Supporting Similarity Queries in MARS

Michael Ortega, Yong Rui, Kaushik Chakrabarti, Sharad Mehrotra, and Thomas S. Huang
Department of Computer Science and Beckman Institute
University of Illinois at Urbana-Champaign
Urbana, IL 61801, USA
E-mail:{ortega-b,kaushikc,sharad}@cs.uiuc.edu, {yrui, huang}@ifp.uiuc.edu

Abstract

To address the emerging needs of applications that require access to and retrieval of multimedia objects, we are developing the *Multimedia Analysis and Retrieval System* (MARS) in our group at the University of Illinois [13]. In this paper, we concentrate on the retrieval subsystem of MARS and its support for content-based queries over image databases. Content-based retrieval techniques have been extensively studied for textual documents in the area of automatic information retrieval [24, 2]. This paper describes how these techniques can be adapted for ranked retrieval over image databases. Specifically, we discuss the ranking and retrieval algorithms developed in MARS based on the Boolean retrieval model and describe the results of our experiments that demonstrate the effectiveness of the developed model for image retrieval.

1 Introduction

While advances in technology allow us to generate, transmit, and store large amounts of digital images, and video, research in content based retrieval over multimedia databases is still at its infancy. Due to the difficulty in capturing the content of multimedia objects using textual annotations and the non-scalability of the approach to large data sets (due to a high degree of manual effort required in defining the annotations), the approach based on supporting content-based retrieval over visual features has become a promising research direction. This is evidenced by several prototypes[25, 18, 12, 13] and commercial systems[7, 1] that have been built recently. Such an approach can be summarized as follows:

1. Computer vision techniques are used to extract visual features from multimedia objects. For example, color, texture, shape features for images, and motion parameters for video.
2. For a given feature, a representation of the feature and a notion of similarity measure are determined. For example, color histogram is used to represent color feature, and intersection distance is used for similarity measure.
3. Objects are represented as a collection of features and retrieval of objects is performed based on computing similarity in the feature space. The results are ranked on the similarity values computed.

Since automatically extracted visual features (e.g., color, texture etc.) are too low level to be useful to the users in specifying their information needs directly, content-based retrieval using visual features requires development of effective techniques to map higher-level user queries (e.g., retrieve images containing field of yellow flowers) to visual features. Mapping a user's information need to a set of features extracted from textual documents have been extensively studied in the information retrieval literature [24]. This paper describes how we have generalized these approaches for content-based retrieval over image features in the *Multimedia Analysis and Retrieval System* (MARS) being developed in our group at the University of Illinois.

1.1 Information Retrieval Models

Before we describe the retrieval approach used in MARS, we briefly review the retrieval process in modern information retrieval (IR) systems [24]. In an IR system, a document is represented as a collection of features (also referred to as terms). Examples of features include words in a document, citations, bibliographic references, etc. A user specifies his information needs to the system in the form of a query. Given a representation of the user's information need and a document collection, the IR system estimates the likelihood that a given document matches the users information needs. The representation of documents and queries, and the mechanism used to compare their similarity forms the *retrieval model* of the system. Existing retrieval models can be broadly classified into the following categories:

Boolean Models Let $\{r_1, r_2, \ldots, r_k\}$ be the set of terms in a collection. Each document is represented as a binary-valued vector of length k where the i^{th} element of the vector is assigned *true* if r_i is assigned to the document. All elements corresponding to features/terms not assigned to a document are set to *false*. A query is a Boolean expression in which operands are terms. A document whose set of terms satisfies the Boolean expression is deemed to be relevant to the user and all other documents are considered not relevant.

Vector-based Models Let $\{r_1, r_2, \ldots, r_k\}$ be the set of terms in a collection. Both documents and queries are represented as a vector of k dimensions where each element in the vector corresponds to a real-valued weight assigned to a term. Several techniques have been proposed to compute these weights, the most common being *tf.idf* weights [24], where *tf* refers to the term frequency in the document, and *idf* is a measure proportional to the inverse of its frequency in the collection. Similarly, many similarity measures have been proposed between the document and the query [24] the most common being the cosine of the angle between the document and the query vectors.

Probabilistic Retrieval Models In these models the system estimates the probability of relevance of a document to the user's information need specified as a query. Documents are ranked in decreasing order of the relevance estimate. Given a document and a query, the system computes $P(R|d,q)$ which represents the probability that the document d will be deemed relevant to query q. Using Bayes' theorem and a set of independence assumptions about the distribution of terms in documents these probabilities are computed and the documents are ranked based on these probabilities.

Traditionally, commercial IR systems have used the Boolean model. Systems based on Boolean retrieval partition the set of documents into either being relevant or not relevant and do not provide any estimate as to the relevance of any document in a partition to the users information need. To overcome this problem, many variations of the term-weighting models and probabilistic retrieval models that provide ranked retrieval have been developed. The boolean model has also been extended to allow for ranking in the text domain (p-norm model [23]). Vector-based models and probabilistic retrieval models are in a sense related and provide similar performance. The primary difference being that while the vector models are ad hoc and based on intuitive reasoning, probability based models have a more rigorous theoretical base.

1.2 Overview of the Retrieval Approach used in MARS

With a large number of retrieval models in the information retrieval literature, MARS attempts to exploit this research for content-based retrieval over images. In MARS, an image is represented as a collection of low-level image features (e.g., color features, texture features, shape and layout features extracted automatically) as well as a manual text description of the image. 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 interpreted as a Boolean expression over image features and a Boolean retrieval model (adapted for retrieval over images) is used to retrieve a set of images ranked based on the degree of match. Boolean queries provide a natural interface for the user to formulate and refine *conceptual queries* to the system using lower-level image features. For example, high level concepts like fields of yellow flowers or a sunset by a lake can be expressed as a boolean combination of lower level features. Such a mapping of high to low level concepts can be provided explicitly by the user or alternatively learned via user interaction by a relevance feedback mechanism. Being able to support such conceptual queries is critical for the versatility of large image databases.

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 $dist(H(I), H(Q))$ be the distance 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:

$$\begin{aligned} \text{I matches Q} &= \textit{true}, \text{if } dist(H(I), H(Q)) \leq \delta_i \\ &= \textit{false}, \text{if } dist(H(I), H(Q)) > \delta_i \end{aligned}$$

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 some 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.

While the above straightforward adaptation of Boolean retrieval can be used, it suffers from many 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 the distance of $\delta_i - \epsilon$ and $\delta_i + \epsilon$ from a query q, where $\epsilon \rightarrow 0$, are essentially very similar will be considered as very different by the system. While I_1 will be considered relevant to the query, I_2 will be considered as not relevant. This problem may be alleviated if instead of fixed *a priori* values for tolerance for a given feature, δ_i was computed dynamically for each query based on the image collection. However, the approach would still suffer from the fundamental restriction of the basic Boolean retrieval in that it produces a unranked set of answers.

