International Journal of Image and Graphics

BINARY IMAGE WATERMARKING THROUGH BLURRING AND BIASED BINARIZATION

HAIPING LU

Multimedia Laboratory
Edward S. Rogers Sr. Department of Electrical and Computer Engineering
University of Toronto, 10 King's College Road, Toronto, Ontario M5S 3G4, Canada
hplu@ieee.org

YUN Q. SHI

Department of Electrical and Computer Engineering
New Jersey Institute of Technology, 323 M. L. King blvd, Newark, NJ 07102, USA
shi@njit.edu

ALEX C. KOT

School of Electrical and Electronic Engineering
Nanyang Technological University, Nanyang Avenue, Singapore 639798
eackot@ntu.edu.sg

LIHUI CHEN

School of Electrical and Electronic Engineering
Nanyang Technological University, Nanyang Avenue, Singapore 639798
elhchen@ntu.edu.sg

Received (26 September 2003)
Revised (07 December 2003)
Accepted (16 December 2003)

Digital watermarking has been proposed for the protection of digital medias. This paper presents two watermarking algorithms for binary images. Both algorithms involve a blurring preprocessing and a biased binarization. After the blurring, the first algorithm embeds a watermark by modifying the DC components of the Discrete Cosine Transform (DCT), followed by a biased binarization, and the second one embeds a watermark by directly biasing the binarization threshold of the blurred image, controlled by a loop. Experimental results show the imperceptibility and robustness aspects of both algorithms.

Keywords: digital watermarking; binary image; binarization.

1. Introduction

There has been a rapid growth in the usage of digital medias recently. Not only multi-level images, video, and audio are in digital forms, but binary images are also digitized in the applications involving electronic documents such as digitized

handwritten signature, legal and financial documents, digital books, maps, and architectural and electronic drawings. As multimedia contents are stored in the digital format, it has become possible to make identical copies, modify the content or forge information using powerful software and digital devices available with great ease. Digital watermarking is proposed to protect digital media contents and prevent or discourage illicit redistribution and reproduction of them through embedding copyright and authentication information within media contents.

Majority of digital image watermarking techniques in the literature are proposed for gray-scale/color images [1,2], while the digital watermarking methods for binary images are quite limited in comparison. One important reason for this difference is that binary images lack rich gray-scale information that can be easily modified imperceptibly. The systems working on gray-level images, in which the pixels may take on a wide range of values, are not directly applicable to binary images, in which there are only two pixel values and no small gray-level variation. Any modification in a binary image is a flipping from one level to the other. Thus, watermark embedding without causing visibly noticeable artifacts becomes more difficult for binary images.

Most of watermarking algorithms for binary images embed a watermark by modifying pixels directly in the spatial domain[3]. A class of algorithms have been developed for formatted text documents to embed data through the modification of certain format features, such as space, height, or width[4,5,6,7,8]. The modification is done by shifting text lines, words or characters, or flipping a group of pixels. These algorithms offer some robustness against copying or other processing/attacks, while they have limited capacity and are applicable to formatted text image only rather than general binary images. A number of other algorithms divide the image into small blocks and embed a watermark through the flipping of individual pixels [9,10,11,12,13]. These algorithms can be applied to general binary images and they have large capacity. However, they are for fragile watermarking applications and usually do not offer robustness. There are also algorithms specially proposed for halftone images [14].

In this paper, we present two binary image watermarking algorithms that use different approaches but share the same working principle that is different from those described above. One is the DC watermarking (DCW) algorithm and the other is the direct biasing watermarking (DBW) algorithm. They both involve a preprocessing that blurs the original binary image to a gray-level image and a binarization process with a biased binarization threshold. DCW embeds a watermark through the modification of the DC components of the Discrete Cosine Transform (DCT) [15,16], and DBW embeds a watermark by the direct biasing on the binarization threshold. The DCW algorithm is analyzed in details and the DBW algorithm is a simplified and enhanced version based on the analysis. Experiments show that the visual distortion introduced by these two algorithms is not obtrusive and they have some robustness against cropping and additive noise.

The organization of this paper is as follows. In the next section, we will present

the DCW algorithm and analyze it in details. Based on the analysis, we propose the DBW algorithm in Sec. 3. Experimental results and the conclusion are in Secs. 4 and 5.

2. Watermark Embedding in DC Components of DCT

The DCW algorithm is developed for binary images based on the embedding strategy proposed for gray-level images by Huang *et al.* [17]. The original binary image is blurred through a low-pass filtering to get a gray-level image and then the watermark is embedded in the DC components of the DCT of the blurred image, followed by an inverse DCT (IDCT) and a binarization process with a biased binarization threshold. The watermark used is a random number sequence with Gaussian distribution. The watermark extraction process requires the presence of the original binary image.

We investigate the feasibility of the DC component embedding for binary images first to show that watermark embedding in DC components is impossible for binary images if the embedding is done directly on binary images, or the binarization threshold is chosen to be simply the mid-point, i.e. the mean of the maximum and minimum intensities. The DCW algorithm is then introduced and analyzed to show why the watermark can survive even after binarization.

2.1. *The feasibility of DC components embedding*

An image $f(x, y)$ of size $N \times N$ can be represented by its IDCT as following[18]:

$$f(x,y) = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} \left\{ \rho(u)\rho(v)C(u,v) \cos\left[\frac{(2x+1)u\pi}{2N}\right] \cdot \cos\left[\frac{(2y+1)v\pi}{2N}\right] \right\}, \quad (1)$$

where $C(u, v)$ is the DCT coefficient and ρ is given as

$$\rho(u) = \begin{cases} \sqrt{1/N} \text{ for } u = 0 \\ \sqrt{2/N} \text{ for } u = 1, 2, ..., N-1. \end{cases} \quad (2)$$

Let $S_{AC}(x, y)$ be the sum of the contributions from all AC components (all values of u, v except $u = v = 0$), then

$$f(x,y) = [\rho(0)]^2 C(0,0) + S_{AC}(x,y), \quad (3)$$

where the DC component $C(0, 0)$ is defined as [18]:

$$C(0,0) = [\rho(0)]^2 \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y) = \frac{\sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y)}{N}. \quad (4)$$

Suppose that only the DC component is modified to $C'(0, 0)$ to embed a watermark and the image after modification of $f(x, y)$ is $f'(x, y)$. Thus,

$$f'(x,y) = [\rho(0)]^2 C'(0,0) + S_{AC}(x,y). \quad (5)$$

Therefore, the change due to the modification is

$$\Delta f(x,y) = f'(x,y) - f(x,y) = [\rho(0)]^2 \Delta C(0,0), \tag{6}$$

where

$$\Delta C(0,0) = C'(0,0) - C(0,0). \tag{7}$$

From the above equations, $\Delta f(x_i, y_j)$, the change in the intensity of a particular pixel $f(x_i, y_j)$, is a constant for all pixels in the image $f(x,y)$ and it is independent of its position (x_i, y_j). Therefore, after modification,

$$f'(x,y) = f(x,y) + const. \tag{8}$$

For binary image watermarking, when $f'(x,y)$ is not binary, it has to be binarized using a binarization threshold T_{bi} to obtain the watermarked binary image $g(x,y)$. Usually uniform (all black/white) image blocks are not considered to be watermarked for imperceptibility. We observe the following two properties regarding binary image watermarking through DC component modification.

