On Similarity Measures for Multimedia Database Applications

K. Selçuk Candan* Wen-Syan Li

C&C Research Laboratories, NEC USA, Inc.
110 Rio Robles, M/S SJ100
San Jose, CA 95134 USA
Email: {candan,wen}@ccrl.sj.nec.com
Tel:(408)943-3008 Fax:(408)943-3099

Abstract

A multimedia database query consists of a set of fuzzy and boolean (or crisp) predicates, constants, variables, and conjunction, disjunction, and negation operators. The fuzzy predicates are evaluated based on different media criteria, such as color, shape, layout, keyword. Since media-based evaluation yields similarity values, results to such a query is defined as an ordered set. Since many multimedia applications require partial matches, query results also include tuples which do not satisfy all predicates. Hence, any fuzzy semantics which extends the boolean semantics of conjunction in a straight forward manner may not be desirable for multimedia databases. In this paper, we focus on the problem of "*given a multimedia query which consists of multiple fuzzy and crisp predicates, how to provide the user with a meaningful overall ranking.*" More specifically, we study the problem of merging similarity values in queries with multiple fuzzy predicates. We describe the essential multimedia retrieval semantics, compare these with the known approaches, and propose a semantics which captures the retrieval requirements in multimedia databases.

Keywords: Fuzzy logic, partial match, multimedia databases, scoring functions, image matching.

1 Introduction

Multimedia data includes image and video data which are complex in terms of visual and semantic contents. Depending on the application, multimedia objects are modeled and indexed using their **(1)** visual properties (or a set of relevant visual features), **(2)** semantic properties, and/or **(3)** the spatial/temporal relationships of sub-objects. Consequently, retrieval in multimedia databases is inherently fuzzy. This fuzziness can be explained in various ways:

- *similarity of media features*, such as correlation between color (red vs. orange) or shape (circle vs. ellipse) features,
- *imperfections in the feature extraction algorithms,* such as the high error rate in motion estimation due to the multitude of factors involved, including camera and object speed, and camera effects,
- *imperfections in the query formulation methods,* such as the Query by Example (QBE) method where user provides an example but is not aware of which features will be used for retrieval,

*This work was performed when the author visited NEC USA, CCRL. The author's current address is Computer Science and Engineering Department College of Engineering and Applied Sciences Arizona State University Box 875406 Tempe, AZ 85287-5406, Email: candan@asu.edu, Tel: (602) 965-2770, Fax: (602) 965-2751.

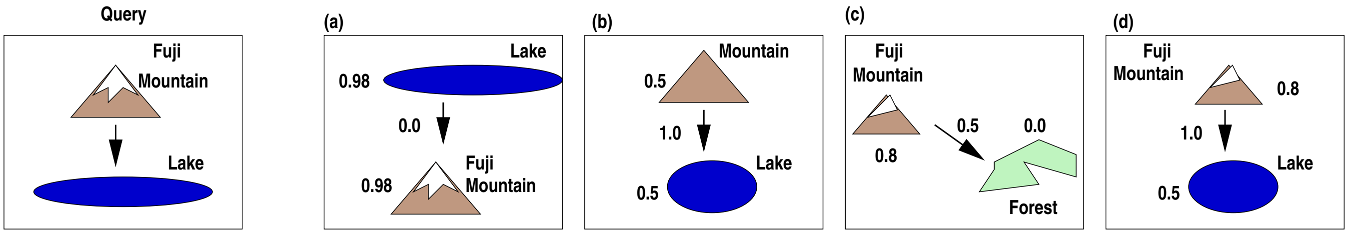


Figure 1: Query image and candidates of partial matches

- *partial match requirements,* where objects in the database fail to satisfy all requirements in the query, and
- *imperfections in the available index structures,* such as low precision or recall rates due to the imperfections in clustering algorithms.

In Section 2, we discuss these reasons in greater details. In many multimedia applications, more than one of these reasons coexist and, consequently, the system must take each of them into consideration. This requires quantification of different sources of fuzziness and merging into a single combined value for user's reference. The following example describes this requirement.

Example 1.1 A query for retrieving images containing *Fuji Mountain* and a lake in the foreground can be specified with an SQL3 like query statement on an object-based multimedia database system [1, 2, 3] as follows:

select image P, *object* object1, *object* object2
where P *contains* object1
and P *contains* object2
and object1.semantical_property *s_like* "mountain"
and object1.image_property *image_match* "Fuji_mountain.gif"
and object2.semantical_property *is* "lake"
and object2.image_property *image_match* "lake_image_sample.gif"
and object1.position *is_above* object2.position

The above query contains two crisp query predicates: *contains* and *is*[1]. It also contains a set of fuzzy query predicates:

- *s_like* (i.e. semantically similar) which evaluates the degree of semantic similarity between two terms. Helps resolving correlations between semantic features, imperfections in the semantics extraction algorithms, in the index structures, and in the user queries;
- *image_match* (i.e. visually like) which evaluates the visual similarity between two images. This helps resolving correlations between visual features and imperfections in the index structures; and
- *is_above* (a spatial condition which compares the spatial position between two objects). Helps resolving correlations between spatial features, imperfections in the spatial information extraction algorithms, imperfections in the index structures, and imperfections in the user queries.

This query returns a set of 3-tuples of the form $\langle P, object1, object2\rangle$ that satisfy all crisp conditions and that has a combined fuzzy score above a given threshold. Figure 1(Query) shows the conceptual representation of the above query. Figures 1(a), (b), (c), and (d) shows examples of candidate images that may match this query. The numbers next to the objects in these candidate images denote the

[1]The keyword, *where*, used in the query does not correspond to a predicate.

similarity values for the object level matching. As explained earlier, in this example, the comparisons on spatial relationships are also fuzzy to account for the correlations between spatial features.

The candidate image in Figure 1(a) satisfies object matching conditions but its layout does not match user specification. Figures 1(b) and (d) satisfy the image layout condition but objects do not perfectly match the specification. Figure 1(c) has structural and object matching with low scores. Note that in Figure 1(a), the spatial predicate, and in Figure 1(c), the image similarity predicate for lake completely fail (i.e., the match is 0.0).

As shown in the above example, most types of fuzziness can be captured using fuzzy predicates. The partial match requirements, on the other hand, have to be handled by the query processing algorithm at the time of merging fuzzy results. The need for partial matches was observed in [4] which proposes a weighting based technique to deal with subgoals with low matches. [4, 5, 6] discuss various ways to handle user-defined weights in multimedia queries. In this approach, if one of the features is not important, that feature is given a relatively low weight so that its effect on the outcome is minimized. Thus, the weighting technique can be used to eliminate such unwanted features. This assumption, however, assumes prior knowledge about the feature distribution of the data as well as user expertise.

Example 1.2 Let us assume that a user wants to find all images in the database that are similar to image $I_{example}$. Let us also assume that the database uses three features, *color*, *shape*, and *edgedistribution* to compare images, and that the database contains three images, I_1, I_2, and I_3. Finally let us assume that the following table gives the matching degrees of the images in the database for each feature:

Image	Shape	Color	Edge
I_1	**0.0**	0.9	0.8
I_2	0.8	**0.0**	0.9
I_3	0.9	0.8	**0.0**

According to this table, it is clear that if the user does not specify a priority among the three features, the system should treat all three candidates equally. On the other hand, since for each of the three features, there is a different image which fails it completely, even if we have prior knowledge regarding the feature distribution of the data, we can not use feature weighting to eliminate low scoring features.

In this paper, we focus on the problem of "*given a query which consists of multiple fuzzy and crisp predicates, how to provide a meaningful final ranking to the users.*" We propose an alternative scoring approach which captures the multimedia semantics well and does not face the above problem while handling partial matches. Although it is not based on weighting, the proposed approach can be used along with weighting strategies, if weighting is requested by the user.

The paper is structured as follows: In Section 2, we provide an overview of the multimedia retrieval semantics. We show the similarities between multimedia queries and fuzzy logic statements. Then, in Section 3, we provide an overview of the popular fuzzy logic semantics and discuss why they are not suitable for multimedia retrieval. In Section 4, we propose a fuzzy logic semantics which captures the essential requirements of multimedia retrieval problem. In Section 5, we compare the statistical properties of the popular fuzzy logic semantics and the proposed semantics and in Section 7 we provide our conclusions.

2 Multimedia Retrieval Semantics

In this section, we first provide an overview of the multimedia retrieval semantics; i.e., we describe what we mean by retrieval of multimedia data. We then review some of the approaches to deal with multimedia queries that involve multiple fuzzy predicates.

2.1 Fuzziness in Multimedia Retrieval

We now give examples to underscore the reasons for fuzziness in the retrieval of multimedia data. In addition, we define the different types of fuzziness that may arise in multimedia applications.