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

Fuzzy Boolean Retrieval distance between the image and the query feature is considered to be 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 the their degree of membership in the set.

Probabilistic Boolean Retrieval distance between the image and the query feature is considered to be a measure of 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 a ranked retrieval over the image sets. Furthermore, as one expects from Boolean models, both the fuzzy and the probabilistic Boolean models rank images based on queries corresponding to semantically equivalent Boolean expressions in the same order.

The rest of the paper is developed as follows. In Section 2, we describe the set of basic image features used in

MARS including technique used to measure similarity between images based on a single feature. Section 3 is devoted to defining the Boolean retrieval models used in MARS and discussing the issues related to their efficient implementation. Section 4 discusses normalization of the low level features necessary to combine with each other. Experimental results that show the retrieval effectiveness of the developed models are discussed in Section 5 and finally Section 6 offers concluding remarks including a discussion of the work we are pursuing in the future.

2 Image Features Used in MARS

The retrieval performance of an image database is inherently limited by the nature and the quality of the features used to represent the image content. In this section, we briefly describe the image features used in MARS and the corresponding distance functions used for comparing similarity of images based on the features. The discussion is kept short since the purpose of this section is only to provide a background for discussing issues related to normalization and ranked retrieval based on Boolean queries. Detailed discussion on the rationale and the quality of the chosen features can be found in references [6, 26, 13, 15, 22].

Color Features: To represent color, we choose the HSV space due to its de-correlated and uniform coordinates. The color feature is represented using a color histogram. Since V coordinate is easily affected by the lighting condition, we use only HS coordinates to form an 8×8 two-dimensional histogram. To measure distance between two color histograms, we compute the amount of non-overlap between the two histograms which is defined as follows:

$$dist_{color} = 1 - \sum_{i=1}^{i=N} min(H_1(i), H_2(i)) \quad (1)$$

where H_1 and H_2 are the two histograms and N is the number of bins used in the histogram. The above intersection based measure of distance provides an accurate and efficient measure of (dis)similarity between two images based on their color [25].

Texture Features: To represent texture of an image, a CCD (*coarseness, contrast*, and *directionality*) texture feature representation has developed in [26, 6]. *Coarseness* is a measure of granularity of the texture, i.e. fine vs coarse. *Contrast* represents the distribution of luminance of the image and is defined as

$$contrast = \sigma/(\alpha_4)^{1/4} \quad (2)$$

where $\alpha_4 = u_4/\sigma^4$. Here σ and u_4 are the standard deviation and the fourth central moment of the luminance, respectively. *Directionality* is a measure of how "directional" the image is. Using the above definitions of CCD, texture is represented as a set of three numbers. A problem with the above described CCD features is that it is sensitive to noise. We have developed and implemented an enhanced version of CCD by using histogram-base features [13]. Experimental results show that our enhanced version is much more robust and accurate than the original definition of CCD [13].

Shape Features: Shape of an object in an image is represented by its boundary. A technique for storing the boundary of an object using modified Fourier descriptor (MFD) is described in [22]. To measure similarity between two shapes, the Euclidean distance between two shape features can be used. In [22], however, we proposed a standard deviation based similarity measure that performs significantly better compared to the simple Euclidean distance. The proposed representation and similarity measure provide invariance to translation, rotation, and scaling of shapes, as well as the starting point used in defining the boundary sequence.

Layout Features: The features discussed so far only describe the global properties of the image. Besides these global properties, MARS also supports features that describe the layout of color and texture in an image. To extract the layout features, the whole image is first split into 5×5 sub-images. Then color and texture features are extracted from each sub-image and stored in the database. For color layout, a two-dimensional HS histogram is constructed for each sub-image, similar to the procedure described earlier.

Since the enhanced CCD representation uses a histogram based measure, it is not suitable for texture layout. This is because the small sub-images may not produce good histograms. Instead, a wavelet-based representation is used, in which the mean and the standard deviation at 10 sub bands are used to represent the texture of each sub-image. The Euclidean distance is used to compute the texture similarity distance for the corresponding sub-images. A weighted sum is then used to form the texture layout distance.

3 Retrieval Models Used In MARS

This section will discuss how to support the Boolean query based on the simple feature distances. MARS supports two mechanisms for generating the ranking of Boolean queries – the first is based on the fuzzy interpretation of the distance and the second is based on a probabilistic interpretation. In the discussion below, we will use the following notation. Images in the collection are represented as $I_1, I_2, \ldots, I_m$. Features over the images are represented as $F_1, F_2, \ldots, F_r$, where F_i is used to represent both the name of the feature as well as the domain of values that the feature can take. For example, say F_1 is the color feature which is represented in the database using an HS histogram. In that case, F_1 is also used to represent the set of all the color histograms. A query is a Boolean expression $Q(v_1, v_2, \ldots, v_n)$, where $v_1, v_2, \ldots, v_n$ are variables. Each variable v_i takes its value from some domain F_j. For example, a query, $Q(v_1, v_2) = v_1 \wedge v_2$ may be a query where v_1 has a value equal to the color histogram associated with image I_2 and v_2 has a value of the texture feature associated with I_5. The query Q then represents a desire to retrieve images whose color matches that of image I_2 and whose texture matches that of image I_5.

3.1 Fuzzy Boolean Model

Let $Q(v1, v2, \ldots, vn)$ 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 I such that the distance between the variable v_i and the corresponding feature in I is used to compute the degree of membership of I in the fuzzy set. That is:

$$f_{v_i}(I) = 1 - dist(I, v_i) \quad (3)$$

where $dist(I, v_i)$ represents the distance measure between v_i and the corresponding feature in the image I. With

the above interpretation of the distance 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:

And: $f_{Q=Q_1 \wedge Q_2}(I) = \min(f_{Q_1}(I), f_{Q_2}(I))$

Or: $f_{Q=Q_1 \vee Q_2}(I) = \max(f_{Q_1}(I), f_{Q_2}(I))$

Not: $f_{Q=not Q_1}(I) = 1 - f_{Q_1}(I)$

Consider for example a query$Q$:

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

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

$$f_Q(I) = min(max(f_{v_1}(I), f_{v_2}(I), f_{v_3}(I)), max(f_{v_4}(I), min(f_{v_5}(I), f_{v_1}(I)))) \quad (4)$$

The value $f_{v_i}(I)$ in the above formula is determined using the equation (3). Once the membership value of the image in the fuzzy set associated with the query have been determined, these values are used to rank the images, where a higher value of $f_Q(I)$ represents a better match of the image I to the query Q. We refer to the fuzzy Boolean model as model **F1**.

