Information Retrieval Beyond the Text Document

YONG RUI, MICHAEL ORTEGA, THOMAS S. HUANG, AND
SHARAD MEHROTRA

ABSTRACT
WITH THE EXPANSION OF THE INTERNET, searching for information goes
beyond the boundary of physical libraries. Millions of documents of vari-
ous media types—such as text, image, video, audio, graphics, and anima-
tion—are available around the world and linked by the Internet. Unfor-
tunately, 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 article reports some of the progress made over the years toward ex-
ploring information retrieval beyond the text domain. In particular, the
following aspects of MARS are addressed in the article: visual feature ex-
traction, retrieval models, query reformulation techniques, efficient ex-
ecution speed performance, and user interface considerations. Extensive
experimental results are reported to validate the proposed approaches.

INTRODUCTION

Huge amounts of digital data are being generated daily. Scanners
convert the analog/physical data into digital form; digital cameras and
camcorders directly generate digital data at the production phase. Owing
to all these multimedia devices, presently information is in all media types,
including graphics, images, audio, and video in addition to the conven-
tional text media type. Not only is multimedia information being generated

Yong Rui, Microsoft Research, One Microsoft Way, Redmond, WA 98052
Michael Ortega, 444 Computer Science, University of California, Irvine, CA 92697-3425
Thomas S. Huang, Beckman Institute for Advanced Science and Technology, University of
Illinois, Urbana, IL 61801
Sharad Mehrotra, Department of Information and Computer Science, University of Cali-
fornia, Irvine, CA 92697-3425

at an ever-increasing rate, it is transmitted worldwide 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 nontextual 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 article reports some of the progress made over the years toward exploring information retrieval (IR) beyond the text domain. In particular, the discussion will concentrate on visual information retrieval (VIR) concepts as opposed to implementation issues. MARS explores many different visual feature representations. A review of these features appears in the next section ("Visual Feature Extraction"). These visual features are analogous to keyword features in textual media. Another section ("Retrieval Models Used in MARS") 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. Results are given in a later section ("Experimental Results"). The last section provides remarks summarizing the overall discussion ("Conclusion").

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. There are many techniques for reliably extracting, for example, keywords from text documents. The *visual counterparts* to textual features in visual based systems are features such as color, texture, and shape.

For each feature, there are several different techniques for representation. The reason for this is twofold: (1) the field is still under development and, more importantly, (2) features are perceived differently by people and thus different representations cater to various 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. Rui et al. (1999) give a detailed description of the visual features, and the following paragraphs emphasize the important ones.

The *color* feature is one of the most widely used visual features in VIR. This 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) and Miyahara (1988). In VIR, color histograms (Swain & Ballard, 1991), color moments (Stricker & Orengo, 1995), and color sets (Smith & Chang, 1995) are the most used representations.

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. It is an innate property of virtually all surfaces, including clouds, trees, bricks, hair, fabric, and so on. It contains important information about the structural arrangement of surfaces and their relationship to the surrounding environment (Haralick et al., 1973). Co-occurrence matrix (Haralick et al., 1973), Tamura texture (Tamura et al., 1978), and Wavelet texture (Kundu & Chen, 1992) are the most popular texture representations.

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. 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 & Kuo, 1996).

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.

The 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 & Ballard, 1991).

Note that, each representation r_{ij} itself may be a vector consisting of multiple components, that is:

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

where K is the length of the vector.

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 (see section entitled "Visual Feature Extraction") extracted automatically using computer vision methods as well as a manual text description of the image.

Figure 1. The Object Model.

Each feature representation is associated with some similarity measure. 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 that they are very similar and a value of 0 means that they are very dissimilar. Revisiting our blue sky and ocean example from the early section ("Visual Feature Extraction"), 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 $M=\{<feature\ representation,$

similarity measure$_i$>, ...} together with the object model *O,* forms *(D, F, R, M),* a foundation on which query models can be built.

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 adaptation 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. The section on Boolean retrieval describes the Boolean model and the section on the "Vector Model" describes that model.

Boolean Retrieval

A user may not only be interested in more than a single feature from a single image. It is very likely that the user may choose multiple features from multiple 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. To cope with composite queries, a Boolean retrieval model is used to interpret the query and retrieve a set of images ranked based on their similarity to the selected feature.

