

Digital Watermarking of MPEG Videos: Empirical Studies and Open Source Tools*

Laura M. Castro¹ Cheng-Fang Yang² Natasha Simon³ Ernst L. Leiss⁴

Abstract

Digital watermarking is a technique that allows the invisible insertion of information into digital audio and video, in particular unique identifiers that allow tracking and facilitate copyright protection. In watermarking efforts, the challenge is creating invisible, robust, time-variant watermarks. It is also essential that the watermark can survive everyday image processing and compression techniques. We focus on watermarking of MPEG videos as a deterrent to theft and as an indicator of content authenticity. Crucial for this purpose is the payload of a watermark. In an effort to assess the amount of information that a watermark may contain, we modify the internal representation of MPEG images in different ways. Our primary objective is assessing the ability of the images to tolerate varying degrees of modification, while preserving the invisibility of the watermark. Besides, in spite of the growing interest in digital watermarking, there are no generic tools available to researchers to help them design and test their algorithms. We therefore developed Java Watermarking Tool, a basic, simple and easy-to-use application that represents a first step in generic testing tools for digital video watermarking.

* Support under NSF grants DUE 0313880, SCI 0453498, and OISE 0519316 is acknowledged, as well as projects MEC TIN2005-08986 and XUGA PGIDIT06PXIC105164PN.

¹ Department of Computer Science, University of A Coruña, Spain; lcastro@udc.es

² The University of Texas at Austin; mcfyang@gmail.com

³ Southern University, New Orleans; natasha_simon1@hotmail.com

⁴ Department of Computer Science, University of Houston; coscel@cs.uh.edu

1. Introduction

Digital watermarking is a technique that allows the insertion of information into digital signal-based objects (audio, video) [1]. In particular, we focus on robust, invisible watermarks [2]. Due to its feasible industrial uses, interest in this field has been growing increasingly; however, the fact that there are no generic applications to help researchers design and test their watermarking algorithms is an inconvenience that forces them, each time, to develop ad-hoc solutions of their own.

In the next section, we give a brief introduction to digital watermarking, with emphasis on its applications. Then we describe the current problems in this research field. After that, goals and objectives of our empirical studies are explained, including the outline of some results of our payload studies; here the emphasis is on the amount of information that can be inserted into images for tracking and identification purposes without affecting the perceived quality of those images. This is followed by a description of the tool we developed, JWMTTool. To sum up, we comment on our conclusions and future work lines.

2. Digital Watermarking

Digital watermarking is based on the notion of storing information in images (or audio) in a way that is imperceptible. We are interested in robust, invisible digital watermarks [2].

Digital watermarking can be used to fill in the gap between copyright protection and digital media distribution. Digital media distribution is being increasingly used in practical applications. It is easier, faster, and, most of all, it is much cheaper than traditional distribution means. However, digital media are also much easily copied and altered; this creates problems of security and integrity [1].

Digital watermarking provides a solution to these problems. Watermarking consists of inserting additional data into the digital media in a way that ensures:

- Robustness
- Non-perceptibility (in ordinary use)
- Non-detectability (in ordinary use)
- Integrity

A watermarking process being *robust* means that if the digital media is altered by the user in any way that does not affect its content (e.g., resizing, cropping, recoding, etc.), the watermark will survive the process. A watermark must also be *non-perceptible* to the viewer who is watching a watermarked digital video or to the listener who is listening to a watermarked audio stream. The *non-detectability* property is very much alike the non-perceptibility property, but instead of referring to the human senses, has to do with the

consistency we need to provide between the original and the watermarked media. Finally, the *integrity* property refers to the necessity of ensuring that the watermarking process does not affect the fundamentals of the media, does not corrupt or damage it.

Apart from those four main properties, one must also bear in mind other factors, such as data payload and complexity. There would be no use in having a watermarking process that would generate watermarked media twice as big as the original one, or that would allow us to insert only a few bytes of data, or that would take two days to produce the watermarked result. Problems arise because some of these features are contradictory or incompatible.

