Managing Uncertanties in Image Databases

Managing Uncertainty in Image Databases

Abstract

In this chapter, we discuss functionalities of multimedia databases, which are not present in traditional databases, but are needed when dealing with multimedia information. Multimedia data are inherently subjective: for example, the association of a meaning and the corresponding content description to an image as well as the evaluation of the difference between two images or two pieces of music usually depend on the user who is involved in the evaluation process. Subjective information usually needs to be combined with objective information, such as image color histograms or sound frequencies, obtained through (generally imprecise) data analysis processes. Therefore, the inherently *fuzzy* nature of multimedia data, both at subjective and at objective levels, may lead to multiple, possibly inconsistent, interpretations of data. We propose the FNF^2 data model, a Non First Normal Form extension of the relational model, which takes into account subjectivity and fuzziness, while being intuitive and enabling user friendly information access and manipulation mechanisms.

INTRODUCTION

In the *multimedia age,* characterized by new emergent kinds of data, such as images, sounds, texts, and video objects, the need for information storage and retrieval requirements cannot be satisfied by simply relying on traditional databases. The various properties of these objects cannot be properly captured by the relational or object oriented models. Therefore, Multimedia Databases have to provide new functionalities, depending on the type of -possibly heterogeneous- multimedia data being stored. Within this context, new challenges, from the problems related to data representation to the challenges related to the indexing and retrieval of such complex information have to be addressed.

In this chapter, we discuss functionalities of multimedia databases which are not present in traditional databases, but are needed when dealing with multimedia information. Multimedia data are inherently subjective: for example, the association of a meaning and the corresponding content description to an image as well as the evaluation of the difference between two images or two pieces of music usually depend on the user who is involved in the evaluation process. Furthermore, such subjective information usually needs to be combined with objective information, such as image color histograms or sound frequencies, obtained through data analysis. Data analysis processes are generally imprecise. Therefore, the inherently *fuzzy* nature of multimedia data, both at subjective and at objective levels, may lead to multiple, possibly inconsistent, interpretations of data. Thus, providing a data model which can take into account subjectivity and fuzziness, while being intuitive and enabling user friendly information access and manipulation mechanisms, is a challenging goal.

Although most of the results presented here also apply to different multimedia information management scenarios, in this chapter we specifically focus on image data, which illustrate the common subjectivity and fuzziness aspects well.
To properly store a collection of images in a database, the system must offer appropriate capabilities to explore the relationships among the different images, to recognize the relevant image features, and to provide methods and techniques to express those relationships and features, and to query on them. As opposed to the classical relational data model, in which queries are usually posed textually (or through some visual interface, which does not increase the expressive power of the textual format in which queries are in fact automatically translated) in image databases queries are usually expressed in *non textual forms*. This is the case, for example, of *Query By Example* or *Query by*

Content forms, in which a query may include an image as part of it and the returned result does not rely on a crisp evaluation process, but relies on a notion of *similarity* between the query and the images in the database. In particular, returned images are associated with a degree of satisfaction of the query, which represents to which extent the image can be considered similar to the given one, according to the chosen notion of similarity.

Fuzziness and uncertainty related to image query processing cannot be directly represented in the relational data model. Therefore, several approaches (Takashi & al., 1993, Raju & al., 1988 and Yang & al., 2001) have been proposed to extend the relational data model to appropriately include these aspects. In particular, (Zaniolo & al., 1997) extend the relational model to incorporate uncertainty at the tuple level, while describing different approaches at the attribute level. In the *tuple-level approaches* the schema of the relations can also include attributes specifically representing uncertainty values. Thus, each tuple may contain one or more uncertainty values for the corresponding attributes, each one representing the fuzziness associated to one interpretation for the data values stored in the other attributes of the tuple. The uncertainty attributes have usually real values or are expressed in terms of intervals on real numbers. In the *attribute-level approaches*, instead of associating a value representing the uncertainty of the data to the tuple as a whole, a degree of uncertainty is associated directly to every single attribute value. In image databases, in which images are represented in terms of their various feature values, extracted from the image using appropriate image processing and analysis processes, attribute level approaches are more applicable: it is easier to store and maintain detailed information about the different relevant aspects of a given image using an attribute-based approach instead of associating a unique, global value to the overall image tuple.

In the next section we present the background on modelling and accessing image databases and the state of the art on dealing with fuzzy information in image databases. After discussing the problem of image representation, we present related work in the area of image retrieval, and comment on relevant fuzzy models for image databases. The following section describes our FNF^2 data model and its suitability for image retrieval. Concluding remarks are given in the final section.

BACKGROUND AND RELATED WORK

Problems related to modelling and accessing image database have been addressed in various scientific communities, from different perspectives. Some aspects, such as the issues related to feature extraction and representation, have been specifically studied in both Computer Vision and Image Processing communities (Del Bimbo, 1998), while aspects related to storage, indexing, and query processing have received great interest in the Database community (Grosky, 1997, Subrahmanian, 1997, Krishnapuram & al., 2005 and Buche & al., 2005).

Given the high dimensionality of the problem, different authors and research groups mainly concentrate on some specific aspects, while making simplifying assumptions on the other aspects. For example, a simplifying assumption widely adopted in the database community is that textual description and representation of images are available; that is, data are *annotated*. From a different perspective, most researchers from the computer vision and the image processing communities make simplifying hypotheses about data modelling and data retrieval methods and work in the context of image repositories or image directories, instead of image databases.

A complete image database systems should benefit from the integration of the methods and results from the different communities. In this section, we introduce the main components of image database systems to motivate the role of fuzziness in image databases and we present alternative approaches for fuzzy data modelling for image retrieval (Raju & al., 1988, Takashi & al., 1993, Petri 1996,Yazici & al., 1999, Yang & al., 2001, Subrahmanian, 1997).

