Blind digital watermarking in PDF documents using Spread Transform Dither Modulation

Ahmad Bitar, Rony Darazi, Jean-François Couchot, Raphael Couturier

Noname manuscript No.
(will be inserted by the editor)

Blind digital watermarking in PDF documents using Spread Transform Dither Modulation.

Ahmad W. Bitar[1], Rony Darazi[1], Jean-François Couchot[2] and Raphaël Couturier[2]

Received: date / Accepted: date

Abstract In this paper, a blind digital watermarking scheme for Portable Document Format (PDF) documents is proposed. The proposed method is based on a variant Quantization Index Modulation (QIM) method called Spread Transform Dither Modulation (STDM). Each bit of the secret message is embedded into a group of characters, more specifically in their x-coordinate values. The method exhibits experiments of two opposite objectives: transparency and robustness, and is motivated to present an acceptable distortion value that shows sufficient robustness under high density noises attacks while preserving sufficient transparency.

Keywords Digital Watermarking, Portable Document Format, Quantization Index Modulation, Spread Transform Dither Modulation, Transparency, Robustness.

1 Introduction

Nowadays, the security of information has become a primordial issue especially with the rapid development of numeric transmission techniques. Among the most important techniques for the protection of information, we can find Digital Watermarking, Cryptography, Fingerprint and Steganography.

Digital Watermarking is the art of concealment [1] which consists in hiding a message (image, text, etc.) inside a digital media (image, text, video, audio, PDF, etc.) for copyright protection, hence the high importance of the cover work. The main idea behind this technique is that once a careful user detects the presence of the hidden message, he should be unable to remove that message without strongly altering the watermarked document.

A. W. Bitar and R. Darazi
Université Antonine, Hadat-Baabda, Lebanon. www.upa.edu.lb
E-mail: ahmad_bittar@hotmail.com, rony.darazi@upa.edu.lb

J.F. Couchot and R. Couturier
Université de Franche Comté, Belfort, France. www.univ-fcomte.fr
E-mail: {jean-francois.couchot, raphael.couturier}@univ-fcomte.fr

Portable Document Format [2], abbreviated as PDF, is a Page Description Language created by Adobe Systems Society, and considered as an evolution of PostScript format and whose specificity is to preserve the formatting of the file.

Several methods of Steganography and Digital Watermarking in PDF and Text documents have been proposed. In [3], a steganographic approach is presented by hiding information using inter-word and inter-paragraph spacing in a text. The main disadvantage of this method is that the hidden message can be destroyed by simply deleting some spaces between the words in the stego text. In [4], two different algorithms are proposed which are considered as an alternative for the original TJ operator method. The TJ operator displays the text string in a PDF document, allows individual character positioning and uses character and word spacing parameters from the text state. The alternative method has less embedding capacity than the original method. In [5], an encryption technique is proposed by combining the information hiding technique in PDF documents and the quadratic residue as basis and then apply it to copyright protection and digital learning. The main drawback of this method is that the hidden message can be easly removed. In [6], an embedding method in source programs using invisible $ASCII$ codes is proposed. This method is very easy to detect by simply extracting the modified text from the document, converting it to hexadecimal, extracting all the inserted invisible $ASCII$ characters, and then, decoding the embedded message. In [7], a data hiding in PDF files and applications by imperceivable modifications of PDF object parameters is proposed. This method serves to hide data by slight modifications of the values of various PDF object parameters such as media box and text matrices. The method is considered to have sufficient transparency while its main drawback is its very low embedding capacity.

Substitutive Quantization Index Modulation (QIM) methods were introduced by Chen and Wornell [8]. The Spread Transform Dither Modulation (STDM) is an implementation of this scheme and it has been considered robust under different watermarking attacks [9][10][11].

In this paper, the goal is to present a blind digital watermarking scheme for PDF documents based on a variant of the Quantization Index Modulation method called Spread Transform Dither Modulation (STDM). The main difficulty in PDF documents is to find a significant watermarking space in order to embed the secret message under a sufficient Transparency-Robustness tradeoff. Our contribution consists in using the x-coordinates of a group of characters to embed each bit of the secret message while choosing the appropriate mean distortion value which gives the strong tradeoff between transparency and robustness.

The remainder of this paper is organised as follows. In section 2, the PDF file structure is briefly summarized. Then, in section 3, a brief explanation on STDM concept is presented. The proposed embedding method is presented in section 4. Experimental results are shown in section 5. Finally, section 6 gives concluding remarks and some directions for future work.

2 PDF File Structure

All PDF files provide a common structure decomposed into 4 components (e.g., see [2]) as shown in Figure 1. Here we give a very simple example in order to understand how a string can be encoded in a PDF file.

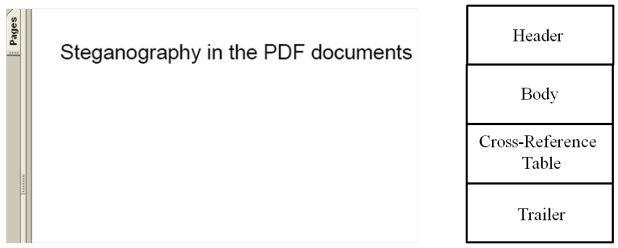

Fig. 1 PDF document and file structure

Header: contains the PDF file version. It also makes the application able to identify the file as being a PDF.
Body: contains series of objects such as Page, Font, etc. that collectively represent a PDF document. A PDF body supports eight types of objects: Boolean, Integer, String, Name, Array, Dictionary, Stream and Null. The 1 0 obj is the Root object having 1 as identifier and 0 as generator. It is a Catalog object (/Catalog) of type dictionary (<< >>). It contains the key version (/version) of value 1.4 (/1.4). Notice that version and 1.4 are two objects of type "name" since they are preceeded by a slash (/). It contains another key named Pages (/Pages) that represents a reference (R) to the object number 2.

