A Fuzzy Algebra for Image Data Bases

A. Picariello
Università Napoli "Federico II"
antonio.picariello@unina.it

M.L. Sapino
Università di Torino
mlsapino@di.unito.it

Abstract

In this paper, we describe how image data may be organized so to facilitate uncertainty management. We develop a fuzzy model for image data and provide an algebra for dealing with such data. New algebraic operators are defined in order to capture the fuzziness related to the semantic descriptors of an image.

1 Introduction

The goal of an image retrieval system is to find images from an image data base while processing a query provided by a user. In the last decade, most of researches are focused on *Content Based Image Retrieval* $CBIR$ [1], [2].

In such a system, the query is in the form of an *image*, that is a sample image, or a user sketch that is somehow related to the image to be retrieved, or a textual description (*annotation*). All the images that are similar to the query according to certain criteria are retrieved.

To perform this task, a set of features encoding certain characteristics is found, and the feature space is endowed with an appropriate metric.

As noted by [3], metric axioms - both geometrical and perceptual - are strongly related to *uncertainty*, which represents the grade of similarity, or dissimilarity, of two images.

In this framework, managing uncertainty becomes a fundamental topic in image database, and Fuzzy set theory can be ideally suited to deal with this kind of uncertainty. As described by [4], this fuzziness is mainly due to similarity of media features, imperfection in the feature extraction algorithms, imperfection in the query methods, partial match requirements, and imperfection in the index structures.

Several extensions of relational databases have been proposed to deal with uncertain or imprecise information [5], [6], [7], [8], [9]. The relational model of data has been extended to incorporate uncertainty either at the tuple level or at the attribute level. In the tuple-level approach, each tuple may have one or more uncertainty attributes. The uncertainty attribute usually is a real number or an interval on real numbers. In the attribute level approach, the information about the certainty degree is associated directly to the single attribute values. In image databases, attribute level approaches seem preferable, since in content based image description approaches, each feature is produced by a distinct image processing technique, that returns multiple possible answers, with different grade of certainty [10].

In this paper we define a data model and an algebra to deal with fuzziness at attribute level in image data bases.

The paper is organized as follows. In section 2 we give some examples in order to describe and motivate the problem we are addressing; in section 3 we propose a data model based on non normal form relations (NF^2). In section 4 algebraic operators for this model are defined. Concluding remarks are reported in section 5 .

2 Preliminary Considerations and Examples

Given a certain space $\mathcal{I}$, and a set of classes of membership $\mathcal{M}$, the retrieval process can be defined as a function performed by a system $\mathcal{V}$ - a human or a computer vision system - which associates to any feature in $\mathcal{I}$ a set of elements from $\mathcal{M}$, each associated with a certain grade of membership $p \in \mathcal{P}$, i.e.:

$$\mathcal{V} : \mathcal{I} \rightarrow 2^{\mathcal{M} \times \mathcal{P}} \tag{1}$$

Figure 1: A sample image

If $\mathcal{V}(i) = \{(c_1, p_1), \ldots, (c_k, p_k)\}$, then for any matched class c_i, p_i is a measure of the membership of i to c_i.

The key point is the meaning of the space $\mathcal{I}$. In our model, $\mathcal{I}$ is the set of all the Images in the given Image Database. Each image can be characterized in terms of (i) its binary data and (ii) the information extracted from the image, using the binary data or provided by the user.

We assume that the given space $\mathcal{I}$ is the set of the bitmaps of all the images in the databases, and the function $\mathcal{V}$ processes binary rough images, to produce a number of descriptions for each of them, with an associated grade of *uncertainty*.

Even if in some cases the image description comes from human written annotations, the task of associating a high level textual feature to a low level visual descriptor is usually performed by means of image processing and analysis algorithms. These methods return high level features, with an associated *grade of confidence*, that is the grade of membership of the given feature to a discovered pattern.

The purpose of this paper is to provide a data model to store and manage the information extracted from image databases. In particular, we aim at introducing a model in which the files containing the bitmap images are directly associated with all the (uncertain) information extracted from it by humans and/or image processing systems, and to manipulate such information to derive new information out of it.

Before moving to the formal definition of the model, and of the operators defined on it, we show some introductory examples.

Example 2.1 Let us consider the image in figure 1.

Usually, a vision system analyzes the given image and extracts information that are useful for a high level description. A useful description of the image may be given by means of colors, textures, shapes and spatial descriptors.

For example, in [11], the system discovers that in the above image there are the colors: *blue*, with confidence 0.82 and *beige* with confidence 0.765 and *green*, with confidence 0.449; it is also discovered that it has a *thin texture* with confidence 0.704, and that it has been recognized as a *SeaCoastPicture* with confidence 0.939.