The basic Boolean retrieval model needs a pre-defined threshold, which has several potential problems (Ortega et al., 1998b). To overcome these 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.

In the discussion below, we will use the following notations. Images in the collection are denoted by $I_1, I_2, ... I_m$. Features over the images are denoted by $F_1, F_2, ... F_r$, where F_i denotes both the name of the feature as well as the domain of values that the feature can take. The jth 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.

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_{ij}). Note that $F_i(I_j) = f_{ij}$. 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 as 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 $<I, S_{F2}(F_2(I), f_{2,5})>$ with S_{F2} 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, nonleaf nodes are one of three forms: $(v_1, v_2, \dots v_n)$: a conjunction of positive literals; $(v_1, v_2, \dots v_p, v_{p+1}, \dots v_n)$, a conjunction consisting of both positive and negative literals; and $(v_1, v_2, \dots v_n)$, which is a disjunction of positive literals. 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.
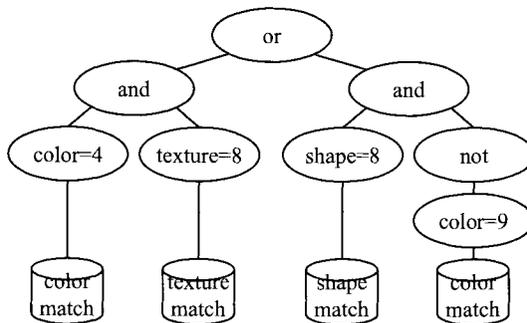
Operators: And, Or, Not
Basic features and representations:
Color histogram, color moment, wavelet texture, ...

Figure 2. Sample Query Tree.

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 *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. Fagin and Wimmers (1997) noted 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. Fagin and Wimmers (1997) 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] → [0,1] of the form:

$$similarity' = similarity^{\frac{1}{weight}}, 0 < weight < \infty \tag{3}$$

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, a weight of 0 means the node can be dismissed. Here, $\lim_{weight \to 0} similarity' = 0$ 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.

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

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 & Wimmers, 1997). A function *F* is monotonic if $F(x_1, ... x_m) \leq F(x'_1, ... x'_m)$ for $x_i \leq x'_i$ for every *i*. Note that the scoring functions for both conjunctive and disjunctive queries for 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).
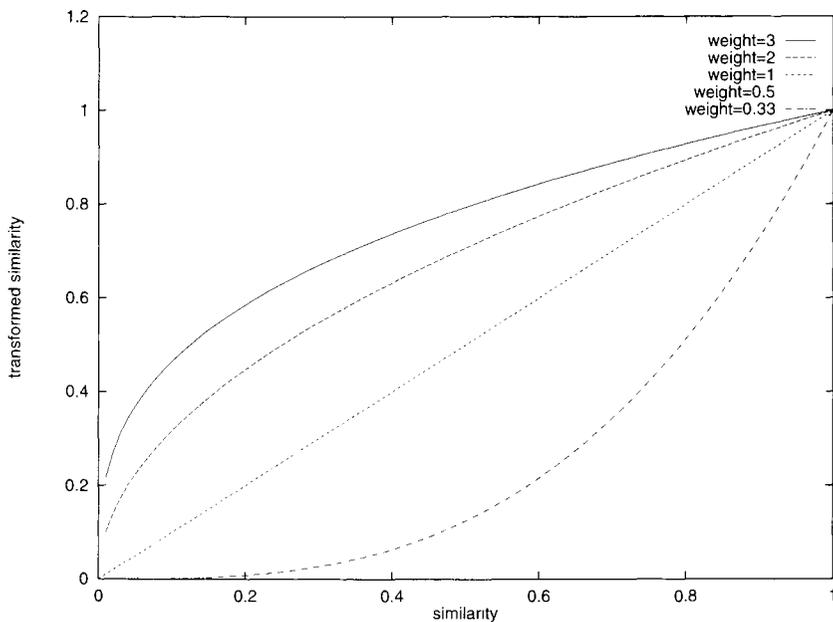
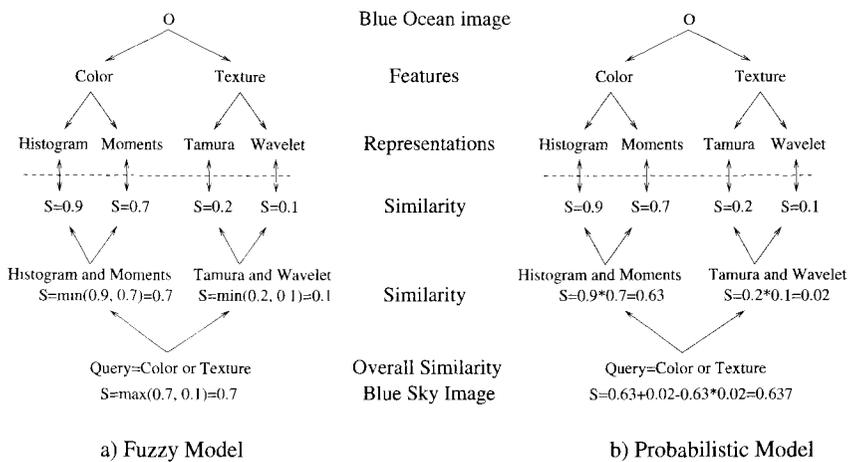Figure 3. Various Samples for Similarity Mappings.

