Multimedia and Metadata Watermarking Driven by Application Constraints

Richard Chbeir
Computer Science Department
LE2I - Bourgogne University
BP 47870
21078 Dijon CEDEX France
rchbeir.@.u-bourgogne.fr

David Gross-Amblard*
Laboratoire CEDRIC – CC 432
Conservatoire national des arts et métiers
292 rue St Martin
75114 Paris CEDEX 3 – France
dgram.@.cnam.fr

Abstract

Providing a fully functional multimedia DBMS (MMDBMS) becomes an emergency with the recent development of distributed environments. In this paper, we address the impact of using watermarking techniques traditionally used to preserve the Intellectual or Industrial Property (IIP) in MMDBMS.

Through a multimedia content and metadata based representation model called M^2, we particularly study: 1) how to watermark all components of a multimedia description, and not only its raw data 2) how watermarking can guarantee the mapping between multimedia objects and their descriptors, avoiding accidental or malevolent mismatch inside crucial documents 3) how to preserve data significance and semantics when altering data for watermarking purposes. We illustrate our approach by providing an example in the medical domain.

1 Introduction

The need for a full fledge multimedia DBMS becomes more apparent when one considers processing environments (such as satellites, surveillance, medical applications, etc.) in which complex multimedia objects are produced massively everyday, shared on demand, and replicated over several sites. To provide multimedia-oriented functionalities, preserve their authenticity, and meet the growing demands for efficient processing of the vast quantities of data in DBMS, documents must respect the following storage process:

- *Multimedia Document Description*: a document should not be considered anymore as a black box. This step allows describing documents by several types of low-level features (colors, textures, shapes, etc.) and metadata. This is vital for multi-criteria queries that use both content-based and metadata representation of multimedia. For example, in a firm time management application, we stock in an `EMPL` table the employees names, addresses, and images and in an `ENTRANCE` table, the video captured by a monitoring camera at different times. A multimedia-join operation between the two tables can then be used to determine the name of employees entering (or leaving) the firm at a given time.
- *Multimedia Access and Intellectual/Industrial Property (IIP) Control*: a multimedia document contains rich content related to the document itself or to its metadata. This step must first provide content-based and related data access mechanisms against document ownership usurpation or falsification. It should e.g. assist legitimate owners of a high quality annotated document in ownership proofs, whenever a suspected copy is discovered. Second, it must guarantee both the mapping between a document and its description, to prevent an accidental or malevolent mismatch. Finally, this should be feasible even when the documents or descriptions are slightly altered (e.g. by scanning, printing, etc.), voluntarily or not.

In this paper, we address these two issues. First we present a multimedia metadata model called M^2, to support the design of efficient multimedia meta-database model able to improve multimedia management. The goal is to provide a modeling framework to express the properties of data items and the metadata that are necessary for organizing multimedia management systems at different levels. Built on the relational-object paradigm, our multimedia meta-database model is independent of (but compatible with) all current data format models (MPEG-4, MPEG-7, etc.) The key feature of the model is that it captures in a single concept the low-level features, the structural and semantic properties, and the relationship descriptions of both multimedia and meta-object.

Second, we study how to access multimedia data and control their IIP using watermarking techniques [1, 2]. Thanks to their ability to hide information into a document, watermarking techniques will permit to identify and tag both documents and descriptions in our M^2 model. This will allow the DBMS to assist a legitimate owner in ownership proofs when a suspect document is discovered, and to guarantee the mapping between data.

*Author supported by the ACI Sécurité & Informatique TADORNE grant (2004-2007)

Finally, applying watermarking techniques should not alter the meaning of initial documents in sensitive application domains. We detail how to express application domain constraints in the M^2 framework, and how watermarking techniques can be integrated in our proposal according to these constraints.

Our goal here is certainly not to provide yet another watermarking method for a specific data-type: we define a formal framework that helps watermarking techniques to be applied on various data-types while respecting formal domain-oriented constraints.

The paper is organized as follows. In Section 2, we give a medical case study that we will use throughout the paper to explain our proposal. Section 3 is dedicated to present the M^2 model, while Section 4 presents which components should be watermarked in M^2 and how domain constraints are defined. In Section 5, we explain how to watermark multimedia data using M^2, and Section 6 addresses the preservation of sensitive components. Section 7 provides the related work and Section 8 concludes.

2 Motivation

In this paper, we address the medical application domain where several and different multimedia data are produced and stored each day. We give here an example of a lungs X-ray associated with descriptors of various types (volume, shapes, pressure, etc.) Fig. 1 shows the X-ray image, the related salient objects, a textual medical record and a table of blood pressure and temperature measurement (i.e. a relational table).