Figure 1 shows the usual watermarking lifecycle. At the beginning of the process, the watermarking system (in this case, a watermark embedder) takes some digital media as an input, as well as the watermark itself and maybe some additional data (such as an encryption key, for example), and generates the watermarked media, which is the media that would be distributed and used (viewed) by the customers. At the other end, we find another watermarking system, namely the watermark extractor. This system should be able to recover the original media from the watermarked media, maybe using the original watermark and additional data, if necessary.

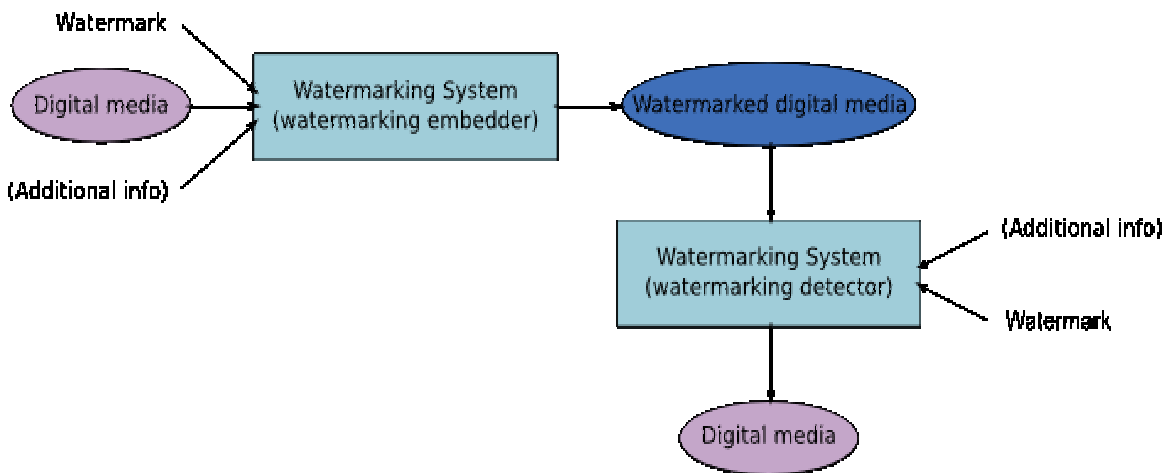


Figure 1: A generic watermarking system

Some applications of digital watermarking techniques include *media fingerprinting*, as a way to identifying uniquely the owner of a media by means of this kind of digital signature, *broadcast authentication*, as a way of monitoring access to restrictively published media, *copy control*, as a solution for allowing the user to download the media, but prevent further unauthorized distribution, or *secret communication*, as another system to ensure that the information comes from a trusted source. Watermarks can also be used in Digital Rights Management (DRM) systems.

3. Research Problems

Most digital watermarking research involves developing new embedding, detection and extraction techniques, paying special attention both to the efficiency and robustness of the techniques and the special characteristics of the media to be watermarked. For example, video streams and audio streams have different properties which must be taken into consideration. Furthermore, since most signal-based information must be compressed, digital watermarking schemes must function within the context of data compression.

A core problem to be studied in any watermarking scheme is the amount of information that one can add to the media, without affecting the perceived quality of the watermarked digital media. An important deficiency in such studies is the lack of generally available tools: researchers have to design and implement their own tools or programs to check out their ideas, to test their algorithms to the limit, to find out the boundaries of their theoretical models.

4. Preliminary Payload Studies

Initial empirical studies were carried out to assess the behavior of time-variant digital watermarks [3], especially focusing on the amount of information that can be inserted into an image without distorting its visual qualities. This essentially means analyzing by how much the coefficients representing the image can be changed without affecting the perceived image quality.

In this study, we used all free software utilities to build an open implementation of an invisible, time-variant watermarker. To implement the watermarking scheme in [3], we augmented the compressor/decompressor XviD, an open source project that enjoys wide use and support [4, 5]. It employs standard MPEG compression, which we sketch in the next section. We obtained our target video for watermarking, consisting of three frames with a cactus moving from left to right, from the same XviD package.

