-based queries over the database of images. Queries supported are a boolean combination of *query terms*. The semantics of the query is to retrieve images ranked on the degree to which the image satisfies the input query. A query term is either *simple* or *complex*. A simple term corresponds to textual annotations or image properties like color and texture. For example, a query:

$$\text{containing\_color(color identifier)} \wedge \text{similar\_texture\_to\_image(image id=4000)}$$

is a boolean combination of the following two query terms, combined using a conjunction operator:

- containing_color(color identifier), and
- similar_texture_to_image(image id=4000).

The first term refers to images that contain a given color (possibly chosen from a color pallet). The second refers to images whose texture matches the texture of the image with the identifier 4000. The system will retrieve images containing the the specified color that also have a texture similar to the image 4000.

The user interface supports many ways in which users can specify query terms. Colors can be chosen from a color pallet or could be chosen from the images that are currently being displayed in the MARS/IRS display window. To specify a color using an image, a user first loads the image from the display into the workspace (by clicking on the image). (S)he can then choose either the global color of the loaded image as a query term (in which case the query term specifies retrieval of images whose global color histogram is similar to that of the loaded image), or, alternatively, the user can choose the average color of some object[2] within the image as a query term by clicking on the object (the objects within an image are highlighted when the image is loaded into the workspace for this purpose). Mechanisms similar to those used for specifying color query can be used for specifying texture query terms as well. Furthermore, MARS/IRS supports mechanisms for both specifying color layout query terms as well as selecting a color layout similar to the layout of a given image.

In contrast to the simple query terms, a complex query term is of the form:

contains_object(object description query)

The complex query term refers to images that contain an object that matches the object description query. The object description query may itself be a boolean combination of both image based as well as textual features associated with the objects. The user interface also supports graphical mechanisms for composing object description queries.

The query mechanism supported by MARS/IRS provides a versatile tool for content-based retrieval. Using the boolean operators, users can form very complex queries. One special complex query is the *similarity query* when a user wishes to retrieve all the images similar to a given input image. Such a query is interpreted to mean images similar to the input image based on *all* the features and objects associated with the input image (obviously, such queries are reasonably inefficient). We are currently exploring information retrieval techniques including the query refinement mechanism of relevance feedback (Salton and McGill, 1983) to meaningfully answer similarity queries effectively and efficiently.

2.5 Query Processing

A query processor takes a query and retrieves the best N images that satisfy the query. Associated with the query is a *query tree*. Leaf nodes of the tree correspond to simple query terms based on a single property — e.g., global color similar to that of an input image I_1. Internal nodes in the tree correspond to boolean operators – **and**, **or**, and **not** as well as complex query terms corresponding to objects contained in the image. The query tree is then evaluated as a pipeline from the leaf to the root. The leaf node n_j return a ranked list of $\langle I, sim(I, Q_{n_j})\rangle$ to its parent, where I is an image and $sim(I, Q_{n_j})$ is a measure of match between the image I and the query represented by the leaf node n_j. For example, a leaf node n_j corresponding to the query term representing the global color of an image I' returns a ranked list of $\langle I, sim(I, Q_{n_j}\rangle$, where $sim(I, Q_{n_j})$ is the measure of the intersection of color histograms corresponding to images I and I'.

The internal nodes n_p, receive such ranked lists from each child and combine then to compute a ranked lists of $\langle I, sim(I, Q_{n_p})\rangle$, where $sim(I, Q_{n_p})$ is a measure of similarity between the image I and the query represented by the internal node n_p. This list is then input to the higher nodes

[2]Identified using the segmentation method described in Section 2.3.

in the pipeline which use it to compute their best matches. To rank the images according to the query represented by parent nodes, first the similarity measures associated with child nodes is *normalized*. Normalized similarity measures of different child nodes are then used to rank the images based on the degree of match to the query represented by the parent node. In our current implementation, a simple approach to normalization and ranking of images is adopted. Let an internal node n_p consist of child nodes $n_1, n_2, \ldots, n_m$. The normalized similarity of an image I to the query corresponding to the child node n_j (represented by $\bar{sim}\ (I, n_j)$) is taken to be the inverse of the rank of I based on its similarity to the query represented by node n_j (notice that the range of the normalized similarity lies between 0 and 1). The similarity of the image I to the query represented by node n_p is computed as follows:

