Soft Query in Image Retrieval Systems*

Cyrus Shahabi and Yi-Shin Chen
Integrated Media Systems Center and Computer Science Department
University of Southern California, Los Angeles, CA 90089-2564
E-mail:{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 traditional database systems, objects/tuples are grouped into classes/relations using "hard" membership. Hence, the result of a query to obtain the members of a class is a fixed set. With multimedia databases, however, an object may belong to different classes with different probabilities ("soft" membership). In addition, alternative users may classify objects differently due to subjectivity of human perception on multimedia objects. In order to remedy for this situation, we propose a unified model that captures both conventional techniques and soft memberships. We implemented the model by extending the traditional database query capabilities such that the result of a query depends on the user who submits the query. We compared our proposed system with conventional image retrieval systems and observed a significant margin of improvement in matching the user expectations.

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

1. INTRODUCTION

Several 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 comparison, and the semantic meaning of the images that are used by humans to query the database.

Several studies[1–16] focused on supporting 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 strived to address this shortcoming[17–25] by proposing various learning mechanisms to modify system parameters iteratively after obtaining user relevance feedbacks. This fine tuning is performed for the entire users body, collectively. However, different humans 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] These classifications are usually "soft" in the sense that the same image can be classified differently by different humans/algorithms. To the best of our knowledge, there is no study on incorporation of all the above factors into a unified model for querying.

We address the above mentioned challenge by proposing a uniform model to use soft computing and user defined classifications in multimedia database systems for content-based queries. We extended a conventional image retrieval system to support soft classification of images by both human users and software procedures. For this extension, we not only rely on the user profile but also on the profiles of other experts trusted by the user. The model captures the experts' evaluations uniformly whether they are human experts or algorithms/functions. Here, we borrow concepts from fuzzy logic[27] to allow a user 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.

*This research was supported in part by cash/equipment gifts from NCR, Intel, Informix, JPL/NASA contract nr. 961518 and NSF grants EEC-9529152 and MRI-9724567.

Our approach to consult the opinions of trusted evaluators when serving a submitted query, is similar in concept to the recommendation systems such as Recommender,[28] Firefly[29] and GroupLens.[30] However, we incorporated an alternative recommendation approach into our model by employing fuzzy logic operations. Therefore, the uniformity of the model is maintained.

To evaluate our approach, we focus on image classification and retrieval as our domain application. We built an experimental setup using an object-relational database system, namely Informix Universal Server, and extended it to support soft query. This system can also support the conventional image retrieval techniques based on image shape, color and texture using the Excalibur Image Datablade. Subsequently, we compare the system results with and without the user profiles. This is to verify that our approach of employing user profiles for image retrieval is better than those of hard membership classification with no profiles. The results show that our soft query model consistently outperformed the conventional image retrieval system in matching the users' expectations. In these experiments we varied the level of noise introduced in the users' profiles to avoid a "perfect knowledge" scenario.

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 unified 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.[27] 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:

$X = \{x | \text{x is a discourse}\}$

$\mu_F(x) =$ membership of x in the fuzzy set F

$F = \{(x, \mu_F(x)) \mid x \in X\}$ ∎

Example 2.3.: The following *"small-integer"* set is a *fuzzy set*:

small-integer = {(0,1), (1,1), (2,0.9), (3,0.7), (4,0.5), (5,0.2)}

From this definition, the membership of 0 and 1 in *small integer* is 1. The membership of 2 in *small integer* is 0.9 and so on. The memberships of those discourses of U that are not found in *small integer* are zero. ∎

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

Definition 2.4.: If $\alpha \in (0,1]$, then

$$(\mu_F)_\alpha(x) = \begin{cases} 1 & \text{if } \mu_F(x) \geq \alpha; \\ 0 & \text{otherwise.} \end{cases}$$

$F_\alpha = \{x \mid (\mu_F)_\alpha(x) > 0\}$■

EXAMPLE 2.5.: Suppose the α-cut of the fuzzy set - "small integer" in Example 2.3 is 0.8. We can rewrite the set as the following:

$small\text{-}integer_{0.8} = \{0, 1, 2\}$ ■

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.