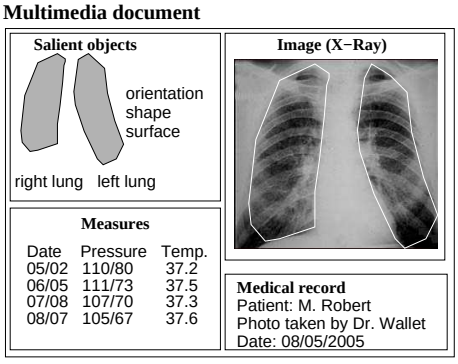

Figure 1: Case study in medical domain

As for several application domains, the medical document follows a specific workflow. It is used by several health professionals (radiologist, physician, insurance, secretariat, etc.) and for different aims. During its life cycle, the medical document may be transformed or altered. A first major concern for professionals is then to guarantee the authenticity, significance, and integrity of the medical document.

Example 1 *A physician has queried the system to obtain the whole description of Mr. Robert's X-rays. After retrieval, he extracted only the image taken by Dr. Wallet for its use in another application. He then printed out the X-ray image in order to submit it to another specialist. After submission, this specialist has mixed up his documents and lost the corresponding X-ray patient's identity. The problem is then, given a printed document, how to infer the identity of the patient and the physician, or how to find all related components.*

A natural solution is to print out the image with the interlaced patient's and document identifiers at a fixed position (e.g. upper side corner). Nevertheless, this solution is not resilient. The related information may disappear with simple common modifications applied to the image (e.g. cropping, smoothing, compression, etc.) Moreover, identifiers can easily be changed by a malevolent "colleague".

A possible solution is the use of watermarking techniques, that allow a pervasive embedding of any information in the multimedia data. Such techniques slightly modify pixels gray-scale in a X-ray, with a little impact on the image quality. The scalability of this technique allows, for instance, extracting the patient id from either the original watermarked image, a compressed version of this image, or even a scanned image. Watermarking can also be applied to other parts of the the multimedia object (textual data, features, relations, numerical tables, etc.)

However, in order to preserve the significance of the resulted image, watermarking operations should be performed with care. This means that *hiding information* in the X-ray image *should not be contradictory with other important data (e.g. medical diagnosis).*

Example 2 *Assisted chest radiography is a typical example of computer-aided diagnosis (see e.g. [3]). A classical processing workflow of chest X-rays includes* image enhancement*,* edge detection*, and* shape classification*. Fig. 2 shows a simplified processing workflow: on the X-ray image (upper left), an edge detection procedure has been applied to determine lungs borders (lower left). On the detailed view (right), four round structures labeled $C1$ to $C4$ can be observed.*

Figure 3 shows the same processing workflow, but on the watermarked X-ray image*, using a popular watermarking plugin. It is noteworthy that the watermarking operation has altered the edge detection process, and that round structures changed with respect to the previous image of Fig. 2. For example, structure $C1$ and $C2$, previously* disjoint*, are now in touch. Structure $C3$ has disappeared, and structure $C4$ has now only one hole, instead of two previously. These differences may impact the computer-aided diagnosis system.*

As we can see, several issues need to be studied before applying watermarking techniques, mainly:

- *what to watermark*: it is necessary to embed information in the whole document, the multimedia object itself and related descriptions.
- *how to watermark*: it is also important to apply appropriate watermarking techniques, so that application domain constraints are preserved.

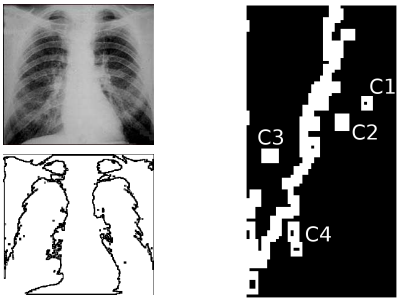

Figure 2: Edge detection on the original X-ray

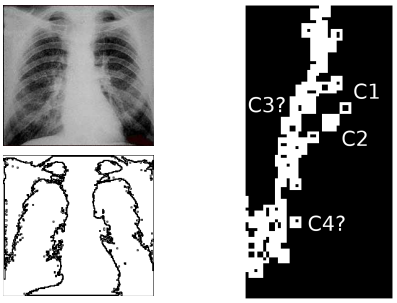

Figure 3: Edge detection on **watermarked** X-ray, showing alterations

Hence, one should devise watermarking techniques that respect and preserve several salient properties. These constraints depend highly on the application domain. One can expect a large number of such constraints in medical applications [4, 5] where only few ones would be sufficient in touristic pictures applications. For this reason, the multimedia DBMS should allow the domain expert to easily express constraints to be preserved by watermarking techniques. These constraints should be designed either between the multimedia objects (and sub-objects) or between the multimedia object and its description. For instance, several properties should not be altered, others should be altered slightly, and others can be modified without any restriction.

