EFFICIENT SUPPORT OF SOFT QUERY IN IMAGE RETRIEVAL SYSTEMS

Cyrus Shahabi and Yi-Shin Chen

Integrated Media Systems Center - Department of Computer Science
University of Southern California
$[shahabi, yishinc]@usc.edu$

ABSTRACT

We explore the use of soft computing and user defined classifications in multimedia database systems for content-based queries. With multimedia databases, due to subjectivity of human perception, an object may belong to different classes with different probabilities ("soft" membership), as opposed to "hard" membership supported by conventional database systems. Therefore, we propose a unified model that captures both hard and soft memberships. In practice, however, in the process of implementing our model by extending a conventional database system, we increase the query processing complexity significantly. To remedy for this increase, we propose a variety of techniques to scale down the complexity by orders of magnitude.

Keywords: image retrieval, fuzzy logic, information retrieval, soft query

1. INTRODUCTION

Numerous applications in digital library, entertainment industry, consumer products and e-commerce domains require access and query of repositories of image data. Examples are virtual museums, movie special effect softwares, family photo search tools, and e-store catalogue search tools, respectively. The challenge is to bridge the gap between the physical characteristics of digital images (e.g., color, texture) that are used for low level comparison, and the semantic meaning of the images conceptualized by humans to query the database.

Several studies [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16] focused on supporting similarity queries on image perceptual features such as color and texture. They either proposed algorithms for automatic extraction of these features or developed efficient indexing techniques for the extracted features. This body of work is orthogonal to our study and can be used by our proposed system as well. The main drawback is, however, the independence of these algorithms of the user perceptions.

Therefore, other studies [17, 18, 19, 20, 21, 22, 23, 24, 25] address this shortcoming by proposing various learning mechanisms to modify system parameters iteratively after obtaining user relevance feedback. This fine-tuning is performed for the entire user body, collectively.

However, different persons have different perceptions on the same set of images. In addition, other semantic properties of images such as the style of a painting are not incorporated into the model. This incorporation can be generalized as different ways of classifying images. Images can be classified into many groups by either humans or algorithms. Examples of automatic image classification algorithms are various techniques proposed to MPEG-7 standard committee [26]. To overcome the above challenges, we proposed a uniform model, *Soft Query Model* [27], which uses soft computing and user defined classifications in multimedia database systems for content-based queries. This model captures the experts' evaluations uniformly whether they are human experts or algorithms/functions. Using the concepts borrowed from fuzzy logic [28], the model allows users to assign different confidence values to different experts in order to support a query. Therefore, the same query might return different results with various rankings depending on the user submitting the query and the fuzzy cut used in the user profiles.

Although our experiments [27] illustrated that the soft query model consistently outperforms the conventional image retrieval system in matching the users' expectations, in practice the complexity of query processing is increased significantly. This is because for every query, the system needs to examine the evaluations of all the experts whom are trusted by the user submitting the query. That is, the time complexity is $O(u \log u)$, where u is the number of users and the space complexity is $O(upq)$, where p is the number of images and q is the number of classes.

In this paper, we address the complexity problem by proposing a novel method of computing fuzzy aggrega-

THIS RESEARCH HAS BEEN FUNDED IN PART BY THE INTEGRATED MEDIA SYSTEMS CENTER, A NATIONAL SCIENCE FOUNDATION ENGINEERING RESEARCH CENTER, COOPERATIVE AGREEMENT NO. EEC-9529152, BY NASA/JPL CONTRACT NR. 961518, AND UNRESTRICTED CASH/EQUIPMENT GIFTS FROM INTEL AND NCR.

tions. With our optimized method, the time complexity can be reduced to $O(f \log u)$ where f is the number of fuzzy sets (which is a small constant). We prove that our technique to compute fuzzy aggregations is equivalent to the conventional methods and hence generates identical results. Finally, we conducted an experimental study to compare the performance of our optimized method with that of the conventional methods. Our results demonstrated that the *optimized Soft Query Model* outperforms the conventional methods by orders of magnitude.

The remainder of this paper is organized as follows. In Section 2, we provide an overview of the fuzzy logic and its operations. Section 3 explains our $optimized$ model for the soft query system. In Section 4, we discuss the behavior of the system and how the proposed model can be used to process various queries. The results of our evaluations as well as the details of the system implementation and our benchmarking method are described in Section 5. Section 6 concludes the paper and presents some of our future research plans.

2. OVERVIEW OF FUZZY SETS

The concept of possibility theory was first introduced by Zadeh[28]. A possibility measure $\prod$ on a universe U is a set function from $p(U)$ to [0,1], where $p(U)$ denotes the set of subsets of U. $\prod$ is defined by Definition 2.1.

Definition 2.1:
$\prod(\emptyset) = 0$
$\prod(U) = 1$
$\forall A \in p(U), \forall B \in p(U)$
Union rule: $\prod(A \bigcup B) = max(\prod(A), \prod(B))$
Conjunction rule: $\prod(A \bigvee B) = max(\prod(A), \prod(B))$
Intersection rule: $\prod(A \bigcap B) = min(\prod(A), \prod(B))$
Disjunction rule: $\prod(A \bigwedge B) = min(\prod(A), \prod(B))$
Negation rule: $\prod(\neg A) = 1 - \prod(A)$ ∎

Generally, a possibility measure $\prod$ can be built from a possibility distribution π, which is a function from U to [0,1]. The term - *fuzzy set* in this paper represents a possibility distribution. A fuzzy set F is defined as:

Definition 2.2: A fuzzy set F in X is expressed as a set of ordered pairs where each pair represents a discourse and its corresponding membership. That is:
$X = \{x |$x is a discourse$\}$
$\mu_F(x) =$ membership of x in the fuzzy set F
$F = \{(x, \mu_F(x)) \mid x \in X\}$ ∎

A fuzzy set *F* can be cut at height α (α-cut), that is:

Definition 2.3: If $\alpha \in$ (0,1], then
$$(\mu_F)_\alpha(x) = \begin{cases} 1 & \text{if } \mu_F(x) \geq \alpha; \\ 0 & \text{otherwise}. \end{cases}$$
$F_\alpha = \{x \mid (\mu_F)_\alpha(x) > 0\}$ ∎

3. SOFT QUERY MODEL

3.1. Image Classes

Each image is associated with two categories of attributes. The first category is called *features* (denoted $\Im$) whose attributes are user-independent. The other category is called *properties* (denoted T) whose attributes are user-dependent. Every image can have several features and/or properties.

