A Normalization Framework for Multimedia Databases

Shi-Kuo Chang[1], Vincenzo Deufemia[2], Giuseppe Polese[2], Mario Vacca[2]

[1] Department of Computer Science, University of Pittsburgh
6101 Sennott Building, Pittsburgh, PA, USA, 15260
chang@cs.pitt.edu

[2] Dipartimento di Matematica e Informatica, Università di Salerno
Via Ponte don Melillo, 84084 Fisciano (SA), Italy
{deufemia, gpolese, mvacca}@unisa.it

Abstract

We present a normalization framework for designing of multimedia database schemas with reduced manipulation anomalies. To this end we introduce new extended dependencies involving different types of multimedia data. Such dependencies are based on distance functions that are used to detect semantic relationships between complex data types. Based upon these new dependencies, we have defined five multimedia normal forms. Finally, we have performed a simulation on a large image dataset to analyze the impact of the proposed framework in the context of content-based retrieval applications and in e-learning applications.

Index Terms - multimedia database management systems (MMDBMS), anomalies, data dependencies, content-based retrieval.

1 Introduction

In the last decade multimedia databases have been used in many application fields. The internet boom has increased this trend, introducing many new interesting issues related to the storage and management of distributed multimedia data. For these reasons data models and database management systems (DBMSs) have been extended in order to enable the modeling and management of complex data types, including multimedia data [29]. In particular, other than working on the extension of data models, the research community has focused on indexing techniques enabling content-based retrieval of multimedia information, query paradigms and languages, clustering techniques, and support for distributed multimedia information management.

Examples of DBMSs extended with functionalities to support multimedia data management (MMDBMSs) include CORE [47], OVID [30], VODAK [27], QBIC [20], ATLAS [36], MIRROR [15], DISIMA [32], and so on, each providing enhanced support for one or more media domains among text, sound, image, and video. At the beginning, many DBMS producers would preferably rely on the object-oriented data model to face the complexity of modeling multimedia data, but there have also been examples of MMDBMSs based on the relational data model and on specific, non-standard data models. However, in order to facilitate the diffusion of multimedia databases within industrial environments, researchers have been seeking solutions based on the relational data model, possibly associated to some standard design paradigm, like those used with traditional relational DBMSs (RDBMSs). Extensible relational DBMSs have been an attempt in this direction. In the ! last decade DBMS vendors have produced extended versions of relational DBMSs [35], with added capabilities to manage complex data types, including multimedia. In particular, these new products extend traditional RDBMSs with mechanisms for implementing the concept of object/relational universal server. In other words, they provide means to enable the construction of user defined Data Types (UDT), and Functions for manipulating them (UDF). New standards for SQL have been created, and SQL3 has become the standard for relational DBMSs extended with object oriented capabilities [16]. The standard includes UDTs, UDFs, LOBs (a variant of Blobs), and type checking on user defined data types, which are accessed through SQL statements. Early examples of extensible RDBMSs include Postgres [42], IBM/DB2 version 5 [14], Informix [35], and ORACLE 8 [31].

As MMDBMSs technology has started becoming more mature, the research community has started focusing on multimedia software engineering issues, with particular emphasis on multimedia databases. In particular, main efforts have been devoted to multimedia data indexing, and content-based retrieval [26], which has led to the development of many data indexing and organization approaches, each specialized on a particular media type, all aiming to guarantee an efficient retrieval of multimedia data based on their contents. Thus, we have had many indexing techniques for images and videos, some based on physical characteristics of media types, and others based on their semantics. However, in spite of these efforts, little attention has been devoted to multimedia databases and multimedia software engineering methodologies in the direction of providing paradigms for designing information systems needing to process many different types of multimedia data together with traditional alphanumeric data. In particular, multimedia software engineering methodologies should not only embed data indexing issues, but also techniques for database schema design, with guidelines to construct the schemas, evaluate their quality, and refactor them. To this end, in this paper we present a generic normalization framework for multimedia databases, which provides guidelines and normal forms to evaluate and improve the quality of schemas. The framework applies in a seamless way to images as well as to other media types. It is based on a new definition of imprecise dependency for multimedia data, named *type-M dependency*, which is parameterized upon the distance functions used to compare multimedia data [6], and it has been exploited to define five new normal forms. The concept of *type-M dependency* generalizes similar concepts of imprecise or fuzzy functional dependencies existing in the literature [3, 9, 34], which turned out to be inadequate to capture important aspects of multimedia data.

Regarding normalization techniques, the ones cited in the literature focus on specific domains [1, 34, 38, 46], and no general purpose normalization framework for multimedia databases is provided. In particular, the technique in [38] focuses on the normalization of image databases by partitioning images so as to enhance search and retrieval operations. To this end it aims to define dependencies among image features, suggesting the designer how to efficiently map them into a database schema. While this technique is based on physical characteristics of images, there are other techniques organizing the multimedia data based on their semantics [7, 19, 23]. However, all these proposals seek adequate index and data organization to provide efficient content based retrieval. Thus, they are complementary with respect to our framework, and can be used in conjunction with it in a synergistic way. Finally, the techniques in [1, 46] focus on the normalization of XML documents.

In order to explicate the framework, in the paper we use it for normalizing medical multimedia databases. Moreover, extensive experiments have been performed on a large image dataset in the context of content-based retrieval applications, and on a multimedia database used in e-learning applications. In the former we aimed to analyze the impact of the proposed framework on retrieval performances and errors, whereas in the second one we aimed to analyze the impact of the normalization process on access performances.

The paper is organized as follows. In Section 2 we introduce some background concepts and preliminary definitions. In Section 3 we present the concept of *type-M dependency*, and compare it with similar dependencies from the literature. In Section 4 we propose new normal forms, whereas experiments for evaluating the framework are presented in Section 5. Finally, discussion is provided in Section 6.

2 Preliminaries

In this section we introduce some basic concepts of multimedia relational databases and similarity theory [39], which will be useful for describing our normalization framework.

In the relational data model the database is viewed as a set of relations of time-varying content. A multimedia database is formed by one or more relations of the form $R(A_1,\ldots, A_n)$, where $A_1,\ldots,A_n$ are attributes. Each A_i has associated a domain denoted by $dom(A_i)$, which is the set of possible values for that attribute. The union of two sets of attributes X and Y is written as XY. An instance of R, that is, its content at a given time, is defined as a subset of the Cartesian product $dom(A_1) \times \ldots \times dom(A_n)$. This instance can be represented as a relation having as rows (named *tuples*) the elements of the subset of $dom(A_1) \times \ldots \times dom(A_n)$, and as columns the attributes of R. If $R = \{A_1, \ldots, A_n\}$ is a database schema, then we write $attr(R)$ the set of attributes of R. If t is a tuple of this table (i.e., an element in an instance of R), then $t[A]$ denotes the value of this tuple in the A-column; $t[\mathrm{A}]$ is called the A-value of t. A schema consists of a set of relations, where each relation is defined by its attribute sets and some semantic constraints.

Tuples of a relation can be compared by means of a set of relevant features Φ. For instance, images can be compared using attributes like color, texture, shape, etc., whereas audio data can be compared using loudness, pitch, brightness, and bandwidth. The values of each feature $F \in \Phi$ belong to a domain $D = dom(F)$.

The similarity between two attribute values a and b in a tuple is based on distance measures or, equivalently, on similarity functions, defined on feature spaces. In particular, given two values a and b belonging to $dom(A)$, we consider distance functions of type $d : dom(A)^2 \rightarrow [0, 1]$, such that for $a, b \in dom(A)$

1. $d(a, a) = 0$ (reflexivity)

2. $d(a, b) = d(b, a)$ (symmetry).

Given an attribute A, in what follows we denote with $D(A)$ the set of distance functions defined on A.

In order to evaluate the similarity between multimedia objects of two tuples we introduce a *tuple distance function*, which summarizes the results produced by the different distance functions applied to the elements of the tuples. In particular, given a relation $R(A_1, \ldots, A_n)$, if $t_1 = (a_1, \ldots, a_n)$ and $t_2 = (b_1, \ldots, b_n)$ are two tuples of R, then $\varpi(t_1, t_2) = g(d_1(a_1, b_1), \ldots, d_n(a_n, b_n))$ measures the distance between t_1 and t_2, where $d_i \in D(A_i)$ and $g : [0, 1]^n \rightarrow [0, 1]$ is an aggregation function that combines the n scores to derive an overall score. Aggregation functions should satisfy the triangular co-norm (t-conorm) properties, that is, the zero identity, monotonicity, commutativity, and associativity. There are several t-conorm aggregation functions defined in fuzzy logic literature [18, 49], among which the *max* function is the most commonly used. Notice that if $n = 1$ then $\varpi(t_1, t_2) = d_1(t_1, t_2)$. Given a set of attributes X, we denote with $TD(X)$ the set of tuple distance functions defined on X.

Definition 2.1 Let $R(A_1, \ldots, A_n)$, ϖ be a tuple distance function on R, τ be a threshold, $t_1 = (a_1, \ldots, a_n)$ and $t_2 = (b_1, \ldots, b_n)$ be two tuples in R, we say that t_1 is *similar* within τ to t_2 with respect to ϖ, denoted with $t_1 \cong_{(\varpi,\tau)} t_2$, iff $\varpi(t_1, t_2) \leq \tau$.

3 Extended Dependencies

A functional dependency on an alphanumeric database is defined as a constraint between two sets of attributes from the database [11]. In particular, given two sets of attributes X and Y, a functional dependency between them is denoted by $X \rightarrow Y$. The constraint says that for any two tuples t_1 and t_2, if $t_1[X] = t_2[X]$ then $t_1[Y] = t_2[Y]$. This concept cannot be immediately applied to multimedia databases, since we do not have similar simple and efficient methods to compare multimedia attributes.

Extensions to the definition of functional dependency have been produced for fuzzy databases [10, 34, 40], leading to several definitions of fuzzy functional dependencies (ffds). However, they have not reached a cogent and largely accepted view, and none of them fits the requirements of our normalization framework. For this reason, in this paper we introduce a new type of imprecise dependency, namely Type-M dependency, on which we developed our normalization framework. In the following subsection, we introduce type-M dependencies and discuss their properties. Then, we discuss why ffds are inadequate for the multimedia domain, and show that type-M dependencies generalize them.

3.1 Type-M Functional Dependencies

The following definition introduces the concept of type-M functional dependency.