Figure 4. Various Samples for Similarity Mappings.

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 models in MARS. This is significant since the experimental results illustrate that the probabilistic model outperforms the fuzzy model in terms of retrieval performance, which is discussed in a later section ("Experimental Results").

Vector Model

An information retrieval 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 & Salton, 1995; Salton & McGill, 1983; Shaw, 1995). Various effective retrieval techniques have been developed for this model. Among these, *term weighting* and *relevance feedback* are of fundamental importance.

Term weighting is a technique for assigning different weights for different keywords (terms) according to their relative importance to the document (Shaw, 1995; Salton & McGill, 1983). If we define w_{ik} to be the weight for term t_k, $k = 1, ..., N$, 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:

$$D_i = [w_{i1}, ... w_{ik}, ... w_{iN}] \tag{4}$$

Experiments have shown that the product of *tf* (term frequency) and *idf* (inverse document frequency) is a good estimation of the weights (Buckley & Salton, 1995; Salton & 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:

$$Q = [w_{q1}, ... w_{qk}, ... w_{qN}] \tag{5}$$

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

$$similarity \ (D, Q) = \frac{D \times Q}{||D|| \times ||Q||} \tag{6}$$

where || || denotes norm-2.

As we can see from the previous subsection ("Computing Boolean Queries"), 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 precisely his information need into a set of terms. To overcome this difficulty, the technique of *relevance feedback* has been proposed (Buckley & Salton, 1995; Salton & McGill, 1983; Shaw, 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. Term weighting and relevance feedback are powerful techniques in IR. We next generalize these concepts to VIR.

Vector Query Model and Integration of Relevance Feedback to VIR

As discussed in a previous section ("The Object Model"), 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 distance is used for comparing vector-based representations, while Histogram Intersection is used for comparing color histogram representations (see the earlier section on "Visual Feature Extraction").

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
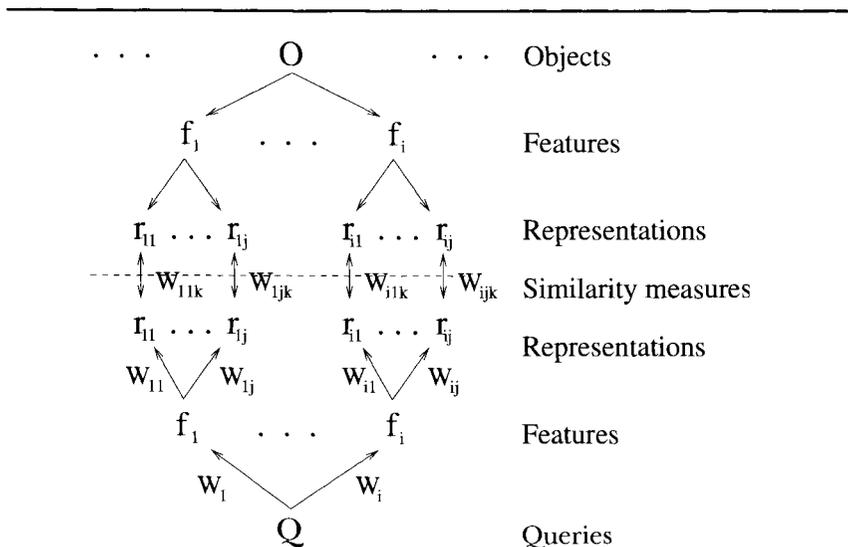
Figure 5. The Retrieval Process.