Image representation

Images are represented in the database as collections of low level features. To detect the prominent features in the stored images, image databases use *feature extraction modules*, implemented on the basis of computer vision techniques. The most frequently used image features are *colors*, *textures*, *shapes*, and *spatial descriptors* (Del Bimbo, 1998).

Colors are usually described through *color histograms* (Swan & al., 1991), which associate a *bin* to each distinct color. Each bin contains those pixels which have the corresponding color in the image, thus the size of the bin reflects how much that color is present in the image. Color histograms capture the color distribution in a given image from a quantitative point of view, but they are sufficient neither to describe spatial color distribution, nor to handle color correlations in the images. These aspects of color, and more specifically spatial relationships between colored pixels in a certain image region, can be better represented by analyzing textures and shapes (Wang Jing & al., 2003, Smeulders & al., 2000, and Del Bimbo, 1998).

According to the *feature contrast model* (Wei Jiang, 2004), a visual stimulus may be characterized as a *set* of binary features, i.e. a stimulus is represented by means of the set of feature properties that it satisfies. For every feature, a set of possible values is fixed. The representation of any visual object is a set of binary values, which denote the fact that the corresponding feature can be considered as having or not having that particular value in the given object. Equivalently, the feature set for a given stimulus can be characterized as a set of *logic predicates* that the stimulus does satisfy (Santini & al., 2001).

On the other hand, binary (Boolean) logic is not always suitable for modeling image features. With the goal of taking into account the noise of visual stimuli perception and representation, fuzzy theory is recognized as a natural modelling framework. In this framework, any image *I* is characterized in terms of a number of fuzzy measurements ν_i on the image features and properties. For example, let us assume that the shape of an object in an image is the interesting feature to be represented. It can be the case that different observers (or different shape-extraction algorithms) provide different shape characterizations for the same object. One observer might say that a given shape is "highly" oval or that it is "almost" rectangular. Expressions such as *highly*, or *almost*, recall this notion of fuzziness, which is implicitly embedded in the similarity evaluation of visual stimuli.

Each image can be characterized in terms of its physical representation and the information extracted from the image, or provided by the user. More formally, let *I* be the set of all the images in the given image database. Given a set *of classes of membership M* and a set of possible *memberships values P*, the analysis process is a function computed by *an* – either a human or a computer vision - *system V,* which associates to any feature in *I* a set of elements from *M*, each one with its specific grade of membership $p \in P$, i.e.:

$$V : I \to (M \to P)$$

If $V(i) = \{(c_1, p_1),...,(c_k, p_k)\}$, then for any class c_i, p_i is a measure of the membership of i to c_i. The function *V* abstracts the process of analyzing the physical representations of the images, to produce a description for each of them with an associated grade of *uncertainty*. In the following discussion and examples, when representing the set *V(i),* we do not list the pairs whose second components (i.e. grades) are 0; these correspond to feature properties which are classified as missing in the considered object. Thus, we read the set $V(i) = \{(c_1, p_1),...,(c_k, p_k)\}$ as the specification of all the properties that are satisfied by the object *i*, with their degree of satisfaction. For retrieval, given a set of low-level features, a notion of *similarity/dissimilarity*, possibly based on distance metrics, is defined over the features. The definition of the similarity between two stimuli depends on the formalisms in which the stimuli are represented. The retrieval process is performed using this dis/similarity concept in the corresponding feature space. Naturally, similarity and dissimilarity are opposite of each other. In many cases, dissimilarity measures characterize the difference between a given pair of images as a distance in some suitable, mostly metric, feature space (Ashby & al., 1988, and Santini & al., 2001).

The image retrieval problem

Intuitively, the image retrieval problem is deciding whether, and to which extent, any image stored in the database matches a criterion, specified by means of a query or an example. Even when low-level features of an image are not fuzzy, image information content (needed for measuring the degree of match between a query and the image) can not always be considered as being univocal. In general, what the image represents and what is important in the image depends on the observer as well as on the context in which the image is inserted into the collection (Itti & al., 2001). Therefore, there is not a unique notion of similarity. To account for this, similarity models can be classified in a way to distinguish between *perceived similarity* and *judged similarity*. If a and b are any two stimuli, and A and B are their representations in the feature space, the perceived similarity between the two stimuli is a function $s(A;B)$ of their representations, while the judged similarity is a function of the perceived similarity, that is a function $\sigma(A,B) = g[s(A;B)]$, where g is a suitable monotonically non decreasing function (Itti & al., 2001). Here, monotonicity ensures that it cannot be the case that two images are subjectively seen as more similar than another pair, while being less similar according to perceived similarity. Specific similarity functions have been defined to capture the uncertainty related to similarity concepts (Wang, 2005).

In some cases, the image description comes from human-provided annotations. In these cases, query processing requires the evaluation of a semantic similarity between the query terms describing the target image and the annotations describing the images in the database. This task is usually done by means of pattern recognition techniques (Boccignone & al., 2005), which return high level features (as opposed to low-level features like color histograms), with associated *grades of confidence*. A grade of confidence denotes the degree of membership of the given feature to the discovered pattern.

Uncertainty-based Models For Image Databases

In the following, we present the literature focussing on the role and importance of uncertainty, necessary for a complete image database definition and implementation.

Buckles & al., (1982) were among the first authors who proposed a formal treatment of fuzziness in relational databases, in a more general setting, not specifically conceived with images and other multimedia data. In their model fuzziness is associated to data by means of linguistic terms (for example, terms like "bad" and "good" are considered in their approach) and they define a notion of *similarity* on linguistic terms. More recently, the authors classified databases in different groups, characterized as dealing with *precise* data, *imprecise* data, and *vague* data respectively (Buckles & al., 1995). The model is best applied to enterprises (or parts of enterprises) in which the linguistic sets are finite and may be extended to continuous domains. The case of precise data virtually includes all the database systems in wide-spread use. On the other hand, *imprecise data* model is the basis of the studies on uncertainty in databases. The key notion is that while only one value applies to the enterprise, the database extension may contain a set and each database object is "*surely*", "*maybe*", or "*surely not*" a response to the query (*possibility theory*). The *vague database* refers to the databases dealing with attribute values, for which it is assumed that no precise value exists. They are represented as *linguistic terms* and these terms are themselves related to each other by similarity relationships.

