

Information Retrieval Beyond the Text Document *

Yong Rui, Michael Ortega, Thomas S. Huang
Beckman Institute for Advanced Science and Technology
University of Illinois at Urbana-Champaign
Urbana, IL 61801, USA

E-mail: {yrui, huang}@ifp.uiuc.edu, {miki, sharad}@ics.uci.edu

Sharad Mehrotra
Department of Information and
Computer Science
University of California, Irvine
Irvine, CA, 92697-3425

Abstract

With the expansion of the Internet, searching for information goes beyond the boundary of physical libraries. Millions of documents of various media types, such as text, image, video, audio, graphics, and animation, are available around the world and linked by the Internet.

Unfortunately, the state of the art of search engines for media types other than text lags far behind their text counterparts. To address this situation, we have developed the Multimedia Analysis and Retrieval System (MARS). This paper reports some of the progress made over the years towards exploring information retrieval beyond the text domain. In particular, the following aspects of MARS are addressed in the paper: visual feature extraction, retrieval models, query reformulation techniques, efficient execution speed performance and user interface considerations. Extensive experimental results are reported to validate the proposed approaches.

1 Introduction

Huge amounts of digital data are being generated every day. Scanners convert the analog/physical data into digital form; digital cameras and camcorders directly generate digital

*This work was supported by NSF CAREER award IIS-9734300; in part by NSF CISE Research Infrastructure Grant CDA-9624396; in part by the Army Research Laboratory under Cooperative Agreement No. DAAL01-96-0003. Yong Rui is supported in part by CSE, College of Engineering, UIUC. Michael Ortega is supported in part by CONACYT grant 89061. Some example images used in this article are used with permission from the Fowler Museum of Cultural History at the University of California–Los Angeles. These images were part of an image database delivery project called the Museum Educational Site Licensing Project (MESL), sponsored by the Getty Information Institute.

data at the production phase. Owing to all these multimedia devices, nowadays information is in all media types, including graphics, images, audio, and video, in addition to the conventional text media type.

Not only is multimedia information being generated at an ever increasing rate, it is transmitted all over the world due to the expansion of the Internet. Experts say that the Internet is the largest library that ever existed, it is however also the most disorganized library ever.

Textual document retrieval has achieved considerable progress over the past two decades. Unfortunately, the state of the art of search engines for media types other than text lags far behind their text counterparts. Textual indexing of non textual media, although common practice has some limitations. The most notable limitations include the human effort required and the difficulty of describing accurately certain properties humans take for granted while having access to the media. Consider how human indexers would describe the ripples on an ocean; these could be very different under situations such as calm weather or a hurricane. To address this situation, we undertook the Multimedia Analysis and Retrieval System (MARS) project to provide retrieval capabilities to rich multimedia data.

Research in MARS addresses several levels including the multimedia features extracted, the retrieval models used, query reformulation techniques, efficient execution speed performance and user interface considerations.

This paper reports some of the progress made over the years towards exploring Information Retrieval (IR) beyond the text domain. In particular, this paper will concentrate on Visual Information Retrieval (VIR) concepts as opposed to implementation issues in this paper.

MARS explores many different visual feature representations. A review of these features appears in section 2. These visual features are analogous to keyword features in textual media. In section 3 describes two broad retrieval models we have explored: the Boolean and vector models and the incorporated enhancements to support visual media retrieval such as relevance feedback. Experimental results are given in section 4. Promising research directions are outlined in section 5 and concluding remarks are discussed in Section 6.

2 Visual Feature Extraction

The retrieval performance of any IR system is fundamentally limited by the quality of the “features” and the retrieval model it supports. This section sketches the features obtained from visual media. In text based retrieval systems, features can be keywords, phrases or structural elements and there are many techniques for reliably extracting for example keywords from text documents. The visual counterparts to textual features in visual based systems are visual features such as color, texture, and shape.

For each feature there are several different techniques for representation. The reason for this is twofold: a) the field is still under development, and b) more importantly, certain features are perceived differently by different people and thus different representations cater to different preferences. Image features are generally considered as orthogonal to each other. The idea is that a feature will capture some dimension of the content of the image, and different features will effectively capture different aspects of the image content. In this way

two images closely related in one feature could be very different in another feature. A simple example of this are two images, one of a deep blue sky and the other of a blue ocean. These two images could be very similar in terms of just color, however the ripples caused by waves in the ocean add a distinctive pattern that distinguishes the two images in terms of their texture. The following sections describe common features.

2.1 Color Features

The Color feature is one of the most widely used visual features in VIR. The Color feature captures the color content of images. It is relatively robust to background complication and independent of image size and orientation. Some representative studies of color perception and color spaces can be found in [McCamy et al., 1976, Miyahara, 1988, Wang et al., 1997].

In VIR, the Color Histogram is the most commonly used color feature representation. Statistically, it denotes the joint probability of the intensities of the three color channels. Swain and Ballard proposed Histogram Intersection, a L_1 metric, as the similarity measure for the Color Histogram representation [Swain and Ballard, 1991]. To take into account the similarities between similar but not identical colors, Ioka [Ioka, 1989] and Niblack et al. [Niblack et al., 1994] introduced a L_2 -related metric for comparing the histograms. Furthermore, considering that most Color Histograms are very sparse and thus sensitive to noise, Stricker and Orengo proposed to use the cumulative Color Histogram. Their research results demonstrated the advantages of the proposed approach over the conventional Color Histogram approach [Stricker and Orengo, 1995].

Besides Color Histogram, several other color feature representations have been considered in VIR, including Color Moments and Color Sets. To overcome the quantization effects present in the Color Histogram representation, Stricker and Orengo proposed the Color Moments approach [Stricker and Orengo, 1995]. The mathematical foundation of this approach is that any color distribution can be characterized by its moments. Furthermore, since most of the information is concentrated on the low-order moments, only the first (mean), second (variance) and third (skewness) central moments are extracted as the color feature representation. Weighted Euclidean distance is then used to calculate the color similarity.

To facilitate fast search over large-scale image collections, Smith and Chang proposed Color Sets as an approximation to the Color Histogram representation [Smith and Chang, 1995a, Smith and Chang, 1995b]. They first transformed the (R,G,B) color space into a perceptually uniform space, such as (H,S,V) [Foley et al., 1990], and then quantized the transformed color space into M bins. A Color Set is defined as a selection of the colors from the quantized color space. Because Color Set feature vectors are binary, a binary search tree was constructed to allow fast search. The relationship between the proposed Color Sets and the conventional Color Histogram was further discussed in [Smith and Chang, 1995a, Smith and Chang, 1995b].

2.2 Texture Features

Texture refers to the visual patterns that have properties of homogeneity that do not result from the presence of only a single color or intensity [Smith and Chang, 1996]. It is an innate property of virtually all surfaces, including clouds, trees, bricks, hair, fabric, etc. It contains

important information about the structural arrangement of surfaces and their relationship to the surrounding environment [Haralick et al., 1973]. Because of its importance and usefulness in Pattern Recognition and Computer Vision, a rich set of research results exists, spanning the past three decades. Now, it further finds its way in VIR. More and more research achievements are being added to it.

In the early 70's, Haralick et al. proposed the co-occurrence matrix representation for the texture feature [Haralick et al., 1973]. This approach explored the gray level spatial dependence of texture. It first constructed a co-occurrence matrix based on the orientation and distance between image pixels and then extracted meaningful statistics from the matrix as the texture representation. Many other researchers followed the same line and further proposed enhanced versions. For example, Gotlieb and Kreyszig [Gotlieb and Kreyszig, 1990] studied the statistics originally proposed in [Haralick et al., 1973] and experimentally found out that *contrast*, *inverse difference moment* and *entropy* had the biggest discriminatory power.

Motivated by the psychological studies in human visual perception of texture, Tamura et al. explored the texture representation from a different angle [Tamura et al., 1978]. They developed computational approximations to the visual texture properties found to be important in psychology studies. The six visual texture properties were *coarseness*, *contrast*, *directionality*, *linelikeness*, *regularity*, and *roughness*. One major distinction between the Tamura texture representation and the co-occurrence matrix representation is that all the texture properties in the Tamura representation are visually meaningful whereas some of the texture properties used in co-occurrence matrix representation may not (for example, entropy). This characteristic makes the Tamura texture representation very attractive in VIR, as it can provide a friendlier user interface. The QBIC [Equitz and Niblack, 1994] and MARS [Huang et al., 1996, Ortega et al., 1997] systems further improved this texture representation.

In the early 90's, after the Wavelet transform was introduced and its theoretical framework established, many researchers began to study its applications to texture representation [Smith and Chang, 1994, Chang and Kuo, 1993, Laine and Fan, 1993, Gross et al., 1994, Kundu and Chen, 1992, Thyagarajan et al., 1994]. In [Smith and Chang, 1994, Smith and Chang, 1996], Smith and Chang used the mean and variance statistics extracted from the Wavelet subbands as the texture representation. This approach achieved over 90% accuracy on the 112 Brodatz texture images.

2.3 Shape Features

In general, the shape representations can be divided into two categories, boundary-based and region-based. The former uses only the outer boundary of the shape while the latter uses the entire shape region [Rui et al., 1996]. The most successful representatives for these two categories are Fourier Descriptor and Moment Invariants.

The main idea of the Fourier Descriptor is to use the Fourier transformed boundary as the shape feature. Some early work can be found in [Zahn and Roskies, 1972, Persoon and Fu, 1977]. To take into account the digitization noise in the image domain, Rui et al. proposed a modified Fourier Descriptor which is both robust to noise and invariant to geometric transformations [Rui et al., 1996].

The main idea of Moment Invariants is to use region-based moments, which are invariant to transformations, as the shape feature. In [Hu, 1962], Hu identified seven such moments. Based on his work, many improved versions emerged. In [Yang and Albrechtsen, 1994], based on the discrete version of Green’s theorem, Yang and Albrechtsen proposed a fast method of computing moments in binary images. Motivated by the fact that most useful invariants were found by extensive experience and trial-and-error, Kapur et al. developed algorithms to systematically generate and search for a given geometry’s invariants [Kapur et al., 1995]. Realizing that most researchers did not consider what happened to the invariants after image digitization, Gross and Latecki developed an approach which preserved the qualitative differential geometry of the object boundary, even after an image was digitized [Gross and Latecki, 1995].