3 A multimedia description model

Looking at the case study given above, we can easily observe that a representation model able to describe all multi-dimensional information related to a multimedia object is required. Below, we explain the M^2 model for structuring the meta-database of multimedia DBMS. The proposed model is built on relational-object paradigm in order to be able to consider both relational and object-oriented DBMS. It can also be used on XML-based DBMS.

3.1 Definition

The multimedia model M^2 extends a previous repository model [6]. In [6], the authors address the management of image databases by providing an algebra where SQL and image-oriented operations can be written. In M^2, we aim to address any multimedia object by providing the concept of a meta-object. A meta-object has a set of properties used to capture the descriptions of a multimedia object at different levels of description and can be related to other meta-objects via one or more relationships. The representation $M^2(id, O, F, A, R)$ of a meta-object consists of:

- id: a unique identifier associated to a meta-object.
- O: a set of references to the raw data of the object (or the file). For complex multimedia data, O is the actual (image, video, or audio) object file which can be stored as `BLOB`. For set oriented data, O is an index for the data structure used to store the elements of the set.
- F: a feature vector representation of the object O. This component contains the physical, visual, spatial and temporal feature data value (color histogram, format, content descriptors, etc.)
- $A(ES, Sem_F)$: contains metadata and semantic feature. The structure of its subcomponents is domain-oriented. ES is the External Space descriptions (either context-oriented, domain-oriented or multimedia oriented). $Sem_F(Type, Description)$ is the type of the semantic feature and its textual description.
- $R(\{(S_1, S_2, Re)\})$, where:
  - $S_1 = \{id_{i|i=1..n}\}$, $S_2 = \{id_{j|j=1..m}\}$;
  - $Re = \{Rel_{k|k=1..p}\}$.

  This component represents zero or more relationships between objects. The description of each relationships consists of the set of the identities of objects participating in the relation, and the relation itself (either spatial (directional, metrical, topological), semantic, temporal, or similarity relation).

3.2 Application

Using the proposed multimedia meta-database model M^2, either multimedia static object (e.g. image), dynamic object (e.g. movie), or a set (or a table) of media objects can be represented in the DBMS. Below, we give the content of each attribute in M^2 when representing the case study given in Section 2. As we will see, the R component of M^2 plays a major role.

Let us study the image and its salient objects appearing in Fig. 1. The hierarchical relations between objects in M^2 are represented by a N-ary tree where the root represents the entire image and where each node is a salient object having one or several outgoing edges (see Figure 4). In Figures 5 we give some content values of the image.

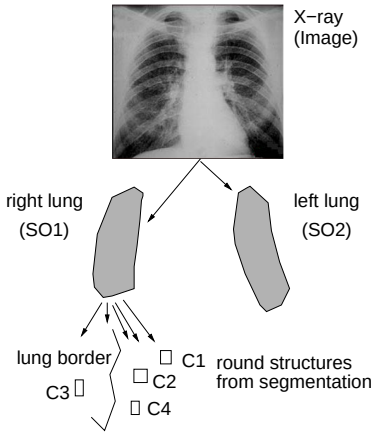

Figure 4: Hierarchy of objects

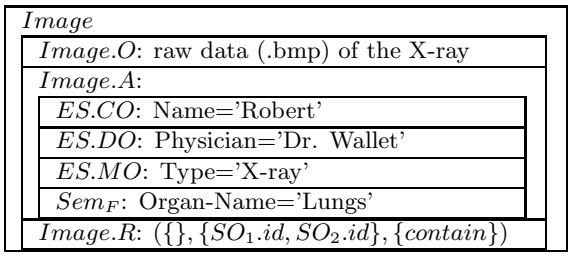

Figure 5: M^2 description of $Image$

4 Expressing application domain constraints in M^2

As M^2 supports representing of both multimedia data and corresponding multi-dimensional description, it is easy then to watermark any of its components. However, as shown in Section 2, data alteration should be achieved according to several application domain constraints. In order to consider these constraints in our approach, we extend $M^2(id, O, F, A, R, S)$ by integrating a sensibility component S. Each atomic value in S belongs to the interval $[0, 1]$: 0 when the related property is to remain unaltered, 1 when there is no restriction. S is composed of several components:

- S_O: identifies the alteration threshold inside the O raw-data. When S_O=0, neither the object O nor its sub-objects O^i must be altered. This introduces the notion of dependency between components as we will see in the next section.

