

Research Article

An Efficient Image Watermarking Method Based on Fast Discrete Cosine Transform Algorithm

S. E. Tsai,¹ K. C. Liu,² and S. M. Yang³

¹Department of Computer Science and Information Engineering, Chang Jung Christian University, Tainan City 701, Taiwan

²Asian Institute of Telesurgery, Changhua County 500, Taiwan

³Department of Aeronautics and Astronautics, National Cheng Kung University, Tainan City 701, Taiwan

Correspondence should be addressed to S. E. Tsai; sean@mail.cjcu.edu.tw

Received 15 February 2017; Revised 24 May 2017; Accepted 21 June 2017; Published 16 August 2017

Academic Editor: Raffaele Solimene

Copyright © 2017 S. E. Tsai et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper proposes an image watermarking method based on the fast discrete cosine transform (DCT) algorithm for implementation in digital signal processor. A digital watermark can be effectively embedded and efficiently extracted without the host image. The keys in watermarking process include four frequency coefficients in DCT, two random permutation vectors, and a quantization matrix for normalizing the watermark and the host image. The fast DCT algorithm has been shown to reduce the complexity of two-dimensional image transformation so that embedding/decoding an image watermark can be completed in real time within 0.33 seconds. The quality of both watermarked image and extracted (retrieved) watermark remains excellent. It is shown that the watermarking method is efficient in and robust to data cropping, transmission loss, and compression/decompression.

1. Introduction

Conventional technique of embedding watermark signals or patterns into digital image is to insert a secret bit string in spatial, frequency, or wavelet domain. In spatial domain watermarking, the human visual system (HVS) is applied to assure that modifications of an original image by embedded watermark are imperceptible. Koz and Alatan [1] applied the temporal contrast threshold of HVS to prevent image distortion after watermarking, but the watermarked image is vulnerable to data transmission. Kaur et al. [2] proposed a technique to substitute the blue component of image pixels with character bits, but it is also vulnerable to signal processing attack. Agarwal [3] embedded a watermark with data encryption which required substantial data capacity thus leading to low image quality. Tyagi et al. [4] also applied the least significant bit (LSB) to achieve high security at the price of lower image quality. Nath et al. [5] proposed a method for hiding encrypted secret message inside a cover file, which again is vulnerable to data compression and attack. Sara et al. [6] selected the pixels with higher intensity to

embed watermark bits by LSB. A major constraint of the above studies is that they all require the original image data to decode the watermark.

In “blind” watermarking, Chadha et al. [7] proposed a method based on LSB and discrete cosine transform (DCT) for decoding watermark without the original image, but the method is sensitive to common image compression. Surrah and Mohamed [8] presented a binary data hiding algorithm by changing the position of every image block to increase security. Wang et al. [9] proposed watermarking by quaternion Fourier transform, but the image quality was reduced significantly by the embedded watermark. With Joint Photographic Experts Group (JPEG) adopting DCT to its compression process, many watermarking methods based on DCT have been developed. Benoraira et al. [10] proposed embedding the watermark bits sequence in both DCT and discrete wavelet transform (DWT) to guard against attack, but the procedure is computational intensive. A watermarking technique based on the so-called wavelet-tree [11] and another based on DWT [12] were also proposed achieving

better robustness; however, both required substantial computation with large data capacity.

Singular value decomposition (SVD) is desirable for implementation together with DCT or DWT because it requires extensive computations if applied separately onto images. Rawat and Raman [13] proposed an SVD-based watermarking scheme to increase security in copyright protection. Makbol and Khoo [14] and Lagzian et al. [15] also developed image watermarking schemes based on DWT and SVD to achieve robustness against image processing, but the computational load was substantial. Hu and Hsu [16] proposed image watermarking with DWT-SVD-DCT features to guard against compression. Mishra et al. [17] applied a firefly algorithm based on DWT and SVD for watermarking, but the encryption is time consuming. The above SVD-based watermarking may have good robustness against JPEG compression; however, the embedded watermark becomes vulnerable because all pixels would be changed when one singular value is modified. This paper presents an efficient digital image watermarking method based on the fast DCT algorithm for real-time application. It is shown that the watermarked image quality, robustness, and computation loading can all be improved.