A natural extension of the previous works is provided by Yazici & al, (1999). This work motivates the need for non-first-normal form models to model and manage uncertainty in the relational data model. It introduces the ExIFO model for conceptual design of fuzzy databases and for logical design based on non-first-normal form logical database models. In particular, the ExIFO model deals with incompleteness, null data, and fuzziness. It represents uncertainty *at the attribute level* by means of new constructors at the conceptual and logical levels, thus defining fuzzy entity types. Raju & al, (1988) represent ambiguities in data values, as well as impreciseness in the association among them, through a fuzzy relational data model. The authors describe and formalize the treatment of the integrity constraints for a fuzzy data model: they define relational operators for fuzzy relations and investigate the applicability of fuzzy logic to capture integrity constraints. Moreover, they address the problem of lossless join decomposition of fuzzy relations for a given set of fuzzy functional dependencies.

In a collection of works edited by Zaniolo (1997), Subrahmanian introduces a foundational theory for managing uncertainty in databases and knowledge-bases in general. In particular, uncertainty is represented at the *tuple level;* that is, each tuple is extended to include one or more uncertainty attributes, representing the likelihood of the overall information associated with the tuple. Given this model, a probabilistic extension of classical relational algebra operators is also provided. These take into account the uncertainty aspects included in the relations. The model is general enough to be applied to different fuzzy knowledge management scenarios, including image databases. However, in this case, it fully relies on an annotated scenario, and does not consider the numerous low-level or intermediate level descriptions (or features) that can be independently extracted from image data by means of image processing techniques.

More recently, Atnafu (2004) proposed an integration of similarity-based queries into image DBMS. In particular, the authors propose an image data repository model, a formal algebra for content-based image operations, and several image processing and retrieval technique. Several "similarity" based operators adapted to image data are discussed within an Object Relational DBMS framework. Uncertainty is incorporated in the definition of similarity (distance) based operators: the authors consider metric space computations returning similar images based on the value of an uncertainty threshold, that can be chosen appropriately based on the peculiarities of an application. Starting from this concept, they define a content-based algebra, with selection and join operators.

Krishnapuram & al. (2004) propose a Content Based Image Retrieval System based on *fuzzy logic*, FIRST (Fuzzy Image Retrieval SysTem). In particular, the authors propose a data model based on Fuzzy Attributed Relational Graph (FARG), in which each image is represented as a graph whose nodes represent the objects and edges represent relations (e.g. spatial relations) between them. A given query is converted to a FARG and the query processing is reduced to a *sub-graph matching* problem. To reduce the NP complexity of the sub-graph matching problem, Krishnapuram also proposes indexing schemes based on a leader clustering algorithm. The underlying data structure, which extends the relational model, is more expressive than the previously existing approaches. As a trade-off, creating and maintaining such data structure is more complex.

Buche (2005) describes a relational model extended with the, so called, *multi-views fuzzy querying*. The model is specifically well suited for biological processes, but it has interesting applications to image databases as well. As an innovative contribution, fuzzy data are integrated and dealt with by referring to ontologies and semantic rules. The model captures incompleteness and impreciseness and it expresses user preferences with a fuzzy model in which an ontology is used to express fuzziness: the values of a domain are connected using the *"a kind of"* semantic link.

In the next section, we discuss an attribute level fuzzy extension of the relational model. Unlike the other approaches, presented above, this model captures the uncertainty in the description of images at different levels.

THE *FNF*^{2} MODEL

In this section, we define the *FNF*^{2} model for image databases. We first introduce motivating examples and then present a formal definition of the model.

Motivating Examples

Let us assume that we are given a database containing a collection of digital pictures and let us consider a set of queries that we may want to pose on this collection:

Query1: *Find those images which contain a high quantity of sun*-light *and also contain a mountain.*

To process this query, the system first needs a *color-based retrieval* module, to find all the images in which those colors that usually are associated to the *light* (red and white) appear, in such a quantity that the system would classify their amount as *high*. This process will involve a fuzzy description of the color content of the digital image. Secondly, the system also needs *semantic information* (extracted automatically or obtained from a repository of human annotations) about the pictures in the database, to tell whether they contain any mountains.

Figure 1: *Ves_pict.gif*, a picture of the Vesuvio Vulcan at the Sunset.

As described earlier, an automatic vision system analyzes a given image and extracts information about *colors, textures, shapes and spatial descriptors,* each associated with *a degree of certainty*. For example, the analysis of the *Ves_pict.gif* image in Figure 1, by means of an automatic vision system developed at the University of Napoli (Chianese & al., 2001), returns fuzzy feature descriptors. Multiple colors appear in the image and the system associates to each color attribute value a certainty degree: *grey* appears with certainty degree 0.6; *red* with certainty 0.75; *white* with certainty 0.7; *blue* with certainty 0.411; and *yellow* with certainty 0.417. Texture attribute is recognized as *thin* with certainty degree 0.6; *mixed thin and coarse* with certainty 0.9; *net* with certainty 0.571; and *crisp* with certainty 0.322. The recognized semantic attribute is *mountain* with a confidence 0.871.

Similarly, the automatic vision system analyzes the image in Figure 2 (the picture of a coast landscape, in which blue is the dominant color), Figure 3 (the picture of a mountain covered with snow, in which white is the dominant color), and Figure 4 (a sunset in a saguaro desert, in which purple and burgundy dominate).

Figure 2: *Coast_pict.gif*, a coast Landscape.