Property 1. *For a binary image $f(x,y)$ such that $f(x_i, y_j) \in \{0,1\}$, where $i,j = 0,1,...,N-1$, a watermark embedded in $f(x,y)$ through DC modification cannot survive as long as the binarization threshold chosen is between the maximum intensity (I_{max}) and the minimum intensity (I_{min}) in $f(x,y)$.*

Proof. From Eq. (8) above, if $f(x,y)$ is binary, $f'(x,y) \in \{const, const+1\}$ has only two levels too and for any binarization threshold T_{bi} between I_{min} ($const$) and I_{max} ($const+1$), $const < T_{bi} < const+1$, the watermarked image

$$g(x_i, y_j) = \begin{cases} 0 \text{ if } f(x_i, y_j) = 0 \\ 1 \text{ if } f(x_i, y_j) = 1. \end{cases}$$

This means that the watermarked binary image is identical to the original binary image and the watermark embedded is removed completely. For other T_{bi}, $T_{bi} \geq const+1$ or $T_{bi} \leq const$, $g(x,y)$ becomes a uniform black/white image, which is not acceptable for imperceptibility. □

Property 2. *For a gray-level image $f(x,y)$, DC component modification has no effect on the image after binarization if a mid-point threshold is used.*

Proof. If a gray-level image $f(x,y)$ is to be binarized using the mid-point threshold, $T_{bi} = (I_{max} + I_{min})/2$, where I_{max} and I_{min} are the maximum intensity and minimum intensity in $f(x,y)$, respectively, from Eq. (8), the maximum and minimum intensities in $f(x,y)$ and $f'(x,y)$ have the similar relation too:

$$I_{max}{}' = I_{max} + const, \quad I_{min}{}' = I_{min} + const$$

and the mid-point threshold $T_{bi}{}'$ (in $f'(x,y)$) is calculated as:

$$\begin{aligned} T_{bi}{}' &= (I_{max}{}' + I_{min}{}')/2 \\ &= (I_{max} + I_{min})/2 + const = T_{bi} + const. \end{aligned} \tag{9}$$

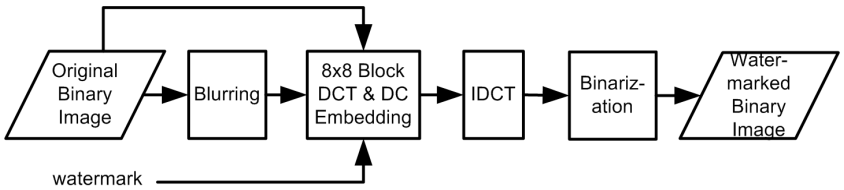


Fig. 1. DC components watermarking (DCW) algorithm for binary images.

We can see that

$$\begin{aligned} f'(x_i, y_j) > T_{bi}{}' \quad if \quad f(x_i, y_j) > T_{bi}, \\ f'(x_i, y_j) = T_{bi}{}' \quad if \quad f(x_i, y_j) = T_{bi}, \\ f'(x_i, y_j) < T_{bi}{}' \quad if \quad f(x_i, y_j) < T_{bi}. \end{aligned}$$

Therefore, if a mid-point threshold is used, doing binarization before or after the modification of the DC component has the same binary output image, which means that the watermark embedding is meaningless and the embedding will fail. □

2.2. *Successful watermark embedding in DC components*

Based on the two properties derived above, we propose the DC components watermarking algorithm. As shown in Fig. 1, the DCW algorithm proposed for binary images is similar to that proposed by Huang *et al.*[17], except that there are a preprocessing that blurs the input binary image to a gray-level image, and a postprocessing that binarizes the image after embedding to a binary image. These two processing are described below.

The blurring preprocessing is to obtain a gray-level image from the input binary image. A low-pass filtering works fine for this purpose. This preprocessing is necessary to avoid the failure of watermark embedding in DC components due to Property 1 discussed in the previous section. We use a Gaussian low-pass filter with a window of size 5×5. We observed that the filtering of a binary image with white background is likely to produce obtrusive noises near the edges of the image. The cause is that when the filter processes the pixels near the image edges where the filter window is only partially within the image, the part of the window that is outside of the image is treated as with pixel value 0 (black). However, black pixels near the image edges are obvious against the white background. To solve this problem, we expand the original image by two pixels with the background color (white) for the four image edges before blurring.

The post-embedding binarization is another critical step for successful watermarking. Binarization is to ensure that the watermarked image is still a binary image. As discussed above, a mid-point threshold will lead to embedding failure. Therefore, it is necessary to find a suitable threshold method such that the watermark embedded can survive after the binarization, yet the visual distortion resulted

is not obtrusive. We succeed by introducing a bias B_{bi} in determining the binarization threshold. Below we give a brief description of the watermarking algorithm.

(i) The original image $f(x, y)$ of size $M \times N$ is expanded with white pixels (two at each image edge) to an image $f_e(x, y)$ of size $(M + 2) \times (N + 2)$. The expanded image $f_e(x, y)$ is then low-pass filtered using a Gaussian filter with a window of size 5×5 and a standard deviation of σ to produce $f_{eb}(x, y)$, from which the blurred version $f_b(x, y)$ is obtained by ignoring two pixels at each image edge. We choose $\sigma = 1$ in the testing of the DCW algorithm. A larger σ offers better robustness but poorer quality, and vice versa. The blurred image $f_b(x, y)$ is then split into non-overlapped blocks of 8×8.

(ii) The non-uniform 8×8 blocks in the original image $f(x, y)$ are identified and those uniform blocks (all black/white) will be skipped in embedding for imperceptibility. Denote each block in $f_b(x, y)$ corresponding to the non-uniform blocks in $f(x, y)$ as $f_{b\_k}(r, s)$, $r, s = 0, 1, ..., 7$, and $k = 0, 1, ..., N_f - 1$, where N_f is the number of non-uniform 8×8 blocks in $f(x, y)$.