$D = \{d \mid \text{d is an image}\}$

$V = \{v \mid v \text{ is a value}\}$

$\Im : d \in D \rightarrow v \in V$ ■

EXAMPLE 3.2.: For an image B, $Size(B) = 100 \times 100$, $Cost(B) = 10,000$, and $Purchase - Date(B) = 9/21/99$, where size, cost and purchase date are various features of B each with unique and user-independent values. ■

DEFINITION 3.3.: 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.■

EXAMPLE 3.4. : The property, *Style*, partitions the set of all images into two classes, $C_{classic}^{Style}$ and C_{modern}^{Style}. The membership of images to these two classes are:

$P(image1 \in C_{classic}^{Style} \mid userA) = High$

$P(image1 \in C_{modern}^{Style} \mid userA) = Low$

$P(image2 \in C_{modern}^{Style} \mid userA) = High$

$P(image1 \in C_{modern}^{Style} \mid userB) = Middle$

This example illustrates that classes $C_{classic}^{Style}$ and C_{modern}^{Style} overlap. It also demonstrates that different evaluators can classify same images 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 3.5.: 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-$\mathcal{A}_0$ color similarity algorithm$) = 1$

$P(image1 \in C_2^{Color-Similarity} \mid H_i$ 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 3.5, 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 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 3.6.: 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 be a fuzzy set or a real value between 0 and 1.

$E = \{e|\text{e is an evaluator}\}\ |E| = n$

$O = \{o|o \in E\}\ O \subseteq E$

$\pi : o \in O, e \in E \rightarrow b \in (F \vee R)$ ∎

EXAMPLE 3.7.:

$\pi(A, C) = 0.67$

$\pi(B, H_i \text{ Function}) = High$

$\pi(D, \text{ QBIC-}\mathcal{A}_0 \text{ color similarity algorithm}) = Low$

In this example, user A has confidence 0.67 in user C; user B has high confidence in H_i function; user D has low confidence in QBIC-$\mathcal{A}_0$ color similarity algorithm. ∎

Due to the subjectivity of human perception, the same fuzzy term might have different meanings according 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 which are lower than α-cut in fuzzy sets.

DEFINITION 3.8.: If $(\pi_{o,e})$ is a fuzzy set, then 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:

$(\pi_{o,e})_\alpha = \{x|x \in X, \mu_{\pi_{o,e}}(x) > \alpha\}$

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

EXAMPLE 3.9.:

$\{low\} = \{(0.4, 0.4), (0.3, 0.68), (0.2, 1), (0.1, 0.75), (0, 0.3)\}$

$\{high\} = \{(0.8, 0.5), (0.9, 0.7), (1, 1), (0.7, 3)\}$

$\pi(A, C) = 0.67, \pi(B, C) = High, \pi(G, A) = Low$

$P(image1 \in C^{Style}_{classic} \mid B) = High$

$P(image1 \in C^{Style}_{modern} \mid G) = Low$

If 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.

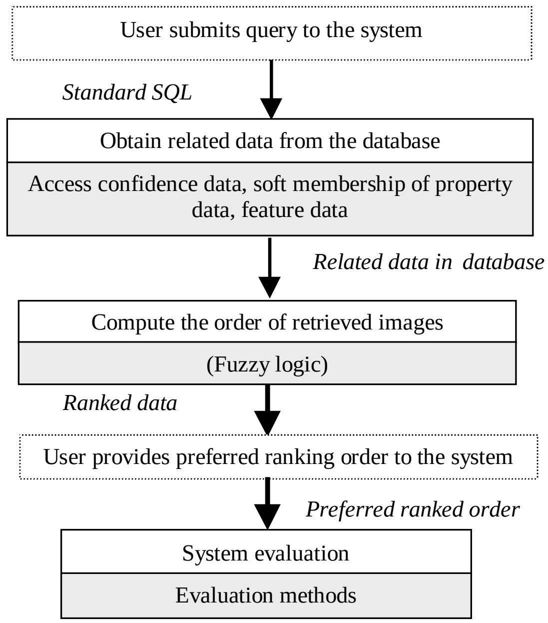


Figure 1. The Flow Diagram of the Soft Query System

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 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.:

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

Example 4.2. : For image1, $Cost(image1) = \$10000$. For image2, $Cost(image2) = \$5000$. Therefore, the membership of image1 to a class $(Cost \geq \$5000)$ is 1, the membership of image1 to a class $(Cost \leq \$5000)$ is 0, and the membership of image2 to a class $(Cost \leq \$5000)$ is 1. ∎

Subsequently, we need to compute the membership values by applying Boolean operations on atomic clauses. The Boolean operations are well-known, and hence we only illustrate the idea through the following example.