Similarity of features. In similarity based retrieval, there usually is a distance function which is used to quantify similarity between two feature instances. For example, in video retrieval, given two object motion vectors, v_1 and v_2, if the user is interested in the orientation of the objects, similarity between these two vectors can be calculated by using the cosine of the angle between v_1 and v_2.

Imperfections in the feature extraction. This imperfection is mostly due to the resource limitations of the feature extraction algorithms. For instance, in the case of object motion estimation in video sequences, such imperfections may be due to the parameters unknown at the motion extraction time, such as the camera motion, or they may be due to low numeric precision of motion vectors due to compression requirements. In either case, the system must quantify and use the estimated error while retrieving objects.

Imperfections in the query formulation. Most query formulation tools, such as query-by-example (QBE), do not have mechanisms to allow users to describe their queries precisely [2]. In most QBE systems, the features for retrieval and their importance are estimated by the system. Even in the systems where such information can be provided by the user, users can not always express, unambiguously, what they are looking for.

Partial matches. An object- and spatial information-based image retrieval query may use two features, object shapes and spatial positions, to retrieve relevant images. If, in the database, there are images which contain objects with the required shapes and other images which contain the required spatial relationships, but if there are no images which satisfy both requirements, the matches returned by the database can not be exact. In other words, given a query, Q, in some cases, users may be willing to accept (of course, with a lower satisfaction) if an object does not satisfy Q (for example, if one of the predicates in a conjunctive query evaluates to `false`) but another query $Q' \subset Q$.

Imperfections in the index structures. Due to the existence of similarity (or distance) functions discussed earlier, indices used in multimedia retrieve feature vectors based on their distances from a given input point. In order to increase the speed of this process, automatic clustering or filtering techniques [8] are usually utilized. As shown in Figure 2, these techniques may result in *false dismissals*, i.e., incorrectly omitted results, and *false positives*, i.e. incorrectly accepted results. If a predicate is implemented using an index structure with a high ratio of false dismissals or positives, then the system must quantify and use the estimated error rate as confidence value for the retrieved objects.

It is possible to re-classify the above fuzziness into three categories: *precision-related*, *recall-related*,

[2]QBE is originally developed for relational databases [7]. Here, we are referring to a similar development in media databases, where instead of providing a explicit textual query, users provide an example media object that describe the features of the object they are looking for implicitly through the concept of similarity.

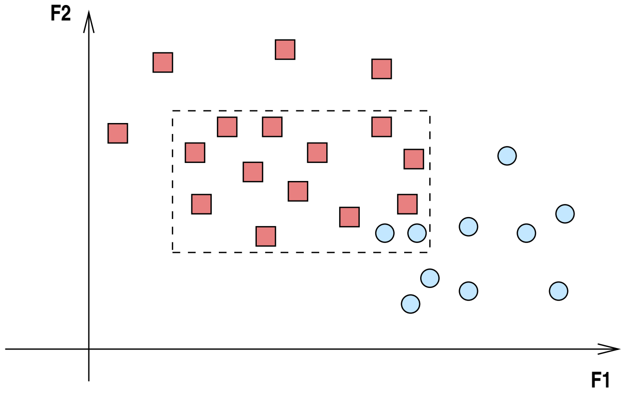


Figure 2: Clustering error which results in imperfections in the index structures: the squares denote the matching objects, circles denote the non-matching objects, and the dashed rectangle denotes the cluster used by the index for efficient storage and retrieval

and *partiality-related* fuzziness. Precision related class captures fuzziness due to similarity of features, imperfections in the feature extraction algorithms, imperfections in the query formulation methods, and the precision rate of the utilized index structures. Recall- and partiality-related classes are self explanatory.

Note that, in information retrieval, the precision/recall values are mainly used in evaluating the effectiveness of a given retrieval operation or the effectiveness of a given index structure. Here, we are using these terms more as statistics which can be utilized to estimate the quality of query results. We have used this approach, in the SEMCOG image retrieval system [2, 1, 3], to provide pre- and post-query feedback to users. Two examples given below show why precision and recall are important in processing multimedia queries.

Example 2.1 (Handling precision related fuzziness) Let us assume that we are given a query of the form

$$Q(X) \leftarrow s\_like(man, X.semantic\_property) \wedge image\_match(X.image\_property, \text{“}a.gif''). $$

Let object I have $semantic\_property$ $=woman$ and $image\_property$ $=im.gif$. Let us assume that the semantic precision of $s\_like(man, woman)$ is 0.8 and the image matching precision $image\_match$ (“$im.gif''$, “$a.gif''$) is 0.6. This means that the index structure and semantic clusters used to implement the predicate s_like guarantee that 80% of the returned results are semantically similar to *man*. These semantic similarities can be evaluated using various algorithms [9, 10]. Similarly, the predicate $image\_match$ guarantees that 60% of the returned results are visually similar to $''a.gif''$. Then, assuming that the two predicates are not correlated, $Q(I)$ should be $0.8 \times 0.6 = 0.48$: By replacing X in $s\_like(man, X)$ with *woman*, we maintain 80% precision; then, by replacing the X in $image\_match(X, \text{“}a.gif'')$ with I, we maintain 60% of the remaining precision. The final precision (or confidence) is 0.48.

Similarly, given a query execution plan and assuming that the predicates are independent, the recall rate can also be found by multiplying the recall rates of the predicates.

2.2 Dealing with Imperfections and Similarities

Traditional query languages are based on boolean logic, where a predicate is treated as a propositional function, that returns one of the two values: *true* or *false*. However, due to the stated imperfections, predicates related to visual or semantic features do not correspond to propositional functions, but to functions which return values between 0.0 and 1.0. Consequently, a *query* is of the form

$$Q(Y_1, \ldots, Y_n) \leftarrow \Theta(p_1(Y_1, \ldots, Y_n), \ldots, p_m(Y_1, \ldots, Y_n)),$$

where p_is are fuzzy or crisp predicates, Θ is a logic formula, and Y_j are free variables. The solution, on the other hand, is defined as an ordered set S_Q of n-tuples of the form $\langle X_1, X_2, \ldots, X_n \rangle$, where (1) n is the number of variables in query Q, (2) each X_i corresponds to a variable Y_i in Q, and (3) each X_i satisfies the type constraints of the corresponding predicates in Q. The order of the set S denotes the relevance ranking of the solutions.

Note that fuzzy sets and the corresponding fuzzy predicates have scoring or membership functions similar to those of the multimedia predicates. Consequently, we will examine the use of fuzzy logic for multimedia retrieval.

3 Overview of Fuzzy Logic

In this section we first provide an overview of fuzzy logic and, then, we introduce properties of different fuzzy logic operators. A fuzzy set, F, with domain D can be defined using a membership function, $\mu_F : D \rightarrow [0, 1]$. A *crisp* (or conventional) set, C, on the other hand, has a membership function of the form $\mu_C : D \rightarrow \{0, 1\}$. When for an element $d \in D$, $\mu_C(d) = 1$, we say that d is in C ($d \in C$), otherwise we say that d is not in D ($d \notin C$). Note that a crisp set is a special case of a fuzzy set.

A fuzzy predicate is defined as a predicate which corresponds to a fuzzy set. Instead of returning *true*(1) or *false*(0) values for propositional functions (or conventional predicates which correspond to crisp sets), fuzzy predicates return the corresponding membership values.

3.1 Fuzzy Logic Operator Semantics and their Properties

There are a multitude of functions [11, 12, 13], each useful in a different application domain, proposed as semantics for fuzzy logic operators ($\wedge, \vee, \neg$). In this section, we describe the popular scoring functions, discuss their properties, and show why these semantics may not be suitable for multimedia retrieval.

Two of the most popular scoring functions are the *min* and *products* semantics of fuzzy logical operators. We can state these two semantics in the form of a table as follows: Given a set, $P = \{P_1, \ldots, P_m\}$ of fuzzy sets and $F = \{\mu_1(x), \ldots, \mu_m(x)\}$ of corresponding membership functions, Table 1 shows the *min* and *products* semantics.