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 express the user's 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 can be described as follows. At the initial query stage, equal weights are associated with the features, representations, and components. Best matches are then displayed back to the user. Depending on his true information need, the user will mark how good the returned matches are (degree of relevance). Based on the user's feedback, the retrieval system will automatically update weights to match the user's true information need. This process is illustrated in Figure 5. In Figure 5, the information need embedded in Q flows up while the content of O's flows
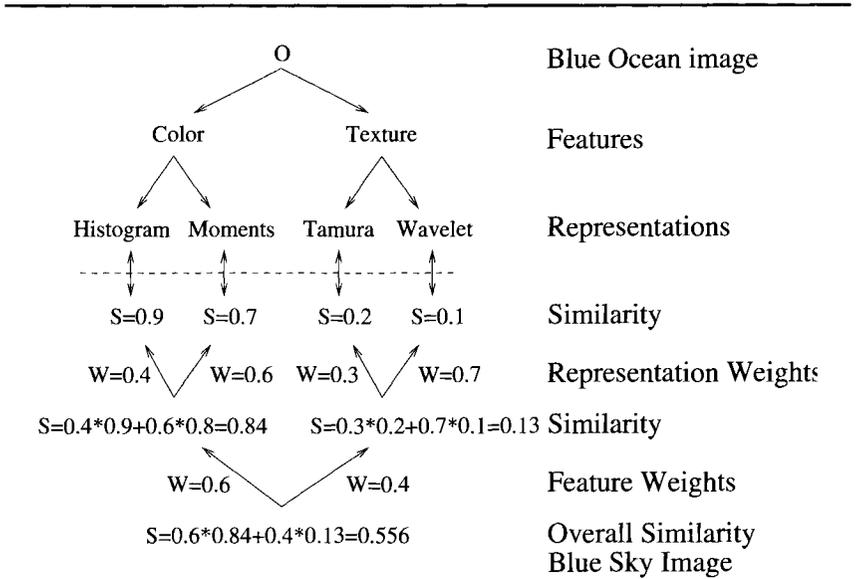

Figure 6. Example Query Calculation of Blue Sky Image against Blue Ocean Image.

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.

Based on the intuition that important representations or components should receive more weight, we have proposed effective algorithms for updating these two levels' weights. Due to page limitation, we refer the readers to Rui et al. (1998b).

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) and Rui et al. (1998b).

In the following experiments, the visual features used are color, texture, and shape of the objects in the image. The representations used are color histogram and color moments (Swain & Ballard, 1991), for the color feature Tamura (Tamura et al., 1978; Equitz & Niblack, 1994), and co-occurrence matrix (Haralick et al., 1973; Ohanian & Dubes, 1992) texture representations for the texture feature, and Fourier descriptor and chamfer shape descriptor (Rui et al., 1997b) for the shape feature.

Boolean Retrieval Model Results

To conduct the experiments, we chose several queries and manually determined the relevant set of images with the 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 & McGill, 1983). These are based on the well-known precision and recall metrics. Precision measures the percentage of relevant answers, and recall measures the percentage 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 interpretations of the same query yield 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_i) *and* layout (I_j)) with different weightings. The three weightings fared quite similarly, 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.

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.

