Abstract—Both watermark structure and embedding strategy affect robustness of image watermarks. Where should watermarks be embedded in discrete cosine transform (DCT) domain in order for the invisible image watermarks to be robust? Though many papers in the literature agree that watermarks should be embedded in perceptually significant components, dc components are explicitly excluded from watermark embedding. In this letter, a new embedding strategy for watermarking is proposed based on a quantitative analysis on the magnitudes of DCT components of host images. We argue that more robustness can be achieved if watermarks are embedded in dc components since dc components have much larger perceptual capacity than any ac components. Based on this idea, an adaptive watermarking algorithm is presented. We incorporate the feature of texture masking and luminance masking of the human visual system into watermarking. Experimental results demonstrate that the invisible watermarks embedded with the proposed watermark algorithm are very robust.

Index Terms—DC components, embedding strategy, image watermarking, robustness, visual masking.

I. INTRODUCTION

Digital watermarking for images, video, and audio has recently drawn extensive attention. Robustness is one of the most basic requirements for invisible image watermarks. Both watermark structure and embedding strategy affect robustness of watermarks. Cox et al. [1] claim that the watermarks composed of Gaussian random sequences are more robust. Our experiments support this statement [2].

The image watermarking algorithms fall into two groups. The first group of methods works in spatial domain by changing the gray levels of some pixels. Another group of techniques modifies the coefficients in transform domain. The methods in transform domain, especially in the discrete cosine transform (DCT) domain, are more popular for the following reasons:

1) The features of human visual system (HVS) can be incorporated into watermarking in the transform domain more effectively.
2) The energy of embedded signal in the transform domain will be spread over all pixels in the spatial domain. This is advantageous to invisibility.
3) They can be implemented in compressed domain since most international image and video compression standards, such as JPEG, MPEG, H. 261, and H. 263 are DCT-based.

In this paper, we address a problem related to the embedding strategy for invisible image watermarking in the DCT domain. Most of the early image watermarking schemes modify the least significant bits (LSB) of original images or their representation in transform domain [3] to meet the requirement of invisibility. These approaches are not robust, since the LSB data are highly sensitive to various noises and common signal processing procedures. Cox et al. [1] argue that the watermarks should be embedded in those perceptually significant components in order for the watermarks to be robust. Some researches [4], [5] make a tradeoff. They embed the watermarks in middle-band coefficients in the DCT domain. Where should watermarks be embedded in the DCT domain? Though the viewpoint that watermarks should be embedded in perceptually significant components has now been well accepted, dc components are explicitly excluded from watermark embedding [1], [2], [4], [5]. The consideration behind this is to avoid block artifacts in watermarked images. Indeed, an unreasonable amount of change in dc coefficients will result in a blocking effect in watermarked images because the average gray level of a block is proportional to the magnitude of the dc component. However, our quantitative analysis and experimental results on the magnitude of DCT components indicate that the watermarks may be more robust if they are embedded in dc components. Based on this observation, an adaptive watermarking algorithm is presented. Simulation results support our novel embedding strategy and demonstrate that the watermarks generated with the proposed watermarking algorithm are invisible and very robust against noise and common image processing techniques such as compression, low-pass filtering, clipping, and subsampling.

This paper is organized as follows. In Section II, we present a quantitative analysis on the magnitudes of DCT components for a few commonly used images with different texture features. Based on the analysis, we propose an embedding strategy that is different from the existing watermarking schemes. As an application of the novel embedding strategy, an adaptive watermarking algorithm utilizing the feature of texture masking of HVS is presented in Section III. The simulation results are shown and the conclusion is drawn in Section IV.

II. WHERE SHOULD WATERMARKS BE EMBEDDED?

In the DCT domain, watermarks should be embedded in those coefficients that meet the following requirements in order for the watermarks to be invisible and robust:

1) having large perceptual capacity [1] that allows strong watermarks to be embedded without perceptual distortion;
2) changing little with common image processing and noise corruption.

While the low-frequency ac components are widely considered to be good locations for watermark placement, dc components are explicitly excluded from watermark embedding. We argue that dc components are more suitable for watermarking than any ac components for the following reasons.

First, the magnitude of dc components is much larger than that of any ac components in general. Fig. 1 shows the average magnitude of 8 x 8 DCT coefficients at different spatial frequencies for a few commonly utilized images, including “Lena,” “Pepper,” and “Baboon.” There, the horizontal axis represents the spatial frequency, specifically, \( u + v, 0 \leq u, v < 8 \). The average magnitudes are computed as follows:

\[
\text{mag}(0) = \frac{1}{K} \sum_{k=0}^{K-1} F_k(0, 0)
\]
\[
\text{mag}(1) = \frac{1}{K} \sum_{k=0}^{K-1} \frac{1}{2} [F_k(0, 1) + F_k(1, 0)]
\]
\[
\text{mag}(2) = \frac{1}{K} \sum_{k=0}^{K-1} \frac{1}{3} [F_k(0, 2) + F_k(2, 0) + F_k(1, 1)]
\]
\[
\vdots
\]
\[
\text{mag}(i) = \frac{1}{K} \sum_{k=0}^{K-1} \frac{1}{8 - |i| - 7} \left( \sum_{u+v=i} F_k(u, v) \right)
\]

where \( K \) is the total number of 8 x 8 blocks in an image.