The output of XviD compression is m4v video files. To play them, we used IBM's free M4Play player [6]. The player demonstrates some of the capabilities of the IBM Toolkit for MPEG-4, a set of tools for generating and using MPEG-4 content. M4Play also allows us to save individual frames as images files, which permits easier analysis.

To analyze in more detail the differences between the original and watermarked frames, we used GIMP, an open source image manipulation program [7]. Once a watermark has been inserted, the information contained in the watermark can be used to demonstrate intellectual property rights. This is where the identification process begins. In the identification process we have used three ways to test that the watermark is properly embedded. The first test signals whether a watermark does or does not exist. The second

test shows the locations where the byte numbers that differ. The last test allows the viewing of the actual picture difference between two frames. Two of the three watermark identifiers are completed by using Linux shell commands and the third involves the use of GIMP.

The first test uses the *diff* command [8] to determine whether there is a difference between the two files or images. This command-line utility outputs differences between files line by line in several formats. For files that are identical, it produces no output; for binary (non-text) files, it will report just that they are different.

The second test signals that the watermark does exist; this is done by using the *cmp* [8] command. This utility is used to show the byte and line numbers where two files differ, and also all the bytes that differ between the two files, side by side. To identify the appearance and location of the watermark, we view the difference. It is necessary to save corresponding frames from both videos, the original and the watermarked one.

This lead us to the third test. Once the difference is taken, we merge down the two images using GIMP, and adjust the white point in the color level until the watermark becomes invisible.

Following the watermarking scheme of [3], we embedded a watermark in the target video in the middle of the MPEG compression process. Standard MPEG compression divides videos into three types of frames: Intra (I), Predicted (P), and Bi-directionally Predicted (B) [3]. The compressor applies JPEG still image compression to the frames. Of particular importance in this process are the quantized discrete cosine coefficients [2]. These are 64 integer coefficients (the DC coefficient and the 63 AC coefficients), which are modified when a digital watermark is inserted. Typically, the watermark insertion occurs in the first p AC coefficients. As discussed in [2], large changes in the AC coefficients affect the image quality in ways that depend on the position of the modified coefficient: coefficients at the beginning of the AC sequence have greater influence on the perceived quality than those towards the end. However, since MPEG compression tends to get rid of the AC coefficients towards the end, it is important to assess how much change can be tolerated at the beginning of the AC sequence (which will not be suppressed in the compression process, or else the image quality will be much affected).

In time-variant watermarking, we embed different watermarks in each I-frame. As shown in [2], time-variant watermarking allows the detection of frame sequencing manipulations that would be undetectable by traditional, time-invariant watermarking schemes. These include deletion of existing frames, as well as permuting and repeating existing frames.

We studied the effect of inserting varying amounts of information on the perceptibility of the watermark in the resulting video. To this end, we varied the magnitude of the number we insert as the watermark. Specifically, we embed watermark numbers of eight digits so we can determine which range of size eight in the sixty-three AC coefficients can hide the most information before a viewer would deem the result unacceptable due to changes

in clarity or color. We consider the ranges $[Start, Stop)$ as listed in tables 1 and 2, considering that the DC coefficient is usually not changed when watermarking [3].

4.1.Insertion: Spatial Frequency

Inserting the watermark after quantization in the midrange of a zigzag traversal of the coefficients matrix likely yields an invisible and more robust watermark [3]. We first consider adding/subtracting increasing integer values of y , starting from $y=1$, to the coefficients from $Start$ to $Stop$, regardless of the original coefficients' values.

Table 1 shows starting at what value of y the embedding of the watermark affects the video noticeably. As expected, it indicates that later coefficients hide more information than earlier coefficients. Here we add (or subtract) the positive value y to the given coefficient (absolute change).