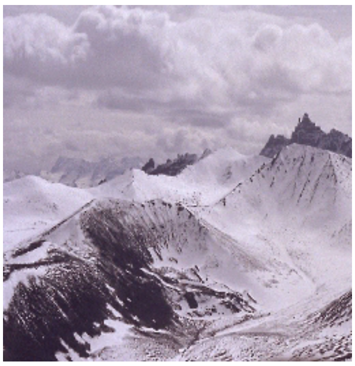

Figure 3: *Snow_pict.gif*, a snow-mountain picture

. Figure 4: *Sunset_pic.gif*, a saguaro-mountain picture.

The information that the automatic system extracts is summarized in Table 1.

File	**Color**	**Texture**	**Content**
Ves_pict.gif	<grey,0.600>, <red,0.750>, <blue,0.750>, <yellow,0.417>, <white, 0.700>	<thin,0.600>, <thinCoar,0.900>, <crisp,0.322>, <net,0.571>	< mountain,0.871>
Coast_pict.gif	<blue,0.827>, <beige,0.765> <green,0.816>	<thin, 0.704>	<SeaCoastPicture,0.939>
Snow_pict.gif	<white,0.939>, <gray, 0.569>	<mixt_texture, 0.674	<SnowMountain,0.918>
Sunset_pict.gif	<purple,0.866> <yellow,0.786> <burgundy,0.856>	<thin,0.704> <crisp,0.346>	<Sunset, 0.839>

Table 1. The fuzzy attribute values extracted from images in figure 1,2,3 and 4.

This automatically extracted information has to be properly stored in the database. Furthermore, to solve Query1, the fuzzy values may need to be retrieved and combined during the query answering process.

Query2: *Find those images that are similar to Ves_pict.jpg.*

This second type of query involves a query by example; the images in the database must be retrieved based on their *similarities* to the given example. The major difference from Query1is that, in this case, the aspects of interest in the image are not explicitly enumerated, but the relevant information on which similarity is to be checked has to be automatically detected. Knowing that our system has extracted from *Ves_pict.gif* color, texture, and semantic information, as described above, the query can be rephrased as: ``*find all the images that have red and white as predominant colors, that have a thin texture and that contain a mountain*''.

To move towards a model that can handle queries like Query1 and Query2, we have to address two main issues. First, we have to deal with uncertainty in data at the attribute level (e.g., uncertainties associated with colors, textures, contents and so on). Secondly, we have to develop mechanisms to process the given queries when uncertainties are associated to the available data at the attribute level.

The *FNF*^{2} Data Model

The fuzzy relational model we are proposing is an extension of the standard relational model. In the proposed extension, the considered attribute value domains are fuzzy. Fuzzy data models can be interpreted as extensions of traditional data models using fuzzy set theory (Zadeh, 1971) and the possibility theory (Zadeh, 1978). We first define a fuzzy tuple.

***FNF*^{2} fuzzy tuple.** Let $D_1, \ldots, D_n$ be n domains. A fuzzy n-tuple is any element of the cartesian product $2^{D_1} \times \ldots \times 2^{D_n}$, where 2^{D_i} is the fuzzy powerset of D_i, that is, the set of all fuzzy subsets of D_i.

According to the definition, any n-tuple is a sequence $< vs_1, ..., vs_n >$, where each vs_i is a set of elements, of the form $<v_j, f_j>$. Here, v_j is a value from the corresponding fuzzy domain D_i, and f_j is its corresponding fuzzy membership value.

For the sake of simplicity and readability, we denote those attribute value sets vs_i that are singletons and where the only member of the set has fuzzy membership degree equal to *1* (which represents full, certain membership), by means of the only domain value. This is the case, for example, for the "Ves_pict.gif", "Coast_pict.gif", "Snow_pict.gif", and "Sunset_pict.gif" values of the **File** field in the data presented in Table 1.

As we have already mentioned, we also consider, as a special case, the presence of a membership degree of *0*. This represents the certain non-membership. In our model, the presence of a pairs $< v, 0>$ in an attribute value does not provide any information in addition to what we could have if we did remove that pair from the set, since domain values that do not appear in an attribute value are implicitly associated to the membership degree *0*. For example, the attribute values $\{< red, 0.5 >\}$, and $\{<red, 0.5 >, < green, 0.0>\}$, provide the same information. Thus, we assume that our attribute value sets do not contain any such pair. Since the fuzzy values are derived from data returned by some automatic vision systems, and since data provided by visual systems are only non zero values (systems only give information about feature values they find in the considered image) this is also not a restriction from an implementation point of view.

***FNF*^{2} fuzzy relation schema**: A *fuzzy relation schema* is used to associate attribute names to the domains. In the following, we also use the term *fuzzy attribute* to denote an attribute A whose domain $dom(A)$ is a fuzzy set.

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

***FNF*^2 fuzzy relation**: A fuzzy relation is an instance of a fuzzy relation schema; that is, a fuzzy relation is a set of fuzzy tuples, as stated in the following definition.

Let $R(\{A_1, A_2, \ldots, A_n\})$ be a relational schema. A *fuzzy relat*ion, defined over R, is a set of fuzzy tuples $t = < vs_1, \ldots, vs_n >$ such that each vs_i is a fuzzy subset of $dom(A_i)$.

An example of a fuzzy relation has already been given in Table 1. The schema of this example relation has four attributes, **File**, **Color**, **Texture** and **Content**. Each attribute value is a set of <domain_value, fuzzy_value> pairs.

***Manipulation of FNF*^2 Fuzzy Relations**

FNF^2 *relations* are accessed and manipulated by means of a corresponding *FNF*^{2} *algebra.* The usual set theoretic operators can be extended to fuzzy sets in different ways, depending on the specific semantics associated with the fuzzy logical connectives. In fact, several semantics have been proposed for fuzzy logical connectives *and* ($\wedge$) and *or* ($\vee$), but it has been proved that the *min-max* semantics is the only semantics for conjunction and disjunction that preserves logical equivalences and satisfies the *idempotency* property (Yager, 1982). This motivates our choice to adopt the following definitions for fuzzy logical operators.