(iii) Each block $f_{b\_k}(r, s)$ is DCT transformed to get

$$C_{b\_k}(u, v) = DCT\{f_{b\_k}(r, s)\}, \qquad 0 \le u, v < 8. \tag{10}$$

The watermark W_G with length $L_W = N_f$ is a random number sequence with Gaussian distribution $N(0, 1)$. The watermark is embedded one element per block by modifying the DC value in $C_{b\_k}(u, v)$ as:

$$C'_{b\_k}(u, v) = \begin{cases} C_{b\_k}(u, v) \cdot (1 + \alpha \cdot W_G(k)) & \text{if } u = v = 0 \\ C_{b\_k}(u, v) & \text{otherwise.} \end{cases} \tag{11}$$

where α is a scaling factor.

(iv) The image block is IDCT transformed to obtain $f'_{b\_k}(r, s)$, the gray-level image block after embedding:

$$f'_{b\_k}(r, s) = IDCT\{C'_{b\_k}(u, v)\}. \tag{12}$$

This gray-level image block is then binarized to obtain the watermarked binary image block $f'_{bb\_k}(r, s)$ using a biased threshold T'_{bib_k}:

$$f'_{bb\_k}(r, s) = \begin{cases} 0, \text{ if } f'_{b\_k}(r, s) < T'_{bib\_k} \\ 1, \text{ if } f'_{b\_k}(r, s) \ge T'_{bib\_k} \end{cases} \tag{13}$$

and

$$T'_{bib\_k} = (I'_{max\_k} + I'_{min\_k}) \cdot (0.5 - B_{bi}), \tag{14}$$

where I'_{max_k} and I'_{min_k} are the maximum and minimum intensities in the block image $f'_{b\_k}(r, s)$, respectively, and B_{bi} is the bias in the binarization ($0 < B_{bi} < 0.5$). A large value of B_{bi} will result in better robustness while poorer image quality, and vice versa. Therefore, we prefer a small value of B_{bi} for less visual distortion. Experimentally, the typical choice of B_{bi} is 0.0004 for text document images.

(v) The whole watermarked image $g(x, y)$ is then obtained by replacing the N_f non-uniform 8×8 blocks in $f(x, y)$ with the modified ones.

2.3. *Watermark detection*

In the watermark detection for a test binary image $g(x, y)$, the original binary image $f(x, y)$ is required. From $f(x, y)$, the non-uniform 8×8 blocks are identified and the corresponding blocks in $g(x, y)$ are used to extract the watermark $\hat{W_G}$. The original image is blurred using the same Gaussian filter to get $f_b(x, y)$ for the extraction. Let $C^*_k(u, v)$ denote the DCT of the corresponding block $g_k(x, y)$, then

$$\hat{W_G}(k) = C^*_k(0, 0) - C_{b\_k}(0, 0). \tag{15}$$

The normalized correlation between $\hat{W_G}$ and W_G is calculated to determine whether $g(x, y)$ is a watermarked copy:

$$corr(\hat{W_G}, W_G) = \frac{\sum_{k=0}^{N_f-1}(\hat{W_G}(k)' \cdot W_G(k))}{\sqrt{\sum_{k=0}^{N_f-1} \hat{W_G}(k)'^2 \cdot \sum_{k=0}^{N_f-1} W_G(k)^2}}, \tag{16}$$

where $\hat{W_G}(k)' = \hat{W_G}(k) - \overline{\hat{W_G}}$ has a zero mean with $\overline{\hat{W_G}}$ denoted as the mean of $\hat{W_G}$. A correlation/similarity threshold T_{wm} is chosen based on experimental results to make a decision. $\hat{W_G}$ is classified as a corrupted version of the true watermark W_G if $corr(\hat{W_G}, W_G) > T_{wm}$.

2.4. *Analysis on the DCW algorithm*

To see why the watermark can survive, the effects of the bias in the binarization need to be examined. Suppose that T_{mid_k} is the mid-point of $f_{b\_k}(r, s)$ and T'_{mid_k} is the mid-point of $f'_{b\_k}(r, s)$ after the embedding of $W_G(k)$. From Eqs. (6), (7), (9) and (11), we have

$$\begin{aligned} T'_{mid\_k} &= T_{mid\_k} + [\rho(0)]^2 \cdot [C'_{b\_k}(0, 0) - C_{b\_k}(0, 0)] \\ &= T_{mid\_k} + (1/8) \cdot [\alpha \cdot W_G(k) \cdot C_{b\_k}(0, 0)] \\ &= T_{mid\_k} + [\alpha \cdot C_{b\_k}(0, 0)/8] \cdot W_G(k) \\ &= T_{mid\_k} + D_k \cdot W_G(k), \end{aligned} \tag{17}$$

where $D_k = \alpha \cdot C_{b\_k}(0, 0)/8$ is a constant for the image block. From Eq. (14),

$$\begin{aligned} T'_{bib\_k} &= (I'_{max\_k} + I'_{min\_k}) \cdot (0.5 - B_{bi}) \\ &= (I'_{max\_k} + I'_{min\_k})/2 - (I'_{max\_k} + I'_{min\_k}) \cdot B_{bi} \\ &= T'_{mid\_k} - 2 \cdot T'_{mid\_k} \cdot B_{bi}. \end{aligned} \tag{18}$$

Suppose we have $|D_k \cdot W_G(k)| > |T_{mid\_k}|$. If $W_G(k) > 0$, then $D_k \cdot W_G(k) > T_{mid\_k}$, and from Eq. (17), $T'_{mid\_k} > 0$. According to Eq. (18), the binarization threshold is lowered by $2 \cdot T'_{mid\_k} \cdot B_{bi}$. Lower threshold raises the probability of increased number of '1's (white pixels) after the binarization, which results in an increased DC value according to Eq. (4). Similarly, if $W_G(k) < 0$, then

$D_k \cdot W_G(k) < -T_{mid\_k}$, $T'_{mid\_k} < 0$, and the binarization threshold is raised by $|2 \cdot T'_{mid\_k} \cdot B_{bi}|$. Higher threshold raises the probability of increased number of '0's (black pixels) after the binarization, which results in a decreased DC value. Therefore, a positive $W_G(k)$ tends to raise the DC value and a negative $W_G(k)$ tends to lower it. Furthermore, the larger the value of $|W_G(k)|$ is, the stronger such tendencies will be. On the other hand, larger magnitudes of α and B_{bi} will strengthen such tendencies too, while causing more visual distortion at the same time. This explains why the watermark embedded can still survive although not all $W_G(k)$ may result in a DC value change and succeed in the embedding.