Triangular-norms. These two semantics (along with some others) have the following properties: Defined as such, binary conjunction and disjunction operators are triangular-norms (t-norms) and triangular-conorms (t-conorms). Table 2 shows the properties of t-norm and t-conorm functions. Intuitively, t-norm functions reflect the properties of the crisp conjunction operation and t-conorm functions reflect those of the crisp disjunction operation. *Although the property of capturing crisp semantics is desirable in many cases, for multimedia applications, this is not always true. For instance, the partial match requirements invalidate the boundary conditions as shown in the following example.*

Min semantics	*Product* semantics
$\mu_{P_i \wedge P_j}(x) = min\{\mu_i(x), \mu_j(x)\}$	$\mu_{P_i \wedge P_j}(x) = \frac{\mu_i(x) \times \mu_j(x)}{max\{\mu_i(x), \mu_j(x), \alpha\}} \quad \alpha \in [0,1]$
$\mu_{P_i \vee P_j}(x) = max\{\mu_i(x), \mu_j(x)\}$	$\mu_{P_i \vee P_j}(x) = \frac{\mu_i(x) + \mu_j(x) - \mu_i(x) \times \mu_j(x) - min\{\mu_i(x), \mu_j(x), 1-\alpha\}}{max\{1-\mu_i(x), 1-\mu_j(x), \alpha\}}$
$\mu_{\neg P_i}(x) = 1 - \mu_i(x)$	$\mu_{\neg P_i}(x) = 1 - \mu_i(x)$

Table 1: *Min* and *products* semantics for fuzzy logical operators

	T-norm binary function N (for $\wedge$)	T-conorm binary function C (for $\vee$)
Boundary conditions	$N(0,0) = 0$, $N(x,1) = N(1,x) = x$	$C(1,1) = 1$, $C(x,0) = C(0,x) = x$
Commutativity	$N(x,y) = N(y,x)$	$C(x,y) = C(y,x)$
Monotonicity	$x \leq x', y \leq y' \rightarrow N(x,y) \leq N(x',y')$	$x \leq x', y \leq y' \rightarrow C(x,y) \leq C(x',y')$
Associativity	$N(x, N(y,z)) \leq N(N(x,y),z)$	$C(x, C(y,z)) \leq C(C(x,y),z)$

Table 2: Properties of triangular-norm and triangular-conorm functions

Example 3.1 Consider the query and candidate images given in Example 1.1, Figure 1. If the boundary conditions given in Table 2 are preserved when merging the scores, Figures 1(a) and (c) will be dropped from consideration as their final score will be 0.0.

In addition, monotonicity is too weak of a condition for multimedia applications. An increase in the score of a single query criterion should increase the combined score; whereas the monotonicity condition dictates such a combined increase only if the scores for all of the query criteria increases simultaneously. A stronger condition ($N(x,y)$ increases even if only x or only y increases) is called strictly increasing property[3]. Clearly, $min(x,y)$ is not strictly increasing.

Example 3.2 If we consider the query and candidate images given in Figure 1, it is clear that the candidate image shown in Figure 1(d) is a better match than a candidate image shown in Figure 1(b). However, monotonicity does not differentiate between these two images.

Logical equivalence. In general, two desirable properties for fuzzy conjunction and disjunction operators is distributivity and idempotency. These enable preservation of the query semantics after query rewrite operations used for optimization purposes. The *min* semantics is known [15, 14, 16] to be the only semantics for conjunction and disjunction that preserves logical equivalence (in the absence of negation) and is monotone at the same time. This property of the *min* semantics makes it the preferred fuzzy semantics for most cases. Furthermore, in addition to satisfying the properties of being t-norm and t-conorm, the *min* semantics also has the property of being idempotent.

Archimedean property. The product semantics [12], on the other hand, satisfies idempotency only if $\alpha = 0$. On the other hand, when $\alpha = 1$, it has the property of being strictly increasing (when x or y is different from 1) and Archimedean ($N(x,x) < x$ and $C(x,x) > x$ foe $x < 1.0$). The Archimedean property is weaker than the idempotency, yet it provides an upper bound on the combined score, allowing for optimizations.

One advantage of Archimedean functions is that they provide a natural way to implement emphasis (which is essential in implementing emphasis adjectives, such as *very*): replication of subgoals. Note that the use of replication to put emphasis is nothing more than a syntactic operation, which does not require the whole predicate to be reevaluated.

[3]This is not the same definition of strictness used in [14].

$\mu_{P_i \wedge \ldots \wedge P_j}(x)$	$\mu_{\neg P_i}(x)$	$\mu_{P_i \vee \ldots \vee P_j}(x)$
$\frac{\mu_i(x)+\ldots+\mu_j(x)}{\lvert\{P_i,\ldots,P_j\}\rvert}$	$1-\mu_i(x)$	$\frac{\lvert\{P_i,\ldots,P_j\}\rvert-\mu_i(x)+\ldots+\mu_j(x)}{\lvert\{P_i,\ldots,P_j\}\rvert}$

Table 3: Arithmetic average semantics for fuzzy logical operators

Example 3.3 Using an Archimedean conjunction function, one can ask for images that are *very similar to "a.gif"* as follows:

$$Q(X) \leftarrow (X.image\_property\ image\_match\ \text{“}a.gif'')\ \ and\ (X.image\_property\ image\_match\ \text{“}a.gif'').$$

Note that the two predicates in the above query are the same. If the conjunction is Archimedean, images must have a higher visual match with "a.gif" to achieve a given score ($min(x, y)$ is not Archimedean).

An alternative to the replication of subgoals would be to assign appropriate weights to the predicates. The advantage of replication to weighting is that replication enables the emphasis to adapt dynamically as the query is modified. To achieve the same affect with weighting, one has to modify the weights when a query is modified. The advantage of weighting, however, is that it can be arbitrarily set by the user, whereas the emphasis achieved by replication is not arbitrary. Furthermore, replication makes certain query rewrite optimizations impossible while making other optimizations, such as runtime caching of non-variant function rewrite necessary. We have investigated performance of intelligent caching and query rewrite operations in [17].

N-ary operators. In information retrieval research (which also shows the characteristics of multimedia applications), other fuzzy semantics, including the arithmetic mean [18] are suggested. The arithmetic mean semantics (Table 3) provides an n-ary scoring function ($|\{P_i, \ldots, P_j\}| = n$). Note that the binary version of arithmetic mean does not satisfy the requirements of being a t-norm: it does not satisfy boundary conditions and it is not associative. Hence, it does not subsume crisp semantics. On the other hand, it is idempotent and strictly increasing.

Arithmetic average semantics emulate the behavior of the *dot product* based similarity calculation popular, in information retrieval: effectively, each predicate is treated like an independent dimension in an n-dimensional space (where n is the number of predicates), and the merged score is defined as the dot-product distance between the complete truth, $\langle 1, 1, \ldots, 1\rangle$, and the given values of the predicates, $\langle \mu_1(x), \ldots, \mu_n(x)\rangle$. Note that, although this approach is shown to be suitable for many information retrieval applications, as we will show in the next section, it does not capture the semantics of multimedia retrieval applications introduced in Section 2.

3.2 Summary

Due to the discussed limitations of

- subsumption of crisp semantics and
- being non-strict increasing

the *min* semantics is not suitable for multimedia applications. The fact that *min* semantics is not Archimedean may also reduce its benefits in certain applications. As discussed in Example 1.1, according to the *min* semantics, the score of the candidate images given in Figures 1(a) and (c) would be 0.0 although they partially match the query. Furthermore, scores of the images in Figure 1(b) and (d) would both be 0.5, although Figure 1(d) intuitively has a higher score.

In the next section, we explain why it should not be the choice for multimedia applications either.

$\mu_{P_i \wedge P_j}$(x)	$\mu_{\neg P_i}(x)$	$\mu_{P_i \vee P_j}(x)$
$\mu_i(x) \times \mu_j(x)$	$1 - \mu_i(x)$	$\mu_i(x) + \mu_j(x) - \mu_i(x) \times \mu_j(x)$

Table 4: The product semantics for multimedia retrieval

4 Fuzzy Logic Operator Semantics for Multimedia Retrieval

The most important parameters for multimedia retrieval are the precision and recall rate of the indexes or the predicates used and the confidence value associated with results (Section 2.1). We have seen that these two parameters are of multiplicative nature. We have also seen that the non-strictness of the *min* semantics, makes it non-suitable for many applications. Consequently, if we need to choose between the *min* and *product* semantics, the product semantics is more suitable for multimedia retrieval. In this section, we look at the product semantics as a candidate for multimedia applications, we use the requirements of multimedia applications as discussed in earlier sections to evaluate product semantics, and we discuss how to improve it whenever it does not provide required functionalities.

4.1 Product Semantics for Multimedia Retrieval

Given a set, $P = \{P_1, \ldots, P_m\}$ of fuzzy sets and $F = \{\mu_1(x), \ldots, \mu_m(x)\}$ of corresponding scoring functions, the product semantics of fuzzy binary logical operators in multimedia retrieval are given in Table 4.

Since this semantics corresponds to the product semantics given in Table 1 (α set to 1), it does not provide idempotency[4]. Furthermore, this semantics does not have distributivity, which means that, teh following equalities hold:

- $[\mu_{(P_1 \wedge P_2) \vee (P_1 \wedge P_3)}(x)] = [\mu_{P_1 \wedge (P_2 \vee P_3)}(x)]$, and
- $[\mu_{(P_1 \vee P_2) \wedge (P_1 \vee P_3)}(x)] = [\mu_{P_1 \vee (P_2 \wedge P_3)}(x)]$.