Definition 3.1 Let R be a relation with attribute set U, and $X, Y \subseteq U$. $\mathrm{X}_{(g_1,\tau')} \rightarrow Y_{(g_2,\tau'')}$ is a *type-M functional dependency* (MFD) relation if and only if for any two tuples t_1 and t_2 in R that have $t_1[X] \cong_{(g_1,\tau')} t_2[X]$, then $t_1[Y] \cong_{(g_2,\tau'')} t_2[Y]$, where $g_1 \in TD(X)$ and $g_2 \in TD(Y)$, whereas τ', $\tau'' \in [0, 1]$ are thresholds.

This means that the features used by g_2 on Y depend on the features used by g_1 on X; or, alternatively, the values of the features used by g_1 on X component imply the range of values for the features used by g_2 on Y component. Notice that given a distance function d_1 and a threshold τ, $t_1 \cong_{(d_1,\tau)} t_2$ and $t_2 \cong_{(d_1,\tau)} t_3$ does not imply $t_1 \cong_{(d_1,\tau)} t_3$. However, we can state that $t_1 \cong_{(d_1,2\tau)} t_3$. In general, if $X_{d_1(\tau')} \rightarrow Y_{d_2(\tau'')}$ holds then for any two tuples t_1 and t_2 that have $t_1[X] \cong_{(d_1, k\tau')} t_2[X]$, then $t_1[Y] \cong_{(d_2, k\tau'')} t_2[Y]$, with $k \in \Re$.

As an example, if we define a functional dependency on a medical database between attributes ECG (electrocardiography) and PULSE (heartbeat), and use fractal dimensionality for comparing ECGs (e.g., [25]), and the similarity measure proposed in [21] for comparing heart sounds, we would write as follows

$$ECG_{(FRACTAL,\tau')} \rightarrow PULSE_{(HS,\tau'')} \tag{1}$$

This constraint says that for any two tuples t_1 and t_2 such that t_1[ECG] is considered similar within the threshold τ' to t_2[ECG] by the FRACTAL, then t_1[PULSE] is considered similar within the threshold τ'' to t_2[PULSE] by the HS.

From definition 3.1 it is clear that $X_I \rightarrow Y_{\varpi_2}$ is a type-M dependency relation where I is the identity relation and ϖ_2 is a tuple distance function. In particular, $X_I \rightarrow Y_I$ is a type-M dependency relation. In other words, if we use identity relations as distance functions, we can regard any dependency relation as a type-M dependency relation. Therefore, some of the type-M based normal forms we define in this paper will be identical to the usual normal forms, as long as we use identity relations as distance functions. In the following we omit the tuple distance function from the MFDs when it corresponds to the identity relation.

As an example, in a multimedia database of dogs, suppose that BREED is an alphanumeric attribute storing the breed of a dog, and PHOTO is an attribute storing its image. It might happen that BREED implies the attribute PHOTO, yielding an MFD. Thus, given two tuples t_1 and t_2, if the two tuples t_1[BREED] and t_2[BREED] are equal then also their photos should be similar according to a tuple distance function ϖ_1. We write

$$BREED \rightarrow PHOTO_{\varpi_1} \tag{2}$$

if every time t_1[BREED] is equal to t_2[BREED] then t_1[PHOTO] $=$ t_2[PHOTO] are similar according to ϖ_1. However, as it can be imagined, the distance function used heavily affects the functional dependencies. In fact, a distance function might consider two dogs similar only because they have a similar color peel, which would not imply they have the same breed. This is why the distance function has to be explicitly represented in the notation.

3.1.1 Inference Rules

The existence of certain MFDs in a relation implies the existence of others. Inference rules are means to construct these implicit dependencies. In the following we define and prove inference rules for MFDs. Given a MFD $X_{(g_1,\tau_1)} \rightarrow Y_{(g_2,\tau_2)}$, we denote with $Dist(g_1, X)$ the sequence of distance functions applied by g_1 on X. Moreover, given two functions $g_1 \in TD(X)$ and $g_2 \in TD(Y)$ we define $g_1 \bullet_h g_2(x, y)$= $h(g_1(x), g_2(y))$ with h a t-conorm aggregation function.

Theorem 3.1 Given the sets of attributes X, Y,Z, and W

1. the *reflexive rule* $XY_{(g_1,\tau_1)} \rightarrow Y_{(g_2,\tau_2)}$ holds if $Dist(g_1, Y) = Dist(g_2, Y)$ and $\tau_2 \geq \tau_1$. That is, the reflexive rule holds if the distance functions used by g_1 and g_2 on the attributes in Y are the same, and the threshold for g_2 is greater than the one for g_1.

2. the *augmentation rule* $\{X_{(g_1,\tau_1)} \rightarrow Y_{(g_2,\tau_2)}\} \models XZ_{(g_3,\tau_3)} \rightarrow YZ_{(g_4,\tau_4)}$, holds if $Dist(g_1, X) = Dist(g_3, X)$, $Dist(g_2, Y) = Dist(g_4, Y)$, $Dist(g_3, Z) = Dist(g_4, Z)$, $\tau_3 \leq \tau_1 + k$ and $\tau_4 = \tau_2 + k$ with $0 \leq k \leq min\{1 - \tau_1, 1 - \tau_2\}$.

3. the *transitive rule* $\{ X_{(g_1,\tau_1)} \rightarrow Y_{(g_2,\tau_2)}, Y_{(g_2,\tau_3)} \rightarrow Z_{(g_3,\tau_4)} \} \models X_{(g_1,\tau_5)} \rightarrow Z_{(g_3,\tau_4)}$, holds if $\tau_2 \leq \tau_3$ and $\tau_5 \leq \tau_1$.

4. the *decomposition rule* $\{ X_{(g_1,\tau_1)} \rightarrow YZ_{(g_2,\tau_2)}\} \models X_{(g_1,\tau_4)} \rightarrow Y_{(g_3,\tau_3)}$, holds if $Dist(g_2, Y) = Dist(g_3, Y)$, $\tau_4 \leq \tau_1$ and $\tau_2 \leq \tau_3$.

5. the *union rule* $\{ X_{(g_1,\tau_1)} \rightarrow Y_{(g_2,\tau_2)}, X_{(g_1,\tau_3)} \rightarrow Z_{(g_3,\tau_4)} \} \models X_{(g_1,\tau_5)} \rightarrow YZ_{(g_4,\tau_6)}$, holds if $g_4 = g_2 \bullet_h g_3$, $\tau_5 \leq \tau_1$, and $\tau_6 \geq \tau_4$.

6. the *pseudotransitive rule* $\{ X_{(g_1,\tau_1)} \rightarrow Y_{(g_2,\tau_2)}, WY_{(g_3,\tau_3)} \rightarrow Z_{(g_4,\tau_4)}\} \models WX_{(g_5,\tau_5)} \rightarrow Z_{(g_4,\tau_4)}$, holds if $Dist(g_1, X) = Dist(g_5, X)$, $Dist(g_2, Y) = Dist(g_3, Y)$, $Dist(g_3, W) = Dist(g_5, W)$, $\tau_3 \geq \tau_2$, and $\tau_5 \leq \tau_1$.

Proof:

(1) *Reflexive rule.*

Suppose that there exist two tuples t_1 and t_2 in a relation instance r of R such that $g_1(t_1[XY]$,$t_2[XY])$

$= g_3(t_1[X]$,$t_2[X]) \bullet_h g_2(t_1[Y]$,$t_2[Y]) \leq \tau_1$ with $Dist(g_1, X)$=$Dist(g_3, X)$ and $Dist(g_1, Y)$=$Dist(g_2, Y)$. Then, $g_2(t_1[Y], t_2[Y]) \leq \tau_1$ since the t-conorm function h satisfies the statement $h(a,b) \geq max\{a,b\}$.

(2) *Augmentation rule.*

Suppose that $X_{(g_1,\tau_1)} \rightarrow Y_{(g_2,\tau_2)}$ holds in a relation instance r of R, but that $XZ_{(g_3,\tau_3)} \rightarrow YZ_{(g_4,\tau_4)}$ does not hold. Then, there must exist two tuples t_1 and t_2 in r such that $g_1(t_1[X]$,$t_2[X]) \leq \tau_1$, $g_2(t_1[Y]$,$t_2[Y]) \leq \tau_2$, $g_3(t_1[XZ]$,$t_2[XZ]) \leq \tau_3$ and $g_4(t_1[YZ]$,$t_2[YZ]) > \tau_4$. This is not possible because $g_3(t_1[XZ]$,$t_2[XZ])$= $g_1(t_1[X]$,$t_2[X]) \bullet_h g_5(t_1[Z]$,$t_2[Z]) \leq \tau_3 \leq \tau_1 + k$ and $g_4(t_1[YZ]$,$t_2[YZ])$ $= g_2(t_1[Y]$,$t_2[Y]) \bullet_h g_5(t_1[Z]$,$t_2[Z]) = \tau_2 + k \leq \tau_4$.

(3) *Transitive rule.*

Let us assume that $X_{(g_1,\tau_1)} \rightarrow Y_{(g_2,\tau_2)}$ and $Y_{(g_2,\tau_3)} \rightarrow Z_{(g_3,\tau_4)}$ hold in a relation instance r of R. Then, for any two tuples t_1 and t_2 in r such that $g_1(t_1[X]$,$t_2[X]) \leq \tau_1$ we have $g_2(t_1[Y]$,$t_2[Y]) \leq \tau_2 \leq \tau_3$, and hence we also have $g_3(t_1[Z]$,$t_2[Z]) \leq \tau_4$. This means that also $X_{(g_1,\tau_5)} \rightarrow Z_{(g_3,\tau_4)}$ holds with $\tau_5 \leq \tau_1$.

(4) *Decomposition rule.*

Let us assume that $X_{(g_1,\tau_1)} \rightarrow YZ_{(g_2,\tau_2)}$ holds in a relation instance r of R. Then, $YZ_{(g_2,\tau_2)} \rightarrow Y_{(g_3,\tau_3)}$ holds from (1) with $Dist(g_2, Y) = Dist(g_3, Y)$ and $\tau_3 \geq \tau_2$. Using (3) $X_{(g_1,\tau_4)} \rightarrow Y_{(g_3,\tau_3)}$ holds with $\tau_4 \leq \tau_1$.