- S_F: contains several values where each value identifies the alteration of one feature. Several low-level features (such as volume, color histogram, etc.) constraints induce some constraints on other components.

- S_A: allows to define alteration possibilities on semantic features.

- S_R: defines the alteration on relationships between multimedia objects and other sub-objects. In our example, the topological relationships between salient objects should not be altered.

It is important to note that the *id* component must remain unaltered.

5 Watermarking issues in sensitive multimedia documents

5.1 General watermarking framework

Basically, a watermarking procedure is defined by:

1. A watermarking algorithm $\mathcal{W}$, whose aim is to hide a message m (a watermark) in a document. The message to hide can include the document identifier, owner or group of owners, or any information related to the application.

2. A detection algorithm $\mathcal{D}$, that extracts, given a watermarked document, the hidden message m.

Algorithms $\mathcal{W}$ and $\mathcal{D}$ use a secret key $\mathcal{K}$ as a parameter, known only by the legitimate owner. They must respect the following criterias:

- **Basic invisibility**: the alteration performed by the marker $\mathcal{W}$ should not alter the quality of the document.

- **Soundness**: the detector $\mathcal{D}$ should extract the message hidden by $\mathcal{M}$ correctly.

- **Robustness**: the detector should work even if the watermarked document has been voluntarily or accidentally altered, up to a reasonable distortion.

- **Specificity**: the detector should not discover false messages from neither third party nor non-watermarked documents.

- **Key-based**: the detection can not be realized without this secret key $\mathcal{K}$.

- **Strong invisibility**: when watermarking several M^2 components simultaneously, the alteration performed by the marker should not break sensitive relationships between these components.

The first five constraints are basically achieved by traditional watermarking methods (see Section 7), while the last one, *strong invisibility*, has not been addressed yet in the literature and will be studied here. In order to describe our global architecture, we assume the existence of a generic watermarking method $\mathcal{W}$, so that, for any simple component $C \in \{O, F, A, R\}$, any message m and any secret key $\mathcal{K}$, the call to $\mathcal{W}(C, \mathcal{K}, m)$ will produce a watermarked component C_m:

$$\mathcal{W}(C, \mathcal{K}, m) = C_m \quad \text{where } C \in \{O, F, A, R\}.$$

However, watermarking a complex component C, with $C = (C^1, \ldots, C^n)$, requires to apply a watermarking algorithm on each sub-object or related object:

$$\mathcal{W}(C, \mathcal{K}, m) = (\mathcal{W}(C^i, \mathcal{K}, m_i))_{i=1,n} = (C^i_{m_i})_{i=1,n}.$$

where C and $C_i \in \{O, F, A, R\}$, and each C_i is a sub-object of C (i.e. C_i *Rel* C, where *Rel* represents any relationship between two components). Message m_i is the message to be hidden in each component (as we will see, this is not the same message for all components).

Conversely, we rely on an abstract detector $\mathcal{D}$, so that $\mathcal{D}(C_m, \mathcal{K})$ returns the hidden message m in a watermarked component C_m, or fails. Observe then that $\mathcal{D}(\mathcal{W}(C, \mathcal{K}, m), \mathcal{K}) = m$.

According to watermarking standards, the watermarked component C_m is reasonably robust against natural transforms or malevolent attacks. Hence, given $\tilde{C}_m$, a malevolent alteration of C_m, the probability that $\mathcal{D}$ outputs m given $\tilde{C}_m$ and $\mathcal{K}$ as input is high. Similarly, given a non-watermarked component C, the probability that the detector does not fail is small.

5.2 Applications of watermarking in MMDBMS

In the classical watermarking literature, the following applications are considered on a unique, single document (e.g. an image). We present here how to achieve these applications on a whole document M^2, i.e. with multimedia and metadata parts, and show the interplay between these different parts.

Adding ownership information The copyright string is encoded into the message m, and possibly encrypted so that only the DBMS manager and the document's owner can read it. The message m is then watermarked into each component of the $M^2(O, F, A, R)$.

Given a suspected altered component $\tilde{C}$, the DBMS manager or the legitimate document owner applies the detector to obtain $\tilde{m} = \mathcal{D}(\tilde{C}, \mathcal{K})$. If the alteration of the component is reasonable (according to a watermarking standards), $\tilde{m}$ is likely to be equal to m, and the copyright can consequently be revealed.