3.2 Probabilistic Boolean Model

Let $Q(v1, v2, \ldots, vn)$ be a query and I be an image. In the probabilistic Boolean model, the distance $dist(I, v_i)$ between the query variable v_i and the corresponding feature in the image is used to compute the probability of the image I matching the query variable v_i, denoted by $P(v_i|I)$. These probability measures are then used to compute the probability that I satisfies the query $Q(v1, v2, \ldots, v_n)$ (denoted by $P(Q(v1, v2, \ldots, v_n)|I)$) which is in turn used to rank the images. To be able to compute $P(Q(v1, v_2, \ldots, v_n)|I)$, an assumption of feature independence is made. That is, we assume that for all variables v_i, v_j such that the domain of v_i is not the same as the domain of v_j, the following holds:

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

Before the probabilistic Boolean model can be used, however, we need to map the distance measure between a query variable and the image into a measure of probability that the image matches the query variable. There are many ways in which such a mapping can be achieved. Three such mechanisms (depicted in Figure 1) implemented in MARS are described below.

$$P(v_i|I) = \left(\frac{1}{1 + dist(I, v_i)} - \frac{1}{2}\right) \times 2 \quad (5)$$

$$P(v_i|I) = 1 - dist(I, v_i) \quad (6)$$

$$P(v_i|I) = 1 - (dist(I, v_i))^2 \quad (7)$$

It is easy to verify that for each of the above interpretations of probability, the range of $P(v_i|I)$ is [0, 1]. In addition, when $dist(I, v_i)$ equals 0 (best match), $P(v_i|I) = 1$;

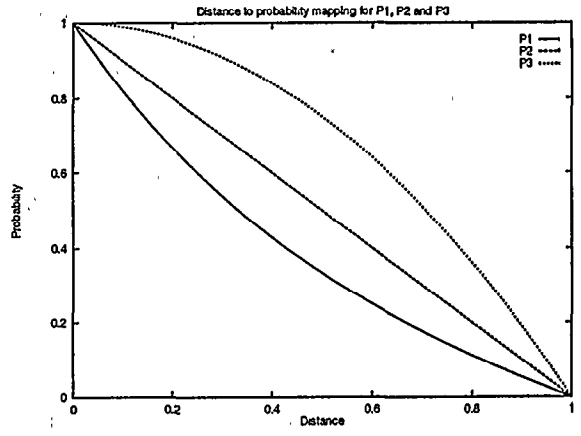

Figure 1: Distance to probability transformation functions.

when $dist(I, v_i)$ equals 1 (worst match), $P(v_i|I) = 0$. Also, $P(v_i|I)$ is a monotonic decreasing function of $dist(I, v_i)$, which fits the physical meanings of $dist(I, v_i)$ and $P(v_i|I)$.

The choice of the mapping has an impact on the image ranking and as a result on the retrieval performance. We will discuss this further in the section on experimental results. We will refer to the three different probabilistic Boolean models resulting from the above equations as models **P1**, **P2**, and **P3** respectively.

Once distance between image and the query variable has been converted to a probability measure, we next need to estimate the probability that the image satisfies the Boolean query $Q(v_1, v_2, \ldots, v_n)$, 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:

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

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:

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

To compute conjunction queries, i.e. $Q = Q_1 \wedge Q_2$ it is desirable that Q_1 and Q_2 are independent so that $P(Q) = P(Q_1)P(Q_2)$. However even though features are independent, the sub queries Q_1 and Q_2 may not be independent. To see this consider for example, a query $Q = Q_1 \wedge Q_2$, where $Q_1 = (v_1 \vee v_2)$ and $Q_2 = (v_1 \vee v_3)$. In such a case since Q_1 and Q_2 are not independent, $P(Q_1 \wedge Q_2)$ cannot be replaced by the product of $P(Q_1)$ and $P(Q_2)$.

The above motivates us to convert the query into a disjunctive normal form (DNF) in which a query is represented as a disjunction of conjuncts[1]. Once the query has been converted to a DNF expression, we can compute the probability of an image satisfying the query based on the probability that the image satisfies query feature variables. We illustrate how computation is done using an example. Consider a query $Q = (v_1 \wedge v_2) \vee (v_1 \wedge v_3) \vee (v_1 \wedge \neg v_3 \wedge v_4)$.

[1]The number of disjuncts, in general, may be exponential in the size of the query. However, if queries are reasonably small, we do not consider this to be an issue. In MARS, we overcome the exponentiality problem by forcing users to input Boolean queries in the DNF form when the probabilistic model is used to rank images.

$$
\begin{aligned}
P(Q) &= P((v1 \wedge v2) \vee (v1 \wedge v3) \vee (v1 \wedge \neg v3 \wedge v4)) \\
&= P(v1 \wedge v2) + P(v1 \wedge v3) + P(v1 \wedge \neg v3 \wedge v4) \\
&\quad -P(v1 \wedge v2 \wedge v1 \wedge v3) - P(v1 \wedge v2 \wedge v1 \wedge \neg v3 \wedge v4) - P(v1 \wedge v3 \wedge v1 \wedge \neg v3 \wedge v4) \\
&\quad +P(v1 \wedge v2 \wedge v1 \wedge v3 \wedge v1 \wedge \neg v3 \wedge v4) \\
&= P(v1)P(v2) + P(v1)P(v3) + P(v1)(1 - P(v3))P(v4) \\
&\quad -P(v1)P(v2)P(v3) - P(v1)P(v2)(1 - P(v3))P(v4) - P(false) + P(false) \\
&= P(v1)P(v2) + P(v1)P(v3) + P(v1)P(v4) - P(v1)P(v3)P(v4) \\
&\quad -P(v1)P(v2)P(v3) - P(v1)P(v2)P(v4) + P(v1)P(v2)P(v3)P(v4)
\end{aligned}
$$

Figure 2: Example Derivation

In the derivation shown in figure 2, we have made the assumption that each pair of variables v_i, v_j are over independent features. For example, v_i may be a color histogram and v_j may be the shape feature. Notice that, in general, the two variables may be over the same feature space. For example, in a query $Q = v_1 \wedge v_2$, v_1 and v_2 may correspond to two color histograms. Our retrieval results (see Section 5) show that even if these variables are considered as independent, the resulting retrieval performance is quite good. Developing a feature dependence model and incorporating it in the system may improve retrieval performance further and is an important extension to our current work.

3.3 Finding the Best N Matches

While the above developed Boolean retrieval models provide a mechanism for ranking the given images based on the query, for the approach to be useful, techniques must be developed to retrieve the best N matches efficiently without having to rank each image. Such a technique consists of two steps:

- retrieve images in a rank order based on each feature variable v_i in the query.
- combine the results of the single feature variable queries to generate ranked retrieval for the entire query.