Table 1: Limits of acceptable integer additions

<i>Start</i>	<i>Stop</i>	$x = x \pm y$
1	9	2
9	17	2
17	25	3
25	33	4
33	41	4
41	49	5
49	57	5
57	64	7

Since the amount of information that we can embed in the video depends also on the target video [3], we considered modifying the matrix by percentages of the original coefficient. We study two ways of making these modifications to a given coefficient x by an increase of z percent (relative change):

$$x = x + z x$$

$$x = x + z/x/$$

The first relative change is symmetric (note that coefficients may be positive and negative). For example, if $z = 10\%$, then $x = 10$ yields a new value of 11 while $x = -10$ yields -11 . The second change essentially consists of a relative shift by a factor of z . For the above values, the two new coefficient values would be 11 and -9 . We study the effects of successively increasing the value of z , starting from $z = 0$ (effectively no watermark). Table 2 shows the percent increase z where we first notice a distortion of the video.

Table 2: Maximum acceptable percent increase

<i>Start</i>	<i>Stop</i>	$x = x + z x$	$x = x + z/x/$
1	9	40	40
9	17	60	60
17	25	200	200
25	33	200	200
33	41	300	380
41	49	610	750
49	57	no limit	no limit
57	64	no limit	no limit

We see that, like the absolute additions (and subtractions) we performed earlier (table 1), later coefficients display the most information potential. On the other hand, later coefficients are typically discarded for compression. Without dramatic changes, information inserted into coefficients 49-64 might be removed in compression. From the results, we infer that coefficients 17-33 can handle a good deal of information without affecting the image quality or interfering with ordinary MPEG compression.

4.2. Insertion: Chrominance and Luminance

In addition to exploring the information potential in various ranges in a zigzag traversal of the coefficient matrix to take advantage of the human interpretation of spatial frequency, we investigate how we may exploit humans perception of luminance (amount of brightness, [9]) and chrominance (refers to the color, [10]) to embed information.

The XviD codec divides quantization process into a quantization step for luminance and a quantization step for chrominance [4]. Figures 2 and 3 illustrate the results of the I-frames with modified luminance and chrominance coefficients, respectively; they show the difference between the original and the watermarked I-frame. Luminance modifications affect the video's texture, creating rougher edges and starker contrast. Chrominance modifications introduce a red tint to the video while maintaining the shape of the objects.

After comparing the original I-frame to watermarked I-frames, we find that, consistent with human higher sensitivity to luminance, we can insert slightly more information in the chrominance coefficients than in luminance coefficients without noticeably distorting the original video.

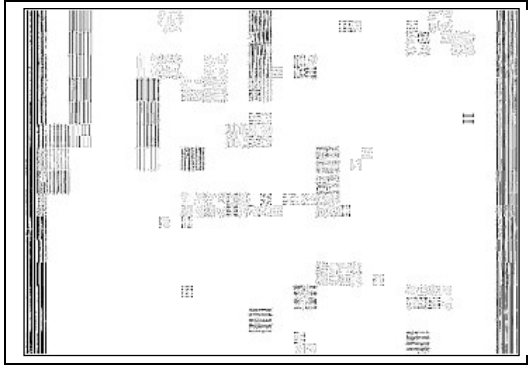


Figure 2: Difference from Chrominance Modification

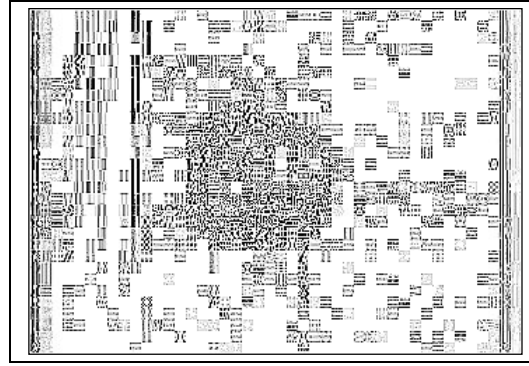


Figure 3: Difference from Luminance Modification

As for the P-frames, the maximum modification possible for I-frames (as reported in tables 1 and 2) minimally distorts the P-frames that follow for a casual viewer. Upon further analysis for example, for coefficients from 25-33 with the maximum increase of 200% in the I-frame coefficients, the P-frame displays more but lighter differences (figure 5) than does the I-frame (figure 4).

