Watermarking and Copyright Labeling of Printed Images

Hagit Z. Hel-Or
Dept. of Computer Science
Haifa University
Haifa 31905, ISRAEL
hagit@cs.haifa.ac.il

Abstract

Digital Watermarking is a labeling technique for digital images which embeds a code into the digital data so the data is marked. Watermarking techniques previously developed deal with on-line digital data. These techniques have been developed to withstand digital attacks such as image processing, image compression and geometric transformations. However, one must also consider the readily available attack of printing and scanning. The available watermarking techniques are not reliable under printing and scanning. In fact one must consider the availability of watermarks for printed images as well as for digital images. An important issue is to intercept and prevent forgery in printed material such as currency notes, bank checks etc and to track and validate sensitive and secret printed material. Watermarking in such printed material can be used not only for verification of ownership but as an indicator of date and type of transaction or date and source of the printed data. In this work we propose a method of embedding watermarks in printed images by inherently, taking advantage of the printing process. The method is visually unobtrusive to the printed image, the watermark is easily extracted and is robust under reconstruction errors. The decoding algorithm is automatic given the watermarked image.

Keywords: Watermarking, Copyright Labeling, Printing, Halftoning, Dithering

1 Introduction

With the development of technology and communication, the availability of digital data has become increasingly accessible. With this accessibility, illegal acquisition reproduction and distribution of digital data (digital images such as art, satellite etc, special fonts, and confidential images or video sequences), has become a problem. The need has evolved for copyright protection; ensuring ownership, detecting corrupted or forged images and allowing tracking of digital data.

The method of *Digital Watermarking* is a labeling technique for digital images which embeds a code into the digital data so the data is marked. The embedded data (known as the *Watermark*) is a code: either specific per image or general for all images. It might be a secret code requiring a key to extract. The hidden code might be a sequence of binary bits, a binary image, a grayscale image or a text sequence (note that all these codes can be, and usually are, translated into a sequence of binary bits). The embedded data usually encodes ownership and additional private information pertaining to the digital data in which it is embedded.

The Watermark embedded in an image, may be visible, in which case it is actually an image pasted onto the base image. This type of watermarking is the source of the name "watermark". It has been used forever by artists who mark their art against forgery (typically, the Watermark is the artist's name, but it might be a symbol, as in Michelangelo's work or even a small icon as the goat in Shagal's work, or Hitchcock appearing discreetly in his own movies). An example of such watermarking in modern digital imaging is the television station symbol which appears in a corner of the broadcasted picture. Some clever visually detectable watermarks have been suggested where the watermark is seen but appears transparent and the base image is unobstructed [5]

Typically, however, the watermark is required to be visually imperceptible and unobtrusive to the digital data in which it is embedded. Although invisible, It should be easily extractable using digital and computerized methods. For copyright and ownership protection, the watermark must be difficult to remove.

Several watermark techniques have been previously presented and are reviewed in Section 2. The digital Watermarks that have been presented were developed to withstand manipulations (*attacks*) on the watermarked images including:

- Geometric transformations such as rotation, translation and scaling.
- Geometric operations such as cropping.
- Image processing manipulations such as blurring and sharpening.
- Color balance and contrast enhancement.
- Compression and requantization of the watermarked image.
- Forgery such as combining or fusing several watermarked images (obtaining a new valid watermarked image or an unmarked version of the original image).

One important class of attacks not thoroughly considered in previous studies on watermarking, is the printing and rescanning of watermarked images. These technologies are readily available and accessible yet have not been specifically considered in the development of existing watermarking techniques. The printing and scanning processes involve complex non-linear operations on the data, which are not easily modeled. Resampling, quantization, and dithering only partially describe the process, which also includes gamut mapping, gamma correction and other spatial effects due to the printing process and later the scanning process. It is imperative that the existing watermarking techniques be tested under printing and scanning attacks before being considered as secure and dependable copyright labeling techniques.

In previous work, only digital data have been considered as warranting watermarking and copyright labeling. However, one must consider printed matter as well. An important issue is to intercept and prevent forgery in printed material such as currency notes, bank checks etc. Also of importance is to track and validate sensitive or secret printed material. Watermarking in such printed material can be used not only for verification of ownership but as an indicator of date and type of transaction, of or date and source of the printed data etc. Watermarking of printed material complements watermarking of digital data. Together with encryption and watermarking of digital data, the watermarking of printed material should form a copyright labeling package for multimedia data.

In this work we study the area of watermarking in a new and important media, namely watermarking and copyright labeling of printed visual data.

As in digital images, the embedded watermark in printed images, may be a sequence of binary bits, a binary image, a grayscale image or a text sequence (again, all these codes can be translated into a sequence of binary bits). The code is either specific per image or general for all images. Possibly the code is similar for sets of printed data (such as for series of checks or bank notes). The code is typically a secret code requiring a key to extract.

Similar to watermarking of digital data, the embedded data may be visible (as in the partially visible imprints which are typical in currency notes). However, we are interested in watermarks that are perceptually insignificant in that they do not obstruct or interfere with the printed image. In contrast with watermarking of digital images, the attacks described in Section 1, which are inherently digital, are irrelevant to printed material. The goal of watermarking of printed images requires the following watermark characteristics:

- The watermark should be perceptually insignificant.
- The watermark should be easily extractable, using either mechanical devices such as physical filters, mirrors, etc, or using digital scanning followed by digital and automatic extracting algorithms.
- The watermarking technique should be hard to forge and reproduce by adversaries and hostile agents.
- Possibly, the watermark should be unavailable for decoding without a proper secret key.

In this paper, a method is proposed to embed watermarks in printed images by inherently exploiting the printing process itself. The watermark is not visibly detectable and is easily reconstructed.

2 Existing Watermarking Techniques

Several approaches to watermarking of images has been suggested. These can be divided into two main approaches: spatial embedding and frequency embedding of watermarks.

2.1 Spatial Embedding of Watermarks

The most straight forward and basic technique for watermarking is to embed a binary sequence by changing the least significant bit (LSB) of some image pixels [9, 32, 24, 35]. These methods are highly sensitive to attacks specifically to compression and to attacks inherently involving low pass filtering and quantization, which destroy the LSB of the data. The decoding of the watermark is performed using cross correlation techniques requiring both the original image and the watermark.

Other techniques were suggested that alter the value of image pixels rather than only the LSB. In [15] a binary sequence is embedded in the blue channel of randomly chosen pixels of the original image. This technique is sensitive to noise especially to compression (JPEG) which greatly deteriorates the blue channel. It is questionable whether this watermark is robust under geometric transformations which tend to intermix pixel values. In [19], a method was suggested in which pixels in patterned regions (textures) are modified. In this manner the code is embedded in high frequency regions of the image.

To increase robustness of the watermark, neighborhoods of pixels (patches) rather than single pixels are modified in an image. In [6], square blocks of the image are increased or decreased in brightness according to the encoded bit. This technique is not robust under collusion (extracting the original image by combining several watermarked versions of the image). To increase robustness, the author randomizes the positions and sizes of the blocks.