-*Fuzzy intersection (and):* $A \wedge B = \{< u, min(\mu_A(u), \mu_B(u))> \mid u \in U\}$

-*Fuzzy union (or):* $A \vee B = \{<u, max(\mu_A(u), \mu_B(u)) > \mid u \in U\}$

- *Fuzzy complementation (not):* $\neg A = \{<u, 1 - \mu_A(u)) > \mid u \in U\}$

In these definitions, U represents the universe, i.e., the domain against which fuzzy memberships is evaluated.

In this chapter, we do not present the details of the algebra. Instead, we concentrate on some primary aspects on which the *FNF*^{2} algebra is founded: (i) tuple comparison, which is at the basis of all the operations that require some logical predicate on tuples to be evaluated; (ii) tuple combination, on which most algebraic operations rely, to put information together after identifying the relevant tuples to be combined, and (iii) tuple ordering, the basis of all the operations having to do with ranking based on a given criterion.

Tuple comparison: Tuple comparison is at the basis of several algebraic operations, including the set oriented ones. In FNF^2 data model**,** attribute values are sets of pairs. Depending on the query, comparison can either involve the complete information represented by the tuples or can be restricted to a particular component. Indeed, it might be necessary to recognize and combine those tuples which contain, for every attribute in their schema, the same sets of values: these values might differ at most because of membership degrees.

This notion is formalized in the definition of *data identical* tuples: Let $R(\{A_1, A_2, \ldots, A_n\})$ be a relation schema, and r be an instance of R. Let $t_1 = <vs_1^1, \ldots, vs_n^1>$ and $t_2 = <vs_1^2, \ldots, vs_n^2>$ denote two tuples of r. t_1 and t_2 are *data identical* iff for every index i, it holds that $data_1 = \{u \mid <u, \mu_{dom(A_i)}(u)> \in vs_i^1\} = data_2 = \{u \mid <u, \mu_{dom(A_i)}(u)> \in vs_i^2\}$. As an example, consider the following *tuples,* t_1 and t_2, which are data identical.

	File	**Color**	**Texture**	**Content**
t1	Ves_pict.gif	<black,0.7>, <red,0.815>, <beige,0.414>, <brown,0.311>, <white, 0.628>	<thin,0.715>, <mixed,0.715>, <net,0.511>, <crisp, 0.121>	< human, 0.95>
t2	Ves_pict.gif	<black,0.75>, <red,0.8>, <beige,0.6>, <brown,0.3>, <white, 0.7>	<thin,0.6>, <mixed,0.715>, <net,0.7>, <crisp,0.1>	< human, 0.9>

As the comparison operator, depending on the intended meaning of the query processing operations, either more traditional *equality* or data identicalness can be used. Consider for example the *set union* operation. Union merges the tuples appearing in the given sets and removes duplicates. If equality is used as the comparison operator, the resulting relation can contain multiple instances of data identical tuples, which are not eliminated because of the different degrees of certainty of their attribute values. This could be the intended meaning if the operation has the goal of keeping track of all the data retrieved by the feature extraction modules. On the other hand, for applications that aim at returning a single certainty information for every feature data value in the result, the existence of data identical tuples can not be acceptable. Therefore data identical tuples should be combined. Similar considerations apply to the other set operators as well. In particular, *intersection* returns the tuples that are in common. Therefore, it depends on the meaning associated to "being in common", and on the treatment chosen for data identical (but distinct) tuples occurring in the relations being intersected. Similarly, set difference returns the tuples appearing in the first relation and not appearing in the second one. Therefore, it also depends on the comparison operator chosen and on the treatment selected for data identical tuples.

Tuple combination: Different alternative combination functions can be defined, depending on the intended meaning of the union operation. In particular, if we want to adopt a *skeptical treatment* towards the combination results, we use *conjunction* as the combination function for data identical tuples, while an *optimistic treatment* of combination would make disjunction preferable:

- *Optimistic combination* ($\oplus_o$): Let $t_1 = <vs_1^1, ..., vs_n^1>$ and $t_2 = <vs_1^2, ..., vs_n^2>$ be two data identical tuples. $t_1 \oplus_o t_2 = <vs_1, ..., vs_n>$,where for each i, $vs_i = \{< u, \mu_1(u) > \vee < u, \mu_2(u) > \mid < u, \mu_1(u) > \in vs_i^1$ and $< u, \mu_2(u) > \in vs_i^2 \}$.

- The *skeptical combination* ($\oplus_s$) is defined in the analogous way, by applying the *fuzzy and* operator on the values instead of the *fuzzy or*: $t_1 \oplus_s t_2 = <vs_1, ..., vs_n>$, where for each i, $vs_i = \{< u, \mu_1(u) > \wedge < u, \mu_2(u) > \mid < u, \mu_1(u) > \in vs_i^1$ and $< u, \mu_2(u) > \in vs_i^2 \}$.

As an example, we consider the two data identical tuples, t_1 and *t2,* introduced above. Their optimistic and skeptical combinations are the following.

	File	**Color**	**Texture**	**Content**
$t_1 \oplus_o t_2$	Ves_pict.gif	<black, 0.75>, <red, 0.815>, <beige, 0.6>, <brown, 0.311>, <white, 0.7>	<thin, 0.715>, <mixed, 0.715>, <net, 0.7>, <crisp, 0.121>	< human, 0.95>
$t_1 \oplus_s t_2$	Ves_pict.gif	<black, 0.7>, <red, 0.8>, <beige, 0.414>, <brown, 0.3>, <white, 0.628>	<thin, 0.6>, <mixed, 0.715>, <net, 0.511>, <crisp, 0.1>	< human, 0.9>

Both optimistic combination and skeptical combination are commutative and associative. They inherit the properties from commutativity and associativity of *fuzzy and* and *fuzzy or*, and from the standard set union. Therefore, combination operators can be straightforwardly extended to sets of data identical tuples.