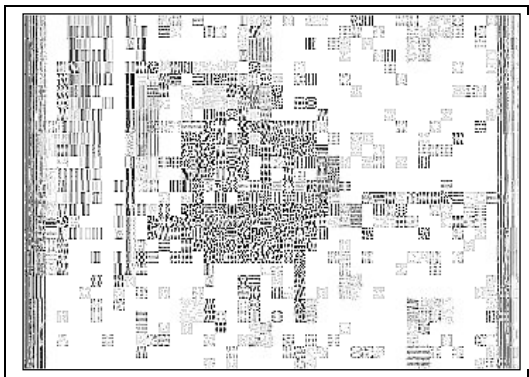


Figure 4: Difference in I-frame

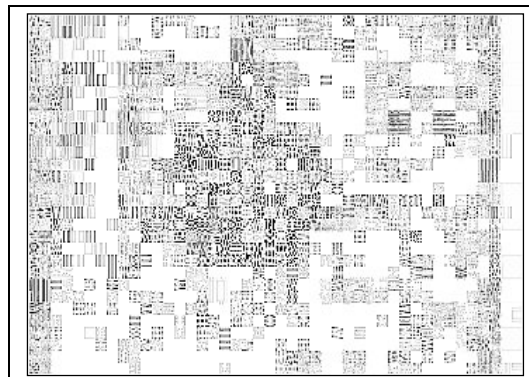


Figure 5: Difference in P-frame

These studies indicated a clear need for a more systematic framework for studying watermarking effects. To this end, we developed JWMTTool.

5. Design and Implementation of JWMTTool

JWMTTool (*Java Watermarking Tool*) is a simple watermarking testing application implemented in Java [11] (see http://www.madsgroup.org/staff/laura/jwmttool_en.html, figure 6). Java technology was chosen as implementation platform for several reasons. First, it is a popular and known framework, so the development effort was likely to be manageable. Second, Java provides a multiplatform and easy interoperable solution, a

very interesting point for us, if our contribution is to be used by third parties, regardless of the operating system they may run.

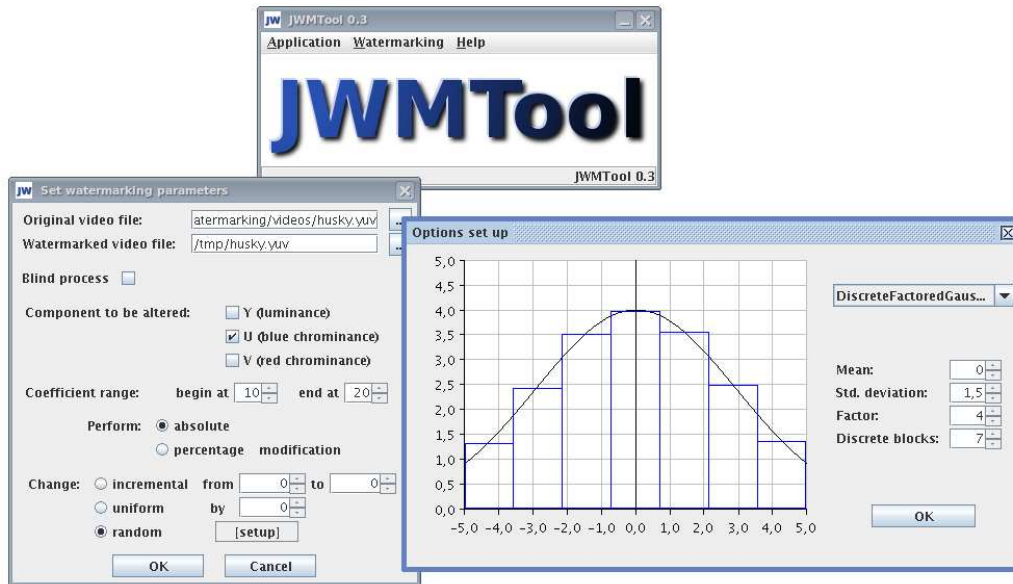
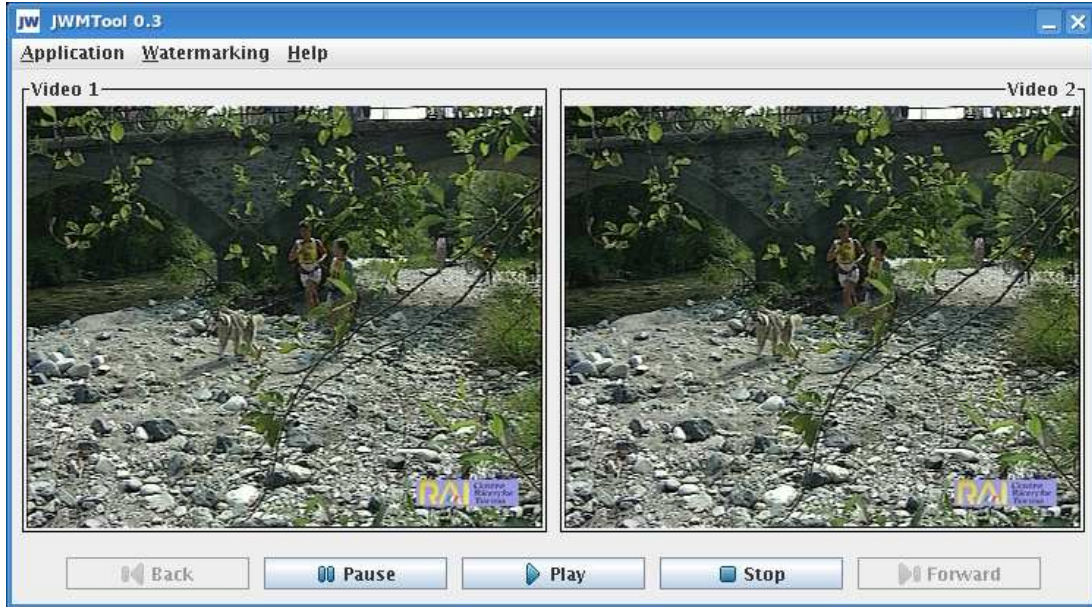