The following table shows how the information extracted from figure 1 is organized in our model.

File	**Color**	**Texture**	**Content**
Im1.gif	{ ⟨blue,0.827 ⟩, ⟨beige, 0.765 ⟩, ⟨green, 0.449 ⟩},	{⟨thin,0.704 ⟩}	{⟨Sea Coast,0.939 ⟩}

Alternatively, the same information could be organized in the following table (in first normal form).

File	**Color**	**unc color**	**Texture**	**unc text**	**Content**	**unc cont**
Im1.gif	blue	0.827	thin	0.704	Sea Coast	0.939
Im1.gif	beige	0.765	thin	0.704	Sea Coast	0.939
Im1.gif	green	0.449	thin	0.704	Sea Coast	0.939

Observe that the first representation provides a more intuitive, flexible, compact and powerful description of image features. It may be easily used for describing a fuzzy data bases.

Example 2.2 Consider the query: *"find all the images that are similar to figure 1"*.

Assume that a certain system $\mathcal{V}$ has produced the data stored in the relation described in the first example. So, the query may be rephrased as: *"find all the images that have as predominant colors blue, beige and green, that have*

a thin texture and that represent a Sea Coast". The problems which arise are: i) how do we manage data at the attribute level, e.g. colors blue, green and beige, and ii)how do we manage the uncertainties associated to these data to process the given query.

In the next sections we address these topics.

3 Fuzzy Data Model

Our data model is based on fuzzy logic [12]. For the sake of clarity, we first provide a short introduction to fuzzy set theory and its fundamental operators.

Fuzzy sets are defined on non fuzzy universes of discourse, which are ordinary sets. Given any universe of discourse $\mathcal{U}$, a fuzzy subset S of $\mathcal{U}$ is characterized by a membership function $\mu_S : \mathcal{U} \rightarrow [0, 1]$, which assigns to every element $x \in \mathcal{U}$ a membership degree. Thus, the subset S is defined as $S = \{\langle u, \mu_S(u)\rangle \mid u \in \mathcal{U}\}$

An element u is said to be in the fuzzy set S if and only if $\mu_S(u) > 0$, and to be a full member of the set if and only if $\mu_S(u) = 1$.

Fuzzy data models can be interpreted as extensions of traditional data models using fuzzy set theory and the possibility theory.

The usual set theoretic operators are extended to fuzzy sets as follows:

Fuzzy union (or) $A \underline{\vee} B = \{\langle u, max(\mu_A(u), \mu_B(u))\rangle \mid u \in \mathcal{U}\}$

Fuzzy intersection (and) $A \underline{\wedge} B = \{\langle u, min(\mu_A(u), \mu_B(u))\rangle \mid u \in \mathcal{U}\}$

Fuzzy complementation (not) $\underline{\neg} A = \{\langle u, 1 - \mu_A(u))\rangle \mid u \in \mathcal{U}\}$

Definition 3.1 (fuzzy tuple) Let $D_1, \ldots, D_n$ be n domains. A fuzzy n-tuple is any element of the cartesian product $V_1 \times \ldots \times V_n$, where each V_i is the set of all fuzzy subsets of D_i.

In other words, any n-tuple is a sequence $\langle v_1, \ldots, v_n\rangle$, where each v_i is either a non fuzzy, atomic value, or a set of elements from the corresponding fuzzy domain D_i.

Definition 3.2 (fuzzy relation schema) A fuzzy relational schema is defined as a symbol R, that is the name of the fuzzy relation, and a set of (names of) fuzzy attributes $X = \{A_1, \ldots, A_n\}$. The schema is usually denoted as $R(X)$.

A fuzzy relation is an instance of a fuzzy relation schema; that is, a fuzzy relation is a set of fuzzy tuples. Formally,

Definition 3.3 (fuzzy relation) Let $R(\{A_1, A_2, \ldots, A_n\})$ be a relational schema. A *fuzzy relation*, defined over R, is a set of fuzzy tuples $t = \langle v_1, \ldots, v_n\rangle$ such that each v_i is a fuzzy subset of $dom(A_i)$

4 Fuzzy Algebra

In this section we propose our operators, specifically conceived for our data model. A fuzzy algebra for the above data model is constructed as an enhancement of the relational algebra. The fundamental algebraic operations provided by our model are union, set difference, cartesian product, join, projection and selection. We first define the set oriented operators union, intersection, and difference. Then we define the algebraic operators selection, join and projection. Before moving to the formal definition of the algebraic operators, we give some preliminary notions.