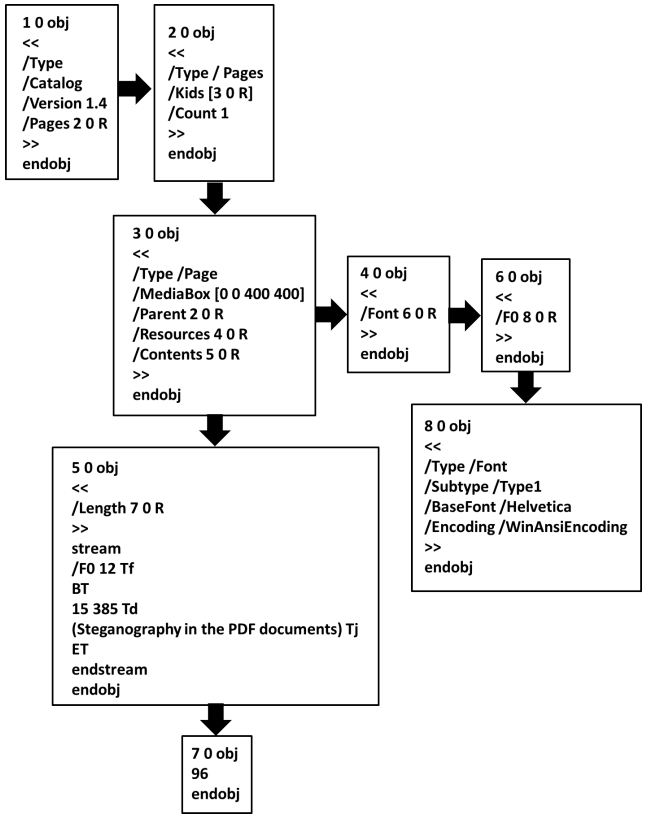

Fig. 2 Body example of the PDF shown in Figure 1

The 2 0 obj is a Pages object (/Pages) of type Dictionary. It contains the key Count (/Count) of value 1 because there is only 1 page in the document. The key Count is an object of type Name while 1 is of type Numeric. The object also contains a reference to the object number 3 (kids [3 0 R]) in order to represent the page in more details. The 3 0 obj is a Page object (/Page) of type Dictionary. It contains the length of the page (/MediaBox), a reference to the parent object number 2, a reference to the object 4 (4 0 obj) that contains a reference to the Font object (6 0 obj). The object 3 also contains a reference to the object 5 (5 0 obj). The object 5 contains a reference to the object 7 (7 0 obj) including the length of the string, and all the information about the stream such as the font and size (Tf operator), the positioning of the string (Td operator), and the text showing (Tj operator). In this example, "15 385 Td" represents the offset of the beginning of the current line "Steganography in the PDF documents" in the document (Td operator: move to the start of the next line and offset from the start of the current line by (tx, ty) [2]). Therefore, 15 and 385 refer to the x and y coordinates of the first character 'S', respectively. The other characters take their corresponding x-coordinates values depending on the spacing in horizontal writing (defined by the Tc operator and which is equal to zero by default (Tc=0)) between the characters. Notice that BT and ET represent the Begin Text and End Text, respectively. Finally the object 6 (6 0 obj) contains a reference to the object 8 (8 0 obj) where this last specifies the font used (Helvetica) and the applied encoding (WinAnsiEncoding).

As a result, all these objects are organized as a linked list where each node represents an object as shown in Figure 3.

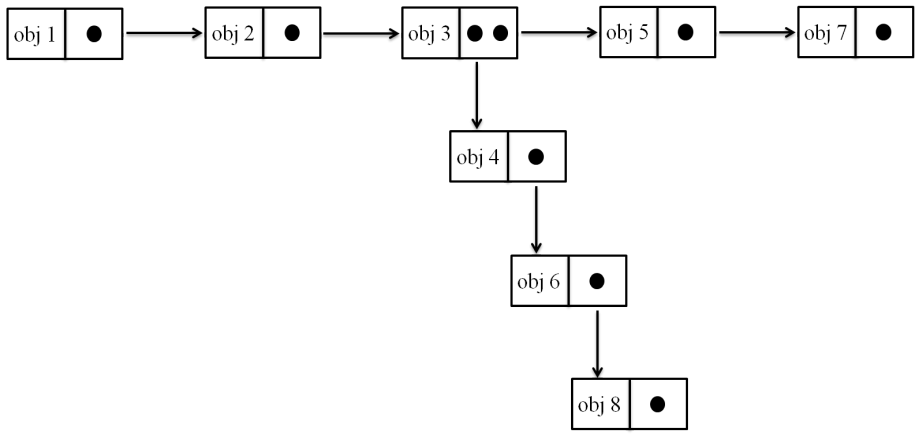

Fig. 3 Body Linked List

Cross-Reference Table: each Cross-Reference table begins with a line containing the keyword xref and all the next lines are exactly 20 bytes long, including the end-of-line marker as shown on the left of Figure 4. The first number after xref says that this list starts at object 0. But a "0 0 obj" does not exist in the PDF file because it is a special sort of entry that represents the head of a linked list. That is why, the first line in this list has a "f" at the end. The second number after xref is a count of how many objects are in this Cross-Reference Table. The lines with "n" at the end refer to the objects existing in the body section. Therefore, each

indirect object has its own line in the Cross-Reference table which includes the location (offset) of the object to be accessed in the body.

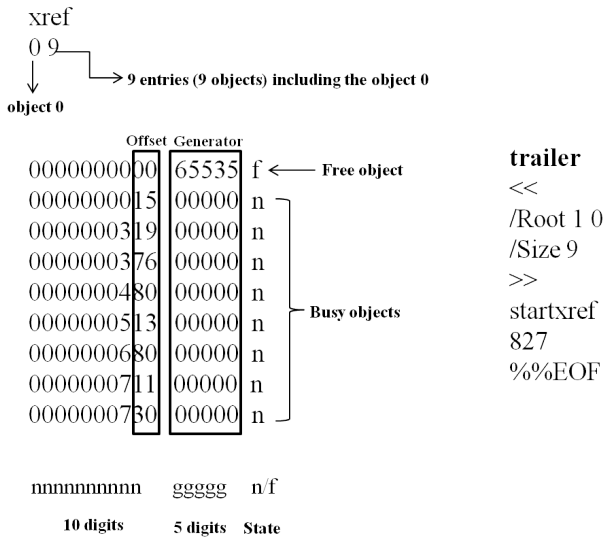

Fig. 4 Cross-Reference Table and Trailer structures

Trailer: the trailer is used to find the xref table which will enable it to locate certain specific objects within the body of the file as shown on the right of Figure 4 . The trailer is a dictionary containing a link to the Root object, the total number of objects (/size 9), the keyword startxref, the offset of the Cross reference table to access it and, finally, the End Of the File (EOF).
The PDF file is therefore executed as follows: Header - Trailer - Cross Ref - Body.