The images can be efficiently retrieved ranked based on a single feature by maintaining an index based on that feature. Since all features in MARS are represented as feature vectors with multiple feature elements, retrieval of images ranked based on a single feature requires search using the values of all the elements in the feature vector. For example, the color feature in MARS is a 64-element vector (8×8 histogram). An alternative is to use several single-attribute indexes, one for each feature element of the feature vector. This is extremely inefficient in terms of the number of I/O accesses needed for the search. The other alternative is to use one or more of the existing multidimensional data structures [16, 10, 4]. However, these data structures do not scale to the high dimensionality of the feature vectors used in MARS. Moreover, since these data structures can only be used to index Euclidean feature vectors, they cannot be used for features defined over non-Euclidean distance measures (e.g. color histograms for which the intersection distance metric is used). Instead MARS uses an incremental clustering approach described in [19].

Once efficient ranked retrieval based on a single feature has been achieved, the ranked lists are *normalized* and then the normalized ranked lists are merged into a ranked set of images corresponding to a query. The normalization process used in MARS is described in the following section. To merge the normalized ranked lists, a query $Q(v_1, v_2, \ldots, v_n)$ 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, nodes are of either of the three forms: $\wedge(v1, v2, \ldots, v_n)$ which is a conjunction of positive literal; $\wedge(v1, v2, \ldots, v_m, \neg v_{m+1} \ldots, \neg v_n)$, which is a conjunction consisting of both positive and negative literals; and $\vee(v1, v2, \ldots, v_n)$ which is a disjunction of positive literals. The query tree is evaluated as a pipeline from the leaf to the root. Each node in the tree provides to its parent a ranked list of images, where the ranking corresponds to the degree of membership (in the fuzzy model), or the measure of probability (in the probabilistic model). For example, in the fuzzy model, a node n in a tree provides to its parent a ranked list $\langle I, f_{query(n)}(I) \rangle$, where $query(n)$ corresponds to the query associated with the node n.

The algorithms used to combine the ranked lists of images from the child to generate a list for parents depend upon the retrieval model used. The algorithms for the fuzzy model are discussed in the Appendix. The corresponding algorithms for the probabilistic model are more complex and not included due to space restrictions. They can be found in [17].

4 Feature Sequence Normalization

The normalization process serves two purposes:

1. It puts an equal emphasis on each feature element within a feature vector. To see the importance of this, notice that in texture layout representation, the feature elements may be totally different physical quantities. For example, one feature can be a mean while the other can be a standard deviation. Their magnitudes can vary drastically, thereby biasing the Euclidean distance measure. This is overcome by the process of *intra-feature* normalization.
2. It maps the distance values of the query from each atomic feature into the range [0,1] so that they can be interpreted as the degree of membership in the fuzzy model or relevance probability in the probability model. While some similarity functions return a value in the range of [0, 1], e.g. the color histogram intersection, others do not, e.g. the Euclidean distance used in texture layout. In the latter case the distances need to be

converted to the range of [0, 1] before they can be used. This is referred to as *inter-feature normalization.*

4.1 Intra-feature Normalization

This normalization process is only needed for *vector based* feature representation, as in the case of wavelet texture feature representation. In other cases, such as color histogram intersection, where all the feature elements are defined over the same physical domain, no intra-feature normalization is needed.

For the vector based feature representation, let $F = [f_1, f_2, ..., f_j, ..., f_N]$ be the feature vector, where N is the number of feature elements in the feature vector and $I_1, I_2, \ldots, I_M$ be the images. For image I_i, we refer the corresponding feature F as $F_i = [f_{i,1}, f_{i,2}, ..., f_{i,j}, ..., f_{i,N}]$. Since there are M images in the database, we can form a $M \times N$ feature matrix $F = f_{i,j}$, where $f_{i,j}$ is the jth feature element in feature vector F_i. Now, each column of F is a length-M sequence of the jth feature element, represented as F_j. Our goal is to normalize the entries in each column to the same range so as to ensure that each individual feature element receives equal weight in determining the Euclidean distance between the two vectors. One way of normalizing the sequence F_j is to find the maximum and minimum values of F_j and normalize the sequence to [0, 1] as follows:

$$f_{i,j} = \frac{f_{i,j} - min_j}{max_j - min_j}, \tag{8}$$

where min_j and max_j refer to the smallest and the biggest value of $f_{i,j}$, $i = 1, 2, \ldots, M$. Although simple, this is not a desirable normalization. Considering the sequence {1.0, 1.1, 1.2, 1.3, 100}, if we use (8) to normalize the sequence, most of the [0, 1] range will be taken away by a single element 100, and most of the useful information in {1.0, 1.1, 1.2, 1.3} will be warped into a very narrow range.

A better approach is to use the Gaussian normalization. Assuming the feature sequence F_j to be a Gaussian sequence, we compute the mean m_j and standard deviation σ_j of the sequence. We then normalize the original sequence to a N(0,1) sequence as follows:

$$f_{i,j} = \frac{f_{i,j} - m_j}{\sigma_j} \tag{9}$$

It is easy to prove that after the normalization according to (9), the probability of a feature element value being in the range of [-1, 1] is 68%. If we use $3\sigma_j$ in the denominator, according to the 3-σ rule, the probability of a feature element value being in the range of [-1, 1] is approximately 99%. In practice, we can consider all of the feature element values are within the range of [-1,1] by mapping the out-of-range values to either -1 or 1. The advantage of this normalization process over (8) is that the presence of a few abnormally large or small values does not bias the importance of the feature element in computing the distance between feature vectors.

4.2 Inter-feature Normalization

Intra-feature normalization ensures equal emphasis of each feature element within a feature. On the other hand, inter-feature normalization ensures equal emphasis of each feature within a composite query.

The feature representations used in MARS are of various forms, such as vector based (wavelet texture representation), histogram based (histogram color representation), irregular (MFD shape representation). To map the distance computations of the heterogeneous features to the same scale and into the range [0,1] the following inter-feature normalization process is used for each feature F_i.

1. For any pair of images I_i and I_j, compute the similarity distance $D_{(i,j)}$ between them:

$$D_{(i,j)} = dist(F_{I_i}, F_{I_j}) \tag{10}$$
$$i, j = 1, ..., M, \quad i \neq j$$

   where F_{I_i} and F_{I_j} are the feature representations of images I_i and I_j.

2. For the $C_2^M = \frac{M \times (M-1)}{2}$ possible distance values between any pair of images, treat them as a value sequence and find the mean m and standard deviation σ of the sequence. Store m and σ in the database to be used in later normalization.