4.1 Preliminary Definitions

Definition 4.1 (Data identical tuples) Let $R(\{A_1, A_2, \ldots, A_n\})$ be a relation schema, and r be an instance of R. Let $t_1 = \langle v_1^1, \ldots, v_n^1\rangle$ and $t_2 = \langle v_1^2, \ldots, v_n^2\rangle$ denote two tuples of r. t_1 and t_2 are *data identical* iff for every index i, it holds that $\{u \mid \langle u, \mu_{dom(A_i)}(u)\rangle \in v_i^1\} = \{u \mid \langle u, \mu_{dom(A_i)}(u)\rangle \in v_i^2\}$

In other words, two tuples are data identical if they are identical, modulo the certainty component of the elements.

We use the notation $DId(t, r)$ to denote the set of all tuples in r that are data identical to the tuple t.

The above notion of data identical tuples, and the following fuzzy or $\underline{\bigvee}$ operator on data identical tuples, play an important role in the definition of some algebraic operators, e.g., in the definition of *union*.

Definition 4.2 Let $t_1 = \langle v_1^1, \ldots, v_n^1 \rangle$ and $t_2 = \langle v_1^2, \ldots, v_n^2 \rangle$ two data identical tuples.

$$t_i \underline{\bigvee} t_2 = \langle v_1, \ldots, v_n \rangle$$

where for each i, $v_i = \{\langle u, \mu_1(u) \rangle \underline{\vee} \langle u, \mu_2(u) \rangle \mid \langle u, \mu_1(u) \rangle \in v_i^1, \langle u, \mu_2(u) \rangle \in v_i^2\}$.

Analogously, the fuzzy "and" operator can be extended to the new operators $\underline{\bigwedge}$, defined on data identical tuples.

Definition 4.3 Let $t_1 = \langle v_1^1, \ldots, v_n^1 \rangle$ and $t_2 = \langle v_1^2, \ldots, v_n^2 \rangle$ two data identical tuples.

$$t_i \underline{\bigwedge} t_2 = \langle v_1, \ldots, v_n \rangle$$

where for each i, $v_i = \{\langle u, \mu_1(u) \rangle \underline{\wedge} \langle u, \mu_2(u) \rangle \mid \mu_1(u) \neq 0, \mu_2(u) \neq 0, \langle u, \mu_1(u) \rangle \in v_i^1, \langle u, \mu_2(u) \rangle \in v_i^2\}$.

According to the definition of data identical tuples, it might be the case that two tuples are not considered data identical simply because of the presence of some datum, associated with a zero membership degree. To give an example, the tuples $\langle\{\langle red, 0.5\rangle\}\rangle$, and $\langle\{\langle red, 0.5\rangle, \langle green, 0.0\rangle\}\rangle$, are not considered data identical, even if intuitively, they are.

The reason of that choice is the following: we assume that all the fuzzy values are returned by some authomatic vision systems, or result from some algebraic operations. Initial values, that is, data provided by visual systems, are only non zero values, since systems give information about features they find, to a certain degree, in the considered image. On the other hand, as it will be clearer in the following, the only algebraic operations that, applied to tuples without zero membership degrees, could introduce zero membership degrees, are those that apply fuzzy conjunction, combined with complementation of certain values (e.g., set difference). But the the fuzzy and operator, applied to tuples, takes care of the problem, and avoids introducing meaningless values.

Here are some examples.

t_1

File	**Color**	**Shape**	**Content**
Im1.gif	{⟨red,0.78 ⟩}	{⟨circle, 0.79 ⟩, ⟨ellipse,0.89 ⟩}	{⟨Sun in the dark, 0.7 ⟩}

t_2

File	**Color**	**Shape**	**Content**
Im1.gif	{⟨red,0.78 ⟩}	{ ⟨circle,0.70 ⟩, ⟨ellipse,0.95 ⟩}	{⟨Sun in the dark, 0.7 ⟩}

$t = t_1 \underline{\bigvee} t_2$

File	**Color**	**Shape**	**Content**
Im1.gif	{⟨red,0.78 ⟩}	{ ⟨circle, 0.79 ⟩, ⟨ellipse,0.95 ⟩}	{ ⟨Sun in the dark, 0.7 ⟩}

$t = t_1 \underline{\bigwedge} t_2$

File	**Color**	**Shape**	**Content**
Im1.gif	{ ⟨red,0.78 ⟩}	{⟨circle, 0.70 ⟩, ⟨ellipse,0.89 ⟩}	{⟨Sun in the dark, 0.7 ⟩}

$t = \underline{\neg} t_1$

File	**Color**	**Shape**	**Content**
Im1.gif	{⟨red,0.22⟩}	{⟨circle,0.21 ⟩, ⟨ellipse,0.05 ⟩}	{⟨Sun in the dark, 0.3⟩}