A different approach to spatially embedding watermarks is to consider embedding more than a single bit of the code at a single pixel. In [21] an N-bit code is embedded by adding N binary pseudo random patterns to the image. Decoding of the watermark is performed by subtracting the original image and then correlating with each of the N pseudo patterns. To increase robustness of the watermark, the author proposes to filter the N pseudo random patterns so that the patterns are within the frequency bands that survive JPEG compression. In [2] the watermarking technique modifies N pairs of pixels by incrementing the value of one of the pair and decrementing the other. Verifying the existence of the watermark is performed by evaluating the sum of differences between the pairs. The authors modify their technique to increase robustness (under compression for example) by choosing and modifying pairs of pixel patches in the image rather than pairs of single pixels. It was commented ([16]) that this technique has a low bit capacity rate.

In [20] the embedded watermark is a binary array the size of the image in which it is embedded. The watermark is restricted to have an equal number of '0' and '1' entries. The embedding is performed by changing the image pixels corresponding to the '1' entries in the watermark. These image pixels are changed by increasing their value by a positive integer factor which is dependent on the variance of the pixel values. Statistical hypotheses testing is used to determine whether a given watermark is indeed embedded in a watermarked image. It was commented ([16]) that this technique also has a low bit capacity rate and that it is resistance to JPEG compression only up to a quality factor of 90% (4:1 compression). This technique is extended in [16] where for every bit in the code, a random set of pixels within a randomly selected image block is increased or decreased. To ensure resistance of the watermark under JPEG compression the embedding is verified in a low quality version of the image block (obtained by JPEG compressing the block at a given quality factor and then uncompressing it).

2.2 Frequency Embedding of Watermarks

Spatially embedded watermarks are seldom resistant to cropping and are not stable under compression. Additionally, using spatial embedding, the capacity (the number of encoded bits) is relatively small - it has been shown that for a 256x256 image, only about 100 bits can be stored efficiently [23]. To overcome these drawbacks, a different approach to watermarking has been suggested in which the embedding is performed in the frequency domain. The frequency embedded watermark is in a sense spread over a wide spatial region in the image. This supports robustness of the watermark under cropping and geometrical attacks. Embedding in the frequency domain can be constrained appropriately to ensure stability under compression (especially under compression techniques such as JPEG which are based on transformation of the image to the frequency domain). Another advantage of frequency embedding is that there is typically a larger capacity, i.e. more bits of code can be embedded than in the spatial embedding techniques.

Frequency embedding of watermarks usually takes the form of transforming the image (or it's sublocks) to the frequency domain, embedding the watermark code in the coefficients of the transform, then inversely transforming the watermarked coefficients to obtain the watermarked image. Initially it was assumed that since the watermark should be invisible, the frequency embedding of the code should be in the high frequency coefficients to which the human visual system (HVS) is less sensitive. However the fact that the HVS is less sensitive to high frequencies is what guides the frequency based compression techniques such as JPEG. They attenuate, if not eliminate, the high frequencies of the image. Thus, if a watermark is to survive compression, it must not be embedded in the high frequencies, rather the code should be hidden in the perceptually significant portions of the image, namely in the mid to low frequency range.

In [3, 10], a binary code is embedded by increasing or decreasing the mid-frequency coefficients of the Discrete Cosine Transform (DCT) of image blocks. In [3] other transforms such as the Walsh-Hadamard, Wavelet and the Discrete Fourier Transform (DFT) are also used. These techniques require the original image to decode the watermark. In [29] the strength of the embedded watermark is adaptively changed according to the sensitivity to perceptual distortion in the embedded region. The N most significant coefficients of the DCT of an image block are increased or decreased according to the watermark bit to be embedded. The amount of change is dependent on the noise sensitivity of the image block. Decoding of the watermark requires both the watermark and the original image.

In [22, 23], a binary sequence is embedded in the most significant portions of the DFT coefficients by modifying the phase component of the coefficients. The phase is perceptually more significant than the magnitude of the DFT, thus the embedded code is more robust to attacks such as compression and change in image contrast. Decoding is based on correlation techniques and is independent of the original image. The authors add an adaptive component to their technique in which the number of embedded bits per block is adaptively chosen according to the local variance in the image. Thus textured and busy areas in the image encode more bits and have a higher information density.

A different approach to watermarking in the frequency domain involves spectrum shaping. In [8, 7], the spectrum of the watermarked image is shaped to be similar to the original image. This is done by non-linearly modifying the most significant portions of the image DCT coefficients. In [25] the code to be embedded is modified so that it's spectrum is appropriate for compressions such as JPEG and in [27] a a pseudo random N-sequence is shaped to conform to the characteristics of the HVS, then linearly added to the DCT coefficients of the image. The technique proposed in [8, 7] uses Gaussian distributed, real-valued sequences as watermarks, which according to the authors is more robust to collusion than binary watermarks.

A different class of frequency embedded watermarks impose constraints on sets of DCT coefficients rather than individually modifying them. In [36, 14, 13], pairs and triplets of the DCT coefficients of randomly selected blocks of an image are modified so that the rank ordering of the coefficients determine the binary bit of the hidden code. The mid-frequency range of the coefficients were chosen and quantization errors were considered to ensure that the embedded code is robust under JPEG compression. Indeed, the watermark resists JPEG compression up to a quality factor of 50%. In [4], linear constraints are enforced between pairs of mid-frequency DCT coefficients of randomly chosen blocks. Statistical testing is used to determine whether a given watermark is indeed embedded in a watermarked image.

2.3 Watermarks in Dithered Images

Very little work has been presented on watermarking of printed images. Concurrent with this work, a similar approach to embedding watermarks in dithered images is described in [1]. In [17, 28] a method is proposed for embedding a watermark in a dithered image which is the typical representation of a printed image. The authors also suggested watermarking techniques for predictive coding and run-length coding which are used to represent fax data. Their methods are based on embedding the watermark by having it resemble quantization noise. In [23] it is commented that this method can be viewed as perceptually adaptive in that the quantization errors and thus the embedded code, are concentrated at the edges and in textured regions of the image. Recently, a method for embedding watermarks in printed images was suggested [12, 34], using correlated stochastic dither cells. In this approach, the watermark is embedded by increasing correlation between dither cells in specified regions of the cell. In this paper, a novel method is presented that embeds the watermark in the dithering threshold during the halftoning process.

3 Watermarking using the Monotone Printing Process

A method is proposed to embed a watermark in printed images by taking advantage of the printing process. In general, printing of color or gray scale images with a finite number of inks is performed by distributing ink dots in different densities and patterns throughout the print. A correct pattern of ink dots reproduces the original colors of the image. There are many possible ink patterns that produce printed images that appear similar to the human eye. This characteristic is exploited to embed a watermark in printed images. In the proposed method, the embedded code locally varies the pattern of ink dots such that correct distribution is still maintained. The variation in ink pattern is highly constrained to allow high probability of correct decoding of the watermark, and to reduce visibility of texture artifacts that may arise.

3.1 Encoding

The printing of digital images involves translating true color values available in the stored image, to colors and patterns available at the printer. Typically, printers have a limited selection of inks (1-6) with which the image is to be reproduced on paper. *Halftoning* is a method for reproducing colors of the original image by a pattern and combination of available printer inks. There are various approaches to Halftoning. One of the basic approaches is the class of *Dithering* methods. The Dithering methods are based on thresholding of the image pixel values according to a predefined thresholding pattern. Classically, the dithering pattern is determined by a collection of threshold values in a square or rectangular block of pixels. This block of threshold values is called the Dither Cell. This cell pattern is then duplicated creating a grid of cells which tessellates the original image and is used as the thresholding pattern of the image (see Figure 1). A number of dither patterns have been introduced having various characteristics in terms of the halftoning patterns they produce. Several well known dither cell patterns are the Void and Cluster, Bayer, and Cluster Dot patterns (see [26, 30] for a review).

