Data Hiding for Printed Binary Documents Robust to Print-Scan, Photocopy and Geometric Attacks

Hae Yong Kim and Joceli Mayer

***Abstract*— This paper presents a data hiding technique for printed bicolor documents. It inserts tiny dots, hardly noticeable at normal reading distance, to embed the message. For message extraction, we employ auto-correlation and tiny registration dots to rectify geometric distortions. This technique is robust to distortions resulting from print-scan operations, good quality photocopies, affine transformations and scribblings/stains on the paper. The technique can be applied to documents with large white (or black) areas and they may present characters, drawings, schematics, diagrams, cartoons, but not halftones. The technique is intended to be neither a robust watermark (because any filtering can remove the dots) nor a covert communication (because the dots are perceptible at short distance). Nevertheless, when combined with a perceptual hashing and a cryptography protocol, it can be applied as semi-fragile authentication watermarking for hardcopy two-tone documents. In some situations, the utilization of the proposed system can substitute the use of notarial authenticated photocopies.**

***Index Terms*— Data hiding, semi-fragile watermarking, geometric attacks, print-scan and photocopy robustness.**

I. Introduction

THIS paper presents a data hiding technique (steganography) for embedding information into documents printed in high-resolution bicolor printers, e.g., conventional laser printers. In the literature, there are several data hiding techniques designed for binary images [1-10]. These techniques can be applied to copy control, annotation, and authentication. However, most of them are designed only for binary images in digital form and cannot be applied for printed documents. They can be divided into three basic classes:

1) Component-wise: Change the characteristics of some pixel groups (connected components, character, words, etc.) Some examples of the characteristics that can be changed are: the thickness of strokes, the position/area of characters/words [1], the brightness of the connected components [2], etc.
2) Pixel-wise: Change the values of individual pixels. Those pixels can be chosen randomly [3, 4] or according to some visual impact measure [5, 6].
3) Block-wise: Divide the cover image into blocks and modify some characteristic of each block to hide the data. Some papers suggest changing the parity or the quantization step of the number of black pixels in each block [7, 8]. Others suggest flipping one specific pixel in the block with m pixels to insert $\lfloor \log_2 (m+1) \rfloor$ bits [9, 10].

Only a few of these techniques can be used to hide data in printed binary images, because print-scan operation introduces many distortions. For instance, geometrical misalignments are introduced by moving parts of the printer and by the user when placing the document at the scanner bed. Physical and chemical deformations of the paper are originated by the high temperature and pressure of the toner fuser. Grayscale magnitude distortions are introduced by the optical and acquisition system of the scanner. Variable toner intensity distribution along the cylinder, paper texture and dirty introduce more distortions.

Most of data hiding techniques for printed binary images are component-wise and are designed only for a specific kind of image (for example, text documents) based on different approaches to modulate the message. Word or character shifting based techniques modulates the horizontal and/or vertical space between words or characters [1, 11, 12]. Character modulation schemes alter the character amplitude, texture or even the halftone [2, 13]. 1-D and 2-D bar codes techniques propose to authenticate the text by modulating a visible image (barcode) introduced into the document [14, 15]. For many of these techniques, the message bit rate is severely reduced when the documents present few characters, large white (or black) areas, handwritings, drawings, equations and official stamps. Another drawback is that many techniques require a perfect segmentation (of connected components, characters or words), which is hard to achieve in practice due to distortions introduced by printing, scanning and non-malicious geometric rotations.

Wu and Liu proposed a block-wise data hiding that can be applied for printed binary images in some specific scenarios [7]. However, this approach requires exceptionally high quality printing and scanning and is not intended to be robust to the aforementioned distortions. It requires precise detection of the boundaries of the document to identify the size, rotation and skewing of the image. The reader is referred to [2, 7] for a discussion about watermarking for printed text and binary documents.

In the literature [16-20], there are many papers on continuous tone (grayscale and color) images watermarking techniques resilient to rotation, scale and translation (for example [16, 17, 18]) and continuous tone data hiding approaches resilient to print-scan (for example [19, 20]). However, none of them can be directly applied to printed binary documents.

Manuscript received October 25, 2007; revised December 3, 2008. This work was supported in part by CNPq under grants 305065/2003-3 and 475155/2004-1.

H.Y. Kim is with the Escola Politécnica, Universidade de São Paulo, Av. Prof. Luciano Gualberto, tr. 3, 158 – CEP 05508-900, São Paulo, SP, Brazil. E-mail: hae@lps.usp.br.

J. Mayer is with LPDS/EEL, Universidade Federal de Santa Catarina, Brazil. E-mail: mayer@eel.ufsc.br.

ISSN 1980-6604/$20.00 © 2008 SBrT/IEEE

This paper proposes a technique named DHDD (Data Hiding for Documents based on Dots). Current printers can print tiny dots hardly noticeable at normal reading distance. We insert tiny dots, pseudo-randomly distributed over the entire document, to embed the message. Our current implementation is able to embed up to 1370 bits in an A4-sized document printed at 600 dpi. As the printing technology evolves, it is expected that future printers will be able to impress even smaller dots, resulting in more visually imperceptible watermarking with more data hiding capacity. We propose to use the entire binary document for embedding the watermark with a high robustness to print-scan, photocopy and geometric attacks

We provide an overall description of the DHDD technique in Section II. Section III describes the embedding process with details while Section IV describes the message extraction approach. Section V provides experiments to validate the proposed technique and Section VI proposes a protocol to be used for authentication of printed documents. Section VII concludes this work.

II. The DHDD Technique

We insert tiny dots, pseudo-randomly distributed over the entire document, to embed the message. In order to extract the message embedded by the proposed technique, the watermarked document is scanned and the tiny dots are detected. If the scanned image were not geometrically distorted, the data extraction would be an easy task. Unfortunately, geometric distortions (small rotation, translation, scale changing, etc.) are inherent to print-scan operations. Thus, we detect and compensate for affine transformations prior to the message extraction. For continuous-tone images, Kutter [16] suggested embedding the same watermark several times at horizontally and vertically shifted locations and to use auto-correlation to detect it. We employ a similar approach to address minor scaling and minor rotation: the document is segmented into four quadrants and the same watermark is embedded into each quadrant. Unfortunately, the auto-correlation approach is unable to withstand translation, cropping or a major rotation (90, 180 or 270 degrees). We propose to detect and compensate for these distortions by inserting some extra registration dots.