Definition 3.1: The domain D is an image set. The domain V is a feature value set. $\Im$ is a function from D onto V.
$D = \{d|$ d is an image$\}$
$V = \{v|$ v is a value$\}$
$\Im : d \in D \rightarrow v \in V$ ∎

Definition 3.2: Each property T_i partitions the set of all images into k classes $C_1^{T_i}, C_2^{T_i} \cdots C_k^{T_i}$, where the classes might overlap and/or be partial.

The membership of an image to a class $C_j^{T_i}$ is determined by an evaluator (denoted as e) via the following probability function: $P(n \in C_j^{T_i} \mid e) \in (F \vee R)$, where the value of $P(n \in C_j^{T_i} \mid e)$ could be either a fuzzy set or a real value within the range of 0 to 1. Moreover, different evaluators can classify same image into different classes. ∎

Using the same model, we can capture conventional image retrieval operations such as finding all images similar to a query image in color, shape or texture. Moreover, the model can capture the presence of multiple algorithms for the same comparison function. To illustrate, consider the following example.

Example 3.1: The property, $Color-Similarity$, partitions the set of all images into k classes, $C_1^{Color-Similarity}$ to $C_k^{Color-Similarity}$. $C_j^{Color-Similarity}$ is a class of all images similar in color to image j. The membership of images to these classes are:
$P(image1 \in C_1^{Color-Similarity} \mid$ QBIC-$\mathcal{A}_0) = 1$
$P(image1 \in C_2^{Color-Similarity} \mid H_i$ Function$) = 0.7$
$P(image1 \in C_2^{Color-Similarity} \mid userA) = Middle$
This example illustrates the flexibility of the model where evaluators can be algorithms, functions or humans. It also shows that the membership value can be a fuzzy set or a real value. ∎

In Example 3.1, although the same model is used to capture the evaluation by both users and functions, the actual processing methods are different. The membership values, which are determined by real users, are stored values. That is, the membership values exist in the system a priori. However, the membership values determined by functions or algorithms are derived values computed on demand when needed.

3.2. User Recommendation and Confidence

The model described in Section 3.1, assumes that there always exists a membership value for any image-user pair to a given class. However, there might be situations where a user does not provide a membership value for an image to a property class. The user might not be familiar with the class, or he/she has not yet assigned membership values for some images to a newly introduced class, or the user may not feel confident enough to perform the classification. In this case, the system requires some reference data (e.g. the values assigned by other evaluators) to estimate the membership values of images to the class. Therefore, storing each user's level of confidence to other users is critical in our model.

Definition 3.3: E denotes a set of evaluators in the database. O represents the set of observers who have assigned reference confidence values to evaluators. π is a confidence value for an observer o to an evaluator e. A confidence value can only be a fuzzy set[1].

$\pi : o \in O, e \in E \rightarrow b \in F$ ∎

Due to the subjectivity of human perception, the same fuzzy term might have different meanings to different users. Therefore, before referencing evaluators' opinions, all fuzzy sets should be converted to real values using the fuzzy α-cut value. The fuzzy α-cut value, representing a standard for human perception, can vary for different users. The system discards the membership values that are less than α-cut in fuzzy sets.

Definition 3.4: The value $(\pi_{o,e})_\alpha$ is a set of values whose memberships to $\pi_{o,e}$ are larger than the α-cut value. Furthermore, if $P(i \in c \mid e)$ is a fuzzy set, then $P(i \in c \mid e)_\alpha$ for image i to class c under the definition of evaluator e is a set of values whose memberships to $P(i \in c \mid e)$ are larger than the α-cut value. The definitions above can be expressed as:

$(\pi_{o,e})_\alpha = \{x | x \in X, \mu_{\pi_{o,e}}(x) > \alpha\}$
$P(i \in c \mid e)_\alpha = \{x | x \in X, \mu_{P(i \in c|e)}(x) > \alpha\}$ ∎

[1] In the original *Soft Query Model*, we also allowed the confidence value to be a real value between 0 and 1. However, we restricted this setting in order to scale down the computation complexity.

Example 3.2:
$\{low\} = \{(0.2, 0.8), (0.1, 0.9), (0, 0.4), (0.3, 0.6)\}$
$\{high\} = \{(0.8, 0.5), (0.9, 0.7), (1, 1), (0.7, 3)\}$
$\pi(B, C) = High, \quad \pi(G, A) = Low$
$P(image1 \in C^{Style}_{classic} \mid B) = High$
$P(image1 \in C^{Style}_{modern} \mid G) = Low$
If the α-cut for user B is 0.65 and α-cut for user G is 0.7, then $(\pi_{B,C})_{0.65}$ is $\{0.9, 1\}$, and $(\pi_{G,A})_{0.7}$ is $\{0.2, 0.1\}$. Similarly, $P(image1 \in C^{Style}_{classic} \mid B)_{0.65}$ is $\{0.9, 1\}$, and $P(image1 \in C^{Style}_{modern} \mid G)$ is $\{0.2, 0.1\}$. ∎

4. SOFT QUERY BEHAVIOR

Figure 1 illustrates the behavior of the system. The flow of image query processing is described in Section 4.1. The system evaluation method is then presented in Section 4.2.