Notice that $\underline{\bigwedge}$ and $\underline{\bigvee}$ are both associative and commutative. Thus, they may be seen as operators that apply to sets of more than two data identical tuples, as follows:

$\underline{\bigwedge}(\{t_1, t_2, \ldots t_k\}) = (\ldots(t_1 \underline{\bigwedge} t_2) \ldots \underline{\bigwedge} t_k)$

$\underline{\bigvee}(\{t_1, t_2, \ldots t_k\}) = (\ldots(t_1 \underline{\bigvee} t_2) \ldots \underline{\bigvee} t_k)$

4.2 Algebra Operators

4.2.1 Compaction

Definition 4.4 (Compaction) Let r be a fuzzy relation. A compaction of r is a relation:

$$K(r) = \{t \mid t \in r \text{ and } DId(t, r - \{t\}) = \emptyset\} \cup \{t_1 \underline{\bigvee} t_2 \mid t_1, t_2 \in r \text{ and } t_1, t_2 \text{ are data identical }\}$$

In other words, a compaction operation normalize a relation r in this way:

- taking the tuples in r for which there is no data identical tuple in r.
- collapsing all data identical tuples in a single tuple, that is their fuzzy "or" $\underline{\bigvee}$.

Definition 4.5 *Compact relation instance* A relation instance r is said to be *compact* iff $K(r) = r$.

4.2.2 Union

Definition 4.6 (Union) Let r_1 and r_2 be two union compatible relations, i.e. let r_1 and r_2 have the same fuzzy relational schema $R(\{A_1, A_2, \ldots, A_n\})$.

$$r_1 \underline{\cup} r_2 = \{t \mid t \in r_1 \text{ and } DId(t, r_2) = \emptyset\} \cup \{t \mid t \in r_2 \text{ and } DId(t, r_1) = \emptyset\} \cup \{t_1 \underline{\bigvee} t_2 \mid t_1 \in r_1, t_2 \in r_2, \text{ and } t_1 \text{ and } t_2 \text{ are data identical}\}$$

Unioning two relations r_1 and r_2 means:

- taking the tuples in r_1 for which there is no data identical tuple in r_2;
- taking the tuples in r_2 for which there is no data identical tuple in r_1;
- for the remaining tuples, taking the fuzzy or $\underline{\bigvee}$ of pairwise data identical tuples.

Example 4.1 Consider the two relations r_1 and r_2 shown in the following two tables:

r_1

File	Color	Shape	Content
Im1.gif	{⟨red,0.78 ⟩, ⟨grey, 0.80 ⟩}	{⟨circle, 0.79 ⟩, ⟨ellipse,0.89 ⟩}	{⟨Sun in the dark, 0.7 ⟩}
Im2.gif	{⟨blue, 0.9 ⟩, ⟨black, 0.3 ⟩}	{⟨rectangle, 0.94⟩, ⟨square, 0.9 ⟩}	{⟨Sea, 0.8 ⟩}

r_2

File	Color	Shape	Content
Im3.gif	{⟨blue,0.78 ⟩, ⟨grey, 0.80 ⟩}	{⟨circle, 0.79 ⟩, ⟨ellipse,0.89 ⟩}	{ ⟨grass, 0.7 ⟩}
Im2.gif	{⟨blue, 0.4 ⟩, ⟨black, 0.5 ⟩}	{⟨rectangle, 0.94 ⟩, ⟨square, 0.9 ⟩}	{⟨Sea, 0.8 ⟩}

r_1 and r_2 are union compatible, and the second tuple in r_1 and the second tuple in r_2 are data identical. $r_1 \underline{\cup} r_2$ is shown in the following table.

File	Color	Shape	Content
Im1.gif	{⟨red,0.78 ⟩, ⟨grey, 0.80 ⟩}	{⟨circle,0.79 ⟩, ⟨ellipse,0.89 ⟩}	{⟨Sun in the dark, 0.7 ⟩}
Im3.gif	{⟨blue,0.78 ⟩,⟨grey, 0.80 ⟩}	{⟨circle,0.79 ⟩, ⟨ellipse,0.89 ⟩}	{⟨grass, 0.7 ⟩}
Im2.gif	{⟨blue, 0.9 ⟩, ⟨black, 0.5 ⟩}	{⟨rectangle, 0.94 ⟩, ⟨square, 0.9 ⟩}	{ ⟨Sea, 0.8 ⟩}

4.2.3 Intersection

The definition of the intersection operator is, analogously, a natural extension of the standard set intersection operator. In this case, the equality requirement for elements in the two given sets, that will appear in their intersection, is replaced by the *data identical* condition.