Some recent work in shape representation and matching includes the Finite Element Method (FEM) [Pentland et al., 1996], Turning Function [Arkin et al., 1991], and Wavelet Descriptor [Chuang and Kuo, 1996]. FEM defines a stiffness matrix, which describes how each point on the object is connected to other points. The eigenvectors of the stiffness matrix are called modes and span a feature space. All the shapes are first mapped into this space and similarity is then computed based on the eigenvalues. Along a similar line to the Fourier Descriptor, Arkin et al. developed a Turning Function based method for comparing both convex and concave polygons [Arkin et al., 1991]. In [Chuang and Kuo, 1996], Chuang and Kuo used the Wavelet transform to describe object shapes. It embraced desirable properties such as multi-resolution representation, invariance, uniqueness, stability, and spatial localization. For shape matching, Chamfer matching attracted much research attention. Barrow et al. first proposed the Chamfer matching technique, which compared two collections of shape fragments at a cost proportional to linear dimension, rather than area [Barrow, 1977]. In [Borgefors, 1988], to speed up the Chamfer matching process, Borgefors proposed a hierarchical Chamfer matching algorithm. The matching was done at different resolutions, from coarse to fine.

2.4 Color Layout Features

Although the global color feature is simple to calculate and can provide reasonable discriminating power in VIR, it tends to give too many false positives when the image collection is large. Many research results suggested that using color layout (both color feature and spatial relations) is a better solution to VIR. To extend the global color feature to a local one, a natural approach is to divide the whole image into sub-blocks and extract color features from each of the sub-blocks [Faloutsos et al., 1993, Chua et al., 1997]. A variation of this approach is the quad-tree based color layout approach [Lu et al., 1994], where the entire image is split into a quad-tree structure and each tree branch has its own histogram to describe the color content. Although conceptually simple, this regular-subblock based approach cannot provide accurate local color information while being expensive in terms of computation and storage. A more sophisticated approach is to segment the image into regions with salient color features by Color Set Back-projection, and then store the position and Color Set feature of each region to support later queries [Smith and Chang, 1995a]. The advantage of this approach is its accuracy while its disadvantage is the difficult general problem of reliable image segmentation.

To achieve a good trade-off between the above two approaches, several other color layout representations were proposed. In [Rickman and Stonham, 1996], Rickman and Stonham proposed a color tuple histogram approach. They first constructed a code book which described every possible combination of coarsely quantized color hues that might be encountered within local regions in an image. Then a histogram based on quantized hues was constructed as the local color feature. In [Stricker and Dimai, 1996], Stricker and Dimai extracted the first three color moments from five predefined partially overlapping fuzzy regions. The usage of the overlapping region made their approach relatively insensitive to small region transformations. In [Pass et al., 1996], Pass et al. classified each pixel of a particular color as either coherent or incoherent, based on whether or not it is part of a large similarly-colored region. By using this approach, widely scattered pixels were distinguished from clustered pixels thus improving the representation of local color features. In [Huang et al., 1997], Huang et al. proposed a color correlogram based on the color layout representation. They first constructed a color co-occurrence matrix and then used the auto-correlogram and correlogram as the similarity measures. Their experimental results showed that this approach was more robust than the conventional Color Histogram approach in terms of retrieval accuracy [Huang et al., 1997].

Along the same line of the Color Layout feature, the layout of texture and other visual features can also be constructed to facilitate more advanced VIR.

3 Retrieval Models used in MARS

With the large number of retrieval models proposed in the IR literature, MARS attempts to exploit this research for content-based retrieval over images. The retrieval model comprises the document or object model (here a collection of feature representations), a set of feature similarity measures, and a query model.

3.1 The Multimedia Object Model

We first need to formalize how an object is modeled [Rui et al., 1998b]. We will use images as an example, even though this model can be used for other media types as well. An image object O is represented as:

$$O = O(D, F, R) \tag{1}$$

- D is the raw image data, e.g. a JPEG image.
- $F = \{f_i\}$ is a set of low-level visual features associated with the image object, such as color, texture, and shape.
- $R = \{r_{ij}\}$ is a set of representations for a given feature f_i , e.g. both color histogram and color moments are representations for the color feature [Swain and Ballard, 1991]. Note that, each representation r_{ij} itself may be a vector consisting of multiple components, i.e.

$$r_{ij} = [r_{ij1}, \dots, r_{ijk}, \dots, r_{ijK}] \tag{2}$$

where K is the length of the vector.

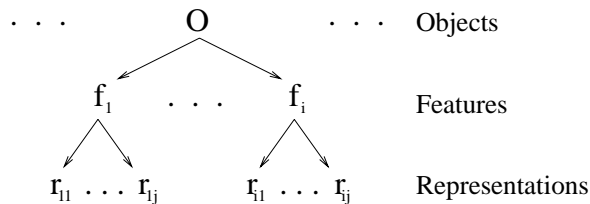


Figure 1: The Object Model

Figure 1 shows a graphic representation of the Object (Image) model. The proposed object model supports multiple representations to accommodate the rich content in the images. An image is thus represented as a collection of low-level image feature representations (section 2) extracted automatically using computer vision methods, as well as a manual text description of the image.

Each feature representation is associated with some similarity measure (see section 2). All these similarity measures are normalized to lie within $[0,1]$ to denote the degree to which two images are similar in regard to the same feature representation. A value of 1 means they are very similar and a value of 0 means they are very dissimilar. Revisiting our blue sky and ocean example from section 2, the sky and ocean images may have a similarity of 0.9 in the Color Histogram representation of Color and 0.2 in the Wavelet representation of Texture. Thus the two images are fairly similar in their color content, but very different in their texture content. This mapping $M = \{ \langle feature\ representation_i, similarity\ measure_i \rangle, \dots \}$ together with the Object model O , forms (D, F, R, M) , a foundation on which retrieval models can be built.

3.2 Query Models

Based on the Object model and the similarity Measures defined above, Query models that work with these raw features are built. These Query models together with the Object model form complete retrieval models used for VIR.

We explore two major models for querying. The first model is an adaption of the Boolean retrieval model to visual retrieval in which selected features are used to build predicates used in a Boolean expression. The second model is a vector (weighted summation) model where all the features of the query object play a role in retrieval. Section 3.3 describes the Boolean model and section 3.4 describes the vector model.

3.3 Boolean Retrieval

A user graphically constructs a query by selecting certain images from the collection. A user may choose specific features from the selected images. For example, using a point-and-click interface a user can specify a query to retrieve images similar to an image A in color and similar to an image B in texture. A user's query is then interpreted as a Boolean expression over image features. A Boolean retrieval model (adapted for retrieval over images) is used to interpret the query and retrieve a set of images ranked based on their similarity to the selected feature.

To see how MARS adapts the Boolean model for image retrieval, consider first a query Q over a single feature F_i (say color represented as a color histogram). Let $H(I)$ be the color histogram of image I and $H(Q)$ be the color histogram specified in the query and $\text{similarity}(H(I), H(Q))$ be the similarity between the two histograms. The simplest way to adapt the Boolean model for image retrieval is to associate a *degree of tolerance* δ_i with each feature F_i such that:

$$\begin{aligned} I \text{ matches } Q &= \text{true, if } \text{similarity}(H(I), H(Q)) \geq \delta_i \\ &= \text{false, if } \text{similarity}(H(I), H(Q)) < \delta_i \end{aligned}$$

Given the above interpretation of a match based on a single feature F_i , an image I matches a given query Q if it satisfies the Boolean expression associated with Q . For example, let $Q = v_1 \wedge v_2$, where v_1 is a color histogram, and v_2 is a texture representation. Image I matches Q if its color and texture representations are within the specified tolerances of v_1 and v_2 .

Although the above straightforward adaptation of Boolean retrieval can be used for retrieval, it has several potential problems. First, it is not clear how the degree of tolerance δ_i , for a given feature F_i , should be determined. If an *a priori* value is set for δ_i , it may result in poor performance – two images I_1 and I_2 at similarity of $\delta_i + \epsilon$ and $\delta_i - \epsilon$ from a query Q , where $\epsilon \rightarrow 0$, are very similar as far as their relevance to Q is concerned but would be considered as very different by the system. While I_1 would be considered relevant to the query, I_2 would not be considered as relevant. This problem may be alleviated by dynamically computing δ_i for each query based on the image collection instead of using fixed *a priori* tolerance values for a given feature. However, the approach still suffers from the fundamental restriction of the basic Boolean retrieval in that it produces an unranked set of answers.

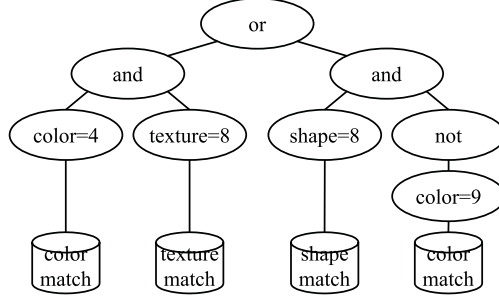
To overcome the above discussed problems, we have adopted the following two extensions to the basic Boolean model to produce a ranked list of answers.

Fuzzy Boolean Retrieval. The similarity between the image and the query feature is interpreted as the degree of membership of the image to the fuzzy set of images that match the query feature. Fuzzy set theory is used to interpret the Boolean query and the images are ranked based on their degree of membership in the set.

Probabilistic Boolean Retrieval. The similarity between the image and the query feature is considered to be the probability that the image matches the user’s information need. Feature independence is exploited to compute the probability of an image satisfying the query which is used to rank the images.

Unlike the basic Boolean model, both the fuzzy and probabilistic Boolean models provide ranked retrieval over the image collection.

In the discussion below, we will use the following notation. Images in the collection are denoted by I_1, I_2, \dots, I_m . Features over the images are denoted by F_1, F_2, \dots, F_r , where F_i denotes both the name of the feature as well as the domain of values that the feature can take. The j^{th} instance of feature F_i corresponds to image I_j and is denoted by f_{ij} . For example, say F_1 is the color feature which is represented in the database using a histogram.



Operators: And, Or, Not
 Basic features and representations:
 Color histogram, color moment, wavelet texture, ...