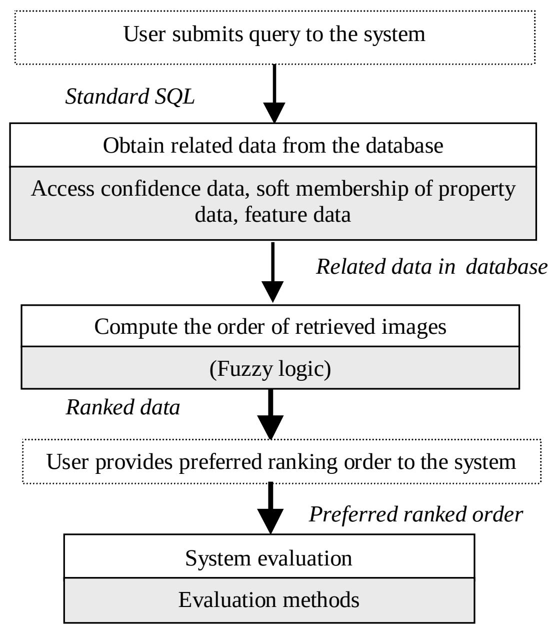

Figure 1: The Flow Diagram of the Soft Query System

4.1. Query Processing Model

Each query statement can be decomposed into several atomic clauses. Some clauses access and process feature-values and the others property classes. Therefore, our soft query model consists of two different query processing methods: feature query and property query. These two query models compute the image membership separately and combine their results at a final step. The final query results, group of images whose membership values are above user perceptual standard, are ordered according to their membership values. The feature query model is described in Section 4.1.1 and the property query model is described in Section 4.1.2. Finally, the combination process is discussed in Section 4.1.3.

4.1.1. Feature Queries

In this section, we discuss the processing of queries on the features of images. For example, "Find all images that cost more than $5000." The results of these queries are independent of the users submitting the queries. Moreover, a binary number determines whether in image belongs to the result set (value 1) or not (value 0).

The membership $\delta_{d,\sigma}$ of an image d to feature σ can be defined as:

Definition 4.1:
$$\delta_{d,\sigma} = \begin{cases} 1 & \Im(d) \text{ satisfies } \sigma \\ 0 & \text{otherwise} \end{cases} \blacksquare$$

Subsequently, we need to compute the membership values by applying Boolean operations on atomic clauses. The Boolean operations are well-known, and hence we will not explicate the details here.

4.1.2. Property Queries

In this section, we discuss the processing of queries on the image properties. For example, "Find all images with the *classic* style." The results of the queries depend on the user submitting the query. Moreover, a real number between 0 and 1 represents the probability of an image belonging to the result set. The challenge is how to take all the information about the trusted evaluators and their classifications into account when processing a query submitted by user u.

The membership $\lambda_{d,c}$ of image d to a property class c given user u can be computed using an averaging equation. The operations in the averaging equations should satisfy *monotonicity*, *commutativity* and *associativity*[2, 28], such as $max()$, $min()$, $+$, $\times$, etc.. There are several equations satisfying these properties. The equations we proposed [27] are:

$$\begin{aligned} \theta_{d,c,e} &= \pi_{u,e} \times P(d \in c \mid e) \\ \lambda_{d,c} &= \max_{e \in E}\{\theta_{d,c,e}\} \end{aligned} \quad (1)$$

$$\begin{aligned} \theta^{'}_{d,c,e} &= \min(\pi_{u,e}, P(d \in c \mid e)) \\ \lambda_{d,c} &= \max_{e \in E}\{\theta^{'}_{d,c,e}\} \end{aligned} \quad (2)$$

Storing the information of $\theta_{d,c,e}$ or $\theta^{'}_{d,c,e}$ is impractical and hence, we need to compute $\theta_{d,c,e}$ or $\theta^{'}_{d,c,e}$ by iterating through all possible values of e. Suppose the complexity of retrieving $\pi_{u,e}$ or $P(d \in c \mid e)$ is $O(1)$ using data structures such as Hash, the computation complexity of processing a query using these two equations is $O(|E| \times |D|)$, where $|E|$ is the size of the user set and $|D|$ is the size of the image set. On the other hand, the computation complexity will be $O(|E|log|E| \times |D|)$ when the complexity of retrieving information from database is $O(log|E|)$ using tree-like data structures such as B^+-tree. Note that according to our experiments, the Equations 1 and 2 are only slightly different in accuracy.

Since the $\lambda_{d,c}$ value comes from the maximum value among $\theta_{d,c,e}$ or $\theta^{'}_{d,c,e}$, we do not need to check all possible values of $\theta_{d,c,e}$ or $\theta^{'}_{d,c,e}$, but only need to focus on potential candidates. From this intuition, we provide two alternative equations in order to improve the efficiency:

$$\begin{aligned} E_f &= \{e \mid e \in E, \pi_{u,e} = f\} \\ \theta^{*}_{d,c,f} &= f \times \max_{e \in E_f}\{P(d \in c \mid e)\} \\ \lambda_{d,c} &= \max_{f \in F}\{\theta^{*}_{d,c,f}\} \end{aligned} \quad (3)$$

$$\begin{aligned} \theta^{**}_{d,c,f} &= \min(f, \max_{e \in E_{u,f}}\{P(d \in c \mid e)\}) \\ \lambda_{d,c} &= \max_{f \in F}\{\theta^{**}_{d,c,f}\} \end{aligned} \quad (4)$$

The value of $\lambda_{d,c}$ from Equations 3 and 4 can be proven to be identical to the corresponding values computed by Equations 1 and 2. Please refer to Appendix A for the proof.

In order to obtain the maximum value of $\theta^{*}_{d,c,f}$ or $\theta^{**}_{d,c,f}$ for one image, we need to compute them by iterating through all possible values of the fuzzy term f. Therefore, using data structures such as Hash, the computation complexity of processing one query using these two equations is $O(|F| \times |D|) = O(|D|)$, where $|F|$ is a small constant representing the number of possible values for the fuzzy term f and $|D|$ is the size of the image set. However, with tree-like data structures such as B^+-tree, the computation complexity will be $O(|F|log|E| \times |D|) = O(log|E| \times |D|)$.