2. Embedding Watermark by the Fast DCT Algorithm

Consider an 8-bit gray-level original image \mathbf{S} and an image watermark \mathbf{W} ,

$$\begin{aligned} \mathbf{S} &= \{s(i, j), 0 \leq i, j \leq N - 1, 0 \leq s(i, j) \leq 255\}, \\ \mathbf{W} &= \{w(p, q), 0 \leq p, q \leq M - 1, 0 \leq w(p, q) \leq 255\}, \end{aligned} \quad (1)$$

where M and N are integers, $M < N$. The original image \mathbf{S} is divided into nonoverlapping pixel blocks of size 8×8 and then transformed into frequency domain. Each block is processed independently by two-dimensional (2D) DCT into matrix \mathbf{F} . The fast DCT algorithm [18] is applied to decompose the 2D DCT into a pair of 1D DCTs for computation efficiency. For a digital image \mathbf{S} , the 2D DCT matrix \mathbf{F} in frequency domain and its inverse transform back to spatial domain \mathbf{S}' can be written in matrix form by

$$\mathbf{F} = \mathbf{M} \mathbf{S} \mathbf{M}^T \quad (2a)$$

$$\mathbf{S}' = \mathbf{M}^T \mathbf{F} \mathbf{M}, \quad (2b)$$

where matrix $\mathbf{M} = \{m(u, v)\}$ with $m(u, v)$ being the elements in the u th row and v th column

$$m(u, v) = \begin{cases} \frac{1}{2\sqrt{2}}, & u = 0 \cup 0 \leq v \leq 7, \\ \frac{1}{2} \cos \frac{(2v+1)u\pi}{16}, & 1 \leq u \leq 7 \cup 0 \leq v \leq 7. \end{cases} \quad (3)$$

The 2D spatial data matrix \mathbf{S}' in (2b) is the linear combination of the base images, which are obtained by the outer product of the column and row vectors of \mathbf{M} . By exploiting the redundancy in DCT coefficients, it can be shown that the fast DCT algorithm can reduce the complexity of a 2D DCT of 8×8 block from \mathbf{S} to \mathbf{F} in (2a) in only 24 multiplications, and the inverse transform from \mathbf{F} to spatial domain \mathbf{S}' in (2b) is simply the transpose of the DCT matrix \mathbf{M} [18].

Watermarking is performed first by decomposing a true color image into RGB components, and the watermark should be embedded in the main composition of the original image to maintain image quality after watermarking. For example, an image rich in blue shall be watermarked with gray-level blue watermarks. Due to the lower sensitivity of human eyes to blue light, it is recommended to embed watermark in blue component if one cannot determine the main composition of the image. The image embedding process of the watermarking method as shown in Figure 1 includes (1) computation optimization by the fast DCT algorithm and its inverse transform, (2) selection of the middle frequency coefficients, (3) permutation of the watermark, and (4) quantization (normalization) of the watermark and the host image. A spatial domain image \mathbf{S} is processed by the fast DCT algorithm into spectral domain \mathbf{F} . Intuitively a watermark should be embedded into the higher frequency coefficients in DCT for better perceptual invisibility. However, the image "energy" is concentrated in the low frequency coefficients (e.g., the 0th~5th coefficients in Figure 2) so that the information embedded (hidden) in the high frequency coefficients (the 28th~63th coefficients) may be vulnerable to data processing. To maintain the quality of a host image after watermarking and to improve the robustness of a watermarked image against data processing, it is recommended to embed the watermark in some of the middle frequency coefficients (the 6th~27th coefficients) as shown in the light color region of Figure 2. Because the host image of size $N \times N$ and the watermark $M \times M$ are divided into nonoverlapping pixel blocks of size 8×8 , the number of the middle frequency coefficients selected for embedding the watermark is $(8M/N)^2$ such that the sizes of the host image and the watermark are compatible. For an 8-bit original image of size 256×256 ($N = 256$) and 8-bit digital watermark of size 64×64 ($M = 64$), 4 middle frequency coefficients in each of the 8×8 blocks from the host image are needed to embed the watermark. The image block composed of these middle frequency coefficients is denoted as \mathbf{F}_m in frequency domain.

