

Watermark Embedding in DC Components of DCT for Binary Images

Haiping Lu*, Xuxia Shi*, Yun Q. Shi†, Alex C. Kot* and Lihui Chen*

*School of EEE, Nanyang Technological University, Singapore 639798

Email: hplu@ieee.org

†Department of ECE, New Jersey Institute of Technology, Newark, NJ07102, USA

Email: shi@njit.edu

Abstract—This paper investigates the feasibility of watermark embedding in the discrete cosine transform (DCT) domain for binary images. Watermark embedding is known difficult for binary images due to their binary nature. For frequency domain approach to binary image watermarking, a post-embedding binarization is a necessary step to ensure that the watermarked image is still a binary image. This step disturbs the watermark embedded and is likely to remove the watermark. We have succeeded in combating this interference by embedding watermarks in the DC components of DCT and employing a biased binarization threshold. This algorithm can be applied to binary images in general and experiments show that the embedding algorithm proposed can not only survive binarization, but also provide some degree of robustness against common image processing.

I. INTRODUCTION

TODAY, digital media are getting more and more popular. Not only multi-level images, video, and audio are in digital form, but binary images are also digitized in the applications including legal documents, digital books, maps, and architectural and electronic drawings. Digital watermarking techniques have been proposed for ownership protection, copy control, annotation and authentication of digital media. Most of digital image watermarking techniques in the literature are proposed for gray-scale/color images, while the digital watermarking for binary images are only addressed by a few authors.

Chen et. al. [1] classify watermarking and data hiding techniques for binary images into various embedding methods: text line, word, or character shifting, boundary modification, fixed partitioning of the image into blocks, modification of character features, modification of run-length patterns, or modification of half-tone images. They compare different methods in terms of embedding techniques, robustness, advantage/disadvantages, data embedding capacity and limitations.

In all the recent works mentioned in [1], the watermarking or data hiding are all done by modifying spatial features in the spatial domain. Liu et. al. [2] have made an attempt to do the watermarking for binary images in the transform-domain using the discrete cosine transform (DCT) watermarking algorithm proposed by Cox et. al. [3]. They have shown by experiments that by cleaning background, i.e. setting all pixels with an intensity below a threshold to white, the strength of the watermark is attenuated. They show further that binarizing a watermarked

image, which goes one step further by setting all pixels with an intensity above the threshold to black, destroys the watermark embedded in text image completely without any other processing or attack for a wide range of threshold values chosen.

In this paper, we study the feasibility to embed watermarks for binary images in DC components of DCT, which is a new embedding strategy proposed by Huang et. al. in [4] for multi-level images. We prove 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 simply chosen to be the mid-point, i.e. the mean of the maximum and minimum intensities. Furthermore, we have proposed a successful watermarking algorithm for binary images, including a blurring pre-processing and a post-embedding binarization with a biased threshold such that the watermark can survive even after binarization and offer some robustness against common processing.

II. THE FEASIBILITY OF DC COMPONENTS EMBEDDING FOR BINARY IMAGES

As defined in [5], an image $f(x, y)$ of size $N \times N$ can be represented by inverse DCT (IDCT) as following:

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

where $C(u, v)$ is the DCT coefficient as defined in [5]. Denote by Sum_{AC} the sum of the contributions from all AC components (all values of u, v except $u = v = 0$), then

$$f(x, y) = [\alpha(0)]^2 C(0, 0) + Sum_{AC}, \quad \alpha(0) = \sqrt{1/N} \quad (2)$$

where the DC component is defined as in [5]:

$$C(0, 0) = [\alpha(0)]^2 \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) = \frac{\sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y)}{N} \quad (3)$$

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

$$f'(x, y) = [\alpha(0)]^2 C'(0, 0) + Sum_{AC} \quad (4)$$

Therefore, the change due to modification is

$$\Delta f(x, y) = f'(x, y) - f(x, y) = [\alpha(0)]^2 \Delta C(0, 0) \quad (5)$$

where

$$\Delta C(0, 0) = C'(0, 0) - C(0, 0) \quad (6)$$

From the above equations, the change in intensity of a particular pixel $f(x_i, y_j)$, $\Delta f(x_i, y_j)$ 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,

$$f'(x, y) = f(x, y) + const \quad (7)$$

For binary image watermarking, $f'(x, y)$ has to be binarized using a binarization threshold T_{bi} to obtain the binary watermarked image $f_{wm}(x, y)$, and uniform (all black/all white) images are not considered to be watermarked for imperceptibility. There are two properties regarding binary image watermarking through DC component modification.