The resulting technique is robust to print-scanning, good quality photocopying, geometric attacks, cropping and scribbling/stains on the paper. It can be applied to documents containing characters, drawings, schematics, cartoons, logos, equations and also to documents with large white (or black) areas, as we can embed black dots over white background or vice-versa. Documents containing halftone elements cannot be watermarked by DHDD, because the resulting halftone patterns cannot be distinguished from the inserted tiny dots. However, this is not a practical problem because there are some data-hiding techniques especially designed for printed halftone images, as [21]. A document can be segmented into halftone and other regions, and a specific data-hiding technique can be applied to each kind of region.

DHDD provides a semi-fragile authentication watermarking. DHDD, by employing a perceptual hashing and a public key cryptography, achieves a complete bicolor hardcopy document authentication system. In some situations, the utilization of the proposed system can substitute the use of notarial authenticated photocopies.

III. Data Insertion

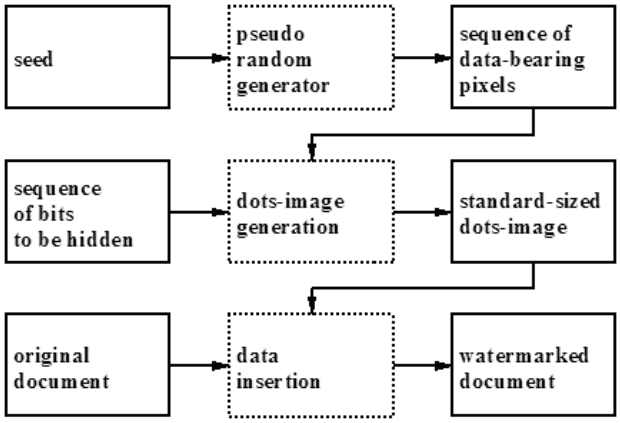


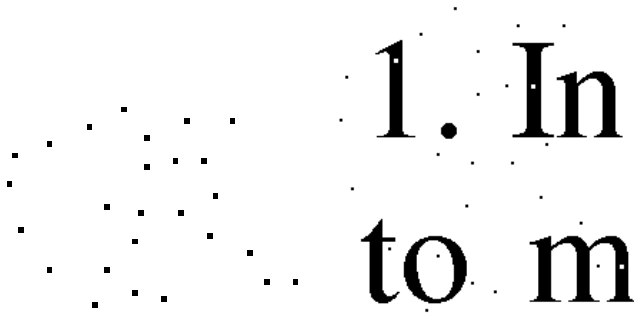
Fig. 1. Data insertion.

Figure 1 depicts the data insertion process. Given the original document, the sequence of bits to be hidden, and a seed of the pseudo-random number generator, the data insertion process consists on obtaining the watermarked document (a binary image) to be impressed by a printer.

In our implementation, sixteen "Presences" or "Absences" of tiny dots in specific data-bearing positions represent each bit: the sequence "PAPAPAPAPAPAPAPA" represents bit 0 and "APAPAPAPAPAPAPAP" represents bit 1. The "dots-image" (Figure 2a) represents a "P" position by a black pixel and an "A" position by white one (an absence cannot be distinguished from the white background). The presence of a tiny dot is translated in document image by a tiny dot, either a black dot in white background or a white dot in black background (Figure 2b). Using more data-bearing pixels per bit, fewer bits can be embedded but the robustness is increased, and vice versa.

The size of the dots-image ($t_w \times t_w$ pixels, 1024×1024 in our implementation) is a parameter known by both the data insertion and extraction algorithms. For simplicity, we are using a square-shaped dots-image. Choosing a dots-image with the same aspect ratio of the document to be watermarked (A4, letter or other) increases the payload. The dots-image is divided in 4 quadrants with $t_2 \times t_2$ pixels each (512×512 in our implementation), and each quadrant receives the same pattern of dots, which allows the data extraction algorithm to detect and compensate for rotation and scaling distortions. Figure 3 depicts a possible configuration of the 16 data-bearing pixels that hides one bit. If pixels 0 and 2 are black (presence of tiny dots) and 1 and 3 are white (absence of tiny dots), the hidden bit is 0. If pixels 0 and 2 are white and 1 and 3 are black, the hidden bit is 1. If the pattern of dots were replicated more than 4 times, the resulting technique would be even more robust, especially against cropping. However, the data hiding capacity would decrease.

The pseudo-random generator chooses a sequence of n_q data-bearing pixels for each quadrant of the dots-image, where $n_q = 6144$ in our implementation. A given distance separates these data-bearing pixels (at least 5 pixels, in our implementation), in order to assure that no pair of tiny dots gets merged during the print-copy-scan operations. Since there are $n_q = 6144$ data-bearing pixels per quadrant in our implementation, a total of $n_q/4 = 1536$ bits can be embedded into the document. However, 16 of these bits are used to indicate the length of the hidden sequence and n_s bits (150 in our case) are used as registration/synchronization dots to detect and compensate for the translation and major rotation. Therefore, the net length of the hidden sequence of bits is $n_l = (n_q/4) - 16 - n_s$, or 1370 bits in our implementation.



(a) Part of the standard sized dots-image.

(b) Part of the watermarked document (before print-scan).

Fig. 2. Some intermediary images of the data insertion.