Figure 6: Java Watermarking Tool configuration overview

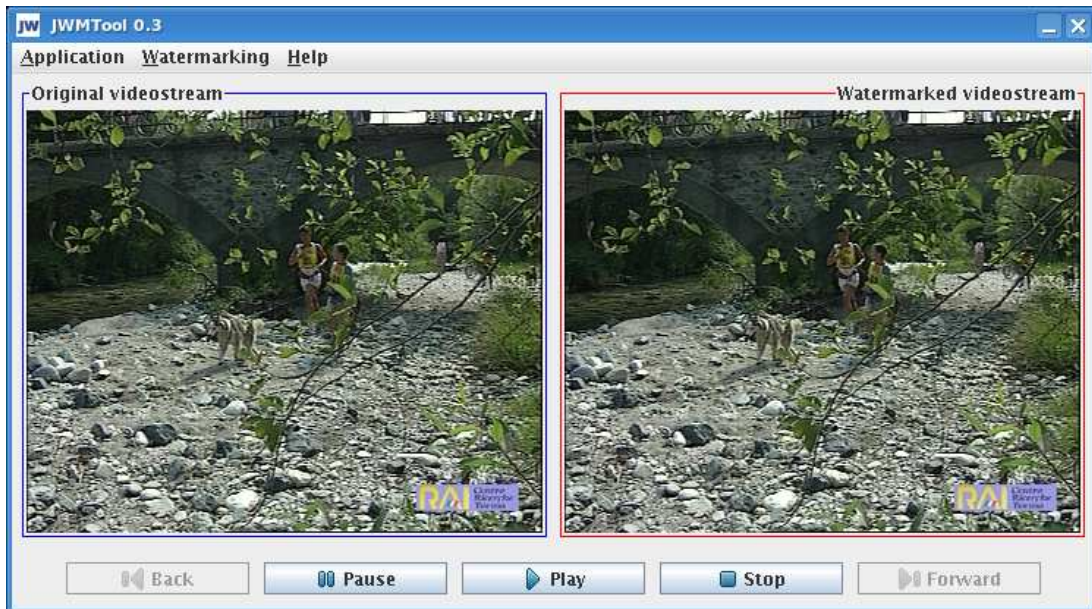
Our *Java Watermarking Tool* has been designed to deal with uncompressed video streams. Previous research [12] in digital watermarking for digital videostreams involves almost exclusively compressed video streams, such as MPEG videos. Thus, even before developing this tool, as one of our next steps, we wanted to check the behavior of uncompressed streams when being watermarked. And since we decided to develop a watermarking test tool, thence JWMTTool which supports the YUV format [13].