We compute the probability of $|D_k \cdot W_G(k)| > |T_{mid\_k}|$ to get

$$\begin{aligned} P[|D_k \cdot W_G(k)| > |T_{mid\_k}|] &= P[|(\alpha/8) \cdot C_{b\_k}(0,0) \cdot W_G(k)| > |T_{mid\_k}|] \\ &= P[|W_G(k)| > (8/\alpha) \cdot |T_{mid\_k}/C_{b\_k}(0,0)|]. \end{aligned}$$

Let I_{max_k} and I_{min_k} denote the maximum and minimum intensities in the block $f_{b\_k}(r,s)$, respectively. We have $T_{mid\_k} = (I_{min\_k} + I_{max\_k})/2$ and from Eq. (4)

$$\begin{aligned} C_{b\_k}(0,0) &= (\textstyle\sum_{r=0}^{7} \sum_{s=0}^{7} f_{b\_k}(r,s))/8 \\ &\geq (63 \cdot I_{min\_k} + I_{max\_k})/8 \\ &\geq (62 \cdot I_{min\_k} + 2 \cdot T_{mid\_k})/8. \end{aligned}$$

In a binary image, a pixel value is either '0' or '1'. After a low-pass filtering, the blurred image will have pixel values ranging from '0' to '1'. Therefore, $f_{b\_k}(r,s) \geq 0$ and $I_{min\_k} \geq 0$. Then we have $C_{b\_k}(0,0) \geq (2 \cdot T_{mid\_k})/8 \geq (T_{mid\_k})/4$. Hence, $0 \leq |T_{mid\_k}/C_{b\_k}(0,0)| \leq 4$.

Since W_G has a normal distribution, when a large α ($= 90$) is selected, we can get a high probability of $|D_k \cdot W_G(k)| > |T_{mid\_k}|$, indicating that the watermark has high possibility to be embedded successfully. Fig. 2 shows the variation of this probability against the full range of $|T_{mid\_k}/C_{b\_k}(0,0)|$. It should be noted that in real images, the values of $|T_{mid\_k}/C_{b\_k}(0,0)|$ are concentrated at the lower range, as shown in the histogram in Fig. 3, which shows the distribution of $|T_{mid\_k}/C_{b\_k}(0,0)|$ among the non-uniform 8×8 blocks of all the eight CCITT standard binary test images [19].

3. Watermark Embedding through Direct Biasing of Binarization Threshold

Based on the insights from the analysis on the DCW algorithm, we can see that DC component modification is a spatial domain technique in effect and the modification of DC components is equivalent to adding constant values to all pixels in the spatial domain. Therefore, we developed the direct biasing watermarking (DBW) algorithm, which is more efficient than the DCW algorithm.

In DBW, the original binary image is blurred to a gray-level image to enable embedding as in DCW. However, DCT and IDCT are not involved, which greatly simplifies the watermarking procedure. The embedding is done by using the watermark information to bias the binarization threshold directly, i.e., the binarization

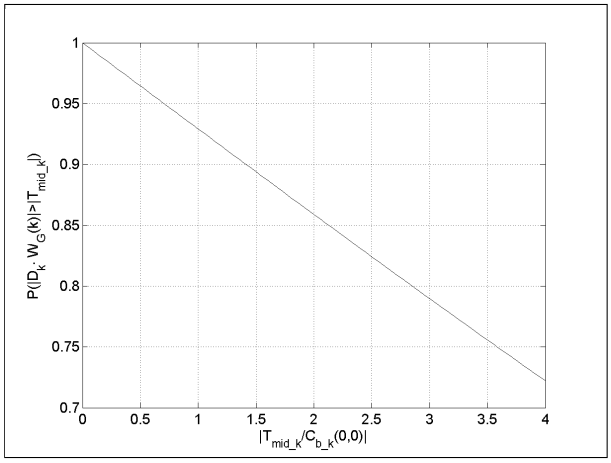


Fig. 2. The probability of $|D_k \cdot W_G(k)| > |T_{mid\_k}|$ against $|T_{mid\_k}/C_{b\_k}(0,0)|$ for $\alpha = 90$.

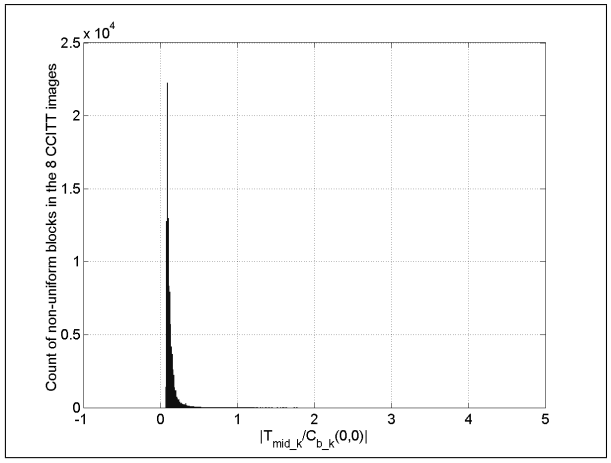


Fig. 3. The count of the non-uniform 8×8 blocks in the eight CCITT binary test images over the range of $|T_{mid\_k}/C_{b\_k}(0,0)|$.

procedure is the same as the embedding procedure, unlike the DCW algorithm, where the binarization procedure follows the embedding procedure. A loop is used to control the visual quality of the watermarked binary image and robustness. A feature vector is extracted as a key to be used in watermark extraction so that the original binary image is no longer necessary in the extraction (this can be done for DCW too). The watermark is a digital bitstream representing any kind of digital information rather than a random number sequence. To improve the extraction accuracy, the watermark is coded using error correction code (ECC).

Fig. 4 shows the system flow of the DBW algorithm. We embed the watermark

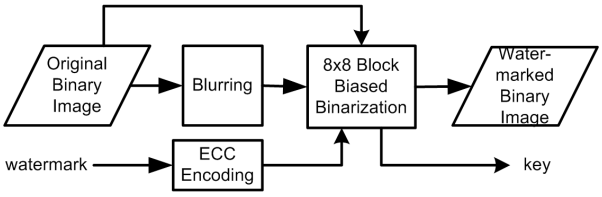


Fig. 4. The direct biasing watermarking (DBW) algorithm.

W into the original binary image $f(x, y)$ of size $M \times N$ to obtain the output image $g(x, y)$.

In DBW, blurring is a necessary preprocessing too. The blurred image $f_b(x, y)$ is obtained from the original image $f(x, y)$ in the same way as in DCW and split into non-overlapped blocks of 8×8. We choose $\sigma = 0.7$ for the low-pass filter in the testing of the DBW algorithm.

As usual, we skip the blocks in $f_b(x, y)$ corresponding to the uniform (all black/white) blocks in $f(x, y)$ to preserve the visual quality of the image after embedding. The watermark is embedded by binarizing the blocks in $f_b(x, y)$ that correspond to the non-uniform 8×8 blocks in $f(x, y)$ with biased thresholds. We denote the number of the non-uniform 8×8 blocks in $f(x, y)$ as N_f and each corresponding block in $f_b(x, y)$ as $f_{b\_k}(r, s)$, where $r, s = 0, 1, ..., 7$, and $k = 0, 1, ..., N_f - 1$.