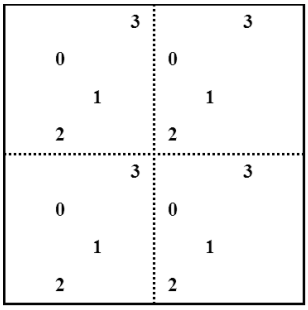
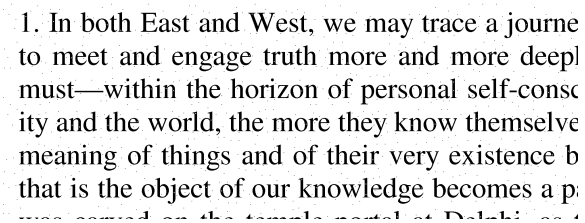


Fig. 3. A dots-image with 16 data-bearing pixels that represent one bit. The image is divided in 4 quadrants and each quadrant receives the same pattern of data-bearing pixels.

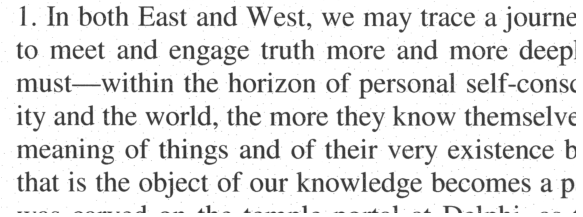
The watermarked document usually has a much larger resolution than the dots-image; for example, an A4 document at 600 dpi has more or less 6600×4400 pixels in the printable area. Thus, the dots-image must be conveniently scaled, before generating the watermarked document. This scaling must keep unchanged the number of black pixels (each black pixel must be mapped into only one black pixel).

The tiny dots can be either black over white background or vice versa. We have observed a practical limit for the size of the small isolated dots. The printers we tested (Oki B4350 and Brother HL-1440) could not print black dots smaller than 2×2 pixels or white dots smaller than 3×3 pixels, at 600 dpi. Thus, each dot in the watermarked document is defined with one of these limiting sizes.

We avoid placing dots near the borders of characters, because the borders of characters may become blurred by the print-scan operations and thus a dot situated at the edge of a character may not be properly detected. So, the data insertion algorithm searches, for each black pixel in the dots-image, the nearest safe location in the document image. In our implementation, a safe pixel in a black document region (where a white dot will be inserted) is situated at least 5 pixels away from the character borders, and a safe pixel in a white region is situated at least 6 pixels away from the border. Sometimes, when no safe location can be found, the dot can be disregarded due to the redundancy of the technique provided by the repetition code and due to the watermark replication on the four quadrants. Figure 4 depicts the appearance of a watermarked document before and after the print-scan operation.



(a) Watermarked binary document before print-scan.



(b) Watermarked grayscale document after print-scan.

Fig. 4. Watermarked documents at approximately normal resolution.

IV. Data Extraction

Figure 5 depicts the data extraction process. Notice that the originally binary documents become grayscale images after the printing and scanning processes and the data extraction algorithm deals with these resulting grayscale images. Given the scanned watermarked document (Figure 6a) and the seed of the pseudo-random generator (the same as in the insertion), the data extraction process consists on recovering the hidden bits.

The data extraction begins by detecting the tiny isolated dots, yielding the "high-resolution dots-image" (Figure 6b). We use a very simple strategy to detect the tiny dots: a pixel p is classified as an isolated tiny dot when its grayscale value is sufficiently brighter or darker than all the 16 neighbor pixels situated at distance 2 from p (using the chessboard distance or 8-connectivity). These detected isolated dots may form small clusters. For each connected component, we classify only its central pixel as a truly isolated dot. The obtained image is resized to yield a standard-sized (but possibly still geometrically distorted) dots-image, in our case with 1024×1024 pixels

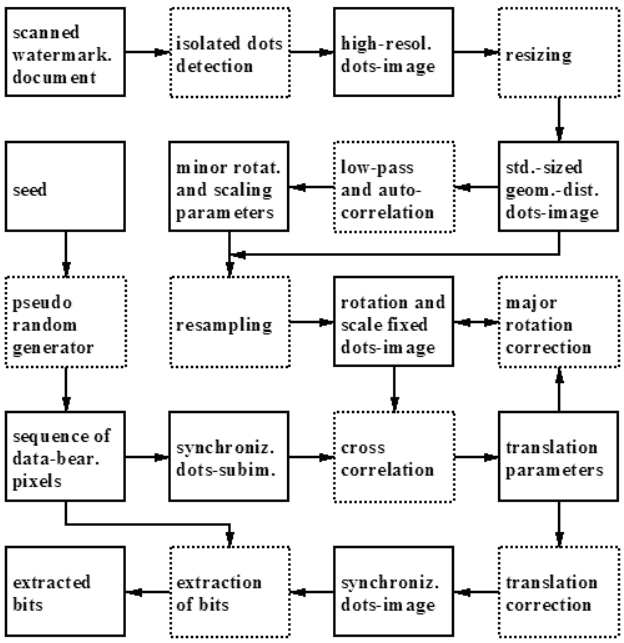


Fig. 5. Data extraction.

(Figure 6c). This resizing must keep unchanged the number of black pixels (each black pixel must be mapped into only one black pixel).

The aim of the following operations is to detect and compensate for the geometrical distortions. The auto-correlation of the standard-sized dots-image will be used to detect the parameters of rotation, scaling and shearing. However, as we know only the positions of black "Present" data-bearing pixels (white "Absent" ones cannot be localized), this image cannot be directly auto-correlated, requiring some preprocessing. First, the margins of the standard-sized dots-image must be detected, in order to avoid their interference in the auto-correlation. These margins appear when the scanned document image extends beyond the printed area of the watermarked document. These margins are characterized by the absence of the tiny dots. We detect the margins using a morphological "closing" operation with a large structuring element (27×27 pixels in our implementation) that merges together the tiny dots. The result of this operation is an image completely white in the margins and completely black in the printed area, which allows us to classify any pixel as belonging to either "margin area" or "printed area." Then, the standard-sized dots-image is low-pass filtered using a Gaussian kernel (with $\sigma = 3$ pixels in our implementation), and the average gray level of the pixels in the printed area is computed. Then, the pixels in the printed area have their values subtracted by the average value, and the pixels in the margin area are set to zero. The resulting image has zero average grayscale value. Figure 6d depicts this image, where the dark, medium and light shades of gray represent respectively negative, zero and positive values. The margins are depicted as regions with constant gray color.