These equations are generally desirable, because they enable a query optimizer to rewrite queries in a way to minimize the execution cost. Unfortunately, this is not always possible for product semantics. Below, we show that distributivity does not hold:

Conjunction does not distribute over disjunction: We can calculate the values of the two sides of the equation as follows:

$$\begin{aligned}
\mu_{(P_1 \wedge P_2) \vee (P_1 \wedge P_3)}(x) &= 1 - (1 - (\mu_{P_1} \times \mu_{P_2})) \times (1 - (\mu_{P_1} \times \mu_{P_3})) \\
&= (\mu_{P_1} \times \mu_{P_2}) + (\mu_{P_1} \times \mu_{P_3}) - (\mu_{P_1} \times \mu_{P_2}) \times (\mu_{P_1} \times \mu_{P_3}) \\
\mu_{P_1 \wedge (P_2 \vee P_3)}(x) &= \mu_{P_1} \times (1 - ((1 - \mu_{P_2}) \times (1 - \mu_{P_3})))
\end{aligned}$$

Note that these two terms do not necessarily evaluate to the same value. In fact, the ratio of difference of these two terms, can be calculated as

$$\frac{[\mu_{(P_1 \wedge P_2) \vee (P_1 \wedge P_3)}(x)] - [\mu_{P_1 \wedge (P_2 \vee P_3)}(x)]}{\mu_{P_1 \wedge (P_2 \vee P_3)}(x)} = \frac{1 - \mu_{P_1}(x)}{\frac{1}{\mu_{P_2}(x)} + \frac{1}{\mu_{P_3}(x)} - 1}.$$

Consequently, the conjunction would distribute on disjunction only in some special cases (when $\mu_{P_1}(x) = 0$ or when the error ratio is 0); i.e., $\mu_{P_1 \wedge (P_2 \vee P_3)}(x) \cong \mu_{(P_1 \wedge P_2) \vee (P_1 \wedge P_3)}(x)$ if

[4] The score for $P_1(x) \wedge P_1(x)$ calculated as such is $(\mu_1(x))^2$, not $\mu_1(x)$ as required by idempotency.

$\mu_{P_i \wedge P_j}(x)$	$\mu_{\neg P_i}(x)$	$\mu_{P_i \vee P_j}(x)$
$(\mu_i(x) \times \mu_j(x))^{\frac{1}{2}}$	$1 - \mu_i(x)$	$1 - ((1 - \mu_i(x)) \times (1 - \mu_j(x))^{\frac{1}{2}})$

Table 5: The modified product semantics for multimedia retrieval

- $\mu_{P_1}(x) \cong 0, \quad \mu_{P_1}(x) \cong 1, \quad \mu_{P_2}(x) \cong 0, \quad$ or $\mu_{P_3}(x) \cong 1$.

Note that, even though distributivity does not hold in general, we can s till distribute conjunction over disjunction when approximate results are accepted. Furthermore, after using distributivity to optimize a query, the distance ratio given above can be used for correcting the error caused by the rewrite operation.

Disjunction does not distribute over conjunction: Similarly, we can see that disjunction also does not distribute on conjunction:

$$\begin{array}{ll} \mu_{(P_1 \vee P_2) \wedge (P_1 \vee P_3)}(x) & = (1 - ((1 - \mu_{P_1}) \times (1 - \mu_{P_2}))) \times (1 - ((1 - \mu_{P_1}) \times (1 - \mu_{P_3}))) \\ \mu_{P_1 \vee (P_2 \wedge P_3)}(x) & = 1 - ((1 - \mu_{P_1}) \times (1 - (\mu_{P_2} \times \mu_{P_3}))) \end{array}$$

The ratio of difference of these two terms, then, can be calculated as

$$\frac{[\mu_{(P_1 \vee P_2) \wedge (P_1 \vee P_3)}(x)] - [\mu_{P_1 \vee (P_2 \wedge P_3)}(x)]}{\mu_{P_1 \vee (P_2 \wedge P_3)}(x)} = \frac{-\mu_{P_1}(x) \times (1 - \mu_{P_1}(x)) \times (1 - \mu_{P_2}(x)) \times (1 - \mu_{P_3}(x))}{1 - ((1 - \mu_{P_1}) \times (1 - (\mu_{P_2} \times \mu_{P_3})))}.$$

Consequently, disjunction distributed on conjunction ($\mu_{P_1 \vee (P_2 \wedge P_3)}(x) \cong \mu_{(P_1 \vee P_2) \wedge (P_1 \vee P_3)}(x)$) when

- $\mu_{P_1}(x) \cong 0, \quad \mu_{P_1}(x) \cong 1, \quad \mu_{P_2}(x) \cong 1, \quad$ or $\mu_{P_3}(x) \cong 1$.

On the other hand, this semantics obeys DeMorgan's laws and the disjunction and conjunction operators are strictly increasing and Archimedean.

4.2 Achieving Idempotency

If idempotency is required by the application, we can give up the Archimedean property, and modify the operator semantics as shown in Table 5. The resulting semantics preserves the order of the product semantics. Furthermore, it satisfies idempotency, De Morgan's laws, and they are strictly increasing.

Note that although the order of the results that the product semantics and the modified product semantics provide are the same, the modified product semantics (or the geometric averaging) have certain desirable statistical and semantical properties. Idempotency is one of these. The statistical properties of geometric averaging will be discussed in the following sections.

One short coming of the modified product semantics, however, is that after the modification, conjunction and disjunction functions are not t-norm and t-conorm any more: they still satisfy the

- boundary conditions (which are actually undesirable due to partial match requirement),
- commutativity requirement, and
- monotone increasing requirement,

whereas, they do not have associativity (unless $\mu_1(x) \cong \mu_3(x)$) :

$$\mu_{P_1(x) \wedge (P_2(x) \wedge P_3(x))} = \sqrt{\mu_1(x) \times \sqrt{\mu_2(x) \times \mu_3(x)}}$$

$\mu_{P_{i1} \wedge \ldots \wedge P_{in}}(x)$	$\mu_{\neg P_i}(x)$	$\mu_{P_{i1} \vee \ldots \vee P_{in}}(x)$
$(\mu_{i1}(x) \times \ldots \times \mu_{in}(x))^{\frac{1}{n}}$	$1 - \mu_{i1}(x)$	$1 - ((1-\mu_{i1}(x)) \times \ldots \times (1-\mu_{in}(x)))^{\frac{1}{n}}$

Table 6: The modified n-ary product semantics for multimedia retrieval

$$\mu_{(P_1(x) \wedge (P_2(x)) \wedge P_3(x)} = \sqrt{\sqrt{\mu_1(x) \times \mu_2(x)} \times \mu_3(x)}$$

$$\mu_{P_1(x) \wedge (P_2(x) \wedge P_3(x))} - \mu_{(P_1(x) \wedge P_2(x)) \wedge P_3(x)} = (\mu_1(x) \times \mu_2(x) \times \mu_3(x))^{\frac{1}{4}} \times (\mu_1(x)^{\frac{1}{4}} - \mu_3(x)^{\frac{1}{4}}).$$

$$\frac{\mu_{P_1(x) \wedge (P_2(x) \wedge P_3(x))} - \mu_{(P_1(x) \wedge P_2(x)) \wedge P_3(x))}}{\mu_{P_1(x) \wedge (P_2(x) \wedge P_3(x))}} = 1 - (\frac{\mu_3(x)}{\mu_1(x)})^{\frac{1}{4}}.$$

Note that, as it was the case in the original product semantics, the modified product semantics is also not distributive.

4.3 N-ary Product Semantics

Since neither the original nor the modified product semantics satisfy distributivity. Therefore, the same query, written in different ways, can have different fuzzy semantics. In order to prevent this, and in order to achieve uniform semantics for multimedia queries, we need to define the uniform fuzzy semantics of a query, Q, as follows.

Definition 4.1 (Uniform Fuzzy Semantics of a Multimedia Query) Let $Q(Y_1, \ldots, Y_n)$ be a query which consists of a set of fuzzy predicates, variables, constants, and conjunction, disjunction, and negation operators. Let also $Q^{\vee}(Y_1, \ldots, Y_n)$ be the disjunctive normal representation of Q. Then, the normal fuzzy semantics of $Q(Y_1, \ldots, Y_n)$ is defined as the fuzzy semantics of $Q^{\vee}(Y_1, \ldots, Y_n)$.

Therefore, whichever way a query is formulated, we define its fuzzy semantics using the fuzzy semantics of its disjunctive normal representation. This provides a uniform definition of fuzziness. Note that, in general, $\mu_Q \neq \mu_{Q^{\vee}}$. The former semantics would be used when logical equivalence of queries is not expected. The latter, on the other hand, would be used when the logical equivalence is required. Note that, as we mentioned earlier, independent of the way a query is rewritten, it is possible to convert the final score into the score, $\mu_{Q^{\vee}}$, of the disjunctive normal representation by using the ratio of differences described earlier.