- $sim(I, Q_{n_p}) = min(\bar{sim}\ (I, Q_{n_1}), \bar{sim}\ (I, Q_{n_2}), \ldots, \bar{sim}\ (I, Q_{n_m}))$, where $Q_{n_p} = Q_{n_1} \wedge Q_{n_2} \wedge \cdots Q_{n_m}$)
- $sim(I, Q_{n_p}) = max(\bar{sim}\ (I, Q_{n_1}), \bar{sim}\ (I, Q_{n_2}), \ldots, \bar{sim}\ (I, Q_{n_m}))$, where $Q_{n_p} = Q_{n_1} \vee Q_{n_2} \vee \cdots Q_{n_m}$)

An advantage of such a simple normalization and ranking algorithm is that it can be implemented very efficiently and does not require that every image be ranked based on each property in order to compute the best N matches. However, the resulting retrieval is not very effective. We are currently exploring usage of more complex retrieval models (e.g., vector space models, inference network retrieval model) used in information retrieval to improve retrieval effectiveness. Effective and efficient retrieval techniques for feature-based queries is one of our primary research concerns in the near future.

2.6 Representation and Compression for Fast Browsing Using Wavelets

Due to the obvious volume of data being stored and processed in the database, it is important to address efficient ways to represent and compress this data. A key goal here is not just to achieve a substantial compression ratio in order to reduce the amount of storage needed, but even more important to do so in a framework that supports some of the important database tasks like browsing and object-based retrieval, i.e. to have a representation data-structure that lends itself to these tasks without needing to completely decompress the data. Toward this end, we propose a novel representation and compression data-structure that is based on wavelets. Wavelets represent a mathematical tool based on multiresolution analysis that permit a natural decomposition of a signal or image into a hierarchy of increasing resolutions, thereby making them very suitable candidates for browsing applications.

Since their introduction, wavelets have become increasingly popular within the image coding community as an effective decorrelating transform to be used in the de-facto standard architecture of lossy coders, consisting of a linear transform followed by a quantization stage, and final entropy coding of the quantized symbol stream. Although initially the performance of wavelet based coders was only marginally better than that of previous existing subband coders, with the introduction of Shapiro's embedded zerotree wavelet (EZW) coder that is based on the Zerotree data-structure, an entire new avenue of research was started, with coders exploiting in different forms the fact that even after decorrelation, in significant structure remains in the subbands. Careful studies of the statistics of image subbands led to many improvements over the standard Zerotree algorithm; however, this increased efficiency in coding often came at the expense of high computational complexity.

There are a number of ways to go about the complexity problem. One possibility, very appealing for the Image Databases application because of the simplicity with which transform domain data is represented, is that of fixing the quantization strategy to something reasonable

(e.g., choose a single uniform quantizer for *all* subbands), and optimizing the entropy coder instead. Probably one of the simplest techniques of lossless data compression is that of runlengths. Zero runlengths have been very successfully applied a few years ago to the JPEG standard; surprisingly, none of the existing high performance wavelet based coders make use of them. To test how useful one such representation can be for our purposes, we took 2 typical test images, and computed the entropy of such a representation:

Image	Lena		Barbara	
Distortion (PSNR)	33.58	36.67	27.32	30.94
Entropy (bpp)	0.2501	0.5042	0.2467	0.4968
Distortion (PSNR)	33.17	36.28	26.77	30.53
Zerotrees (bpp)	0.2500	0.5000	0.2500	0.5000

It is clear from these numbers that *any* decent entropy coding scheme will do a good job at compressing this symbol stream, since by taking such a straightforward approach we are obtaining a performance improvements over the standard Zerotrees: it is conceivable that some work along these lines will yield further improvements. Besides, if low complexity implementations are sought, there are computationally more efficient entropy coders than the adaptive arithmetic coder. We are currently exploring this approach. Preliminary versions of a coder based on these ideas show that performance comparable to that achieved by much more complex schemes can be achieved while taking less than 5 seconds to run on a PC-like machine. Our encoder/decoder requires only 1 floating point multiplication/division per pixel, does not require an arithmetic coder (only static Huffman coding, no run-time adaptation), and is entirely based on table lookup operations with tables computed at the encoder and encoded in the bitstream, thus avoiding hard to justify choices of prestored parameters. Yet under such stringent complexity constraints, its coding performance on typical test images is superior to that of the state of the art Zerotree wavelet based algorithm, and less than 1dB lower than that of the absolute best coders published in the literature, while drastically outperforming them in terms of speed.