Figure 2: Sample query tree

In that case, F_1 is also used to denote the set of all the color histograms, and $f_{1,5}$ is the color histogram for image 5. Query variables are denoted by $v_1, v_2, \dots, v_n \mid v_k \in F_i$ so each v_k refers to an instance of a feature F_i (an $f_{i,j}$). Note that $F_i(I_j) = f_{i,j}$. During query evaluation, each v_k is used to rank images in the collection based on the feature domain of f_i (F_i), that is v_k 's domain. Thus, v_k can be thought of being a list of images from the collection ranked based on the similarity of v_k to all instances of F_i . For example, say F_2 is the set of all wavelet texture vectors in the collection, if $v_k = f_{2,5}$, then v_k can be interpreted as being both, the wavelet texture vector corresponding to image 5 and the ranked list of all $\langle I, S_{F_2}(F_2(I), f_{2,5}) \rangle$ with S_{F_2} being the similarity function that applies to two texture values. A query $Q(v_1, v_2, \dots, v_n)$ is viewed as a query tree whose leaves correspond to single feature variable queries. Internal nodes of the tree correspond to the Boolean operators. Specifically, non-leaf nodes are of one of three forms: $\wedge(v_1, v_2, \dots, v_n)$, a conjunction of positive literals; $\wedge(v_1, v_2, \dots, v_p, \neg v_{p+1} \dots \neg v_n)$, a conjunction consisting of both positive and negative literals; and $\vee(v_1, v_2, \dots, v_n)$, which is a disjunction of positive literals. Notice that we do not consider an unguarded negation or a negation in the disjunction (that is, $p \geq 1$), since it does not make much sense. Typically, a very large number of entries will satisfy a negation query virtually producing the universe of the collection [Beyer et al., 1998]. We therefore allow negation only when it appears within a conjunctive query to rank an entry on the positive feature discriminated by the negated feature. The following is an example of a Boolean query: $Q(v_1, v_2) = (v_1 = f_{1,5}) \wedge (v_2 = f_{2,6})$ is a query where v_1 has a value equal to the color histogram associated with image I_5 and v_2 has a value of the texture feature associated with I_6 . Thus, the query Q represents the desire to retrieve images whose color matches that of image I_5 and whose texture matches that of image I_6 . Figure 2 shows an example query $Q(v_1, v_2, v_3, v_4) = ((v_1 = f_{1,4}) \wedge (v_2 = f_{2,8})) \vee ((v_3 = f_{3,8}) \wedge \neg(v_4 = f_{1,9}))$ in its tree representation.

3.3.1 Weighting in the query tree

In a query, one feature can receive more importance than another according to the user's perception. The user can assign the desired importance to any feature by a process known as

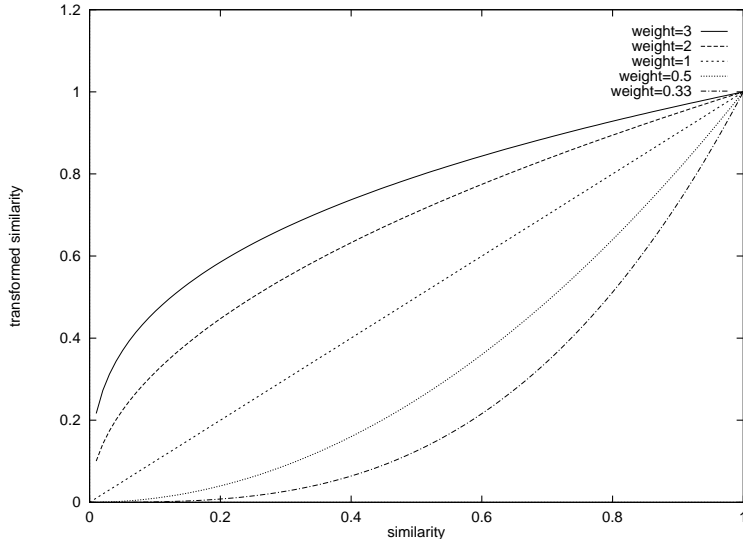


Figure 3: Various samples for similarity mappings

feature weighting. Traditionally, retrieval systems [Flickner et al., 1995, Bach et al., 1996] use a linear scaling factor as feature weights. Under our Boolean model, this is not desirable. It has been noted [Fagin and Wimmers, 1997] that such linear weights do not scale to arbitrary functions used to compute the combined similarity of an image. The reason is that the similarity computation for a node in a query tree may be based on operators other than a weighted summation of the similarity of the children. For example if the fuzzy model is used, and the node is \wedge , the similarity computation is done as $similarity_{\wedge} = \min(S_{F_i}, S_{F_j})$. If F_i carries a weight α , F_j a weight β and the above method is used, then $similarity_{\wedge} = \min(\alpha \times S_{F_i}, \beta \times S_{F_j})$ will be in the range $[0, \min(\alpha, \beta)]$ which is distinct from $[0, 1]$ in general. In [Fagin and Wimmers, 1997], the authors present a way to extend linear weighting to the different components for arbitrary scoring functions as long as they satisfy certain properties. We are unable to use their approach since their mapping does not preserve orthogonality properties on which our algorithms rely [Ortega et al., 1998b]. Instead, we use a mapping function from $[0, 1] \rightarrow [0, 1]$ of the form

$$similarity' = similarity^{\frac{1}{weight}}, \quad 0 < w < \infty \quad (3)$$

which preserves the range boundaries $[0,1]$ and boosts or degrades the similarity in a smooth way. Sample mappings are shown in figure 3. This method preserves most of the properties explained in [Fagin and Wimmers, 1997], except it is undefined for a weight of 0. In [Fagin and Wimmers, 1997], a weight of 0 means the node can be dismissed. Here, $\lim_{weight \rightarrow 0} similarity' = 0$ for $similarity \in [0, 1)$. A perfect similarity of 1 will remain at 1. This mapping is performed at each link connecting a child to a parent in the query tree.

3.3.2 Fuzzy Boolean Query Model

Let $Q(v_1, v_2, \dots, v_n)$ be a query and I be an image. In the fuzzy retrieval model, a query variable v_i is considered to be a fuzzy set of images and the relevance of any image I to Q with respect to v_i is interpreted as the degree of membership of I in that fuzzy set.

With the above interpretation of the similarity measure between the image feature and the feature specified in the query, a Boolean query Q is interpreted as an expression in fuzzy logic and fuzzy set theory is used to compute the degree of membership of an image to the fuzzy set represented by the query Q . Specifically, the degree of membership for a query Q is computed as follows:

$$\begin{aligned} \text{And } S_{Q=Q_1 \wedge Q_2}(I) &= \min(S_{Q_1}(I), S_{Q_2}(I)) \\ \text{Or } S_{Q=Q_1 \vee Q_2}(I) &= \max(S_{Q_1}(I), S_{Q_2}(I)) \\ \text{Not } S_{Q=\neg Q_1}(I) &= 1 - S_{Q_1}(I) \end{aligned}$$

Consider for example a query Q :

$$Q = (v_1 \vee v_2 \vee v_3) \wedge (v_4 \vee (v_5 \wedge v_1)) \quad (4)$$

The degree of membership of an image I in the fuzzy set corresponding to Q can be determined as follows:

$$S_Q(I) = \min(\max(S_{v_1}(I), S_{v_2}(I), S_{v_3}(I)), \max(S_{v_4}(I), \min(S_{v_5}(I), S_{v_1}(I)))) \quad (5)$$

The value $S_{v_i}(I)$ in (5) is determined using the appropriate similarity or distance measure for the feature v and appropriately normalized. Once the membership value of the image in the fuzzy set associated with the query is determined, these values are used to rank the images, where a higher value of $S_Q(I)$ represents a better match of the image I to the query Q .

Figure 4a) shows how the fuzzy model would work with our running example of blue sky and blue ocean images.

3.3.3 Probabilistic Boolean Query Model

Let $Q(v_1, v_2, \dots, v_n)$ be a query and I an image. In the probabilistic Boolean model, the similarity $S_{F_i}(F_i(I), v_j)$ between the query variable v_j and the corresponding feature in the image is taken to be the probability of the image I matching the query variable v_j , denoted by $P(v_j|I)$. These probability measures are then used to compute the probability that I satisfies the query $Q(v_1, v_2, \dots, v_n)$ (denoted by $P(Q(v_1, v_2, \dots, v_n)|I)$) which is in turn used to rank the images. To enable computation of $P(Q(v_1, v_2, \dots, v_n)|I)$, an assumption of *independence* is made. That is, we assume that for all variables v_i, v_j following holds:

$$P(v_i \wedge v_j|I) = P(v_i|I) \times P(v_j|I) \quad (6)$$

Developing a term and feature dependence model and incorporating it may improve retrieval performance further and is an important extension to our work.

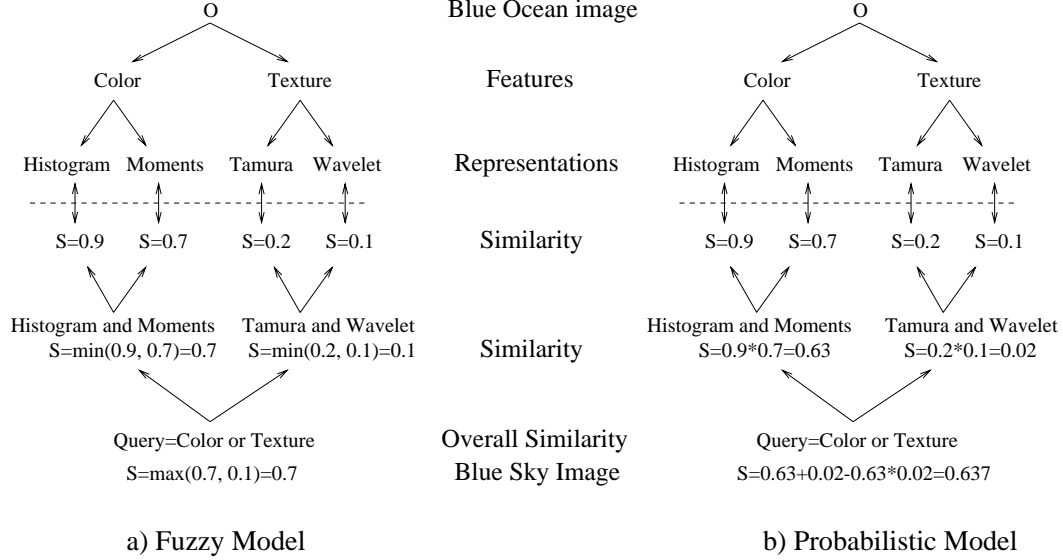


Figure 4: Various samples for similarity mappings

Once the probability of match is known for a basic feature, we next need to estimate the probability that the image satisfies the Boolean query $Q(v_1, v_2, \dots, v_n)$, denoted by $P(Q|I)$. If Q is a disjunction ($Q = Q_1 \vee Q_2$), following the laws of probability, $P(Q_1 \vee Q_2|I)$ can be estimated as follows:

$$P(Q_1 \vee Q_2|I) = P(Q_1|I) + P(Q_2|I) - P(Q_1 \wedge Q_2|I) \quad (7)$$

Since all probabilities are conditioned on the image I , we will omit this for brevity from now on. Similarly, $P(\neg Q)$ can be computed as follows:

$$P(\neg Q_1) = 1 - P(Q_1) \quad (8)$$

To compute conjunction queries, i.e. $Q = Q_1 \wedge Q_2$ we use

$$P(Q_1 \wedge Q_2) = P(Q_1) \times P(Q_2) \quad (9)$$

Taking the similarity values to mean the probability that the image matches the query, we use the following equations to compute the final result:

$$\begin{aligned} \mathbf{And} \quad S_{Q=Q_1 \wedge Q_2}(I) &= S_{Q_1}(I) \times S_{Q_2}(I) \\ \mathbf{Or} \quad S_{Q=Q_1 \vee Q_2}(I) &= S_{Q_1}(I) + S_{Q_2}(I) - S_{Q_1}(I) \times S_{Q_2}(I) \\ \mathbf{Not} \quad S_{Q=\neg Q_1}(I) &= 1 - S_{Q_1}(I) \end{aligned}$$

Our retrieval results (see section 4) show that even if query terms are considered as independent, the resulting retrieval performance is quite good. It should be noted that although $S_{F_i}(F_i(I_j), v_k)$ has the same value for the fuzzy and probabilistic models, their interpretation is different and yields different results (see section 4).