Example 4.3.: For an image 1, $Cost(image1) = \$10000$, $Size(image1) = 100 \times 100$. For image2, $Cost(image2) = \$5000$, $Size(image2) = 30 \times 30$.

Therefore, the membership of image1 to a class where $((Cost \geq \$5000) \wedge (Size > 30 \times 30))$ is $(1 \wedge 1) = 1$, the membership of image1 to a class where $((Cost \leq \$5000) \wedge (Size > 30 \times 30))$ is $(0 \wedge 1) = 0$, and the membership of image2 to a class $((Cost < \$5000) \wedge (Size > 30 \times 30))$ is : $(0 \wedge 0) = 0$ and so on. ∎

4.1.2. Property Queries

In this section, we discuss the processing of queries on the properties of images. 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 determines 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*.[2,27] We provide two alternative equations satisfying these properties below.

$$\lambda_{d,c} = \max_{e \in E}\{\pi_{u,e} \times P(d \in c \mid e)\} \tag{1}$$

$$\lambda_{d,c} = \max_{e \in E}\{\min(\pi_{u,e}, P(d \in c \mid e))\} \tag{2}$$

If the value of $\pi_{u,e}$ is a fuzzy set, the system assigns the value of $\max\{(\pi_{u,e})_{\alpha_u}\}$ as the value of $\pi_{u,e}$. Instead, if the value of $P(d \in c \mid e)$ is a fuzzy set, the system assigns the value of $\max\{P(d \in c \mid e)_{\alpha_e}\}$. (The α value for the evaluator j is denoted as α_j). If the set $P(d \in c \mid e)_{\alpha_e}$ is empty, the system assigns zero as the value of $\max\{P(d \in c \mid e)_{\alpha_e}\}$. Likewise, if the set $(\pi_{u,e})_{\alpha_u}$ is empty, the system assigns zero as the value of $\max\{(\pi_{u,e})_{\alpha_u}\}$.

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 form its "*system profile*". Below, we first list these cases and then explain two alternative methods to estimate the value from *system profile*.

- No user 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 for image d to class c in the *system profile*, then the membership value is considered to be zero.

Example 4.4.: 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 1) 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:

$\lambda_{image1,C^{style}_{classic}} = \max\{(1 \times 1)(0.3 \times 0.2)(0.67 \times 0)\} = 1$, $\lambda_{image2,C^{style}_{classic}} = 0.3$, $\lambda_{image4,C^{style}_{classic}} = 0.67$

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

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

Confidence Information

Observer	Evaluator	π(u,e)
A	C	0.67
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

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)

Figure 2. Sample data for Example 4.4

$\lambda_{image3,C^{style}_{classic}} = 0.6$ (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 4.5. : Consider the same data as in Example 4.4. 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 4.4, 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). To illustrate, consider the following example.

Example 4.6. : Consider the same parameters and results as and Example 4.5. When user A submits a query requesting all the images in class m where m is $(C^{style}_{classic} \vee C^{style}_{modern} \wedge (Cost > 5000))$, the system computes the membership value for each image on feature clauses and computes the soft membership value for each image on property clauses separately. Next, the system computes the membership value by applying fuzzy logic for combination as follows.

The membership of image1 to m is : $\min(1,1) = 1$, image2 to m is 0, image3 to m is 0.6, image4 to m is 0.67.