This substantial speedup basically enables the incorporation of high performance wavelet based image coding techniques into applications which, like in the Image Databases case, no hardware implementations are possible. Furthermore, our coder supports a key requirement of the Databases application: progressive mode transmission. This feature is important for browsing, since low resolution images are encoded at the beginning of the compressed bitstream; if network delays occur, the user can view partial reconstructions of his query, and if it turns out that the retrieved image is not the one he was looking for, then transmission can be aborted *before* the whole image is received, thus making the interactive process much faster.

3 Future Research

The MARS project was started to address the growing need for developing an effective multimedia database management system. Such an effort requires an integrated approach encompassing the fields of image analysis and coding, computer vision, information management, and database systems. As a first step towards MARS, we have developed an image retrieval systems (MARS/IRS) which incorporates some novel approaches to image segmentation, object representation, image coding and query processing. However, the prototype system built is only at its infancy and further investigation is required before we come close to our goals of developing an effective multimedia database management system. Below we discuss future research directions within the MARS project.

- *Coding for Retrieval:* Work in the coding aspects will focus on making evaluation of content-based queries on images possible directly on the compressed domain, without having to fully decompress the image. The coding methods being explored for this application make use of a feature unique to the wavelet transform; i.e., the structure in the transform domain is related to the spatial structure in the image. Unlike in other transforms, this makes it feasible to obtain easy access to shape representations directly in the wavelet domain. Research will be done to determine to what extent this is feasible and/or practical. It has been observed empirically that object structure can be clearly recognized in the wavelet domain. However, heavy use will need to be made of the semantic content of the scene being coded to make the task of identifying shapes feasible.
- *Automated Image Feature Extraction:* Automated feature extraction is one of the most important requirements for a scalable multimedia database system. We will focus our attention primarily on automated *texture* feature extraction. Methods dealing with texture extraction fall into two main categories. The first one is Statistics-based method, such as Markov Random Field model, Coocurrent Matrix, Fractal Model, etc. The second one is Transform-based method, including Discrete Fourier Transform (DFT), Gabor Filter, and DWT models, etc. The statistics-based methods are normally computationally expensive and the accuracy is lower than that of Transform-based methods. Therefore, the Transform-based methods are preferred.

  Among the Transform-based methods, DFT can not achieve localization in the transformed domain and Gabor Filter involves complex number computation whereas the DWT is both localized in the transformed domain and easy to compute. Almost all of the existing DWT models use quad-tree decomposition in the spatial domain (Pyramid and Tree structures in the transformed domain). An obvious disadvantage of quad-tree based methods is that the segmentation that they can perform must be of square shape (Egger, Ebrahimi, and Kunt, 1996; Gever and Kajcovski, 1994). However, the majority of the natural images contain texture regions of arbitrary shapes. It is almost impossible to find a square texture region inside a natural image. Besides, rotation-invariance is also an almost ignored research issue (Haley and Manjunath, 1995).

  We will explore a DWT model which can achieve the following goals:

  - Automated feature extraction;
  - Texture region can be of arbitrary shape;
  - Texture feature is rotation-invariant.
- *Efficient Feature Indexing:* A primary retrieval technique in multimedia databases is to use features extracted from the images. Hence, efficient indexing and retrieval of the features is very crucial for scalability of the system. The feature space normally is very high dimensional and, therefore, usage of conventional multidimensional and spatial indexing methods (e.g., R-trees, quad trees, grid files) is not feasible for feature indexing. Existing multidimensional index method are only useful when the number of dimensions are reasonably small. For example, the R-Tree based methods, which are among the most robust multidimensional indexing mechanisms, work well only for multidimensional spaces with dimensionality around 20. Other methods do not even scale to 20 dimensions.