When a suspect M^2 model with several components $C^1, \ldots, C^k$ is discovered, the detector is recursively applied to each component, giving sub messages $\tilde{m}_1, \ldots, \tilde{m}_k$. The original message m is extracted with high probability by performing a majority voting on each bit of messages $\tilde{m}_1, \ldots, \tilde{m}_k$.

Embedding an M^2 component into another Since any component in M^2 is basically a binary string m, it can be encoded as a message m for watermarking. This allows watermarking techniques to hide a component into another component in a persistent manner. For instance, one would like to hide A component (the patient's personal information) into O (the X-rays). For this, the message m_A, encoding the patient's informations in A, can be watermarked in O as follows: $O_{m_A} = \mathcal{W}(O, \mathcal{K}, m_A)$.

A physician can then print out the watermarked raw image O_{m_A} and distribute it to another specialist. This latter may scan the printed version of the image, and obtain $\tilde{O}$. With a high probability, the embedded description A can be recovered by a call to the detector: $A = m_A = \mathcal{D}(\tilde{O}, \mathcal{K})$. Instead of embedding all component data into another component, it is better to create a link (or use an existing one) with the watermark component. Only a URI (or the identifier) of a component is then hidden into another. This solve the problem of Example 1.

Detecting descriptions mismatch A third application of watermarking is to maintain the consistency of data and to certify that all components within a multimedia object M^2 correspond. An alteration can be voluntarily introduced in the system by a malicious user, in order to discredit another user or the overall system. For example, the lungs X-ray of a healthy patient can be stored in the database inside a sick patient record in order to discredit the diagnosis. Certification is done using hashing of the components by the following function:

$$\text{for } C \in \{O, F, A, R\}, h_C = hash(C||\mathcal{K}),$$

where $||$ denotes concatenation and $hash(C)$ is a shortcut for the hashing of the binary representation of component C. The *hash* function can be any known one-way hash functions like `SHA` or `MD5`. Numbers h_C are typically a few bits long. We also compute the complete signature $m_S = h_O \# h_F \# h_A \# h_R$, where $\#$ is a special character as separator. Then, in order to maintain consistency between data, we watermark each object M^2 as follows:

$$M^2_{m_S} = \mathcal{W}(M^2, \mathcal{K}, m_S).$$

Hence m_S is spread in all components. When a component $C \in (O, A, F, R)$ is downloaded from the DBMS, a user can perform the following operations:

- check the authenticity of one component C_0: we consider a component R_0 as an example. By detecting the hidden message $m_S = \mathcal{D}(R_0, \mathcal{K})$, the user is able to recover the signature of the original document $m_S = h_O \# h_F \# h_A \# h_R$. By extracting h_R, he can compare its value with the signature of the component R_0, by computing $hash(R_0||\mathcal{K})$. If they do not correspond, i.e. if the signature of the original R component (hidden in the whole M^2) and the signature of the discovered R_0 differ, the authenticity of the R_0 component can be declared suspect.
- check the correspondence between components: if for example only A and R components are available, h_S and then h_R can be extracted from A. Hence, h_R and $hash(R||\mathcal{K})$ can be compared. If

they differ, the user can conclude that A and R do not correspond. The same operation can be followed for the symmetric case.

6 Watermarking while preserving sensibility factors

Up to now, we did not consider the *strong invisibility* constraints; we simply applied known watermarking techniques on components without considering their relationships. However, we may specify which components or specific objects in the M^2 description model should be impacted or not by the watermarking process.

Considering our medical example, several parts must be preserved from watermarking, particularly the X-ray texture and the topological relationships between salient objects of the X-ray. For example, the `disjoint` relationship between C_1 and C_2 should not be impacted. In order to protect the X-ray texture, one must specify a correct sensibility value S_O on the image component O, as introduced in Section 4. For a given visual quality metric, like signal to noise ratio, setting $S_O = 0.3$ limits watermarking alterations to 3 dB (Peak Signal-to-Noise Ratio (PSNR) alteration). Similarly, to preserve the topological relationships between shapes or salient objects in the image, we should set their corresponding S_R components to zero.

It is obvious that *strong invisibility* constraints on one component can impact all other M^2 components: if we watermark the X-ray (component O), segmented objects C_1 and C_2 may change and topological relationships may be lost. We now focus on watermarking methods that handle such watermarking with *strong invisibility* constraints.