Figure 4b) shows how the probabilistic model would work with our running example of blue sky and blue ocean images.

3.3.4 Computing Boolean Queries

[Fagin, 1996] proposed an algorithm to return the top k answers for queries with monotonic scoring functions that has been adopted by the Garlic multimedia information system under development at the IBM Almaden Research Center [Fagin and Wimmers, 1997]. A function F is monotonic if $F(x_1, \dots, x_m) \leq F(x'_1, \dots, x'_m)$ for $x_i \leq x'_i$ for every i . Note that the scoring functions for both conjunctive and disjunctive queries for both the fuzzy and probabilistic Boolean models satisfy the monotonicity property. This algorithm relies on reading a number of objects from each branch in the query tree until it has k objects in the intersection. Then it falls back on probing to enable a definite decision. In contrast, our algorithms [Ortega et al., 1998b] are tailored to specific functions that combine object scoring (here called fuzzy and probabilistic models).

Another approach to optimizing query processing over multimedia repositories has been proposed in [Chaudhari and Gravano, 1996]. It presents a strategy to optimize queries when users specify thresholds on the grade of match of acceptable objects as filter conditions. It uses the results in [Fagin, 1996] to convert top- k queries to threshold queries and then process them as filter conditions. It shows that under certain conditions (uniquely graded repository), this approach is expected to access no more objects than the strategy in [Fagin, 1996]. Furthermore, while the above approaches have mainly concentrated on the fuzzy Boolean model, we consider both the fuzzy and probabilistic model in MARS. This is significant since the experimental results illustrate that the probabilistic model outperforms the fuzzy model in terms of retrieval performance (discussed in section 4).

3.4 Vector Model

An IR model consists of a document model, a query model, and a model for computing similarity between the documents and the queries. One of the most popular IR models is the vector model [Buckley and Salton, 1995, Salton and McGill, 1983, Shaw, 1995]. Various effective retrieval techniques have been developed for this model. Among them, *term weighting* and *relevance feedback* are of fundamental importance.

3.4.1 Term Weighting in Textual Media

Term weighting is a technique for assigning different weights for different keywords (terms) according to their relative importance to the document [Shaw, 1995, Salton and McGill, 1983].

If we define w_{ik} to be the weight for term t_k , $k = 1, \dots, N$, in document i (D_i), where N is the number of terms. Document i can be represented as a weight vector in the term space:

$$D_i = [w_{i1}, \dots, w_{ik}, \dots, w_{iN}] \quad (10)$$

To correctly estimate the weights, we need to consider two aspects. First, if term t_k occurs frequently in document i , then w_{ik} should be assigned a high value. This intuition suggests that a term frequency (*tf*) factor should be included in the estimation of w_{ik} . Second, *tf* alone cannot ensure an acceptable estimation. When the high frequency term is not concentrated in a few documents, but is instead spread over all documents, then this term

should receive a low weight. This introduces the well-known inverse document frequency (*idf*), which varies inversely with the number of documents in which a term appears.

$$idf_k = \log_2 \frac{M}{df_k} + 1 \quad (11)$$

where df_k is the document frequency for term t_k and M is the total number of documents in the collection. Experiments have shown that the product of *tf* and *idf* is a good estimation of the weights [Buckley and Salton, 1995, Salton and McGill, 1983, Shaw, 1995].

The query Q has the same model as that of a document D , i.e. it is a weight vector in the term space:

$$Q = [w_{q1}, \dots, w_{qk}, \dots, w_{qN}] \quad (12)$$

The similarity between D and Q is defined as the Cosine distance.

$$Similarity(D, Q) = \frac{D \times Q}{\|D\| \times \|Q\|} \quad (13)$$

where $\| \cdot \|$ denotes norm-2.

3.4.2 Relevance Feedback for term weighting

As we can see from the previous subsection, in the vector model, the specification of w_{qk} 's in Q is very critical, since the similarity values ($Similarity(D, Q)$'s) are computed based on them. However, it is usually difficult for a user to map his information need into a set of terms precisely. To overcome this difficulty, the technique of *relevance feedback* has been proposed [Salton and McGill, 1983, Shaw, 1995, Buckley and Salton, 1995]. Relevance feedback is the process of automatically adjusting an existing query using information feedback by the user about the relevance of previously retrieved documents.

The mechanism of this method can be described elegantly in the vector space. If the sets of relevant documents (D_R) and non-relevant documents (D_N) are known, the optimal query can be proven to be [Buckley and Salton, 1995, Salton and McGill, 1983, Shaw, 1995]:

$$Q_{opt} = \frac{1}{N_R} \sum_{i \in D_R} D_i - \frac{1}{N_T - N_R} \sum_{i \in D_N} D_i \quad (14)$$

where N_R is the number of documents in D_R and N_T the number of the total documents.

In practice, D_R and D_N are not known in advance. However, the relevance feedback obtained from the user furnishes approximations to D_R and D_N , which are referred as, D'_R and D'_N .

Putting more weight on the relevant terms and less weight on the non-relevant terms can modify the original query Q .

$$Q' = \alpha Q + \beta \left(\frac{1}{N_{R'}} \sum_{i \in D'_{R'}} D_i \right) - \gamma \left(\frac{1}{N_{N'}} \sum_{i \in D'_{N'}} D_i \right) \quad (15)$$

where α, β and γ are suitable constants [Salton and McGill, 1983, Shaw, 1995]; $N_{R'}$ and $N_{N'}$ are the numbers of documents in $D'_{R'}$ and $D'_{N'}$. Q' approaches Q_{opt} , as the relevance feedback

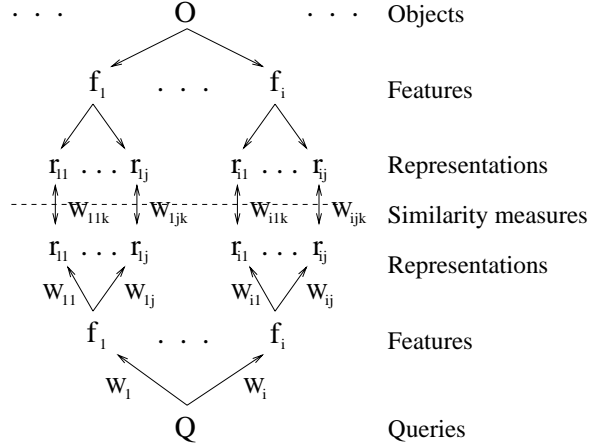


Figure 5: The retrieval process

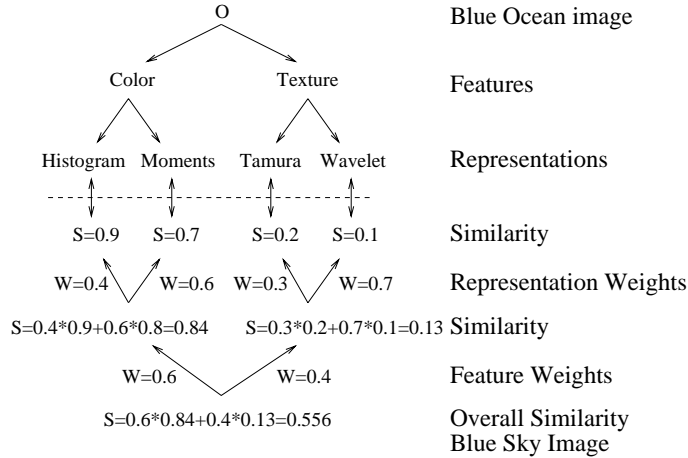


Figure 6: Example Query calculation of Blue Sky image against Blue Ocean image

iteration moves on. Experiments show that the retrieval performance can be improved considerably by using relevance feedback [Buckley and Salton, 1995, Salton and McGill, 1983, Shaw, 1995].

Term weighting and relevance feedback are powerful techniques in IR. We next generalize these concepts to VIR.

3.4.3 Vector Query Model and Integration of Relevance Feedback to VIR

As discussed in section 3.1, an object model $O(D, F, R)$, together with a set of similarity measures $M = \{m_{ij}\}$, provides the foundation for retrieval (D, F, R, M) . The similarity measures are used to determine how similar or dissimilar two objects are. Different similarity measures may be used for different feature representations. For example, Euclidean is used for comparing vector-based representations while Histogram Intersection is used for comparing color histogram representations (see section 2).

The Query model is shown in figure 5. The query has the same form as an object, except it has weights at every branch at all levels. W_i , W_{ij} , and W_{ijk} , are associated with features f_i , representations r_{ij} , and components r_{ijk} respectively. The purpose of the weights is to reflect as closely as possible the combination of feature representations that best represents the users information need. The process of relevance feedback described below aims at updating these weights to form the combination of features that best captures the user’s information need.

Intuitively, the similarity between query and object feature representations is computed, and then the feature similarity computed as the weighted sum of the similarity of the individual feature representations. This process is repeated one level higher when the overall similarity of the object is the weighted sum over all the feature similarities. The weights at the lowest level, the component level, are used by the different similarity measures internally. Figure 6 traces this process for our familiar example of a blue sky image as a query and a blue ocean image in the collection.

Based on the image object model and the set of similarity measures, the retrieval process is described below and also illustrated in Figure 5.