A random number generator is applied to simultaneously generate two vectors to permute the watermark \mathbf{W} into \mathbf{W}_p in spatial domain. These vectors with elements in $[1, M]$ representing the new coordinates (x, y) serve as a key to watermark decoding. The permuted watermark is divided into nonoverlapping 4×4 blocks and processed by the fast DCT algorithm into \mathbf{W}_f in spectral domain. Quantization is required to normalize the image intensity of the host image and watermark thereby increasing the quality of watermarked image and reducing the watermark memory size. The process is to scale the watermark \mathbf{W}_f into \mathbf{W}_q by the host image property, represented by the average of the middle frequency

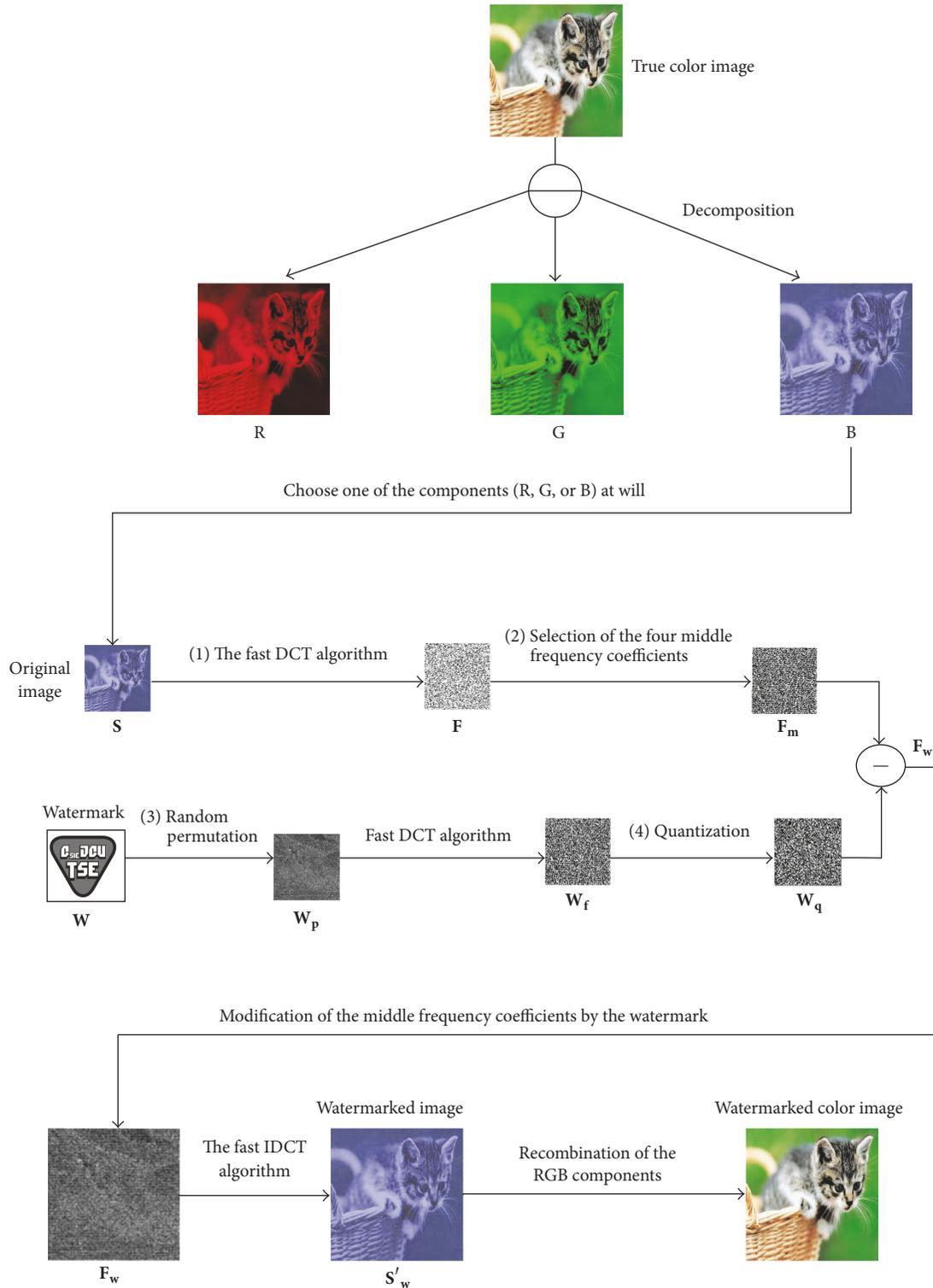


FIGURE 1: The watermarking embedding process that integrates (1) the fast DCT algorithm, (2) selection of the four middle frequency coefficients, (3) random permutation of the watermark, and (4) normalization by quantization.