Algorithm 1: $\mathcal{W}(C, S, \mathcal{K}, m)$ // Brute-force method

Input: component C, sensitivity S, message m, key $\mathcal{K}$
Output: watermarked component C_m, or $FAIL$

```
for α from 1 downto 0 step 0.1 do
    C_m=W(C, K, m, α) // try to hide m in C
                      // using strength α
    if (S is checked on C_m) then
        return C_m and exit;
    end
end
return FAIL // the component cannot be watermarked
```

Brute-force method Watermarking methods often use a *strength* factor $\alpha \in [0, 1]$, that controls the impact of watermarking on the document quality. For image watermarking methods, if the strength is small, the visual impact is low and the watermark is less resilient to malicious attacks. Conversely, a huge strength implies a robust watermarking, but the image quality and usability are reduced. Hence, a basic method for watermarking while respecting sensitivity factors is to iteratively apply the watermarking algorithms with variable strength, and to check the sensibility (Algorithm 1).

This method is general. However, its major drawbacks are its time-consuming and brute-force aspects: it is not tailored to respect constraints in $\mathcal{S}$.

Sensibility-driven watermarking method As discussed before in our case study, watermarking the raw document O should preserve topological constraints on salient objects, as specified in the sensibility component S. We will first present an abstract settings to solve this problem, then we will illustrate the method using our case study. We rely on the following notions: (1) a description of the relationships between the various components of an M^2 object, (2) sensibility annotations on each important component, and (3) a set of sensibility driven watermarking algorithms.

Given an M^2 object with n components $\{C^1, \ldots, C^n\}$, we consider its *annotation dependency graph*. This graph depicts explicitly what are the relationships between components, specially when one component is *computed* from another one. This graph is linked to the intended application domain, and differs from one application to another.

More formally, let $G = (V, E)$ be a directed graph, where the set of vertices V is the set of components $\{C^1, \ldots, C^n\}$, and the set of edges E are of the following form:

$$C^i \overset{m_{ij}}{\rightarrow} C^j,$$

where m_{ij} is an *annotation method*.

The annotation dependency graph asserts that component C^j is computed by method m_{ij} from component C^i. The method m_{ij} may be manual or semi-automatic. Illustrated in our example, the salient objects C_1 and C_2 are obtained from segmentation of the raw image O by a classical automatic segmentation method m_{OF} (e.g. [7]). The topological relationships in R are obtained from regions C_1 and C_2 by an automatic topological decomposition method m_{FR}.

$$O(\text{X-ray}) \overset{\text{segmentation}}{\underset{m_{OF}}{\rightarrow}} C_i \overset{\text{topological analysis}}{\underset{m_{FR}}{\rightarrow}} R(\text{topology}).$$

Figure 6: Annotation dependency graph for X-ray medical applications

A *sensibility annotation* on G is simply the restriction of the S part of M^2 on each component. In our example, there is a sensibility annotation on the `disjoint` relation between objects C_1 and C_2:

relationships	sensibility
C_1 `disjoint` C_2	0
C_1 `disjoint` C_3	1

These relationships are boolean, hence their sensibility is either "modification allowed" (value 1) or

"no modification at all" (0). Here, the relationships between C_1 and C_2 should absolutely be respected, but relationships between C_1 and C_3 are not as much important.

Sensibility annotations on a component may impact other components. This can be discovered by traversing the graph backward. For example, the fact C_1 `disjoint` C_2 appears in the R component, and is annotated as sensitive. By traversing the dependency graph backward, we discover that this fact is due to the relationships between two salient objects, C_1 and C_2. Hence these two objects are also sensible. Finally, going backward one step further, the raw X-ray image is also annotated as sensible.

Let $C^i \overset{m_{ij}}{\rightarrow} C^j$. A *preservation strategy* p_{ji} allows to translate a sensibility annotation S_j on C^j to a new sensibility annotation S_i on C^i. Clearly, the preservation strategy p_{ji} and the watermarking method m_{ij} are closely related. In our example, preserving the topological relationships between C_1 and C_2 induces preserving the gray-scale distribution of C_1 and C_2.

Our general watermarking method is expressed in Algorithm 2: for the intended application, the annotation dependency graph is computed once for all. Given an M^2 model, the graph is labeled with the corresponding sensibility annotations. Then the preservation strategies are recursively applied. Once closure is reached, all the sensitive components are labeled.