The watermark W is a bitstream of '0's and '1's, and it is encoded with BCH(31,6) [21] to reduce the extraction error. The coded watermark W_c is of length $L_{Wc} \leq N_f$.

As mentioned, the DCW algorithm requires the original image in watermark extraction, which may not be convenient in practice. In the DBW algorithm, we eliminate this limitation by extracting a key to be used in the extraction. This key, K_N, is extracted as the number of white pixels in each block of $f(x, y)$ (both uniform and non-uniform) and it is of length L_{KN}, where L_{KN} is equal to the total number of 8×8 blocks in $f(x, y)$.

For each block $f_{b\_k}(r, s)$, the maximum and minimum intensities are I_{max_k} and I_{min_k}, respectively. The initial value of the binarization bias B_k depends on the watermark signal $W_c(k)$ as following:

$$B_k = \begin{cases} 0.05, & \text{if } W_c(k) = 1 \\ -0.05, & \text{if } W_c(k) = 0. \end{cases} \tag{19}$$

This bias is adjusted through a loop to control the amount of visual distortion and the robustness, as shown in Fig. 5. In the figure, C_L is a counter and it is initialized to 0. The maximum number of iterations is limited to M_L. The binarization threshold T_{b_k} is calculated as following:

$$T_{b\_k} = (I_{max\_k} + I_{min\_k}) \cdot (0.5 - B_k). \tag{20}$$

Thus, bit '1' in W_c will lower the threshold and bit '0' in W_c will raise it. The block

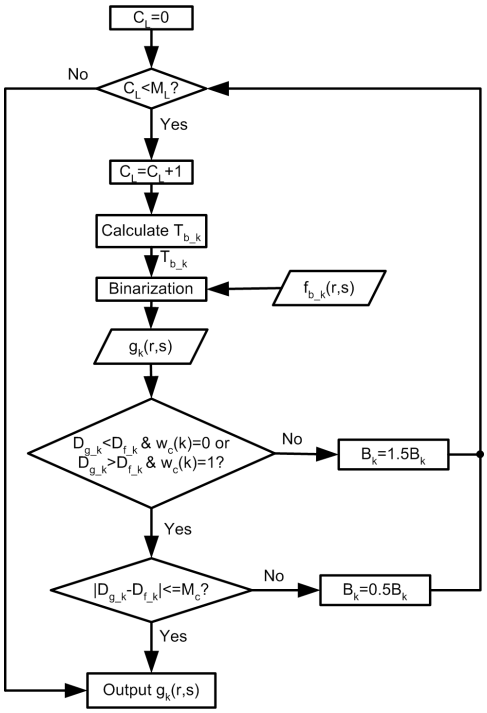


Fig. 5. Loop to control the embedding in the DBW algorithm.

$f_{b\_k}(r,s)$ is then binarized to $g_k(r,s)$ using T_{b_k}:

$$g_k(r,s) = \begin{cases} 0, \text{ if } f_{b\_k}(r,s) < T_{b\_k} \\ 1, \text{ if } f_{b\_k}(r,s) \geq T_{b\_k}. \end{cases} \tag{21}$$

Therefore, a lowered threshold (by bit '1' in W_c) tends to increase the number of white ('1's) pixels and a raised threshold (by bit '0' in W_c) tends to reduce it.

We denote the number of white pixels in $f_k(r,s)$ and that number in $g_k(r,s)$ as D_{f_k} and D_{g_k}, respectively. For successful embedding, we need to have the following condition:

$$\begin{cases} D_{g\_k} < D_{f\_k}, \text{ if } W_c(k) = 0 \\ D_{g\_k} > D_{f\_k}, \text{ if } W_c(k) = 1. \end{cases} \tag{22}$$

If this condition is not satisfied after the biased binarization, we increase the bias to $\lambda_I \cdot B_k$ ($\lambda_I > 1$) until the condition is satisfied or $C_L \geq M_L$.

On the other hand, when the condition in (22) is satisfied, it is possible that the visual distortion in the block is too much. Hence, for better visual quality, we should reduce the amount of bias when this happens. We denote the maximum acceptable number of flipped pixels in a block as M_c. Then we reduce the bias to $\lambda_D \cdot B_k$ ($0 < \lambda_D < 1$) if $|D_{g\_k} - D_{f\_k}| > M_c$. A larger M_c provides better robustness while resulting in poorer visual quality.

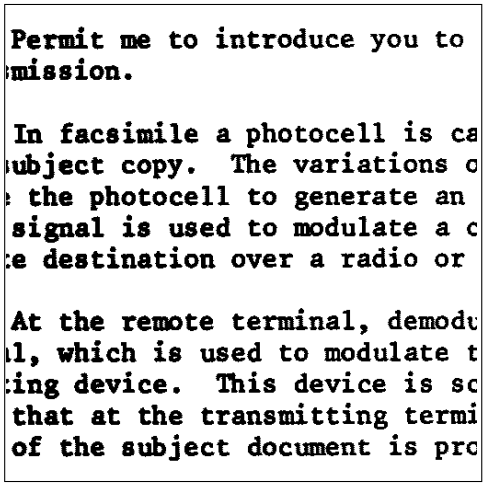


Fig. 6. The original binary image.

In our experiments, we choose $\lambda_I = 1.5$ and $\lambda_D = 0.5$. Inappropriate values of λ_I and λ_D may require more iterations to reach satisfactory results.

In case that the condition in (22) is not satisfied after the loop, we keep $g_k(r,s) = f_k(r,s)$, where $f_k(r,s)$ is the corresponding non-uniform 8×8 block in $f(x,y)$. The watermarked binary image $g(x,y)$ is obtained by replacing $f_k(r,s)$ in $f(x,y)$ with $g_k(r,s)$.

Watermark extraction is a simple process. To extract watermark $\hat{W}_c$ from a test binary image $g(x,y)$, we split $g(x,y)$ into 8×8 blocks. The key K_N is required and each element in K_N corresponds to an 8×8 block in $g(x,y)$. If the element from K_N is either 0 or 64, the corresponding block in $g(x,y)$ is skipped since there is no embedding in uniform blocks. Otherwise, a '1' is extracted if the number of white pixels in the block is greater than the value of the element from K_N, and a '0' is extracted otherwise.

4. Experimental Results

We use the text image shown in Fig. 6, which is a cut of the CCITT1 image [19], as the original image $f(x,y)$ to test the two proposed algorithms. Its size is 512×512. There are 1830 ($\approx 44.7\%$) non-uniform 8×8 blocks out of 4096 in $f(x,y)$.