- 1) If $f(x, y)$ is binary, i.e. $f(x_i, y_j) \in \{0, 1\}$ and $i, j \in \{0, 1, \dots, N-1\}$, watermark embedded 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 (7), if $f(x, y)$ is binary, $f'(x, y) \in \{const, const + 1\}$ is binary too and for any T_{bi} , $const < T_{bi} < const + 1$,

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

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

- 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 $f(x, y)$ 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. Also from (7), the maximum and minimum intensities in $f(x, y)$ and $f'(x, y)$ have the similar relations too:

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

and the mid-point threshold T_{bi}' is calculated as:

$$\begin{aligned} T_{bi}' &= (I_{max}' + I_{min}')/2 \\ &= (I_{max} + I_{min})/2 + const = T_{bi} + const \end{aligned} \quad (8)$$

We can see that

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

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

III. SUCCESSFUL WATERMARK EMBEDDING IN DC COMPONENTS FOR BINARY IMAGES

Based on the two properties derived in Section II, we have designed a DC components watermark embedding system successfully. As shown in Fig. 1, the watermarking algorithm proposed for binary images is similar to that proposed in [4], except that there are a pre-processing that blurs the binary image into a gray-level image, and a post-processing that binarizes the image after embedding into a binary image. These two processing are described below.

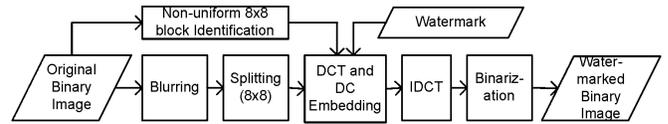


Fig. 1. DC components embedding for binary images

The blurring pre-processing is to obtain a gray-level image from the input binary image. A low-pass filtering works fine for this purpose. This pre-processing is necessary to avoid the failure of watermark embedding in DC components due to property 1 discussed in Section II.

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, we need to find a suitable threshold method such that the watermark embedded can survive after binarization yet the distortion resulted is not obtrusive. We succeed by introducing a bias B_{bi} in determining the binarization threshold. After a brief description of the watermarking algorithm, we will explain why it works.

- 1) The original image $f(x, y)$ is low-pass filtered using a Gaussian filter with window size of 5×5 and standard deviation of 1 to obtain the blurred version $f'(x, y)$. This blurred image is then split into non-overlapped blocks of 8×8 .
- 2) 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 $f'(x, y)$ corresponding to the non-uniform blocks in $f(x, y)$ as $f'_k(r, s)$, $r, s = 0, 1, \dots, 7$,

and $k = 0, 1, \dots, K - 1$, where K is the number of non-uniform blocks in $f(x, y)$.

- 3) Each block $f'_k(r, s)$ is DCT transformed as

$$C'_k(u, v) = DCT\{f'_k(r, s)\}, \quad 0 \leq u, v < 8$$

The watermark $W = \{w_k, 0 \leq k < K\}$ with length K 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 $C'_k(u, v)$ as:

$$C''_k(u, v) = \begin{cases} C'_k(u, v) \cdot (1 + \beta \cdot w_k) & \text{if } u = v = 0 \\ C'_k(u, v) & \text{otherwise.} \end{cases} \quad (9)$$

where β is a scaling factor.

- 4) The image block is IDCT transformed to obtain the gray-level image block after embedding $f''_k(r, s)$:

$$f''_k(r, s) = IDCT\{C''_k(u, v)\}$$

This gray-level image block is then binarized to obtain the watermarked binary image block $f'''_k(r, s)$ using a biased threshold $T_{bib.k}$.

$$f'''_k(r, s) = \begin{cases} 0, & \text{if } f''_k(r, s) < T_{bib.k} \\ 1, & \text{if } f''_k(r, s) \geq T_{bib.k}. \end{cases}$$

and

$$T_{bib.k} = (I''_{max.k} + I''_{min.k}) \cdot (0.5 - B_{bi}) \quad (10)$$

where $I''_{max.k}$ and $I''_{min.k}$ are the maximum and minimum intensities in block image $f''_k(r, s)$, and B_{bi} is the bias in binarization, and $0 < B_{bi} < 0.5$.

- 5) The whole watermarked image $f'''(x, y)$ is then obtained by replacing the K non-uniform blocks in $f(x, y)$.

In watermark detection for a binary image $f^*(x, y)$, the original binary image $f(x, y)$ is required. From $f(x, y)$, the non-uniform blocks are identified and the corresponding blocks in $f^*(x, y)$ are used to extract the watermark W^* . The original image is blurred using the same Gaussian filter to get $f'(x, y)$ for the extraction. Let $C_k^*(u, v)$ denote the DCT of the corresponding block $f_k^*(x, y)$, then

$$w_k^* = C_k^*(0, 0) - C'_k(0, 0), \quad k = 0, 1, \dots, K - 1$$

The normalized correlation between W^* and W is calculated to determine whether $f^*(x, y)$ is a watermarked copy.

$$corr(W^*, W) = \frac{\sum_{k=0}^{K-1} (w_k^* \cdot w_k)}{\sqrt{\sum_{k=0}^{K-1} w_k^{*2} \cdot \sum_{k=0}^{K-1} w_k^2}}$$

where $w_k^* = w_k^* - \overline{W^*}$ and the mean, $\overline{W^*}$, of W^* is subtracted to make the mean of the sequence to be zero. A threshold T_{wm} is then used to make a decision. W^* is classified as a corrupted version of the true watermark W if $corr(W^*, W) > T_{wm}$.