The comparison and combination operators defined so far are mainly needed for set oriented operations. To move towards a complete algebra on the fuzzy data model, we need to be able to express more general conditions on the tuple and on their components. The fuzzy attribute based model is very flexible and allows us to express may interesting relationships and conditions. A sample is provided next.

Selection conditions on tuples. Selection conditions are either basic or complex. Basic conditions are those that are *atomic*. Complex conditions are defined by means of conjunction, disjunction, and negation of atomic conditions. The FNF^2 model admits different classes of conditions. We list some illustrative examples as starting points to help define selection conditions.

Comparing with constants. These conditions are defined on individual attribute values (i.e., a single set of value/degree pairs).

1 - *Fuzzy pair membership condition* tests whether a given pair belongs to a given set . This condition is generally used to retrieve images with a specific feature value. For example, in any query like "*Find all images in the relation* r *in which the color* red *appears with certainty* 0.8". Syntactically, this selection condition can be expressed as <red,0.8> $\in$ r.Color.

2 - *Fuzzy pair membership condition, restricted to data-value,* tests whether a pair with a specific data value belongs to a given attribute value. For example, this condition allows the retrieval of images in which some quantity of red is identified by the feature extraction module. This query could be expressed as <red, _> $\in$ *r.Color*, where *"_"* is a wildcard.

3 - *Fuzzy pair membership condition, restricted to certainty-value,* tests whether a pair with specific certainty value belongs to a given attribute value. For example, we might be interested in knowing if there is any color which has been recognized with certainty 0.9, in a given collection of pictures. This query can be written as <_, 0.9 > $\in$ *r.Color.*

4 - *Fuzzy pair membership condition, restricted to certainty value thresholds,* tests whether there is any pair with a certainty value above a given threshold (or, similarly, below the threshold). Conditions of this sort allow the users to retrieve those images whose data certainty values are above the threshold of interest for the specific application. For example, the condition <_, 0.5> $\in_{<=}$ *r.Color* would retrieve those images which have at least one color with a certainty value at least 0.5.

When we need to combine different tuples (as it is the case for join operations in the relational algebra), we need to express conditions that relate those attribute values relevant to the query, in the given tuples. The following is a (non exhaustive) list of conditions that can be expressed in our FNF^2 model to compare attribute values.

Comparing two attribute values *(i.e., comparing two sets of fuzzy pairs)*:

1 **-** *Equality of the sets.* This condition can be used for testing whether two tuples have exactly the same value, for a given attribute. For example, the condition $r_1.Color = r_2.Color$ could be used to retrieve those image pairs, in relations r_1 and r_2, which are described as having the same colors with the same certainty values,.

2 **-** *Equality of the two sets, restricted to the data component.* If we are interested in image pairs described as having the same colors, but we do not care about the certainty values associated to their descriptions, we can restrict the attribute value comparison to data values as follows: $r_1.Color =_d r_2.Color$.

3 **-** *Equality of the two sets, restricted to the certainty component.* This case is similar to the previous one, but applies when we are interested in comparing the degrees of certainty, without checking the corresponding data values: $r_1.Color =_c r_2.Color.$

4 **-** *Set inclusion.* If we are interested, for example, in checking whether the color descriptions of an image is included, with the same certainty degrees, in the description of another, we could express this condition as $r_1.Color \subseteq r_2.Color$

5 **-** *Set inclusion, restricted to the data component.* We can use this type of conditions, for instance, if we want to check whether all the colors which describe a given image also appear in the description of another image (possibly with different certainty values) . A restricted set inclusion type condition which expresses this requests is as follows: $r_1.Color \subseteq_d r_2.Color.$

6 **-** *Set inclusion, restricted to the certainty component. In some cases*, we might want to check whether the data about two images have been collected with the same degrees of certainty, no matter what the corresponding data values are. We would express this as $r_1.Color \subseteq_c r_2.Color.$

7 **-** *Overlapping of the two sets.* This condition allows us, for instance, to test whether two images have some color in common, described with the same certainty; that is, if their Color attributes have a non empty intersection. Syntactically, this condition would be expressed as $(r_1.Color \cap r_2.Color) \neq \emptyset.$

8 *- Overlapping of the two sets, restricted to the data component.* If we are interested in the presence of common colors, but we can ignore their degrees of certainty, we can restrict the test to the data component of the common pairs: $(r_1.Color \cap_d r_2.Color) \neq \varnothing$.

9 *- Overlapping of the two sets, restricted to the certainty component.* Analogously, if we are interested in the presence of any color described with the same certainty, we can use $(r_1.Color \cap_c r_2.Color) \neq \varnothing$.

10 *- Relative ordering of the two set.* In many cases, different value sets might result from the same data processed by different vision systems. FNF^2 model allows us to test whether data in common are more reliable (i.e., more certain) according to one set or the other. The condition $r_1.Color \leq r_2.Color$ checks, for example, if all the common the values in the Color attributes in the two tuples (one from r_1 and the other from r_2) have higher certainty values in the second than the first.

The above conditions, and other conditions that we could express in FNF^2, can be seen as atomic conditions that can be combined in a complex condition, by means of conjunction, disjunction, and negation connectives. Thus *any combination of predicates are applicable either on the data elements or on the certainty elements.* For example, FNF^2 allows us to check whether two tuples "have exactly the same color information, but the granularity of texture in the first one is finer than the second one." Naturally, for the execution of this query a partial order over granularity of texture description must be defined a priori. Thus, FNF^2 model enables a natural extension, suitable for managing image data, of the relational algebra, with both standard (both atomic and complex) selection conditions and ad hoc (atomic and complex) conditions described in this section.