3 Spread Transform Dither Modulation

In order to present the QIM based method, a bit message $m \in \{0, 1\}$ is considered to be embedded in a host signal x. Therefore, according to the value of the embedded bit m, two different dither quantizers are used. To embed the bit message m=0, the dither quantizer Q_0 is used as:

$$Q_0(x,\ \Delta) = \lfloor \frac{(x - d_0)}{\Delta} \rfloor \Delta\ + d_0 \tag{1}$$

While Q_1 is used to embed the bit message m=1

$$Q_1(x,\ \Delta) = \lfloor \frac{(x - d_1)}{\Delta} \rfloor \Delta\ + d_1 \tag{2}$$

where Δ is the Quantization Step Size, also called Quantization Factor. $\lfloor . \rfloor$ denotes a rounding operation. The real values d_0 and d_1 represent the dither levels

$$d_0 = -\frac{\Delta}{4} \quad \text{and} \quad d_1 = \frac{\Delta}{4} \tag{3}$$

Notice that d_0 can also be chosen pseudo randomly from a uniform distribution over $[-\Delta/2, \Delta/2]$. In such a situation, according to the sign of d_0, $\Delta/2$ can be either added or subtracted from d_0 to form d_1.

$$d_1 = \begin{cases} d_0 + \Delta/2, & \text{if } \ d_0 < 0 \\ d_0 - \Delta/2, & \text{otherwise} \end{cases}$$

In the STDM method, each bit of the message is inserted into a sample vector x of length L of the host signal and the quantization occurs entirely in the projection of the host signal using projection vector p. The most important advantage of this method is that the embedding-induced distortion is spread into all the groups of samples instead of into one sample only. That is why this type of dither modulation is called Spread Transform Dither Modulation.

The quantized signal is given by :

$$x' = x + (Q_m(x^T p,\ \Delta) - x^T p)p \quad m \in \{0, 1\} \tag{4}$$

The equation (4) can be re-written as:

$$x' = x + ((\lfloor(\frac{(x^T p) - d_m}{\Delta})\rfloor \Delta + d_m)\ - x^T p)p \tag{5}$$

The extraction of the embedded message can be performed by using a minimum distance decoder as of the form:

$$ExtMessage = arg \min_{m \in \{0,1\}} \mid x'^T p - Q_m(x'^T p,\ \Delta) \mid \tag{6}$$

The average expected distortion [7] is:

$$D_s = \Delta^2/12L \tag{7}$$

4 Proposed Method

4.1 Embedding concept

The embedding process can be divided into 6 steps:
Step 1 – The message is ciphered by applying a XOR operation between the binary message and a random secret key. Any other Cryptographic algorithm can be used.

Step 2 – The original document is read, and then all the necessary resources (x-coordinate, y-coordinate, width, height, etc.) are founded of each character that exists in the document. Let k be the length of the binary cipher message. Thus the algorithm requires $k \times L$ ressources to embed the whole secret message.

Step 3 – The host signal is created and which corresponds to the x-coordinates of all the selected characters to be modified or quantized.

Step 4 – Each bit of the encoded message is embedded into L different values ($L \geq 1$) of the host signal created in step 3 corresponding to the x-coordinate of the characters to be modified. The embedding function is applied as shown in

equation (4).
Assume that m_0 and m_1 shown in Figure 5 are two bits of the secret message to be embedded in the x-coordinate values, where L=8. Thus, to embed m_0, both the quantizer Q_0 and the dither level d_0 are used, while Q_1 and d_1 are used to embed m_1.

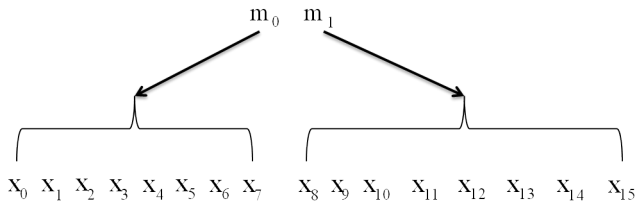

Fig. 5 Basic embedding example using 2 bits and L=8

As a result, to embed a message formed by k bits into the document where each bit is embedded into L samples, we need $k \times L$ characters to modify. In other words, each character in the document has its own (x, y), therefore, if L is chosen to be 8, each bit of the encoded message being inserted into 8 values that correspond to the x-coordinate of 8 characters $(x_0, y_0), (x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4), (x_5, y_5), (x_6, y_6), (x_7, y_7)$.
After the embedding process, the 8 characters become:
$(x'_0, y_0), (x'_1, y_1), (x'_2, y_2), (x'_3, y_3), (x'_4, y_4), (x'_5, y_5), (x'_6, y_6), (x'_7, y_7)$ where x'_i is calculated as in equation (5).

Step 5 – After the embedding process, each character a_i takes its corresponding modified coordinate (x'_i, y) and be re-written separately in the document as shown in (b) of Figure 6,

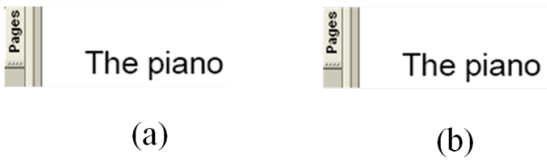

Fig. 6 (a) Original document, (b) Watermarked document

Step 6 – Finally, the embedded message can be extracted by applying equation (6).

4.2 Discussion problem

Equation (7) has shown that the distortion is quadratic in Δ for a given L. We have represented this function in Figure 7 with $0 \leq \Delta \leq 3$ and $0 \leq L \leq 100$.