Next, we need to convert all fuzzy terms appearing in equations to real values for computation. Because the value of $\pi_{u,e}$ is a fuzzy term f^*, the system assigns the value of $\max\{\ f^*_{\alpha_u}\}$ to it. Likewise, if the value of $P(d \in c \mid e)$ is a fuzzy term f', the system assigns $\max\{f'_{\alpha_e}\}$ as its value. (The α value for the evaluator j is denoted as α_j). If the set $f^*_{\alpha_u}$ is empty, the system assigns zero as the value of $\max\{\ f^*_{\alpha_u}\}$. In the same way, if the set f'_{α_u} is empty, the system assigns zero as the value of $\max\{\ f'_{\alpha_u}\}$.

Note that there are cases where the system does not have enough information to compute $P(d \in c \mid e)_{\alpha_e}$ or $(\pi_{u,e})_{\alpha_u}$. In these cases, the system estimates the value from the "*system profile*". Below, we first list these cases and then explain two alternative methods to estimate the values from the *system profile*.

- No evaluator in E provides a membership value for image d to class c (i.e., $\forall u \in E, P(d \in c \mid e) = \emptyset$).
- There is no confidence value from user u who submitted the query to any evaluator (i.e.,$\forall e \in E, \pi_{u,e} = \emptyset$).
- The confidence value from user u who submitted the query only exists for those evaluators who have not assigned membership value for image d to class c (i.e.,$\forall e \in E \mid \pi_{u,e} = \emptyset$ then $P(d \in c \mid e) = \emptyset$).

The membership values provided by *system profile* for image d to class c can be computed using one of the following approaches:

- Averaging the membership values provided by all of the evaluators for image d to class c.
- Using some algorithms for computing the membership value of image d to class c automatically.

If there is no membership value either stored or derived for image d to class c in the *system profile*, then the membership value is considered to be zero.

Fuzzy Set Data				
Low	(0.4, 0.45)	(0.3,0.68)	(0.2,1)	(0.1,0.75)
High	(0.8,0.5)	(0.9,0.7)	(1,1)	(0.7, 0.3)

Image Membership Information

Image	Class	Evaluator	P(d ∈ C | e)
1	$C^{style}_{classical}$	A	High
		B	Low
	C^{style}_{modern}	A	Low
2	$C^{style}_{classical}$	A	Low
		B	Low
		C	Low
	C^{style}_{modern}	A	High
4	$C^{style}_{classical}$	B	High
		C	High

System Profile

Image	Class	P(d ∈ C | e)
1	$C^{style}_{classical}$	0.7
2	$C^{style}_{classical}$	0.3
3	$C^{style}_{classical}$	0.6
4	$C^{style}_{classical}$	0.5

Fuzzy Cut Information

Evaluator	α
A	0.65
B	0.7
C	0.5
D	0.4

Confidence Information

Observer	Evaluator	π(u,e)
A	C	Low
A	B	Low
A	A	High

Image Feature Information

Feature	Image	Value
Cost	1	\$10,000
	2	\$5,000
	3	\$7,000
	4	\$8,000

Figure 2: Sample data for Example 4.1

Example 4.1: Figure 2 represents a set of sample data assumed for this example. Suppose user A submits a query requesting the list of all images in class $C^{style}_{classic}$. The system computes the soft membership value for each image (using Equation 3) in two phases. In phase one, the system first retrieves $\pi_{A,A}$ from the database. Next, since the value of $\pi_{A,A}$ is a fuzzy set *High*, the value of $\max\{(High)_{\alpha_A}\}$ should be computed as:

$\max\{(High)_{\alpha_A}\}$
$= \max\{\{(0.8, 0.5), (0.9, 0.7), (1, 1), (0.7, 0.3)\}_{0.65}\}$
$= \max\{0.9, 1\} = 1$

Hence, the system assigns the value 1 to $\pi_{A,A}$. The same procedure is applied to compute $\pi_{A,B}$ and $\pi_{A,C}$.

In phase two, the system first retrieves $P(image1 \in C^{style}_{classic} \mid A)$. Next, it identifies the value of $P(image1 \in C^{style}_{classic} \mid A)$ as a fuzzy set *High*, thus the value of $\max\{(High)_{\alpha_A}\}$ which is 1 should be assigned to $P(image1 \in C^{style}_{classic} \mid A)$. The same procedure can be applied to compute $P(image1 \in C^{style}_{classic} \mid B)$ and $P(image1 \in C^{style}_{classic} \mid C)$. Finally, we compute the membership values:

$\lambda_{image1, C^{style}_{classic}}$
$= \max\{(1 \times 1), (0.3 \times 0.2), (0.67 \times 0)\} = 1$,
$\lambda_{image2, C^{style}_{classic}} = 0.3$,
$\lambda_{image4, C^{style}_{classic}} = 0.67$
$\lambda_{image3, C^{style}_{classic}}$ = 0.6 (the system profile value is retrieved because $\forall e \in E, P(d \in c \mid e) = \emptyset$) ∎

We also need to consider the cases where the query includes a Boolean combination of atomic clauses. We use the standard rules of fuzzy logic, which are described in Definition 2.1, instead of traditional Boolean operations. The idea is illustrated by the following example.

Example 4.2: Consider the same data as in Example 4.1. Suppose user A submits a query requesting the list of all images in either class $C^{style}_{classic}$ or class C^{style}_{modern}. Similar to Example 4.1, membership values are computed for C^{style}_{modern} as follows:

$\lambda_{image1, C^{style}_{modern}} = \max\{1 \times 0.3, 0.3 \times 0, 0.67 \times 0\} = 0.3$,
$\lambda_{image2, C^{style}_{modern}} = 1$
$\lambda_{image3, C^{style}_{modern}} = \lambda_{image4, C^{style}_{modern}} = 0$

(Because $\forall e \in E, P(d \in c \mid e) = \emptyset$ and no such data in system profile)

Now, we need to combine the two membership values to satisfy the query as:

$\lambda_{image1,(C^{style}_{classic} \vee C^{style}_{modern})}$
= $\max$(1,0.3)= 1, $\lambda_{image2,(C^{style}_{classic} \vee C^{style}_{modern})}$ = 1,
$\lambda_{image3,(C^{style}_{classic} \vee C^{style}_{modern})}$ = 0.6
$\lambda_{image4,(C^{style}_{classic} \vee C^{style}_{modern})}$ = 0.67 ∎

4.1.3. Combination of Feature and Property Queries

Finally, in this section, we explain the processing of queries on both features and properties of images. For example, "Find all *classic* images that cost less than $5000." For these cases, we continue to use the standard rules of fuzzy logic as described in Definition 2.1. The only difference is that the possibility of an image belonging to a feature class is either 0 or 1. After computing probabilities of all images the system only returns the images whose probabilities are higher than user perceptual standard (user fuzzy cut).