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

FIGURE 2: Distribution of coefficients in DCT, (a) low frequency (heavy color), (b) middle frequency (light color), and (c) high frequency (white).

coefficients, denoted as F_{ave} . The element of quantization matrix $\mathbf{Q} = \{Q(a, b)\}$ is defined

$$Q(a, b) = \frac{16}{M^2 F_{ave}} \left(\sum_{i=0}^{M/4-1} \sum_{j=0}^{M/4-1} \mathbf{W}_f(a + 4i, b + 4j) \right), \quad (4)$$

such that \mathbf{W}_f is scaled element by element or block by block into \mathbf{W}_q . Note that the watermark \mathbf{W}_q now contains the host image information F_{ave} so the impact on image quality after watermarking will be minimized. When embedding the watermark image, the four coefficients selected in \mathbf{F}_m will be substituted by the corresponding element in \mathbf{W}_q into \mathbf{F}_w . The final step is inverse transform of \mathbf{F}_w by the fast DCT algorithm in (2b) into watermarked image \mathbf{S}'_w in spectral domain. In summary, the parameters in the embedding process are the four middle frequency coefficients, two random vectors, and one quantization matrix for normalization. All are to be stored as keys for later decoding process.

A quantitative measurement for evaluating the fidelity of the watermarked image \mathbf{S}'_w is the peak to signal noise ratio

$$\text{PSNR} = 10 \log_{10} \left(\frac{255^2}{\text{MSE}} \right), \quad (5)$$

where MSE is the mean-square error. For an image of block size 8×8 , its MSE is

$$\text{MSE} = \frac{1}{64} \sum_{i=0}^7 \sum_{j=0}^7 (s(i, j) - s'_w(i, j))^2, \quad (6)$$

where $s(i, j)$ and $s'_w(i, j)$ represent the pixel value in the original image \mathbf{S} and watermarked image \mathbf{S}'_w , respectively. The higher the PSNR, the better the image quality. In general, human visual system cannot differentiate a watermarked image from the host image when $\text{PSNR} \geq 30$ dB.

3. Decoding Watermark

The step of decoding process as shown in Figure 3 is reverse to the embedding process. A watermarked image is first

divided into nonoverlapped blocks of size 8×8 , and each block is processed by the fast DCT algorithm. The middle frequency coefficients in each block are extracted by using the key in the embedding process. Inverse quantization of the extracted watermark is by matrix $\mathbf{Q} = \{Q(a, b)\}$ from the key in embedding process. Inverse DCT can then be applied to obtain the watermark image in spatial domain. Reverse permutation by using the key of the two random vectors is to extract the watermark embedded in the image, denoted as \mathbf{W}' . Note that no original image is required in decoding a watermarked image.

The normalized correlation (NC) is defined to evaluate the fidelity of the extracted watermark

$$\text{NC} = \frac{\sum_{i=0}^{M-1} \sum_{j=0}^{M-1} w(i, j) w'(i, j)}{\sum_{i=0}^{M-1} \sum_{j=0}^{M-1} (w(i, j))^2}. \quad (7)$$

This is to represent the cross-correlation of the reference and extracted watermarks. If the extracted watermark $w'(i, j)$ is the same as the reference (registered) watermark image $w(i, j)$, then $\text{NC} = 1.0$. Note that NC is a quantitative measurement; it is not necessarily related to image quality. $\text{NC} = 1.0$ indicates that the watermarking method can extract the embedded watermark faithfully. In summary, the steps of decoding process are (1) selection of the RGB component of the watermarked image, (2) computation by the fast DCT algorithm, (3) extraction of the middle frequency coefficients, and (4) inverse quantization, inverse DCT, and reverse permutation.

4. Watermarking Implementation

4.1. PC-Based Watermarking. In order to validate the proposed watermarking method, experiments are conducted on computer (CPU: AMD AM3 Athlon II X4-640, RAM: DDR3 4G-1333, and VGA: MSI R5770 HAWK 1 G GDDR5) in Windows® 8 with Matlab® 2012. The digital image can be stored in Windows Bitmap (BMP), Graphics Interchange Format (GIF), Joint Photographic Experts Group (JPEG), or Tagged Image File Format (TIFF). The use of color image processing is motivated by two factors: (1) color is a powerful descriptor that often simplifies object identification and extraction from a scene and (2) human can distinguish thousands of color shades and intensities, compared to about only two dozen shades of gray. From the absorption characteristics of human eyes, colors are seen as variable combinations of the primary colors: red (R), green (G), and blue (B). In practice, a color model is a specification of coordinate system within which the color is represented by a single point.