Definition 4.7 (Intersection) Let r_1 and r_2 be two union compatible relations, i.e. let r_1 and r_2 have the same fuzzy relational schema $R(\{A_1, A_2, \ldots, A_n\})$.

$r_1 \underline{\cap} r_2 = \{t_1 \underline{\bigwedge} t_2 \mid t_1 \in r_1, t_2 \in r_2,$ and t_1 and t_2 are data identical$\}$

Example 4.2 Consider again the above relations r_1 and r_2. The intersection $r_1 \underline{\cap} r_2$ is the relation r_3 shown in the following table.

$r_1 \underline{\cap} r_2$

File	Color	Shape	Content
Im2.gif	{⟨blue, 0.4 ⟩, ⟨black, 0.3 ⟩}	{⟨rectangle, 0.94 ⟩, ⟨square, 0.9 ⟩}	{⟨Sea, 0.8 ⟩}

4.2.4 Difference

Our difference operator extends the standard set difference operator in the same way as intersection extends the corresponding standard operator: given two relations, tuples from the first one that do not have any corresponding data identical tuple in the second one belong to the difference set, as well as the "difference", expressed in terms of conjunction with the complement, of the data identical ones.

Definition 4.8 (Difference) Let r_1 and r_2 be two union compatible relations,

$r_1 \underline{\setminus} r_2 = \{t \mid t \in r_2$ and $DId(t, r_1) = \emptyset)\}$

$\cup \{t \underline{\bigwedge} \underline{\neg} t' \mid t \in r_1$ and $t' \in r_2$and t,t' are data identical $\}$

Example 4.3 Consider again the above relations r_1 and r_2. The difference $r_1 \underline{\setminus} r_2$ is the relation r_3 shown in the following table.

$r_1 \underline{\setminus} r_2$

File	Color	Shape	Content
Im3.gif	{⟨blue,0.78 ⟩, ⟨grey, 0.80 ⟩}	{ ⟨circle,0.79 ⟩, ⟨ellipse,0.89 ⟩}	{⟨grass, 0.7 ⟩}
Im2.gif	{⟨blue, 0.6 ⟩, ⟨black, 0.3 ⟩}	{⟨rectangle, 0.06 ⟩, ⟨square, 0.1 ⟩}	{⟨Sea, 0.2 ⟩}

4.2.5 Value Conditions

We are now ready to move to the definition of the algebraic operators. As in the standard relational algebra, the selection operator is defined in terms of some conditions that have to be satisfied by the selected tuples. Thus, we will first introduce our notions of selection condition, and selection condition satisfaction.

Definition 4.9 (atomic value condition) Let $r(X)$ be a a relation, $A \in X$ and $B \in X$ be two attributes of r, and $\theta \in \{=, \neq, <, >, \leq, \geq, =_d, \subset, \subseteq, \subset, \supseteq\}$ be any relationships defined on the domain of A and B. ($=_d$ denotes the equality restricted to the data components).

Thus the following are *atomic value conditions*: (i) $A\theta B$; (ii) $A = c$, for any constant c from the domain of A; (iii)$A =_{cd} \{c_1, \ldots, c_k\}$, where and each c_j is a constant value of the same type as A's domain datum component; (iv) $(u, \mu_u) \in A$, where (u, μ_u) is a fuzzy element compatible with $dom(A)$ (v) $A\ relop_c\ B$, being *relop* an operator defined on the universe of the domains of A and B.

Definition 4.10 (value condition) Every *atomic value condition* is a *value condition*.

Let vc_1 and vc_2 be two value conditions. Then so are $vc_1 \wedge vc_2$, $vc_1 \vee vc_2$, and $\neg vc_1$ (being $\wedge$, $\vee$, and $\neg$ the standard logical connectives.)

Definition 4.11 (Satisfaction of a selection condition) Let $R(X)$ be a relation schema, r be a relation instance of R, and t be a tuple in r, A and B be attributes in X, and c be a constant. Let χ be any selection condition on A, B and c.

The satisfaction of χ by t ($t \models \chi$) is defined by induction on the structure of χ, as follows;

- $t \models A\theta B$ iff $t[A]\theta t[B]$, for any $\theta \in \{=, \neq, <, >, \leq, \geq, \subset, \subseteq, \subset, \supseteq\}$
- $t \models A =_d B$ iff for every pair $\langle d_i^A, \mu_i^A \rangle$ in $t[A]$ there is a pair $\langle d_i^B, \mu_i^B \rangle$ in $t[B]$ such that $d_i^A = d_i^B$, and viceversa.