(5) *Union rule.*

Let us assume that $X_{(g_1,\tau_1)} \rightarrow Y_{(g_2,\tau_2)}$, $X_{(g_1,\tau_3)} \rightarrow Z_{(g_3,\tau_4)}$ hold in a relation instance r of R. Then, $X_{(g_1,\tau_1)} \rightarrow XX_{(g_5,\tau_7)}$ holds with $g_5 = g_1 \bullet_h g_1$ and $\tau_7 = h(\tau_1, \tau_1)$, and $XX_{(g_5,\tau_8)} \rightarrow XY_{(g_6,\tau_9)}$ holds from (2) with $g_7 = g_1 \bullet_h g_2$, $Dist(g_5, X) = Dist(g_6, X)$, $\tau_8 \leq \tau_1 + k$ and $\tau_9 = \tau_2 + k$ where $0 \leq k \leq min\{1 - \tau_1, 1 - \tau_2\}$. Moreover, $XY_{(g_6,\tau_{10})} \rightarrow YZ_{(g_4,\tau_6)}$ holds from (2) with $Dist(g_1, X) = Dist(g_6, X)$, $Dist(g_3, Z) = Dist(g_4, Z)$, $Dist(g_6, Y) = Dist(g_4, Y)$, $t_{10} \leq \tau_3 + k$ and $\tau_6 = t_4 + k$ where $0 \leq k \leq min\{1 - \tau_3, 1 - \tau_4\}$. Using (3) $X_{(g_1,\tau_5)} \rightarrow YZ_{(g_4,\tau_6)}$ holds with $\tau_5 \leq \tau_1$, $\tau_7 \leq \tau_8$ and $\tau_9 \leq \tau_{10}$.

(6) *Pseudotransitive rule.*

Let us assume that $X_{g_1(\tau_1)} \rightarrow Y_{(g_2,\tau_2)}$, $WY_{(g_3,\tau_3)} \rightarrow Z_{(g_4,\tau_4)}$ hold in a relation instance r of R with $Dist(g_2, Y) = Dist(g_3, Y)$. Then, $WX_{(g_5,\tau_5)} \rightarrow WY_{(g_3,\tau_7)}$ holds from (2) with $Dist(g_5, W) =$

$Dist(g_3, W)$, $\tau_5 \leq \tau_1 + k$ and $\tau_7 = \tau_2 + k \leq \tau_3$ where $0 \leq k \leq min\{1 - \tau_1, 1 - \tau_2\}$. Using (3) $WX_{(g_5,\tau_5)} \rightarrow Z_{(g_4,\tau_4)}$ holds with $\tau_7 \leq \tau_3$ and $\tau_5 \leq \tau_1$. □

3.1.2 Multivalued and Join Type-M Dependencies

Multivalued dependencies (MVDs) were introduced in traditional relational databases as a generalization of functional dependencies to capture a significant amount of semantic information useful for normalization [17]. In the following we extend the notion of MVD to multimedia databases.

Definition 3.2 Let R be a multimedia relation with attribute set U, and X, $Y \subseteq U$.
We say that $X_{(g_1,\tau')} \longmapsto Y_{(g_2,\tau'')[(g_3,\tau''')]}$ is a *type-M multivalued dependency* (MMD) relation if and only if for any two tuples t_1 and t_2 in R such that $t_1[X] \cong_{(g_1,\tau')} t_2[X]$, there also exist two tuples t_3 and t_4 in R with the following properties:

- $t_3[X], t_4[X] \in [t_1[X]]_{\cong_{(g_1,\tau')}}$
- $t_3[Y] \cong_{(g_2,\tau'')} t_1[Y]$ and $t_4[Y] \cong_{(g_2,\tau'')} t_2[Y]$
- $t_3[R - (XY)] \cong_{(g_3,\tau''')} t_2[R - (XY)]$ and $t_4[R - (XY)] \cong_{(g_3,\tau''')} t_1[R - (XY)]$.

where $g_1 \in TD(X)$, $g_2 \in TD(Y)$ and $g_3 \in TD(R - (XY))$, whereas τ', τ'', and $\tau''' \in$ [0,1] are thresholds.

Because of the symmetry in the definition, whenever $X_{(g_1,\tau')} \longmapsto Y_{(g_2,\tau'')[(g_3,\tau''')]}$ holds in R, so does $X_{(g_1,\tau')} \longmapsto [R - (XY)]_{(g_3,\tau''')[(g_2,\tau'')]}$.

An MMD $X_{(g_1,\tau')} \longmapsto Y_{(g_2,\tau'')[(g_3,\tau''')]}$ in R is called *trivial* if (a) $Y \subseteq X$ or (b) $X \cup Y = R$. An MMD that satisfies neither (a) nor (b) is called *non trivial.*

Similarly to multimedia functional dependencies (MFDs), we can define inference rules for MMDs.

1. $X_{(g_1,\tau_1)} \longmapsto Y_{(g_2,\tau_2)[(g_3,\tau_3)]} \models X_{(g_1,\tau_4)} \longmapsto [R - (XY)]_{(g_3,\tau_5)[(g_2,\tau_6)]}$, where $g_1 \in TD(X)$, $g_2 \in TD(Y)$, $g_3 \in TD(R - (XY))$, $\tau_4 \leq \tau_1$, $\tau_5 \geq \tau_3$, and $\tau_6 \geq \tau_2$.

2. If $X_{(g_1,\tau_1)} \longmapsto Y_{(g_2,\tau_2)[(g_3,\tau_3)]}$ and $W \supseteq Z$ then $WX_{(g_4,\tau_4)} \longmapsto YZ_{(g_5,\tau_5)[(g_6,\tau_6)]}$ where $Dist(g_4, Z) = Dist(g_5, Z)$, $Dist(g_1, X) = Dist(g_4, X)$, and $Dist(g_2, Y) = Dist(g_5, Y)$, $\tau_4 \geq \tau_1$, $\tau_5 \geq \tau_2 + (\tau_4 - \tau_1)$, and $\tau_6 \geq \tau_3$.

3. $\{X_{(g_1,\tau_1)} \longmapsto Y_{(g_2,\tau_2)[(g_3,\tau_3)]}, Y_{(g_2,\tau_4)} \longmapsto Z_{(g_4 5,\tau_5)[(g_5,\tau_6)]}\} \models X_{(g_1,\tau_7)} \longmapsto (Z - Y)_{(g_6,\tau_8)[(g_7,\tau_9)]}$ where $Dist(g_3, Z - Y) = Dist(g_4, Z - Y) = Dist(g_6, Z - Y)$, $g_7 \in TD(R - (Z - Y))$, $\tau_2 \leq \tau_4$, $\tau_7 \leq \tau_1$, $\tau_8 \geq \tau_5$, and $\tau_9 \geq \tau_6$.

4. $X_{(g_1,\tau_1)} \rightarrow Y_{(g_2,\tau_2)} \models X_{(g_1,\tau_1)} \longmapsto Y_{(g_2,\tau_2)[(g_3,1)]}$.

5. If $X_{(g_1,\tau_1)} \longmapsto Y_{(g_2,t\tau_2)[(g_3,\tau_3)]}$ and there exists W with the properties that (a) $W \cap Y = \emptyset$, (b) $W_{(g_4,\tau_4)} \rightarrow Z_{(g_5,\tau_5)}$, and (c) $Y \supseteq Z$, then $X_{(g_1,\tau_6)} \rightarrow Z_{(g_5,\tau_7)}$ with $\tau_6 \leq \tau_1$ and $\tau_7 \geq \tau_1$.

Given a set D of MFDs and MMDs specified on a relation schema R, we can use the inference rules to infer the set of all dependencies D^+ that will hold in every relation instance of R satisfying D.

In order to present the notion of multimedia join dependency, we need to introduce the multimedia operations of projection and join. Given a relation r over a multimedia relation $R(X)$, a subset Y of X, a tuple distance function $g \in TD(Y)$, and a threshold τ, the *multimedia projection* of R on Y respect to (g, τ), denoted with $\Pi_{Y,(g,\tau)}(R)$, is defined by

$$\Pi_{Y,(g,\tau)}(R) = \{v(Y) \mid v \in r \text{ and } g(v, w) \leq \tau \text{ for each tuple } w \text{ in } u[Y]\}$$

Note that the duplicate elimination is performed according to the function g, and the associated threshold τ. Obviously, if $\tau = 0$ then $w = v$, and the tuple distance function g corresponds to exact match for the particular features it considers.

Let $R(X, Y)$ and $S(Y, Z)$ be multimedia relations where X, Y, and Z are disjoint sets of attributes, $g \in TD(Y)$ be a tuple distance function, and τ be a threshold. The *multimedia join* of R and S respect to (g, τ), denoted with $R \bowtie_{(g,\tau)} S$, is the relation defined by

$$R \bowtie_{(g,\tau)} S = \{(x, y, z, k) \mid (x, y) \in R, (y', z) \in S \text{ with } y \cong_{(g,\tau)} y', \text{and } k = g(y, y')\}$$

That is, the multimedia join is created by linking tuples of R with tuples of S that have similar values, within a threshold τ with respect to a function g, for all the attributes that are common to the two multimedia relations. The parameter k is introduced in the joined tuples, and represents a fuzzy value describing their degree of similarity. Obviously, in case R and/or S are the result of previous join operations, the fuzzy values used to produce them do not concur to the result of $R \bowtie_{(g,\tau)} S$.

Notice that the multimedia join raises many new issues and problems. In fact, we have higher probability to generate spurious tuples due to false alarms. Moreover, false dismissals lead to a new type of manipulation anomaly, not existing in traditional alphanumeric databases, namely the problem of *dismissed tuples*. These are tuples that should have been generated as a result of the multimedia join, but indeed they were discarded because a false dismissal occurred. We empirically analyze these issues in the evaluation section (Section 5), where we describe how these anomalies manifest under different thresholds. In these experiments conducted on a large real-world image dataset we have always been able to find a threshold interval where such anomalies were acceptable.

In the following we give the definition of Type-M join dependency.