Therefore, our proposed query processing model for *soft query* can capture user queries on both features and properties in a unified manner. To compare our model with the conventional image retrieval systems, consider the typical query by example on color, shape, and texture of images. Conventional systems compute a weighted average over these perceptual features to measure the similarity distance between two images. The weights are assigned and fine-tuned either directly by the user, or by the system after several iterations monitoring the user feedbacks. For example, the system will assign higher weights to the color feature for a color oriented user. These perceptual features can also be modeled within our system as different properties. Subsequently, their weights are assigned not only by the user and his/her previous feedbacks but also by the other users/evaluators trusted by this user. In addition, our model can capture various feature extraction algorithms, semantic classes (e.g., image style) and soft memberships.

4.2. Query Evaluation

To evaluate the accuracy of our proposed approach, we need to compare the results returned by our system for a query submitted by a user u with the exact results that u expects to observe. Two most commonly used metrics in information retrieval are precision and recall. In image databases, for example, precision is used to examine the system's ability to find images that match the query image. Recall is a measure of the extent to which all the images of the class matching the query image are found. This evaluation method is not quite applicable to our soft query system. Unlike current image retrieval systems, the soft query system deals with the ambiguities associated with classifying images. The classification problem has no standard answer.

Therefore, we use an alternative relevant performance measure: *prediction quality*. To illustrate, consider Table 1, which contains the results of a query and the corresponding user feedback.

Table 1: An example of a query result and the user feedback

	I1	I2	I3	I4	I5
Query Result Q	0.92	0.89	0.82	0.77	0.73
User Feedback B	0.92	0.89	0.9	0.85	0.88

The query result row reflects the list of soft membership values per image for a given query class computed by the soft query system. The feedback row represents user's expectation. For example, user thinks image $I3$ is more likely to belong to the query class than image $I2$. Therefore, the user modifies the rating of $I3$ to 0.9[2], which is higher than that of $I2$. Subsequently, the system measures the degree of similarity between these two vectors as:

$$\text{Similarity Distance}(Q, B) = \frac{\sum_{i=1}^{n} Q_i \times B_i}{\sqrt{\sum_{i=n}^{n} Q_i^2 \times \sum_{i=n}^{n} B_i^2}} \quad (5)$$

This similarity value can be considered as an acceptance rate. For example, if the user does not modify any rating in the query result, the similarity value will be 1. This suggests that the user is in total agreement with the query result. On the other hand, if the measure is less than 1, it represents that the user does not totally agree with the query result.

5. PERFORMANCE EVALUATION

In this section, we first describe our conceptual schema designed to implement the model discussed in Section 5.1. Subsequently, we explain how our benchmarking method is used to populate the schema. Finally, the details of our experimental results on the benchmark are discussed.

5.1. Experimental Setup

Our soft query system is implemented in Java and on top of Informix Universal Database Server V9.14[3] with Excalibur Image DataBlade Module[4] which is running on a 300MHZ dual processor Sun Enterprise 450 Server with SunOS 5.6.

In our experimental setup, we use the ternary relationship for recording the membership values of images in property

[2] For the real application, the user might simply drag the image and move it up in the list using a couple of mouse clicks.

[3] http://www.informix.com/informix/resource/udo914.htm

[4] http://www.intraware.com/app/shop?page=product&plne=000110

classes. The confidence values for users/evaluators assigned by each observer are stored in the *Confidence-Data* relation. The relationship *User* contains the user information as well as the user's perceptual standard - user fuzzy cut. The image features are stored along with the image itself (or its handle) in the *Image* entity set. This is because the features are independent of the users. The experimental system also has an entity *Fuzzy-Set* for maintaining the system fuzzy information.

5.2. Benchmarking Method

In order to populate our schema for evaluation purposes, we propose a parametric algorithm. By changing the parameters of the algorithm, we can simulate various database contents. The algorithm first randomly generates a cube that contains the perfect knowledge about the classification behaviors of all the users. Next, it utilizes the cube to populate *Confidence-Data* and *Soft-Membership-Data* relations. However, during this process, it introduces some noise into the data.

The cube has the following three dimensions: users, classes and images. Each cell $\mathcal{C}(i,j,k)$ determines the membership value of image k to class j assigned by user i. The cells are each assigned a random value uniformly distributed between 0 and 1. The *system profile* is also considered as a user row in this cube. Its membership value for each image x to class c is obtained by averaging over all the users' membership values for x to c. Since the system profile is the only information utilized by conventional image retrieval systems, its membership value is computed assuming the perfect knowledge (however, simply averaged) on users classification behaviors.

To assign confidence values for evaluators in the relation, *Confidence-Data*, we compare their classification behaviors (randomly selected from the cube) to the classification behavior of the observer. The closer their behaviors, the higher the confidence values. Subsequently, we **randomly** select cells for every user to populate the *Soft-Membership-Data* relation. The number of cube cells that we use to populate both *Soft-Membership-Data* and *Confidence-Data* is determined by a *selectivity factor*. Thus, the lower the selectivity factor, the less knowledge our system has about its users. To incorporate imperfect or wrong knowledge into our system, prior to populating the cube information to relations, all information will be tuned in a noisy process. Finally, all the real values between 0 and 1 must be translated into fuzzy terms. To achieve this, we use the randomly populated *Fuzzy-Cut* attribute of the *User* relation, which is bounded by [0.3 0.8].