Algorithm 2: $\mathcal{W}(M^2, G, S, \mathcal{K}, m)$ // Sensibility-driven algorithm

Input: M^2 model, dependency graph G, sensibility S, key $\mathcal{K}$, message m
Output: watermarked model M^2, or $FAIL$
1 label G with sensitivity annotations in S
2 recursively apply preservation strategies on G
3 **for** *each component C to watermark with m,$\mathcal{K}$* **do**
4 | Call $\mathcal{W}(C, S_C, \mathcal{K}, m)$
5 **end**
6 **return** the new M^2 model

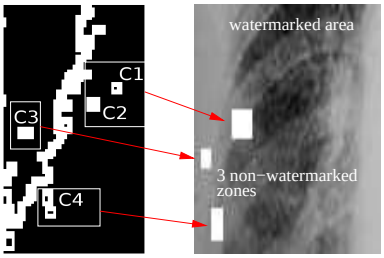

Figure 7: Automatic relevant stencil computation for original image masking

Real example – the stencil algorithm We end this section by giving the complete sensibility driven algorithm for our medical example. Our algorithm, that we called the *stencil algorithm*, rely on the annotation dependency graph of Figure 6. First, the set of important topological relationships are identified (the `disjoint` between C_1 and C_2 in our example). Then these constraints are propagated to the important salient objects (C_1 and C_2). Finally, the set of fragile gray-scale pixels are identified in the raw X-ray O. These pixels are then protected by *stencils*, i.e. regions of the image that should not be altered by watermarking. Given these stencils, the watermarking algorithm for O simply acts as usual, but never introduces any modification on pixels under a stencil. This method clearly leads to a watermarked image where salient objects and topological relationships do not differ from the original one. This solve the problem of Example 2.

Of course, the occurrence of stencils reduces the watermarking space. However, a wide number of pixels are still available for watermarking. Similarly, when detecting a document that has been watermarked using stencils, one should observe that the detector does not know positions of stencils. Hence, watermarked pixels and un-watermarked pixels under a stencil are not distinguished. However, the lack of watermarked pixels can be analyzed / is similar to an attack that voluntarily alters the watermark at stencils positions. Hence, using any attack-resilient watermarking technique resists to the occurrence of a reasonable number of stencils during the watermarking process.

7 Related Work

A lot of work has been done to increase the efficiency of multimedia management in DBMS [8, 9, 10]. Early research in multimedia data processing has been carried out separately in the database and computer vision communities. The database approach focuses on metadata management and semantic-based annotations for storage and retrieval of multimedia data. This approach has several inadequacies as it is time-consuming, subjective, and cannot adequately describe the content of multimedia data [9, 11]. The computer vision approach has addressed content-based issues such as features extraction, information coding, lossless data compression, image segmentation. This approach is based on low level features [12, 13, 10]. To integrate the two approaches, several research activities have focused on defining new representation formats and standards allowing the description of multimedia data through several dimensions (e.g. MPEG-4, 7 and 21). However, their adaptability for traditional DBMS remains difficult and requires major internal modifications. This is why a DBMS oriented-model is required for multimedia multi-facets data representation.

Traditional DBMS-oriented access control techniques are also no longer appropriate and should be extended. One of the important issues to be studied is the Intellectual or Industrial Property (IIP) protection particularly addressed by the MPEG-21. A wide pool of watermarking methods exists in the literature to preserve the IIP by hiding information in each datatype, including still images, sound files, video, MPEG-4 extensions, geometrical objects, textual data and nu-

merical datasets [1, 14, 15, 16, 17, 18, 19, 20, 21].

To the best of our knowledge, only few watermarking techniques consider relations between documents, and spread the message to hide over several data-types et not only over one specific type. In [22], the authors study how watermarking medical images without impacting the corresponding diagnosis. The method consists of defining an exclusion zone where watermarking is not to be performed (basically an ellipse). However, the provided method is not fine-grained (only one zone can be defined) and does not support all possible constraints such as the relations between salient objects.

8 Conclusion and future work

In this paper, we described a multimedia DBMS-oriented description model called M^2 allowing to pinpoint and represent the sensitive information and features contained in a multimedia document. Using this model, we explored how watermarking can be applied on multimedia objects in order to enforce their security and improve reliability in a DBMS. We demonstrated that watermarking could not be applied without considering and preserving several application domain constraints. We also showed that watermarking should consider linked components in a description model. For all theses reasons, we proposed a general watermarking framework allowing constraints preservation and propagation within the whole multimedia document.

In order to study the efficiency and limits of our proposal, we are currently implementing our approach and studying how to integrate effective preservation strategies.

References

[1] I. J. Cox, M. L. Miller, and J. A. Bloom, *Digital Watermarking*. Morgan Kaufmann Publishers, Inc., 2001.