Definition 3.3 Let R be a relation on U, and $\{X_1, \ldots, X_n\} \subseteq U$, with the union of X_i's being U. If $R = \Pi_{X_1,(g_1,\tau_1)}(R) \bowtie_{(g_1,\tau_1)} \Pi_{X_2,(g_2,\tau_2)}(R) \bowtie_{(g_2,\tau_2)} \ldots \bowtie_{(g_{n-1},\tau_{n-1})} \Pi_{X_n,I}(R)$, we say that R satisfies a *Type-M join dependency* (MJD), denoted by $\bowtie_{[(g_1,\tau_1),\ldots,(g_{n-1},\tau_{n-1})]} [X_1, \ldots, X_n]$, where $g_i \in TD(X_i \cap X_{i+1})$ and $\tau_i \in [0, 1]$ for each $1 \leq i \leq n-1$.

An MVD is a special case of an MJD. An MVD $X_{(g_1,\tau_1)} \longmapsto Y_{(g_2,\tau_2)[(g_3,\tau_3)]}$ for a relation on R is the MJD $\bowtie_{[(g_1,\tau_1)\bullet_h(g_2,\tau_2),(g_1,\tau_1)\bullet_h(g_3,\tau_3)]} (XY, X(R-Y))$.

In the following we provide some inference rules to infer MJDs. Let $S = \{X_1, \ldots, X_n\}$ and $R = \{Y_{n+1}, \ldots, Y_m\}$.

1. $\emptyset \models \bowtie_{[(g,\tau)]} [X]$, for any finite set of attributes X, and with $g \in TD(X)$, $\tau \in [0, 1]$.

2. $\bowtie_{[(g_1,\tau_1),\ldots,(g_{n-1},\tau_{n-1})]} [S] \models \bowtie_{[(g_1,\tau_1),\ldots,(g_{n-1},\tau_{n-1}),(g_n,\tau_n)]} [S, Y]$ if $Y \in attr(S)$ and $g_n \in TD(Y)$.

3. $\bowtie_{[(g_1,\tau_1),\ldots,(g_{n-1},\tau_{n-1}),(g_n,\tau_n),(g_{n+1},\tau_{n+1})]} [S, Y, Z] \models \bowtie_{[(g_1,\tau_1),\ldots,(g_n,\tau_n)\bullet_h(g_{n+1},\tau_{n+1})]} [S, YZ]$.

4. $\{\bowtie_{[(g_1,\tau_1),\ldots,(g_{n-1},\tau_{n-1}),(g_n,\tau_n)]} [S,Y],\ \bowtie_{[(g_{n+1},\tau_{n+1}),\ldots,(g_{m-1},\tau_{m-1})]} [R]\} \models$

   $\bowtie_{[(g_1,\tau_1),\ldots,(g_{n-1},\tau_{n-1}),(g_n,\tau_n),(g_{n+1},\tau_{n+1}),\ldots,(g_{m-1},\tau_{m-1})]} [S,R]$ if $Y = attr(R)$.

5. $\bowtie_{[(g_1,\tau_1),(g_2,\tau_2)]} [S,YA] \models \bowtie_{[(g_1,\tau_1),(g_2,\tau_2)]} [S,Y]$ if $A \notin attr(S)$.

3.2 Comparing MFD with other Extended Dependencies

In this section we compare the type-M dependency with other fuzzy functional dependencies (ffds) [10, 34, 40], which were introduced for fuzzy databases. In particular, we refer to three among the most relevant ffds, and provide theorems showing that type-M dependency generalizes them. Moreover, we discuss why ffds are not suitable for normalizing multimedia databases, motivating the introduction of type-M dependency. These arguments can be easily applied to other ffds since we do not pose constraints on distance functions and type-M dependencies thresholds.

For sake of uniformity, we will use the notation used in [4]: $RES_X(t_1[X], t_2[X])$ is the resemblance on tuples computed on the subset X of attributes; $\Rightarrow_{RG}$ is the $Rescher - Gaines'$ implication that is equal to one if the value on the left-hand side is less or equal than that on the right-hand side, otherwise it is equal to zero; $\Rightarrow_G$ denotes the *Gödel* implication, whose result is one if the left-hand side value is less or equal to the right-hand side one, otherwise it is equal to the value on the right-hand side.

Proposition 3.1 If any relation instance r on a schema R satisfies the Raju-Majumdar fuzzy functional dependency $X \rightarrow Y$ [34], then it also satisfies a type-M functional dependency $X_{(g_1,\tau')} \rightarrow Y_{(g_2,\tau'')}$.

Proof: If the Raju - Majumdar ffd $X \rightarrow Y$ holds in a relation instance r, then for all tuples t_l and t_2 of r we have $RES_X(t_1[X], t_2[X]) \Rightarrow_{RG} RES_Y(t_1[Y], t_2[Y])$, where $RES_X(t_1, t_2) = min_{i \in X}\{\mu^i_{EQ}(t_1[A_i], t_2[A_i])\}$, with μ^i_{EQ} similarity functions over attributes.

From the definition it follows that $1 - RES_X(t_1[X], t_2[X]) \geq 1 - RES_Y(t_1[Y], t_2[Y])$.

Thus, by setting

$g_1(t_1[X], t_2[X]) = 1 - RES_X(t_1[X], t_2[X])$,

$g_2(t_1[Y], t_2[Y]) = 1 - RES_Y(t_1[Y], t_2[Y])$

we have that the tuple distance functions g_1 and g_2 are composed of a t-conorm aggregation function, plus reflexive and symmetric distance functions, then
$g_1(t_1[X], t_2[X]) \leq \tau'$ implies $g_2(t_1[Y], t_2[Y]) \leq \tau''$ with $\tau' \leq \tau''$.

Therefore, the MFD $X_{(g_1,\tau')} \rightarrow Y_{(g_2,\tau'')}$ also holds. □

Thus, a Raju-Majumdar ffd can always be rewritten as a Type-M functional dependency. Vice versa, a Type-M functional dependency cannot always be rewritten as a Raju-Majumdar ffd. In fact, if the first threshold τ' is greater than the second one τ'', and $g_1(t_1[X], t_2[X]) \leq \tau'$ implies $g_2(t_1[Y], t_2[Y]) \leq \tau''$, then it could result that $g_1(t1[X], t2[X]) \geq g2(t1[Y], t2[Y])$.

In other words, type-M functional dependencies allows a similarity between Y-representations to be weaker than a similarity between X-representations.

Proposition 3.2 If any relation instance r on a schema R satisfies a Chen fuzzy functional dependency $X \rightarrow_q Y$ [10], then it also satisfies a MFD $X_{(g_1,\tau')} \rightarrow Y_{(g_2,\tau'')}$.

Proof: Recall that a Chen fuzzy functional dependency $X \rightarrow_q Y$ with $q \in [0, 1]$ is valid in r iff $\forall t_1, t_2 \in r$ if $t_1[X] = t_2[X]$ then $t_1[Y] = t_2[Y]$ else $I(RES_X(t_1[X], t_2[X]), RES_Y(t_1[Y], t_2[Y])) \geq q$, where I denotes the *Gödel* implication, ($I(a, b) = 1$ if $a \leq b$, b otherwise).

The interpretation of this FFD is: when two tuples have the same value (or representation) on X, then they should have the same value (or representation) on Y due to the *if* part of the definition.

If $q = 1$, then the Chen ffd reduces to the Raju and Majumdar ffd [34], and the proposition holds.

If $q < 1$, then $RES_X(t_1[X], t_2[X]) > RES_Y(t_1[Y], t_2[Y])$, and
$I(RES_X(t_1[X], t_2[X]), RES_Y(t_1[Y], t_2[Y])) = RES_Y(t_1[Y], t_2[Y])$. Thus, we obtain
$RES_X(t_1[X], t_2[X]) > RES_Y(t_1[Y], t_2[Y]) \geq q \; \forall t_1, t_2 \in R$, and,
$1 - RES_X(t_1[X], t_2[X]) \leq 1 - RES_Y(t_1[Y], t_2[Y]) \leq 1 - q$.
By setting
$g_1(t_1[X], t_2[X]) = 1 - RES_X(t_1[X], t_2[X])$,
$g_2(t_1[Y], t_2[Y]) = 1 - RES_Y(t_1[Y], t_2[Y])$, and

we have that the tuple distance functions g_1 and g_2 are composed of a t-conorm aggregation function, plus reflexive and symmetric distance functions, then
$g_1(t_1[X], t_2[X]) \leq \tau'$ implies $g_2(t_1[Y], t_2[Y]) \leq \tau''$ holds for $\tau' = 1 - q$ and $\tau'' \geq \tau'$, and the proposition holds. □

Proposition 3.3 If any relation instance r on a schema R satisfies a Sözat and Yazici fuzzy functional dependency $X \overset{\theta}{\rightarrow}_F Y$ [40], then it also satisfies a MFD $X_{(g_1,\tau')} \rightarrow Y_{(g_2,\tau'')}$.

Proof: If the Sözat and Yazici ffd $X \overset{\theta}{\rightarrow}_F Y$ holds in a relation instance r, where θ is a real number in $[0, 1]$ describing the linguistic strength, then $\mathcal{C}(t_1[Y], t_2[Y]) \geq min(\theta, \mathcal{C}(t_1[X], t_2[X]))$ for every pair of tuples t_1 and t_2 in r.

By setting
$g_1(t_1[X], t_2[X]) = 1 - \mathcal{C}(t_1[X], t_2[X])$ and $g_2(t_1[Y], t_2[Y]) = 1 - \mathcal{C}(t_1[Y], t_2[Y])$
we have that the tuple distance functions g_1 and g_2 are composed of a t-conorm aggregation function. Moreover, if $g_1(t_1[X], t_2[X]) \leq \tau'$, then $1 - \mathcal{C}(t_1[X], t_2[X]) \leq \tau'$. Since $1 - \mathcal{C}(t_1[Y], t_2[Y]) \leq 1 - min(\theta, \mathcal{C}(t_1[X], t_2[X]))$ it follows that:

1. if $\mathcal{C}(t_1[X], t_2[X]) < \theta$ then $g_2(t_1[Y], t_2[Y]) \leq 1 - \mathcal{C}(t_1[X], t_2[X]) \leq \tau'$
2. if $\mathcal{C}(t_1[X], t_2[X]) \geq \theta$ then $g_2(t_1[Y], t_2[Y]) = 1 - \mathcal{C}(t_1[Y], t_2[Y]) \leq 1 - \theta$

Hence, $g_1(t_1[X], t_2[X]) \leq \tau'$ implies $g_2(t_1[Y], t_2[Y]) \leq \tau''$ for $\tau' = 1 - \theta$ and $\tau' \geq \tau''$, and the proposition holds. □

The ffds analyzed here are not able to capture some cases of dependency [13], because of the strong relations between the similarity on the input values and that on the output values [10, 34, 40]. For instance, it is forbidden the existence of an ffd between two sets of attributes where on the first one there is a similarity function stronger than the second one. On the contrary, multimedia databases call for many kinds of dependencies involving both alphanumeric and multimedia attributes and whose relation between the two resemblance values cannot be determined in advance. As an example, we would not be able to define an MFD like $ECG_{(FRACTAL,\tau')} \rightarrow PULSE_{(HS,\tau'')}$, where $\tau'' > \tau'$.