3. After a query Q is presented, compute the raw (unnormalized) similarity value between Q and the images in the database. Let $s_1, ..., s_M$ denote the raw similarity values.

4. Normalize the raw similarity values as follows:

$$s_i' = \frac{s_i - m}{3\sigma} \tag{11}$$

   As explained in the intra-feature normalization section, this Gaussian normalization will ensure 99% of s_i' to be within the range of [-1,1]. An additional shift will guarantee that 99% of similarity values are within [0,1]:

$$s_i'' = \frac{s_i' + 1}{2} \tag{12}$$

   After this shift, in practice, we can consider all the values are within the range of [0,1], since an image whose distance from the query is greater than 1 is very dissimilar and can be considered to be at a distance of 1 without affecting retrieval.

4.3 Weights for feature and feature elements

After the intra- and inter-feature normalization processes discussed above, the feature elements within a feature as well as the features within a composite query are of equal weights. This *objective* equality allows us to further associate *subjective* unequal intra- and inter-feature weights for a particular query.

Inter-feature weights associated with the composite query reflect the user's different emphasis of the atomic feature in the composite query. For example, for a composite query based on color and texture, a user may put the weight for color equals 90% and weight for texture equals 10%. The support of different inter-feature weights enables the user to specify his/her information need more precisely.

Intra-feature weights associated with each feature vector reflect the different contributions of the feature elements to the feature vector. 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

(a) Sample 1 (b) Sample 2 (c) Sample 2

Figure 4: Three sample images used in conjunction with known relevant sets to evaluate precision and recall

deviation of a sub-band is independent of the lighting condition. Therefore the user may want to put more weight on the standard deviation feature element, and less weight on the mean feature element. The support of the different intra-feature weights enables the system to have more reliable feature representation and thus better retrieval performance.

In MARS, we have explored many techniques to associate subjective weights with feature elements and feature vectors. Associating subjective weights improves the retrieval performance considerably. Due to page limitations we do not discuss this any further. Instead we refer the readers to [21, 20].

5 Experimental Evaluation

For our experiments we used a collection of images of ancient African artifacts from the Fowler Museum of Cultural History. We used a total of 286 images of such artifacts. A sample of our collection is shown as the result of a query in figure 3.

To demonstrate the retrieval quality of our system, we chose 13 typical conceptual queries, three examples of which are "all stools", "stone masks" or "golden pots". Sample images satisfying these three concept queries are shown in figure 4.

To determine the relevant answer sets for each of the conceptual queries, we browsed through the collection and marked those images relevant to a concept. The concept queries were then mapped into a Boolean formulation that best represents them (e.g. stone masks are expressed as 'texture=image X and shape=image Y').

As in the text based retrieval systems, the retrieval performance is defined based on *precision* and *recall* [24, 2].

Precision is the ratio of the number of relevant images retrieved to the total number of images retrieved.

$$\frac{|relevant\ retrieved|}{|retrieved|}$$

Perfect precision (100%) means that all retrieved images are relevant.

Recall is the ratio of the number of relevant images retrieved to the total number of relevant images.

$$\frac{|relevant\ retrieved|}{|relevant|}$$

Perfect recall (100%) can be obtained by retrieving the entire collection, but the precision will be poor.

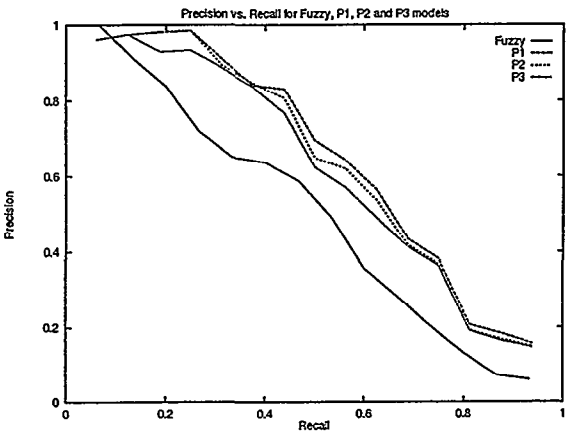

Figure 5: Precision for various levels of recall averaged over 13 queries

We have conducted experiments in which we calculate precision at various levels of recall. These results are reported in figure 5.

As we can see from the figure, the probability models consistently outperform the fuzzy model at equal recall or precision points. An interpretation of this is that the fuzzy model includes only partial information. For *and* (*or*) queries, the fuzzy model uses the min (max) operation that selects an image based on the worst (best) degree of membership in both operand sets. This ignores any contribution of the other operand set in computing the final degree of membership of an image. The probability model instead combines the information provided by both terms in a better way. This allows for an improved ranking as more information about an image is captured.

We can also see that P1 is the best among the three probability models. This relates to figure 1 where $P1 < P2 < P3$ at the same distance. Model P1 has a sharper drop in probability for distance less than 0.414. This has the effect of "warping" space close to the optimum match ($Pr = 1$) which results in slight ranking differences in the *and* and *or* operations. By this effect, good matches are drawn closer by a larger factor which has the effect of increasing precision. P1 gives more importance to images ranking good in any of the operand sets while models P2 and P3 give progressively less importance to good rankings, trying to equate them to worse rankings in the hope that the combination might be better. A faster decreasing function will negatively impact recall. In the limiting case ($[1 - dist(I, v_i)]^p, p \rightarrow \infty$) precision will be 1, but recall very low since even small distance reductions between a feature and the query feature will result in a large probability drop. As a result, similar images (unless they match perfectly) would rank lower and thus not be returned.

6 Related Work

Content-based retrieval of images is an active area of research being pursued independently by many research teams. Similar to MARS, most existing content-based image retrieval systems also extract low-level image features like color, texture, shape, and structure [6, 26, 13, 7, 15, 12, 18, 25]. However, compared to MARS the retrieval tech-