1. Initialize the weights $W = [W_i, W_{ij}, W_{ijk}]$ to $W0$, which is a set of no-bias weights. That is, every entity is initially of the same importance.

$$W_i = W0_i = \frac{1}{I} \quad (16)$$

$$W_{ij} = W0_{ij} = \frac{1}{J_i} \quad (17)$$

$$W_{ijk} = W0_{ijk} = \frac{1}{K_{ij}} \quad (18)$$

where I is the number of features in set F ; J_i is the number of representations for feature f_i ; K_{ij} is the length of the representation vector r_{ij} .

2. The user’s information need, represented by the query object Q , is distributed among different features f_i , according to their corresponding weights W_i .
3. Within each feature f_i , the information need is further distributed among different feature representations r_{ij} , according to the weights W_{ij} .
4. The objects’ similarity to the query, in terms of r_{ij} , is calculated according to the corresponding similarity measure m_{ij} and the weights W_{ijk} :

$$S(r_{ij}) = m_{ij}(r_{ij}, W_{ijk}) \quad (19)$$

5. Each representation’s similarity values are then combined into a feature’s similarity value:

$$S(f_i) = \sum_j W_{ij} S(r_{ij}) \quad (20)$$

6. The overall similarity S is obtained by combining individual $S(f_i)$ ’s:

$$S = \sum_i W_i S(f_i) \quad (21)$$

7. The objects in the database are ordered by their overall similarity to Q . The N_{RT} most similar ones are returned to the user, where N_{RT} is the number of objects the user wants to retrieve.
8. The user marks each of the retrieved objects as *highly relevant*, *relevant*, *no-opinion*, *non-relevant*, or *highly non-relevant*, according to his information need and subjective perception.
9. The system updates the weights (described in section 3.4.4) according to the user's feedback so that the adjusted Q is a better approximation to the user's information need.
10. Go to Step 2 with the adjusted Q and start a new iteration of retrieval.

In Figure 5, the information need embedded in Q flows up while the content of O 's flows down. They meet at the dashed line, where the similarity measures m_{ij} are applied to calculate the similarity values $S(r_{ij})$'s between Q and O 's.

Note that in the proposed retrieval algorithm, both S and $S(f_i)$ are linear combinations of their corresponding lower level similarities. The basis of the linear combination is that the weights are proportional to the entities relative importance [Fagin and Wimmers, 1997]. For example, if a user cares twice as much about one feature (color) as he does about another feature (shape), the overall similarity would be a linear combination of the two individual similarities with the weights being 2/3 and 1/3, respectively [Fagin and Wimmers, 1997]. Furthermore, because of the nature of linearity, these two levels can be combined into one, i.e.:

$$S = \sum_i \sum_j W_{ij} S(r_{ij}) \quad (22)$$

where W_{ij} 's are now *re-defined* to be the weights by which the information need in Q is distributed directly into r_{ij} 's. Note that it is not possible to absorb W_{ijk} into W_{ij} , since the calculation of $S(r_{ij})$ can be a non-linear function of W_{ijk} 's, such as Euclidean or Histogram Intersection.

3.4.4 Update of W_{ij}

The W_{ij} 's associated with the r_{ij} 's reflect the user's different emphasis of a representation in the overall similarity. The support of different weights enables the user to specify his or her information need more precisely. We will next discuss how to update W_{ij} 's according to the user's relevance feedback.

Let RT be the set of the most similar N_{RT} objects according to the overall similarity value S :

$$RT = [RT_1, \dots, RT_l, \dots, RT_{N_{RT}}] \quad (23)$$

Let $Score$ be the set containing the relevance scores fed-back by the user for RT_l 's (see section 3.4.3):

$$= 3, \quad \text{if highly relevant} \quad (24)$$

$$= 1, \quad \text{if relevant} \quad (25)$$

$$Score_l = 0, \quad \text{if no-opinion} \quad (26)$$

$$= -1, \quad \text{if non-relevant} \quad (27)$$

$$= -3, \quad \text{if highly non-relevant} \quad (28)$$

The choice of 3, 1, 0, -1, and -3 as the scores is arbitrary. Experimentally we find that the above scores capture the semantic meaning of *highly relevant*, *relevant*, etc. In Equations (24-28), we provide the user with five levels of relevance. Although more levels result in more accurate feedback, it is less convenient for the user to interact with the system. Experimentally we find that 5 levels is a good trade-off between convenience and accuracy.

For each r_{ij} , let RT^{ij} be the set containing the most similar N_{RT} objects to the query Q , according to the similarity values $S(r_{ij})$:

$$RT^{ij} = [RT_1^{ij}, \dots, RT_l^{ij}, \dots, RT_{N_{RT}}^{ij}] \quad (29)$$

To calculate the weight for r_{ij} , first initialize $W_{ij} = 0$, and then use the following procedure:

$$W_{ij} = W_{ij} + Score_l, \quad \text{if } RT_l^{ij} \text{ is in } RT \quad (30)$$

$$= W_{ij} + 0, \quad \text{if } RT_l^{ij} \text{ is not in } RT \quad (31)$$

$$l = 0, \dots, N_{RT} \quad (32)$$

Here, we consider all the images outside RT as marked with *no-opinion* and have the score of 0. After this procedure, if $W_{ij} < 0$, set it to 0. Let $W_{Tij} = \sum W_{ij}$ be the total weights. The raw weights obtained by the above procedure are then normalized by the total weight to make the sum of the normalized weight equal to 1.

$$W_{ij} = \frac{W_{ij}}{W_{Tij}} \quad (33)$$

As we can see, the more the overlap of relevant objects between RT and RT^{ij} , the larger the weight of W_{ij} . That is, if a representation r_{ij} reflects the user's information need, it receives more emphasis.

3.4.5 Update of W_{ijk}

The W_{ijk} 's associated with r_{ijk} 's reflect the different contributions of the components to the representation vector r_{ij} . For example, in the wavelet texture representation, we know that the mean of a sub-band may be corrupted by the lighting condition, while the standard deviation of a sub-band is independent of the lighting condition. Therefore more weight should be given to the standard deviation component, and less weight to the mean component. The support of different weights for r_{ijk} 's enables the system to have more reliable feature representation and thus better retrieval performance.

A standard deviation based weight updating approach has been proposed in our previous work [Rui et al., 1997a]. Out of the N_{RT} returned objects, for those objects that are marked with *highly relevant* or *relevant* by the user, stack their representation vector r_{ij} 's to form a $M' \times K$ matrix, where M' is the number of objects marked with *highly relevant* or *relevant*. In this way, each column of the matrix is a length- M' sequence of r_{ijk} 's. Intuitively, if all

the relevant objects have similar values for the component r_{ijk} , it means that the component r_{ijk} is a good indicator of the user’s information need. On the other hand, if the values for the component r_{ijk} are very different among the relevant objects, then r_{ijk} is not a good indicator. Based on this analysis, the inverse of the standard deviation of the r_{ijk} sequence is a good estimation of the weight W_{ijk} for component r_{ijk} . That is, the smaller the variance, the larger the weight and vice versa.

$$W_{ijk} = \frac{1}{\sigma_{ijk}} \quad (34)$$

where σ_{ijk} is the standard deviation of the length- M' sequence of r_{ijk} ’s. Here we assume that the user will mark at least one image, besides the query image, as relevant or highly relevant, such that σ_{ijk} will not be zero. The assumption is valid since otherwise the user would re-start a new query if nothing relevant is retrieved. Furthermore, just as in Equation (33), we need to normalize W_{ijk} ’s in the same way.

$$W_{ijk} = \frac{W_{ijk}}{W_{Tijk}} \quad (35)$$

where $W_{Tijk} = \sum W_{ijk}$.

4 Experimental Results

In the experiments reported here, we test our approaches over the image collection from the Fowler Museum of Cultural History at the University of California-Los Angeles. It contains 286 ancient African and Peruvian artifacts and is part of the Museum Educational Site Licensing Project (MESL), sponsored by the Getty Information Institute.

The size of the MESL test set is relatively small but it allows us to explore all the color, texture, and shape features simultaneously in a meaningful way. More extensive experiments with larger collections have been performed and reported in [Ortega et al., 1998b, Rui et al., 1998b].

In the following experiments, the visual features used are color, texture and shape of the objects in the image. That is,

$$F = \{f_i\} = \{\text{color, texture, shape}\} \quad (36)$$

The representations used are color histogram and color moments [Swain and Ballard, 1991] for the color feature; Tamura [Tamura et al., 1978, Equitz and Niblack, 1994] and co-occurrence matrix [Haralick et al., 1973, Ohanian and Dubes, 1992] texture representations for the texture feature, and Fourier descriptor and chamfer shape descriptor [Rui et al., 1997b] for the shape feature.

$$\begin{aligned} R = \{r_{ij}\} &= \{r_1, r_2, r_3, r_4, r_5, r_6\} \\ &= \{\text{color histogram, color moments, Tamura,} \\ &\quad \text{co-occurrence matrix, Fourier descriptor,} \\ &\quad \text{chamfer shape descriptor}\} \end{aligned}$$

Our proposed framework is open in that other visual features or feature representations can be easily incorporated, if needed. The similarity measures used for the corresponding representations are the following. Color Histogram Intersection [Swain and Ballard, 1991] is used for the color histogram representation; weighted Euclidean is used for the color moments, Tamura texture, co-occurrence matrix, and Fourier shape descriptor [Rui et al., 1997b] representations; and Chamfer matching [Rui et al., 1997b] is used for the chamfer shape representation.

4.1 Boolean Retrieval Model Results

To conduct the experiments we chose several queries and manually determined the relevant set of images with help of experts in librarianship as part of a seminar in multimedia retrieval. With the set of queries and relevant answers for each of them, we constructed precision-recall curves [Salton and McGill, 1983]. These are based on the well known precision and recall metrics. Precision measures the percentage of relevant answers and recall measures the percent of relevant objects returned to the user. The precision recall graphs are constructed by measuring the precision for various levels of recall.

We conducted experiments to verify the role of feature weighting in retrieval. Figure 7(a) shows results of a *shape or color* query i.e. to retrieve all images having either the same shape or the same color as the query image. We obtained four different precision recall curves by varying the feature weights. The retrieval performance improves when the shape feature receives more emphasis.

We also conducted experiments to observe the impact of the retrieval model used to evaluate the queries. We observed that the fuzzy and probabilistic interpretation of the same query yields different results. Figure 7(b) shows the performance of the same query (a *texture or color* query) in the two models. The result shows that neither model is consistently better than the other in terms of retrieval.