5.3. Results

We conducted several sets of experiments to compare our soft query system with one conventional image retrieval approach (denoted hereafter as CIR). In these experiments, we observed a significant margin of improvement in matching the user expectations in various settings. Since we proved that the query results are equivalent to those we proposed in [27], we only provide a summary of the results in Figure 3.

The results shown for each set of experiments are averaged over many runs, where each run is executed with different seeds for the random generator functions. The coefficient of variance among these runs is always smaller than 5% [5], which shows the independence of our results from a specific run.

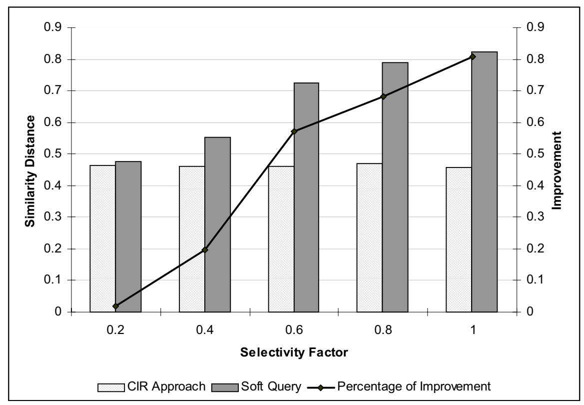

Figure 3: Impacts of Selectivity Factors

In Figure 3, the X-axis is the selectivity factor varing from 0.2 to 1. We fix the number of images (X) at $X = 50$, the number of users (Y) at $Y = 20$, the number of classes (Z) at $Z = 5$ and introduce no noise into the data. The Y-axis depicts both the similarity distance computed by Equation (5) and the percentage of improvement over CIR, which is computed as:

$$\text{Percentage of Improvement}(SQ, CIR) = \frac{SQ - CIR}{CIR} \tag{6}$$

Figure 3 depicts that the higher the selectivity factor, the better the performance of our soft query system. This is because CIR cannot benefit from the additional information provided by the higher values of selectivity factor. When the selectivity factor is 20%, CIR and soft query perform almost identically because in most cases, soft query does not have enough information to avoid consulting the system profile. That is, its approach is reduced to that of CIR.

In addition, we also compare the computation time between the *optimized* and the *original* Soft Query models. In

[5] The variance among these runs is smaller than 1.5%.

theory, the *original* Soft Query model increases the query processing complexity significantly. When we fix the number of images and use a tree-like data structure the time complexity is $O(|E|\log|E|)$. Contrastly, the time complexity of the *optimized* Soft Query Model is $O(|F|\log|E|) = O(\log|E|)$, where $|F|$ is a small constant representing the number of fuzzy terms. Our experimental results (Figure 4) shows the difference between the *original* and the *optimized* models. Figure 5 verifies the above argument.

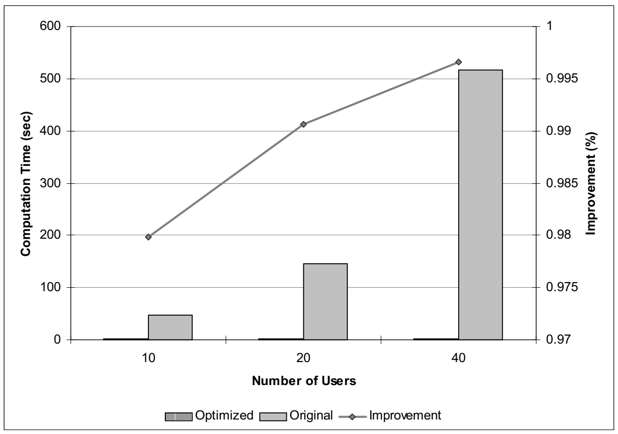

Figure 4: Computation complexity experiments

In Figure 4, the X-axis is the number of users varying from 10 to 40. The Y-axis depicts both the computation time (left axis) and the percentage of improvement (right axis). Since the selectivity factor and the level of noise will not affect the results, we fix the selectivity factor at 0.6 and the level of noise at 0. Moreover, because the computation complexity of both the optimized and the original models are linear to the number of images (see Appendix A), we fixed the number of images at 50 for simplicity. As shown in Figure 4, the larger the number of users, the higher the percentage of improvement. This is because the computation complexity of the *optimized* model is less than that of the *original* model.

In Figure 5, the X-axis is the number of users. The Y-axis depicts the computation time. For the same reasons as in Figure 4, we fixed the selectivity factor at 0.6, the level of noise at 0, and the number of images at 50. We notice that the curve for the computation time of the *optimized* model is similar and close to the curve of $log_{8.5}|E| + 0.1$. Therefore, the computation complexity of *optimized* model is $O(log|E|)$ while the number of users is fixed.

6. CONCLUSION

We proposed a unified and efficient model to support soft querying and classification on image databases. Towards this end, we formally described our model and demon-

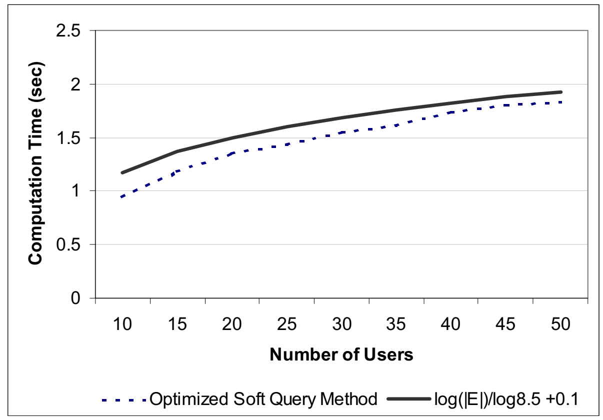

Figure 5: Optimized computation time and $log_{8.5}|E| + 0.1$

strated its power in capturing various query and retrieval scenarios. The model borrows heavily from the field of fuzzy logic. The model maintains a profile for each user, which contains the membership value for every image-class pair. For the cases where values for some of these pairs are missing, the model provides estimation based on the level of confidence that this user has on other users. To support a user query, the model combines the profile information with conventional feature similarity algorithms, in a unified manner. In the very worst case where no profile is available for a user, the model reduces itself to a conventional image retrieval system and only consults system provided algorithms and profiles. We showed that if conventional methods are used for fuzzy aggregations, the complexity of soft query will be significant. Therefore, we proposed a novel method of computing the fuzzy aggregations, which reduces the complexity of soft queries by orders of magnitudes. The performance improvement of the optimized model was verified by our experimental study. In addition, the accuracy of the optimized model in generating identical results to those of the conventional methods has been formally proved.