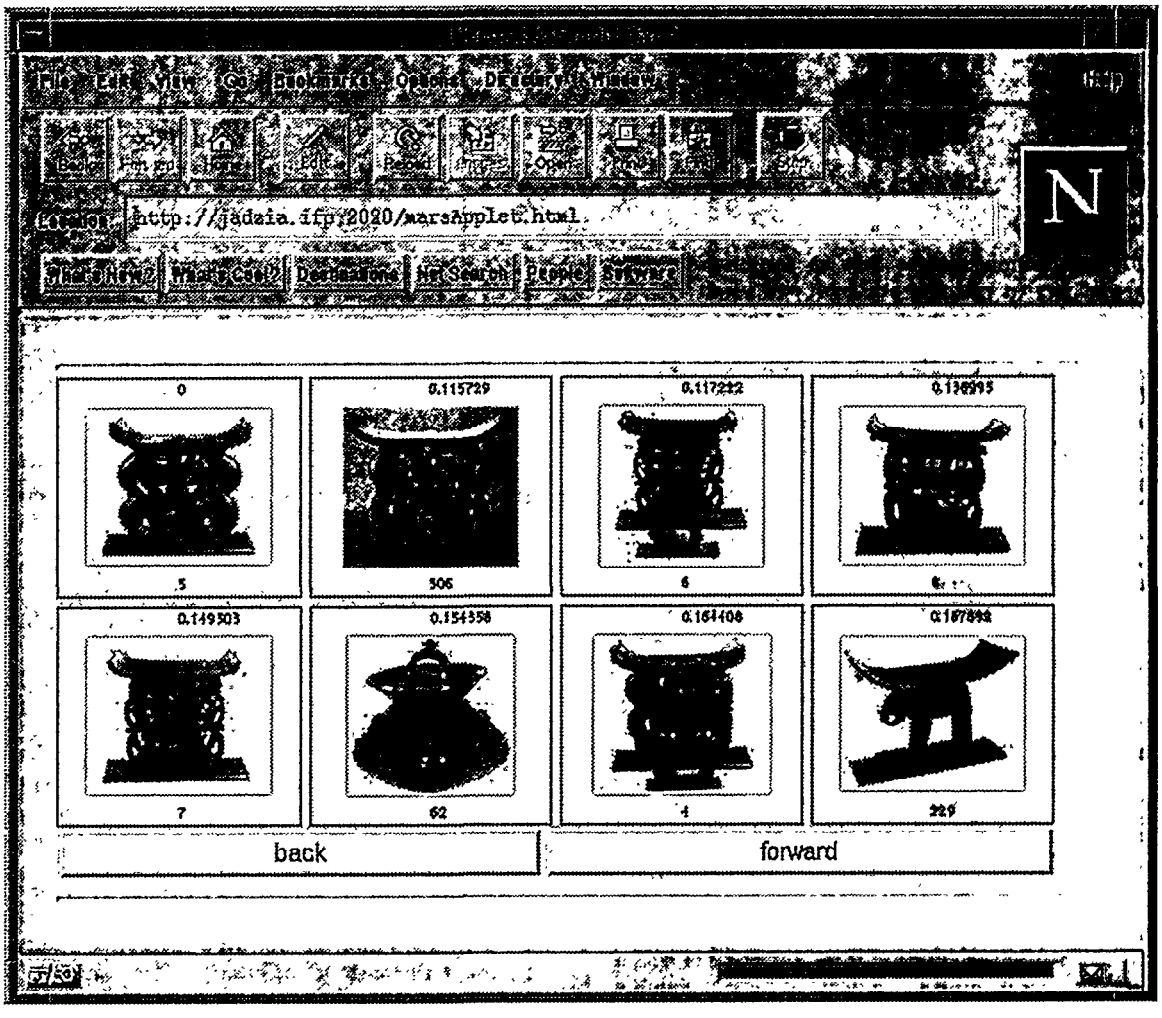

Figure 3: Screen shot of the query "texture or shape" with a stool image as parameter

niques supported in some of these systems are quite primitive. Many of these systems support queries only on single features separately. Certain other systems allow queries over multiple feature sets by associating a degree of tolerance with each feature. An image is deemed similar to the query if it is within the specified tolerance on all the query features. As discussed in section 1.2, this approach has many drawbacks.

Some commercial systems have been developed. QBIC [7], standing for Query By Image Content, is the first commercial content-based Image Retrieval system. Its system framework and techniques had profound effects on later Image Retrieval systems. QBIC supports queries based on example images, user-constructed sketches and drawings and selected color and texture patterns, etc. The color features used in QBIC are the average (R,G,B), (Y,i,q),(L,a,b) and MTM (Mathematical Transform to Munsell) coordinates, and a k element Color Histogram. Its texture feature is an improved version of the Tamura texture representation [26], i.e. combinations of coarseness, contrast and directionality. Its shape feature consists of shape area, circularity, eccentricity, major axis orientation and a set of algebraic moments invariants. QBIC is one of the few systems which take into account high dimensional feature indexing. In its indexing subsystem, the KL transform is first used to perform dimension reduction and then R^*-tree is used as the multi-dimensional indexing structure.

Virage is a content-based image search engine developed at Virage Inc. Similar to QBIC, Virage [1] supports visual queries based on color, composition (color layout), texture, and structure (object boundary information). But Virage goes one step further than QBIC. It also supports arbitrary combinations of the above four atomic queries. Users can adjust the weights associated with the atomic features according to their own emphasis. In [1], Jeffrey et al. further proposed an open framework for image management. They classified the visual features ("primitive") as general (such as color, shape, or texture) and domain specific (face recognition, cancer cell detection, etc.). Various useful "primitives" can be added to the open structure depending on the domain requirements. To go beyond the query-by-example mode, Gupta and Jain proposed a nine-component *query language* framework in [9].

Photobook [18] is a set of interactive tools for brows-

ing and searching images developed at the MIT Media Lab. Photobook consists of three sub-books, from which shape, texture, and face features are extracted respectively. Users can then query based on corresponding features in each of the three sub-books. In its more recent version of Photobook, FourEyes, Picard et al. proposed to include human in the image annotation and retrieval loop [14]. The motivation of this was based on the observation that there was no single feature which can best model images from each and every domain. Furthermore, human perception is subjective. They proposed a "society of models" approach to incorporate the human factor. Experimental results show that this approach is very effective in interactive image annotation.

In [11] the authors propose an image retrieval system based on color and shape. Their color measure is based on the RGB color space and euclidean and histogram intersection measures are used. For shape, they use a polygonal description that is resilient to scaling, translation and rotation. The proposed integration uses a weighted sum of shape and color to arrive at the final result. They address high dimensional feature indexing with a clustering approach, where clusters are build upon database creation time.

To date, no systematic approach to answering content based queries based on image features has emerged. To address this challenge, similar to the approaches taken in information retrieval system, the approach we have taken in developing MARS is to support an "intelligent retrieval" model using which a user can specify their information need to the image database and the database provides a ranked retrieval of images to user's request. The retrieval model supported is a variation of the Boolean model based on probabilistic and fuzzy interpretation of distances between the image and the query.

Recently, in parallel to our work, the problem of processing boolean queries over multimedia repositories has also been studied in [5] and [3]. These approaches have, however, restricted themselves to a boolean model based on a fuzzy interpretation of boolean operators. Our experimental results illustrate that the probabilistic model outperforms the fuzzy model in terms of retrieval performance (see figure 5). Furthermore, the query evaluation approach used in MARS differs significantly from the approaches developed in [5, 3]. As will become clear in the appendix, MARS follows a *demand-driven data flow* approach [8]; i.e., data items are never produced before they are needed. So the wait in a temporary file or buffer between operators in the query tree for each item is minimized. This model is efficient in its time-space-product memory costs [8]. In this model, the operators are implemented as *iterators* which can be effectively combined with parallel query processing. On the other hand, the strategy presented in [5] requires intermediate storage of data items at internal nodes of the query tree to evaluate the best N answers. This is also true for the approach followed in [3].