Efficiency of the Algorithm

As we have discussed in the section "The Object Model," the image object is modeled by the combinations of representations with their corresponding weights. If we fix the representations, then a query can be
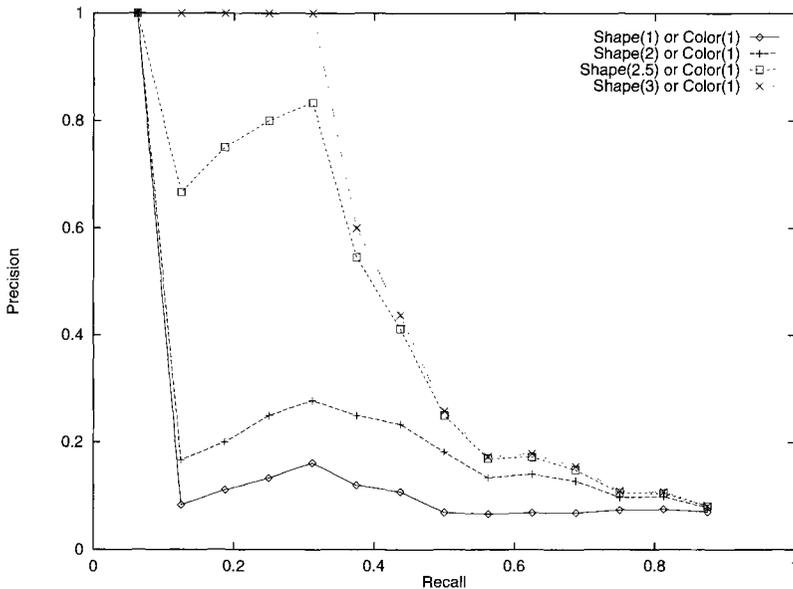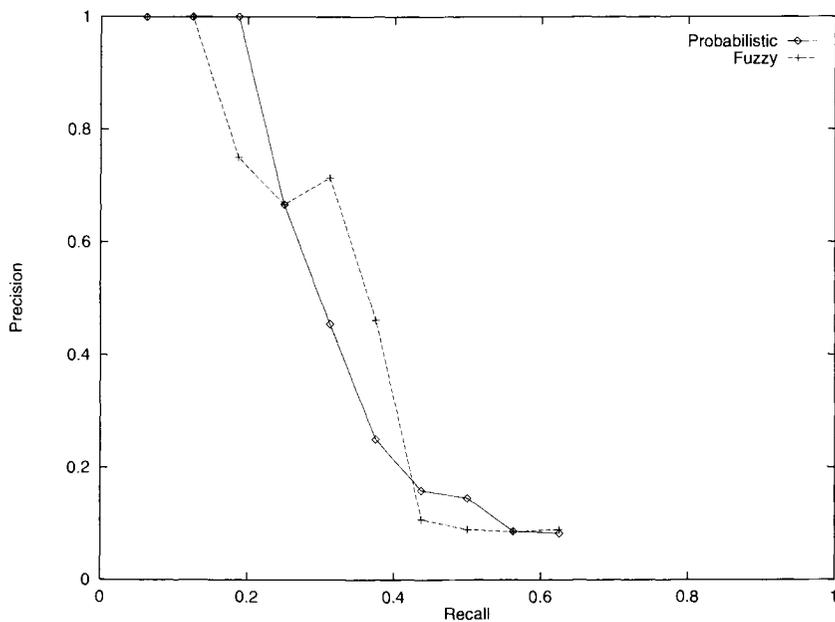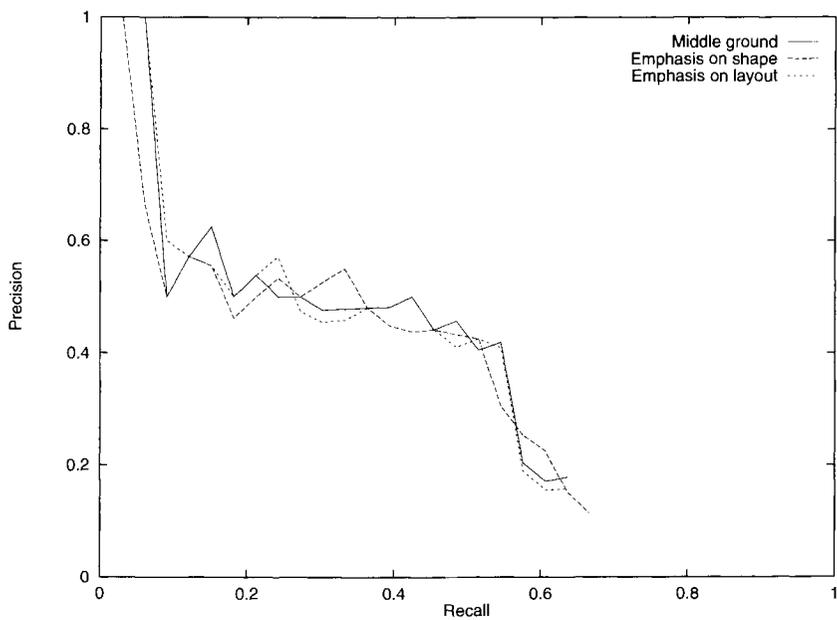