This image is auto-correlated, yielding an image with 9 peaks, because the watermarked document contains 2×2 repeated pattern of tiny dots. Figure 6e depicts the nine peaks of a non-rotated stego image, and Figure 6f depicts the nine peaks of a rotated stego image. The auto-correlation can be computed efficiently using FFT (Fast Fourier Transform) and the correlation theorem. Note that computing the auto-correlation of the "high-resolution dots-image" (with approximately 6600×4400 pixels) would be quite time-consuming, and this is one of the reasons why we resized it to a smaller standard size (1024×1024 pixels). The central correlation peak (zero lag) is simply ignored. The remaining 8 peaks appear because the same watermark was periodically replicated at 4 different locations. Those peaks are much stronger and clearer than in continuous-tone watermarking [16], because in our process there is no interference of the host image. Thus, we obtained a high robustness even using only the four corner peaks to detect rotation, scaling and shearing parameters. We use the locations of these four corner peaks, and the locations where these four peaks should be without the geometric distortions, to perform a bilinear resampling, yielding the "rotation and scale fixed dots-image."

The obtained image is almost a translated version of the original dots-image, but it may be rotated by 90, 180 or 270 degrees. To detect the translation and major rotation parameters, we first generate the "synchronization dots-subimage" with $t_2 \times t_2$ pixels (512×512 in our implementation). All pixels of this image are zeroes except in the $4 \times n_s$ (4×150, in our case) registration data-bearing pixels where the pixel values are -1 (if the data-bearing pixel is black or "Present") or +1 (white or "Absent"). This image is low-pass filtered using a Gaussian kernel (in our case, with $\sigma = 1.125$ pixels). Figure 6g depicts this image, where black, gray and white represent respectively values -1, 0 and +1. This image is cross-correlated with the "rotation and scale fixed dots-image." The resulting image must contain four peaks that form a square with sides of approximate t_2 pixels (512 in our case, Figure 6h), unless the image has undergone a major rotation. In this case, the four peaks will not be detected and the "rotation and scale fixed dots-image" is rotated by 90 degrees (then by 180 and 270 degrees) and cross-correlated again until detecting the four peaks. These trial rotations end up by identifying the major rotation parameter. After identifying the major rotation parameter, we use the positions of the four peaks and where they should be without the translation, to detect and compensate for the translation, yielding the "synchronized dots-image."

Figure 6i depicts in black the "synchronized dots-image," in magenta the expected locations of the data-bearing pixels, and in red where there is a superposition of both. The "synchronized dots-image" is low-pass filtered using a Gaussian kernel (in our case, with $\sigma = 1$ pixel), because the dots may not be positioned exactly at the data-bearing locations. To extract a bit b, the sequence of the 16 data-bearing pixel values corresponding to b (extracted from the filtered synchronized dots-image by accessing the expected positions of the data-bearing pixels) is correlated with the sequence (+1, -1, +1, -1, +1, -1, +1, -1, +1, -1, +1, -1, +1, -1, +1, -1). The signal of the correlation coefficient indicates if the hidden bit is 0 or 1. If the absolute value of this coefficient is too small, the extracted bit may be erroneous.

V. Experimental Results

A. Preliminary Tests

We adjusted the parameters of our implementation primarily to use the Oki Data B4350 as the printing device and the Lexmark X83 as the scanning device, both working at 600 dpi. We have tested also the Brother laser printer HL-1440 and the HP ScanJet 3400C. We have used the Xerox WorkCenter Pro 420 as the photocopy machine.

First, we tested the proposed technique in two documents. An MS-Word document written in "Times New Roman 12 pts" (document A) and a page of an electronic version (PDF) of the IEEE Transactions containing a non-halftone figure and some equations (document B) were converted in two 600 dpi binary images. 1344 bits were embedded in each binary image. Some black data-bearing pixels could not be translated into tiny dots because no safe location for them could be found in the cover images. Specifically, 1.9% of the black data-bearing pixels in document A and 5% in document B could not be transformed in tiny dots.

We tested extracting the hidden data from the watermarked binary documents A and B (without print-scanning them). The results were successful. Then, we printed the both watermarked images using the Oki printer and scanned them with the Lexmark scanner at 600 dpi, and all bits were correctly extracted. Then, we scanned the two documents with slight rotations, and the data were correctly extracted. We scribbled some handwritings on the documents, scanned them, resulting in perfect detection.

To test the resistance to different scalings, the two printed documents were scanned at 500 and 700 dpi, and the extractions were successful. To test the non-uniform scaling, the two images at 700 dpi were downsized to 600×500 dpi and the extractions were successful. To evaluate the robustness to rotations, we rotated the two documents 30 and 135 degrees using bilinear resampling. All the four extractions were successful. We applied cropping at different borders of the documents. We cropped 10%, 20%, 30% and 40% of the documents and all extractions were successful. The extractions failed only when cropping over 40% of the print-scanned documents. We also tested painting parts of the documents with black squares. Painting up to 30% of the document areas resulted in successful extractions. Painting over 40% of the document resulted in detection errors.

We photocopied the printed documents using Xerox photocopier, scanned them, and the bits were correctly extracted. The brightness of the photocopy machine was manually adjusted so that the tiny dots do not disappear in the photocopied documents. It was not possible to preserve both black and white dots, so we adjusted the brightness to preserve only the black dots, because the black dots were far more numerous than the white ones. Only good quality photocopies are able to preserve the hidden data. We stress that the user of our technique must intend to preserve the hidden data (semi-fragile watermarking for authentication), because it is easy to remove them (for example, adjusting the brightness of the photocopy machine so that the tiny dots are removed).

Similar results were obtained using Brother printer and HP scanner. However, documents scanned by the HP are darker than those scanned by the Lexmark and we made a manual brightness correction. We also successfully tested embedding the watermark in eight other A4-sized documents, printing them in Oki Data, scanning in Lexmark and extracting the hidden information.

Typical processing time to insert data to an A4-sized document is 12 seconds and the data extraction takes 110 seconds, using a 3GHz Pentium-4 computer. The programs used in this paper are available at www.lps.usp.br/~hae/software/dhdd.