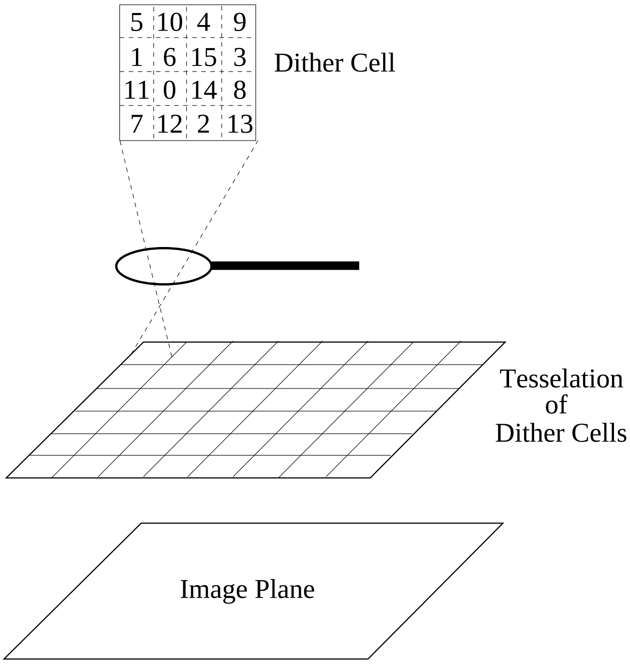


Figure 1: A dither cell pattern is duplicated creating a grid of cells which tessellates the original image and is used as the thresholding pattern of the image to create a halftoned image for printing.

We propose a method for copyright labeling of printed images which embeds a watermark by using a number of different dither cells to create a threshold pattern in the halftoning process. The pattern of the dither cells used for dithering is determined by the watermark. A schematic diagram

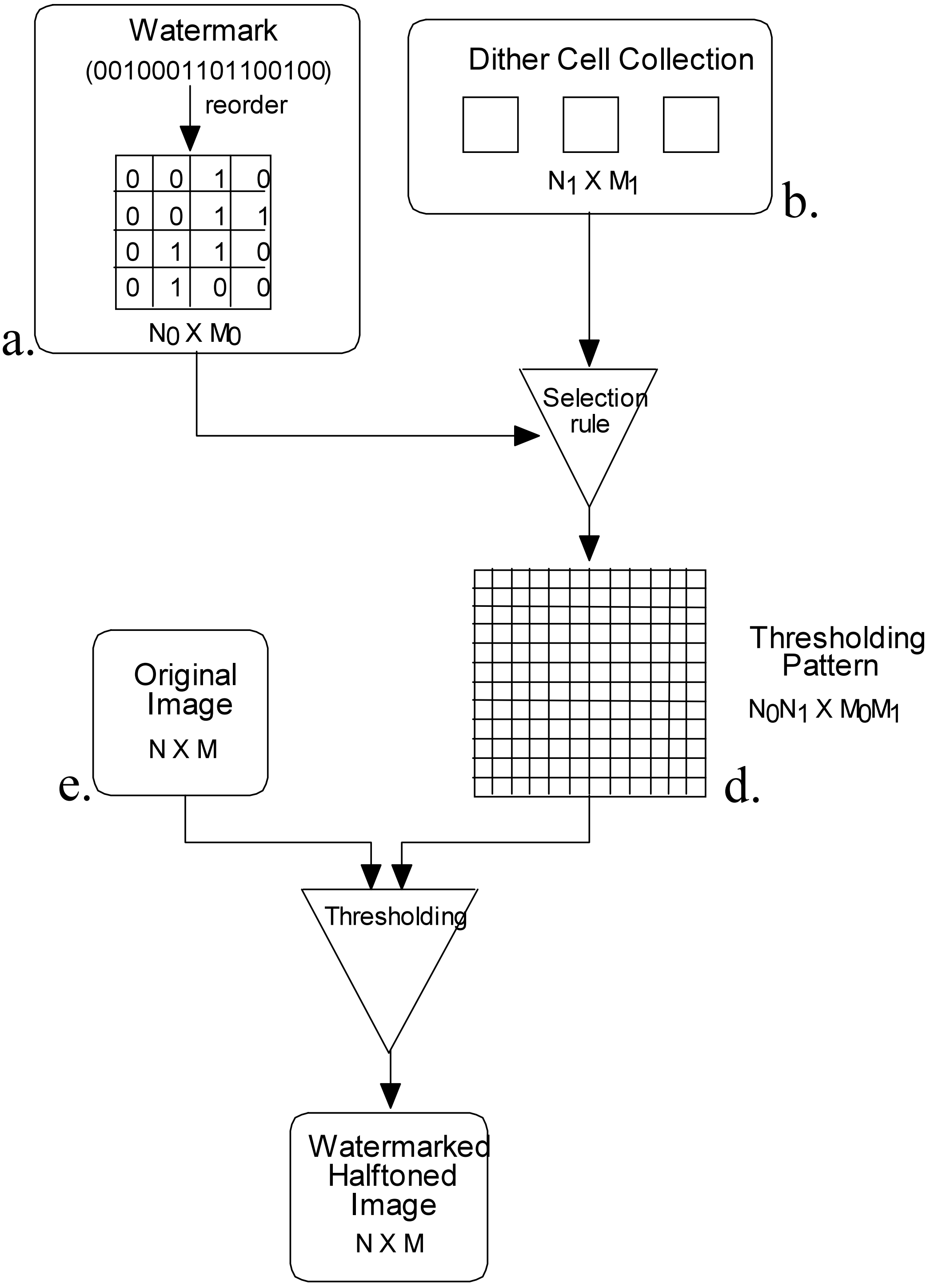
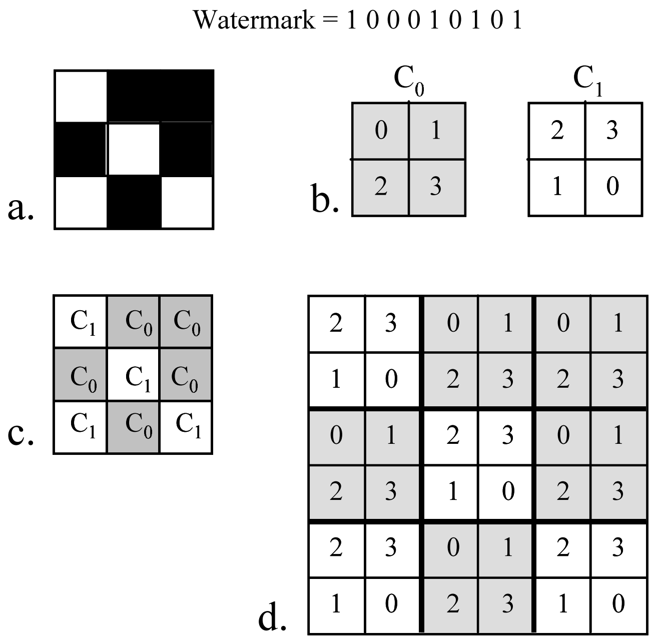


Figure 2: Schematic diagram of the watermarking technique for printed images.



