Data Hiding Techniques: an Innovative Approach to Video Interactivity

MIHAI MITREA¹,², SORIN DUȚĂ¹, MARIUS PREDA¹, FRANÇOISE PRÊTEUX¹
¹ ARTEMIS Project Unit,
GET/INT
9, Rue Charles Fourier, 91011, Evry-France
² Faculty of Electronics and Telecommunications,
POLITEHNICA University of Bucharest-Romania


Abstract: Nowadays, the multimedia consumer is introduced to an entirely new multimedia concept, namely the enriched media. The versatility of the digital networks alongside with the new, intelligent terminals, have allowed the creation and distribution of the classic media types enhanced with additional content information, introducing new features, from the trivial media-related information (e.g. EXIF data for the still images, ID3 tag for the MP3 music) to display/play information and interactive behaviour. However, these new features are hardly available on classic distribution networks. This paper presents an innovative approach to multimedia enrichment, allowing not only the distribution of this type of media over older networks, but its compact and unitary handling as well. The enriching information is inserted directly into the host media by means of data hiding techniques. The experiments were carried out in cooperation with the SFR (Vodafone group) wireless service provider in France.

Key-words: multimedia interactivity, data hiding.

1 Introduction

The advent of the Internet technologies and the continuously increasing non-linear multimedia content consumption require new types of digital products. These are the so called enriched multimedia products. They are obtained by enhancing the traditional media types (still image, video, audio, 3D) with additional information: content presentation attributes and/or description, interactive behaviour, outside references, etc. Consequently, new types of objects (visual, audio, metadata) complete the traditional ones.

This new paradigm naturally complies with the Internet related applications: any type of information (regardless its format or representation) can be reliably transmitted, stored, and processed. On the contrary, it is incompatible with several other networks or service types; for instance, for the Digital Television, where only audio and video data can be properly dealt with, no more room is left for the additional data.

At our best knowledge, this paper is the first to address this problem by means of data hiding techniques. The principle consists in transmitting the metadata (the enrichment information) via in-band channels obtained by means of data hiding (watermarking) techniques. The challenge is to design data hiding techniques reaching the trade off among transparency (the enrichment process should not alter the perceptual quality of the host media), robustness (possibility to recover the metadata at the end user even when high distortions occur through the channel) and data payload (the amount of metadata which can be inserted). Moreover, this watermarking method should also allow various data types (e.g. video, audio, and 3D) to be processed in a unitary manner. The illustration in this paper refers to video data.

The present paper has the following structure. Section 2 presents a new watermarking method (patent pending) while Section 3 experimentally evaluates it. Section 4 presents how the method can cope with the interactivity applications and concludes the paper.
2 A unified approach to robust watermarking

2.1 Watermarking basic concepts

The term robust watermarking designates any method that imperceptibly inserts some data - referred to as mark or watermark - into a host document [1] (video, audio, 3D ...); it should not be possible for these additional data to be removed by normal or malicious processing. The watermark is generated starting from some public information (e.g. copyright information) and from some secret information - the key - available only to the copyright holder.

For a real life copyright application, a watermarking technique should feature good transparency, high robustness, a quite large data payload, very low probability of false alarm, and obliviousness.

- **The transparency** refers to the imperceptibility of the watermark in the document.
- **The robustness** is the ability of the watermark to survive any processing that may be applied to the document: mundane processing and attacks (i.e. malicious transformations applied to the document with the stated purpose of removing the watermark).
- **The data payload** is the quantity of information that is inserted into the host document.
- **The probability of false alarm** is the probability of taking an unmarked document for a marked one.
- **The obliviousness** refers to the possibility of detecting the mark in the absence of the original document.

From the theoretical point of view, a generic watermarking system can be modelled as a noisy channel, Fig. 1. The copyright information is a sample from the information source and should be recovered at the detection side (i.e. it should be detected in the marked document). The elements that make the watermark detection difficult can be modelled as the channel noise: the original document, the mundane processing, and the attacks. The watermarking procedure itself plays the role of the modulation technique (i.e. the way in which the mark is inserted into the host document).

The transparency requirement means a power constraint on mark insertion: the mark should not perceptually alter the original data. The literature provides a great number of solutions to this problem. Among them, the highest degree of generality is provided by the Spread Spectrum (SS) and the Informed Embedding (IE) methods.

The SS methods have already been used in telecommunication applications (i.e. CDMA) providing a good solution for very low power signal transmission over noisy channels. Their principle consists in spreading the signal over a very large band, thus providing a very low power signal (hence imperceptible) in any frequency sub-band. The IE methods take advantage of the side information paradigm [2] stated by Shannon.

The side information principle states that a channel noise known at the transmitter and unknown at the receiver does not decrease the channel capacity. Thus, the original video (channel #1 noise) should not be considered as an impediment to the watermark detection.

In practice, the SS methods feature a very small data payload, although affording very good robustness and transparency. Conversely, the IE methods feature a huge data payload, but with very poor robustness and transparency.