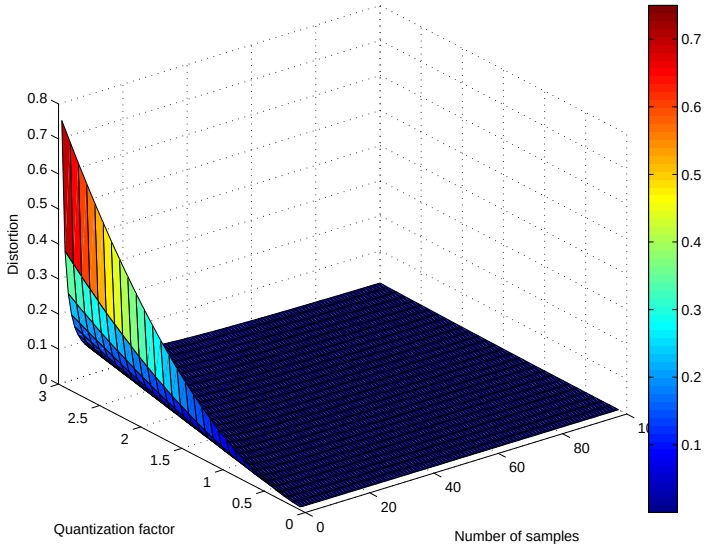

Fig. 7 3D representation of the distortion D_s

The user is then left to choose a D_s value that would lead a sufficient robustness with sufficient transparency. In the proposed method, some distortions are considered acceptable whereas others are not. But the remaining question to be solved is "What makes a distortion acceptable". In other words, what is the value of D_s for which the method shows sufficient robustness with sufficient transparency. However, transparency and robustness are two opposite objectives. In our method, there are basically two threshold levels to consider, namely a and b. The transparency threshold level a is always computed by the transparency experiments while the robustness threshold level b is computed by the robustness experiments.

If the distortion D_s is inferior to a, we thus have a sufficient transparency. On the opposite, if D_s is greater than b, the method ensures sufficient robustness (but weak transparency). There are thus two cases to consider: the former is when a is inferior to b. In such a situation, for any value of D_s, the corresponding distortion is either inferior to b (and the robustness is not established) or greater than a and the transparency is weak. The latter is when $b \leq a$. In such a situation, the interval $b \leq D_s \leq a$ corresponds to the acceptable distortion values that can show sufficient robustness with sufficient transparency. Let us consider an example which includes both cases: *If* $b = 0.5$ *and* $a = 0.2$ in this case, we have $b > a$ and D_s can either be greater than or equal to 0.5 (sufficient robustness with weak transparency) or inferior to 0.5. In this latter case, D_s can either belong to the interval [0.2 0.5[(weak robustness with weak transparency), or be inferior to 0.2 (weak robustness with sufficient transparency). *If* $b = 0.2$ *and* $a = 0.5$ in this case, we have $b \leq$ a and D_s can either be greater than 0.5 (sufficient robustness with weak transparency) or inferior than or equal to 0.5. In this latter case, D_s can either belong to the interval [0.2 0.5] (**sufficient robustness with sufficient transparency**), or be inferior to 0.2 (weak robustness with sufficient transparency).

5 Experiments

Several transparency and robustness experiments are performed in order to deduce the strong approximation values of a and b. All the experiments were computed by function of Δ. Three cases can be considered:

- **Case 1**: a balance between the number of characters (length) in the document and the message to be embedded.
- **Case 2**: the number of characters in the document is increased in order to have a large document while keeping the same message length used in case 1.
- **Case 3**: the length of the message is shortened while keeping the same length of the document used in case 1.

The threshold values of a and b are thus deduced from case 1 since they are always accepted by both cases 2 and 3. It can be explained by the fact that both cases 2 and 3 are able to represent better transparency-robustness tradeoff than case 1. In order to argument our approach, we present a brief example on a PDF document and message of case 1. The proposed method has been implemented in JAVA using the Netbeans program. Let us consider the original document : Violin.pdf shown in the top-left hand side of Figure 8 and the message to be embedded : UFC. The violin document contains $n = 947$ characters. Each character of the message is encoded into 8 bits in order to form a total of $k = 24$ bits. Each bit message is then embedded into $L = E(n/k = 39.458) = 39$ characters' x-coordinates extracted during step 2 of section 4. Therefore, a total of $k \times L = 936$ characters are used from the document to embed the whole 24 bits of the message.

5.1 Tests of Transparency (Violin.pdf, UFC)

Three different kinds of experiments (error measurements, perceptual PDF differences and distortion plots) are presented in order to test the transparency of the proposed method under several values of Δ: 0.1, 0.5, 1, 1.5, 2, 2.5, 3, 5 and 10. Table 1 presents error measurements between the original and the modified documents after watermarking using three different metrics: Mean Square Error (MSE), Root Square Error (RSE) and Mean Absolute Error (MAE). The results show that error values increase when Δ increases. Figure 8 exhibits a perceptual difference between the original and modified document and the results show a slight modification in the characters' position when Δ is small while notable modification when Δ is high (equal to 5 or 10 for example). Figure 9 exhibits clearly how the positioning of some characters after watermarking is affected by simply comparing the deviation of the x marks in relation to the center of o marks. The x marks are exactly centered into the o marks when the distortion is very low.

Table 1 Error computations between the original and modified violin documents in terms of their x-coordinate values

Error tests	MSE	RSE	MAE
Δ=0.1 (D_s = 0.00002)	$\mathbf{2.5527 \times 10^{-5}}$	**0.0051**	**0.0038**
Δ=0.5 (D_s = 0.00053)	$\mathbf{6.2815 \times 10^{-4}}$	**0.0251**	**0.0195**
Δ=1 (D_s = 0.00214)	**0.0020**	**0.0449**	**0.0338**
Δ=1.5 (D_s = 0.00481)	**0.0063**	**0.0794**	**0.0619**
Δ=2 (D_s = 0.00855)	**0.0118**	**0.1085**	**0.0898**
Δ=2.5 (D_s = 0.01335)	**0.0127**	**0.1129**	**0.09**
Δ=3 (D_s = 0.01923)	0.0222	0.1491	0.1082
Δ=5 (D_s = 0.05342)	0.0537	0.2317	0.1675
Δ=10 (D_s = 0.21367)	0.1696	0.4118	0.2904

Fig. 8 Perceptual PDF difference– violin.pdf and modified_violin.pdf using $\Delta = 0.1$, $\Delta = 0.5$, $\Delta = 1$, $\Delta = 1.5$, $\Delta = 2$, $\Delta = 2.5$, $\Delta = 3$, $\Delta = 5$ and $\Delta = 10$, respectively. The document shown in the top-left hand side is the original document.