To answer why the watermark can survive, the effects of bias in binarization need to be examined. Suppose that $T'_{mid.k}$ is the mid-point for $f'_k(r, s)$ and $T''_{mid.k}$ is the mid-point for $f''_k(r, s)$ after embedding of w_k . From (5), (6), (8) and (9), we have

$$T''_{mid.k} = T'_{mid.k} + CW_k \cdot w_k \quad (11)$$

where $CW_k = \beta \cdot C'_k(0, 0)/8$ is a constant for the image block. From (10),

$$\begin{aligned} T_{bib.k} &= (I''_{max.k} + I''_{min.k}) \cdot (0.5 - B_{bi}) \\ &= (I''_{max.k} + I''_{min.k})/2 - (I''_{max.k} + I''_{min.k}) \cdot B_{bi} \\ &= T'_{mid.k} - 2 \cdot T'_{mid.k} \cdot B_{bi} \end{aligned} \quad (12)$$

Suppose we have $|CW_k \cdot w_k| > |T'_{mid.k}|$. If $w_k > 0$, $CW_k \cdot w_k > T'_{mid.k}$, and from (11), $T''_{mid.k} > 0$. According to (12), the binarization threshold is lowered by $2 \cdot T'_{mid.k} \cdot B_{bi}$. Lowered threshold raises the probability of increased number of 1s after binarization, which means increased DC value according to (3). Similarly, if $w_k < 0$, $CW_k \cdot w_k < -T'_{mid.k}$, $T''_{mid.k} < 0$, and the binarization threshold is raised by $|2 \cdot T'_{mid.k} \cdot B_{bi}|$. Raised threshold raises the probability of increased number of 0s after binarization, which means decreased DC value. Therefore, a positive w_k tends to raise the DC value and a negative w_k tends to lower it. Furthermore, the larger $|w_k|$, the stronger such tendencies. This explains why the watermark embedded can survive although not all w_k can result in DC value change and survive. On the other hand, the larger the magnitudes of β and B_{bi} , the stronger the tendencies too, while the more distortions resulted at the same time.

The probability of $|CW_k \cdot w_k| > |T'_{mid.k}|$ can be derived from the expected value of CW_k , where the expected value of a pixel in $f'_k(r, s)$ can be estimated as $T'_{mid.k}$:

$$\begin{aligned} E(CW_k) &= E[\beta \cdot C'_k(0, 0)/8] = \beta \cdot E[C'_k(0, 0)]/8 \\ &= \beta \cdot E\left[\sum_{r=0}^7 \sum_{s=0}^7 f'_k(r, s)/8\right]/8 \\ &= \beta \cdot E[64 \cdot T'_{mid.k}/8]/8 = \beta \cdot T'_{mid.k} \end{aligned}$$

Therefore,

$$\begin{aligned} P(|CW_k \cdot w_k| > |T'_{mid.k}|) &= P(|\beta \cdot T'_{mid.k} \cdot w_k| > |T'_{mid.k}|) \\ &= P(|\beta \cdot w_k| > 1) = P(|w_k| > 1/\beta) \end{aligned}$$

Because W has normal distribution, when a large β ($= 90$) is selected, we can get

$$\begin{aligned} P(|CW_k \cdot w_k| > |T'_{mid.k}|) &= P(|w_k| > 1/90) \\ &\approx 0.991 = 99.1\% \end{aligned}$$

The high probability of $|CW_k \cdot w_k| > |T'_{mid.k}|$ indicates that watermark has high possibility to be embedded successfully.

IV. EXPERIMENTAL RESULTS

The proposed algorithm has been tested on a wide range of binary images, including the eight CCITT images [6] and some

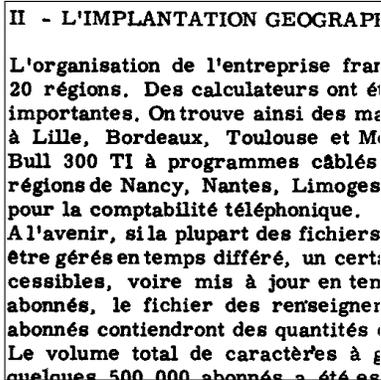


Fig. 2. Original binary image

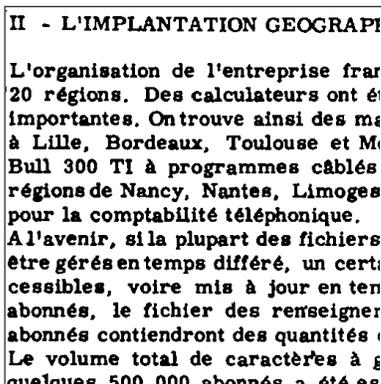


Fig. 3. Watermarked binary image

other types of binary images that we collected ourselves. In Fig. 2, we show a text image of 512×512 cut from one of the CCITT binary images.