Obviously, Fig. 1 displays a huge discrepancy between dc components and any ac components in terms of magnitude.

Image watermarking can be viewed as superimposing a weak signal (watermark) onto a strong background signal (image). The superimposed signals can be detected by HVS only if they exceed the detection threshold of HVS. According to Weber’s law [6], the detection threshold of visibility for an embedded signal is proportional to the magnitude of the background signal. In other words, compared with ac components, although dc components cannot be changed by a larger percentage in order to avoid block artifacts under the constraint of invisibility, they can be modified by a much larger quantity due to their huge peak in the magnitude distribution. This indicates that dc components have much larger perceptual capacity than ac components. Fig. 2 compares the invisibility of watermarked images, using dc and ac components for watermarking, where both watermarked images are 44.1 dB in PSNR. It illustrates that the block artifacts in watermarked image can be avoided when embedding watermark into dc components. In other words, embedding watermarks in the dc components of the DCT does not necessarily cause visible block artifacts.

Second, according to the theory of signal processing, common image processing procedures, which watermarked images may encounter, such as data compression, low-pass filtering, subsampling, digital-to-analog (D/A) and analog-to-digital (A/D) conversions, tend to change dc components less than ac components.

We conducted experiments to compare the performance of different embedding strategies in terms of robustness against JPEG compression and additive noise. Fig. 3 demonstrates that the watermark cast into dc components (0,0) is more robust than that cast into low frequency ac components (1,0) or (2,0). There, the vertical axes are the similarity (correlation) between the extracted corrupted watermark and the original watermark. To compare these two cases fairly, the experimental conditions are set up as follows.

1) Embedding formula is chosen to be \( F'_i = F_i(1 + \alpha x_i) \), where \( F_i \) denotes the DCT coefficients of the original image.
2) Watermark \( W = \{ x_i \} \) is composed of a binary random sequence, i.e., \( x_i = \{-1, 1\} \).
3) Scaling factor \( \alpha \) is changed for embedding in dc and ac components, respectively, in order to have the PSNR’s of the watermarked images the same (44.1 dB in all cases) before JPEG compression and additive noise corruption.

The results in Fig. 3 are from the experiments on “Lena.” Similar results can be obtained on “Pepper” and “Baboon.” These results support the above observations. Our experiments show that the similar results can be achieved for embedding watermarks in different \( (u, v) \) as long as the sum of \( u \) and \( v \) keeps the same.

III. A WATERMARKING ALGORITHM USING DC COMPONENTS

The visibility of the superimposed watermark signal is affected by the luminance, the spatial-frequency, and the texture of the background in the following ways.

1) The brighter the background, the lower the visibility of the embedded signal (luminance masking) [6], [7].
2) The stronger the texture in the background, the lower the visibility of the embedded signal (texture masking) [6], [8].
3) The effect of the spatial-frequency distribution on the visibility of the embedded signal is modeled by contrast sensitivity function (CSF) [9], [10](frequency masking). An example of watermarking utilizing frequency masking can be found in [10].

In this section, we present an adaptive watermarking algorithm using spatial masking (both luminance and texture masking). The algorithm is featured by the following two elements: embedding watermarks in dc components and casting watermarks adaptively based on a block classification using edge-point density. It is implemented in the following three steps: 1) image splitting and block classification; 2) DCT and watermark embedding; and 3) inverse DCT.

The original image $f(x, y)$ is split into nonoverlapped blocks of $8 \times 8$, denoted as $B_k$, $k = 0, 1, \cdots, K - 1$. That is

$$f(x, y) = \bigcup_k B_k = \bigcup_k f_k(x', y'), \quad 0 \leq x', y' < 8,$$

In an image, the different blocks usually have different features. The features will affect the visibility of superimposed watermark signals. To embed watermarks as strongly as possible, we classify all blocks into two categories: $S_1$ with weak texture, and $S_2$ with strong texture. The strength of watermark components embedded into a block in $S_1$ should be weaker than that...
embedded into a block in $S_2$, since HVS is more sensitive to a gray-level change in the block in $S_1$ than in the block in $S_2$. On the other hand, a block in $S_2$ permits stronger watermark components to be inserted.

The classification is based on edge point density [11]. That is, $B_k \in S_1$ if

$$\text{number}\{c(x, y) \neq 0, (x, y) \in B_k\} < T_1$$

where $\text{number}\{p\}$ denotes the number of points satisfying the condition specified by $p$. $c(x, y)$ is a binary edge map of $f(x, y)$ obtained by applying a gradient operator followed by a thresholding; $T_1$ is a preset criterion. Otherwise, $B_k \in S_2$.