C_0	
0	1
2	3

C_1	
2	3
1	0

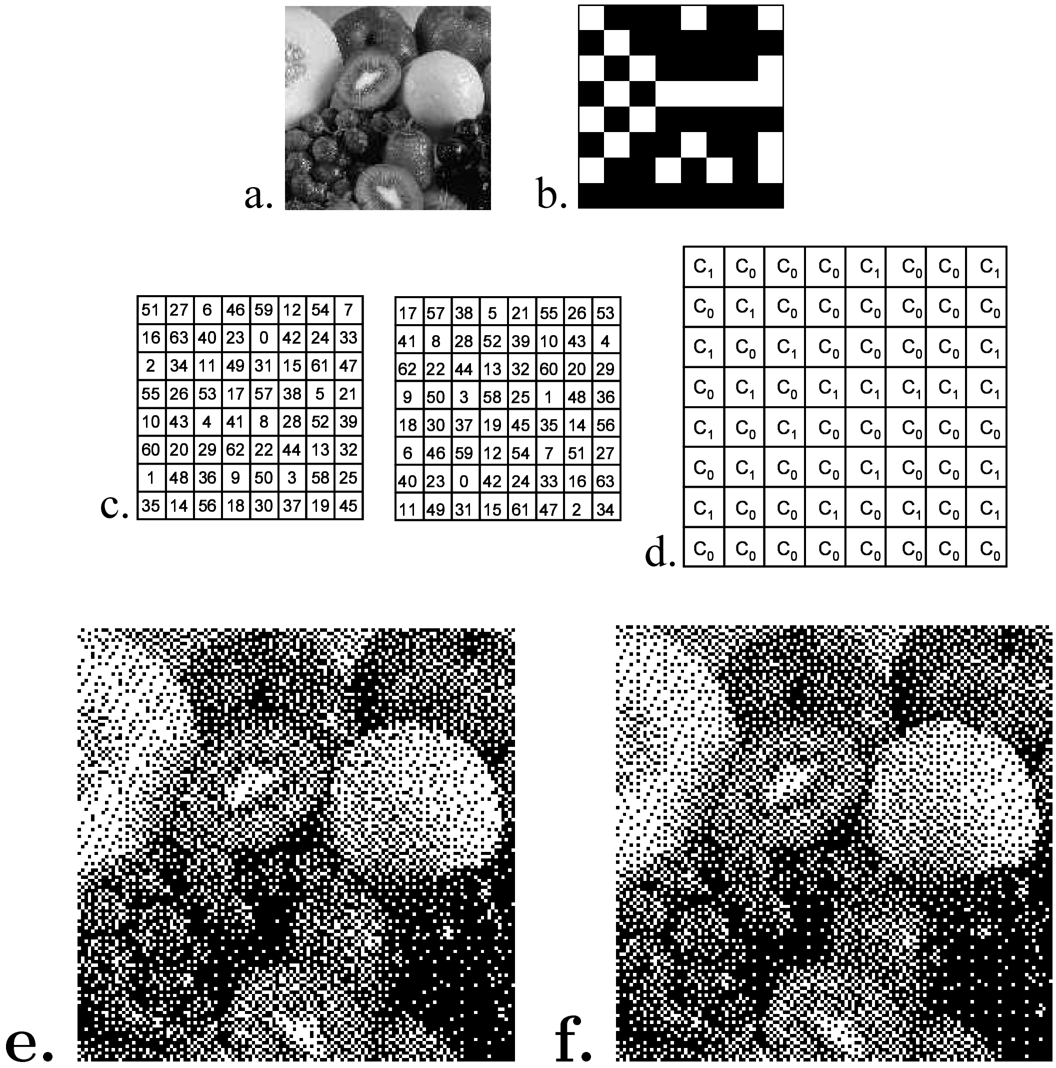
C_1	C_0	C_0
C_0	C_1	C_0
C_1	C_0	C_1

2	3	0	1	0	1
1	0	2	3	2	3
0	1	2	3	0	1
2	3	1	0	2	3
2	3	0	1	2	3
1	0	2	3	1	0

Figure 3: The watermark encoding scheme
a) The 9-bit watermark (100010101) is reordered as a 3×3 array. Black pixels represent '0', white pixels represent '1'.
b) Two 2×2 dither cells C_0 and C_1 are available for dithering. The threshold values, chosen for demonstration, are in 0..3.
c) A single dither cell is selected for each entry in the watermark array. Cell C_0 is chosen for bit '0' and cell C_1 is chosen for bit '1'.
d) The selected dither cells are tiled in correspondence to the watermark array to form a thresholding pattern of size 6×6.

of the proposed method is shown in Figure 2. A schematic example is shown in Figure 3. The watermark is a binary sequence reordered into an $N_0 \times M_0$ array (Figures 2a and 3a). This array is used to select dither cells from a set of predefined dither cells of size $N_1 \times M_1$ (Figures 2b and 3b). A single dither cell is selected for each entry in the watermark array (Figure 3c). The selected dither cells are tiled in correspondence to the watermark array to form a thresholding pattern of size $N_0N_1 \times M_0M_1$ (Figures 2d and 3d). This pattern is repetitively used to tile the original image $(N \times M)$ and serve as the threshold pattern with which the image (Figure 2e) is halftoned.

Figure 4 shows an example where an original 128×128 monochrome image (Figure 4a) is watermarked with a binary sequence of 64 bits. Figure 4b shows the watermark reordered as an 8×8 array (black and white pixels denote '0' and '1' respectively). The set of available dither cells consists of the two 8×8 arrays shown in Figure 4c and denoted C_0 and C_1 (threshold values have been normalized to the range 0..63 for convenience). A single dither cell is selected for each entry in the watermark. The rule for selection of the dither cell is to select C_0 for bit '0' and C_1 for bit '1'. The selected dither cells are tiled in correspondence to the watermark array to form a thresholding pattern of size 64×64, shown in Figure 4d. This pattern is tiled in the image plane (2×2) to produce the threshold values that are used to halftone the original 128×128 image. The watermarked halftoned image is shown in Figure 6a.



51	27	6	46	59	12	54	7
16	63	40	23	0	42	24	33
2	34	11	49	31	15	61	47
55	26	53	17	57	38	5	21
10	43	4	41	8	28	52	39
60	20	29	62	22	44	13	32
1	48	36	9	50	3	58	25
35	14	56	18	30	37	19	45

17	57	38	5	21	55	26	53
41	8	28	52	39	10	43	4
62	22	44	13	32	60	20	29
9	50	3	58	25	1	48	36
18	30	37	19	45	35	14	56
6	46	59	12	54	7	51	27
40	23	0	42	24	33	16	63
11	49	31	15	61	47	2	34

C_1	C_0	C_0	C_0	C_1	C_0	C_0	C_1
C_0	C_1	C_0	C_0	C_0	C_0	C_0	C_0
C_1	C_0	C_1	C_0	C_0	C_0	C_0	C_1
C_0	C_1	C_0	C_1	C_1	C_1	C_1	C_1
C_1	C_0	C_1	C_0	C_0	C_0	C_0	C_0
C_0	C_1	C_0	C_0	C_1	C_0	C_0	C_1
C_1	C_0	C_0	C_1	C_0	C_1	C_0	C_1
C_0	C_0	C_0	C_0	C_0	C_0	C_0	C_0

Figure 4: Watermarking a printed image - Example

a) Original 128×128 monochrome image.

b) The watermark to be embedded is a binary sequence of 64 bits which is reordered as an 8×8 array. Black pixels represent '0', white pixels represent '1'.

c) The set of available dither cells consisting of two 8×8 arrays. The threshold values have been normalized to the range 0..63.

d) The 64×64 threshold pattern produced by the watermarking algorithm.

e) The watermarked halftone image.

f) The original image halftoned for comparison.