Figure 7a. Effects of Varying the Weighting on a Query.

7b. Fuzzy Versus Probabilistic Performance for Query.

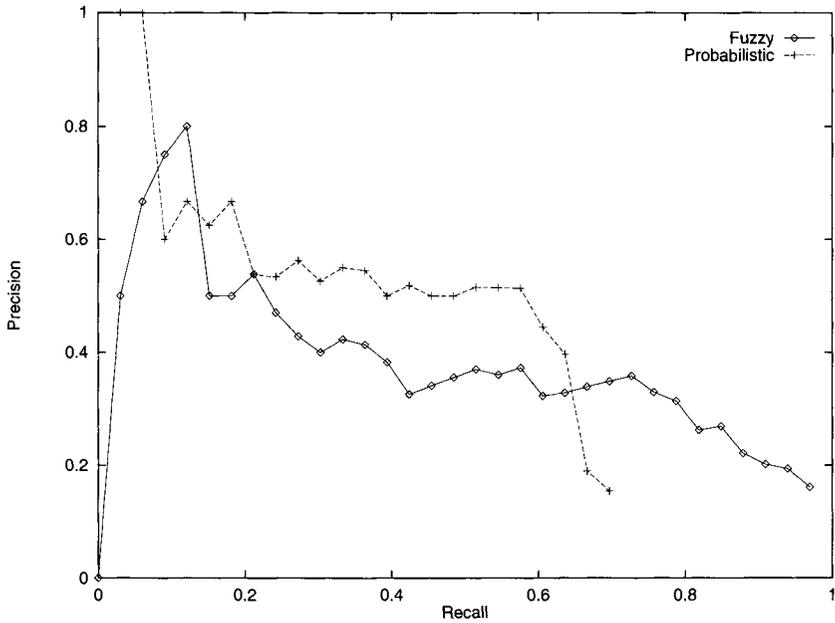7c. Complex Query with Different Weights.

Figure 7d. Fuzzy Versus Probabilistic for SameComplex Query.

completely characterized by the set of weights embedded in the query object Q. Obviously, the retrieval performance is affected by the offset of the true weights from the initial weights. We thus classify the test into two categories—i.e., moderate offset and significant offset—by considering how far away the true weights are from the initial weights. The convergence ratio (recall) for these cases is summarized in Figure 8. Based on the curves, some observations can be made:

- In all the cases, the convergence ratio (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.
- CR is affected by the degree of offset. The lower the offset, the higher the final absolute CR. However, the more the offset, the higher the relative increase of CR.

Effectiveness of the Algorithm
    Extensive experiments have been carried out. Users from various disciplines, such as computer vision, art, library science, and so on, 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.
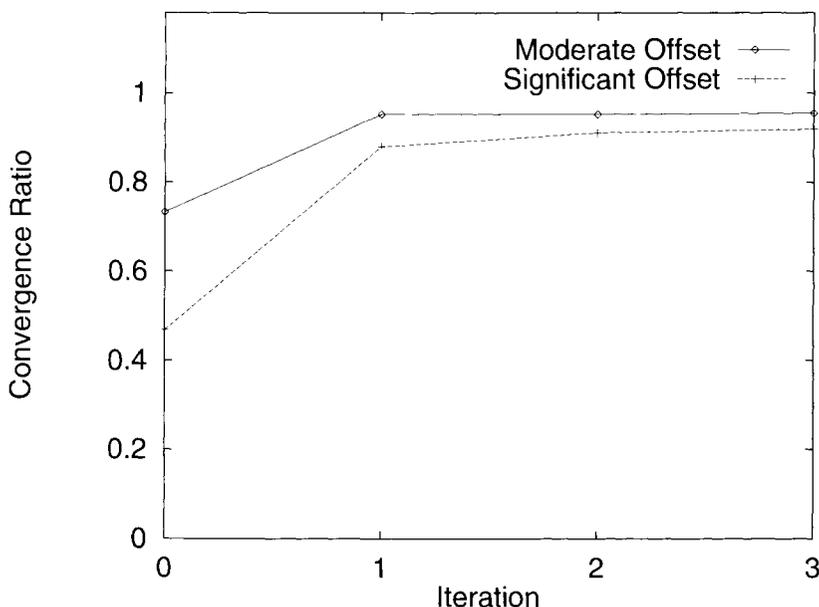
Figure 8. Convergence Ratio Curves.