B. Stirmark Tests

We carried on more exhaustive tests using documents distorted by Stirmark 4.0, release 129 [22, 23]. Stirmark is a popular watermark testing program. We distorted 6 watermarked documents using Stirmark: the original binary documents A1 and B1 (before the print-scan operations); the grayscale images A2 and B2 obtained by print-scanning documents A1 and B1; the grayscale images A3 and B3 obtained by print-photocopy-scanning documents A1 and B1. Each one of these 6 images was distorted in 90 different ways, using the default settings that come with the software. From the original settings, we discarded some tests that do not apply to our case (PSNR, Embed-Time and Self-Similarity). We inserted more parameters in Cropping (80% to 95%). We modified the parameters of convolution-filters, because the original ones were erroneous. The results are depicted in Table 1. In this table, the symbol "-" means errorless extraction; PE-n means partial error with n wrong bits; PE-* means partial error with 10 or more wrong bits; TE means total error; $k \times$ transform means that the "transform" were applied using k different parameters. In this table, we also present the results of Small-Random-Distortions and Latest-Small-Random-Distortions, although DHDD was not designed to resist to these distortions. We can conclude that:

- DHDD is very robust against affine transformations, rotations, rotations followed by small cropping, and rotations followed by small scaling. Each image was distorted in 38 different ways and the extractions were always successful.
- Surprisingly, DHDD seems to be robust against random removal of rows and columns of the image. One in each 10 to 100 rows and columns were removed and all extractions were always successful.
- DHDD is robust against JPEG compression, using 35 or higher quality parameter.
- DHDD is robust against rescaling, from 75% to 110%.
- DHDD is robust against cropping using the original or print-scanned images, keeping 75% or more of the original documents' areas. DHDD is not robust against cropping of photocopied images. This behavior is due to the feeble photocopy quality, because if the photocopy preserved all the tiny dots, the same robustness against cropping would be measured using either the original or the photocopied documents.
- In some cases, DHDD can resist convolution with Gaussian and sharpening kernels.

TABLE I
DATA EXTRACTION FROM IMAGES DISTORTED BY STIRMARK 4.0.

	A1	A2	A3	A4	A5	A6
8 × affine	-	-	-	-	-	-
10 × rotcrop	-	-	-	-	-	-
10 × rotscale	-	-	-	-	-	-
10 × rotation	-	-	-	-	-	-
10 × rem-lines (10 to 100)	-	-	-	-	-	-
jpeg-15	TE	PE-3	PE-*	TE	-	PE-*
jpeg-20	TE	-	PE-2	TE	-	PE-3
jpeg-25	TE	-	PE-1	PE-4	-	PE-3
jpeg-30	TE	-	-	PE-6	-	-
8 × jpeg (35 to 100)	-	-	-	-	-	-
rescaling-50	-	TE	PE-6	-	TE	TE
rescaling-75	-	-	-	-	-	-
rescaling-90	-	-	-	-	-	-
rescaling-110	-	-	-	-	-	-
cropping-25	TE	TE	TE	TE	TE	TE
cropping-50	TE	TE	TE	TE	TE	TE
cropping-75	-	-	PE-2	-	-	PE-*
cropping-80	-	-	PE-1	-	-	PE-*
cropping-85	-	-	-	-	-	PE-6
cropping-90	-	-	-	-	-	PE-5
cropping-95	-	-	-	-	-	-
conv. (3 × 3 Gaussian)	-	-	PE-5	-	-	PE-5
conv. (3 × 3 sharpening)	-	-	-	-	-	-
median-3	TE	PE-6	PE-2	TE	-	PE-2
median-5	TE	TE	TE	TE	TE	TE
median-7	TE	TE	TE	TE	TE	TE
median-9	TE	TE	TE	TE	TE	TE
noise-4	-	TE	TE	-	TE	TE
4 × noise (8 to 20)	TE	TE	TE	TE	TE	TE
rnddist-0.95	PE-2	PE-*	TE	PE-*	PE-*	PE-*
rnddist-1.05	TE	PE-*	TE	PE-*	PE-*	PE-*
rnddist-1.1	PE-3	PE-*	TE	PE-*	PE-*	PE-*
rnddist-1	PE-7	PE-*	TE	PE-*	PE-*	PE-*
latestrnddist-0.95	-	-	-	PE-3	-	-
latestrnddist-1	-	-	-	PE-4	-	-
latestrnddist-1.05	-	PE-2	-	PE-4	-	-
latestrnddist-1.1	-	PE-1	-	PE-6	-	-

- As expected, DHDD is not robust against median filtering (because it removes the tiny dots), and noise (because noise is indistinguishable from the data-bearing tiny dots).

C. Inkjet Tests

Finally, we tested our technique using Epson Stylus CX4900 all-in-one (scanner, photocopier and inkjet printer). We watermarked 8 different A4 documents, printed them in CX4900 using glossy papers, scanned them, extracted the hidden bits, and all extractions were correct. Then, we photocopied the 8 watermarked documents using CX4900 and glossy papers, scanned them, and successfully extracted the hidden bits. Figure 7 depicts some of the original, watermarked, print-scanned and photocopied documents used in our tests.

We took one of the documents, and copied it successively, until obtaining incorrect bit extraction. After 3 successive copies 4 bits were extracted incorrectly. The tiny dots become larger and larger with successive photocopies.

VI. AUTHENTICATED PHOTOCOPY

In this section we describe a protocol to be used with DHDD for authentication of printed and scanned binary documents. Semi-fragile authentication watermarking is used to verify the originator of the image and to certify that its visual content has not been changed maliciously or accidentally. However, semi-fragile watermarking must not detect harmless alterations of the image (such as those generated by lossy compression, brightness/contrast adjusting, etc.) as image adulterations.