2.2 The watermarking method

In order to pass from some side information theoretical concepts to a real life application, this paper adapts and extends the principles in [3,4]. Fig. 2 is a synoptic representation of the method.
We shall further detail each of the five blocks there presented, for the particular case of video watermarking. For clarity, they will be discussed in the following order: mark generation, salient vector extraction, detection, informed embedding, and channel.

Fig. 2. The advanced method synopsis.

**Block #1: Mark generation**

*Be there a message of M bits and be there the key. The aim of this block is to compute the mark to be embedded, starting from both the message and the key.*

In order to be embedded as a mark into the original document, the M bits are encoded by means of a modified trellis code [3,5]. The trellis has K states and 2 arcs exiting each state (each transition codes one bit). Each arc is labelled with an N length vector whose components are real numbers (and not bits like in basic trellis encoders). These labels are computed starting from the key, *i.e.* they are known only by the true document owner.

Note: The output of a trellis encoder depends on the input bit and on the previous $K^2 \log_2 K$ transitions.

Each combination of ($\log_2 K + 1$) adjacent bits from the message to be embedded is replaced by an N length label. Consequently, the mark is a vector denoted by $g$, with real components, having an $M \times N$ length.

**Block #2: Salient characteristic vector representing the document**

*The aim of this block is to extract a vector denoted by $c_0$ which has the same $M \times N$ length as the mark and which contains salient information representing the document.*

The watermark is inserted into some transformation coefficients of the video sequence, and not directly into the video itself. The transformation here considered is the DWT (Discrete Wavelet Transform).

Be there a colour video sequence consisting in L frames. Each frame is represented in the HSV (hue-saturation-value) space [6]; the V component is normalised to [0,1] interval.

In order to obtain the $c_0$ salient vector (one of the inputs of the embedding algorithm, Fig 3) the following steps should be gone through:
1. The (9,7) 2D-DWT [7,8] is individually applied to each frame in the video sequence, at an $N_r$ resolution level.
2. The coefficients belonging to the $HL_{N_r}$ and $LH_{N_r}$ low frequency sub-bands are sorted in a decreasing order of their values. The largest D coefficients in each frame are (randomly) shuffled and then recorded into the $c_0$ vector.
3. The original locations of the $c_0$ vector components are stored into a $\nu$ vector.

**Block #3: The detection**

*Be there a document which is supposed to be marked. The aim of this block is to establish whether the M bit message has been embedded into the considered document or not.*

The first task is to extract from the document the salient vector susceptible to convey the mark, see Block #2 above. Then, the coefficients corresponding to the locations where the mark might have been inserted are recorded, thus obtaining a $\tilde{c}_w$ vector with $M \times N$ real components. This vector is the input of a Viterbi decoder [5]. The decoder is pair designed with the trellis encoder. The cost involved in the Viterbi algorithm is the (un-normalised) correlation coefficient between the input sequence and the transition labels. This cost is to be maximised. Hence, high detection performances are obtained when these labels are uncorrelated.
Block #4: Informed embedding

This block [9] is designed by adapting the principles in [3,4]. Its aim is to embed the mark (the \( g \) vector) into the document (represented by the \( c_0 \) vector). Under the informed watermarking framework, the crucial issue is to find a \( c_w \) vector which is as close as possible to the \( c_0 \) vector and for which the Viterbi decoder produces the same output as for the \( g \) vector.

This \( c_w \) vector is computed by an iterative algorithm. In the first iteration, \( c_w \) is initialised with \( c_0 \). Further on, a vector denoted by \( b \) is computed by applying the Viterbi decoder to \( c_w + n \), and by trellis encoding the resulting bits. Here, \( n \) is a vector of \( M \times N \) length, whose components are sampled from a noise source modelling the channel perturbations. This noise is computed as a sum of a Gaussian noise - considered until recently as a universal model for the watermarking channel noise - and a noise that models the non-Gaussian effects [10,11] of some transformations or attacks (e.g. the StirMark attack). The \( c_w \) vector is now modified according to the following formula:

\[
c_w \leftarrow c_w + \alpha \cdot (g - b)/|g - b|.
\]

The \( \alpha \) scalar value is computed as follows:

\[
\alpha = R_l - R(g,b,c_w),
\]

where \( R(g,b,c_w) = c_w \cdot (g - b)/|g - b| \) and \( R_l \) is a scalar. The dot product between the \( c_w \) and the \((g-b)\) vectors is the un-normalised correlation coefficient.

The loop of \( b \) computation and \( c_w \) modification is repeated until the condition \( R(g,b,c_w) \geq R_l \) is reached several times successively (e.g. 100 times – \( N_j = 100 \)). If the equality between the \( g \) and the \( b \) vectors is reached before the \( R(g,b,c_w) \geq R_l \) condition is achieved, then the \( b \) vector is computed without modifying \( c_w \). If such a situation is encountered many times successively (e.g. 100 times – \( N_j = 100 \)) then we consider that the \( g \) mark was successfully embedded into the \( c_w \) vector: regardless the added noise, the decoder can recover the message. The thus computed \( c_w \) vector replaces the \( c_0 \) salient vector and the marked document is obtained by performing the inverses of the operations in Block #2.

Block #5: The channel

The marked document withstands a large variety of transformations.