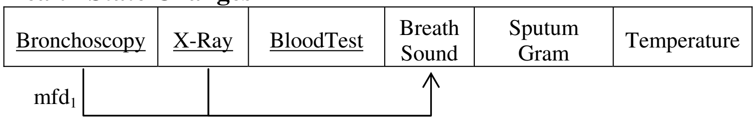
4 Normal Forms in Multimedia Databases

In traditional alphanumeric databases, normal forms are used to derive database schemas preventing manipulation anomalies [16]. Similar anomalies can arise in multimedia databases. Thus, a multimedia database designer should take all the precautions at database design time to avoid such anomalies.

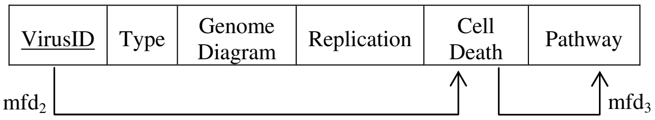
As an example, let us consider a database of viral lung diseases. Important data in medical databases are those related to health state changes and causes of diseases. According to Thagard [45], the representation of state changes is multimodal, and it involves both multimedia data - like the breathing and cough sound, the health state of the lungs (presence of pneumonia), detected by bronchoscopy and/or X-ray images - and alphanumerical ones, like the temperature. Furthermore, the causes of diseases are also represented by multimedia data [43, 44, 45]: information about the virus structure; information about the mechanisms that cause the disease. Mechanisms are, for example, those related to the phases of the virus attack (attachment, entry, assembly, replication, release) or to the relationship between the cell damage and the symptoms, or between the immune response and the symptoms.

In what follows, we show a simplified portion of the database schema for viral lung diseases, on which we highlight some relevant MFDs associated to its attributes:

Health State Changes



Viruses



A tuple in the first relation represents a particular health status. The latter is identified through a combination of a Bronchoscopy, X-Ray, and BloodTest. The Bronchoscopy is a video, whereas X-Ray is an image. For sake of simplicity, also the BloodTest is represented as an image, since it is a compound document possibly containing both alphanumeric and diagrammatic information. Notice that these are multimedia attributes, hence their values are representative samples. In other words, for a given health status we can choose the values for the attributes of its key among many similar candidates. BreathSound is a sound, SputumGram is an image, and Temperature is an alphanumeric string.

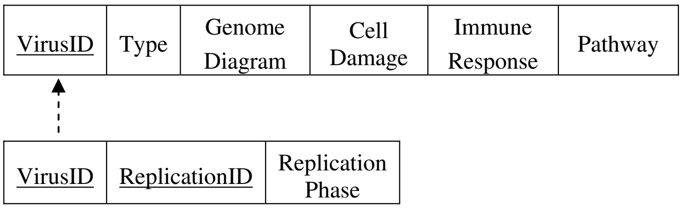
The second relation contains data of viruses. Each of them is identified through an alphanumeric ID representing the technical name of the virus, such as H5N1 for aviary flu, SARS-CoV for the SARS disease, etc. The Type is an alphanumeric attribute representing virus family, the GenomeDiagram is an image, the Replication is a video describing all the phases of a virus attack (attachment, entry, assembly, replication, release), the CellDamage is an image showing the damages that the virus causes on cells, and the Pathway is an image explaining the virus activity in terms of biochemical reactions [44].

On the schema we have shown some relevant MFDs, and it is easy to imagine how they yield manipulation anomalies. In fact, mfd_1 reveals that a strict correlation exists between the BreathSound and the combination of Bronchoscopy and X-Ray test results. This suggests to put these attributes together on a separate relation, in order to avoid possible manipulation anomalies. As an example, if we want to insert a new breath sound that is associated to particular patterns appearing in bronchoscopy and/or X ray, we cannot add a tuple without first having a sample blood test, since this attribute is part of the key. Mfd_2 and mfd_3 together reveal an indirect dependency of the attribute Pathway from the key, which suggests to put CellDeath and Pathway on a different relation in order to avoid other possible anomalies. In fact, we might have data redundancy when different viruses share the same cell damage. This causes not only a waste of disk space, but also problems if we will to update the values of these two attributes with different images, since we should make this update for all the tuples containing the same combination of values for these two attributes. Moreover, the deletion of the last tuple containing a given combination of CellDeath and Pathway causes loss of information from the database.

In this section we present three normal forms for multimedia databases. They are based on the type-M dependencies defined above. Therefore, their results depend upon the distance functions used to derive dependencies, and can be used to derive multimedia database schemas with reduced manipulation anomalies.

The first normal form (1MNF) regards the granularity of multimedia attributes. We say that a multimedia database schema is in *first multimedia normal form* if each attribute A is single valued and contains an elementary value. The latter is a relative concept, because a multimedia object can be elementary for certain application domains, whereas it might require a further segmentation due to frequent queries on its content. This issue also arises in traditional alphanumeric databases, where for example a civic address might be modeled by a single string containing the street name, the civic number, and the ZIP code; such a value might be elementary for certain application domains, but not for those requiring to inquire on single address components.

The application of the normalization process is based on a specific segmentation function: image attributes can be decomposed in a certain number of k image components, which will be stored as separated attributes; video attributes can be split into several multimedia components. For instance, the relation *Virus* shown above is not in 1NF since *Replication* is a video attribute. We could normalize the schema by segmenting the video into images representing the various phases of virus replication. Thus, we obtain the following schema:



where the obtained images are stored in a separate relation, with a foreign key *VirusID* on the original table, since there is a 1 to N relationship between viruses and their replication phase images. The ReplicationID attribute is used to reconstruct the video sequence starting from the stored images.

Obviously, the application of these normal forms requires the availability of specific segmentation functions. Moreover, as said above the decomposition of a composite multimedia attribute may require the storing of additional data structures to enable the reconstruction of the original attribute format. In particular, such data structure should store the relations between the different attribute components.

We say that a multimedia database schema is in *second multimedia normal form* (2MNF) if it is in 1MNF, and each non prime attribute A is fully dependent on the primary key. In case there is a partial dependency of A from a subset $\{k_i, \ldots, k_j\}$ of key attributes, then the designer can decide to normalize the schema by splitting the original schema R into two sub-schemas $R_1 = R - T$ and $R_2 = \{k_i, \ldots, k_j\} \cup T$, where $T = \{A\} \cup \{B_i | B_i \in R, \{k_i \ldots k_j\}_{s_1} \rightarrow \{B_i\}_{s_2}\}$. For brevity, in the following we omit the threshold from the similarity expressions.

As an example, let us analyze the MFD of relation schema *Health State Changes* from the database seen above.

$\{$*Bronchoscopy, X-Ray*$\}$ → *Breath Sound* is a partial dependency, which leads to the decomposition of the relation into the following relations, each of which is in 2MNF.

Bronchoscopy	X-Ray	Blood Tests	Sputum Gram	Temperature

Bronchoscopy	X-Ray	Breath Sound

We say that a multimedia database schema is in *third multimedia normal form* (3MNF) if it is in 2MNF, and the non prime attributes are not mutually dependent. Equivalently, we can say that the schema R is in third normal form if, whenever a MFD $X_{s_1} \rightarrow A_{s_2}$ holds in R, either

(a) X is a superkey of R, or

(b) A is a prime attribute of R.

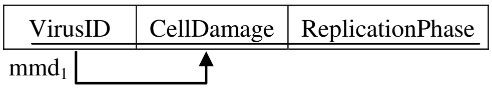
As an example, the dependency mfd_3 violates 3MNF because *CellDeath* is not a superkey of the relation, and *Pathway* is not a prime attribute. We can normalize the relation schema by decomposing it into the following two 3MNF relation schemas. We construct the first relation by removing the attributes violating 3MNF, namely *Pathway*, from the original relation, and placing them with *CellDeath* into the second relation, as shown in the following relation.

VirusID	Type	Genome Diagram	Cell Damage

Cell Damage	Pathway

Notice that by iteratively transforming a database schema to put it in 1MNFs may cause the introduction of MMDs. Such undesirable dependencies can be detected by the fourth multimedia normal form. We say that a multimedia database schema R is in *fourth multimedia normal form* (4MNF) with respect to a set of multimedia dependencies D if, for every nontrivial MMD $X_{(g_1,\tau_1)} \rightarrow Y_{(g_2,\tau_2)[(g_3,\tau_3)]}$ in D^+, X is a superkey for R. In case there is a nontrivial MMD $X_{(g_1,\tau_1)} \rightarrow Y_{(g_2,\tau_2)[(g_3,\tau_3)]}$ in D^+ with X not superkey for R, then the designer can decide to normalize the schema by splitting the original schema R into two sub-schemas $R_1 = (X \cup Y)$ and $R_2 = (R - Y)$.

As an example, let us consider the following simple multimedia relation:



The multivalued dependency *mmd*$_1$ violates 4MNF because *VirusID* is not a superkey of the relation. We can normalize the relation schema by decomposing it into the following two 4MNF relation schemas:

VirusID	CellDamage

VirusID	ReplicationPhase

Finally, a multimedia database schema R is said to be in *fifth multimedia normal form* (5MNF) if it guarantees the lossless join properties, and prevents the problem of dismissed tuples. Formally, we say that R is in 5MNF with respect to a set D of MFDs, MMDs, and MJDs if, for every nontrivial type-M join dependency $\bowtie_{[(g_1,\tau_1),\ldots,(g_{n-1},\tau_{n-1})]} (X_1, \ldots, X_n)$ in D^+, each X_i is a superkey for R.