4.1. *Watermarking results*

4.1.1. *The DCW algorithm*

The test image is watermarked using the DCW algorithm with $\alpha = 90$ and $B_{bi} = 0.0004$ for a balance between the robustness and imperceptibility, and the watermarked image is shown in Fig. 7(a). The watermark length $L_W = 1830$. In

the embedding, 1814 out of 1830 ($\approx 99.1\%$) blocks satisfy $|D_k \cdot W_G(k)| > |T_{mid\_k}|$, which agrees with our analysis in Sec. 2.4.

In the watermarked image, 2854($\approx$ 1.1%) pixels are flipped. The PSNR is $19.63dB$ and the DRDM measure [20] with $m = 5$ is 0.66. Visual inspection shows that the watermarked image is still in good quality and the visual distortion is not obtrusive. However, some of the "holes" in some characters, such as 'a', 'e', 'g' and 's', are filled completely by the embedding process, which is not desired. Fig. 7(b) shows the pixels flipped due to the watermark embedding. The original black pixels are brightened and the pixels flipped are shown as black dots. We can see that most of the pixels flipped are near the contours of the characters except some "hole-filling" defects.

The detector response in Fig. 8 shows the correlations between the extracted watermark and the true or false watermarks. The $500th$ watermark is the true one and all the others are false watermarks randomly generated. We can see that the detector gives a quite strong response, after undergoing the binarization process, which is a very strong interference as pointed out by Liu *et al.* [4]. From the detector response, the threshold T_{wm} in the detection is safely set to 0.1.

4.1.2. *The DBW algorithm*

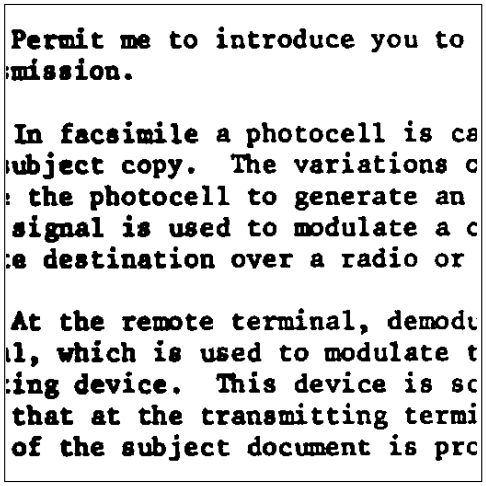
In the testing of the DBW algorithm, the key length $L_{KN} = 4096$ and $N_f = 1830$. We choose $M_L = 20$ and $M_c = 2$ for the control loop in our experiments. We generate a random bitstream W of length 354, and after BCH(31,6) coding we have the watermark W_c of length $L_{Wc} = 1829$ for embedding.

The image after watermark embedding is shown in Fig. 9(a). There are 2361($\approx$ 0.90%) pixels flipped in the image. We have the measured quality $PSNR = 20.45dB$ and $DRD = 0.58$ with $m = 5$ [20]. The perceived visual quality of the watermarked image is quite good and the visual distortion is not obtrusive. The "hole-filling" defects in the results for the DCW algorithm do not occur here, attributing to the loop control on the visual quality. Fig. 9(b) shows the pixels flipped due to the watermark embedding as in Fig. 7(b). Most of the pixels flipped are near the contours of the characters too and the embedding does not affect the visual quality of the binary image much.

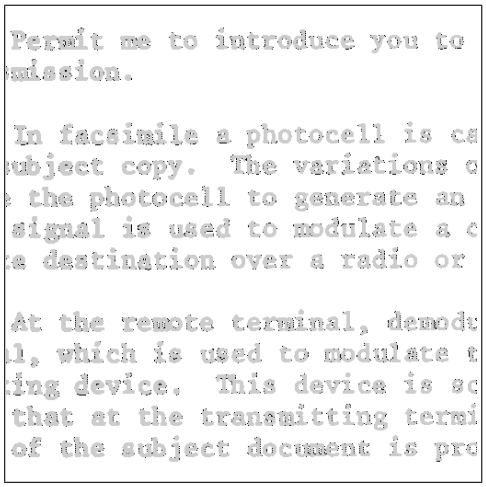
There is no error in the decoded $\hat{W}$ after the extraction of $\hat{W}_c$, attributing to the BCH coding. There are 121($\approx 6.62\%$) bits error in $\hat{W}_c$.

4.2. *Robustness test results*

The robustness against cropping and additive noise are tested. The cropping test is implemented as in the experiments by Liu *et al.* [4], where a number of rows are cropped from the watermarked image and the cropped portion is inserted into the original image to extract the watermark. The robustness against additive noise is tested by adding Gaussian white noise of mean 0 and variances ranging from 0.01



(a) Watermarked binary image.



(b) Flipped pixels.

Fig. 7. The watermarked binary image and the flipped pixels in it for the DCW algorithm.

to 0.1. The image after adding noise is a gray-level image and needs to be binarized with a mid-point threshold.

The results of the robustness tests against cropping and additive noise for the DCW algorithm are shown in Fig. 10(a) and Fig. 10(b), respectively. The ratio of pixels changed (flipped) in the watermarked binary image before and after adding noise is shown by the dash line in Fig. 10(b). We can see that mostly the detector responses are well above the threshold T_{wm}.

The results of the robustness tests against cropping and additive noise for the DBW algorithm are shown in Fig. 11(a) and Fig. 11(b), respectively. The Bit Error

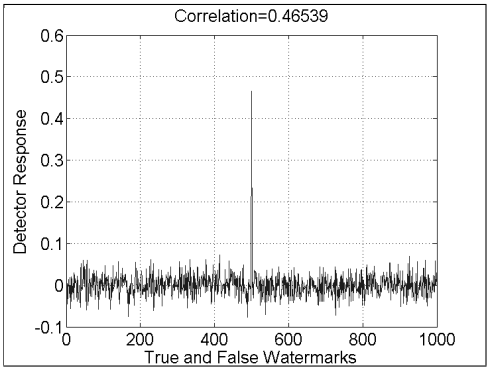


Fig. 8. Detector response for the watermarked image of Fig. 7(a).

Rates (BERs) are shown for both with ECC and without ECC. The ratio of pixels changed (flipped) in the watermarked binary image before and after adding noise is shown by the dash-dot line in Fig. 11(b). We can see that the ECC coding, BCH(31,6), is more effective with random additive noise than with cropping, and its effectiveness (improvement in BER) decreases as the amount of additive noise increases.