A semi-fragile watermarking typically uses a perceptual image hashing (also called robust visual hashing or media hashing [24, 25, 26, 27]). The perceptual hashing $h(A)$ is a value that identifies the image A. Moreover, given two images A and B, the distance D between the hashings $D[h(A), h(B)]$ must be somehow proportional to the perceptual visual difference of the images A and B. However, for the semi-fragile authentication watermarking, it is enough that the function D assumes one of the two possible values: 0 if A and B are visually equivalent and 1 if they are not equivalent. If the documents to be watermarked contain only characters, an OCR followed by a one-way cryptographic hashing can be used as a Boolean perceptual hashing function.

Let us suppose that Alice wants to authenticate a document A. She computes the hashing $h(A)$ and encodes it using her private-key, yielding the digital signature $DS(A)$. She embeds $DS(A)$ into the document using DHDD. She prints the watermarked document and sends it to Bob.

Bob receives the printed document and scans it, resulting in scanned image B. He extracts the hidden digital signature $DS(A)$ from the scanned image B using DHDD and decodes it using the Alice's public-key, obtaining the extracted hashing value $h(A)$. If the data extraction is unsuccessful, the document is deemed not authentic, because it has undergone distortions strong enough to hinder extracting the hidden bits. Bob also filters out the data-bearing tiny dots from the scanned document B, obtaining the filtered document $\hat{B}$. Many different filters can be used for this purpose, for example, the median filter. Bob computes the perceptual hashing $D(\hat{B})$ of the filtered document. Only if $D[h(A), h(\hat{B})] = 0$, the printed document is considered authentic.

Bob makes a good quality photocopy of the printed document. Bob must perform this operation carefully, in order to preserve the tiny dots. Bob send to Carol the photocopy of the document. Carol can verify the authenticity of the photocopy (that Alice generated the document and it has not been modified) by the same means.

VII. CONCLUSIONS

We have presented a data hiding technique for printed binary documents. Tiny barely visible dots are used both to recover synchronism and to carry the information. A sequence of precise image processing operations is proposed to achieve the decoding of the information. The current implementation of the proposed approach is able to robustly embed up to 1370 bits in an A4 sized document printed at 600 dpi. In order to evaluate the robustness of the technique, we used the popular Stirmark benchmarking to generate many types of attacks. Experiments illustrate the method's robustness to printing-scanning operations, good quality photocopying, affine transformations, cropping and scribbling/stains on the paper. We have described its application to semi-fragile authentication of printed documents.

REFERENCES

[1] N.F. Maxemchuk, and S. Low, "Marking Text Documents," *IEEE Int. Conf. Image Processing*, vol. 3, pp. 13-17, 1997.

[2] P.V.K. Borges, J. Mayer, "Text Luminance Modulation for Hardcopy Watermarking," *Signal Processing*, vol. 87 , no. 7, pp. 1754-1771, 2007.

[3] M.S. Fu, and O.C. Au, "Data Hiding by Smart Pair Toggling for Halftone Images," *IEEE Int. Conf. Acoustics Speech and Signal Processing*, vol. 4, pp. 2318-2321, 2000.

[4] H.Y. Kim, and A. Afif, "Secure Authentication Watermarking for Halftone and Binary Images," *Int. J. Imaging Systems and Technology*, vol. 14, no. 4, pp. 147-152, 2004.

[5] H.Y. Kim, "A New Public-Key Authentication Watermarking for Binary Document Images Resistant to Parity Attacks," *IEEE Int. Conf. on Image Processing*, vol. 2, pp. 1074-1077, 2005.

[6] Q. Mei, E.K. Wong, and N. Memon, "Data Hiding in Binary Text Documents," *Proceedings of SPIE*, vol. 4314, pp. 369-375, August 2001.

[7] M. Wu and B. Liu, "Data Hiding in Binary Image for Authentication and Annotation," *IEEE Trans. on Multimedia*, vol. 6, no. 4, pp. 528-538, Aug. 2004.

[8] H.Y. Kim and R.L. de Queiroz, "Alteration-Locating Authentication Watermarking for Binary Images," *Int. Workshop on Digital Watermarking*, Lecture Notes in Computer Science 3304, pp. 125-136, 2004.

[9] Y.C. Tseng, Y.Y. Chen, and H.K. Pan, "A Secure Data Hiding Scheme for Binary Images," *IEEE Trans. on Communications*, vol. 50, no. 8, pp.1227-1231, Aug. 2002.

[10] C.-C. Chang, C.-S. Tseng, and C.-C. Lin, "Hiding Data in Binary Images," *Lecture Notes in Computer Science* 3439, pp. 338-349, 2005.

[11] J.T. Brassil, S. Low, N.F. Maxemchuk, "Copyright Protection for the Electronic Distribution of Text Documents," *Proc. of IEEE*, vol. 87, no. 7, pp. 1181-1196, July 1999.

[12] D. Huang and H. Yan, "Interword Distance Changes Represented by Sine Waves for Watermarking Text Images," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 11, no. 12, pp 1237-1245, Dec. 2001.

[13] R. Vllan, S. Voloshynovskiy, O. Koval, J. Vila, E. Topak, F. Deguillaume, Y. Rytsar and T. Pun, "Text data-hiding for digital and printed documents: theoretical and practical considerations," in *Proc. of SPIE*, Elect. Imaging, USA, 2006.

[14] R. Vllan, S. Voloshynovskiy, O. Koyal and T. Pun, "Multilevel 2D Bar Codes: Towards High Capacity Storage Modules for Multimedia Security and Management," in *Proc. of SPIE*, Elect. Imaging, USA, 2005.

[15] N.D.Quintela and F. Prez-Gonzlez, "Visible Encryption: Using Paper as a Secure Channel," in *Proc. of SPIE*, Elect. Imaging, USA, 2003.

[16] M. Kutter, "Watermarking Resisting to Translation, Rotation, and Scaling," *SPIE Conf. Multimedia Syst. and App.*, vol. 3528, pp. 423-431, Nov. 1998.

[17] C.Y. Lin, M. Wu, J.A. Bloom, I.J. Cox, M.L. Miller and Y.M. Lui, "Rotation, Scale, and Translation Resilient Watermarking for Images," *IEEE T. Image Processing*, vol. 10, no. 5, pp. 767-782, May 2001.

