Implementation of a User-friendly GUI for a Multiple Image Watermarking Tool Addressing Healthcare Applications

Aggeliki Giakoumaki, *Member, IEEE,* Anastassios Tagaris, *Member, IEEE,*
and Dimitris Koutsouris, *Senior Member, IEEE*

***Abstract*—Digital watermarking is a technology with many different application domains; nevertheless, its potential in providing value-added services towards secure and efficient health data management only recently started to be realized. The paper illustrates the perspectives of digital watermarking in the healthcare sector and presents a number of user-friendly graphical user interfaces needed to implement a multiple watermarking tool; this tool is aimed to be integrated in health information management systems, in order to enhance security of sensitive information and facilitate medical data management. The Graphical User Interface (GUI) allows the physician to define a Region Of Interest (ROI) in the image, which can either be totally ignored by the watermarking process or just accommodate the minimum payload (a reference watermark) needed to enable ROI integrity control. In the rest of the image, apart from the reference watermark, three other types of watermarks may also be embedded, providing origin identification, enhanced security of sensitive personal data, and efficient image indexing and data retrieval. The paper presents a number of graphical user interfaces that enable the watermark embedding and extraction modules with the required functionality, and describes the options that become available, towards secure and efficient medical data management.**

I. Introduction

THE landscape of healthcare delivery and information management has radically changed over the recent years, as a result of the constant evolvement of information and communication technologies. The wide distribution and easy access of sensitive healthcare data poses critical issues as far as their secure and efficient management is concerned. Among the different solutions the research community seeks so as to effectively face the emerging challenges, digital watermarking technology appears to have a very promising potential. Digital watermarking has already been implemented in a wide range of application domains; nevertheless, until recently it had not been realized that a number of issues including origin and data authentication, image archiving and retrieval, and sensitive data protection, could be effectively addressed through watermarking of medical images with the appropriate set of data. A review of the medical-oriented watermarking approaches recorded in the literature can be found in [1]. The proposed paper illustrates the perspectives of digital watermarking in the healthcare sector and presents a number of user-friendly graphical user interfaces needed to implement a multiple watermarking tool; this tool is intended to be integrated in health information management systems, in order to enhance security of sensitive information and facilitate medical data management.

II. Multiple Watermarking Tool for Medical Images

Watermarking medical images with the appropriate sets of information is an approach that provides alternative and/or complementary solutions to a number of issues relating to healthcare information management, including: origin and data authentication, enhanced security of critical personal and medical data, efficient image archiving and retrieval. A detailed discussion on the perspectives of digital watermarking in healthcare-oriented applications can be found in [2]. In order to simultaneously address these diverse medical application fields, multiple watermarks with varying characteristics and requirements need to be jointly embedded in a single image; these watermarks should be extractable independently from each other, as they are intended to be used for different purposes and at different levels of the health information management chain.

Our proposed watermarking Software Development Kit (SDK), whose architecture has been described in detail in [3], incorporates a watermarking engine that implements the embedding and extraction of multiple purpose-specific watermarks. The algorithms invoked by the engine for the embedding and the extraction of the watermarks have been presented in [2]; these algorithms subject the images to wavelet decomposition, in order to exploit the coefficient properties of different decomposition levels and subbands in distributing the different types of watermarks according to their desired characteristics and requirements.

In the following sections, an outline of the watermarking SDK architecture is provided, followed by the presentation of a number of graphical user interfaces that enable the watermark embedding and extraction modules with the

A. Giakoumaki is with the Biomedical Engineering Laboratory, School of Electrical and Computer Engineering, National Technical University of Athens, 9 Iroon Polytechniou str., 15773, Athens, Greece (phone: +30 210 7723516; fax: +30 210 7722431; e-mail: agiakoum@biomed.ntua.gr).

A. Tagaris is with the Biomedical Engineering Laboratory, School of Electrical and Computer Engineering, National Technical University of Athens, 9 Iroon Polytechniou str., 15773, Athens, Greece (e-mail: tassos@biomed.ntua.gr).