Only the probability of image 1 and image 4 are above the fuzzy cut of user A 0.65, therefore, class m only includes image1 and image 4 with probabilities of 100% and 67%, respectively. ∎

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 if 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 containing the results of a query and the user feedback

	I1	I2	I3	I4	I5	I6	I7	I8	I9
Query Result Q	0.92	0.89	0.82	0.77	0.73	0.71	0.69	0.64	0.5
User Feedback B	0.92	0.89	0.9	0.85	0.88	0.71	0.78	0.64	0.5

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[†], which is higher than that of $I2$. Subsequently, the system measures the degree of similarity between these two vectors as:

$$\text{Similarity Distance}(Q, B) = \frac{\sum_{i=1}^{n} Q_i \times B_i}{\sqrt{\sum_{i=n}^{n} Q_i^2 \times \sum_{i=n}^{n} B_i^2}} \tag{3}$$

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[‡] with Excalibur Image DataBlade Module[§] which is running on a 300MHZ dual processor Sun Enterprise 450 Server with SunOS 5.6.

Figure 3 depicts the conceptual schema of our experimental setup using the Entity-Relationship data model. In this figure, the ternary relationship *Soft-Membership-Data* (User, Image, Property-Class, Membership) is used for recording the membership values of images in property classes. The confidence values for users/evaluators assigned by each observer are stored in the relationship *Confidence-Data* (Observer, User, Confidence). The relationship *User* (User, Fuzzy-Cut) 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* (Fuzzy-Name, Fuzzy-Value, Possibility) for maintaining the system fuzzy information.

[†]For the real application, the user might simply drag the image and move it up in the list using a couple of mouse clicks.
[‡]http://www.informix.com/informix/resource/udo914.htm
[§]http://www.informix.com/informix/products/options/udo/datablade/dbmodule/excalibur2.htm

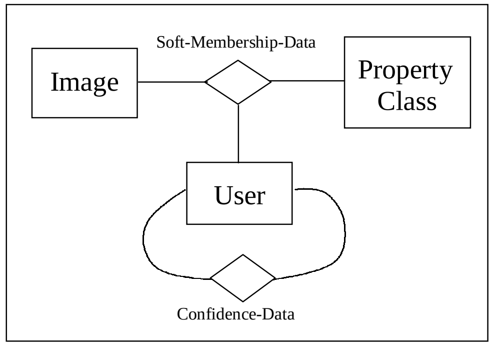


Figure 3. The experiment Entity-Relationship schema

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 assigned a random value between 0 and 1, using a uniform distribution. 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 value for x to c. Since the system profile is the only information utilized by conventional image retrieval systems, its value is computed assuming the perfect knowledge (however, simply averaged) on users classification behaviors.

Figure 4a. shows the algorithm to populate the *Confidence-Data* relation from the cube. First, we randomly select an observer and $\mathcal{P} \times \mathcal{Y}$ number of corresponding evaluators, where $\mathcal{Y}$ is the total number of users and $\mathcal{P}$ is a real number between 0 and 1, which is a system parameter denoted as *selectivity factor*. To assign confidence values for these evaluators, we compare their classification behaviors (from the cube) to the classification behavior of the observer. The closer their behaviors, the higher the confidence values. This is of course the case when the data is completely accurate. However, to introduce noise into this data, we use the Gaussian distribution shown in Figure 4c. This process repeats itself until the entire relation is populated.

Figure 4b. shows the algorithm to populate the *Soft-Membership-Data* relation from the cube. First, for a given user u, we randomly select $\mathcal{P} \times \mathcal{X} \times \mathcal{Z}$ number of cells out $\mathcal{C}(u, j, k)$, where j (a class number) varies from 1 to $\mathcal{Z}$, k (an image number) varies from 1 to $\mathcal{X}$ and $\mathcal{P}$ is the selectivity factor. Next, each randomly selected cell $\mathcal{C}(u, c, x)$ is used to determine the membership value of image x to class c by the user. Again, this is assuming a completely accurate data. To introduce noise into this data we use the Gaussian distribution shown in Figure 4c.

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 three sets of experiments to compare our soft query system with the conventional image retrieval approach (denoted hereafter as CIR). For these experiments, we investigated the impact of varying the selectivity factor, level of noise and number of users, respectively. In addition, we varied the number of images ($\mathcal{X}$) and classes ($\mathcal{Z}$) within these experiments, however, the trends remained unchanged. Hence, for the remainder of this section we fix the values at $\mathcal{X}$=50 and $\mathcal{Z}$=5. We also performed all the experiments using both Equations 1 and 2, but the differences in the results were marginal and hence we only report the results for Equation 1.