Note also that, as shown in the following example, the proposed uniform semantics allow implementation of emphasis adjectives, such as *very*.

Example 4.1 A query asking for objects which are semantically similar to *man* and looks like *"a.gif"* can be formulated as

$$Q(X) \leftarrow (X.semantic\_property\ s\_like\ man)\ \ and\ (X.image\_property\ image\_match\ \text{"}a.gif''),$$

whereas a query asking for objects which are semantically very similar to *man* and looks like *"a.gif"* as

$$Q'(X) \leftarrow (X.semantic\_property\ s\_like\ man)\ \ and\ (X.semantic\_property\ s\_like\ man)\ \ and$$
$$(X.image\_property\ image\_match\ \text{"}a.gif'').$$

The scoring function for $Q(X)$ is $\mu_{s\_like}^{\frac{1}{2}} \times \mu_{image\_match}^{\frac{1}{2}}$, whereas the scoring function for $Q'(X)$ is $\mu_{s\_like}^{\frac{2}{3}} \times \mu_{image\_match}^{\frac{1}{3}}$.

The disadvantage of N-ary semantics, however, is that achieving uniformity, it sacrifices the opportunities for query optimizations.

4.4 Accounting for Partial Matches

Both the *min* and the proposed geometric mean functions have weaknesses in supporting partial matches. When, one of the involved predicates return zero, then both functions return 0 as the combined score. However, in multimedia retrieval, partial matches are required (see Section 1). In such cases, having a few number of terms with a score value of 0 in a conjunction should not eliminate the whole conjunctive term from consideration.

As we mentioned earlier, one proposed [4] way to deal with the partial match requirement is to weigh different query criteria in such a way that those criterion that are not important for the user are effectively omitted. For instance, in the query given in Figure 1(Query), if the user knows that spatial information is not important, the user can choose to provide a lower weight to spatial constraints. Consequently, using a weighting technique, the image given in Figure 1(a) can be maintained although the spatial condition is not satisfied. This approach, however, presupposes that users can identify and weigh different query criteria. This assumption may not be applicable to many situations, including databases for naive users or retrieval by QBE (query by example). Furthermore, it is always possible that for each feature or criterion in the query, there may be a set of images in the database that fails it. In such a case, no weighting scheme will be able to handle the partial match requirement for all images.

To account for partial matches, we need to modify the semantics of the n-ary logical operators and eliminate the undesirable nullifying effect of 0. Note that a similar modification can also be done for the *min* semantics.

4.4.1 Supporting Partial Matches

Given a set, $P = \{P_1, \ldots, P_m\}$ of fuzzy sets and $F = \{\mu_1(x), \ldots, \mu_m(x)\}$ of corresponding scoring functions, the semantics of n-ary fuzzy conjunction operator used in SEMCOG is as follows:

$$\mu_{(P_{i1} \wedge \ldots \wedge P_{in})}(t, r_{true}, \beta) = \frac{((\prod_{\mu_k(t) \geq r_{true}} \mu_k(t)) \times (\prod_{\mu_k(t) < r_{true}} \beta))^{1/n} - \beta}{1 - \beta}$$

where

- n is the number of predicates in Q,
- r_{true} is the truth cutoff point, i.e., it is the minimum valid score,
- β is an offset value greater than 0.0 and less than r_{true}; it corresponds to the fuzzy value of *false* and it prevents the combined score to be 0 when one of the predicates has a 0 score, and
- $\mu_k(t)$ is the score of predicate $P_k \in \{P_{i1}, \ldots, P_{in}\}$ for n-tuple t.

The term $(\prod_{\mu_k(t) \geq r_{true}} \mu_k(t)) \times (\prod_{\mu_k(t) < r_{true}} \beta))^{1/n}$ returns a value between β and 1.0. Subtraction of β from it and the subsequent division to $1 - \beta$ normalizes the result to values between 0.0 and 1.0.

Figure 3, shows how the truth cut-off points, r_{true} and β, and the normalization can be utilized to achieve different behavior for the conjunction.

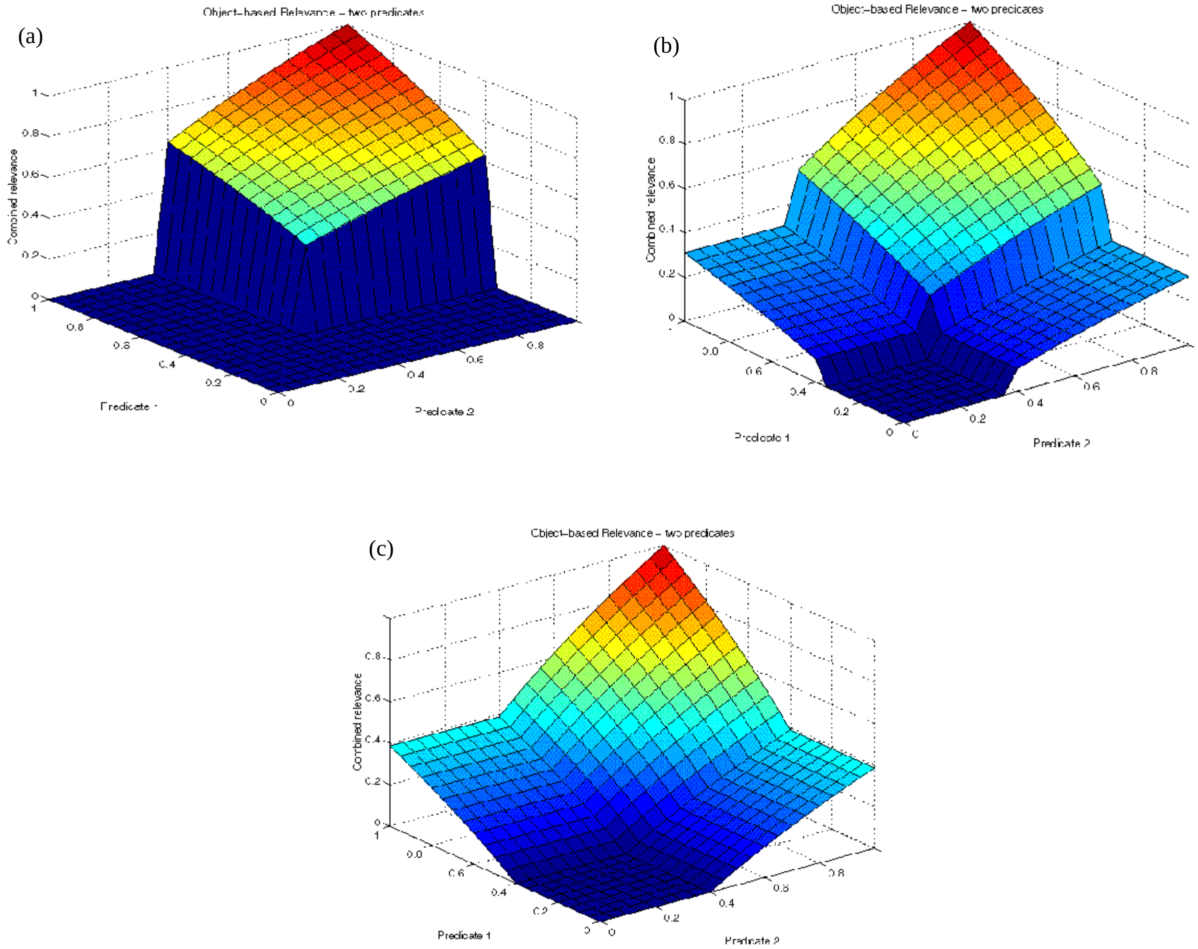


Figure 3: Geometric averaging with different truth cut-off points: (a) $r_{true} = 0.4$ and $\beta = 0.4$, (b) $r_{true} = 0.4$ and $\beta = 0.2$, and (b) $r_{true} = 0.4$ and $\beta = 0.0$ (after normalization). Horizontal axes correspond to the values of the two input predicates and the vertical axis corresponds to the value of the conjunct according to the respective function.