D. Koutsouris is with the Biomedical Engineering Laboratory, School of Electrical and Computer Engineering, National Technical University of Athens, 9 Iroon Polytechniou str., 15773, Athens, Greece (e-mail: dkoutsou@biomed.ntua.gr).

required functionality. As will be described, the GUI allows the physician to define a region of diagnostic significance (Region of Interest – ROI) in the image, which can either be totally ignored by the watermarking process or just accommodate the minimum payload needed to enable ROI integrity control. In this way, the quality and diagnostic value of the image is explicitly preserved throughout the watermarking process. The watermark destined to enable image integrity control is the so-called *reference* one [4]; in the rest of the image, apart from the reference watermark, three other types of watermarks may also be embedded, each conveying application-dependent information, namely: a) a *signature* watermark conveying origin identification data (e.g. physician's digital signature, identification code of an image acquisition device or laboratory, etc.), b) an *index* watermark carrying keywords to be used for efficient image indexing and retrieval (e.g. ICD 9/10 diagnostic codes, a patient's unique identifier such as his/her Medical Record or Social Security Numbers, etc.), and c) a *caption* watermark conveying information about the patient and/or the examination data, additional comments about the diagnosis or the diagnostically significant image parts, and so on. Section IV presents the graphical user interfaces that have already been developed and describes the options that become available, towards secure and efficient medical data management.

III. Watermarking SDK

The watermarking functionality may be delivered by the architecture that is presented in Fig. 1. The core of the system is the watermarking engine along with the API, acting both as watermark embedder and extractor [3]. The different components illustrated in Fig. 1 are briefly presented below:

a) Watermarking Engine: It comprises the core component of the SDK, which implements the embedding and the extraction of the multiple watermarks (signature, index, caption, and reference).

b) The API of the Engine: Actually, the API is not a separate component, as it is included in the watermarking engine; however, it is presented here as such for reasons of clarity. The role of the API is to allow the direct communication between external software and the engine.

c) Access & Security Component: This component authenticates users, in order for them to gain access to the API of the watermarking engine. In case of a version where images are stored in the local database, it could be extended, to define access rights on those images.

d) Local Database: The local database of the system is used as a repository, where the original and the watermarked images are stored and archived; it may contain the access rights of the SDK users as well. It could also be extended to support storing of patients' medical and demographic data. In that case the engine, the user interface, and the viewer, could make use of the database, in order to store and retrieve patient's data, images, health record, etc.

e) Image Viewer: This component is a tool to display both the original and the watermarked images. It may also be considered as a part of the user interface.

f) User Interface: Apart from the image viewer, the user interface includes also the required controls to embed as well as to extract the multiple watermarks from images. The functionality and the design of the user interface are presented in more detail in the next section.

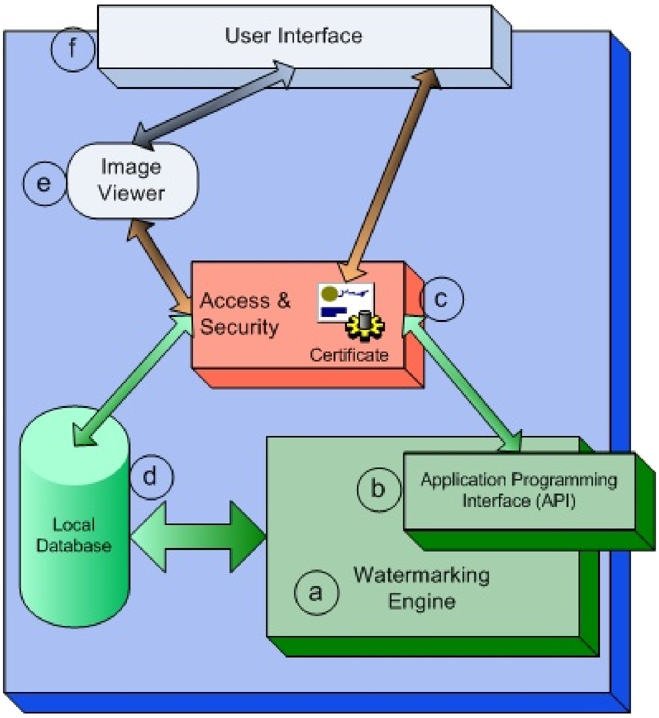


Fig. 1. Watermarking SDK architecture

IV. GUI Description

The user interface supports the everyday workflow of the user, in order to either embed or extract the previously described watermarks. Thereupon, there are two separate operating modes of the user interface, namely the embedding and extraction modes, which are described below:

1) Embedding mode

In order for the watermarking engine to implement image watermarking, the user has to provide the following input through the user interface:

a) A username and a password to enable authentication.

b) Selection of the original image that the user wishes to watermark (either from the file system or the local database).

c) Definition of a set of options regarding the embedding procedure, namely:

--Define if a ROI will be used and for what purpose (i.e., determine whether the selected region will be subjected to reference watermarking to enable integrity control, or will be totally excluded from the watermarking process).