EXPERIMENTAL STUDIES

The FNF^2 model is a data model on which many different systems can be implemented. In order to directly experience the expressive power of our model, and without any performance evaluation goal (since the performance of the resulting model would nott be a property of the model, but would depend on the chosen feature extraction, indexing, and clustering mechanisms), we developed a prototype system, called FIB (Fuzzy Image database).

FIB extends relational databases both in terms of the data model (fuzzy relations are used instead of standard ones), and in terms of the implemented algebraic operators, which include selection conditions of all the forms we listed in the previous section. As for the tuple combination functions, in the existing prototype only *skeptical* version is currently available.

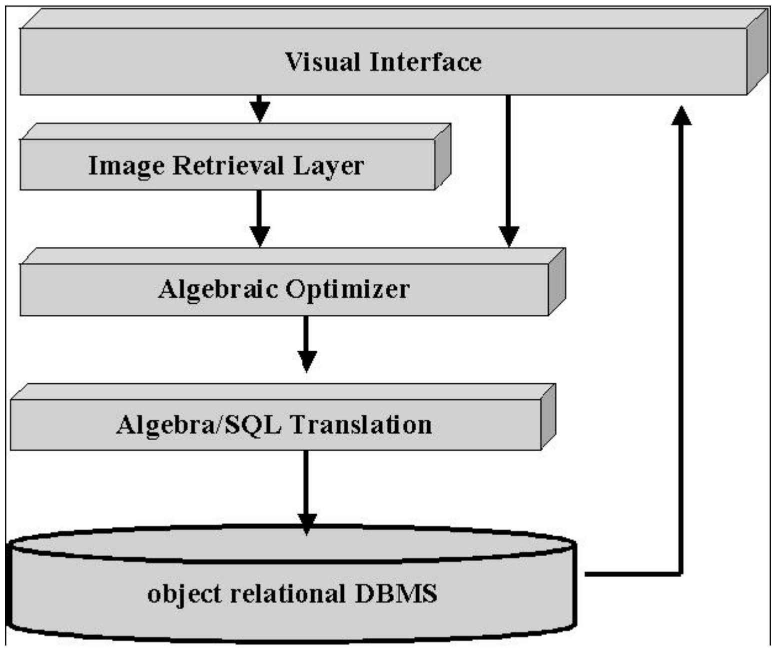

Figure 5: system architecture.

Figure 5 illustrates the architecture of the system, which consists of the following major components.

- With the goal of supporting visual, textual, as well as visual/textual queries, *a graphical user interface* is available for users' query expression. The interface code consists of approximately 7500 lines of Java code.

- An image *query processing engine* implements the various feature extraction algorithms described in (Chianese & al., 2001). This module provides the fuzzy information to be stored in the fuzzy relations.
- An *algebraic optimizer*, rewrites the queries, taking into account a number of algebraic equivalences. These equivalences are basically extensions (to the FNF^2 model) of the well known equivalences of the standard relational algebra. They allow significant cost reduction in the query evaluation process by properly choosing the ordering of the operators (for example, by anticipating projections and selections over Cartesian products and joins).
- A *query translator* transforms algebraic queries in a sequence of PL/SQL statements, that is, in queries over the object relational database.
- An *object relational database* system contains the information about the images and their contents. The database engine is written in PL/SQL code, in an Oracle 9i environment.

Figure 6 shows a screen dump of the system GUI, with the menu to query the system.

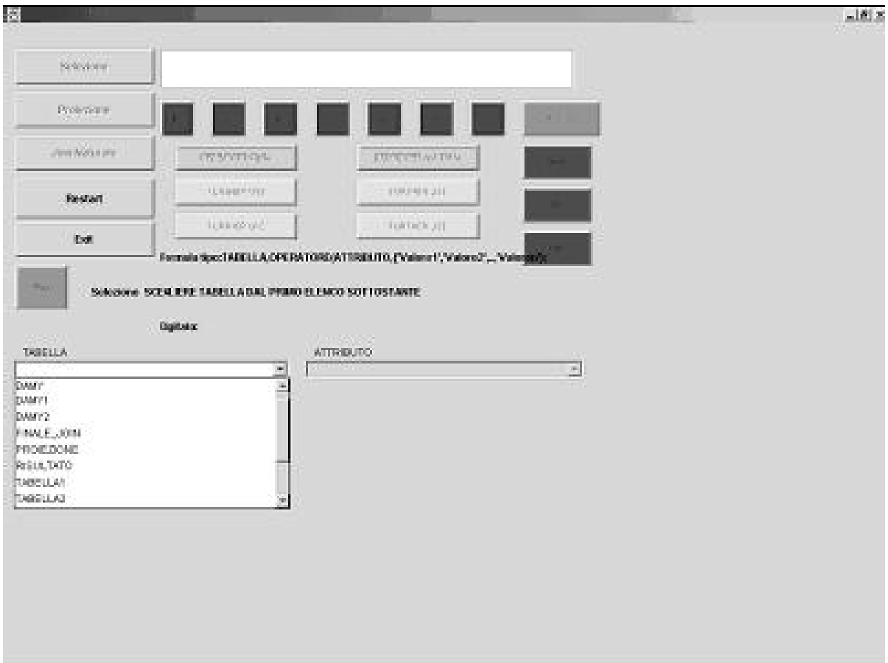

Figure 6: a screen dump of the described prototype system.

FUTURE TRENDS AND CONCLUSIONS

The problem of managing uncertainty in image databases, to define image retrieval methods that better match the users' expectations, is becoming more and more important. Unfortunately, it is far from being solved in a way that can be acceptable for the needs of the database and image processing communities.

The major current trends in the image database research include

a) development of a theoretical framework to manage uncertainty using several theories (probability, fuzziness);

b) development of a unified data model based on relational or object oriented databases; and

c) development of a model to measure the complexity of image database content and image database retrieval (similarity) algorithms.