  An approach used by the QBIC to overcome the dimensionality curse of the feature space is to transform the high dimensional feature space to a lower dimensional space using, for example, a K-L transform. An R^* tree is then used for indexing and retrieval in a lower dimensional space. The retrieval over the index provides a superset of the answers which can then be further refined in the higher dimensional space. While the approach is attractive and the QBIC authors report good retrieval efficiency over small image databases, it is not clear whether it will scale to large databases and complex feature spaces that

are very highly multidimensional. In such situations, the high number of false hits in the lower dimensional space might make the approach unusable. We will explore extensions to the QBIC approach and/or alternate methods to overcoming the dimensionality curse. One important direction of research is methods for selecting optimal ways to map a high dimensional feature space to a lower dimensional space based on the nature of commonly occurring queries and the nature of the feature vectors.

- *Effective Retrieval Models:* As discussed earlier, the retrieval model used to implement complex boolean queries in our current implementation is very simple. The choice of the retrieval model has been dictated by issues of efficiency, simplicity and quick prototyping. We are now examining more complex retrieval models developed in the information retrieval literature for supporting boolean queries over the image feature database. Among the models being examined is the inference network model used by the INQUERY system (Callan, Croft, and Harding, 1992). We will also explore how index structures can be used to support the developed retrieval model efficiently.
- *Integration with SQL*: An important consideration in the design of the multimedia database system is its integration with the organization's existing databases. This requires integration of the query language developed for the multimedia database (which allows content-based and similarity retrieval) with SQL (a popular database query language). Such an integration will allow users to develop complex applications in which images as well as other multimedia data can be considered simply as another data type and the applications have a mechanism for retrieving information based on both visual as well as traditional non-visual properties of data in the same query. Another related concept that we will explore is correlation of concepts from one media to another media.
- *Support for Concurrent Access:* Scalable design requires that concurrent operations (indexing new images, retrievals, updates) be supported over the multimedia database. Supporting concurrent operations over the feature database is challenging since it contains multidimensional data and uses multidimensional access structures (e.g., R-trees) for efficient retrieval. Concurrent access of multidimensional access methods is an important open research problem.

  A common requirement for concurrent access in database systems is to provide phantom protection to achieve degree 3 consistency, or repeatable read (RR) (Gray and Reuter, 1993). Key-range locking employed in B-tree, a mature dynamic indexing mechanism in single attribute database system, is a well-known and robust solution. The major issue is that this scheme depends on the linear order of keys. However, in R-tree, a dynamic index structure used in multi-dimensional space, the linear order of keys does not exist. As a result new mechanisms to overcome the phantom problem for multidimensional data needs to be developed. One promising direction is to use two versions of R-tree where all operations can concurrently run in the new version's R-tree after they set the locks on the proper entry in the old version's R-tree. The old version R-tree is essentially used as a partitioning of the space into lockable granules. The old version can either be used to provide static partitioning of the multidimensional space, which will result in a simpler solution but will result in lower concurrency, or could be updated by a periodic version switch resulting in a dynamically changing space partitioning. This technique will support higher concurrency but will be significantly more complex.
- *Supporting Concept Queries:* In a large number of applications of multimedia retrieval systems, users seldom use low-level image features (i.e., shape, color, texture) directly to query the database. Instead, user interacts with the system using high-level concepts (e.g., a beach, forest, yellow flowers, a sunset) in specifying a particular image content. These concept queries, in turn, need to be translated into queries over the low-level features so as

to be answered using the feature database. Such a translation results in a complex query over the low-level feature space. Providing capability to support concept queries over the feature database is one of the prime reasons we chose to implement support for complex boolean queries in MARS/IRS. However, in the current implementation, the MARS/IRS system does not provide any help to the user in mapping a high-level concept query into an equivalent query over the low-level feature space. We are currently investigating user interface extensions that can (partially) automate such a translation. In the approach being investigated, the system uses relevance feedback from users to learn concepts.

4 Acknowledgements

The authors would like to acknowledge Yong Rui's help in writing of this paper.

5 References

Callan, J. P.; Croft, W. B.; and Harding, S. M. (1992). The INQUERY retrieval system. In Proceedings of the Third International Conference on Database and Expert Systems Applications. Valencia, Spain.

Gray, J. and Reuter, A. (1993). Transaction Processing: Concepts and Techniques. Morgan Kaufmann, San Mateo, CA.