--Determine which, if any, types of watermarks will be subjected to Error Correction Coding (ECC) for increased robustness.

--If ECC is selected, choose the ECC scheme to be used in each case.

--Specify the key to be used for watermark embedding, by either selecting it from a list of predefined keys, or defining a new one.

--Select the watermarks that will be embedded in the selected image.

--Define the data that each of the selected watermarks will convey. Depending on the nature of the information, the user can type the embeddable data, select them from a list, or specify a text file from which they will be read.

2) Extraction mode

In order for the watermarking engine to extract the information from a watermarked image, the user has to provide the following input through the user interface:

a) A username and a password to enable authentication.

b) Select the watermarked image (either from the file system or the local database).

c) Define which of the data watermarks (signature, index, caption) will be extracted.

d) Define whether the watermarked image will be checked for integrity.

e) If the image is found to have been tampered with, decide whether the viewer illustrates the image with the tampered regions highlighted.

The image viewer, which is a significant part of the user interface, should enable the user to:

a) See the original and the watermarked image.

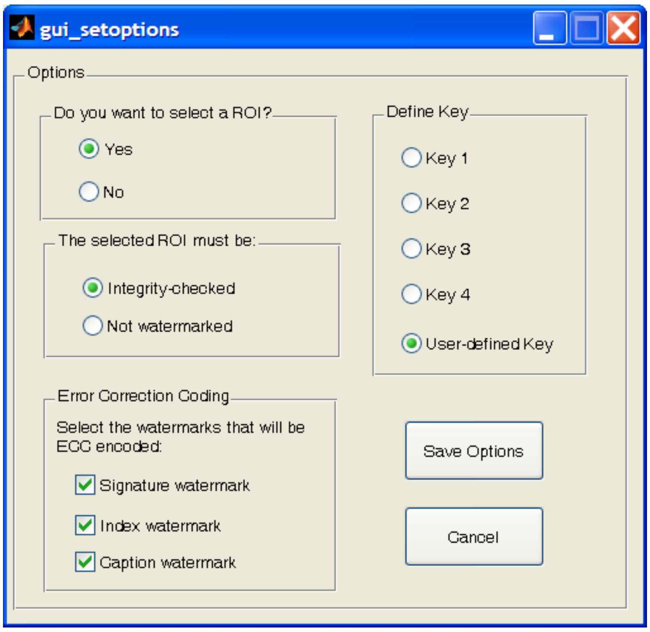
b) Define the ROI in the original image.

In order for the watermarks embedding to take place, the user interface prompts the user to provide input regarding the information to be conveyed in each watermark type. For instance, in the case that the user selects to embed a signature watermark, which aims at data origin identification, an input dialog box providing two options could appear; the one option may refer to the insertion of the machine ID, and the other one may involve embedding of the physician's ID or digital signature. The user can select either one or both options. In the case of the index watermark, the user interface could prompt the user to define whether he/she wants to type the appropriate keywords, or select indices from a list of ICD9/10 disease codes. Finally, in the case of the caption watermark, a corresponding input dialog box could appear, providing the user with the option of typing the desirable data in the corresponding boxes. For instance, the GUI could prompt the user to insert the patient's identification code and demographics, examination data, and additional comments that would facilitate the accurate evaluation of the patient's health status. For all of the above watermark types, the information to be embedded could also be read from a data file stored in the system.

The design and implementation of the current version of the user interface is based on its required functionality, which was described above. According to these, the application comprises of two modules, each accomplishing the embedding and the extraction procedure, respectively. The description of these two modules is provided below, thus illustrating their functionality and the user-system interaction.

A. Embedding module

When the user selects to run the watermarks embedding application, a *Set Options Form* (see Fig. 2) appears in the screen.


gui_setoptions

Options

Do you want to select a ROI?
● Yes
○ No

The selected ROI must be:
● Integrity-checked
○ Not watermarked

Error Correction Coding
Select the watermarks that will be ECC encoded:
☑ Signature watermark
☑ Index watermark
☑ Caption watermark

Define Key
○ Key 1
○ Key 2
○ Key 3
○ Key 4
● User-defined Key

Fig. 2. Set options form

This form prompts the user to define a set of parameters, based on which the watermarks embedding will take place. These parameters include the following:

1) ROI Selection: the user selects whether he/she wishes to select a ROI in the original image.