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

[¶]The variance among these runs was smaller than 1.5%.

```
Generate-Reference-Confidence(int K )
{  // K is a knob for introducing noise into our data
   For i=1 to User-Number Y do{
      generate a random number θ  (0.3 to 0.8)
      INSERT (UserI, θ) INTO USER
      Pick-Number = P * Y
      choose Pick-Number users randomly and save in Yb
      for each user u in Yb
      {
       Calculate the similarity distance χ between i and u
       χ=F(χ, K )
       Translate χ into corresponding fuzzy term
       Save the fuzzy term in Confidence-Data
      }}
}
```

a.

```
Generate-Soft-MemberShip(int K )
{// K is a knob for introducing noise into our data
   For i=1 to UserNumber do{
      Pick-Number = P * X  * Z
      choose Pick-Number cells from C randomly
        and save in Xb
      for each cell of Xb
      {
         χ is the value of the cell
         χ=F(n, K )    //n= the cell value n of M
         Translate χ into corresponding fuzzy term
         Save the fuzzy term in Soft-Membership-Data.
      }}
}
```

b.

```
F(double n,int K )
{  //F(x,k)= x  + K  *E/10  where E is a random variable who has Gaussian distribution(with zero mean and variance 1.
   //0<= K <=10
   generate a random number E within –1 to 1
   x= n + E* K /10
   if  (x > 1) x=1
   else if (x <0) x=0
   return(x)
}
```

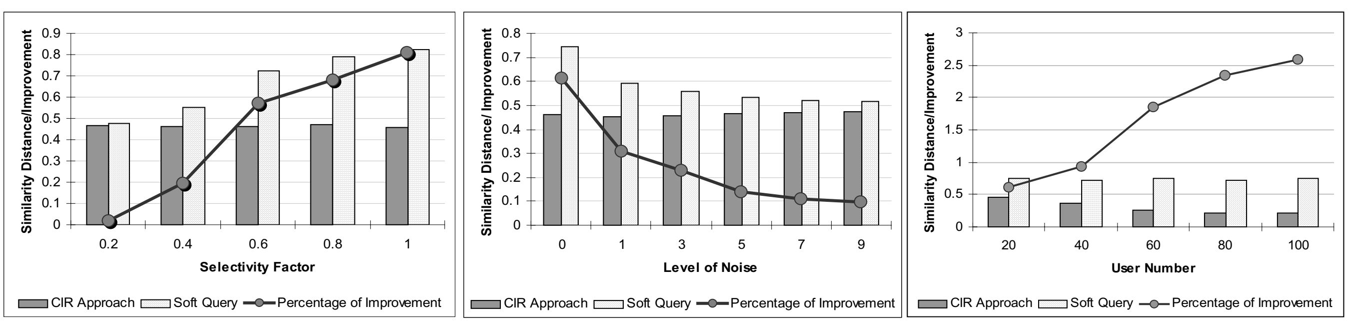
c.

Figure 4. The benchmark generating algorithm

For all the results reported in this section, the Y-axis depicts both the similarity distance computed by Equation 3 and the percentage of improvement over CIR computed as:

$$\text{Percentage of Improvement}(SQ, CIR) = \frac{SQ - CIR}{CIR} \tag{4}$$

In Figure 5a., the X-axis is the selectivity factor varied from 0.2 to 1. We fixed the number of users at $\mathcal{Y} = 20$ and introduced no noise into the data. The results show 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 due to 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.



a) Y=20, Noise Level = 0   b) Y=20, P = 0.6   c) P = 0.6, Noise Level =0

Figure 5. Impacts of Selectivity Factor, Noise and Number of Users

The results of our experiments to investigate the impact of noise are summarized in Figure 5b. We fixed the number of users at $\mathcal{Y} = 20$ and the selectivity factor at $\mathcal{P} = 0.6$. The X-axis of this figure is the noise level varied from 0 to 9 (see Figure 4c.). As expected, soft query suffers from higher noise levels, while it always outperformed CIR by at least 10%. The reason is that soft query uses Equation 1 as opposed to a naive averaging performed by the system profile. Therefore, it cancels out the impact of noisy data by considering the entire population.