3.2 Decoding

Decoding of the watermark is performed by scanning the halftoned image and determining the sequence of dither cells used to create the halftone pattern. Assuming the the size of the watermark array ($N_0 \times M_0$) and the size of each dither cell ($N_1 \times M_1$) are known, the decoding reduces to the following problem: Given an appropriate $N_1 \times M_1$ region of the halftoned image, decide which of the dither cells in the available set was used. In the above example, given an 8×8 region, decide whether C_0 or C_1 was used.

Deciding which of the dither cells was used, is not trivial since, the underlying grayscale image, used to create the halftone pattern is unknown. To make this decision, the halftoning process is again exploited. The halftoning process is based on the HVS characteristic of a limited spatial resolution [33]. Reproduction of a color of the original image using halftoning is obtained by producing a pattern of available printer inks such that the local average of the pattern is similar to the original color. The HVS, being limited in spatial resolution performs a spatial blurring and in effect the reproduced (halftoned) image is seen as similar in color to the original.

Assuming the dither cells used in the watermarking described above, follow the rule of creating a pattern whose local average is equal to the original in color, the decoding of the watermark can be performed based on this rule. For a given region of the halftoned image, the decoding is performed as follows:

- Calculate the average color.
- Simulate the halftoning of a similar sized region having constant color equal to the average color using each of the available dither cells in turn.
- From among the simulated halftoning results, chose the one most similar to the original halftoned image.
- Select the appropriate dither cell as the one used to halftone the particular region of the halftoned image.

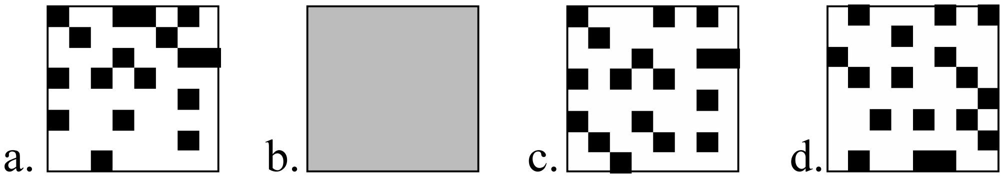


Figure 5: Decoding a bit of the watermark - Example
a) An 8×8 region of the watermarked halftoned image. The average gray level is 0.734 (in the range $0 \ldots 1$).
b) A constant 8×8 patch of gray value 0.734.
c) The halftone pattern obtained by thresholding constant patch b) with dither cell C_0.
d) The halftone pattern obtained by thresholding constant patch b) with dither cell C_1.
It can be seen that the original halftone is more similar to that produced by dither cell C_0. Thus the watermark bit is deduced to be '0'.

As an example consider the 8×8 region of the halftoned image of Figure 4e shown in Figure 5a. The average gray level of the region is 0.734 (in the range $0 \ldots 1$). An 8×8 image of constant gray level 0.734 (Figure 4b) is halftoned using dither cells C_0 and C_1 producing the halftoned images in Figure 5c and 5d respectively. A Hebb-metric is used to decide that C_0 produces a more similar result to the original halftoned region, than C_1 (34 pixels are inconsistent with C_1 while only 4 are inconsistent with C_0). Thus the appropriate bit of the watermark binary sequence is deduced to be '0'.

The decoding algorithm applied to the example of Figure 4 is shown in Figure 6. The watermark was decoded with no errors (compare Figure 6b and Figure 4b).

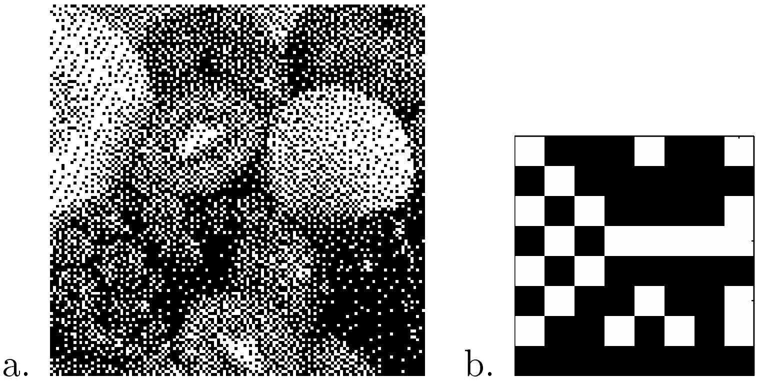


Figure 6: Decoding a watermark
a) The watermarked halftoned image.
b) The watermark is decoded with no errors (compare with Figure 4b).

3.3 Issues to Consider

On implementing, the watermarking technique for printed images suggested above, one must decide on the set of available dither cells to be used in the encoding and on the selection rule which chooses cells from this set according to the value of the watermark bit. When choosing the set of dither cells, the following issues and criteria must be considered:

- The dither cells should produce <u>different</u> halftone patterns for <u>all</u> colors or gray levels. It is obvious that the greater the difference between the patterns, the easier it is to distinguish between the dither cells. Note that at extreme gray levels the difference between dither cell patterns always decreases (for a white region all dither cells will produce a white halftone with no ink).
- The dither cells should be visually continuous, i.e. false contours and edges should not appear. As an extreme case, intermixing dither cells in the creation of the thresholding pattern for a constant colored image, should not produce textural artifacts in the halftone (see Figure 7 for an example where artifacts are visible when a single C_0 dither cell is used in an array of C_1 cells). This constraint is very hard to achieve due to the high sensitivity of the HVS to changes in texture pattern [33] and due to the difficulty in defining a measure to evaluate the visual perception of such inconsistencies.

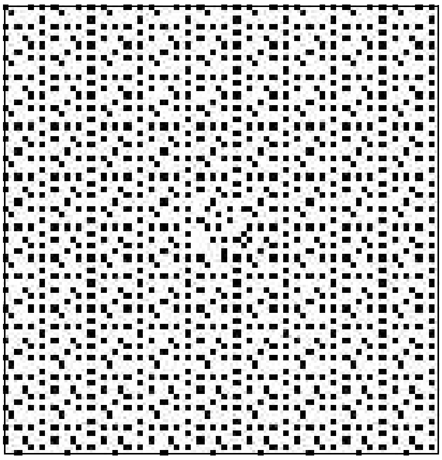


Figure 7: Artifacts are visible when a constant colored image is halftoned using a threshold pattern obtained using a single C_0 dither cell in an array of C_1 cells.

- The dither cells should have corresponding threshold values. This constraint is very subtle and requires some explanation. Watermark decoding is based on the rule that the local average color of the halftoned image is equal to the local average color in the original. If the original grayscale image region is a constant color then (exempting extreme color values) the decoding rule described above will provide a good selection between the dither cells. However, generally, this is not so. Typically there is some variation in color or gray value within an image region. This may give rise to cases where a wrong selection is made in the decoding process (see Figure 8). To minimize the instances where such errors occur, threshold values in the set of dithercells should be similar, i.e. the position of a given threshold value in one dither cell should not be distant from its position in other dithercells. In the example of Figure 8, the decoding error occurred due to the very large spatial distances between the threshold values of C_0 and C_1 (e.g. the disparity in position of threshold value 0 is very large). Note that this constraint conflicts with the first constraint. A compromise is typically chosen based on the smoothness model of the input grayscale image.