The goal of this chapter was to highlight uncertainty-related challenges inherent in image databases and discuss several cutting-edge solutions, illustrating the theory, tools, and technology available to support various types of uncertainty. In particular, we have introduced a new and powerful model, *FNF*^{2}, developed by the authors, and we have described the main features and technical contributions of the model. The model is well suited for the definition of an extension of the relational algebra, for managing image data. A preliminary version of the extended algebra has already been defined by the authors (Chianese & al., 2004) under some simplifying hypotheses about comparison and combination operators. In particular, in the algebra the tuple combination operators have been limited to their *skeptical* versions. A prototype system based on the FNF^2

model, and including the extended relational algebra operators, has also been implemented at the University of Napoli “Federico II”. An extended version of the algebra, with the presence of both skeptical and optimistic versions of the operators, is work in progress.

BIBLIOGRAPHY

Ashby, F.G., & Perrin, N.A. (1988). Toward a unified theory of similarity and recognition. *Psychological Review*, 95(1):124-150.

Atnafu, S. & Chbeir, R., & Coquil, D. & Brunie, L. (2004). Integrating Similarity-Based Queries in Image DBMSs, *2004 ACM Symposium on Applied Computing*, pp. 735-739.

Boccignone, G. & Chianese, A., & Moscato, V., & Picariello, A. (2005) : Foveated shot detection for video segmentation. *IEEE Trans. on Circuits and Systems for Video Technology*, 15 (3): 365-377.

Buche, P, & Dervin, C., & Haemmerlé, O., & Thomopoulos, R., (2005). Fuzzy Querying of Incomplete, Imprecise, Heterogeneously Structured Data in the Relational Model Using Ontologies and Rules, *IEEE Trans. on Fuzzy Systems*, 13(3), pp. 373:383.

Buckles, B.P., & Petry, F. E. (1982). A fuzzy representation of data for relational databases. *Fuzzy Sets and Systems*, 7:213-226.

Buckles, B.P., & Petry, F. E. (1995) . Fuzzy databases in the new era. *Proceedings of the 1995 ACM symposium on Applied computing,* pages 497-502.

Chianese, A., & Picariello, A. & Sansone, L. (2001). Query by examples in image database using a fuzzy knowledge base. *In Proc. of SCI 2001*, volume 16, pages 362-367.

Chianese, A., & Picariello, A. & Sansone, L, & Sapino,M.L. (2004). Managing Uncertainties in Image Databases: A Fuzzy Approach. *Multimedia Tools Appl.* 23(3): 237-252.

Del Bimbo, A. (1999). Visual Information Retrieval. Morgan Kaufmann.

Grosky, W. I (1997). Managing multimedia information in database systems. *Communications of the ACM*, 40(12):72-80.

Itti, L. & Koch, C. (2001). Computational modelling of visual attention. *Nature Reviews - Neuroscience*, 2:1-11.

Krishnapuram, R., & Medasani, S. , & Jung, S. H., & Choi, Y.S., & Balasubramaniam, R. (2004). Content-Based Image Retrieval Based on a Fuzzy Approach, *IEEE Trans. on Knowledge and Data Engineering*, 16(10), 1185-1199.

Petry, F. E., (1986). Fuzzy Databases: Principles and Applications. Kluwer Academic Publishers, Norwell, MA.

Raju, K. V. S. V. N., & Majumdar, A. K. (1998). Fuzzy functional dependencies and lossless join decomposition of fuzzy relational database systems. *ACM Transactions on Database Systems (TODS),* 13(2):129-166.

Santini, S, & Gupta, A. & Jain, R. (2001). Emergent Semantics trhough Interactions in Image Databases. *IEEE Trans. on Knowledge and Data Engineering*, 13:337-351.

Smeulders, A.W., & Worring, M., & Santini, S., & Gupta, A., & Jain, R. (2000). Content-based image retrieval at the end of the early years. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 22(12):1349-1380.

Subrahmanian, V.S. (1998). Principles of Multimedia Database Systems. Morgan Kaufmann Publishers.

Swain, M. J. , & Ballard, D. H. (1991). Color indexing. *Int Journal of Computer Vision*.

Takahashi, Y. (1993). Fuzzy database query languages and their relational completeness theorem. *IEEE Trans. on Knowledge and data engineering*, 5(1):122-125.

Van de Wouwer, J. & al. (1999). Wavelet correlation signature for color texture characterization. *Pattern Recognition*, 32:175-183.

Wang Jing, & Dana, K. J., (2003). A novel approach for texture shape recovery, *Proc. Of 9th Int, Conference on Computer Vision*, 2:1374-1380.

Wang Zhou, & Simoncelli, E. P. (2005), Translation Insensitive Image Similarity in Complex Wavelet Domain, *IEEE Int. Conference on Acoustic, Speech and Signal Processing, 2005, 573-576.*

Wei Jiang, & Mingjing Li & Hongjiang Zhan & Jinwei Gu, (2004), On line feature selection based on generalized feature contrast model, *IEEE Int. Conf. on Multimedia and Expo, 3:1995:1998*.

Yager, R. R. (1982). Some procedures for selecting fuzzy set-theoretic operators, *Int. J. General Systems* 8:235-242.

Yang, Q., & Zhang, C. , & Wu, J., & Nakajima, C. , & Rishe, H. (2001). Efficient processing of nested fuzzy sql queries in a fuzzy database. *IEEE Trans. on Knowledge and data engineering*, 13(6):884-901.

Yazici, A. & Buckles, B.P. & Petry, F.E. (1999). Handling complex and uncertain information in the Exifo and nf2. *IEEE Transactions on Fuzzy Systems*, 7(1):659-676.

Zadeh, L. (1971). Quantitative fuzzy semantics. *Information Systems*, 3:159-176.

Zadeh, L. (1978). Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets and Systems*, 1:3-28.

Zaniolo, C. & Ceri, S. & Faloutsos, C., & Snodgrass, R. , & Subrahmanian, V.S. , & Zicari, R. (1997). Advanced Database Systems. Morgan Kaufmann Publishers.