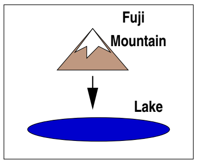
4.4.2 Weighting in the Structural Match

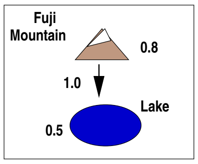
An additional improvement could be made by allowing the conjunction to give extra importance to the predicates which are above the truth cutoff value, r_{true}. To achieve this, one can modify the scoring function as follows:

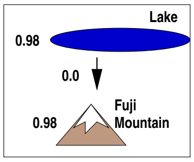
$$\mu_{(P_{i1}\wedge\ldots\wedge P_{in})}(t, r_{true}, \alpha_1, \alpha_2, \beta) = \alpha_1 \times \frac{((\prod_{\mu_k(t)\geq r_{true}} \mu_k(t)) \times (\prod_{\mu_k(t)<r_{true}} \beta))^{1/n} - \beta}{1-\beta} + \alpha_2 \times \frac{\sum_{\mu_k(t)\geq r_{true}} 1}{n},$$

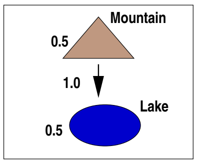
where α_1 and α_2 are values between 0.0 and 1.0 such that $\alpha_1 + \alpha_2 = 1.0$,

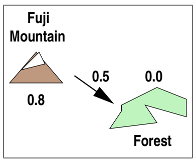
The following is an example where various semantics for logical operators are compared. Among other things, this example clearly shows that any semantics which capture the crisp semantics are not suitable for multimedia retrieval.











Query	Candidate 4		Candidate 1		Candidate 2		Candidate 3	
Semantics	*Score*	*Rank*	*Score*	*Rank*	*Score*	*Rank*	*Score*	*Rank*
min	0.50	**1-2**	0.00	**3-4**	0.50	**1-2**	0.00	**3-4**
product	0.40	**1**	0.00	**3-4**	0.25	**2**	0.00	**3-4**
arithmetic average	0.76	**1**	0.65	**3**	0.66	**2**	0.43	**4**
geometric average	0.74	**1**	0.00	**3-4**	0.63	**2**	0.00	**3-4**
geometric average with cutoff	0.56	**1**	0.55	**2**	0.38	**3**	0.24	**4**
geometric average with weights	0.65	**1**	0.57	**2**	0.51	**3**	0.32	**4**

Figure 4: Comparison of different scoring mechanisms

Example 4.2 In this example, we use the query which was presented in the introduction section to compare scores corresponding to different approaches.

Figure 4 shows a set of candidate images and the associated scores computed by different methods. Numbers next to the objects in the candidate images denote the similarity values for the object level matching. The figure shows the score of the candidate images as well as their relative ranks. The cutoff parameters used in this example are $r_{true} = 0.4$ and $\beta = 0.4$, and the structural weights are $\alpha_1 = 0.8$ and $\alpha_2 = 0.2$.

5 Comparison of the Statistical Properties of Fuzzy Logical Operators

Figure 5 depicts three mechanisms to evaluate conjunction for multimedia retrieval. Figure 5(a) depicts the geometric averaging method, (b) depicts the arithmetic averaging mechanism used by other researchers [18], and (c) the minimum function as described by Zadeh [19] and Fagin [14, 20]. In the previous section, we have compared semantics of these mechanism. In this section, we compare various statistical properties of these semantics and evaluate their applicability to multimedia databases. These statistical properties are especially important to judge the effectiveness of thresholds set for media retrieval. Furthermore, they give an idea about how various database histograms will look.

5.1 Average Score or Cardinality

The first statistical property that we consider is the *average score*, which intuitively measures (assuming a uniform distribution of input scores) the average output score. This value is especially important in setting up thresholds.

The *average score*, or the relative cardinality, of a fuzzy set with respect to its discourse (or domain) is defined as the cardinality of the set divided by the cardinality of its discourse. We can define the

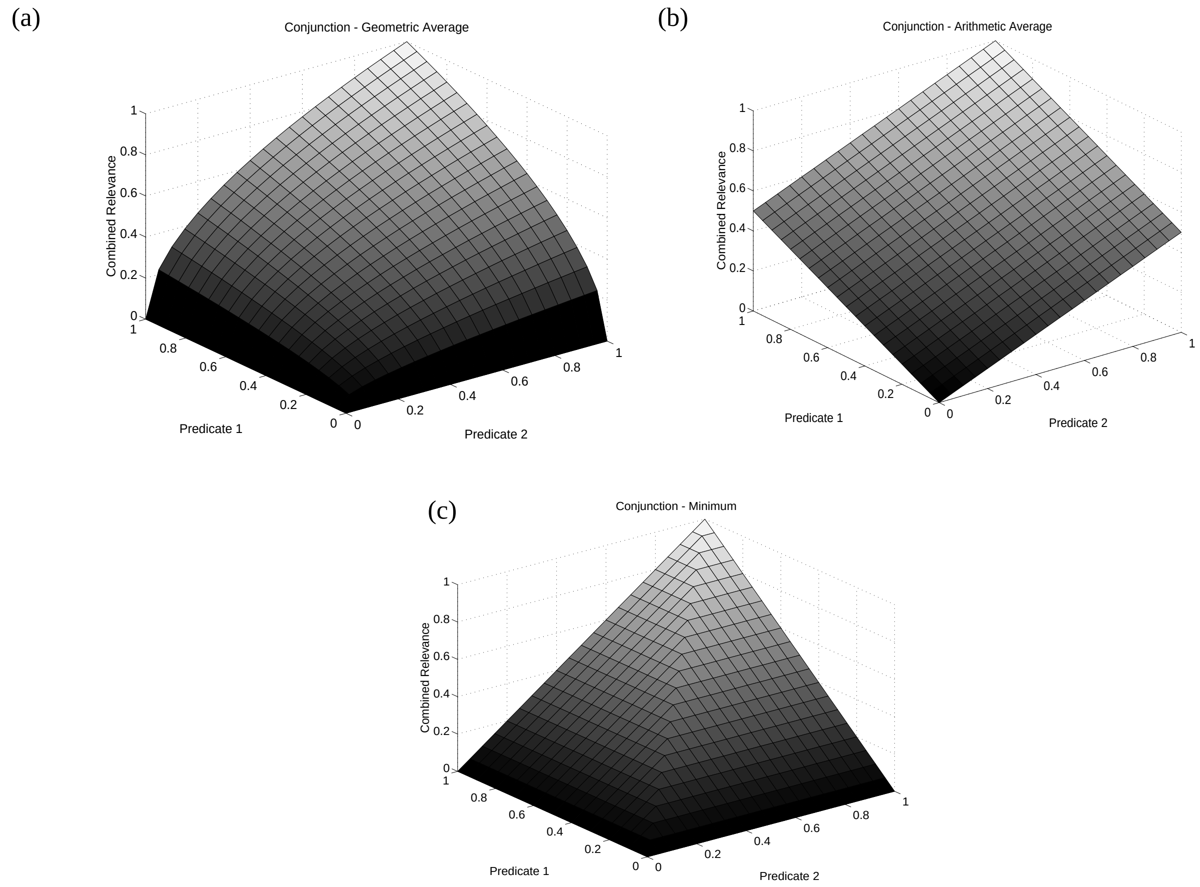


Figure 5: The effect of (a) geometric average, (b) arithmetic average, and (c) minimum function with two predicates. Horizontal axes correspond to the values of the two input predicates and the vertical axis correspond to the value of the conjunct according to the respective function.

relative cardinality of a fuzzy set S with a scoring function $\mu(x)$, where x ranges between 0 and 1 as

$$\frac{\int_0^1 \mu(x)dx}{\int_0^1 dx}.$$

Consequently, the average score of alternative conjunction semantics can be computed as shown in Table 7. Note that, if analogously defined, the relative cardinality of the crisp conjunction would be

$$\frac{\mu_{(false \wedge false)} + \mu_{(false \wedge true)} + \mu_{(true \wedge false)} + \mu_{(true \wedge true)}}{|\{(false \wedge false), (false \wedge true), (true \wedge false), (true \wedge true)\}|} = \frac{0+0+0+1}{4} = \frac{1}{4}.$$

This means that min semantics (Figure 5(c)) is closer to the crisp conjunction semantics, yet it underestimates the geometric average (Figure 5(a)). The arithmetic mean (Figure 5(b)), on the other hand, slightly overestimates the scores. This indicates that the (min, crisp conjunction) and (geometric average, arithmetic average) pairs behave similarly.

Arithmetic average	Min	Geometric average
$\frac{\int_0^1 \int_0^1 \frac{x+y}{2} dydx}{\int_0^1 \int_0^1 dydx} = \frac{1}{2}$	$\frac{\int_0^1 \int_0^1 min\{x,y\} dydx}{\int_0^1 \int_0^1 dydx} = \frac{1}{3}$	$\frac{\int_0^1 \int_0^1 \sqrt{x \times y}\, dydx}{\int_0^1 \int_0^1 dydx} = \frac{4}{9}$

Table 7: Average score of various scoring semantics

Arithmetic average	Min	Geometric average
$1 - \frac{8 \times \alpha^3}{3} \quad (\alpha \leq 0.5)$ $\frac{4}{3} - 4 \times \alpha^2 + \frac{8 \times \alpha^3}{3} \quad (\alpha \geq 0.5)$	$1 - 3 \times \alpha^2 + 2 \times \alpha^3$	$1 - \alpha^3 + 3 \times \alpha^3 \times ln(\alpha)$

Table 8: Score distribution (relative cardinality of a strong α-cut) of various scoring semantics

5.2 Score Distribution

The study of the score distribution of fuzzy algebraic operators is essential in creating histograms that can be used in query optimization and processing. For instance, in the next section, we will use histograms to find the set of approximate results to a fuzzy query.