For the example case given above, C_0 and C_1 were chosen to fulfill these constraints to the extent possible, as follows: To satisfy the first constraint, C_0 and C_1 were chosen from a class of dither cells known as Stochastic dither cells [30]. In contrast to Cluster Dot dither cells [30] which have threshold values increasing systematically from the center of the cell, the Stochastic dither cells have threshold values that increase in a stochastic pattern throughout the cell [18, 31]. The Cluster Dot halftoning produces patterns in the form of clusters of same colored ink whereas the Stochastic halftoning produces patterns in which same colored ink dots are spread throughout a region. Thus, using Stochastic dither cells it is easier to find cells which have different halftone patterns for each gray level as required by the first constraint.

To satisfy the third constraint we chose to select C_0 and C_1 from among a set of cyclically rotated versions of a single Stochastic dither cell. The stochastic pattern of threshold values and the cyclic rotation produce correlated cells as required by the third constraint.

To satisfy the second constraint, we tested pairs from the set of cyclically rotated dither cells, by halftoning a constant valued image with a threshold pattern produced from a mixture of the

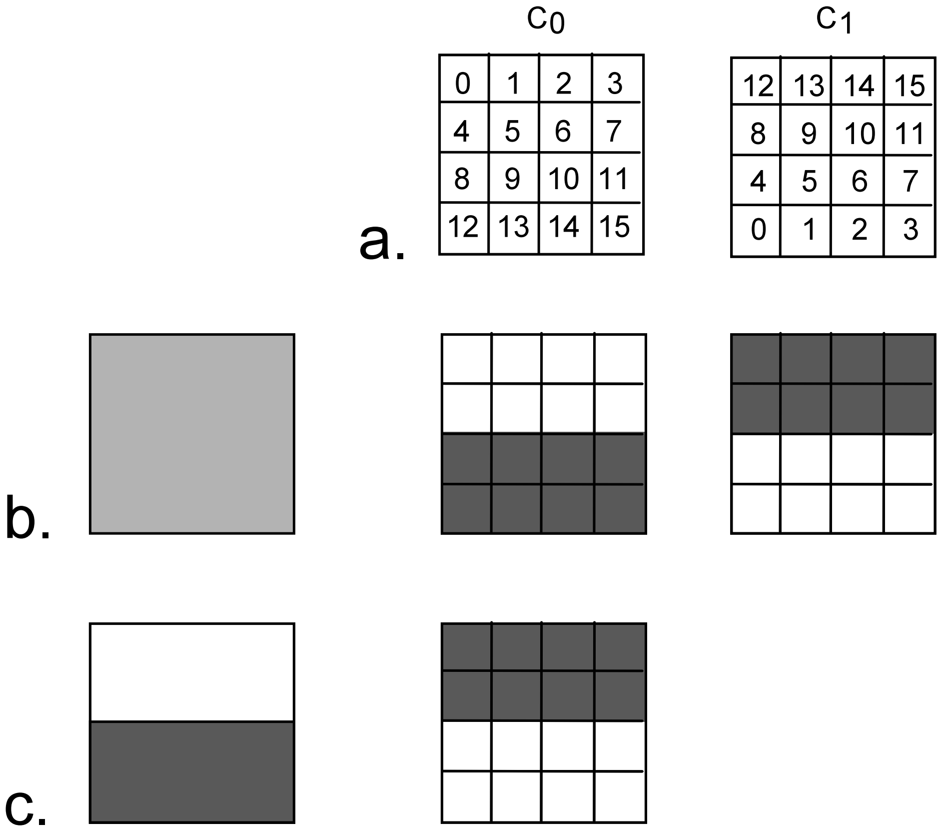


Figure 8: Incorrect watermark decoding of an image patch.
a) Two 4 × 4 dither cells (C_0 and C_1) whose threshold values have been normalized to $0 \dots 15$ for convenience.
b) A constant gray image of gray value 8 (in the normalized range) is halftoned using dither cells C_0 (middle) and C_1 (right). Image pixels that pass the corresponding threshold value are marked as white, otherwise they are marked as black.
c) A gray valued image (left) whose average is 8 (in the normalized range) but is not constant is halftoned using dither cell C_0 (right). Such an image would be classified incorrectly using the decoding algorithm (see text); The image halftoned using C_0 is more similar to the pattern obtained by halftoning the constant valued image of the same gray average, using dither cell C_1.

dither cell pair. We chose as C_0 and C_1, the pair of cells which produced a halftone with minimal distortion (distortion was evaluated visually). Two approaches can be used to create dither cells that have seamless boundries. In [11], a method is presented for creating seamless dither cells which can be used to tile thresholding patterns without visual artifacts. Using these dither cells, however would require a number of cells to be assigned to each possible watermark symbol (0 and 1). In [12, 34] given dither cells are modified within predefined regions in the cell. This approach can be used to create dither cells which vary only in their internal region while maintaining the threshold values at the cell boundary, again, allowing a seamless tiling.

It has been noted that the smaller the stochastic dither cell used in halftoning an image, the greater the visual artifacts such as periodic patterns. However, using the watermarking technique presented in this paper, the number of watermark bits that can be embedded within a fixed sized image increases with the decrease in dither cell size. Thus there is a need to combine the visual advantages of the large dither cell, while minimizing the the size in order to maintain a feasible number of embedded watermark bits. To overcome this problem two approaches are suggested both based on the fact that any sub region of a stochastic dither cell is itself stochastic. One approach creates several dither cells to represent each possible watermark symbol. A watermark symbol is embedded by randomly choosing from the cells that represent that symbol. Thus periodic artifacts can be reduced or overcome. A second approach uses large dither cells, each dither cell representing a sequence or array of two or more symbols. Thus a dither cell must be created for each combination of symbols (e.g. 2^n cells must be created when each dither cell represents a sequence of n binary symbols). In both approaches, the decoding process must determine which dither cell was used for encoding from within a larger set of dither cells. This, however did not increase the error rate in decoding (see Section 4).

The dither cells C_0 and C_1 that were selected were used in the example of Figure 4.

Finally, one must consider the case where extreme color values appear in an image region. In this case the halftoning pattern produced is similar under any dither cell (as explained in the first constraint above). To overcome such cases and to increase robustness under errors in the decoding, the thresholding pattern which was produced according to the watermark, is used repeatedly to tile the entire image for halftoning. Thus every bit in the watermark is embedded repeatedly in the image and the decoding is performed by voting for the appropriate dither cell. Thus extreme color valued regions might produce an error in the cell selection however the repeated encoding increases the chance that the overall vote will be for the correct dither cell.

4 Results

In addition to the example shown in Figure 4, the printed watermarking technique was applied to over 2000 additional images of size 128 × 128, 1500 of which were natural images and 500 random images. Example images from the test set are shown in Figure 9. The embeded watermark was of size 8×8 thus 4 multiple embeddings were performed in each image. The watermark was perfectly decoded in 99% of the trials. In all trials, including the failures, at most 36% of the watermark bits were incorrectly decoded.