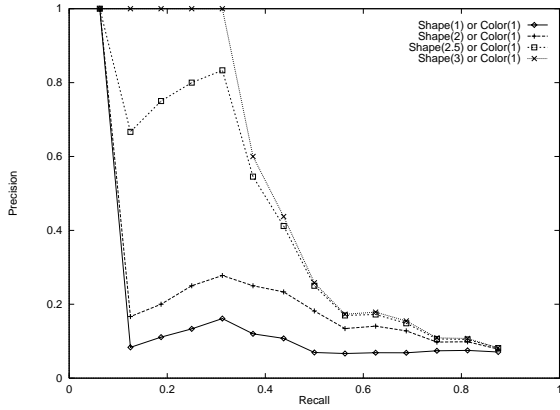
Figure 7(c) shows a complex query (*shape(I_i) and color(I_i) or shape(I_j) and layout(I_j)* query) with different weightings. The three weightings fared quite similar, which suggests that complex weightings may not have a significant effect on retrieval performance. We used the same complex query to compare the performance of the retrieval models. The result is shown in Figure 7(d). In general, the probabilistic model outperforms the fuzzy model.

4.2 Vector Retrieval Model with Relevance Feedback Results

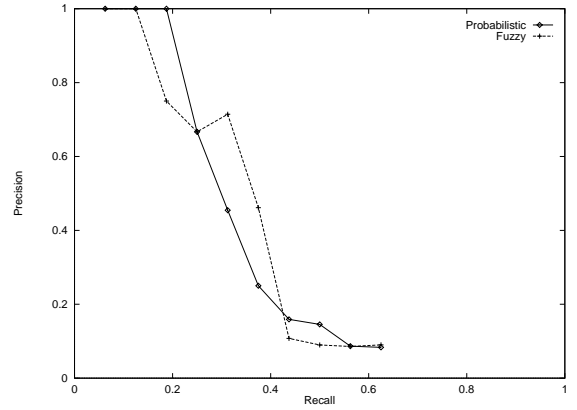
There are two sets of experiments reported here. The first set of experiments is on the efficiency of the retrieval algorithm, i.e. how fast the retrieval results converge to the true results. The second set of experiments is on the effectiveness of the retrieval algorithm, i.e. how good the retrieval results are subjectively.

4.2.1 Efficiency of the Algorithm

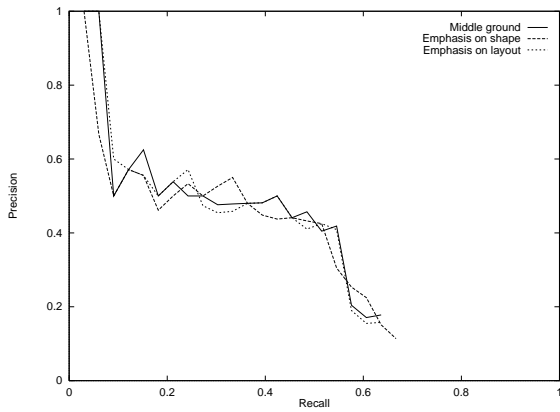
The ultimate goal of the relevance feedback technique is to help the user retrieve what he or she wants. Because of this, it is very important to verify that the above proposed relevance feedback retrieval algorithm converges to the user's true information need fast.



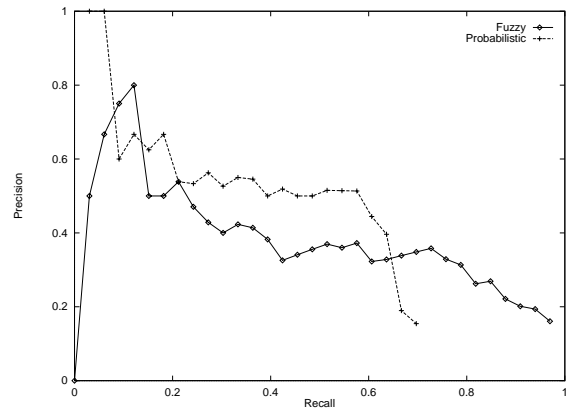
a) Effects of varying the weighting on a query



b) Fuzzy vs. Probabilistic performance for query



c) Complex query with different weights



d) Fuzzy vs. probabilistic for same complex query

Figure 7: Experimental result graphs

The only assumption that we make in the experiments is that the user is consistent when doing relevance feedback. That is, the user does not change his or her information need during the feedback process, such that a computer can simulate the feedback process.

As we have discussed in Section 3.1, the image object is modeled by the combinations of representations with their corresponding weights. If we fix the representations, then a query can be completely characterized by the set of weights embedded in the query object Q . Let set s_1 be the *highly relevant* set; set s_2 be the *relevant* set; set s_3 be the *no-opinion* set; set s_4 be the *non-relevant* set; and set s_5 be the *highly non-relevant* set. The testing procedure is described as follows:

1. Retrieval results of the ideal case.

Let W^* be the set of weights associated with the query object Q . The retrieval results based on W^* are the ideal case and serve as the baseline for comparing other non-ideal cases.

- (a) Specify a set of weights, W^* , to the query object.
- (b) Set $W = [W_{ij}, W_{ijk}]$ to W^* .
- (c) Invoke the retrieval algorithm.
- (d) Obtain the best N_{RT} returns, RT^* .
- (e) From RT^* , find the sizes of sets $s_i, i = 1, \dots, 5$, $n_i, i = 1, \dots, 5$. s_i 's are marked by the human user for testing purpose.
- (f) Calculate the ideal weighted relevant count as:

$$count^* = 3 \times n_1 + 1 \times n_2 \quad (37)$$

Note that 3 and 1 are the scores of the *highly relevant* and *relevant* sets, respectively (see section 3.4.3). Therefore, $count^*$ is the maximal achievable weighted relevant count and serves as the baseline for comparing other non-ideal cases.

2. Retrieval results of the relevance feedback case.

In the real retrieval situation, neither the user nor the computer knows the specified weights W^* . However, the proposed retrieval algorithm will move the initial weights W_0 to the ideal weights W^* via relevance feedback.

- (a) Set $W = W_0$.
- (b) Set the maximum number of iterations of relevance feedback, P_{fd} .
- (c) Initialize the iteration counter, $p_{fd} = 0$.
- (d) Invoke the retrieval algorithm and get back the best N_{RT} returns, $RT(p_{fd})$ (see Section 3).
- (e) Compute the weighted relevant count for the current iteration:

$$count(p_{fd}) = 3 \times n_1(p_{fd}) + 1 \times n_2(p_{fd}) \quad (38)$$

where $n_1(p_{fd})$ and $n_2(p_{fd})$ are the number of *highly relevant* and *relevant* objects in $RT(p_{fd})$. These two numbers can be determined by comparing $RT(p_{fd})$ against RT^* .

(f) Compute the convergence ratio $CR(p_{fd})$ for the current iteration:

$$CR(p_{fd}) = \frac{count(p_{fd})}{count^*} \times 100\% \quad (39)$$

- (g) Set $p_{fd} = p_{fd} + 1$. If $p_{fd} \geq P_{fd}$, quit; otherwise continue.
- (h) Feed the current 5 sets $s_i, i = 1, \dots, 5$, back into to the retrieval system.
- (i) Update the weights W according to Equations (28-35). Go to step 2(d).

In all the experiments reported here, 100 randomly selected images are used as the query images and the values of CR listed in the tables are the averages of the 100 cases. In this paper, we concentrate on the effect of W_{ij} . The effect of W_{ijk} has been studied in our previous research in [Rui et al., 1997a]. Specifically, only W_{ij} is specified for W^* in the experiments. In the MESL test set, there are 6 r_{ij} 's as described at the beginning of this section. Therefore, both W^* and $W0$ have 6 components. In addition,

$$W0 = \left[\frac{1}{6} \ \frac{1}{6} \ \frac{1}{6} \ \frac{1}{6} \ \frac{1}{6} \ \frac{1}{6} \right] \quad (40)$$

where each entry in the vector $W0$ is the weight for its corresponding representation.

Obviously, the retrieval performance is affected by the offset of the specified weights W^* from the initial weights $W0$. We classify W^* into two categories, i.e. moderate offset, and significant offset, by considering how far away they are from the initial weights $W0$.

The six moderate offset testing weights are:

$$\begin{aligned} W_1^* &= [0.5 \ 0.1 \ 0.1 \ 0.1 \ 0.1 \ 0.1] \\ W_2^* &= [0.1 \ 0.5 \ 0.1 \ 0.1 \ 0.1 \ 0.1] \\ W_3^* &= [0.1 \ 0.1 \ 0.5 \ 0.1 \ 0.1 \ 0.1] \\ W_4^* &= [0.1 \ 0.1 \ 0.1 \ 0.5 \ 0.1 \ 0.1] \\ W_5^* &= [0.1 \ 0.1 \ 0.1 \ 0.1 \ 0.5 \ 0.1] \\ W_6^* &= [0.1 \ 0.1 \ 0.1 \ 0.1 \ 0.1 \ 0.5] \end{aligned}$$

The six significant offset testing weights are:

$$\begin{aligned} W_7^* &= [0.75 \ 0.05 \ 0.05 \ 0.05 \ 0.05 \ 0.05] \\ W_8^* &= [0.05 \ 0.75 \ 0.05 \ 0.05 \ 0.05 \ 0.05] \\ W_9^* &= [0.05 \ 0.05 \ 0.75 \ 0.05 \ 0.05 \ 0.05] \\ W_{10}^* &= [0.05 \ 0.05 \ 0.05 \ 0.75 \ 0.05 \ 0.05] \\ W_{11}^* &= [0.05 \ 0.05 \ 0.05 \ 0.05 \ 0.75 \ 0.05] \\ W_{12}^* &= [0.05 \ 0.05 \ 0.05 \ 0.05 \ 0.05 \ 0.75] \end{aligned}$$

The experimental results for these cases are summarized in Figure 8. Based on the curves, some observations can be made:

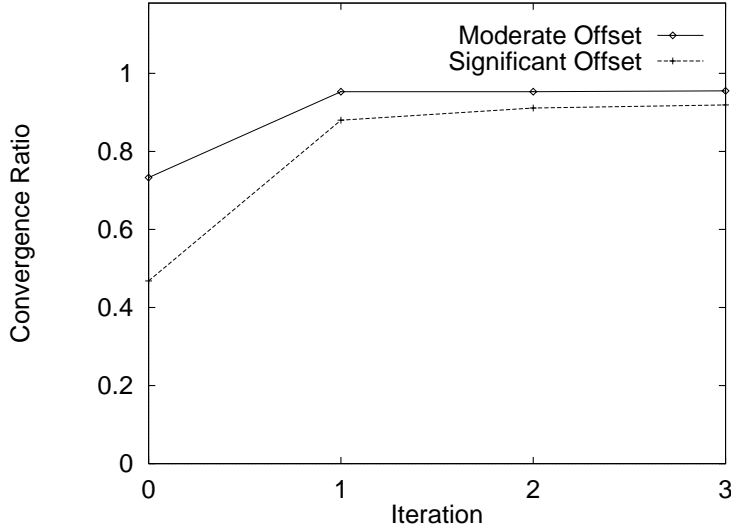


Figure 8: Convergence Ratio curves.