7 Conclusions

To address the emerging needs of applications that require access to and retrieval of multimedia objects, we are developing the *Multimedia Analysis and Retrieval System* (MARS) in our group at the University of Illinois [13]. In this paper, we described the retrieval subsystem of MARS and its support for content-based queries over image databases. To support content-based retrieval, in MARS many visual features are extracted from images– color, texture, shape, color and texture layout. Information retrieval (IR) techniques, modified to work over visual features, are then used to map user's queries to a collection of relevant images. Specifically, extended boolean models based on a probabilistic and fuzzy interpretation of boolean operators are used to support ranked retrieval. Our results show that using IR techniques for content-based retrieval in image databases is a promising approach.

The work reported in this paper is being extended in many important directions. In our current system, we have concentrated on adapting the boolean retrieval model for content-based retrieval of images. Many other retrieval models that have a better retrieval performance compared to the boolean approach have been developed in the IR literature for textual databases [24, 2, 27]. We are currently exploring how these models can be adapted for content-based image retrieval. Furthermore, our current work has concentrated on image databases. We are also generalizing our approach to content-based retrieval in multimedia databases. Finally, we are also exploring the use of relevance feedback techniques in our extended boolean model.

8 Acknowledgments

This work was supported in part by the Army Research Laboratory under Cooperative Agreement No. DAAL01-96-2-0003; in part by NSF/DARPA/NASA Digital Library Initiative Program under Cooperative Agreement No. 94-11318; in part by NSF CISE Research Infrastructure Grant CDA-9624396. Yong Rui is supported in part by CSE, College of Eng. UIUC. Michael Ortega is supported in part by CONACYT grant 89061. The 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. This goal of the two-year MESL project was to test the licensing and delivery of digital images and meta data from seven U. S. museums to seven U. S. universities. The University of Illinois was selected as a participant in the MESL project.

References

[1] Jeffrey R. Bach, Charles Fuller, Amarnath Gupta, Arun Hampapur, Bradley Horowitz, Rich Humphrey, Ramesh Jain, and Chiao fe Shu. The virage image search engine: An open framework for image management. In *SPIE Storage and Retrieval for Still Image and Video Databases IV.*

[2] J. P. Callan, W. B. Croft, and S. M. Harding. The inquery retrieval system. In *In Proceedings of the Third International Conference on Database and Expert Systems Applications*, Valencia, Spain, 1992.

[3] Surajit Chaudhari and Luis Gravano. Optimizing queries over multimedia repositories. *Proc. of SIGMOD*, 1996.

[4] Lomet D. The hb-tree: A multiattribute indexing mechanism with good guaranteed performance. *ACM Transactions on Database Systems*, 15(4), 1990.

[5] Ronald Fagin. Combining fuzzy information from multiple systems. *Proc. of the 15th ACM Symp. on PODS*, 1996.

[6] C. Faloutsos, M. Flocker, W. Niblack, D. Petkovic, W. Equitz, and R. Barber. Efficient and effective querying by image content. Technical Report RJ 9453 (83074), IBM Research Report, Aug. 1993.

[7] M. Flickner et al. Query by image and video content: The qbic system. *IEEE Computer*, September 1995.

[8] Goetz Graefe. Query evaluation techniques for large databases. *ACM Computing Surveys, Vol. 25, No. 2, 1993*, 1996.

[9] Amarnath Gupta and Ramesh Jain. Visual information retrieval. *Communications of the ACM*, 40(5), 1997.

[10] A. Guttman. R-tree: a dynamic index structure for spatial searching. In *SIGMOD*, pages 47–57, 1984.

[11] Anil K. Jain and Aditya Vailaya. Image retrieval using color and shape. *Pattern Recognition*, 29(8), 1996.

[12] B.S. Manjunath and W.Y. Ma. Texture features for browsing and retrieval of image data. Technical Report TR-95-06, CIPR, July 1995.

[13] Sharad Mehrotra, Kaushik Chakrabarti, Michael Ortega, Yong Rui, and Thomas S. Huang. Towards extending information retrieval techniques for multimedia retrieval. In *3rd International Workshop on Multimedia Information Systems, Como, Italy*, 1997.

[14] T. P. Minka and R. W. Picard. Interactive learning using a "society of models". Technical Report 349, MIT Media Lab, 1996.

[15] Makoto Miyahara et al. Mathematical transform of (r,g,b) color data to munsell (h,v,c) color data. In *SPIE Visual Communications and Image Processing'88*, volume 1001, 1988. 650.

[16] J. Neivergelt et al. The grid file: an adaptable, symmetric multikey file structure. *ACM Transactions on Database Systems*, March 1984.

[17] Michael Ortega, Sharad Mehrotra, Yong Rui, Kaushik Chakrabarti, and Thomas S. Huang. Processing similarity queries in mars. Technical Report TR-MARS-97-12, University of Illinois at Urbana-Champaign, Aug. 1997.

[18] A. Pentland, R. W. Picard, and S. Sclaroff. Photobook: Tools for content-based manipulation of image databases. In *Proc. Storage and Retrieval for Image and Video Databases II*, volume 2(185), pages 34–47, Bellingham, Wash, 1994. SPIE.

[19] Yong Rui, Kaushik Chakrabarti, Sharad Mehrotra, Yunxin Zhao, and Thomas S. Huang. Dynamic clustering for efficient retrieval in high dimensional multimedia databases. Technical Report TR-MARS-97-10, University of Illinois at Urbana Champaign, 1997.

[20] Yong Rui, Thomas S. Huang, Sharad Mehrotra, and Michael Ortega. Automatic matching tool selection via relevance feedback in mars. *to appear in the 2nd Int. Conf. on Visual Information Systems*, 1997.

[21] Yong Rui, Thomas S. Huang, Sharad Mehrotra, and Michael Ortega. A relevance feedback architecture in content-based multimedia information retrieval systems. In *Proceedings of the IEEE Workshop on Content-based Access of Image and Video Libraries*. IEEE, 1997.

[22] Yong Rui, Alfred C. She, and Thomas S. Huang. Modified fourier descriptors for shape representation – a practical approach. In *Proceeding of First International Workshop on Image Databases and Multi Media Search*, 1996. Amsterdam, The Netherlands.