In evaluating the results, it was found that most of the images over which the watermark decoding failed, had characteristic image statistics, namely, extreme mean gray values. Examples of images over which decoding failed are shown in Figure 10. The mean gray level of these images is bimodal with the regional mean being either under 10% or over 80% of the gray level range (i.e. mean gray level under 25 or over 200 from within the 0..255 range of gray values). Failure of the decoding process over these images is not surprising, since, as mentioned in Section 3.3, at extreme gray values, all dither cells produce the same pattern, and distinguishing between the cells used in encoding the watermark, is impossible.

A systematic study of the effect of mean image gray level on the success rate of the watermark decoding process is shown in Figure 11. Random images were created having gray level distributions with a fixed variance and a systematically increasing mean. For each mean gray level several images were created. Each image was embedded with a watermark as described above and decoded using the watermark decoding process presented in this paper. The number of watermark bits that were not decoded correctly in each image was averaged over those images with the same mean gray level. Figure 11 plots the average number of watermark bits that were incorrectly decoded, as a function of the mean gray level of the image (on a scale of 0..255). It can be seen that the rate of failure

Figure 9: Example images from the test set.

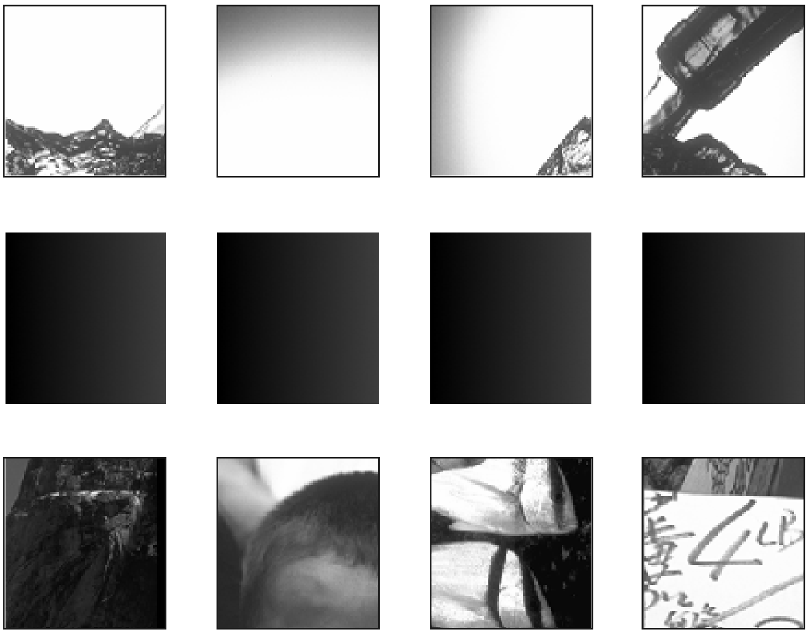


Figure 10: Example images from the test set for which the watermark decoding failed.

increases dramatically at the extreme mean gray values, as expected.

In an additional experiment, the same set of over 2000 images was used to test the watermarking technique when several dither cells are available for each watermark symbol ('0' and '1'). The watermark bits were embedded in every image by randomly selecting a dither cell from the set of cells associated with the specific watermark bit. The number of cells associated with each watermark symbol ranged from 2 to 10. It was found that the rate of successful decoding was maintained at around 99%. However the greater the number of cells per watermark symbol, the more sensitive the decoding is to noise. This is shown in Figure 12 where the average number of incorrectly decoded watermark bits is plotted as a function of the noise added to the halftoned image. The noise is given as the percentage of halftone dots that have been flipped. The average

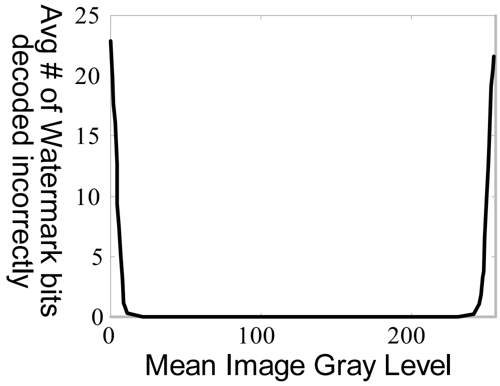


Figure 11: The average number of watermark bits decoded incorrectly as a function of the mean gray level of randomly created images.

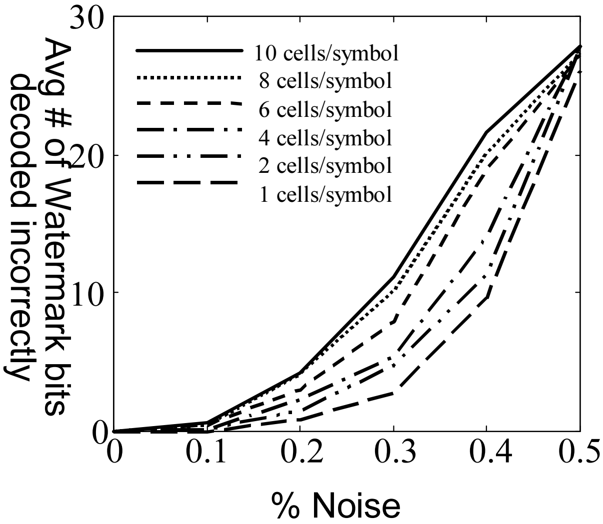


Figure 12: The average number of watermark bits decoded incorrectly as a function of the mean gray level of randomly created images.

was computed over 50 watermarked images at each noise level. It can be seen that decoding failure rate increases as a function of noise, as expected. Failure rate also increases as the number of dither cells per watermark symbol increases.

5 Conclusion

A watermarking technique was presented which embeds a binary code in a printed image. The method produces a watermark which is inherently part of the printed image and is thus visually unobtrusive. The decoding algorithm is automatic following scanning of the image and was developed to be robust to errors. It is still left to verify that the method is not susceptible to forgery and that decoding is not possible without apriori knowledge which includes, the set of possible dither cells. Variants of the method can be developed to increase security of the watermark: the bits of the watermark should be ordered with a random number generator and the selected dither cell should be used in randomly selected positions of the image. The seed for the generator must then be provided as a key for decoding the watermark.

The technique presented above, was developed for monotone printing. Color printing involves additional issues such as gamut mapping and screening. These must be considered and possibly incorporated in techniques for watermarking printed color images. Typically, color printing is considered an extension of monochrome printing, namely the halftoning technique for monochrome printing is applied a number of times to produce several overlay halftone images (one for each printer ink). The above presented method can be extended to color printing by applying the encoding algorithm to each color band or extending directly to three dimensional dither cells. Issues that need to be considered include problems of aliasing which might arise when different colored dither cells are overlayed. These might produce visual artifacts that could call attention to the watermark. On the other hand, the varying sensitivity of the HVS to different colors and spatial frequencies can be exploited in embedding the watermark. For example, the HVS is least sensitive to errors in the short spectral wavelengths, thus the watermark can be embedded in the blue color band similar to the technique presented in [15]. In contrast to [15] where digital images are coded, compression attacks are not a hazard for watermarked printed images.

Finally, the watermaking technique presented here determines the printing parameters, thus it should be intergerated into the printer driver or should override the driver altogether.

References

[1] Z. Baharav and D. Shaked. Watermarking of dither halftoned images. In *Proc. of IS&T/SPIE Conf. on Security and Watermarking of Multimedia Content*, volume 3657, pages 307–316, 1999.