Salton, G. and McGill, M. J. (1983). Introduction to Modern Information Retrieval. McGraw Hill Computer Science Series.

Stonebraker, M. and Kemnitz, G. (1991). The POSTGRES Next-Generation Database Management System. Communications of the ACM, 34(10): 78–92.

Faloutsos, C.; Flicker, M.; Niblack, W.; Petkovic, D.; Equitz, W.; and Barber, R. (1993). Efficient and Effective Querying By Image Content, IBM Research Report RJ 9453 (83074).

Flickner, M. etal. (1995). Query by Image and Video Content: The QBIC System, Computer, September.

Smith, John R. and Chang, Shih-Fu (1994) Tools and Techniques for Color Image Retrieval, In IS & T/SPIE Proceedings Vol.2670, Storage & Retrieval for Image and Video Databases IV.

Smith John R. and Chang, Shih-Fu (1995). Single Color Extraction and Image Query, In Proc. ICIP.

Smith John R. and Chang, Shih-Fu (1996). Automated Binary Texture Feature Sets for Images Retrieval, In Proc. ICASSP.

Wang, Hualu (1995). Compressed-Domain Image Search and Applications, Columbia University Technical Report.

Pentland, A.; Picard, R.W.; and Sclaroff, S. (1995). Photobook: Tools for Content-Based Manipulation of Image Databases, Proc. Storage and Retrieval for Image and Video Databases II, Vol. 2, 185, SPIE, Bellingham, Wash, pp34-47.

Manjunath, B.S. and Ma, W.Y. (1995). Texture Features for Browsing and Retrieval of Image Data, CIPR TR-95-06, July.

Rui, Yong; She, Alfred C. and Huang, Thomas S. (1996). Automated Region Segmentation Using Attraction-Based Grouping in Spatial-Color- Texture Space, to appear in Proc. ICIP.

Rui, Yong; She, Alfred C. and Huang, Thomas S. (1996). Modified Fourier Descriptor for Shape Representation – A Practical Approach, accepted to First International Workshop on Image Databases and Multi Media Search, Amsterdam, The Netherlands.

Scharz, Bruce and Chen, Hsinchun (1996). Special Issue on Digital Library Initiative, Computer, May.

Egger, Olivier; Ebrahimi, Tourdadj and Kunt, Murat (1996). Arbitrarily-Shaped Wavelet Packets for Zerotree Coding, Proc. ICASSP.

Gevers, T.; and Kajcovski, V.K. (1994). Image Segmentation by Directed Region Subdivision, Proc. ICIP.

Haley, George M. and Manjunath, B.S. (1995). Rotation-Invariant Texture Classification Using Modified Gabor Filters, Proc. ICIP.

6 Vitae

Thomas S. Huang is William L. Everitt Distinguished Professor in the Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, and Research Professor at the Coordinated Science Laboratory and the Beckman Institute for Advanced Science and Technolgy at the University. He is the Chair of the Human Computer Intelligent Interaction major research theme at the Beckman Institute. Before joining UIUC, he served on the faculties of MIT and Purdue University. His research interests lie in the broad area of information processing, esp. in the acquisition, representation, analysis, manipulation, and visualization of multidimensional signals and data. He has published 11 books and more than 300 journal and conference papers in digital filtering, digital holography, image/video compression, image enhancement, image databases, and vision/speech-based human computer interface.

Sharad Mehrotra is an assistant professor in the Department of Computer Science, University of Illinois at Urbana Champaign since 1994. Before joining the University of Illinois, he worked as a scientist for Matsushita Information Technology Laboratory, Princeton from 1993-94. His research interests include database management, distributed systems, and information retrieval. He has authored over 20 journal and conference papers in transaction processing, multidatabase systems, text indexing systems, and distributed information systems.

Kannan Ramchandran has been an Assistant Professor in the Electrical and Computer Engineering Department, and a Research Assistant Professor in the Beckman Institute at the University of Illinois at Urbana Champaign since 1993, when he received his Ph.D. in Electrical Engineering from Columbia University. He was a Member of the Technical Staff at AT&T Bell Labs, New Jersey, from 1984-1990. His research interests include image and video compression, multirate signal processing and wavelets, telecommunications, and image communications. He has over 40 journal and conference publications in these areas, and holds 3 patents.