- In all the cases, CR increases the most in the first iteration. Later iterations only result in minor increases in CR . This is a very desirable property, which ensures that the user gets reasonable results after only one iteration of feedback. No further feedback iterations are needed, if time is a concern.
- CR is affected by the degree of offset. The less the offset, the higher the final absolute CR . However, the more the offset, the higher the relative increase of CR .

4.2.2 Effectiveness of the Algorithm

Experiments in the previous subsections focus on the convergence of the algorithm. This sub-section will focus on how good the results are *subjectively*. The only way of performing subjective tests is to ask the user to evaluate the retrieval system subjectively. Extensive experiments have been carried out. Users from various disciplines, such as Computer Vision, Art, Library Science, etc., as well as users from industry, have been invited to judge the retrieval performance of the proposed *interactive* approach. A typical retrieval process on the MESL test set is given in Figures 9 and 10.

The user can browse through the image database. Once he or she finds an image of interest, that image is submitted as a query. Alternatively to this query-by-example mode, the user can also submit images outside the database as queries. In Figure 9, the query image is displayed at the upper-left corner and the best 11 retrieved images, with $W = W_0$, are displayed in the order from top to bottom and from left to right. The retrieved results are obtained based on their overall similarities to the query image, which are computed from all the features and all the representations. Some retrieved images are similar to the query image in terms of the shape feature while others are similar to the query image in terms of color or texture feature.

Assume the user’s true information need is to “retrieve similar images based on their shapes”. In the proposed retrieval approach, the user is no longer required to explicitly map



Figure 9: The retrieval results before the relevance feedback



Figure 10: The retrieval results after the relevance feedback

his information need to low-level features, but rather he or she can express his intended information need by marking the relevance scores of the returned images. In this example, images 247, 218, 228 and 164 are marked *highly relevant*. Images 191, 168, 165, and 78 are marked *highly non-relevant*. Images 154, 152, and 273 are marked *no-opinion*.

Based on the information fed-back by the user, the system *dynamically* adjusts the weights, putting more emphasis on the *shape feature*, possibly even more emphasis to one of the two shape representations which better matches the user's subjective perception of shape. The improved retrieval results are displayed in Figure 10. Note that our shape representations are invariant to translation, rotation, and scaling. Therefore, images 164 and 96 are relevant to the query image.

5 Future Research Directions

We have conducted research in several areas. This paper presented a broad overview of our work on the retrieval model aspect of VIR. There is however much further work required. Specifically, we are developing an integrated multimedia retrieval model that allows queries not on single media, but truly on multimedia documents. A query would be a document containing text, one or more images, possibly audio and video data as well. Our work in the video domain [Rui et al., 1998a] provides a foundation for the video domain. Our current work revolves around complete integration of these disparate media types in a single unified model.

We are also actively pursuing research in the evaluation methods for such multimedia retrieval models. While our observations indicate that combining textual and visual retrieval enhances retrieval performance, there is no hard evidence. Our initial work on metrics designed specifically for such models incorporating multiple media types is promising [Ortega et al., 1998a], but still requires much further work. How to reliably measure the extent to which incorporating distinct media types enhances retrieval is an important research area.

To complement the research presented, we are also working on the efficient implementation of these retrieval models [Chakrabarti and Mehrotra, 1999], and their incorporation into database systems.

6 Conclusion

In this paper we have discussed techniques to extent information retrieval beyond the text document. Specifically, we have discussed how to extract visual features from images and video; how to adapt a Boolean retrieval model (enhanced with Fuzzy and Probabilistic concepts) for VIR systems; and how to generalize the relevance feedback technique to VIR.

In the past decade, two general approaches to VIR emerged. One is based on text (titles, keywords, and annotation) to search for visual information indirectly. This paradigm requires much human labor and suffers from vocabulary inconsistency problems across human indexers. The other paradigm intends to build fully automated systems by completely discarding the text information and performing the search on visual information only. Neither paradigm has been very successful. In our view, these two paradigms have both their

advantages and disadvantages; and sometimes are complimentary to each other. For example, in the MESL database, it will be much more meaningful if we first do a text-based search to confine the category and then use visual feature based search to refine the result. Another promising research direction is the integration of the human user into the retrieval system loop. A fundamental difference between an old Pattern Recognition system and today's VIR system is that the end user of the latter is human. By integrating human knowledge into the retrieval process, we can bypass the unsolved problem of image understanding. Relevance feedback is one technique designed to deal with this problem.

References

- [Arkin et al., 1991] Arkin, E. M., Chew, L., Huttenlocher, D., Kedem, K., and Mitchell, J. (1991). An efficiently computable metric for comparing polygonal shapes. *IEEE Trans. Patt. Recog. and Mach. Intell.*, 13(3).
- [Bach et al., 1996] Bach, J. R., Fuller, C., Gupta, A., Hampapur, A., Horowitz, B., Humphrey, R., Jain, R., and fe Shu, C. (1996). The Virage image search engine: An open framework for image management. In *Proc. SPIE Storage and Retrieval for Image and Video Databases*.
- [Barrow, 1977] Barrow, H. G. (1977). Parametric correspondence and chamfer matching: Two new techniques for image matching. In *Proc 5th Int. Joint Conf. Artificial Intelligence*.
- [Beyer et al., 1998] Beyer, K., Goldstein, J., Ramakrishnan, R., and Shaft, U. (1998). When Is “Nearest Neighbor” Meaningful? *Submitted for publication*.
- [Borgefors, 1988] Borgefors, G. (1988). Hierarchical chamfer matching: A parametric edge matching algorithm. *IEEE Trans. Patt. Recog. and Mach. Intell.*
- [Buckley and Salton, 1995] Buckley, C. and Salton, G. (1995). Optimization of relevance feedback weights. In *Proc. of SIGIR'95*.
- [Chakrabarti and Mehrotra, 1999] Chakrabarti, K. and Mehrotra, S. (1999). High dimensional feature indexing using hybrid trees. In *Proceedings of the International Conference on Data Engineering (ICDE) Sydney, Australia*.
- [Chang and Kuo, 1993] Chang, T. and Kuo, C.-C. J. (1993). Texture analysis and classification with tree-structured wavelet transform. *IEEE Trans. Image Proc.*, 2(4):429–441.
- [Chaudhari and Gravano, 1996] Chaudhari, S. and Gravano, L. (1996). Optimizing Queries over Multimedia Repositories. *Proc. of SIGMOD*.
- [Chua et al., 1997] Chua, T. S., Tan, K.-L., and Ooi, B. C. (1997). Fast signature-based color-spatial image retrieval. In *Proc. IEEE Conf. on Multimedia Computing and Systems*.
- [Chuang and Kuo, 1996] Chuang, G. C.-H. and Kuo, C.-C. J. (1996). Wavelet descriptor of planar curves: Theory and applications. *IEEE Trans. Image Proc.*, 5(1):56–70.
- [Equitz and Niblack, 1994] Equitz, W. and Niblack, W. (1994). Retrieving images from a database using texture – algorithms from the QBIC system. Technical Report RJ 9805, Computer Science, IBM Research Report.
- [Fagin, 1996] Fagin, R. (1996). Combining Fuzzy Information from Multiple Systems. *Proc. of the 15th ACM Symp. on PODS*.
- [Fagin and Wimmers, 1997] Fagin, R. and Wimmers, E. L. (1997). Incorporating user preferences in multimedia queries. In *Proc of Int. Conf. on Database Theory*.
- [Faloutsos et al., 1993] Faloutsos, C., Flickner, M., Niblack, W., Petkovic, D., Equitz, W., and Barber, R. (1993). Efficient and effective querying by image content. Technical report, IBM Research Report.
- [Flickner et al., 1995] Flickner, M., Sawhney, H., Niblack, W., Ashley, J., Huang, Q., Dom, B., Gorkani, M., Hafine, J., Lee, D., Petkovic, D., Steele, D., and Yanker, P. (1995). Query by image and video content: The QBIC system. *IEEE Computer*.
- [Foley et al., 1990] Foley, J., van Dam, A., Feiner, S., and Hughes, J. (1990). *Computer Graphics*. Addison Wesley, 2nd edition.