By the image watermarking method illustrated in Figure 1, the three components of an image are stored in data array that defines RGB components for each individual pixel. Embedding a watermark is dependent on the main composition of the color image. An original image of size 256×256 and a digital watermark of 8-bit gray-level of size 64×64 as shown in Table 2 are to demonstrate the watermarking method. The original image divided into nonoverlap blocks of size 8×8 shall select 4 coefficients in the middle frequency range after DCT. The middle range coefficients in coordinates

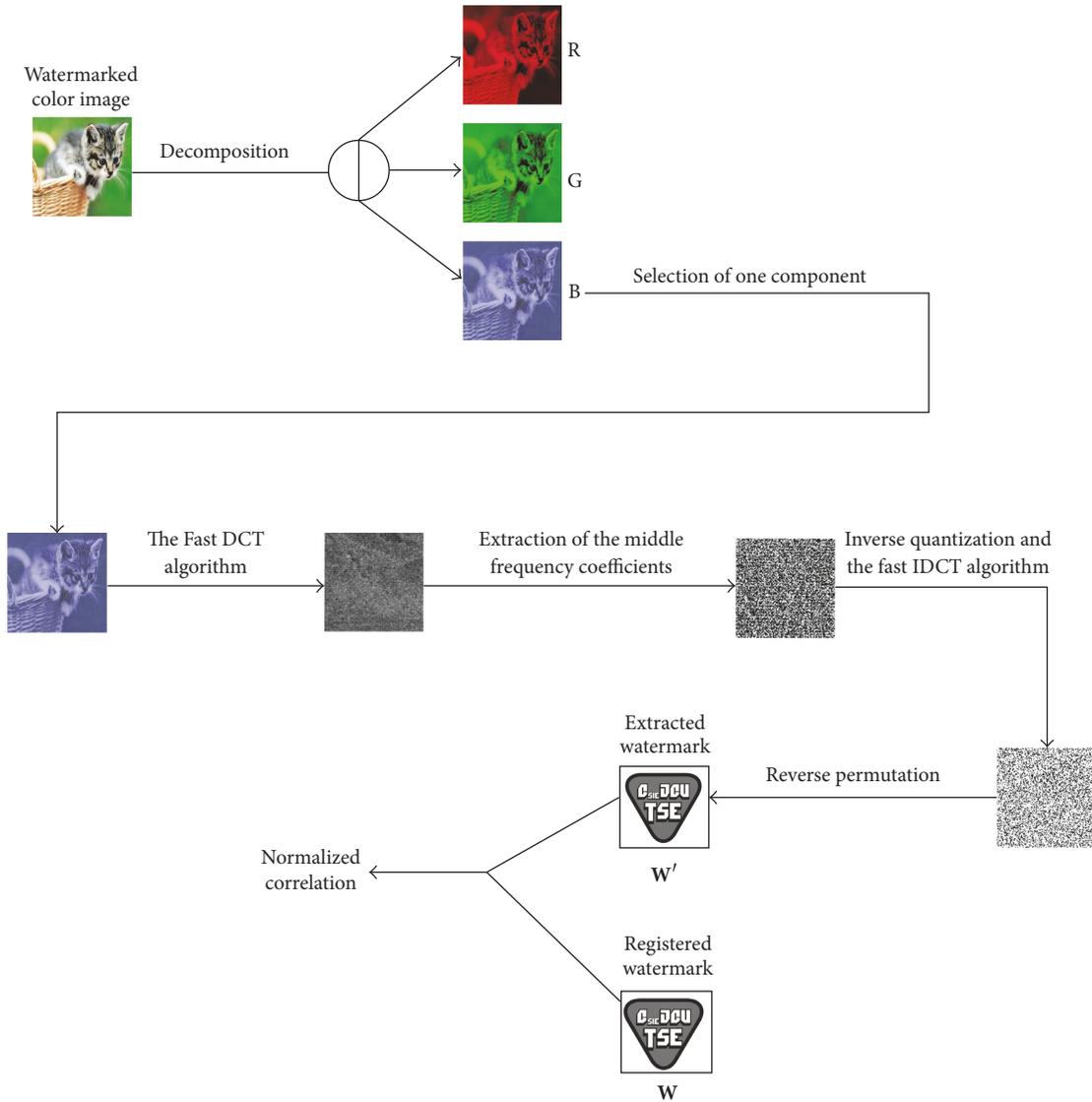


FIGURE 3: Watermark decoding process by the fast DCT algorithm with the key of (1) the four middle frequency coefficients, (2) the random permutation vectors, and (3) the quantization matrix.