[2] S. Katzenbeisser and F. A. P. Petitcolas, Eds., *Information hiding: techniques for steganography and digital watermarking*, ser. Computer Security Series. Artech house, 2000.

[3] B. van Ginneken, B. M. ter Haar Romeny, and M. A. Viergever, "Computer-Aided Diagnosis in Chest Radiography: A Survey," *IEEE Transactions on Medical Imaging*, vol. 20, no. 12, December 2001.

[4] R. Chbeir, Y. Amghar, and A. Flory, "A prototype for medical image retrieval," *Methods of Information in Medicine*, vol. 3, pp. 178–184, 2001.

[5] R. Chbeir, S. Atnafu, and L. Brunie, "Image data model for an efficient multimcriteria query: A case in medical databases," *International Conference on Scientific and Statistical Database Management, IEEE Computer Society Press*, vol. 3, pp. 165–174, 2002.

[6] S. Atnafu, R. Chbeir, D. Coquil, and L. Brunie, "Integrating similarity-based queries in image dbms," *ACM SAC*, pp. 735–739, 2004.

[7] M. Borga, H. Malmgren, and H. Knutsson, "FSED - feature selective edge detection," in *Proceedings of 15th International Conference on Pattern Recognition*, vol. 1. Barcelona, Spain: IAPR, September 2000, pp. 229–232.

[8] A. Yoshitaka and T. Ichikawa, "A survey on content-based retrieval for multimedia databases," *IEEE Transactions on Knowledge and Data Engineering*, vol. 11, no. 1, pp. 81–93, 1999.

[9] Y. Rui, T. Huang, and S. Chang, "Image retrieval: Past, present, and future," *Journal of Visual Communication and Image Representation*, vol. 10, pp. 1–23, 1999.

[10] W. I. Grosky, "Managing multimedia information in database systems," *Communications of the ACM*, vol. 40, no. 12, pp. 72–80, 1997.

[11] J. P. Eakins and M. E. Graham, *Content-Based Image Retrieval: A Report to the JISC Technology Applications Program*. Inst. for Image Data Research, Univ. of Northumbria at Newcastle, 1999.

[12] J.K.Wu, A. Narasimhalu, B. Mehtre, C. Lam, and Y. Gao, "Core: A content-based retrieval engine for multimedia information systems," *Multimedia Systems*, vol. 3, pp. 25–41, 1995.

[13] S. Berchtold, C. Boehm, B. Braunmueller, D. A. Keim, and H. P. Kriegel, "Fast parallel similarity search in multimedia databases," *SIGMOD Conference*, vol. 3, pp. 1–12, 1997.

[14] D. Kirovski and H. S. Malvar, "Spread-spectrum watermarking of audio signals," *IEEE Trans. Signal processing*, vol. 51, no. 4, pp. 1020–1033, April 2003.

[15] F. Hartung, *Digital Watermarking and Fingerprinting of Uncompressed and Compressed Video*. Shaker Verlag, Aachen, Germany, 2000.

[16] F. Hartung, P. Eisert, and B. Girod, "Digital watermarking of mpeg-4 facial animation parameters," *Computers & Graphics*, vol. 22, no. 4, pp. 425–435, August 1998, (Special issue on "Data Security in Image Communication and Network").

[17] O. Benedens, "Robust watermarking and affine registration of 3d meshes," in *Information Hiding, 5th International Workshop, IH 2002, Noordwijkerhout, The Netherlands, October 7-9, 2002, Revised Papers*, ser. Lecture Notes in Computer Science, F. A. P. Petitcolas, Ed., vol. 2578. Springer, 2003.

[18] R. Agrawal, P. J. Haas, and J. Kiernan, "Watermarking relational data: framework, algorithms and analysis," *VLDB J.*, vol. 12, no. 2, pp. 157–169, 2003.

[19] R. Sion, M. Atallah, and S. Prabhakar, "Rights protection for relational data," in *International Conference on Management of Data (SIGMOD)*, 2003.

[20] D. Gross-Amblard, "Query-Preserving Watermarking of Relational Databases and XML Documents," in *Symposium on Principles of Databases Systems (PODS)*, 2003, pp. 191–201.

[21] C. Constantin, D. Gross-Amblard, and M. Guerrouani, "Watermill: an optimized fingerprinting system for highly constrained data," in *ACM MultiMedia and Security Workshop*, New York City, New York, USA, January 1–2 2005.

[22] G. Coatrieux, H. Maître, and B. Sankur, "Strict Integrity Control of Biomedical Images," in *SPIE - 2001: Security and Watermarking of Multimedia Contents III*, vol. 4314, January 2001, pp. 229–240.