- [Gotlieb and Kreyszig, 1990] Gotlieb, C. C. and Kreyszig, H. E. (1990). Texture descriptors based on co-occurrence matrices. *Computer Vision, Graphics, and Image Processing*, 51.
- [Gross and Latecki, 1995] Gross, A. and Latecki, L. (1995). Digital geometric invariance and shape representation. In *Proc. of Int'l Symposium on Computer Vision*, Coral Gables, FL.
- [Gross et al., 1994] Gross, M. H., Koch, R., Lippert, L., and Dreger, A. (1994). Multiscale image texture analysis in wavelet spaces. In *Proc. IEEE Int. Conf. on Image Proc.*
- [Haralick et al., 1973] Haralick, R. M., Shanmugam, K., and Dinstein, I. (1973). Texture features for image classification. *IEEE Trans. on Sys, Man, and Cyb*, SMC-3(6).
- [Hu, 1962] Hu, M. K. (1962). Visual pattern recognition by moment invariants, computer methods in image analysis. *IRE Transactions on Information Theory*, 8.
- [Huang et al., 1997] Huang, J., Kumar, S., Mitra, M., Zhu, W.-J., and Zabih, R. (1997). Image indexing using color correlogram. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*.
- [Huang et al., 1996] Huang, T. S., Mehrotra, S., and Ramchandran, K. (1996). Multimedia analysis and retrieval system (MARS) project. In *Proc of 33rd Annual Clinic on Library Application of Data Processing - Digital Image Access and Retrieval*.
- [Ioka, 1989] Ioka, M. (1989). A method of defining the similarity of images on the basis of color information. Technical Report RT-0030, IBM Research, Tokyo Research Laboratory.
- [Kapur et al., 1995] Kapur, D., Lakshman, Y. N., and Saxena, T. (1995). Computing invariants using elimination methods. In *Proc. IEEE Int. Conf. on Image Proc.*
- [Kundu and Chen, 1992] Kundu, A. and Chen, J.-L. (1992). Texture classification using qmf bank-based subband decomposition. *CVGIP: Graphical Models and Image Processing*, 54(5):369–384.
- [Laine and Fan, 1993] Laine, A. and Fan, J. (1993). Texture classification by wavelet packet signatures. *IEEE Trans. Patt. Recog. and Mach. Intell.*, 15(11):1186–1191.
- [Lu et al., 1994] Lu, H., Ooi, B., and Tan, K. (1994). Efficient image retrieval by color contents. In *Proc. of the 1994 Int. Conf. on Applications of Databases*.
- [McCamy et al., 1976] McCamy, C. S., Marcus, H., and Davidson, J. G. (1976). A color-rendition chart. *Journal of Applied Photographic Engineering*, 2(3).
- [Miyahara, 1988] Miyahara, M. (1988). Mathematical transform of (r,g,b) color data to munsell (h,s,v) color data. In *SPIE Visual Communications and Image Processing*, volume 1001.
- [Niblack et al., 1994] Niblack, W., Barber, R., and et al. (1994). The QBIC project: Querying images by content using color, texture and shape. In *Proc. SPIE Storage and Retrieval for Image and Video Databases*.
- [Ohanian and Dubes, 1992] Ohanian, P. P. and Dubes, R. C. (1992). Performance evaluation for four classes of texture features. *Pattern Recognition*, 25(8):819–833.
- [Ortega et al., 1998a] Ortega, M., Chakrabarti, K., Porkaew, K., and Mehrotra, S. (1998a). Cross media validation in a multimedia retrieval system. *ACM Digital Libraries 98 Workshop on Metrics in Digital Libraries*.
- [Ortega et al., 1997] Ortega, M., Rui, Y., Chakrabarti, K., Mehrotra, S., and Huang, T. S. (1997). Supporting similarity queries in MARS. In *Proc. of ACM Conf. on Multimedia*.
- [Ortega et al., 1998b] Ortega, M., Rui, Y., Chakrabarti, K., Porkaew, K., Mehrotra, S., and Huang, T. S. (1998b). Supporting ranked boolean similarity queries in mars. *IEEE Trans. on Knowledge and Data Engineering*, 10(6).
- [Pass et al., 1996] Pass, G., Zabih, R., and Miller, J. (1996). Comparing images using color coherence vectors. In *Proc. ACM Conf. on Multimedia*.
- [Pentland et al., 1996] Pentland, A., Picard, R. W., and Sclaroff, S. (1996). Photobook: Content-based manipulation of image databases. *International Journal of Computer Vision*.
- [Persoon and Fu, 1977] Persoon, E. and Fu, K. S. (1977). Shape discrimination using fourier descriptors. *IEEE Trans. Sys. Man, Cyb*.
- [Rickman and Stonham, 1996] Rickman, R. and Stonham, J. (1996). Content-based image retrieval using colour tuple histograms. In *Proc. SPIE Storage and Retrieval for Image and Video Databases*.

- [Rui et al., 1997a] Rui, Y., Huang, T. S., and Mehrotra, S. (1997a). Content-based image retrieval with relevance feedback in MARS. In *Proc. IEEE Int. Conf. on Image Proc.*
- [Rui et al., 1998a] Rui, Y., Huang, T. S., and Mehrotra, S. (1998a). Exploring video structures beyond the shots. In *Proc. of IEEE conf. Multimedia Computing and Systems.*
- [Rui et al., 1997b] Rui, Y., Huang, T. S., Mehrotra, S., and Ortega, M. (1997b). Automatic matching tool selection using relevance feedback in MARS. In *Proc. of 2nd Int. Conf. on Visual Information Systems.*
- [Rui et al., 1998b] Rui, Y., Huang, T. S., Ortega, M., and Mehrotra, S. (1998b). Relevance feedback: A power tool in interactive content-based image retrieval. *IEEE Tran on Circuits and Systems for Video Technology*, 8(5).
- [Rui et al., 1996] Rui, Y., She, A. C., and Huang, T. S. (1996). Modified fourier descriptors for shape representation – a practical approach. In *Proc of First International Workshop on Image Databases and Multi Media Search.*
- [Salton and McGill, 1983] Salton, G. and McGill, M. J. (1983). *Introduction to Modern Information Retrieval.* McGraw-Hill Book Company.
- [Shaw, 1995] Shaw, W. M. (1995). Term-relevance computations and perfect retrieval performance. *Information Processing and Management*, 31(4).
- [Smith and Chang, 1994] Smith, J. R. and Chang, S.-F. (1994). Transform features for texture classification and discrimination in large image databases. In *Proc. IEEE Int. Conf. on Image Proc.*
- [Smith and Chang, 1995a] Smith, J. R. and Chang, S.-F. (1995a). Single color extraction and image query. In *Proc. IEEE Int. Conf. on Image Proc.*
- [Smith and Chang, 1995b] Smith, J. R. and Chang, S.-F. (1995b). Tools and techniques for color image retrieval. In *IS & T/SPIE proceedings Vol.2670, Storage & Retrieval for Image and Video Databases IV.*
- [Smith and Chang, 1996] Smith, J. R. and Chang, S.-F. (1996). Automated binary texture feature sets for image retrieval. In *Proc ICASSP-96, Atlanta, GA.*
- [Stricker and Dimai, 1996] Stricker, M. and Dimai, A. (1996). Color indexing with weak spatial constraints. In *Proc. SPIE Storage and Retrieval for Image and Video Databases.*
- [Stricker and Orenco, 1995] Stricker, M. and Orenco, M. (1995). Similarity of color images. In *Proc. SPIE Storage and Retrieval for Image and Video Databases.*
- [Swain and Ballard, 1991] Swain, M. and Ballard, D. (1991). Color indexing. *International Journal of Computer Vision*, 7(1).
- [Tamura et al., 1978] Tamura, H., Mori, S., and Yamawaki, T. (1978). Texture features corresponding to visual perception. *IEEE Trans. on Sys, Man, and Cyb*, SMC-8(6).
- [Thyagarajan et al., 1994] Thyagarajan, K. S., Nguyen, T., and Persons, C. (1994). A maximum likelihood approach to texture classification using wavelet transform. In *Proc. IEEE Int. Conf. on Image Proc.*
- [Wang et al., 1997] Wang, J., Yang, W.-J., and Acharya, R. (1997). Color clustering techniques for color-content-based image retrieval from image databases. In *Proc. IEEE Conf. on Multimedia Computing and Systems.*
- [Yang and Algreghsen, 1994] Yang, L. and Algreghsen, F. (1994). Fast computation of invariant geometric moments: A new method giving correct results. In *Proc. IEEE Int. Conf. on Image Proc.*
- [Zahn and Roskies, 1972] Zahn, C. T. and Roskies, R. Z. (1972). Fourier descriptors for plane closed curves. *IEEE Trans. on Computers.*

Yong Rui received the B.S. degree from Southeast University, P. R. China in 1991 and the M.S. degree from Tsinghua University, P. R. China in 1994, both in Electrical Engineering. He is currently finishing the Ph.D. degree in Electrical and Computer Engineering at the University of Illinois at Urbana-Champaign. Since 1995 he has been a Graduate Research Assistant at the Image Formation and Processing Group of the Beckman Institute for Advance Science and Technology at UIUC.

His research interests include multimedia information retrieval, multimedia signal processing, computer vision and artificial intelligence. He has published over 30 technical papers in the above areas.

He is a Huitong University Fellowship recipient 1989-1990, a Guanghua University Fellowship recipient 1992-1993, and a CSE Engineering College Fellowship recipient 1996-1998.

Michael Ortega Received his B.E. degree with honors from the Mexican Autonomous Institute of Technology in Aug. 1994 with a SEP fellowship for the duration of the studies. Currently he is pursuing his graduate studies at the University of Illinois at Urbana Champaign. Michael Ortega received a Fulbright/CONACYT/García Robles scholarship to pursue graduate studies as well as the Mavis Award at the University of Illinois and is a member of the Phi Kappa Phi honor society, the IEEE computer society and member of the ACM. His research interests include multimedia databases, database optimization for uncertainty support and content based multimedia information retrieval.

Thomas S. Huang received his B.S. Degree in Electrical Engineering from National Taiwan University, Taipei, Taiwan, China; and his M.S. and Sc.D. Degrees in Electrical Engineering from the Massachusetts Institute of Technology, Cambridge, Massachusetts. He was on the Faculty of the Department of Electrical Engineering at MIT from 1963 to 1973; and on the Faculty of the School of Electrical Engineering and Director of its Laboratory for Information and Signal Processing at Purdue University from 1973 to 1980. In 1980, he joined the University of Illinois at Urbana-Champaign, where he is now William L. Everitt Distinguished Professor of Electrical and Computer Engineering, and Research Professor at the Coordinated Science Laboratory, and Head of the Image Formation and Processing Group at the Beckman Institute for Advanced Science and Technology.

During his sabbatical leaves, Dr. Huang has worked at the MIT Lincoln Laboratory, the IBM Thomas J. Watson Research Center, and the Rheinishes Landes Museum in Bonn, West Germany, and held visiting Professor positions at the Swiss Institutes of Technology in Zurich and Lausanne, University of Hannover in West Germany, INRS-Telecommunications of the University of Quebec in Montreal, Canada and University of Tokyo, Japan. He has served as a consultant to numerous industrial firms and government agencies both in the U.S. and abroad.

Dr. Huang's professional interests lie in the broad area of information technology, especially the transmission and processing of multidimensional signals. He has published 12 books, and over 300 papers in Network Theory, Digital Filtering, Image Processing, and Computer Vision. He is a Fellow of the International Association of Pattern Recognition, IEEE, and the Optical Society of American; and has received a Guggenheim Fellowship, an A.V. Humboldt Foundation Senior U.S. Scientist Award, and a Fellowship from the Japan Association for the Promotion of Science. He received the IEEE Acoustics, Speech, and Signal Processing Society's Technical Achievement Award in 1987, and the Society Award in 1991. He is a Founding Editor of the International Journal Computer Vision, Graphics, and Image Processing; and Editor of the Springer Series in Information Sciences, published by Springer Verlag.

Sharad Mehrotra received his M.S. and PhD at the University of Texas at Austin in 1990 and 1993 respectively, both in Computer Science. Subsequently he worked at MITL, Princeton as a scientist from 1993-1994. He is an assistant professor in the Computer Science department at the University of Illinois at Urbana-Champaign since 1994. He specializes in the areas of database management, distributed systems, and information retrieval. His current research projects are on multimedia analysis, content-based retrieval of multimedia objects, multidimensional indexing, uncertainty management in databases, and concurrency and transaction management. Dr. Mehrotra

is an author of over 50 research publications in these areas. Dr. Mehrotra is the recipient of the NSF Career Award and the Bill Gear Outstanding junior faculty award in 1997.