2) Actions in ROI: in case that a ROI has been defined, the user decides on whether this region will be excluded from the watermarking process, or will carry the reference watermark, in order to allow integrity control of the specific region.

3) Error Correction Coding (ECC): the user selects if ECC will be applied for increased robustness and if yes, in which of the data watermarks (signature, index, caption).

4) Key Selection: the user has the option of defining the key that will be used for watermark embedding and extraction, either by picking one out of a list of keys saved in a system repository, or by selecting to define a new key.

After the user selects to save his/her choices in the Set Options form, these parameters are stored in the system and are recalled from it during the multiple watermarks embedding procedure. After the options of the user are defined, the *Watermarks Embedding Form*, which is illustrated in Fig. 3, appears in the screen.

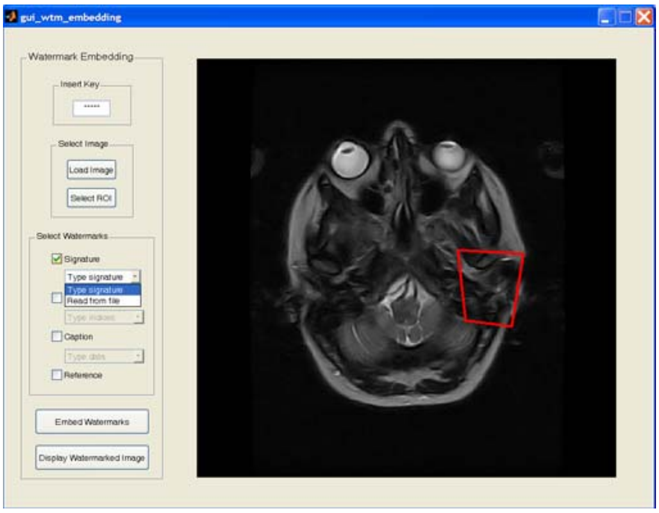


Fig. 3. Watermarks embedding form

This form prompts the user to define the following:

1) Define the key: in case that the user's selection in the Set Options Form was to use a new watermark key, an edit box captioned "Insert key" gets active, in order for the user to type the new key.

2) Select the image: when the user clicks on the "Load Image" button, a dialog box appears which allows him/her to browse the system folders and select the image that is to be watermarked. The selected image is then displayed in the form.

3) Select the ROI: if the user's selection in the Set Options Form was to define a ROI in the image, a button captioned "Select ROI" is active, thus allowing the user to define this region, by clicking the mouse on the appropriate points of the displayed image. The selected ROI is then displayed.

4) Select the watermarks to be embedded: the user selects which of the available watermark types he/she wishes to embed in the image. Except from the reference watermark that is a predefined bit array, known at both the embedding and the extraction sites, the user defines the way of specifying the information to be embedded in each type of the selected watermarks. Specifically, as far as the three data watermarks are concerned, the corresponding choices are as follows:

a) Signature watermark: if a signature watermark is selected to be inserted in the image, a pop-up list appears, which allows the user to define whether he/she will type the information to be embedded, or it will be read from a data file. Fig. 4 illustrates the dialog box that appears when the user selects to input the data oneself; this dialog box prompts the user to insert two different types of information, namely his/her identification code and the device identifier. The user may fill in both fields, or even ignore one if it is inapplicable in the specific case.

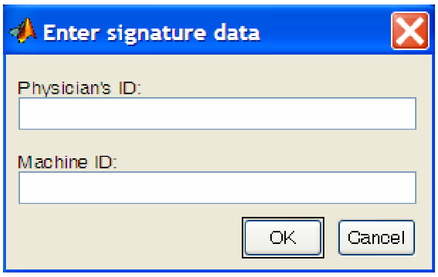
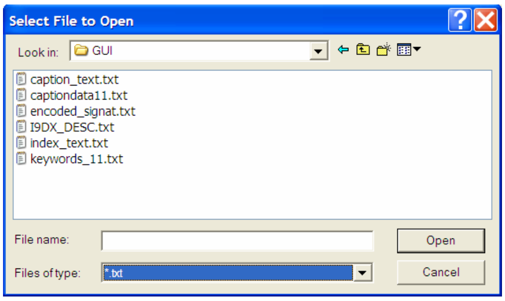