The user can browse through the image database. Once the user finds
an image of interest, that image is submitted as a query. In Figure 9, the
query image is displayed at the upper-left corner as well as the best eleven
retrieved images. The top eleven best matches are displayed in 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 the 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 his or her information need to low-level
features, but rather the user can express the 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

Figure 9. The Retrieval Results Before the Relevance Feedback.

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.

CONCLUSION

This article discussed techniques to extend information retrieval beyond the textual domain. Specifically, it 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 seeks 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

Figure 10. The Retrieval Results After the Relevance Feedback.

paradigms both have 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 a 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, E. M.; Chew, L.; Huttenlocher, D.; Kedem, K.; & Mitchell, J. (1991). An efficiently computable metric for comparing polygonal shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 13*(3), 209-216.

Bach, J. R.; Fuller, C.; Gupta, A.; Hampapur, A.; Horowitz, B.; Humphrey, R.; Jain, R.; & Shu, C-F. (1996). The Virage image search engine: An open framework for image management. In *Storage and retrieval for image and video databases IV* (Proceedings held February 1-2, 1996, San Jose, CA) (pp. 76-87). Bellingham, WA: SPIE

Buckley, C., & Salton, G. (1995). Optimization of relevance feedback weights. In *SIGIR '95* (Proceedings of the 18ᵗʰ Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, July 9-13, 1995, Seattle, WA) (pp. 351-357). New York: Association for Computing Machinery Press.

Chaudhari, S., & Gravano, L. (1996). Optimizing queries over multimedia repositories. In *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data* (June 4-6, 1996, Montreal, Quebec, Canada) (pp. 91-102). New York: Association for Computing Machinery Press.

Chuang, G. C-H., & Kuo, C-C. J. (1996). Wavelet descriptor of planar curves: Theory and applications. *IEEE Transactions of Image Processing, 5*(1), 56-70.

Equitz, W., & Niblack, W. (1994). *Retrieving images from a database using texture-algorithms from the QBIC system* (IBM Computer Science Tech. Rep. No. RJ 9805). San Jose, CA: IBM.

Fagin, R. (1996). Combining fuzzy information from multiple systems. In *Proceedings of the Fifteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems* (PODS 1996, conference held June 3-5, 1996, Montreal, Canada) (pp. 216-226). New York: Association for Computing Machinery Press.

Fagin, R., & Wimmers, E. L. (1997). Incorporating user preferences in multimedia queries. In F. N. Afrati (Ed.), *Database Theory–ICDT '97* (Proceedings of the 6ᵗʰ International Conference, January 8-10, 1997, Delphi, Greece) (pp. 247-261). Berlin, Germany: Springer.

Flickner, M.; Sawhney, H.; Niblack, W.; Ashley, J.; Huang, Q.; Dom, B.; Gorkani, M.; Hafine, J.; Lee, D.; Petkovic, D.; Steele, D.; & Yanker, P. (1995). Query by image and video content: The QBIC system. *Computer, 28*(9), 23-32.

Haralick, R. M.; Shanmugam, K.; & Dinstein, I. (1973). Texture features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics, 3*(6), 610-621.

Kundu, A., & Chen, J-L. (1992). Texture classification using QMF bank-based subband decomposition. *Graphical Models and Image Processing, 54*(5), 369–384.

McCamy, C. S.; Marcus, H.; & Davidson, J. G. (1976). A color-rendition chart. *Journal of Applied Photographic Engineering, 2*(3), 95-99.