5 Framework Evaluation

Advantages and drawbacks of normalization have been widely discussed in the relational database literature [2, 16]. The multimedia database field shares many of the issues and problems discussed in such literature, but it also provides several new specific aspects to be analyzed. In particular, as in traditional databases, the multimedia database normalization process entails a design overhead, and its benefits heavily depend on the characteristics of the specific application context. For example, in some application contexts it might happen that a highly fragmented normalized database schema makes data navigation inefficient. Moreover, since multimedia databases are mainly targeted to applications involving content-based retrieval of multimedia data, it becomes crucial to evaluate the impact of normalization over these application domains [26]. In this context, the impreciseness of comparisons involving multimedia attributes is another important issue to be considered. Thus, the multimedia database designer needs more sophisticated guidelines and tools to be able to understand the degree of normalization guaranteeing the right compromise among quality of data organization, correctness of results, and time performances, for each specific application context.

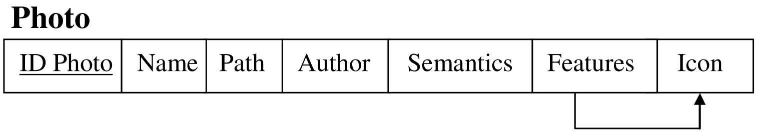
Many are the factors affecting retrieval effectiveness in multimedia databases as opposed to those of traditional alphanumeric databases. In fact, since alphanumeric databases use exact match paradigms, errors are mainly caused by inappropriate data organization, which may cause manipulation anomalies, data redundancy, inconsistencies, accidental data deletion, and so on. The designer can use normalization to prevent such errors, but only to the extent that keeps adequate time performances. On the contrary, querying of multimedia databases is typically accomplished by using approximate match paradigms, which are inherently error-prone. Thus, errors might not only due to inappropriate data organization, but also to potential failures of the matching functions. The normalization process presented in the paper can prevent the formers, but it could introduce additional retrieval errors since it relies itself on approximate matching functions. The designer can reduce such phenomena by narrowing thresholds used by such functions. Moreover, by still acting on thresholds, s/he can affect not only retrieval effectiveness, but also database size and time performances. However, despite complexity and size of multimedia data, hardly ever database size is a major concern, also due to the low cost of recent storage devices. On the other hand, time performances and retrieval effectiveness are both important issues, and often it is difficult achieving them together. To this end, each application context gives different relevance to these two parameters. There are contexts in which retrieval effectiveness is so critical that the designer can also tolerate poor time performances, whereas in other domains it is more important to gain faster although less accurate query responses.

In what follows we describe some experiments to analyze the impact of thresholds on retrieval effectiveness and time performances. In particular, we describe the application of our normalization framework in two domains: content-based retrieval of images from a heterogeneous dataset, and an e-learning application context. In the latter we have focused on time performances, whereas in the former we have mainly focused on retrieval effectiveness.

5.1 Evaluating the framework in an image retrieval context

In the following we analyze the impact of the proposed normalization framework in the context of a content-based retrieval application. In particular, we performed a simulation through an application entailing content-based retrieval from a large MPEG-7 image dataset, aiming to analyze the effect of the normalization process on retrieval effectiveness. The latter is measured in terms of precision and recall. The image dataset used in this experiment contained about 24k images, and it has been derived by converting the Berkeley's CalPhotos collection [5] to the MPEG-7 format. We first describe a relation taken from the database schema of the selected application, and then describe the normalization process applied to it.

We considered the schema of a database whose data describe the characteristics of the above mentioned image dataset, including information about semantics of images, authors, etc. In particular, for describing our experiment we focus on the following significant relation extracted from the whole database schema:



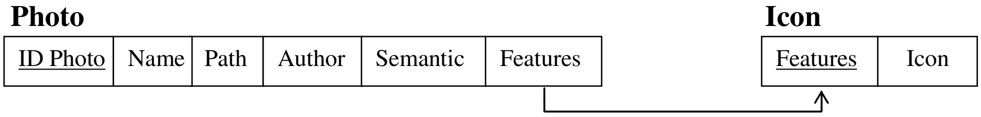
Photo

ID Photo	Name	Path	Author	Semantics	Features	Icon

The attribute *Semantics* abstractly represents a set of keywords describing the semantics of the photo, and their values have been automatically extracted by using the GCap tool [33]. With the attribute *Features* we abstractly refer to a subset of MPEG-7 features used to index the images for retrieval purposes [8]. At implementation level we have one attribute for each type of feature. For our experiment we have used the following three image features: *color* *layout*, *edge histogram*, and *scalable color*. The attribute *Icon* is an image characterizing the category to which each single image belongs. These attributes yield the following M-type functional dependency:

$$Features_{(g,\tau')} \rightarrow Icon_I \qquad (3)$$

where g is the composite function $g = c_1{*}d_1{+}c_2{*}d_2{+}c_3{*}d_3$, d_1 is the *Meehl index* [28], computed on the color layout feature, d_2 and d_3 are the *Pattern difference* functions [41], computed on the edge histogram feature and the scalable color feature, respectively; c_1,c_2, and c_3 are constants such that $c_1 + c_2 + c_3 = 1$; I is the *Identity* function. In our simulation we achieved best performances with respect to errors by setting $c_1 = 0.4$, $c_2 = 0.2$, $c_3 = 0.4$. The functional dependency highlighted above reveals that the relation schema *Photo* is not in 2MNF. The application of our normalization technique leads to the following relation schemas:



Photo

ID Photo	Name	Path	Author	Semantic	Features

Icon

Features	Icon

It is worth noting that when the database is populated it will be necessary to set the thresholds because they will affect the way data will be distributed across relations. Thus, we have produced several versions of the database by populating it through different thresholds. Then, we have performed twenty-five content-based queries on each version of the database, and on the non normalized one, by using the Query by Example paradigm on a common set of query images, randomly selected from the dataset. Effectiveness of results was evaluated by computing precision and recall measures [37] with a cutoff value of 50 on the ranked list of result images. We achieved a precision of 78% and a recall of 38% on the non normalized database, whereas the results for the different versions of the normalized database are plotted in Figure 1. In particular, the figure shows percent values of precision and recall measured on versions of the database generated by varying the threshold of the M-type functional dependency in the range 0.09-0.22.

In general, the normalized database trades-off between retrieval errors and errors due to manipulation anomalies, reducing the latter and increasing the formers. Since in the threshold range 0.09-0.18 the retrieval effectiveness of the normalized database is comparable to that of the non normalized one, we expect that in this interval the benefits of normalization overcome the drawbacks due to additional retrieval errors. On the other hand, for threshold values above 0.18 the additional retrieval errors induced by the normalization process start becoming more considerable, hence it will be necessary to evalulate the goals of the specific application context to decide the extent to which normalization is convenient. Finally, with threshold values below 0.09 there is no considerable variation in the retrieval performances with respect to the non normalized database.

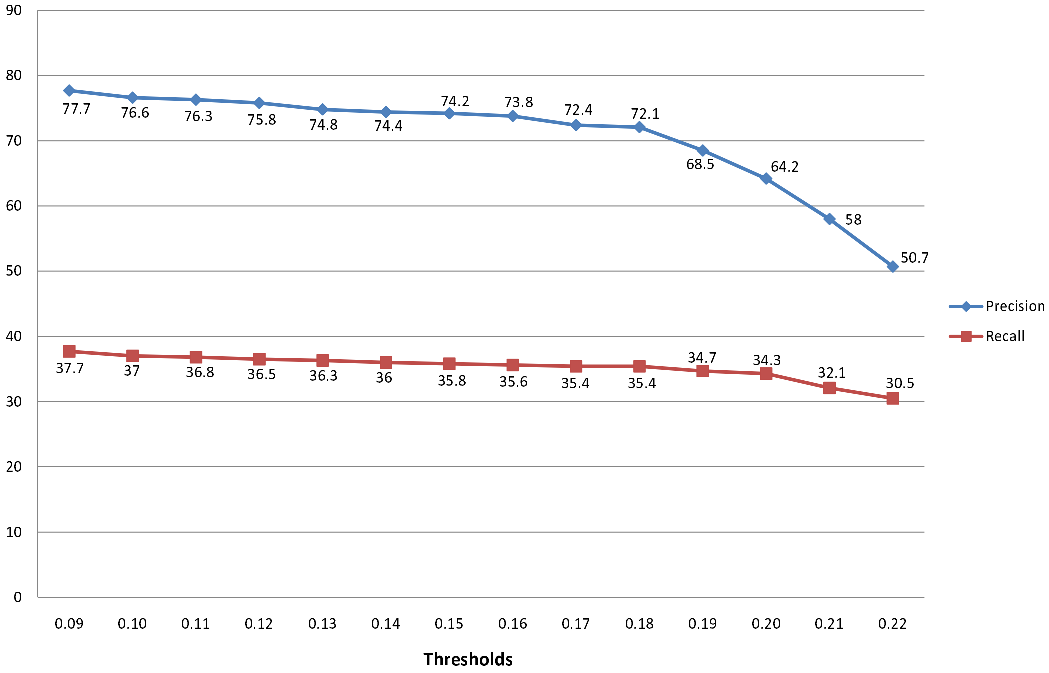


Figure 1: Precision and recall of the retrieval results with respect to threshold.

Obviously, the threshold also affects the size of the database and the average response time of queries, especially when the normalization process yields the splitting of relations. In fact, a higher threshold yields a lower number of tuples in the relation created by the splitting process. In our experiment we have used an algorithm for the *Disk Cover Problem* to select the pivot features to be inserted as tuples in the relation *Icon* [48]. Moreover, in order to make join operations more efficient we have performed them starting from the relation *Icon* and using a similarity join algorithm based on the grid-join algorithm proposed in [24]. Figure 2 shows how the average query response time changes by varying the threshold. Notice that with higher thresholds we gain reduced query response times. This is mainly due to the fact that we have less tuples in the *Icon* relation, which reduces the comparisons that are necessary to perform the join operations. Moreover, on the non normalized database we have observed an average query response time of 19.38 seconds. Thus, the average query response times observed on the normalized versions of the database in the above mentioned threshold interval 0.09-0.18 are close to the one of the non normalized database.