We intend to extend this study in three ways. First, we plan to populate our database with real images and compare our approach with conventional systems using a real application (Figure 6 depicts the graphical user interface of the application). Second, we intend to investigate alternative clustering technology to further reduce the complexity and storage requirements of our approach. We expect that clustering would reduce the storage complexity from $O(Y^2 + X \times Y \times Z)$ to $O(X \times Z)$. Finally, we want to extend the system to utilize the users' relevance feedbacks to improve the profiles iteratively.

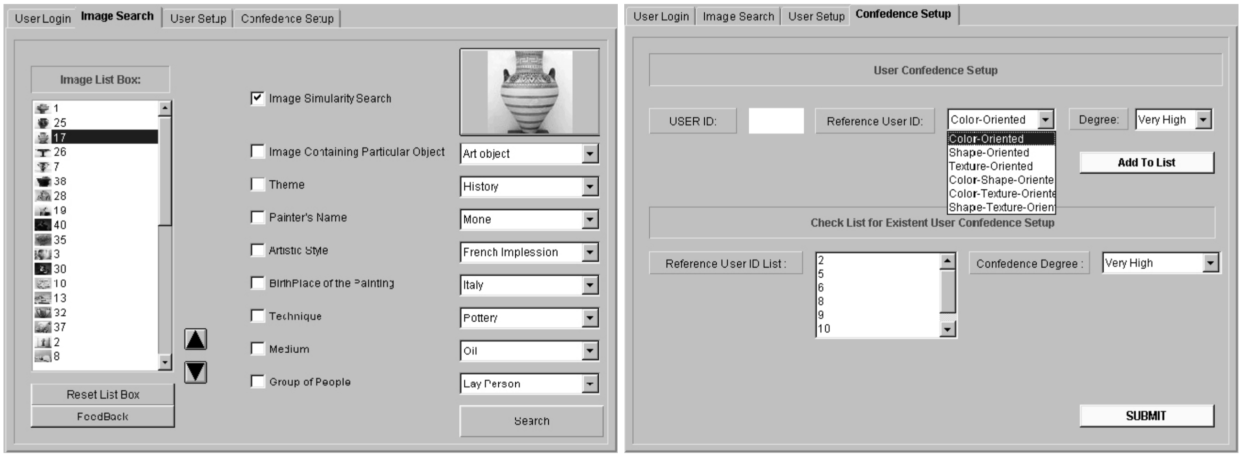

Figure 6: Interfaces of the soft query application

A. PROOF OF EQUIVALENCE

Given, $E_{u,f} = \{e \mid e \in E, \pi_{u,e} = f\}$

$$\lambda_{d,c} = \max_{f \in F}\{f \times \max_{e \in E_{u,f}} \{P(d \in c \mid e)\}\}$$
$$\lambda^*_{d,c} = \max_{e \in E}\{\pi_{u,e} \times P(d \in c \mid e)\}$$

We want to prove $\lambda^*_{d,c} = \lambda_{d,c}$

Proof:

First, we prove that $\lambda_{d,c} \geq \lambda^*_{d,c}$

Suppose $e^* \in E$ is an evaluator where the value of $\pi_{u,e^*} \times P(d \in c \mid e^*)$ is the maximum among all the other evaluators $e \in E$.

Hence, denoting

$$f^* = \pi_{u,e^*} \in F \ \ and$$
$$p^* = P(d \in c \mid e^*)$$

then

$$\lambda^*_{d,c} = f^* \times p^* \tag{7}$$

$$Given \ \ \lambda_{d,c} = \max_{f \in F}\{f \times \max_{e \in E_{u,f}} \{P(d \in c \mid e)\}\}$$
$$and \ \ f^* \in F \ \ then$$
$$\lambda_{d,c} \geq f^* \times \max_{e \in E_{u,f^*}} \{P(d \in c \mid e)\}\} \tag{8}$$

$$and \ \ given \ \ p^* = P(d \in c \mid e^*)$$
$$\leq \max_{e \in E_{u,f}} \{P(d \in c \mid e)\}\}$$
$$and \ (8) \ then \ \ \lambda_{d,c} \geq f^* \times p^* \tag{9}$$
$$from \ (9) \ \ and \ (7) \ \Rightarrow \ \lambda_{d,c} \geq \lambda^*_{d,c} \tag{10}$$

Next, we prove that $\lambda^*_{d,c} \geq \lambda_{d,c}$.

Suppose

$$\lambda_{d,c} = f^{'} \times p^{'} \tag{11}$$

Hence, we can find an evaluator $e^{'}$ who satisfies these following conditions:

$$f^{'} = \pi_{u,e^{'}} \tag{12}$$
$$e^{'} \in E_{u,f^{'}} \in E \tag{13}$$
$$p^{'} = P(d \in C \mid u^{'}) \tag{14}$$

$$Give \ \ \lambda^*_{d,c} = \max_{e \in E}\{\pi_{u,e} \times P(d \in c \mid e)\}$$
$$by \ (13) \ \Rightarrow \lambda^*_{d,c} \geq \pi_{u,e^{'}} \times P(d \in c \mid e^{'}) \tag{15}$$
$$and(12)(14)(15) \ \Rightarrow \ \lambda^*_{d,c} \geq f^{'} \times p^{'} \tag{16}$$
$$from \ (16) \ and \ (11) \ \Rightarrow \ \lambda^*_{d,c} \geq \lambda_{d,c} \tag{17}$$

From (10) and (17) $\Rightarrow$ $\lambda^*_{d,c} = \lambda_{d,c}$ ∎

B. REFERENCES

[1] J. Assfalg, A. Del Bimbo, and P. Pala, "Querying by photographs: Using virtual reality for content-based image retrieval," in *Proceedings of the IEEE International Conference on Multimedia Computing and Systems*, 1999, vol. 1.