[18] S. Pereira, J.J.K.O. Ruanaidh, F. Deguillaume, G. Csurka, and T. Pun, "Template Based Recovery of Fourier-Based Watermarks Using Log-Polar and Log-Log Maps," *IEEE Int. Conf. Multimedia Comp. Systems*, vol. 1, pp. 870-874, Jun. 1999.

[19] K. Solanki, U. Madhow, B. S. Manjunath, S. Chandrasekaran, and I. El-Khalil, "Print and Scan Resilient Data Hiding in Images," *IEEE T. Information Forensics and Security*, vol. 1, no. 4, pp. 464-478, Dec. 2006.

[20] C.-Y. Lin, "Public Watermarking Surviving General Scaling and Cropping: An Application for Print-and-Scan Process," *Multimedia and Security Workshop at ACM Multimedia*, Orlando, FL, Oct 1999.

[21] I. G. Chun, S. H. Ha, "A Robust Printed Image Watermarking Based on Iterative Halftone Method," *Int. Workshop on Digital Watermarking*, Lecture Notes in Computer Science 2939, pp. 200-211, 2003.

[22] F.A.P. Petitcolas, "Watermarking Schemes Evaluation," *IEEE Signal Processing*, vol. 17, no. 5, pp. 58-64, September 2000. http://www.cl.cam.ac.uk/ fapp2/publications/ieeespm00-evaluation.doc

[23] F.A.P. Petitcolas, M. Steinebachb, F. Raynalc, J. Dittmannb, C. Fontained, N. Fats, "A Public Automated Web-Based Evaluation Service for Watermarking Schemes: StirMark Benchmark," in *Proc. Electronic Imaging, Security and Watermarking of Multimedia Contents*, vol. 4314, San Jose, CA, January 2001. http://www.cl.cam.ac.uk/ fapp2/publications/ei01-automated.doc

[24] M. Schneider and S.-F. Chang, "A Robust Content Based Digital Signature for Image Authentication," *IEEE Int. Conf. Image Processing*, vol. 3, pp. 227-230, Sep. 1996.

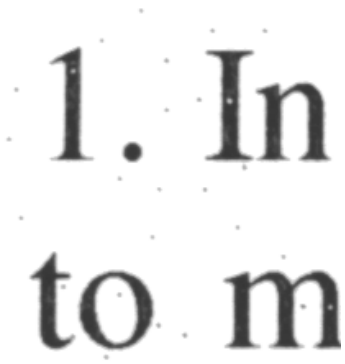
[25] C.-S. Lu, C.-Y. Hsu, "Geometric Distortion-Resilient Image Hashing Scheme and Its Applications on Copy Detection and Authentication," *Multimedia Systems*, vol. 11, no. 2, pp. 159-173, 2005.

[26] M. K. Mihcak and R. Venkatesan, "A tool for robust audio information hiding: a perceptual audio hashing algorithm," in *Proc. 4th Information Hiding Workshop'01*, Portland, OR, USA, April 2001.

[27] R. Venkatesan, S.-M. Koon, M. H. Jakubowski, and Pierre Moulin, "Robust Image Hashing," in *Proc. IEEE Int. Conf. Image Processing*, 2000.

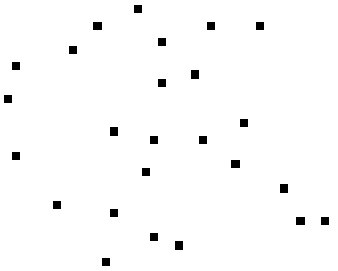
Hae Yong Kim was born in Korea in 1964 and migrated to Brazil in 1975. He received the best rating in the university ingressing examination for Computer Science, Universidade de São Paulo (USP), Brazil, and graduated with the best average marks in 1988. He received MSc in Applied Mathematics (1992) and PhD in Electrical Engineering (1997) from USP. Since 1989 he has been teaching in USP, and currently he is an associate professor with the Dept. of Electronic Systems Engineering, USP. Since 2002, CNPq (National Council for Scientific and Technological Development) has granted him a "research productivity award" scholarship. His research interests include the general area of image and video processing and analysis, authentication watermarking, and machine learning.

Joceli Mayer graduated in Electrical Engineering from the Universidade Federal de Santa Catarina (UFSC) in 1998, received the Masters in Electrical Engineering degree from UFSC in 1991, received the Masters in Computer Engineering degree from the University Of California (UCSC) in 1998 and received the Doctor of Philosophy degree in 1999 from UCSC. Currently he is an Associate Professor of Electrical Engineering at UFSC since 1993 and teaches at the undergraduate and graduate programs. He has published over 70 articles in scientific conferences and periodicals. He is coordinating projects on super-resolution, speech compression, image watermarking, and hardcopy document authentication with support of the industry and government agencies.

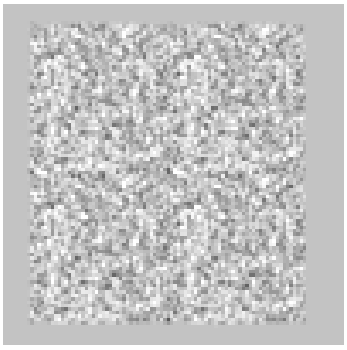


(a) Scanned image.

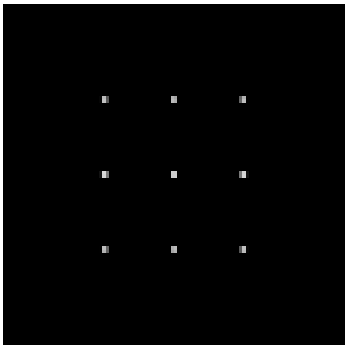
(b) High-resolution dots-image.



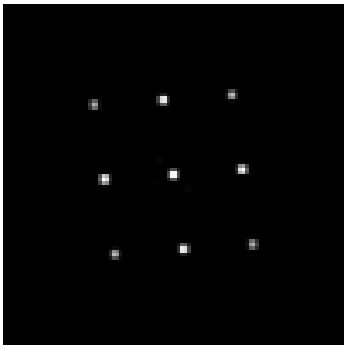
(c) Standard-sized geometrically distorted dots-image.