- $t \models A\theta c$ iff $t[A]\theta c$, for any $\theta \in \{=, \neq, <, >, \leq, \geq, \subset, \subseteq, \subset, \supseteq\}$
- $t \models A =_d c$ iff for every pair $\langle d_i^A, \mu_i^A \rangle$ in $t[A]$ there is a pair $\langle d_i^c, \mu_i^c \rangle$ in c such that $d_i^A = d_i^c$, and viceversa.
- $t \models A =_{cd} \{c_1, \ldots, c_k\}$ iff $t[A] = \{\langle c_1, \mu_1 \rangle, \ldots, \langle c_k, \mu_k \rangle\}$, no matter what the membership degrees are.
- $t \models \langle u, \mu_u \rangle \in A$ iff $\langle u, \mu_u \rangle \in t[A]$
- $t \models A\ relop_c\ B$ iff $\exists \langle d^A, \mu^A \rangle \in t[A]$, $\exists \langle d^B, \mu^B \rangle \in t[B]$, and the relation $d^A\ relop\ d^B$ holds (no matter what the membership degrees are)
- $t \models \chi_1 \wedge \chi_2$ iff $t \models \chi_1$ and $t \models \chi_2$.
- $t \models \chi_1 \vee \chi_2$ iff either $t \models \chi_1$, or $t \models \chi_2$.
- $t \models \neg\chi$ iff it is not the case that $t \models \chi$

According to the above definitions, selection conditions may be expressed to check attribute values (which are sets), to check elements belonging to attribute values (which are pairs of data and membership degrees), as well as to test single data values within pairs in attribute values. For example, the condition $\theta_1 \equiv Color = \{\langle red, 0.78 \rangle, \langle grey, 0.80 \rangle\}$ checks the whole value of the attribute *Color*. The first tuple of the above relation r_1 satisfies θ_1, since its value for *Color* coincides with the constant set appearing in the selection condition.

The same tuple does not satisfy the condition $\theta_2 \equiv Color = \{\langle red, 0.5 \rangle, \langle grey, 0.50 \rangle\}$, since the constant set appearing in θ_2 differs from the attribute value in the tuple because of the membership degrees.

But the same tuple satisfies $\theta_3 \equiv Color =_d \{\langle red, 0.5 \rangle, \langle grey, 0.50 \rangle\}$, because in this case the membership degrees are not influent.

If we are interested in tuples that contain the pair $\langle red, 0.5 \rangle$ in the set *Color* , we want those tuples that satisfy the condition $\theta_4 \equiv \langle red, 0.5 \rangle \in Color$.

If we are interested in tuples whose attribute *Color* includes an element with data value *red*, no matter what the membership degree is, we look for tuples that satisfy the condition $\theta_5 \equiv Color =_c red$.

4.2.6 Selection

Definition 4.12 (Selection) Let r be an instance of the relation schema $R(X)$, and χ be any value condition involving attributes from X.

The selection $\underline{\sigma}_\chi(r)$ is the set of tuples t in r satisfying the condition χ:

$$\underline{\sigma}_\chi(r) = \{t \in r \mid t \models \chi\}$$

Example 4.4 Given the following relation r, the query "find the tuples with color = (red, grey)", can be expressed as

$$\underline{\sigma}_{Color =_{cd} \{red, grey\}}(r)$$

File	Color	Shape	Content
Im1.gif	{ ⟨red, 0.78 ⟩, ⟨grey, 0.80 ⟩}	{⟨circle, 0.79 ⟩, ⟨ellipse, 0.89 ⟩}	{ ⟨Sun in the dark, 0.7⟩}
Im2.gif	{⟨blue, 0.89 ⟩, ⟨black, 0.90 ⟩}	{⟨rectangle,0.94 ⟩, ⟨square, 0.90 ⟩}	{⟨Sea, 0.8 ⟩}

The selection produces as its result the following relation

File	Color	Shape	Content
Im1.gif	{⟨red, 0.78 ⟩, ⟨grey, 0.80 ⟩}	{⟨circle, 0.79 ⟩, ⟨ellipse, 0.89 ⟩}	{ ⟨Sun in the dark, 0.7 ⟩}

That is, the result is the relation containing only one tuple, concerning the image file Im1.gif with confidence 0.78 for red and 0.80 for grey.

4.2.7 Projection

Definition 4.13 (projection) Let r be a relation with schema $R(X)$, and let $A \subseteq X$.

$$\underline{\pi}_A(r) = K(\{t[A] \mid t \in r\})$$

As usual, in the above definition $t[A]$ denotes the sub-tuple of t, obtained by dropping out from t all the values of attributes that do not belong to the set A.