[2] R. Fagin, "Fuzzy queries in multimedia database systems," *ACM SIGACT-SIGMOD-SIGART Symposium on Principle of Database Systems*, 1998.

[3] R. Fagin, "Combining fuzzy information from multiple systems," in *Proc. Fifteenth ACM Symp. on Principles of Database Systems*, 1996, pp. 216–226.

[4] C. Faloutsos and et al., "Efficient and effective querying by image content," *J. of Intelligent Information System*, vol. 3, pp. 231–262, 1994.

[5] K. Hirata, S. Mukherjea, W. Li, and Y. Hara, "Integrating image matching and classification for multimedia retrieval on the web," in *Proceedings of the IEEE International Conference on Multimedia Computing and Systems*, 1999, vol. 1.

[6] W. Hu, M.S. Schmalz, and G.X. Ritter, "Image retrieval using the longest approximate common subsequences," in *Proceedings of the IEEE International Conference on Multimedia Computing and Systems*, 1999, vol. 2.

[7] D. Huttenlocher, W. Rucklidge, and G. Klanderman, "Comparing images using the hausdorff distance under translation," in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1992, pp. 654–656.

[8] D. Maeda, M. Mukunoki, and K. Ikeda, "A classification method of images based on composition and its application to image retrieval," in *Proceedings of the IEEE International Conference on Multimedia Computing and Systems*, 1999, vol. 2.

[9] R. McConnell, R. Kwok, J.C. Curlander, W. Kober, and S.S. Pang, "ψ-s correlation and dynamic time warping: Two methods for tracking ice floes in sar images," *IEEE Trans. Geoscience and Remote Scensing*, vol. 29, pp. 1004–1012, 1991.

[10] S. Medasani and R. Krishnapuram, "A fuzzy approach to content-based image retrieval," in *Proceedings of the IEEE International Conference on Multimedia Computing and Systems*, 1999, vol. 2.

[11] S. Nepal, M.V. Ramakrishna, and J.A. Thom, "A fuzzy object query language (foql) for image databases," in *Proceedings of the Sixth International Conference on Database Systems for Advanced Applications*, April 1999, pp. 117–124.

[12] S. Nepal and M.V. Ramakrishna, "Query processing issues in image(multimedia) databases," in *Proceeding of 15th International Conference on Data Engineering (ICDE-99)*, 1999, pp. 22–29.

[13] S. Nepal and M.V. Ramakrishna, "Query processing issues in image(multimedia) databases," in *21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR98)*, 1998, p. 386.

[14] W. Niblack, R. Barber, W. Equitz, M. Flickner, E. Glasman, D. Petkovic, and P. Yanker, "The qbic project: Querying imgaes by content using color, texture and shape," in *Proceeding of SPIE Conference on Storage and Retrieval for Image and Vedio Databases*, 1993, vol. 1908, pp. 173–187.

[15] M. Ortega, Y. Rui, K. Chakrabarti, K. Porkaew, and S. Mehrotra, "Supporting ranked boolean similarity queries in mars," *IEEE Transactions on Knowledge and Data Engineering*, vol. 10, no. 6, 1998.

[16] M. Stricker and M. Orengo, "Similarity of color images," in *Proceeding of SPIE Conference on Storage and Retrieval for Image and Vedio Databases III*, 1995, vol. 2420, pp. 381–392.

[17] C. Brambilla, A. Della Ventura, I. Gagliardi, and R. Schettini, "Multiresolution wavelet transform and supervised learning for content-based image retrieval," in *Proceedings of the IEEE International Conference on Multimedia Computing and Systems*, 1999, vol. 1.

[18] T. Chua, C. Chu, and M. Kankanhalli, "Relevance feedback techniques for image retrieval using multiple attributes," in *Proceedings of the IEEE International Conference on Multimedia Computing and Systems*, 1999, vol. 1.

[19] A.D. Doulamis, Y.S. Avrithis, N.D. Doulamis, and S.D. Kollias, "Interactive content-based retrieval in video databases using fuzzy classification and relevance feedback," in *Proceedings of the IEEE International Conference on Multimedia Computing and Systems*, 1999, vol. 2.

[20] R. Fagin and E.L.Wimmers, "Incorporating user preferences in multimedia queries," in *Proc. 1997 International Conference on Database Theory*, 1997.

[21] J.Huang, S.R.Kumar, and M. Mitra, "Combining supervised learning with color correlograms for content-based image retrieval," *ACM Multimedia 97 - Electronic Proceedings*, 1997.

[22] C. Meilhac and C. Nastar, "Relevance feedback and category search in image databases," in *Proceedings of the IEEE International Conference on Multimedia Computing and Systems*, 1999, vol. 1.

[23] K. Porkaew, S. Mehrotra, and M. Ortega, "Query reformulation for content based multimedia retrieval in mars," in *Proceedings of the IEEE International Conference on Multimedia Computing and Systems*, 1999, vol. 2.

[24] Y. Rui, T.S. Hung, and S. Mehrotra, "Content-based image retrieval with relevance feedback in mars," in *IEEE proc. of Int. Conf. on Image Processiong*, 1997.

[25] D. Squire, W. Muller, and H. Muller, "Relevance feedback and term weighting schemes for content-based image retrieval," in *ACM Multimedia 97 - Electronic Proceedings*, 1997.

[26] Rob Koenen, "Mpeg-7: Context, objectives and technical roadmap, v.12," *ISO/IEC JTC1/SC29/WG11/ N2861*, 1999.

[27] Cyrus Shahabi and Yi-Shin Chen, "Soft query in image retrieval systems," in *Proceeding of SPIE Internet Imaging , Electronic Imaging 2000*, 2000, vol. 3964, pp. 57–68.

[28] L.A. Zadeh, "Fuzzy sets as a basis for a theory of possibility," *Fuzzy Sets and Systems 1(1)*, pp. 3–28, 1978.