All the transparency experiments shown in Table 1, Figure 8 and 9 prove that the higher the value of Δ is, the more the transparency decreases. Based on these experiments, we assume that for a distortion D_s greater than 0.01335, any perceptual difference between the original and the watermarked document can be

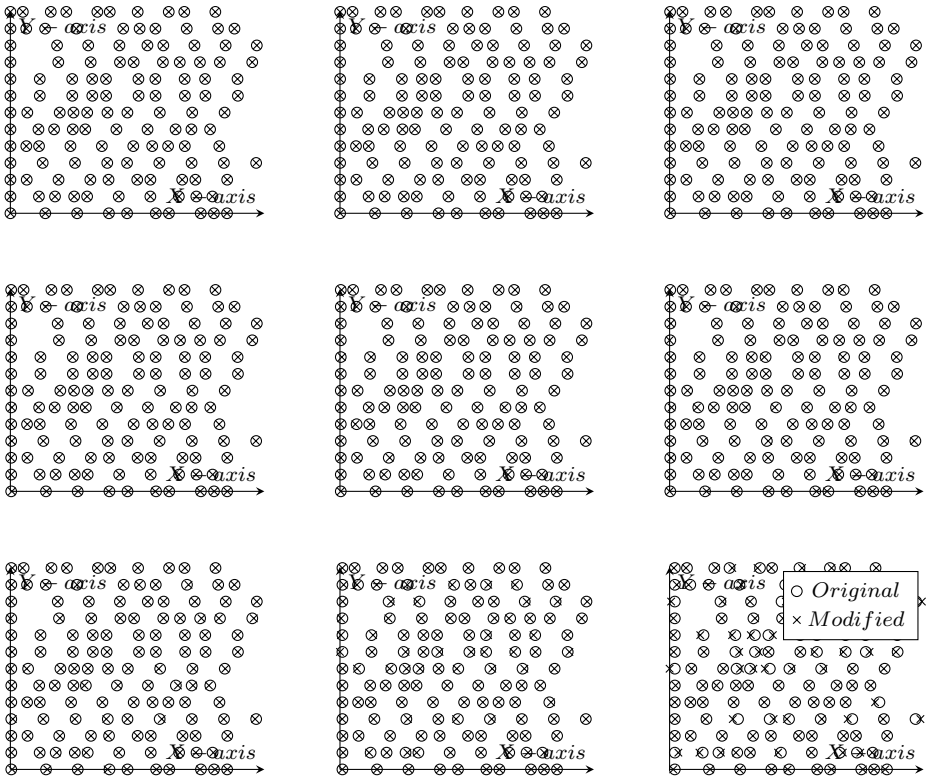

Fig. 9 Distortion plots– For L=39, the distortion is computed using some of the characters using $\Delta = 0.1$, $\Delta = 0.5$, $\Delta = 1$, $\Delta = 1.5$, $\Delta = 2$, $\Delta = 2.5$, $\Delta = 3$, $\Delta = 5$ and $\Delta = 10$. The x marks are exactly centered into the o marks when the distortion is very low.

noticed. Thus the transparency threshold level a is equal to 0.01335. The distortion values that are selected to show good transparency are shown in bold in Table 1.

5.2 Tests of Robustness

Experiments are done on the Violin PDF document shown in the top-left hand side of Figure 8 and the embedded message "UFC" using L=39. Two different watermarking attacks: Gaussian and Salt&Pepper noises are applied to the x-coordinates of the characters in the watermarked document. Only the digits after the decimal point are modified. After the attacks, the extracted message is compared to the original message by computing the Pearson's linear correlation coefficient (corr), the Mean Square Error (MSE) and the Bit Error Rate (BER). The simulations were repeated 500 times. Table 2 and 3 illustrate an average of all the robustness results (corr, MSE and BER), and from their values, we notice that the higher the value of Δ is, the more the robustness increases. Since two noises attacks (Gaussian and Salt&Pepper) under two densities (0.1 and 0.25) are applied, therefore we will get four different robustness threshold levels: b_1, b_2, b_3 and b_4. b_1 and b_3 are computed respectively from the experiments of Gaussian and Salt&Pepper

noises under a density equal to 0.1, while b_2 and b_4 under a density equal to 0.25. The robustness threshold level b is therefore computed. It corresponds to the best robustness under all the watermarking attacks. In our experiments, we consider that BER = 12.5% can be tolerated to deduce the values of b_1, b_2, b_3 and b_4. This is motivated by the fact that this percentage of BER which corresponds in our experiments to a total of 4 error bits from $k = 24$, can be corrected by the majority of error correcting codes. Each robustness threshold level of each noise attack under each density value is thus equal to the distortion D_s from which all the error bits (inferior than or equal to 4) can be corrected.