JWMTTool allows the user to run watermarking tests to check the amount of variation that it is possible to add to a particular stream before an observer is able to realize there is a difference between the original and the resulting video. In order to do so, our program configuration has multiple options, as shown in figure 6, so that not only the component (luminance, blue chrominance, red chrominance [13]) to be modified can be chosen, but also the AC coefficient range to be affected and the nature of the modification (sequentially increasing, constant, or random; by an absolute value, by a percentage). Furthermore, a blind or non-blind execution may be selected; in blind mode, after performing the alterations on the input stream, the system displays two image streams (the original one and the watermarked one) side-by-side without indicating which is watermarked and which is the original; in non-blind, the user is informed which is which (see figure 7).

Figure 7: JWMTTool: blind vs. nonblind execution



(a) Blind test



(b) Non-blind test

6. Conclusions and Future Work

The potential use of digital watermarking technology as an authentication and data verification mechanism has attracted a growing interest into this research field. Multimedia theft is a growing problem in the entertainment industry. By watermarking videos, a means is provided by which true media ownership can be determined. By comparing the raw video to the watermarked data, the watermark can be extracted. The extracted watermark can be used to authenticate against a third party registry of watermarks, thereby establishing the true ownership of intellectual property.

Another potential use is tracking the distribution of media. By watermarking media differently from different sources, by surveying the end users the success of various distribution systems can be determined. This can also be useful in tracking unauthorized distributions.

However, few tools are available to perform basic tests that allow researchers to check out new ideas, new algorithms. This became clear during our initial payload studies. The tool presented in this paper should help in remedying this inconvenience. Additional work needs to be done, as JWMTTool is a very simple and basic functional research tool that could benefit from adding new features as the need for them arises.

Bibliography

- [1] E. L. Leiss: Protecting Digital Content, IFIP World Computing Congress, Tutorial, Security Stream, Santiago, Chile, August 20-25, 2006.
- [2] E. L. Leiss: Time-Variant Watermarking of MPEG-Compressed Digital Videos, CLEI 2004 – Conferencia Latinoamérica de Informática, Sept. 27 – Oct. 1, 2004, Arequipa, Peru. Also in CLEI Electronic Journal, Vol. 1, 2005 (<http://www.clei.cl/cleiej>).
- [3] E. L. Leiss: Time-Variant Watermarks for Digital Videos: An MPEG-Based Approach. In Seitz J. *Digital Watermarking for Digital Media*. Hershey, Information Science Publishing, pp. 215-232.
- [4] XviD Development Team. <http://www.xvid.org>, 2006.
- [5] Doom9. XviD guides. <http://www.doom9.net>, 2006.
- [6] IBM Toolkit for MPEG-4. <http://www.alphaworks.ibm.com/tech/tk4mpeg4>, 2006.
- [7] The GIMP. <http://www.gimp.org>, 2006.
- [8] GNU. <http://www.gnu.org/software/diffutils/>, 2006.
- [9] Luminance. <http://en.wikipedia.org/wiki/Luminance>, 2006.
- [10] Chrominance. <http://en.wikipedia.org/wiki/Chrominance>, 2006.
- [11] Sun Microsystems Java technology, <http://java.sun.com>, 2006.

- [12] Juergen Seitz. *Digital Watermarking for Digital Media*. Hershey, Information Science Publishing, 2005.
- [13] Yuv model. <http://en.wikipedia.org/wiki/YUV>, 2006.