Fig. 4. Signature watermark input data typed by the user

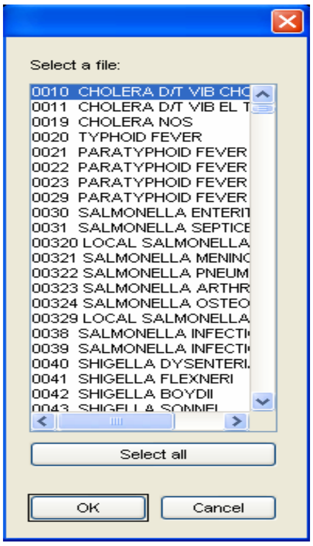
If the user selects from the pop-up list the option of embedding into the image a signature watermark that will convey data contained in a text file, the dialog box illustrated in Fig. 5 appears. This dialog box prompts the user to specify the text file, where the application can find the signature watermark data.



caption_text.txt
captiondata11.txt
encoded_signat.txt
I9DX_DESC.txt
index_text.txt
keywords_11.txt

Fig. 5. Signature watermark input read from a text file

b) Index watermark: if an index watermark is selected to be embedded in the image, a pop-up list is displayed, which allows the user to specify the source of the input data that will comprise the watermark.

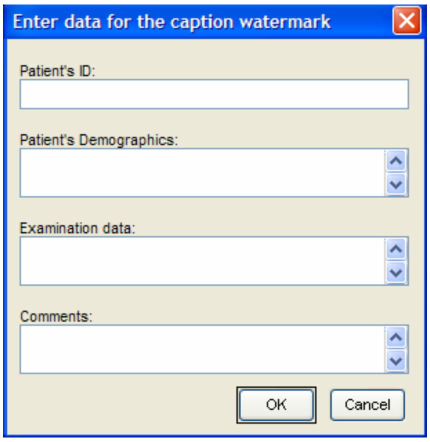


0010 CHOLERA D/T VIB CHC
0011 CHOLERA D/T VIB EL T
0019 CHOLERA NOS
0020 TYPHOID FEVER
0021 PARATYPHOID FEVER
0022 PARATYPHOID FEVER
0023 PARATYPHOID FEVER
0029 PARATYPHOID FEVER
0030 SALMONELLA ENTERIT
0031 SALMONELLA SEPTICE
00320 LOCAL SALMONELLA
00321 SALMONELLA MENINC
00322 SALMONELLA PNEUM
00323 SALMONELLA ARTHR
00324 SALMONELLA OSTEO
00329 LOCAL SALMONELLA
0038 SALMONELLA INFECTI
0039 SALMONELLA INFECTI
0040 SHIGELLA DYSENTERI.
0041 SHIGELLA FLEXNERI
0042 SHIGELLA BOYDII
0043 SHIGELLA SONNEI

Fig. 6. Index watermark input data selected from the ICD9/10 disease codes list

Specifically in this case, the user chooses among three options of defining the input source, namely: he/she can type the keywords, select them from an ICD9/10 disease codes list, or specify a data file that contains the keywords to be embedded. Fig. 6 illustrates the dialog box that appears when the user chooses to pick the keywords from the ICD9/10 disease codes list. For reasons of brevity, the other two options are not illustrated in figures, as they are analogous to the corresponding options for the signature watermark.

c) Caption watermark: as in the other two cases, if the user selects to embed a caption watermark, a pop-up list is displayed, allowing the determination of the input data source. Two options are available, namely the typing of the data to be embedded directly from the user, and the selection of a data file that contains this information. Again, the dialog box appearing in the case that the user specifies a data file from which the caption data will be read, is similar to the one corresponding in the signature case (Fig. 5), and is not illustrated here for brevity reasons. Fig. 7 illustrates the dialog box that appears when the user selects to input the data oneself; this dialog box prompts the user to insert four different types of information, namely the patient's identification code and demographics, examination data, and additional comments for future reference or other physicians' guidance. The user may select to fill in all four fields, or leave blank those that do not fit the specific application.



Enter data for the caption watermark

Patient's ID:

Patient's Demographics:

Examination data:

Comments:

OK Cancel

Fig. 7. Caption watermark input data typed by the user

5) Embed the selected watermarks: when the user clicks on the "Embed Watermarks" button, the multiple watermarks embedding procedure takes place.

6) Display the watermarked image: when the user clicks on the button captioned "Display Watermarked Image", a new window opens, displaying the resulting watermarked image, thus allowing the user to compare it with the original.