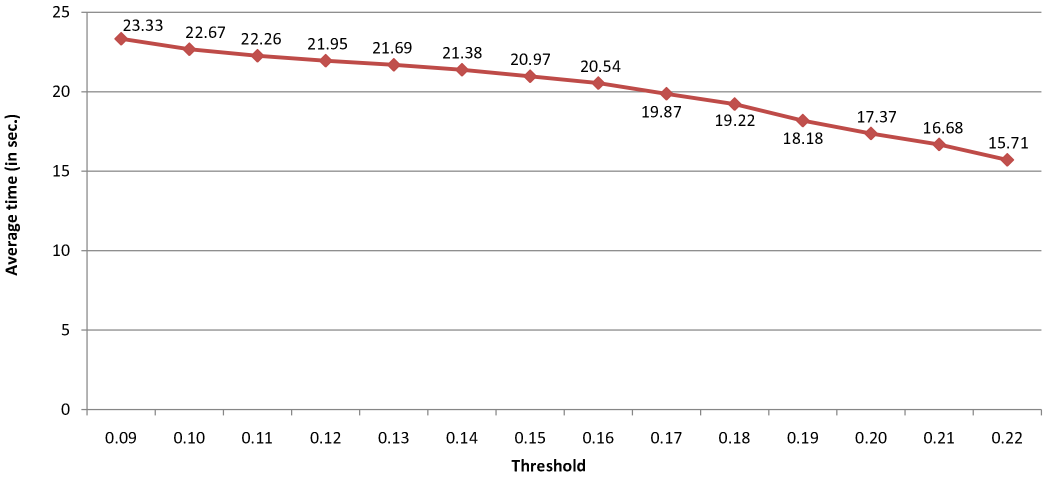


Figure 2: Trend of query response time with respect to threshold.

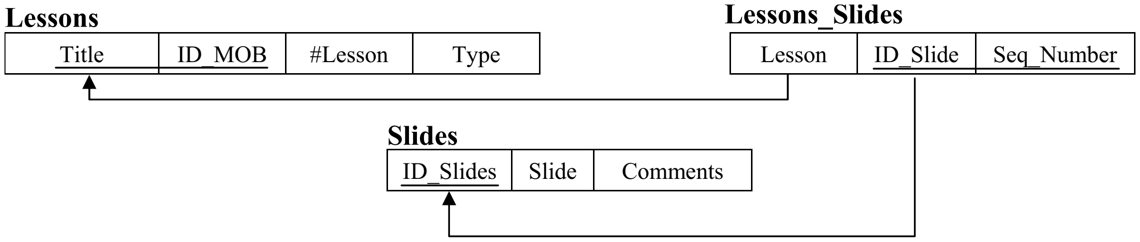
5.2 Evaluating the framework in an e-learning application

In this second experiment we aimed to analyze access performances to a multimedia database used in e-learning applications. The database stores data on lessons, such as Title, Speaker name, Speaker's photo, fingerprint (SFP), and finally the multimedia presentation of the lesson, which is stored at different levels of resolution. The following was the starting database schema:

Lessons

#Lesson	Title	ID_MOB	Type	Speaker	SFP	S_Photo	Presentation_Low	Presentation_Medium	Presentation_High

After the application of our framework we derived the following normalized database schema:



Then, we performed simulation experiments on an instance of the database containing twelve lessons, two instructors, and thirty students. We have divided the students into two groups based on the connection bandwidth they could use: dial up connection (56kbps), and DSL connection (640kbps). Then, we have performed several simulations by varying the mix of the two groups of students accessing the multimedia database simultaneously. For each simulation we have estimated the minimum, average, and maximum time needed by each group of students to access the lessons. Such parameters have always been computed twice, once on the initial database schema (whose size is 178 Mb), and once on the normalized schema (whose size is 163 Mb). In particular, the tuple selected from the initial *Lessons* schema has two types of presentations: the *Presentation_Low* attribute of 4.35 Mb, and the *Presentation_High* attribute of 11 Mb, whereas the tuple selected from the normalized schema has an *Audio* attribute of 1.27 Mb, a *Slide* attribute of 0.5 Mb, a *Video_Low* attribute of 2.63 Mb, and a *Video_High* attribute of 9.70 Mb.

The results of the simulation are summarized in Table 1. They show that in this case normalization enhanced access performances. Moreover, with the non-normalized database we gained worse average performances by increasing the number of students with high connection bandwidth. More precisely, we observed that if more than 20 students had DSL connection the performances of the DBMS decrease, mainly due to the fact that the database had to serve more requests simultaneously, whereas in other cases requests from slow connections could be served later. On the other hand, we observed that the normalized database allowed more than 20 fast connections before decreasing performances.

The histogram in Figure 3 shows how the average access time to the multimedia lessons changed by varying the "mix" of the two groups of students. In this case the normalized database provided lower average access times and smaller variations across different mix of

Non-Normalized Database

	Student's group mix		*Access Times*		
	$56kbps$	$640kbps$	*Min*	*Avr*	*Max*
Simulation 1	30	0	6.26	14.75	28.99
Simulation 2	20	10	2.97	8.09	16.89
Simulation 3	10	20	3.27	6.88	21.39
Simulation 4	0	30	4.40	8.45	12.29

Normalized Database

	Student's group mix		*Access Times*		
	$56kbps$	$640kbps$	*Min*	*Avr*	*Max*
Simulation 1	30	0	1.90	5.09	12.84
Simulation 2	20	10	1.09	4.97	9.64
Simulation 3	10	20	1.76	3.98	7.38
Simulation 4	0	30	1.15	3.74	7.78

Table 1: Simulation Results.

students.

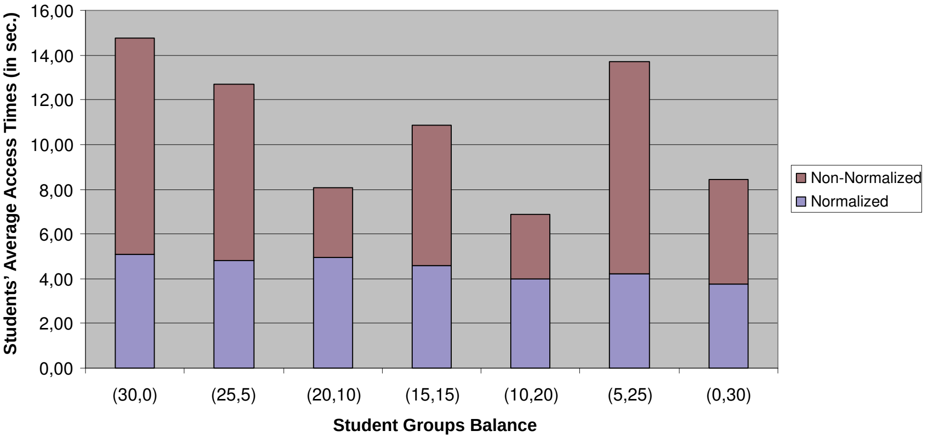


Figure 3: Histogram showing average access performances.

Further, we have observed higher performance gaps with bigger multimedia attributes. To this end we have performed further simulations to monitor the average access time with bigger multimedia objects. In particular, we have considered a non-normalized database of 869 Mb, whose normalized version is of 788Mb. Figure 4 shows performances gained on an entry of a non-normalized (normalized, resp.) database containing a *Presentation_High* (*Video_High*, resp.) attribute of 61 Mb (59.7 Mb, resp.).

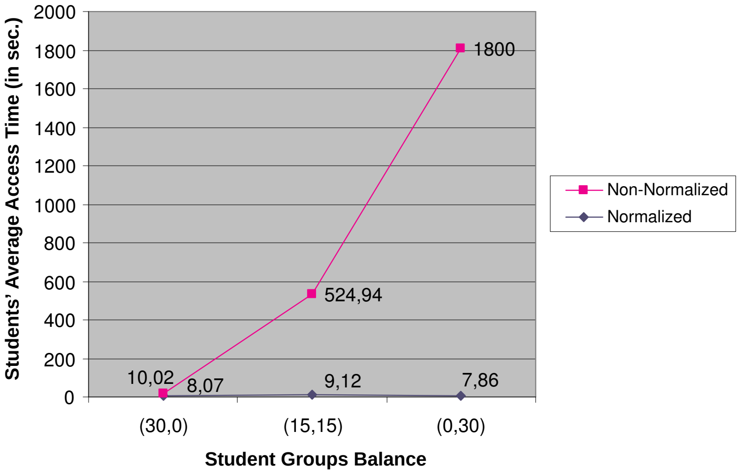


Figure 4: Performances comparison on a large multimedia object.

In conclusion, this experiment provides an application context in which it is necessary to continuously manipulate the multimedia database, hence it is more important to reduce manipulation anomalies. Moreover, the data reorganization induced by the normalization process has resulted more suitable for this specific context, because it has led to an improvement of access performances. Thus, we can conclude that in this application context the benefits of normalization have been more remarkable.

6 Discussion

In this paper we have proposed a normalization framework for multimedia databases. Our goal has been to derive proper design guidelines to improve the quality of multimedia database schemas. The framework provides designers with several means to let them derive the normalized database schema that is more suitable to the specific application domain. It is based on the concept of type-M dependency, which has been introduced to overcome some limitations of previous imprecise dependencies. We have also shown that type-M dependency generalizes previous ffds. The framework has been evaluated in the context of two important domains using multimedia databases, that is, content-based image retrieval and e-learning, for which we have provided experimental data to analyze several performance parameters. In particular, in the first application domain we have shown the threshold interval guaranteeing an appropriate compromise among retrieval effectiveness, manipulation anomalies, and time performances.

In practice, the detection of MFDs might not be a simple activity. In fact, by following the traditional approach of alphanumeric relational databases, the designer could detect type-M functional dependencies based on his/her knowledge on the application context. Alternatively, the detection of candidate type-M dependencies could be accomplished by finding feature correlations. The latter is a problem widely analyzed in the literature, and several techniques have been proposed, such as principal component analysis (PCA) [22], independent component analysis (ICA) [12], and so on. They are all based on training sets, and aim to identify independent features for indexing, querying, and retrieving multimedia data based on their contents. Thus, after performing this type of analysis, the designer knows the features having some correlation, between which there must exist a bi-directional dependency, valid for a large subset of tuples. This does not guarantee the existence of a type-M dependency, because the latter should apply to all tuples. However, the designer can still exploit the statistics collected for identifying feature correlations, since they provide him/her with a set of candidate type-M dependencies, but they require further validation.