Table 2 Tests of robustness under gaussian attack

DELTA	Density	corr	MSE	BER
0.1 ($D_s = 0.00002$)	0.1	-0.0714	0.5233	12.5600
	0.25	-0.0545	0.5162	12.3900
0.5 ($D_s = 0.00053$)	0.1	-0.0503	0.5161	12.3860
	0.25	-0.0508	0.5242	12.5800
1 ($D_s = 0.00214$)	0.1	0.2713	0.3528	8.4680
	0.25	0.0503	0.4638	11.1320
1.1 ($D_s = 0.00258$)	0.1	0.3587	0.3081	7.3940
	0.25	0.1204	0.4281	10.2740
1.2 ($D_s = 0.00308$)	0.1	0.4222	0.2709	6.5020
	0.25	0.1525	0.4105	9.8520
1.3 ($D_s = 0.00361$)	0.1	0.4834	0.2415	5.7960
	0.25	0.2187	0.3661	8.7860
1.4 ($D_s = 0.00418$)	0.1	0.5555	0.2066	4.9580
	0.25	0.2836	0.3498	8.3960
1.5 ($D_s = 0.00481$)	0.1	0.6271	0.1716	4.1180
	0.25	0.3596	0.3062	7.3480
1.6 ($\mathbf{D_s = 0.00547}$)	**0.1**	**0.6510**	**0.1587**	**3.8080**
	0.25	0.3779	0.2953	7.0880
1.7 ($D_s = 0.00617$)	0.1	0.7019	0.1358	3.2580
	0.25	0.4577	0.2539	6.0940
1.8 ($D_s = 0.00692$)	0.1	0.7381	0.1180	2.8320
	0.25	0.5400	0.2133	5.1200
1.9 ($D_s = 0.00770$)	0.1	0.7701	0.1042	2.5000
	0.25	0.5593	0.2036	4.8860
2 ($D_s = 0.00855$)	0.1	0.7956	0.0921	2.2100
	0.25	0.5881	0.1914	4.5940
2.1 ($D_s = 0.00942$)	0.1	0.8113	0.0851	2.0420
	0.25	0.6330	0.1672	4.0140
2.2 ($\mathbf{D_s = 0.01034}$)	0.1	0.8328	0.0755	1.8120
	0.25	**0.6688**	**0.1503**	**3.6060**
2.3 ($D_s = 0.01130$)	0.1	0.8509	0.0670	1.6080
	0.25	0.6917	0.1397	3.3520
2.4 ($D_s = 0.01230$)	0.1	0.8698	0.0585	1.4040
	0.25	0.7307	0.1221	2.9300
2.5 ($D_s = 0.01335$)	0.1	0.8715	0.0578	1.3860
	0.25	0.7589	0.1089	2.6140
3 ($D_s = 0.01923$)	0.1	0.8972	0.0463	1.1100
	0.25	0.8425	0.0708	1.7000
5 ($D_s = 0.05342$)	0.1	0.9075	0.0417	1
	0.25	0.9062	0.0423	1.0140
10 ($D_s = 0.21367$)	0.1	0.9075	0.0417	1
	0.25	0.9075	0.0417	1

Table 2 and 3 present respectively the tests of robustness under Gaussian and Salt&Pepper noises attacks with two density values: 0.1 and 0.25. For a density equal to 0.1, we notice from Table 1 that for $D_s \geq 0.00547$, the average BER is less than or equal to 3.8080, while 3.3620 for $D_s \geq 0.00308$ from Table 2. Therefore b_1 and b_2 are equal to 0.00547 and 0.00308, respectively. For a density equal to 0.25, Table 1 shows that the average BER that can be entirely corrected is less than or equal to 3.6060 for the interval of $D_s \geq 0.01034$, while 3.6200 for the interval of $D_s \geq 0.00692$ from Table 2. Therefore b_3 and b_4 are equal to 0.01034 and 0.00692, respectively. The four threshold levels are shown in bold in Table 2 and 3.

Table 3 Tests of robustness under Salt&Pepper attack

DELTA	Density	corr	MSE	BER
0.1 ($D_s = 0.00002$)	0.1	-0.0502	0.5113	12.2720
	0.25	-0.0634	0.5208	12.4980
0.5 ($D_s = 0.00053$)	0.1	0.0915	0.4400	10.5600
	0.25	-0.0510	0.5143	12.3420
1 ($D_s = 0.00214$)	0.1	0.5701	0.1976	4.7420
	0.25	0.1810	0.3889	9.3340
1.1 ($D_s = 0.00258$)	0.1	0.6305	0.1691	4.0580
	0.25	0.2580	0.3649	8.7580
1.2 ($D_s = 0.00308$)	**0.1**	**0.6914**	**0.1401**	**3.3620**
	0.25	0.3268	0.3212	7.7080
1.3 ($D_s = 0.00361$)	0.1	0.7401	0.1178	2.8260
	0.25	0.4042	0.2816	6.7580
1.4 ($D_s = 0.00418$)	0.1	0.7796	0.0996	2.3900
	0.25	0.4715	0.2469	5.9260
1.5 ($D_s = 0.00481$)	0.1	0.8034	0.0884	2.1220
	0.25	0.5065	0.2263	5.4320
1.6 ($D_s = 0.00547$)	0.1	0.8216	0.0802	1.9260
	0.25	0.5716	0.1969	4.7260
1.7 ($D_s = 0.00617$)	0.1	0.8467	0.0689	1.6540
	0.25	0.6088	0.1793	4.3020
1.8 ($D_s = 0.00692$)	0.1	0.8657	0.0603	1.4460
	0.25	**0.6682**	**0.1508**	**3.6200**
1.9 ($D_s = 0.00770$)	0.1	0.8730	0.0570	1.3680
	0.25	0.7114	0.1307	3.1380
2 ($D_s = 0.00855$)	0.1	0.8759	0.0558	1.3400
	0.25	0.7242	0.1247	2.9920
2.1 ($D_s = 0.00942$)	0.1	0.8845	0.0519	1.2460
	0.25	0.7621	0.1074	2.5780
2.2 ($D_s = 0.01034$)	0.1	0.8942	0.0476	1.1420
	0.25	0.7746	0.1014	2.4340
2.3 ($D_s = 0.01130$)	0.1	0.8989	0.0455	1.0920
	0.25	0.8030	0.0885	2.1240
2.4 ($D_s = 0.01230$)	0.1	0.9019	0.0442	1.0600
	0.25	0.8261	0.0782	1.8760
2.5 ($D_s = 0.01335$)	0.1	0.9032	0.0436	1.0460
	0.25	0.8388	0.0724	1.7386
3 ($D_s = 0.01923$)	0.1	0.9066	0.0421	1.0100
	0.25	0.8804	0.0537	1.2880
5 ($D_s = 0.05342$)	0.1	0.9075	0.0417	1
	0.25	0.9075	0.0417	1
10 ($D_s = 0.21367$)	0.1	0.9075	0.0417	1
	0.25	0.9075	0.0417	1

Figure 10 and 11 present the results of the BER and correlation (shown in Table 2 and 3) by function of Δ, respectively. The figures serve to compare between the Gaussian and Salt&Pepper noises attacks under the two density values: 0.1 and 0.25. They exhibit 4 different curves. The red (dashed) and blue (dashdotted) curves represent the Gaussian noise under the two densities 0.1 and 0.25, respectively. The green (dotted) and black (dash pattern) curves represent the Salt&Pepper noise under the two densities 0.1 and 0.25, respectively. The plotted curves prove that the Salt&Pepper attack is always more robust than the Gaussian attack even under the two density values.