B. *Extraction module*

When the user selects to run the watermark extraction application, a *Watermarks Extraction Form*, which is illustrated in Fig. 8, appears in the screen. This form prompts the user to do the following:

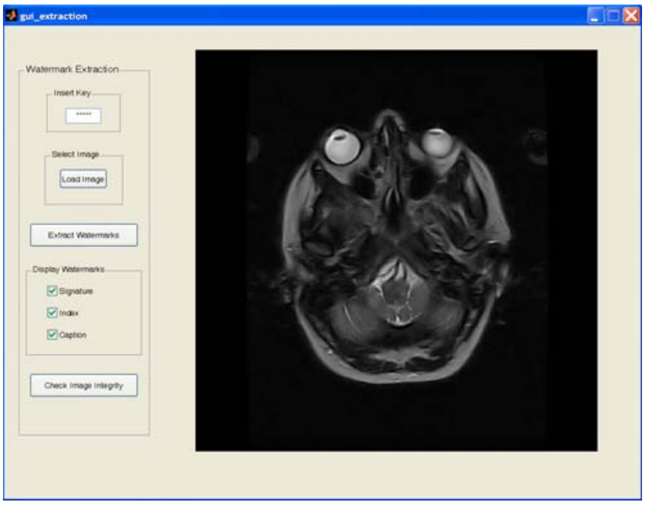
1) Specify the key: in case that a user-defined key was used in the watermarks embedding procedure, the user has to type this key in the edit box captioned "Insert key".

2) Select the image: when the user clicks on the "Load Image" button, a dialog box appears which allows him/her to browse the system folders and select the watermarked image, from which the watermarks will be extracted. The selected image is then displayed in the form.

3) Extract the watermarks: when the user clicks on the "Extract Watermarks" button, the multiple watermarks extraction procedure takes place.

4) Selection of the extracted watermarks to be displayed: the user selects which of the extracted data watermarks (signature, index, and caption) will be displayed on the screen. For each of the selected watermark types, a text box presenting the corresponding extracted watermark appears.

5) Check image integrity: in case that the user wishes to perform integrity control of the watermarked image, he/she clicks on the button captioned "Check Image Integrity". Based on the comparison of the extracted reference watermark with the originally embedded one (which, as mentioned above, is a priori known at the extraction site), the application calculates a *difference image* illustrating the tampered image parts; finally, the watermarked image with the tampered regions highlighted, is displayed on the screen.



gui_extraction

Watermark Extraction

Insert Key: *****

Select Image

Load Image

Extract Watermarks

Display Watermarks

☑ Signature

☑ Index

☑ Caption

Check Image Integrity

Fig. 8. Watermarks extraction form

The current version of the GUI has been implemented in Matlab; however, it is now being implemented in Microsoft .NET Framework (v2.0). Microsoft® Visual Studio® .NET is a comprehensive tool set that allows developers to organize the functionality of their application in unique ways and create smart and extensible application layouts that are easy for the users to use [5]. Moreover, it offers great capabilities in terms of portability and easy integration with other either windows or web applications, which is a necessity in the healthcare domain.

V. Discussion

The paper pinpoints the perspectives of digital watermarking in the healthcare sector and presents a number of preliminary user-friendly graphical user interfaces needed to operate a multiple watermarking tool; these interfaces enable the user with the capability to simultaneously embed multiple watermarks in medical images, each of them addressing different purposes. The functionality provided by the tool results in the enhancement of security and access control of sensitive data, and enables source and data authentication; thus, its integration into health information management systems is anticipated to provide value-added services to a range of critical health data management issues.

References

[1] A. Giakoumaki, S. Pavlopoulos, D. Koutsouris, “Secure and efficient health data management through multiple watermarking on medical images,” *Medical & Biological Engineering & Computing*, to be published. Available: http://dx.doi.org/10.1007/s11517-006-0081-x

[2] A. Giakoumaki, S. Pavlopoulos, D. Koutsouris, “Multiple image watermarking applied to health information management,” *IEEE Transactions on Information Technology in Biomedicine*, to be published.

[3] A. Tagaris, A. Giakoumaki, L. Karle, D. Koutsouris, “Watermarking SDK implementation to facilitate integration in a secure healthcare environment,” in *Proc. 28th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBC’06*, New York, USA, August 30 - September 3, 2006.

[4] D. Kundur, D. Hatzinakos, “Diversity and attack characterization for improved robust watermarking,” *IEEE Trans. Signal Processing*, vol. 49, no. 10, pp. 2383-2396, Oct. 2001.

[5] http://msdn.microsoft.com/netframework/