[23] G. Salton, Edward Fox, and E. Voorhees. Extended boolean information retrieval. *Communications of the ACM*, 26(11):1022–1036, November 1983.

[24] G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval*. McGraw Hill Computer Science Series, 1983.

[25] John R. Smith and Shih-Fu Chang. Tools and techniques for color image retrieval. In *IS & T/SPIE proceedings*, volume 2670, 1994. Storage & Retrieval for Image and Video Databases IV.

[26] Hideyuki Tamura et al. Texture features corresponding to visual perception. *IEEE Trans. Systems, Man, and Cybernetics*, SMC-8(6), June 1978.

[27] C. J. Van Rijsbergen. *Information Retrieval*. London: Butterworths, 1979.

A Fuzzy model evaluation algorithms

In this appendix, we present the algorithms used to compute the nodes in the query tree in the case of the fuzzy Boolean retrieval model. For simplicity we restrict ourselves to compute only binary nodes. That is, we assume that the query node Q has exactly two children, A, and B. Algorithms are presented for the following three cases: $Q = A \wedge B$, $Q = A \wedge \neg B$ and $Q = A \vee B$. Notice that we do not present an algorithm for only an unguarded negation (i.e., $Q = \neg A$) or a negation in the disjunction (i.e., $Q = (A \vee \neg B)$). Presence of an unguarded negation or negation in a disjunction does not make much intuitive sense. Typically, a very large number of images will satisfy the query and if such a negation is present in the query, it is best to rank each image and sort the answer set based on this ranking. We therefore only consider a restricted notion of negation when it appears within a conjunctive query.

In describing the algorithms the following notation is used.

- An image I is represented by two components, a key ($I.image$) and a degree of membership ($I.degree$). The key identifies the image and the degree of membership describes the match between the query feature and the database entries.
- A and B are assumed to be image streams from the child nodes. Each of these streams support the operations *Peek* and *GetNext* which look at and extract the next best element based on the degree of membership (i.e. they are retrieved in sorted order by degree of membership).

- Associated with each query node Q are three sets S_a, S_b and S_{res}. Initially each of these sets are empty. The query node Q extracts images from the child streams (that is, A and B) and may buffer them into S_a and S_b. The set S_{res} acts as a buffer of the images for the query node Q. Once a query node Q is able to estimate the degree of membership of image I for Q (that is, $f_Q(I)$), it places I in S_{res}. Thus, $I.degree$ refers to the degree of membership of I according to Q, where $I \in S_{res}$.

In describing the algorithms below we omit some critical error and boundary checking for clarity purposes, which have been addressed in the implementation.

A.1 Conjunctive Query with Positive Sub queries

The following algorithm computes the set of images ranked on their degree of membership to the query $Q = A \wedge B$, given input streams A and B which are ranked based on the degree of membership of images in A and B. In the algorithm, at each stage, the best image out of the sources A and B is chosen and added to a set of images S_a and S_b which function as buffers of images already observed from the corresponding stream. When an image is found that was already observed in the other stream, the loop is terminated since this is the next best image according to the query node Q. The resulting image is returned (that is, placed in the set S_{res}) with the degree equal to the minimum degree of the image in both streams.

```
Algorithm GetNextAnd_Fuzzy(A, B)
;returns: next best image in A and B
  while (TRUE)
    I_a= Peek (A), I_b= Peek (B)
    if I_a.degree > I_b.degree then
      I_a= GetNext(A)
      S_a = S_a ∪ I_a
      if I_a.image ∈ S_b then ;image already seen in B
        I_b= image S_b[I_a.image]
        exit loop
      end if
    else
    if I_b.degree > I_a.degree then
      I_b= GetNext(B)
      S_b = S_b ∪ I_b
      if I_b.image ∈ S_a then ;image already seen in A
        I_a= image S_a[I_b.image]
        exit loop
      end if
    end if
  end while
  ; reached upon finding a common image in S_a and S_b
  I.image =I_a.image
  I.degree = min(I_a.degree, I_b.degree)
  S_a = S_a − I_a, S_b = S_b − I_b, S_res = S_res ∪ I
  return I
```

A.2 Conjunctive Query with Negative Sub query

We next develop the algorithm for computing the query $Q = A \wedge \neg B$. The algorithm is different compared to the one developed for the conjunctive query with no negative sub query. Unlike the algorithm discussed earlier, only the stream for the node A is used in computing the degree of membership of images according to $A \wedge \neg B$. Images are retrieved from the input stream A in ranked order. For a given image I its degree of membership in the set associated with $\neg B$ is evaluated using the following function:

Probe(I, query): the Probe function returns the degree of membership of the image identified by $I.image$ in the fuzzy set associated with the query.

Let I be the image in S_{res} with the highest degree according to Q and let I_a be the next best image in rank order in stream A. An image is the next best match if either of the two conditions hold:

1. If $I.degree > I_a.degree$ then I is the next best match according to Q since all other images to follow will have the degree of membership according to Q less than $I_a.degree$.
2. If $I.degree \leq I_a.degree$ but $I_a.degree < Probe(I_a, \neg B)$ then I_a is the next best match according to Q since the membership of I_a is better than all the images already in S_{res}, and is also higher than all other images that will be produced in the future.

To find the next best match according to Q, As stream is traversed until one of the two conditions become true.

```
Algorithm GetNextAnd_Not_Fuzzy(A, B)
;returns: next best image in A and not B
  while (TRUE)
    I_a= Peek (A)
    if S_res ≠ ∅ ∧ I_a.degree < MaximumDegree(S_res) then
      I = image from S_res with maximum degree
      S_res = S_res − I
      exit loop
    else
      I_a= GetNext(A) ; consume from A
      I_b.image = I_a.image
      I_b.degree = Probe(I_a, ¬B)
      if I_a.degree ≤ I_b.degree then
        I = I_a
        exit loop
      else
        S_res = S_res ∪ I_b
      end if
    end if
  end while
  return I
```

A.3 Disjunctive Query

The following algorithm computes the set of images ranked on their degree of membership to the query $Q = A \vee B$, given input streams A and B which are ranked based on the degree of membership of images in A and B. The algorithm essentially consists of a merge but makes sure that an image that was already retrieved is ignored. This accomplishes the desired *max* behavior of the degree function associated with the disjunction in the fuzzy model.

```
Algorithm GetNextOr_Fuzzy(A, B)
;returns: next best image in A or B
  flag = TRUE
  while (flag)
    I_a= Peek (A), I_b= Peek (B)
    if I_a.degree > I_b.degree then
      I= GetNext(A)
    else
      I= GetNext(B)
    end if
    flag = FALSE
    if I.image ∈ S_res then
      flag = TRUE
    end if
  end while
  S_res = S_res ∪ I
  return I
```