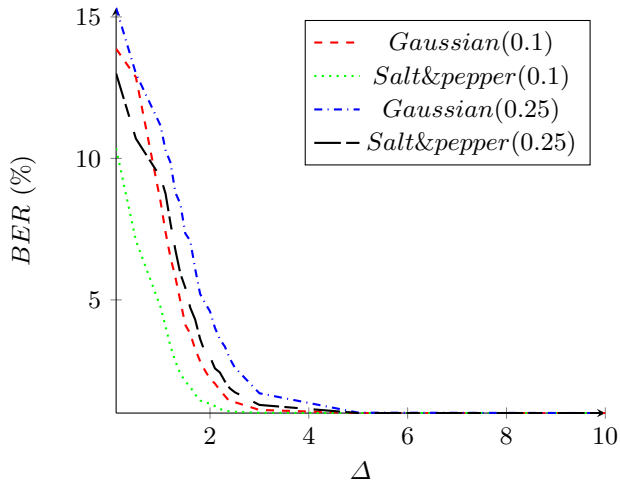

Fig. 10 Gaussian and Salt&Pepper comparisons in terms of BER

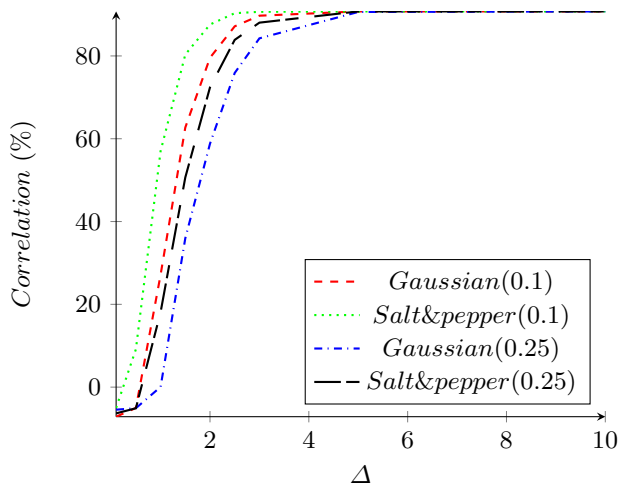

Fig. 11 Gaussian and Salt&pepper comparisons in terms of correlation

5.3 Robustness with Transparency

We have found $0.00547 \leq D_s \leq 0.01335$ for $1.6 \leq \Delta \leq 2.5$ and $0.00308 \leq D_s \leq 0.01335$ for $1.2 \leq \Delta \leq 2.5$ under Gaussian and Salt&Pepper attacks with density $= 0.1$, respectively. While $0.01034 \leq D_s \leq 0.01335$ for $2.2 \leq \Delta \leq 2.5$ and $0.00692 \leq D_s \leq 0.01335$ for $1.8 \leq \Delta \leq 2.5$ under Gaussian and Salt&Pepper attacks with density $= 0.25$, respectively.
The final acceptable distortion interval that can show sufficient rorbustness and transparency under all the watermarking attacks at the same time is the distortion D_s that belongs to the interval [0.01034 0.08] and it is represented with the blue curve in Figure 12. The red (dashed), green (dotted), black (dash pattern) and blue (dashdotted) curves refer to the Gaussian (density =0.1), Salt&Pepper (density=0.1), Salt&Pepper (density=0.25) and Gaussian (density = 0.25), respectively.

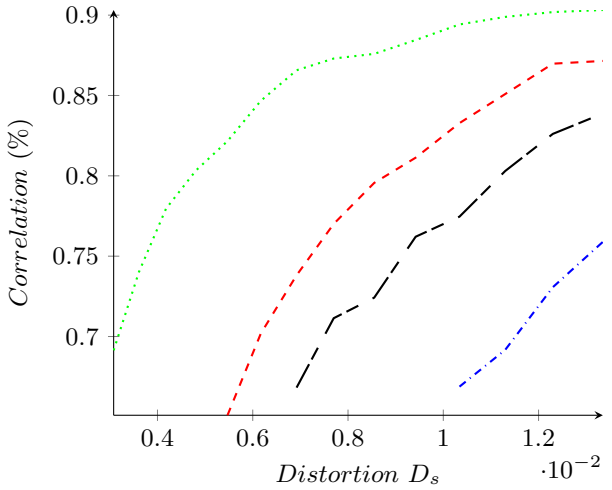

Fig. 12 Robustness correlations of all possible acceptable distortion under all the watermarking attacks

5.4 Our method Vs Related work

Our method has shown an new watermarking scheme to embed the secret message under a sufficient Transparency-Robustness tradeoff. In contrast to what has been proposed in [4] and [7], our method presents better transparency and higher embedding capacity. For example in [7], the message was embedded by slightly modifying the decimal values of the media box and text matrices, which means that the increase in the number of characters in the document does not affect the embedding capacity of the method. That is why, we exploited the characters for the embedding. More specifically, we exploited the x-coordinates of the characaters for the embedding and we used each group of them to embed one bit message by taking advantage of the STDM concept. In this case our method shows sufficient transparency and sufficient robustness at the same time where the embedded message becomes hard to be removed in contrast to what is deduced in [5]. It also

provides an efficient solution of [3] and [6] by making the detectability of the message more difficult. The y-coordinates values were not used because they are constant for the characters of the same line, which can increase the detectability.