Miyahara, M. (1988). Mathematical transform of (R,G,B) color data to munsell (H,S,V) color data. In R. Hsing (Ed.), *Proceedings of SPIE: The Visual Society for Optical Engineering, Vol. 1001* (Visual Communications and Image Processing '88, November 9-11, 1988, Cambridge, MA) (pp. 650-657). Bellingham, WA: SPIE.

Ortega, M..; Rui, Y.; Chakrabarti, K.; Porkaew, K.; Mehrotra, S.; & Huang, T. S. (1998). Supporting ranked Boolean similarity queries in MARS. *IEEE Transactions on Knowledge and Data Engineering, 10*(6), 905-925.

Pentland, A.; Picard, R. W.; & Sclaroff, S. (1996). Photobook: Content-based manipulation of image databases. *International Journal of Computer Vision, 18*(3), 233-254.

Rui, Y.; She, A. C.; & Huang, T. S. (1996). Modified Fourier descriptors for shape representation—a practical approach. In A. Smeulders & R. Jain (Eds.), *Image databases and multi media search* (pp. 165-180). River Edge, NJ: World Scientific.

Rui, Y.; Huang, T. S.; Ortega, M.; & Mehrotra, S. (1998). Relevance feedback: A power tool in interactive content-based image retrieval. *IEEE Transactions on Circuits and Systems for Video Technology, 8*(5), 644-655.

Rui, Y.; Huang, T. S.; & Chang, S-F. (1999). Image retrieval: Current techniques, promising directions, and open issues. *Journal of Visual Communication and Image Representation, 10*(1), 39-62.

Salton, G., & McGill, M. J. (1983). *Introduction to modern information retrieval.* New York: McGraw-Hill Book Company.

Shaw, W. M. (1995). Term-relevance computations and perfect retrieval performance. *Information Processing and Management, 31*(4), 491-498.

Smith, J. R., & Chang, S-F. (1996). Tools and techniques for color image retrieval. In *Storage & retrieval for image and video databases IV* (Proceedings of the International Society for Optical Engineering, vol. 2670) (pp. 426-437). Bellingham, WA: SPIE.

Stricker, M., & Orengo, M. (1995). Similarity of color images. In W. Niblack & R. C. Jain (Eds.), *Storage and retrieval for image and video databases III* (Proceedings of the International Society for Optical Engineering, vol. 2420) (pp. 381-392). Bellingham, WA: SPIE.

Swain, M., & Ballard, D. (1991). Color indexing. *International Journal of Computer Vision, 7*(1), 11-32.

Tamura, H.; Mori, S.; & Yamawaki, T. (1978). Texture features corresponding to visual perception. *IEEE Transactions on Systems, Man, and Cybernetics, 8*(6), 460-473.

ADDITIONAL REFERENCES

Hu, M. K. (1962). Visual pattern recognition by moment invariants, computer methods in image analysis. In *IRE Transactions on Information Theory* (316 p.). New York: Institute of Radio Engineers.

Ortega, M.; Chakrabarti, K.; Porkaew, K.; & Mehrotra, S. (1998). *Cross media validation in a multimedia retrieval system.* Unpublished paper presented at the ACM Digital Libraries '98 Workshop on Metrics in Digital Libraries.

Ortega, M.; Rui, Y.; Chakrabarti, K.; Mehrotra, S.; & Huang, T. S. (1997). Supporting similarity queries in MARS. In *Proceedings of ACM Multimedia '97* (November 9-13, 1997, Seattle, WA) (pp. 403-413). New York: Association for Computing Machinery.

Rui, Y.; Huang, T. S.; & Mehrotra, S. (1997). Content-based image retrieval with relevance feedback in MARS. In *Proceedings of the International Conference on Image Processing* (October 26-29, 1997, Santa Barbara, CA) (pp. 815-818). Los Alamitos, CA: IEEE Computer Society.

Rui, Y.; Huang, T. S.; & Mehrotra, S. (1998). Exploring video structure beyond the shots. In *Proceedings of the International Conference on Multimedia Computing and Systems* (June 28-July 1, 1998, Austin, TX) (pp. 237-240). Los Alamitos, CA: IEEE Computer Society.