(1, 4), (4, 2), (3, 3), and (1, 5), corresponding to block numbers 6, 11, 12, and 14, are selected to embed the image watermark. Table 1 also lists the two random permutation vectors and the quantization matrix Q of (4). The experiment result is shown in Table 2, where $PSNR = 31.256$ dB and $NC = 1.0$, and the watermark embedding/decoding can be completed within 1.6 seconds. The result indicates that the proposed watermarking method can maintain the image quality and the embedding/decoding process is efficient.

Experiments are also conducted to validate the robustness of the watermarking method against image cropping and data compression. Table 3 shows that, after 5% cropping of watermarked image, $NC = 0.931$. For 15% and 25% cropping, the watermarked image yields $NC = 0.777$ and 0.707 , respectively, and the quality of the retrieved watermark remains satisfactory. These results show that selection of the four middle frequency coefficients for embedding watermark is

robust to data compression. Although some distinct features in the retrieved watermark are lost after 25% cropping or 1/4 data compression, the method can still correctly extract and identify the watermark.

Experiments of color image embedding in different components are also conducted. For the color image in Table 4, the watermark can be embedded into R, G, or B component in the middle frequency coefficients. The watermarking in any of the components can be completed within 2 seconds, with $PSNR$ higher than 41 dB.

4.2. DSP-Based Watermarking. With rapid development of digital signal processor (DSP), implementation of real-time processing is feasible. The watermarking method is most effective when implemented in digital signal processor (TMS320C6701), which includes eight functional units CPU, instruction packing to reduce code size, 40-bit arithmetic

TABLE 1: The key in watermark embedding/decoding process includes (1) four frequency coefficients, (2) two random permutation vectors, and (3) a quantization matrix.

(1) Frequency coefficients	6, 11, 12, 14 [44, 19, 34, 9, 62, 7, 50, 57, 39, 20, 40, 30, 10, 49, 53, 27, 52, 43, 6, 13, 4, 26, 23, 47, 55, 36, 38, 64, 58, 61, 21, 14, 45, 41, 8, 59, 12, 18, 24, 31, 1, 32, 11, 29, 56, 48, 42, 15, 5, 46, 63, 51, 25, 37, 28, 2, 54, 21, 60, 3, 16, 33, 35, 17]
(2) Random permutation vectors	[57, 47, 39, 40, 3, 32, 15, 50, 37, 64, 56, 36, 23, 11, 63, 42, 9, 51, 46, 10, 2, 30, 61, 60, 55, 41, 43, 59, 26, 25, 27, 4, 35, 45, 20, 53, 7, 12, 8, 31, 21, 44, 62, 5, 33, 16, 54, 6, 22, 18, 14, 17, 13, 1, 19, 49, 29, 52, 58, 38, 34, 28, 48, 24]
(3) Quantization matrix Q	$\begin{bmatrix} 988.2351 & -6.4521 & 1.1211 & -1.1129 \\ 6.3722 & 3.6798 & -0.6523 & -3.0758 \\ -6.3219 & 1.6128 & -0.9834 & -1.7245 \\ 3.4911 & 1.1675 & -1.5218 & 2.3147 \end{bmatrix}$

TABLE 2: Validation of the watermarking method, where the watermarked image remains in excellent quality with PSNR ≥ 30 dB and the retrieved watermark with perfect NC = 1.0.

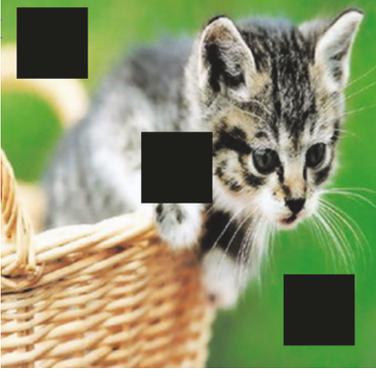
The original image of size 256×256 and the watermark of size 64×64 .		
The watermarked image using middle frequency coefficients (6, 11, 12, and 14 in Figure 2) with PSNR = 31.256 dB, and NC = 1.0.		

options for precision, hardware support for floating point operations, and large on-chip RAM. The C/C++ compiler is able to perform optimization ($n = 0, 1, 2,$ and 3) in different level of clock cycles and code size. $n = 0$: the lowest level optimization provides the operations of performing loop rotation, allocating variables to registers and simplifying expressions and statements. $n = 1$: the first level performs the optimization on constant propagation and unused assignments. $n = 2$: the second level performs the optimization on software pipelining, unused global assignments, loop unrolling, and incremented pointer. $n = 3$: the highest

level performs the optimization by simplifying functions with return values never used, removing all functions never called. $n = 0$ and 1 level can efficiently reduce the code size, while $n = 2$ and 3 enhance the execution speed with larger code size. Higher levels generally result in the smallest code size.