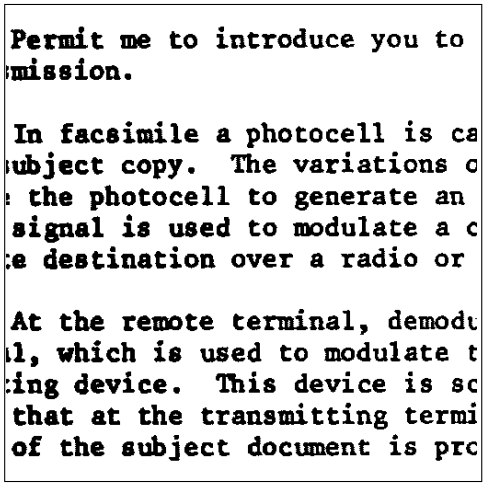
5. Conclusion

This paper presents two binary image watermarking algorithms that embed a watermark through blurring the original binary image to a gray-level one and introducing a bias in the binarization process.

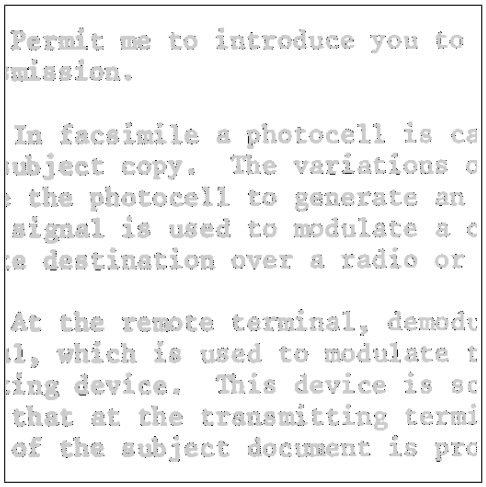
The feasibility of watermark embedding in the DC components of DCT for binary images is studied. It has been shown that direct embedding in the DC components of DCT for binary images is not feasible. Also, a mid-point threshold in binarization will remove the watermark information embedded in the DC components even if the original image is converted to a gray-level image. We propose the DC watermarking algorithm based on these two properties. The blurring preprocessing transforms a binary image into a gray one, removing the first obstacle, and the introduction of a biased binarization threshold combats the second obstacle.

Based on the analysis of the DCW algorithm, we propose an enhanced algorithm, the direct biasing watermarking algorithm. The original binary image is blurred to a gray-level image to enable embedding, and the watermark bitstream is then embedded by directly biasing the binarization threshold, without DCT and IDCT. Hence it is more efficient. A loop is used to control the visual quality of the watermarked image and the robustness. A key is extracted for watermark extraction so that the original binary image is not required in the extraction. For higher extraction accuracy, error correction code is used.

We have validated our analysis on the DCW algorithm in our experiments. The



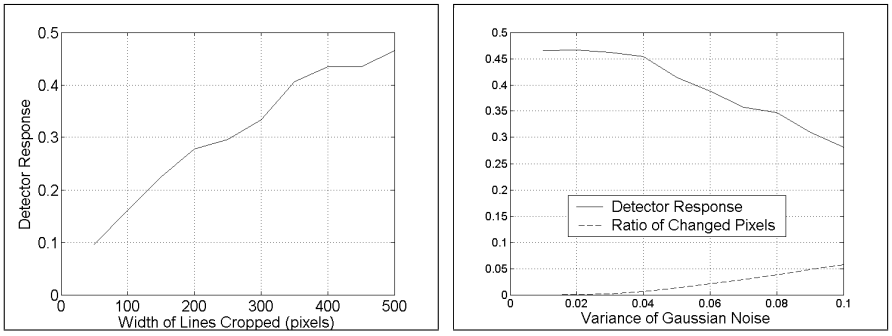
(a) Watermarked binary image.



(b) Flipped pixels.

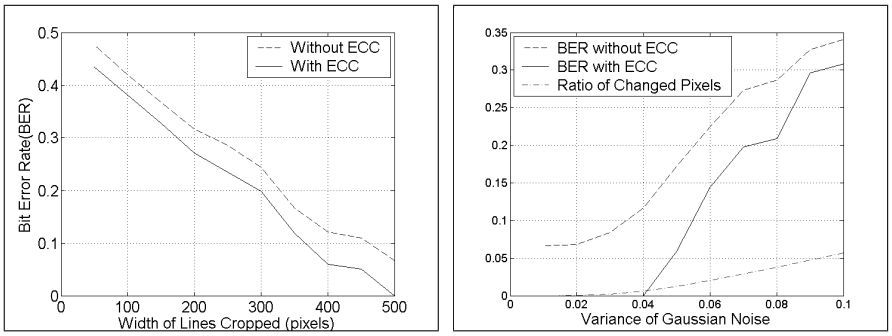
Fig. 9. The watermarked binary image and the flipped pixels in it for the DBW algorithm.

experimental results show that the visual distortion in the watermarked binary images is not obtrusive for both algorithms and they both have some degree of robustness against cropping and additive noise. In comparison, the DBW algorithm is more efficient than the DCW algorithm. Moreover, with the control loop, the DBW algorithm does not have the problem of "hole-filling" defects occurring in the DCW algorithm.



(a) Robustness against cropping. (b) Robustness against additive white Gaussian noise degradation.

Fig. 10. Robustness test results for the DCW algorithm.



(a) Robustness against cropping. (b) Robustness against additive white Gaussian noise degradation.

Fig. 11. Robustness test results for the DBW algorithm.

Acknowledgements

The authors would like to thank Xuxia Shi for her contribution in this paper.

References

1. F. A. P. Petitcolas, R. J. Anderson, and M. G. Kuhn, "Information hiding - a survey," *Proceedings of the IEEE*, vol. 87, no. 7, pp. 1062–1078, July 1999.
2. C. D. Vleeschouwer, J.-F. Delaigle, and B. Macq, "Invisibility and application functionalities in perceptual watermarking an overview," *Proceedings of the IEEE*, vol. 90, no. 1, pp. 64–77, Jan. 2002.
3. M. Chen, E. K. Wong, N. Memon, and S. Adams, "Recent developments in document image watermarking and data hiding," in *Proc. SPIE Conference on Multimedia Systems and Applications IV*, vol. 4518, Aug. 2001, pp. 166–176.
4. Y. Liu, J. Mant, E. Wong, and S. H. Low, "Marking and detection of text documents