The proposed approach also enables the designer analyze the impact of feature selection on database redesign and application development. In particular, when new features are added, the designer can understand whether they will lead to database redesign. As an example, in a multimedia relation $R = \{photo, name, \mathit{address}\}$, where the mfd: $photo_{(g_1,\tau_1)} \rightarrow \{name, address\}_{(g_2,\tau_2)}$ holds, suppose that *photo* can be in turn characterized by independent features such as *eye*, *nose* and *mouth*. In other words, three new features are selected. This can be incorporated into database design as follows. R can be replaced by $R_1 = \{photo, eye, nose, mouth\}$ and $R_2 = \{eye, nose, mouth, name, address\}$, on which mfd_1: $photo_{(g_1,\tau_1)}\rightarrow\{eye, nose, mouth\}_{(g_3,\tau_3)}$, and mfd_2: $\{eye, nose, mouth\}_{(g_3,\tau_3)} \rightarrow \{name, address\}_{(g_2,t_2)}$ hold, respectively. If *photo* and $\{$*eye*, *nose*, *mouth*$\}$ are well-behaved in the sense that *eye*, *nose* and *mouth* are independent features totally characterizing *photo*, then the above normalization can always be carried out. Moreover, by looking at the redesigned database schema, the designer can tell whether the application programs will be affected. In particular, s/he can contrast the redesigned schema against class and use-case diagrams of the whole application to precisely detect the application programs that the normalization step affects. Thus, our approach makes the design problem more a systematic and integrated multimedia software engineering activity.

The proposed framework is flexible enough to accommodate the requirements of different applications. In fact, since the multimedia dependencies and multimedia normal forms depend upon the tuple distance functions, by imposing additional constraints on tuple distance functions we can introduce more restricted multimedia dependencies and multimedia normal forms. For example, to support gesture languages in a virtual classroom for e-learning applications we can introduce different tuple distance functions to classify gestures as similar or dissimilar, leading to different protocols for gesture languages supported by the same underlying multimedia database.

Another important issue regards the normalization of multimedia databases in adaptive multimedia applications, where a media data may be replaced/combined/augmented by another type of media for people with different sensory capabilities. To this end, the normalization process yields a partitioning of the database that facilitates the management of adaptiveness.

References

[1] M. Arenas and L. Libkin. A normal form for XML documents. *ACM Trans. Database Syst.*, 29(1):195–232, 2004.

[2] P. Atzeni, S. Ceri, S. Paraboschi, and R. Torlone. *Database Systems: Concepts, Languages and Architectures.* McGraw-Hill, Inc., Hightstown, NJ, 1999.

[3] O. Bahar and A. Yazici. Normalization and lossless join decomposition of similarity-based fuzzy relational databases. *International Journal of Intelligent Systems*, 19(10):885 – 917, 2004.

[4] P. Bosc, D. Dubois, and H. Prade. Fuzzy functional dependencies - an overview and a critical discussion. In *Proc. 3rd IEEE International Conference on Fuzzy Systems*, pages 325–330. IEEE Press, 1994.

[5] CalPhotos. A database of photos of plants, animals, habitats and other natural history subjects. http://elib.cs.berkeley.edu/photos/, 2000.

[6] K. S. Candan and W. Li. On similarity measures for multimedia database applications. *Knowl. Inf. Syst.*, 3(1):30–51, 2001.

[7] E. Chang, K. Goh, G. Sychay, and G. Wu. Cbsa: content-based soft annotation for multimodal image retrieval using bayes point machines. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(1):26–38, 2003.

[8] S. F. Chang, T. Sikora, and A. Puri. Overview of the MPEG-7 standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(6):688–695, 2001.

[9] G. Chen, E. E. Kerre, and J. Vandenbulcke. Normalization based on fuzzy functional dependency in a fuzzy relational data model. *Information Systems*, 21(3):299–310, 1996.

[10] G.Q. Chen. Fuzzy functional dependencies and a series of design issues of fuzzy relational databases. In *Fuzziness in Database Management Systems*, pages 166–185. Physica Verlag, 1995.

[11] E. F. Codd. Further normalization of the database relational model. In R. Rusum, editor, *Data Base Systems*, pages 33–64. Prentice Hall, Englewood Cliffs, N.J., 1972.

[12] P. Comon. Independent component analysis a new concept? *Signal Processing*, 36:287–314, 1994.

[13] J.C. Cubero and M.A. Vila. A new definition of fuzzy functional dependency in fuzzy relational databases. *International Journal of Intelligent Systems*, 9(5):441–448, 1994.

[14] J. Davis. IBM/DB2 universal database: Building extensible, scalable business solutions. http://www-306.ibm.com/software/data/pubs/papers/db2udb/db2udb.pdf, February 2000.

[15] A. P. de Vries, M. G. L. M. van Doorn, H. M. Blanken, and P. M. G. Apers. The MIRROR MMDBMS architecture. In *Proc. of the International Conference on Very Large Databases*, pages 758–761, Edinburgh, Scotland, 1999.

[16] R. Elmasri and S. B. Navathe. *Fundamentals of Database Systems.* Addison-Wesley, Reading, MA, fifth edition, 2006.

[17] R. Fagin. Multivalued dependencies and a new normal form for relational databases. *ACM Transactions on Database Systems*, 2(3):262–278, 1977.

[18] R. Fagin. Combining fuzzy information from multiple systems. In *Proc. Fifteenth ACM Symp. on Principles of Database Systems*, pages 216–226. ACM Press, 1996.

[19] J. Fan, H. Luo, and A.K. Elmagarmid. Concept-oriented indexing of video database towards more effective retrieval and browsing. *IEEE Trans. on Image Processing*, 13(7):974–992, 2004.

[20] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker. Query by image and video content: The QBIC system. *IEEE Computer*, 28(9):23–32, sep 1995.

[21] A. Haghighi-Mood and J.N. Torry. Coherence analysis of multichannel heart sound recording. In *Proc. of Computers in Cardiology*, pages 377–380. IEEE Press, 1996.

[22] J. E. Jackson. *A User's Guide to Principal Components.* John Wiley & Sons, Inc., 1991.

[23] A. Jaimes and S.-F. Chang. Learning structured visual detectors from user input at multiple levels. *Int. J. Image Graphics*, 1(3):415–444, 2001.

[24] D. V. Kalashnikov and S. Prabhakar. Fast similarity join for multi-dimentional data. *Information Systems*, 32(1):160–177, 2007.

[25] M. J. Katz. Fractals and the analysis of waveforms. *Computers in biology and medicine*, 18(3):145–156, 1988.

[26] Michael S. Lew, Nicu Sebe, Chabane Djeraba, and Ramesh Jain. Content-based multimedia information retrieval: State of the art and challenges. *ACM Trans. Multimedia Comput. Commun. Appl.*, 2(1):1–19, 2006.

[27] M. Löhr and T. C. Rakow. Audio support for an object-oriented database-management system. *Multimedia Systems*, 3(6):286–297, 1995.

[28] P. E. Meehl. The problem is epistemology, not statistics: Replace significance tests by confidence intervals and quantify accuracy of risky numerical predictions. In L.L. Harlow, S.A. Mulaik, and J.H. Steiger, editors, *What if there were no significance tests?*, pages 393–425. Erlbaum, Mahwah, N.J., 1997.

[29] A. D. Narasimhalu. Multimedia databases. *Multimedia Systems*, 4(5):226–249, 1996.

[30] E. Oomoto and K. Tanaka. OVID: Design and implementation of a video-object database system. *IEEE Transactions on Knowledge and Data Engineering*, 5(4):629–643, August 1993.

[31] Oracle. Oracle 8i^{TM} release 2 features overview. http://www.oracle.com, November 1999.

[32] Vincent Oria, M. Tamer Ozsu, Paul J. Iglinski, Shu Lin, and Bin Yao. DISIMA: a distributed and interoperable image database system. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data (SIGMOD'00)*, page 600, New York, NY, USA, 2000. ACM Press.

[33] J.-Y. Pan, H.-J. Yang, C. Faloutsos, and P. Duygulu. Gcap: Graph-based automatic image captioning. In *Proceedings of the 4th International Workshop on Multimedia Data and Document Engineering (MDDE 04)*, 2004.

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

[35] M. Rennhackkamp. Extending relational DBMSs. *DBMS Online*, 10(13), 1997.

[36] R. Sacks-Davis, A. Kent, K. Ramamohanarao, J. Thom, and J. Zobel. ATLAS: A nested relational database system for text applications. *IEEE Transactions on Knowledge and Data Engineering*, 7(3):454–470, June 1995.

[37] G. Salton and M. McGill. *Introduction to Modern Information Retrieval.* McGraw Hill, 1983.

[38] S. Santini and A. Gupta. Principles of schema design for multimedia databases. *IEEE Transactions on Multimedia*, 4(2):248–259, June 2002.

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

[40] M. Sözat and A. Yazici. A complete axiomatization for fuzzy functional and multivalued dependencies in fuzzy database relations. *Fuzzy Sets and Systems*, 117(2):161–181, 2001.

[41] P. P. Sint. *Similarity structures and similarity measures.* Austrian Academy of Sciences Press, Vienna, Austria, 1975.

[42] M. Stonebraker and G. Kemnitz. The POSTGRES next-generation database management system. *Communications of ACM*, 34(10):78–92, 1995.

[43] P. Thagard. *How Scientists Explain Disease.* Princeton University Press, 2000.

[44] P. Thagard. Pathways to biomedical discovery. *Philosophy of Science*, 70:235–254, 2003.

[45] P. Thagard. What is a medical theory? In *Multidisciplinary approaches to theory in medicine*, chapter 4, pages 47–62. Elsevier, Amsterdam, 2006.

[46] M. W. Vincent, Jixue Liu, and Chengfei Liu. Strong functional dependencies and their application to normal forms in XML. *ACM Trans. Database Syst.*, 29(3):445–462, 2004.

[47] J. K. Wu, A. D. Narasimhalu, B. M. Mehtre, C. P. Lam, and Y. J. Gao. CORE: a content-based retrieval engine for multimedia information system. *Multimedia Systems*, 3(1):25–41, 1995.

[48] B. Xiao, J. Cao, Q. Zhuge, Y. He, and E. Hsing-Mean Sha. Approximation algorithms design for disk partial covering problem. In *Proc. of the 7th International Symposium on Parallel Architectures, Algorithms, and Networks*, pages 104–110. IEEE Computer Society Press, 2004.

[49] L. Zadeh. Fuzzy sets. *Information and Control*, 8:338–353, 1965.