4.2.8 NF^2 Join Operators

Definition 4.14 (Natural join) Let r_1 and r_2 be two relation instances with schema $R_1(X_1)$ and $R_2(X_2)$ respectively. Let $X = X_1 \cap X_2$. The natural join $r = r_1 \underline{\otimes} r_2$ is the relation with schema $R = X_1 \cup X_2$, defined as follows

$r = \{t = \langle t_y, t_x, t_z \rangle \mid Y = R_1 \setminus X, Z = R_2 \setminus X, \exists t_1 \in r_1, t_2 \in r_2$ such that $t_1[Y] = t_y, t_2[Z] = t_z, t_1[X] =_{di} t_2[X], t_x = t_1[X] \underline{\bigwedge} t_2[Y]\}$

NF^2 natural join is an extension of standard relational natural join, that replaces the equality requirement for common attributes with the data identical requirement, and joins tuples that are data identical on the common attributes by associating to those data the membership grade resulting from the fuzzy *and* of the joined ones.

¿From the above definition, if the schemas of the given relations are disjoint, the natural join degenerates in the cartesian product.

Example 4.5 Consider the following two NF^2 relations r_1 and r_2.

r_1:

File	Color
Im1.gif	{⟨red, 0.78 ⟩, ⟨grey, 0.80 ⟩}
Im2.gif	{ ⟨blue, 0.89 ⟩, ⟨black, 0.90 ⟩}

r_2:

Color	Shape
{⟨red, 0.6 ⟩, ⟨grey, 0.88 ⟩}	{ ⟨circle, 0.79 ⟩, ⟨ellipse, 0.89 ⟩}
{ ⟨blue, 0.89 ⟩, ⟨black, 0.99 ⟩}	{⟨rectangular. 0.94 ⟩, ⟨square, 0.90 ⟩}

The natural join $r = r_1 \underline{\otimes} r_2$ is shown in the following table:

File	Color	Shape
Im1.gif	{⟨red, 0.6 ⟩, ⟨grey, 0.80 ⟩}	{⟨circle, 0.79 ⟩, ⟨ellipse, 0.89 ⟩}
Im2.gif	{⟨blue, 0.89 ⟩, ⟨black, 0.90 ⟩}	{⟨rectangular, 0.94 ⟩, ⟨square, 0.90 ⟩}

As a useful, derived operator, we also define the Theta-Join operator as follows.

Definition 4.15 (θ-join) Let r_1 and r_2 be two relation instances, with disjoint schemas $R_1(X_1)$ and $R_2(X_2)$ respectively. Let θ be a selection condition involving attributes from $X_1 \cup X_2$.

The θ-join $r = r_1 \underline{\otimes}_\theta r_2$ is the relation with schema $R(X_1 \cup X_2)$, defined as follows

$$r = r_1 \underline{\otimes}_\theta r_2 = \underline{\sigma}_\theta(r_1 \underline{\otimes} r_2)$$

To conclude this section, we introduce some simple examples of use of our algebra. Consider the following relations:

Image Catalogue: r_2

File	Color	Shape	Content
Im1.gif	{(red,0.78), (grey, 0.80)}	{(circle,0.79), (ellipse,0.89) }	{(Sun in the dark,0.7) }
Im2.gif	{(blue, 0.89),(black, 0.90)}	{(rectangle, 0.94), (square, 0.9)}	{(Sea, 0.8)}
Im3.gif	{(blue, 0.6), (yellow, 0.4)}	{(rectangle, 0.9), (square, 0.5)}	{(Sea, 0.8)}

Colors Significance: r_1

ColorType	Sensation
red	⟨hot, 0.8 ⟩
yellow	⟨hot, 0.6 ⟩
blue	cold
white	⟨clearness,0.7 ⟩
{⟨blue,0.827 ⟩, ⟨green, 0823 ⟩}	⟨calm, 0.8 ⟩

The following are examples of queries in our algebra.

Query 4.1 *Find the name of the images having a* yellow *color with membership grade = 0.4, and representing* sea *with membership grade = 0.8.*

$$\underline{\pi}_{File}(\underline{\sigma}_{\langle yellow,0.4\rangle \in Color \wedge \langle sea,0.8\rangle \in Content}(r_1))$$

Query 4.2 *Find the name of the images having* hot *colors .*

$$\underline{\pi}_{File}(\underline{\sigma}_{Sensation=_{cd}hot}(r_1 \underline{\otimes}_{color=_{cd}ColorType} r_2))$$

Query 4.3 *Find the name of the images having* hot *colors and* Content= *sea with membership = 0.8 .*

$$\underline{\pi}_{File}(\underline{\sigma}_{Sensation=_{cd}hot \wedge \langle sea,0.8\rangle \in Content}(r_1 \underline{\otimes}_{color=_{cd}ColorType} r_2))$$

5 Concluding Remarks

Uncertainties and imprecision management in databases has been an important topic of research for many years, yet to date there is not a widely accepted model.