using transform-domain techniques," in *Proc. SPIE Conference on Security and Watermarking of Multimedia Contents III*, vol. 3657, Jan. 1999, pp. 317–328.
5. J. T. Brassil, S. H. Low, and N. F. Maxemchuk, "Copyright protection for the electronic distribution of text documents," *Proceedings of the IEEE*, vol. 87, no. 7, pp. 1181–1196, July 1999.
6. J. Brassil and L. O'Gorman, "Watermarking document images with bounding box expansion," in *Proc. the 1st International Workshop on Information Hiding*, May 1996, pp. 227–235.
7. D. Huang and H. Yan, "Interword distance changes represented by sine waves for watermarking text images," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, no. 12, pp. 1237–1245, Dec. 2001.
8. N. Chotikakamthorn, "Document image data hiding techniques using character spacing width sequence coding," in *Proc. IEEE International Conference on Image Processing*, vol. 32, 1999, pp. 250–254.
9. M. Wu, E. Tang, and B. Liu, "Data hiding in digital binary image," in *Proc. IEEE International Conference on Multimedia and Expo*, vol. 1, July 31 to August 2, 2000, pp. 393–396.
10. H. Lu, A. C. Kot and J. Cheng, "Secure Data Hiding in Binary Document Images for Authentication", in *Proc. IEEE International Symposium on Circuits and Systems*, vol. 3, May 2003, pp. 806–809.
11. Y. C. Tseng, Y. Y. Chen, and H. K. Pan, "A secure data hiding scheme for binary images," *IEEE Transactions on Communications*, vol. 50, no. 8, pp. 1227–1231, Aug. 2002.
12. J. Zhao and E. Koch, "Embedding robust labels into images for copyright protection," in *Proc. International Congress on Intellectual Property Rights for Specialized Information, Knowledge and New Technologies*, Aug. 1995, pp. 242–251.
13. Q. G. Mei, E. K. Wong, and N. D. Memon, "Data hiding in binary text documents," in *Proc. of SPIE Conference on Security and Watermarking of Multimedia Contents III*, vol. 4314, Aug. 2001, pp. 369–375.
14. M. S. Fu and O. C. Au, "Data hiding watermarking for halftone images," *IEEE Transactions on Image Processing*, vol. 11, no. 4, pp. 477–484, Apr. 2002.
15. H. Lu, X. Shi, Y. Q. Shi, A. C. Kot and L. Chen, "Watermark Embedding in DC Components of DCT for Binary Images", in *Proc. IEEE International Workshop on Multimedia Signal Processing*, Dec. 2002, pp. 300–303.
16. X. Shi, "Digital watermarking in binary document images," Master's thesis, Nanyang Technological University, Singapore, 2001.
17. J. Huang, Y. Shi, and Y. Shi, "Embedding image watermark in dc components," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 10, no. 6, pp. 974–979, Sept. 2000.
18. R. C. Gonzalez and R. E. Woods, *Digital Image Processing.* Addison-Wesley Publishing Company, Inc., 1992.
19. P. Franti and E. Ageenko, "On the use of context tree for binary image compression," in *Proc. IEEE International Conference on Image Processing*, Oct. 1999, pp. 752–756.
20. H. Lu, J. Wang, A. C. Kot and Y. Q. Shi, "An Objective Distortion Measure for Binary Document Images Based on Human Visual Perception", in *Proc. International Conference on Pattern Recognition*, vol. 4, Aug. 2002, pp. 239–242.
21. B. Sklar, *Digital communications : fundamentals and applications.* Upper Saddle River, NJ: Prentice Hall, 2001.

Photo and Bibliography

Haiping Lu received the B.Eng. degree with first-class honors in the School of Electrical and Electronic Engineering (EEE) from Nanyang Technological University (NTU), Singapore, in 2001. He has just submitted his M.Eng. thesis for examination to the School of EEE, NTU, in 2003. Currently, he is pursuing the Ph.D. degree in the Edward S. Rogers Sr. Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON, Canada.

His research interests include image processing, data hiding/watermarking, information security, pattern recognition, and computer vision.

Yun Q. Shi has joined the Department of Electrical and Computer Engineering at the New Jersey Institute of Technology, Newark, NJ since 87, and is currently a professor there. He obtained his B.S. degree and M.S. degree from the Shanghai Jiao Tong University, Shanghai, China; his M.S. and Ph.D. degrees from the University of Pittsburgh, PA. His research interests include visual signal processing and communications, digital multimedia data hiding, digital image processing, computer vision and pattern recognition, theory of multidimensional systems and signal processing. Prior to entering graduate school, he had industrial experience in a radio factory as a principal design and test engineer in numerical control manufacturing and electronic broadcasting devices. Some of his research projects are currently supported by several federal and New Jersey State funding agencies.

He is an author/coauthor of more than 140 journal and conference proceedings papers in his research areas and a book on *Image and Video Compression for Multimedia Engineering*. He has been an IEEE senior member, the chairman of Signal Processing Chapter of IEEE North Jersey Section, an editorial board member of *International Journal of Image and Graphics*, a member of IEEE Circuits and Systems Society's Technical Committee of Visual Signal Processing and Communications as well as Technical Committee of Multimedia Systems and Applications, a member of IEEE Signal Processing Society's Technical Committee of Multimedia Signal Processing. He is currently an IEEE CASS Distinguished Lecturer, the guest editor of special issue on Image Data Hiding for *International Journal of Image and Graphics*, and the guest editor of special issue on Multimedia Signal Processing for *Journal of VLSI Signal Processing Systems*. He was a co-general chair of IEEE 2002 International Workshop on Multimedia Signal Processing, a formal reviewer of the *Mathematical Reviews*, an Associate Editor for *IEEE Transactions on Signal Processing* in the area of Multidimensional Signal Processing, the guest editor of the special issue on Image Sequence Processing for the *International Journal of Imag-*

ing Systems and Technology, one of the contributing authors in the area of Signal and Image Processing to the *Comprehensive Dictionary of Electrical Engineering.*

Alex C. Kot was educated at the University of Rochester, New York, and at the University of Rhode Island, Rhode Island, USA, where he received the Ph.D. degree in electrical engineering in 1989.

He was with the AT&T Bell Company, New York, USA. Since 1991, he has been with the Nanyang Technological University (NTU), Singapore, where he is Head of the Information Engineering Division. His research and teaching interests are in the areas of signal processing for communications, signal processing, watermarking, and information security.

Dr. Kot served as the General Co-Chair for the Second International Conference on Information, Communications and Signal Processing (ICICS) in December 1999, the Advisor for ICICS'01 and ICONIP'02. He received the NTU Best Teacher of the Year Award in 1996 and has served as the Chairman of the IEEE Signal Processing Chapter in Singapore. He is the General Co-Chair for the IEEE ICIP 2004 and served as Associate Editor for the *IEEE Transactions on Signal Processing* and the *IEEE Transactions on Circuits and Systems for Video Technology.*

Lihui Chen received the B.Eng. degree in Computer Science and Engineering at Zhejiang University, China, in 1982 and the Ph.D. degree in Computational Science at University of St. Andrews, UK, in 1992, respectively.

She is currently an Associate Professor of the Division of Information Engineering, School of Electrical and Electronic Engineering at Nanyang Technological University, Singapore. Her current research interests include Neural and Internet computing, especially in the areas of learning algorithms, data clustering, web intelligence and hybrid intelligent systems.