Generic transformations include format or representation changes, compression, document editing and changing, malicious attacks. Generally, in watermarking, all these transformations are implicitly assumed to be Gaussian distributed. However, our recent studies [10,11] on multimedia data statistical behaviour brought into evidence that this Gaussian assumption does not hold for challenging attacks, like the StirMark attack [12], for instance. Consequently in our watermarking scheme we consider two types of perturbations: (1) Gaussian distributed which can model the generic transformations and (2) attack specific noise sources.

3 Experimental results

3.1 Parameter instantiation

The experimental data consists of 20 video sequences, each of them having 1000 frames (40 sec). The frame sizes are 192×160 pixels, corresponding to a Motorola V550 cell phone.

The 2D-DWT is applied at a \( N_f = 3 \) resolution level.

The original message to be inserted is represented on \( M = 1000 \) bits and corresponds to the binary SFR logo (just for illustration see Fig. 6.a). Each bit from this message is trellis encoded by a \( N = 360 \) real number label. These numbers are extracted from a random generator obeying a Gaussian distribution of \( \mu = 0 \) mean and \( \sigma = 0.005 \) standard deviation. The \( D \) number of DWT coefficients selected from each frame is \( D = M \times N / L = 360 \).

The \( R_l \) parameter involved in the embedding scheme was set to \( R_l = 2 \). The noise generator considers an \( n_g \) Gaussian noise of \( \mu = 0 \) mean and \( \sigma = 0.2 \) standard deviation and a \( n_a \) StirMark noise [10-12].

3.2 Experimental results

In order to subjectively evaluate the transparency, 25 human observers of different ages were involved in our experiments: 5 researchers deeply involved in the image/video...
processing, 5 researchers working in fields not connected with video processing, 5 persons with various educational backgrounds (foreign languages, history, law), 6 students, 1 film director, 1 film producer and 2 painters. They agreed that the method features fidelity. In order to also offer an objective measure of the transparency, the UIQI [13] (Universal Image Quality Index) was computed for each frame in the video sequence: their minimal, maximal and mean values are 0.9798, 0.9994, and 0.9981 respectively (a UIQI of 1 corresponds to identical images). Frames from original and marked Advertising sequences are represented in Figs. 4 and 5.

The method also features very good robustness. First, we check up the resistance against the mundane video processing: change of file format (from mpg to avi), linear and non-linear filtering (Gaussian, Laplace, median), small rotations (each frame was randomly rotated with up to 2 degrees), noise addition, spatial and temporal cropping (up to 25% of frames have been randomly dropped). Each and every time, the visual logo has been successfully recovered. Secondly, the StirMark [12] attack was individually applied to each frame in the sequence: although the commercial value of the video sequence was completely destroyed during this attack, the logo was still recovered. Fig. 6 illustrates the robustness. The logo recovered after the file format changing, Laplace filtering and the StirMark attack are represented in Fig. 6 a, b, and c, respectively.

The upper limit of the false alarm probability was evaluated at $10^{-12}$.

![Fig. 4. Original frames sampled from the Advertising sequence.](image1)

![Fig. 5. Transparency for video watermarking: marked frames sampled from the marked Advertising sequence, and corresponding to the originals in Fig. 5.](image2)

![Fig. 6. Robustness for video watermarking: the SFR logo recovered after the file format changing (a), Laplace filtering (b) and the StirMark attack (c). Note that the logo (a) is practically identical to the original logo.](image3)

4 Final remarks

This paper presents an outstanding watermarking method (joint GET/INT and SFR patent pending): it affords a very large quantity of information (about 1000 bits) to be inserted into a short video sequence (about 40s), while observing to the strongest constraint on robustness (the StirMark attack). These performances are obtained even for very low bitrates (e.g. 64kbit/s). Moreover, the method also proved to be very easily adaptable for audio and 3D data.

For video interactivity, the robustness constraints are greatly alleviated: there is no
more interest in removing the mark (hence, no more attacks, and the mark should only survive mundane transformations). Actually, this application deals with data hiding rather than watermarking. Our experiments show that under these circumstances the data payload can be increased about 100 times, i.e. about 12.5 kbytes can be inserted into a short video sequence (40s). Moreover, when the frames have larger dimensions (e.g. 1280x960), this quantity is increased by a factor of 24, thus allowing about 288 kbytes to be inserted into a 40s video.

These performances allow us to exploit this method in order to ensure video interactivity, as described in the two examples [14] below.

A first application consists in creating a video presentation, see Fig. 7.a: each slide represents a picture and the user can navigate among the slides (of course, this application is similar to a photo album). The interactivity information (scene specification and related Java script code) requires about 2kbytes. In order to use our method, the presentation (or the photo album) should have at least 7 slides (or photos), which is always the case.

Let us consider now the case of a video sequence enriched with a pinball game, Fig. 7.b. The user can watch the movie into a very small round window which enlarges according to the points the user scores. Such an application would require about 6kbytes of extra data which can be very easily inserted in any video sequence (even in the worst case scenario, it would require only 80s of video).

Fig. 7. Two scenarios for video data enriched with interactivity information; notice the interactivity information: the navigation buttons (on left) and the score, the ball and pinball control (on right).

References