Several extensions of relational databases have been proposed to deal with uncertain or imprecise information [5], [6], [7], [8], [9]. Relational model may be extended in many ways, depending on "where" fuzziness is introduced and "what" one means with uncertainty. In most papers in the literature, the membership is an attribute expressed by single numerical values or by an interval contained in $[0, 1]$. These values are calculated by using a possibility measure in query processing. [13], [6]. In [14] fuzziness in attribute values and mutual relationships between domain elements are taken into account, and dealt with by using possibility distributions and closeness relations respectively. [13] proposes a generalized possibility fuzzy-relational model, in which a database schema is constructed from a set of pairs of an attribute and its membership values, generalizing fuzzy relational schemas introduced in GEFRED [15]. We focused on a first attempt to create such a model for image database purposes, by introducing:

- a theory based on NF^2 relational model and fuzzy logic;
- new algebraic operators in order to take into account the fuzziness implicitly related to image descriptors, both at low level and at high level;

In order to test the validity of our model, we have built an Image Data Base containing about 1000 images picked out from the Internet and from several commercial archives.
In order to facilitate the image retrieval, each image has been processed by a vision system and by humans, and a complex catalogue containing semantic description of the pictures has been produced. The catalogue has been stored in a prototype system which implements the proposed algebra.
The system contains approximately 1000 lines of PL/SQL code running under Windows 2000 platform. The implementation consists of the following major parts:

- A graphical user interface for expressing fuzzy algebraic queries.
- An object relational database system in which the catalogue is stored.
- A query processing stage that implements the various algebraic operators described here: queries expressed in our algebra are converted into queries over the relational database.

The work reported in this paper, is being extended in several directions:

- a study of several fuzzy membership for a more suitable modelling of image database purposes;
- an integration of the system with computer vision environments for content based image retrieval.

References

[1] A. Yoshitaka and T. Ichikawa. A survey on content based retrieval for multimedia databases. *IEEE Trans. on Knowledge and Data Engineering*, 11(1):81–93, 1999.

[2] P. Aigrain, H.J. Zhang, and D. Petkovic. Content based representation and retrieval of visual media: a state of the art review. *Multimedia Tools and Apllication*, 3:179–202, 1996.

[3] S. Santini and R. Jain. Similarity measures. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 21(9):871–883, 1999.

[4] K. S. Candan and W.S. Li. On similarity measures for multimedia database application. Technical report, C & C Research Laboratories, NEC USA, Inc., 2001.

[5] K.V.S.V.N. Raju and A.K Majumdar. Fuzzy relational dependencies and lossless join decomposition of fuzzy relational database systems. *ACM Trans. Database Syst.*, 13,(2):129–166, 1988.

[6] Y. Takahashi. Fuzzy database query languages and their relational completeness theorem. *IEEE Trans. on Kowledge and data engineering*, 5(1):122–125, 1993.

[7] F. E. Petry. *Fuzzy Databases: Principles and Applications.* Kluwer Academic Publishers, Norwell, MA, 1996.

[8] A. Yazici, B.P. Buckles, and F.E. Petrym. Handling complex and uncertain information in the exifo and nf^2. *IEEE Transactions on Fuzzy Sustems*, 7(1):659–676,, 1999.

[9] Q. Yang, C. Zhang, W. adn Liu, J. Wu, C. Nakajima, and H. Rishe. Efficient processing of nested fuzzy sql queries in a fuzzy database. *IEEE Trans. on Kowledge and data engineering*, 13(6):884–901, 2001.

[10] V.S. Subrahmanian. *Principles of Multimedia Database Systems.* Morgan Kaufmann Publishers, 1998.

[11] A. Chianese, A. Picariello, and L. Sansone. Query by examples in image database using a fuzzy knowledge base. In *Proc. of SCI 2001*, volume 16, pages 362–367, 2001.

[12] L. Zadeh. Quantitative fuzzy semantics. *Information Systems*, 3:159–176, 1971.

[13] M. Nakata. Eliminitation of semantic ambiguity in fuzzy relational models. In *Proc. of IEEE ISUMA-NAFIPS 95*, pages 643–648, 1995.

[14] G.Q. Chen, E.E. Kerre, and J. Vamdenbulcke. *Studies in Fuzziness : fuzzy sets and possibility theory in data base management systems, Bosc, P. and Kacprzyk, J (ed)*, chapter An overview of fuzzy data models, pages 23–41. Physica. Springer Verlag, Germany, 1995.

[15] J.M. et al. Medina. Gefred: A generalized model of fuzzy relational databases. *Information Systems*, pages 87–109, 1976.