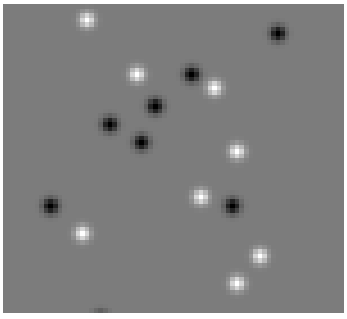
(d) Low-passed dots-image with detected margins.



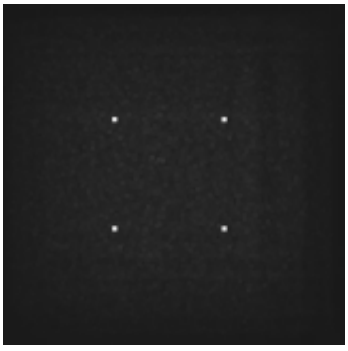
(e) The nine peaks of the autocorrelation image.



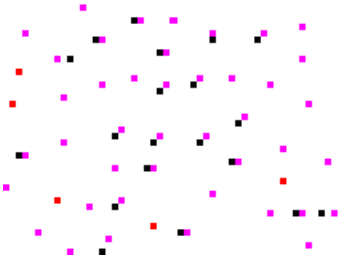
(f) The nine peaks of the autocorrelation obtained from a rotated stego image.



(g) Low-pass filtered synchronization dots-subimage.



(h) The four peaks of the cross-correlation.



(i) The synchronization dots-image (black), data-bearing pixels (magenta), and the superposition of both (red).

Fig. 6. Various intermediary images of the data extraction.

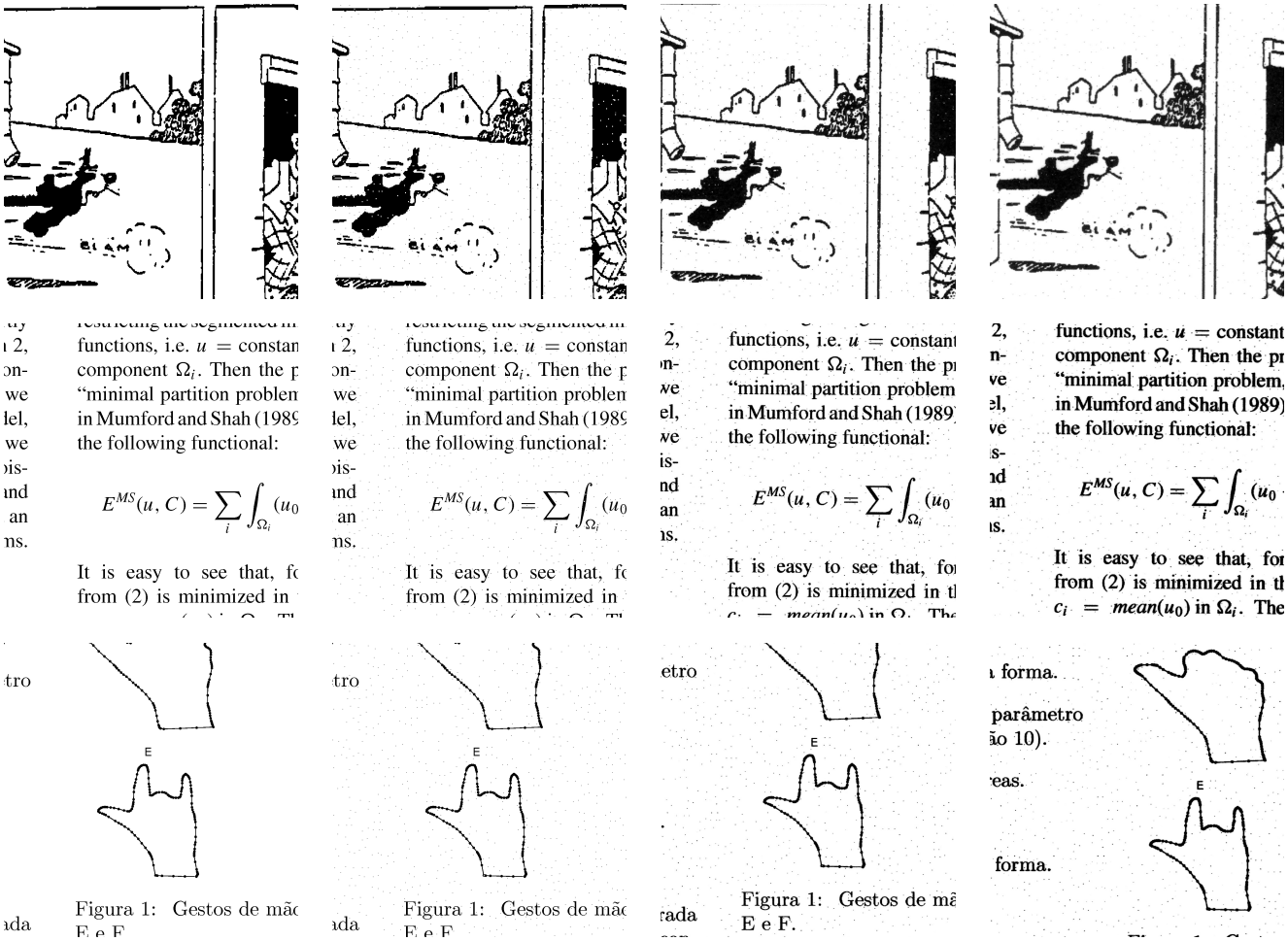


Fig. 7. Some parts of the documents used to test the proposed technique using inkjet printers: 1st column depicts the original binary documents; 2nd column depicts the watermarked binary documents; 3rd column depicts the print-scanned grayscale documents; 4th column depicts the print-photocopy-scanned grayscale documents. The hidden data were successfully extracted from the images in 2nd, 3rd and 4th columns.