This test image is watermarked using our algorithm with $\beta = 90$ and $B_{bi} = 0.0004$ for a balance between the robustness and imperceptibility, and the watermarked image is shown in Fig. 3. There are 2512 ($\approx 61.3\%$) non-uniform blocks available for watermark embedding. Hence the watermark length $K = 2512$. In the embedding, 2488 out of 2512 ($\approx 99.045\%$) blocks satisfy $|CW_k \cdot w_k| > |T'_{mid,k}|$. In the watermarked image, 4182 ($\approx 1.595\%$) pixels are changed. The detector response is shown in Fig. 4, where number 500 watermark is the true one and others are false. We can see that the distortion is not obtrusive and the detector gives a quite strong response, after undergoing the binarization process, which is a very strong interference as pointed out in [2]. From the detector response, we can safely set the threshold T_{wm} in detection to 0.1 in the following robustness tests.

The robustness against cropping and noises are shown in Fig. 5. The cropping test is implemented as in [2], where a number of complete rows are cropped from the watermarked image and the cropped portion is inserted into the original image to extract the watermark. The robustness against noises is tested by adding Gaussian white noise of mean 0 and variances ranging

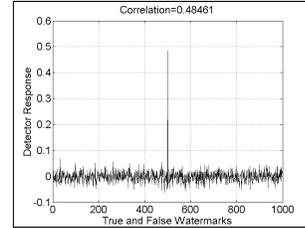


Fig. 4. Detector response for watermarked image

from 0.03 to 0.3. The image after adding noise is gray and needs to be binarized. The percentage of pixels changed in the binary image after adding noise, compared with the original image, is shown by the dashed line in Fig. 5(b).

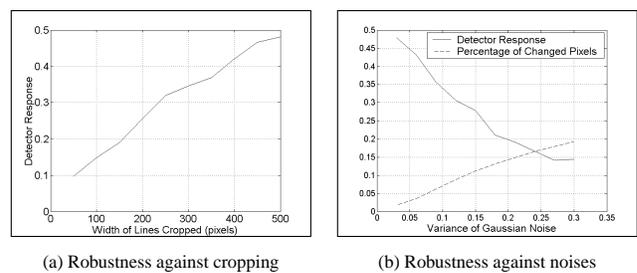


Fig. 5. Robustness tests

V. CONCLUSION

In this paper, we present a study of the feasibility of watermark embedding in DC components for binary images. It has been shown that direct embedding in DCT DC components for binary images is not feasible. Also, a mid-point threshold in binarization will remove the watermark information embedded in DC components. Based on these two observations, we propose an algorithm for binary images that can embed watermark successfully in DC components. A blurring pre-processing transforms a binary image into a gray one, removing the first obstacle. The introduction of a binarization threshold dynamically biased by watermark information is to combat the second obstacle. The experimental results show that the watermarking algorithm proposed offers some robustness with no obtrusive distortions.

REFERENCES

- [1] M. Chen, E.K. Wong, N. Memon, and S. Adams, "Recent developments in document image watermarking and data hiding," in *Proc. SPIE Conf. 4518: Multimedia Systems and Applications IV*, pp. 166-176, Aug. 2001.
- [2] Y. Liu, J. Mant, E. Wong, and S. Low, "Marking and detection of text documents using transform-domain techniques," in *Proc. SPIE Conf. on Security and Watermarking of Multimedia Contents*, pp. 317-328, Jan. 1999.
- [3] I.J. Cox, J. Kilian, T. Leighton, and T. Shamoan, "Secure spread spectrum watermarking for multimedia," *IEEE Trans. Image Processing*, vol. 6, pp. 1673-1687, Dec. 1997.
- [4] J. Huang, Y.Q. Shi, and Y. Shi, "Embedding Image Watermark in DC Components," *IEEE Trans. CSVT*, vol. 10, no. 6, pp. 974-979, Sep. 2000.
- [5] R.C. Gonzalez and R.E. Woods, *Digital Image Processing*, Addison-Wesley Publishing Company, Inc., 1992.
- [6] P. Franti and E.I. Ageenko, "On the use of context tree for binary image compression," in *Proc. ICIP*, pp. 752-756, Oct. 1999.