The code of the proposed watermarking method is written in C language. After compiler, assembler, and linker, the development software generates a machine file and stores it in DSP. The original image, the watermark, and the decoding key are loaded and stored in SDRAM. As a safety protection in communication, the decoding key as listed in Table 1

TABLE 3: Robustness of the watermarking method against image cropping and data compression.

Image attack on watermarked image	Retrieved watermark
5% cropping at the upper right area	  <p data-bbox="1297 534 1394 555">NC = 0.931</p>
15% cropping from the upper right to the lower left	  <p data-bbox="1297 921 1394 942">NC = 0.777</p>
1/4 data compression on the watermarked image	  <p data-bbox="1297 1293 1394 1315">NC = 0.707</p>

is the protocol between the deliverer and the receiver. The computation time and code size in DSP at different levels of optimization are listed in Table 5. The code size and computation time are 26 KB and 0.89 seconds in the lowest level ($n = 0$). The computation time in first level ($n = 1$) reduces to 0.61 seconds with code size 29 KB. In the second and third ($n = 2$ and 3), the computation time reduces to 0.35 and 0.33 seconds, but the code size increases to 43 and 44 KB, respectively. Implementation on DSP is much faster than PC-based operation. The higher the level of optimization, the less the time required. By comparison, implementation using Matlab on PC requires 1.6 seconds' computation time. The experiment starts with the highest optimization ($n = 3$), and the watermarked image and extracted watermark are shown in Table 5 with PSNR = 31.531 dB and NC = 0.951. The calculation uses 57,149,130

clocks, corresponding to 0.33 seconds in 44 KB code size. In conclusion, the watermarking method on DSP is efficient and effective for real-time implementation.

5. Conclusions

- (1) A digital image watermarking method based on the fast DCT algorithm is developed to maintain the image quality and improve image robustness against signal processing. The decoding keys include (i) the four middle frequency coefficients, (ii) two random permutation vectors, and (iii) a quantization matrix for normalizing the watermark and the original image. A watermark with ownership information is embedded into the host image in an unnoticeable way. An index of peak signal to noise ratio (PSNR) is

TABLE 4: Watermarked images with watermark embedded in red, green, or blue component.

Watermark embedded in the red component		PSNR = 43.156
Watermark embedded in the green component		PSNR = 41.251
Watermark embedded in the blue component		PSNR = 43.944

employed to measure the quality of the watermarked image, and another, the normalized correlation (NC), is also applied to evaluate the similarity and articulation of the extracted watermark with the reference (registered) watermark. The “blind” watermarking method is shown to be robust to image data cropping, transmission loss, and compression/decompression.

- (2) The watermarking method is first by decomposing a true color image into RGB components such that a

watermark can be embedded in the main composition for better image quality. Due to lower sensitivity of human eyes to blue light, it is recommended to embed watermark in blue component if one cannot determine the main composition. The image watermarking method is applicable to images in Windows Bitmap (BMP), Graphics Interchange Format (GIF), Joint Photographic Experts Group (JPEG), or Tagged Image File (TIF) format. To maintain image quality

TABLE 5: Processing time comparison in DSP and in PC of the watermarking method.

(a)			
Optimization level	Clocks (cycle)	Time (s)	Code size (KB)
0	154,645,548	0.89	26
1	112,044,218	0.61	29
2	59,152,281	0.35	43
3	57,149,130	0.33	44
PC with Matlab	N.A.	1.60	N.A.

(b)		
Optimization level $n = 3$	Watermarked image	Retrieved watermark
<p>Four middle frequency coefficients for embedding watermark</p>		 <p>NC = 0.951</p>
PSNR = 31.531		

after watermarking and to improve the robustness of a watermarked image, it is recommended to embed the watermark in some of the middle frequency coefficients (the 6th~27th coefficients) in DCT as shown in the light color region of Figure 2. Selection of the frequency coefficients should be based on the desired watermarked image quality and robustness. For an 8-bit original image of size 256×256 ($N = 256$) and 8-bit digital watermark of size 64×64 ($M = 64$), 4 middle frequency coefficients of the host image after DCT are needed to embed the watermark.