The strong α-cut of a fuzzy set is defined as the set of elements of the discourse that have score values equal to or larger than α. The relative cardinality of a strong α-cut ($\mu_{x \wedge y} \geq \alpha$) of a conjunction with respect to its overall cardinality ($\mu_{x \wedge y} \geq 0$) describes the concentration of scores above a threshold α.

Table 8 shows the score distribution of various scoring semantics. When, α is close to 1, the relative cardinality of the strong α-cut of the geometric average behaves like $1 - \alpha^3$. Consequently, both arithmetic average and min semantics have a lower concentration ratio of scores above a given α compared to the geometric average.

Figure 6 visually shows the score distribution of the geometric average and the minimum functions for a conjunction of three predicates. According to this figure, higher scores are confined to a smaller region in the min function. This implies that, as intuitively expected, given a threshold, the min function is likely to eliminate more candidates than the geometric average.

5.3 Relative Importance of Query Criteria

An important advantage of the geometric average, against the arithmetic average and the min functions, is that although it shows a linear behavior when the similarity values of the predicates are close to each other:

- $(x = y) \quad \longrightarrow \quad \frac{d\mu(x,y)}{dxdy} = \frac{1}{2 \times \sqrt{x \times x}} \times (x + x) = \frac{1}{2 \times x} \times (2 \times x) = 1,$

it shows a non-linear behavior when one of the predicates has lower similarity compared to the others:

- $(x << y) \quad \longrightarrow \quad \frac{d\mu(x,y)}{dx} = \frac{\sqrt{y}}{2 \times \sqrt{x}} = \infty \qquad (x >> y) \quad \longrightarrow \quad \frac{d\mu(x,y)}{dx} = 0.$

Example 5.1 The first item below shows the linear increase in the score of the geometric average when the input values are closer to each other. The second item, on the other hand, shows that non-linearity of the increase when input values are different:

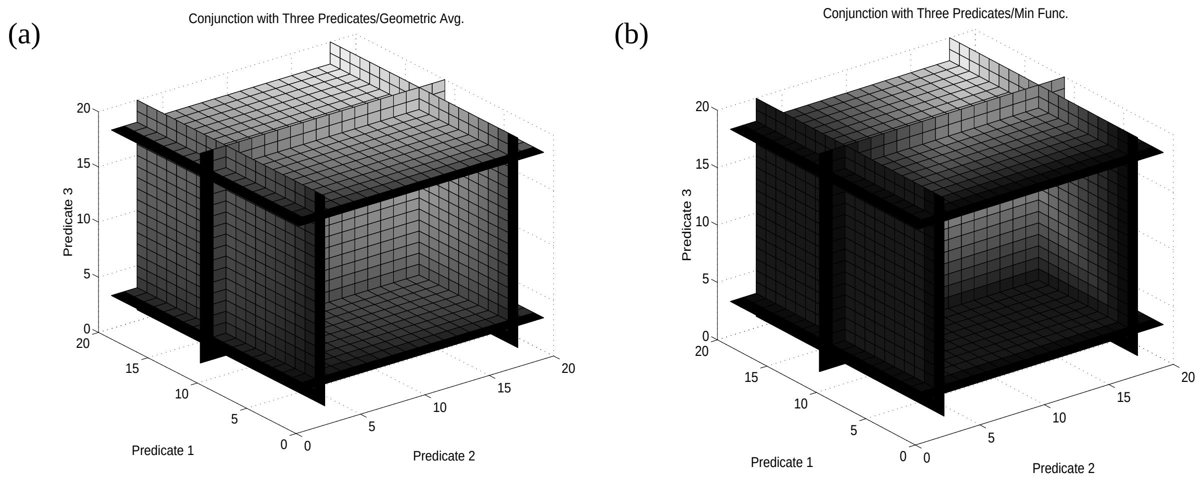


Figure 6: (a) Geometric averaging vs. (b) minimum with three predicates. Each axis correspond to an input predicate and the gray level represents the value of the combined score (the brighter the gray, the higher the score).

- $(0.5 \times 0.5 \times 0.5)^{1/3} = 0.5,\ (0.6 \times 0.6 \times 0.6)^{1/3} = 0.6,$ and $(0.7 \times 0.7 \times 0.7)^{1/3} = 0.7.$

- $(1.0 \times 1.0 \times 0.5)^{1/3} = 0.79,\ (1.0 \times 1.0 \times 0.6)^{1/3} = 0.85,$ and $(1.0 \times 1.0 \times 0.7)^{1/3} = 0.88.$

It has been claimed that according to real-world and artificial nearest-neighbor workloads, the highest-scoring predicates are interesting and the rest are not interesting [21]. This implies that the *min* semantics which gives the highest importance on the lowest scoring predicate may not be suitable for real workloads. The geometric average semantics, unlike the *min* semantics, on the other hand, does not suffer from this behavior.

Furthermore, the effect of an increase in the score of a subquery (due to a modification/relaxation on the query by the user or the system) with a small score value is *larger* than an equivalent increase in the score of a subquery with a large score value. This implies that, although the subqueries with a high score have a larger role in determining the final score, relaxing a non-satisfied subquery may have a significant impact on improving the final score. This makes sense as an increase in a low scoring subquery increases the *interestingness* of the subquery itself.

6 Implementation

The similarity evaluation concepts were implemented as part of the Semantic and Cognition-based Image Retrieval (SEMCOG) system. SEMCOG query processor performs the following tasks: (1) Query relaxation: It reformulates query statements containing semantic, such as *is_a* or *semantically_like*, predicates by consulting a *terminology manager*; (2) Query forwarding: It forwards query statements containing *image_like* or *contains* (visually) predicates to a *image data query processor* to perform image matching while forwarding other non-image-matching operation statements to *textual and spatial data query processor*; and (3) Result integration: It merges results of image-matching query statements and non-image-matching operation statements. This task includes computing overall similarity values and eliminating images based on specified filtering constraints (Figure 7).