6 Conclusion and Future work

In this work, we have shown in details the four different components of a PDF file structure: Header, Body, Cross-Reference Table and Trailer. The structure has been exploited to be used for an efficient blind digital watermarking scheme in terms of Transparency-Robustness tradeoff. The proposed scheme was based on a variant of the Quantization Index Modulation (QIM) method called Spread Transform Dither Modulation (STDM). Since the x-coordinates values of the characters presented in the document are non-constant especially those belonging on the same line, they have been exploited to embed each bit of the secret message.

The main contribution of this work was to achieve sufficient resistance against very high density noises attacks while preserving sufficient transparency at the same time. One of the biggest difficulties was to perform multiple transparency and robustness evaluations in order to estimate the strong value of distortion that would lead a sufficient robustness with sufficient transparency. That is why this work relies on two distinct threshold levels a and b which are computed by exploiting the transparency and robustness experiments, respectively. The strong distortion value D_s that would lead to a sufficient robustness with sufficient transparency should be neither greater than a nor inferior to b. The value satisfying this condition is called "The acceptable distortion".

As for future enhancements, we plan to extend this work into both practical and theoretical directions. In the practical part, we plan to find how robust is the approach against the JPEG compression. This hard task is challenging and presents direct applications into newspaper watermarking for instance. In the theoretical part, we plan to study how secure the STDM based approach is, *i.e.*, how many bit are sufficient to find the encoding key as in a classical cryptographic approach.

References

1. I. Cox, M. Miller, J. Bloom, J. Fridrich and T. Kalker. *Digital Watermarking and Steganography.* second edition, 624p, November 27, 2007.
2. Document management-Portable Document Format-Part1: PDF1.7. `http://www.adobe.com/content/dam/Adobe/en/devnet/acrobat/pdfs/PDF32000_2008.pdf`, 2008.
3. L. Y. POR and B. Delina, "Information Hiding: A New Approach in Text Steganography", *7th WSEAS Int. Conf. on APPLIED COMPUTER & APPLIED COMPUTATIONAL SCIENCE* (ACACOS'08), 7p, April 6-8, 2008.
4. F. Alizadeh, N. Canceill, S. Dabkiewicz and D. Vandevenne, "Using Steganography to hide messages inside PDF files", 34p, pp.1-11, December 30, 2012.
5. H. F. Lin, L. W. Lu, C. Y. Gun and C. Y. Chen, "A Copyright Protection Scheme Based on PDF", *International Journal of Innovative Computing, Information and Control*, Vol. 9, No.1, ISSN 1349-4198, pp.1-6, January 2013.
6. I.S. Lee and W. H. Tsai , "a new approach to covert communication via pdf files", *In Signal processing*, 557-565, 2010.
7. C. T. Wang and W. H. Tsai , "Data Hiding in PDF Files and Applications by imperceivable modifications of PDF Object Parameters", *Proceedings of 2008 Conference on Computer Vision, Graphics and Image Processing*, Ilan, Taiwan, Republic of China, 8p, pp.1-6, 2008.

8. B. Chen and G. W. Wornell, "Quantization Index Modulation Methods for Digital Watermarking and Information Embedding of Multimedia", *Journal of VLSI Signal Processing 27*, 7-33, 2001.
9. R. Darazi, R. H., Benoît Macq, "Applying Spread Transform Dither Modulation for 3D-MESH Watermarking by using Perceptual Models", *In proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing* (ICASSP), No. 1742-1745, pp.1742-1745, 2010.
10. W. Wan, J. Liu, J. Sun, X. Yang, X. Nie, F. Wang, "Logarithmic Spread-Transform Dither Modulation Watermarking Based On Perceptual Model", *In proceeding of 20th IEEE International Conference on Image Processing* (ICIP'2013), pp. 4522-4526, 15-18 September 2013.
11. B. Chen and G. W. Wornell, "Provably robust digital watermarking", *In proceeding of the International Society for Optics and Photonics* (SPIE'99), pp. 43-54, vol. 3845, 1999.

Ahmad W. BITAR was born in 1990. In 2013, he received the M.S degree in telecommunication engineering at Université Antonine (UA), Hadat-Baabda, Lebanon. From March 2013 to September 2013, he did a research internship entitled "Robust watermarking in PDF documents" at Université Antonine, and collaborated with the University of Franche-Comté (UFC).

Rony DARAZI is actually an associate professor at Université Antonine (UA). He received the M.S degree in computer science and telecommunication engineering in 2005, and the Ph.D. degree from the Université catholique de Louvain (UCL), Belgium. He was a researcher in the ICTEAM institute at UCL since 2006, and a member of the TICKET lab at UA since 2010. His research interests include information security and digital watermarking, digital 2D and 3D image processing. In 2009, he was granted the best paper award, 2nd price by the Digital Watermarking Alliance (DWA) and the IS&T/SPIE International Conference on Media Forensics and Security XII. Rony Darazi is an IEEE member; He has been actively involved as a reviewer in Signal, Image and Video Processing Journal by Springer, IEEE Transactions on Information Foren-

sics & Security and International Conference on Image Processing (ICIP).

Jean-François COUCHOT is an Associate Professor in the Department of Computer Science (DISC) of the FEMTO-ST institute (UMR 6174 CNRS) at the university of Franche-Comté. He received a Ph.D. in Computer Science in 2006 in the FEMTO-ST institute. He has applied for a postdoctoral position at INRIA Saclay Ile de France in 2006. His research focuses on discrete dynamic systems (with applications into data hiding, pseudorandom number generators, hash function) and on bioinformatics, especially in gene evolution prediction. He has written more than 25 scientific articles in these areas.

Raphaël COUTURIER received the Ph.D. degree in 2000 in Computer science from the Henri Poincare University in Nancy, France. From 2000 to 2006 he was an assistant professor at the University of Franche-Comté. Then he has been a professor at the same university. His research interests include parallel and distributed algorithms with a strong knowledge on asynchronous iterative methods, GPU and FPGA computing, sensor networks and watermarking. Raphaël authored or co-authored more than 80 papers in conferences and journals and two books. He has also served in many program committees for conferences.