- (3) Experiments show that PC-based watermarking can be completed within 1.6 seconds and the quality of image remains pristine with PSNR = 31.256 dB and the retrieved watermark NC = 1.0. The method is shown to be robust to image data cropping and compression/decompression. The method is also optimal for real-time applications in DSP, where implementation shows that the embedding and extraction can be completed within 0.33 seconds. The watermarking method is efficient and effective against illegal pirating, destruction, and copy.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

- [1] A. Koz and A. A. Alatan, "Oblivious spatio-temporal watermarking of digital video by exploiting the human visual system," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 3, pp. 326–337, 2008.
- [2] A. Kaur, R. Dhir, and G. Sikka, "A new image steganography based on first component alteration technique," *International Journal of Computer Science and Information Security*, vol. 6, no. 3, pp. 53–56, 2009.
- [3] A. Agarwal, "Security enhancement scheme for image steganography using S-DES technique," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 2, pp. 164–169, 2012.
- [4] V. Tyagi, A. Kumar, R. Patel, S. Tyagi, and S. S. Gangwar, "Image steganography using least significant bit with cryptography," *Journal of Global Research in Computer Science*, vol. 3, no. 3, pp. 53–55, 2012.
- [5] J. Nath, S. Das, S. Agarwal, and A. Nath, "A challenge in hiding encrypted message in LSB and LSB+1 bit positions in various cover files," *International Journal of Computer Applications*, vol. 14, no. 7, pp. 31–35, 2011.
- [6] K. Sara, A. D. Mashallah, and Y. M. Hossein, "A new steganography method based on HIOP (Higher Intensity Of Pixel) algorithm and strassen's matrix multiplication," *Journal of Global Research in Computer Science*, vol. 2, no. 1, p. 12, 2011.
- [7] A. Chadha, N. Satam, R. Sood, and D. Bade, "An efficient method for image and audio steganography using Least Significant Bit (LSB) substitution," *International Journal of Computer Applications*, vol. 77, no. 13, pp. 37–45, 2013.

- [8] H. A. Surrah and I. N. Mohamed, "A new steganography data hiding algorithm," *Journal of Global Research in Computer Science*, vol. 5, no. 6, pp. 1–6, 2014.
- [9] X.-Y. Wang, C.-P. Wang, H.-Y. Yang, and P.-P. Niu, "A robust blind color image watermarking in quaternion Fourier transform domain," *Journal of Systems and Software*, vol. 86, no. 2, pp. 255–277, 2013.
- [10] A. Benoraira, K. Benmahammed, and N. Boucenna, "Blind image watermarking technique based on differential embedding in DWT and DCT domains," *Eurasip Journal on Advances in Signal Processing*, vol. 2015, no. 1, article 55, 2015.
- [11] H. M. Al-Otum and N. A. Samara, "A robust blind color image watermarking based on wavelet-tree bit host difference selection," *Signal Processing*, vol. 90, no. 8, pp. 2498–2512, 2010.
- [12] C.-S. Lu, S.-K. Huang, C.-J. Sze, and H.-Y. M. Liao, "Cocktail watermarking for digital image protection," *IEEE Transactions on Multimedia*, vol. 2, no. 4, pp. 209–224, 2000.
- [13] S. Rawat and B. Raman, "A blind watermarking algorithm based on fractional Fourier transform and visual cryptography," *Signal Processing*, vol. 92, no. 6, pp. 1480–1491, 2012.
- [14] N. M. Makbol and B. E. Khoo, "Robust blind image watermarking scheme based on redundant discrete wavelet transform and singular value decomposition," *AEU—International Journal of Electronics and Communications*, vol. 67, no. 2, pp. 102–112, 2013.
- [15] S. Lagzian, M. Soryani, and M. Fathy, "A new robust watermarking scheme based on RDWT-SVD," *International Journal of Intelligent Information Processing*, vol. 2, no. 1, pp. 22–29, 2011.
- [16] H.-T. Hu and L.-Y. Hsu, "Exploring DWT-SVD-DCT feature parameters for robust multiple watermarking against JPEG and JPEG2000 compression," *Computers and Electrical Engineering*, vol. 41, pp. 52–63, 2015.
- [17] A. Mishra, C. Agarwal, A. Sharma, and P. Bedi, "Optimized gray-scale image watermarking using DWT-SVD and firefly algorithm," *Expert Systems with Applications*, vol. 41, no. 17, pp. 7858–7867, 2014.
- [18] S. E. Tsai and S. M. Yang, "A fast DCT algorithm for watermarking in digital signal processor," *Mathematical Problems in Engineering*, vol. 2017, Article ID 7401845, 7 pages, 2017.



Hindawi

Submit your manuscripts at
<https://www.hindawi.com>