[2] W. Bender, D. Gruhl, N. Morimoto, and A. Lu. Techniques for data hiding. *IBM Systems Journal*, 35(3/4):313–336, 1996.

[3] F. M. Boland, J. J. K. O Ruanaidh, and C. Dautzenberg. Watermarking digital images for copyright protection. In *Fifth International Conference on Image Processing and its Applications*, pages 326–330, Edinburgh, July 1995.

[4] A. G. Bors and I. Pitas. Image watermarking using dct domain constraints. In *IEEE International Conference on Image Processing*, volume 3, pages 231–234, Lusanne, Switzerland, September 1996.

[5] G. W. Braudaway, K. A. Magerlein, and F. Mintzer. Protecting publicly-available images with a visible image watermark. In *Proc. SPIE - The International Society for Optical Engineering*, volume 2659, pages 126–133, San Jose, CA, Feb, 1996. http://www.research.ibm.com/image_apps/wmsample.html.

[6] G. Caronni. Assuring ownership rights for digital images. In H. H. Brüggemann and W. Gerhardt-Häckl, editors, *Reliable IT Systems, VIS '95.* Vieweg Publishing Co., 1995.

[7] I. Cox, J. Kilian, T. Leighton, and T. Shamoon. A secure, robust watermarking for multimedia. In R. Anderson, editor, *First International Workshop on Information Hiding*, volume 1174 of Lecture Notes in Computer Science, pages 185–206, Cambridge, UK, may, 1996. Springer-Verlag.

[8] I. Cox, J. Kilian, T. Leighton, and T. Shamoon. Secure spread spectrum watermarking for images, audio and video. In *IEEE International Conference on Image Processing*, volume 3, pages 243–246, Lusanne, Switzerland, September 1996.

[9] C. Dautzenberg and F. M. Boland. Watermarking images. Technical report, Trinity College Dublin, Department of Electronic and Electrical Engineering, 1994.

[10] C. T. Hsu and J.L. Wu. Hidden signatures in images. In *IEEE International Conference on Image Processing*, volume 3, pages 223–226, Lusanne, Switzerland, September 1996.

[11] D. Kacker and J.P. Allebach. Aperiodic micro screen design using dbs and training. In *Proc. of SPIE*, volume 3300, pages 386–97, 1998.

[12] K.T. Knox and S.G. Wang. Digital watermarks using stochastic screens. In *Color Imaging: Device Independent Color, Color Hard Copy and Graphic Arts, Proc. of SPIE*, volume 3018, pages 316–322, San-Jose, CA, 1997.

[13] E. Koch, J. Rindfrey, and J. Zhao. Copyright protection for multimedia data. In *International Conference on Digital Media and Electronic Publishing*, pages 203–213, Leeds, UK, December 1994.

[14] E. Koch and J. Zhao. Towards robust and hidden image copyright labeling. In *IEEE Workshop on Non-linear Signal and Image processing*, pages 452–455, Neos Marmaras, June 1995.

[15] M. Kutter, F. Jordan, and F. Bossen. Digital signature of color images using amplitude modulation. In *Proc. SPIE - The International Society for Optical Engineering*, volume 3022, pages 518–526, San Jose, CA, Feb, 1997.

[16] G. C. Langelaar, J. C. A. Van der Lubbe, and J. Biemond. Copy protection for multimedia data based on labeling techniques. In *17th Symposium on Information Theory in the Benelux*, Enschede, The Netherlands, May, 1996. http://www-it.et.tudelft.nl/people/gerhard/home.html.

[17] K. Matsui and K. Tanaka. Video steganography. In *IMA Intellectual Property Project*, volume 1, pages 187–206, January 1994.

[18] T. Mitsa and K.J. Parker. Digital halftoning technique using a blue-noise mask. *Journal of the Optical Society of America A*, 9(11):1920–1929, 1992.

[19] O. Paatelma and R. H. Borland. Method and apparatus for manipulating digital data works. Technical Report WIPO WO 95/20291, World Intellectual Property Organization, Patent, 1995.

[20] I. Pitas and T. H. Kaskalis. A method for signature casting on digital images. In *IEEE International Conference on Image Processing*, volume 3, pages 215–218, Lausanne, Switzerland, September 1996.

[21] D. C. Rhoads. Method and apparatus for manipulating digital data works. Technical Report WIPO WO 95/14289, World Intellectual Property Organization, Patent, 1995.

[22] J. J. K. O Ruanaidh, F. M. Boland, and O. Sinnen. Watermarking digital images for copyright protection. In *The Electronic Imaging and Visual Arts Conference*, Florence, Febuary 1996.

[23] J. J. K. O Ruanaidh, W. J. Dowling, and F. M. Boland. Phase watermarking of images. In *IEEE International Conference on Image Processing*, volume 3, pages 239–242, Lausanne, Switzerland, September 1996.

[24] R. G. Van Schnydel, A. Z. Tirkel, and C. F. Osborne. A digital watermark. In *IEEE International Conference on Image Processing*, volume 2, pages 86–90, Austin, Texas, 1994.

[25] J. R. Smith and B. O. Comiskey. Modulation and information hiding in images. In R. Anderson, editor, *First International Workshop on Information Hiding*, volume 1174 of Lecture Notes in Computer Science, pages 207–226. Springer-Verlag, 1996.

[26] Stoffel and Moreland. A survey of electronic techniques for pictorial reprodu ction. *IEEE Transactions on Communication*, COM-29:1898–1925, 1981.

[27] M. D. Swanson, B. Zhu, and A. H. Tewfik. Transparent robust image watermarking. In *IEEE International Conference on Image Processing*, volume 3, pages 211–214, Lusanne, Switzerland, September 1996.

[28] K. Tanaka, Y. Nakamura, and K. Matsui. Embedding secret information into a dithered multi-level image. In *IEEE Military Communications Conference*, pages 216–220, 1990.

[29] B. Tao and B. Dickenson. Adaptive watermarking in the dct domain. In *International Conference on Accoustics Speech and Signal Processing*, volume 4, page 2985, 1997.

[30] R.A. Ulichney. *Digital Halftoning.* Mit press, Cambridge, Mass, 1987.

[31] R.A. Ulichney. The void-and-cluster method for dither array generation. In *Human Vision, Visual Processing and Digital Display IV, Proc. of SPIE*, volume 1913, pages 332–343, San-Jose, CA, 1993.

[32] S. Walton. Image authentication for a slippery new age. *Dr Dobb's Journal*, April:18–2, 82–87, 1995.

[33] B. A. Wandell. *Foundations of Vision.* Sinaur Press, Sunderland, MA, 1995.

[34] S.G. Wang and K.T. Knox. Embedding digital watermarks in halftone screens. In *Proc. of IS&T/SPIE Conf. on Security and Watermarking of Multimedia Contents II*, volume 3971, pages 218–227, 2000.

[35] R. B. Wolfgang and E. J. Delp. A watermark for digital images. In *IEEE International Conference on Image Processing*, volume 3, pages 219–222, Lausanne, Switzerland, September 1996.

[36] J. Zhao and E. Koch. Embedding robust labels into images for copyright protection. In *International Congress on Intellectual Property Rights for Specialized Information, Knowledge and New Technologies*, pages 242–251, Vienna, Austria, August 1995.