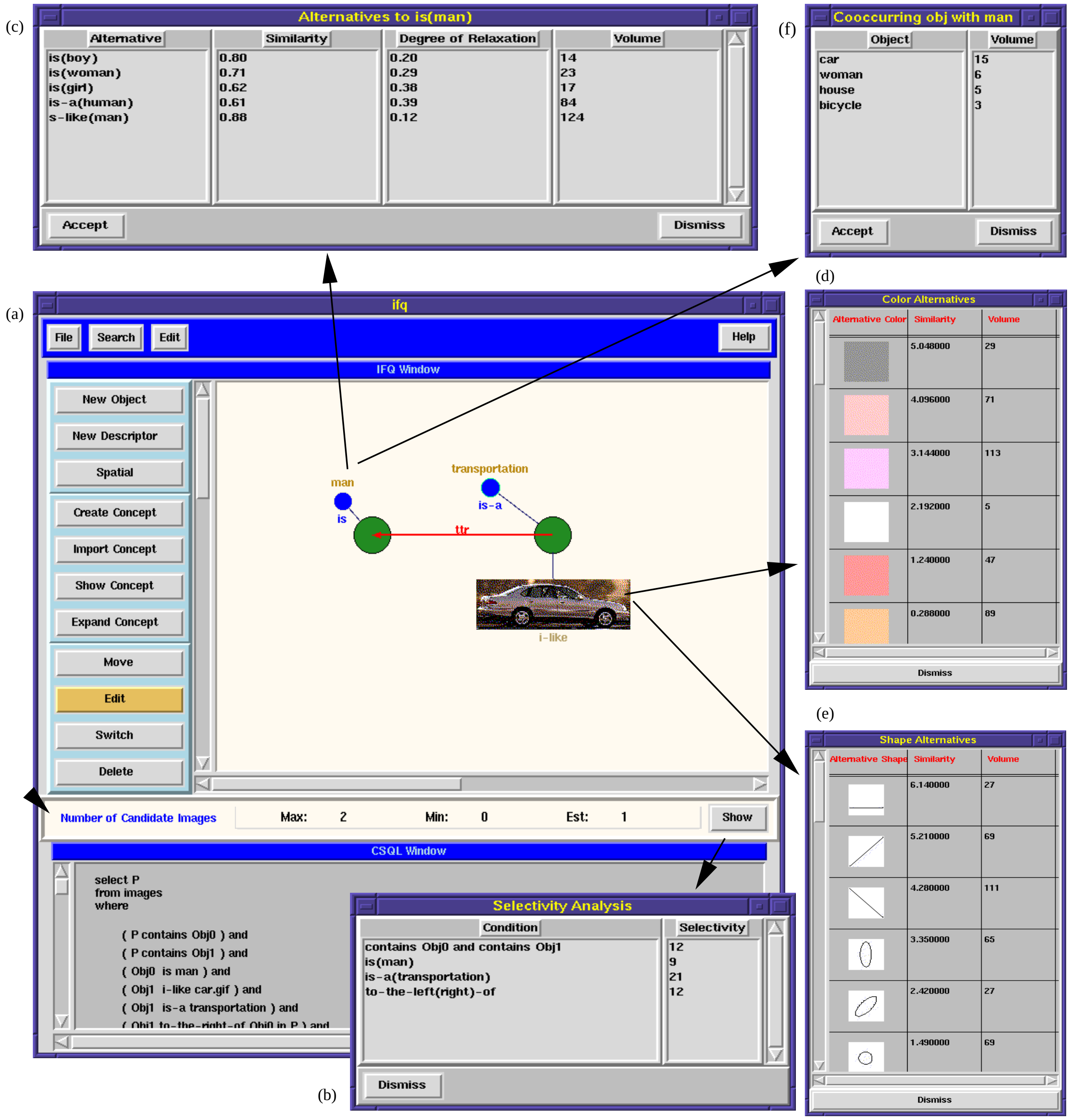


Figure 7: A SEMCOG query that contains multiple predicates to be merged

Note that the merge function used has an effect on the precision and recall of the algorithm used for retrieval. There are four factors that affect precision and recall: (1) user preferences, (2) application semantics, (3) the distance function that compares objects in the database, and (4) the indices and algorithms used. In SEMCOG, as described in earlier sections, the distance function is a combination of a merge function and the individual fuzzy predicates (such as *image_like* or *semantically_like*). Although user preferences are hard to formalize, we see that the proposed merge function captures the application semantics well:

- The structural information (such as spatial predicates) are accounted for by the scoring function.
- As discussed in the previous section, the geometric average captures the behavior of real-world workloads (the highest-scoring predicates are interesting and the rest is not interesting [21]) better than the other merge functions.

Furthermore, the α and β parameters of the proposed merge function, allow merge function to be fine tuned to the actual application semantics or user preferences.

7 Conclusion

In this paper, we have presented the difference between the general fuzzy query and multimedia query evaluation problems. More specifically, we have pointed to the multimedia precision/recall semantics, partial match requirement, and unavoidable necessity of fuzziness, but non-progressive predicates. We have, then, shown that the popular fuzzy logic semantics do not fulfill the required multimedia semantics and the partial match requirement. We have proposed an alternative semantics that is capable of capturing the needs of multimedia queries and we have discussed the properties of the proposed semantics.

References

[1] Wen-Syan Li and K. Selçuk Candan. SEMCOG: A Hybrid Object-based Image Database System and Its Modeling, Language, and Query Processing. In *Proceedings of the 14th International Conference on Data Engineering*, Orlando, Florida, USA, February 1998.

[2] Wen-Syan Li, K. Selçuk Candan, Kyoji Hirata, and Yoshinori Hara. Facilitating Multimedia Database Exploration through Visual Interfaces and Perpetual Query Reformulations. In *Proceedings of the 23th International Conference on Very Large Data Bases*, pages 538–547, Athens, Greece, August 1997. VLDB.

[3] Wen-Syan Li, K. Selçuk Candan, Kyoji Hirata, and Yoshinori Hara. Hierarchical Image Modeling for Object-based Media Retrieval. *Data & Knowledge Engineering*, 27(2):139–176, July 1998.

[4] R. Fagin and E. L. Wimmers. Incorporating User Preferences in Multimedia Queries. In F. Afrati and P. Koliatis, editors, *Database Theory – ICDT '97*, volume 1186 of *LNCS*, pages 247–261, Berlin, Germany, 1997. Springer Verlag.

[5] R. Fagin and Y. S. Maarek. Allowing Users to Weight Search Terms. Technical Report RJ10108, IBM Almaden Research Center, San Jose, CA, 1998.

[6] S. Y. Sung. A Linear Transform Scheme for Combining Weights into Scores. Technical Report TR98-327, Rice University, Houston, TX, 1998.

[7] Moshé M. Zloof. Query-by-example: the invocation and definition of tables and forms. In Douglas S. Kerr, editor, *Proceedings of the International Conference on Very Large Data Bases, September 22-24, 1975, Framingham, Massachusetts, USA*, pages 1–24. ACM, 1975.

[8] Christos Faloutsos. *Searching Multimedia Databases by Content.* Kluwer Academic Publishers, Boston, 1996.

[9] R. Richardson, Alan Smeaton, and John Murphy. Using Wordnet as a Knowledge base for Measuring Conceptual Similarity between Words. In *Proceedings of Artificial Intelligence and Cognitive Science Conference*, Trinity College, Dublin, 1994.

[10] Weining Zhang, Clement Yu, Bryan Reagan, and Hiroshi Nakajima. Context-Dependent Interpretations of Linguistic Terms in Fuzzy Relational Databases. In *Proceedings of the 11th International Conference on Data Engineering*, Taipei, Taiwan, March 1995. IEEE.

[11] H. Bandermer and S. Gottwald. *Fuzzy Sets, Fuzzy Logic, Fuzzy Methods with Applications.* John Wiley and Sons Ltd., England, 1995.

[12] U. Thole, H.-J. Zimmerman, and P. Zysno. On the Suitability of Minimum and Product operators for the Intersection of Fuzzy Sets. *Fuzzy Sets and systems*, pages 167–180, 1979.

[13] J. Yen. Fuzzy Logic–A Modern Perspective. *IEEE Transactions on Knowledge and Data Engineering*, 11(1):153–165, January 1999.

[14] Ronald Fagin. Fuzzy Queries in Multimedia Database Systems. In *17^{th} ACM Symposium on Principles of Database Systems*, pages 1–10, June 1998.

[15] D. Dubois and H. Prade. Criteria Aggregation and Ranking of Alternatives in the Framework of Fuzzy Set Theory. *Fuzzy Sets and Decision Analysis, TIMS Studies in Management Sciences*, 20:209–240, 1984.

[16] R.R. Yager. Some Procedures for Selecting Fuzzy Set-Theoretic Operations. *International Jounral General Systems*, pages 115–124, 1965.

[17] S. Adali, K.S. Candan, Y. Papakonstantinou, and V.S. Subrahmanian. Query Caching and Optimization in Distributed Mediator Systems. In *Proceedings of the 1996 ACM SIGMOD Conference*, pages 137–147, Montreal, Canada, June 1996.

[18] Y. Alp Aslandogan, Chuck Thier, Clement Yu, Chengwen Liu, and Krishnakumar R. Nair. Design, Implementation and Evaluation of SCORE. In *Proceedings of the 11th International Conference on Data Engineering*, Taipei, Taiwan, March 1995. IEEE.

[19] L. Zadeh. Fuzzy Sets. *Information and Control*, pages 338–353, 1965.

[20] Ronald Fagin. Combining Fuzzy Information from Multiple Systems. In *15^{th} ACM Symposium on Principles of Database Systems*, pages 216–226, 1996.

[21] K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft. When is “Nearest Neighbor” Meaningful? In *Database Theory – ICDT '99*, Berlin, Germany, 1999. Springer Verlag. to appear.

Kasım Selçuk Candan is a tenure track assistant professor at the Department of Computer Science and Engineering at the Arizona State University. He joined the department in August 1997, after receiving his Ph.D. from the Computer Science Department at the University of Maryland at College Park. His dissertation research concentrated on multimedia document authoring, presentation, and retrieval in distributed collaborative environments. He received the 1997 ACM DC Chapter award of Samuel N. Alexander Fellowship for his Ph.D. work. His reserach interests include development of formal models, indexing schemes, and retrieval algorithms for multimedia and Web information and development of novel query optimization and processing algorithms. He has published various articles in respected journals and conferences in related areas. He received his B.S. degree, first ranked in the department, in computer science from Bilkent University in Turkey in 1993.

Wen-Syan Li is a Senior Research Staff Member at Computers & Communications Research Laboratories (CCRL), NEC USA Inc. He also affiliates with NEC USA's Venture Development Center in San Jose, CA. He received his Ph.D. in Computer Science from Northwestern University in December 1995. He also holds an MBA degree. His main research interests include Web acceleration technologies, multimedia/hypermedia/document databases, WWW, and Internet/intranet search engines.