Figure 5c. depicts the impact of varying the number of users (from 20 up to 100 on the X-axis). Although the performance of soft query remained unchanged as the number of users grows, the performance of CIR dropped. This is because soft query only takes into consideration the useful information per user, thus the accuracy remains independent of the number of users, while the CIR approach considers the average information which is dependent to the number of users.

6. CONCLUSION

We proposed a unified model to support soft querying and classification on image databases. Towards this end, we formally described our model and demonstrated 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 that no profile is available for a user, the model reduces itself to the conventional image retrieval systems and only consults system provided algorithms and profile.

We intend to extend this study in three ways. First, we plan to populate our database with real images and utilizing a real application, compare our approach with conventional systems. The results reported in this paper are at the preliminary stages. Second, we intend to investigate alternative clustering and indexing techniques to reduce the complexity and storage requirements of our approach. Currently, the complexity of processing a soft query is $O(\mathcal{X} \times \mathcal{Y})$, and the complexity of storage requirement for the profiles is $O(\mathcal{Y}^2 + \mathcal{X}\mathcal{Y}\mathcal{Z})$. Finally, we want to extend the system to utilize the users' relevance feedback to improve the profiles, iteratively.

REFERENCES

1. J. Assfalg, A. D. 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*, vol. 1, 1999.
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*, pp. 216–226, 1996.
4. C. Faloutsos and et al., "Efficient and effective querying by image content," *J. of Intelligent Information System* **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*, vol. 1, 1999.
6. W. Hu, M. Schmalz, and G. Ritter, "Image retrieval using the longest approximate common subsequences," in *Proceedings of the IEEE International Conference on Multimedia Computing and Systems*, vol. 2, 1999.
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*, pp. 654–656, 1992.
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*, vol. 2, 1999.
9. R. McConnell, R. Kwok, J. Curlander, W. Kober, and S. Pang, "ψ-s correlation and dynamic time warping: Two methods for tracking ice floes in sar images," *IEEE Trans. Geoscience and Remote Scensing* **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*, vol. 2, 1999.
11. S. Nepal, M. Ramakrishna, and J. Thom, "A fuzzy object query language (foql) for image databases," in *Proceedings of the Sixth International Conference on Database Systems for Advanced Applications*, pp. 117–124, April 1999.

12. S. Nepal and M. Ramakrishna, "Query processing issues in image(multimedia) databases," in *Proceeding of 15th International Conference on Data Engineering (ICDE-99)*, pp. 22–29, 1999.
13. S. Nepal and M. Ramakrishna, "Query processing issues in image(multimedia) databases," in *21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR98)*, p. 386, 1998.
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 *SPIE Conference on Storage and Retrieval for Image and Vedio Databases*, vol. 1908, pp. 173–187, 1993.
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* .
16. M. Stricker and M. Orengo, "Similarity of color images," in *SPIE Conference on Storage and Retrieval for Image and Vedio Databases III*, vol. 2420, pp. 381–392, 1995.
17. C. Brambilla, A. D. 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*, vol. 1, 1999.
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*, vol. 1, 1999.
19. A. Doulamis, Y. Avrithis, N. Doulamis, and S. 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*, vol. 2, 1999.
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*, vol. 1, 1999.
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*, vol. 2, 1999.
24. Y. Rui, T. 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. R. Koenen, "Mpeg-7: Context, objectives and technical roadmap, v.12," *ISO/IEC JTC1/SC29/WG11/ N2861* , 1999.
27. L. Zadeh, "Fuzzy sets as a basis for a theory of possibility," *Fuzzy Sets and Systems 1(1)* , pp. 3–28, 1978.
28. W. Hill, L. Stead, M. Rosentien, and G. Furnas, "Recommending and evaluationg choices in virtual community of user," in *Procedding of the CHI95 Conference*, 1995.
29. U. Shardanand and P. Maes, "Social information filtering: Algorithm for automating "word of mouth"," in *Procedding of the CHI95 Conference*, 1995.
30. P. Resnick, N. Iacovou, M. Suchak, P. Bergstorm, and J. Riedl, "Grouplens: An open architecture for collaborative filtering of netnews," in *Proceeding of ACM 1994 Conference on Computer Supported Cooperative Work*, pp. 175–186, 1994.