Each $B_k$ is DCT transformed

$$F_k(u, v) = \text{DCT}\{f_k(x', y')\}, \quad 0 \leq u, v < 8,$$

The watermark $W = \{x_i, 0 \leq i < n\}$ is composed of a random number sequence of length $n$, which obeys the Gaussian distribution $N(0, 1)$, where $n = K$. The watermark is embedded by modifying the dc coefficients as follows:

$$F_k^e(u, v) = \begin{cases} F_k(u, v) \cdot (1 + \alpha \cdot x_k), & \text{if } u = v = 0 \\ F_k(u, v), & \text{otherwise} \end{cases}$$

where $\alpha$, as introduced before, is a scaling factor. Since the variation of dc coefficients $\alpha \cdot x_k \cdot F_k(0, 0)$, i.e., the strength of embedded signals, is proportional to the values of dc coefficients, which is in turn proportional to the average brightness information of the background, the embedding formula has automatically utilized the luminance masking. We incorporate the texture masking into watermarking by changing the scaling factor adaptively. Based on many experiments, $\alpha$ is selected to be 0.006 for those blocks with weak texture ($S_1$) and 0.015 for the blocks with strong texture ($S_2$), respectively. The values are appropriate for various images.

The watermarked image is obtained by

$$f(x, y) = \bigcup_k \text{DCT}\{F_k^e(u, v)\}, 0 \leq u, v < 8.$$  

The detection of watermarks is carried out straightforwardly by employing correlation technique. Let $f_k(x', y')$ denote the DCT coefficients of the corrupted watermarked image in block $B_k$. The corrupted watermark $W^*$ is extracted by

$$W_k^* = F_k^e(0, 0) - F_k(0, 0), \quad W^* = \bigcup_k W_k^* = \{x_i^*, 0 \leq i < n\}$$

where $x_i^*$ is the corrupted version of $x_i$.

To determine if a watermark exists, we compute the similarity, denoted by $\rho$, between $W^*$ and $W$

$$\rho(W^*, W) = \frac{\sum_{i=0}^{n-1} (x_i^* \cdot x_i)}{\sqrt{\sum_{i=0}^{n-1} (x_i^*)^2} \cdot \sqrt{\sum_{i=0}^{n-1} x_i^2}}.$$  

If $\rho(W^*, W) > T_2$, a preset threshold, it indicates that there is a watermark existing in the testing image. According to [1], while the detected $W^*$ is not the corrupted version of the watermark $W$, the probability that $\rho(W^*, W) > T_2$ is equal to the probability of a Gaussian distributed random variable exceeding its mean by more than $T_2$ times of standard deviations. So $T_2$ is selected to be five in our work.

IV. SIMULATION RESULTS AND CONCLUSIONS

The proposed algorithm has been tested on various images. The experimental results are shown here with “Lena” and “Baboon” images of $256 \times 256 \times 8$ bits. Note that these two images represent images with quite different content “complexity.” The former contains mainly smooth regions, representing low complexity, while the latter contains rich high-frequency components, representing high complexity.
Fig. 4 demonstrates that the watermark embedded with the proposed algorithm is invisible, where (a) is the watermarked “Lena” images. Fig. 4(b) and (c) are the original and watermarked “Baboon” images. The PSNR’s of Fig. 4(a) and (c) are of 44.1 and 42.62 dB, respectively. In both cases, block artifacts are not noticeable. Fig. 5 demonstrates the robustness of watermark generated by using the proposed algorithm against JPEG compression and additive Gaussian noise corruption. Even though the PSNR’s of watermarked images are as low as 13.0 dB for “Lena” (11.0 dB for “Baboon”) due to noise corruption, the similarities in these cases are still above the threshold ($T_2 = 5$). For JPEG compression, the watermark can be well detected if the PSNR of coded watermarked “Lena” image is above 25.5 dB (20.5 dB for “Baboon” image).

We also test the robustness of the proposed watermarking against low-pass filtering, clipping, and subsampling. For “Lena” image, after $7 \times 7$ mean filtering, clipping to a quarter (lost 75% of the watermarked image), or 2:1 subsampling along both horizontal and vertical directions in the watermarked image, the similarities detected are 7.9, 14.5, and 11.3, respectively, all well above the threshold. Better robustness performance can be obtained for “Baboon” image since it has more complex texture features than “Lena.”
In summary, the main contributions of this paper are the following.

1) Proposing a new embedding strategy. We demonstrate that dc components should be used to place watermarks in order to improve robustness of watermarks, while all the previous methods avoided doing so.

2) Combining this discovery with the features of HVS, we present an adaptive watermarking algorithm. The watermarks embedded with the proposed algorithm are invisible and very robust.

It is noted that both dc and ac components can be used to place watermarks in the case, say, when watermarks are large in size. In doing so, the embedding formula should be different for dc and ac embedding according to our experiments.

REFERENCES


