Hindawi Publishing Corporation
The Scientific World Journal
Volume 2014, Article ID 454867, 14 pages
http://dx.doi.org/10.1155/2014/454867

Research Article

Combining Digital Watermarking and Fingerprinting Techniques to Identify Copyrights for Color Images

Shang-Lin Hsieh,[1] Chun-Che Chen,[1,2] and Wen-Shan Shen[1]

[1] *Department of Computer Science and Engineering, Tatung University, Taipei 10452, Taiwan*
[2] *Taipei College of Maritime Technology, New Taipei 25172, Taiwan*

Correspondence should be addressed to Shang-Lin Hsieh; sunny6677@gmail.com

Received 15 March 2014; Accepted 20 May 2014; Published 8 July 2014

Academic Editor: Fei Yu

This paper presents a copyright identification scheme for color images that takes advantage of the complementary nature of watermarking and fingerprinting. It utilizes an authentication logo and the extracted features of the host image to generate a fingerprint, which is then stored in a database and also embedded in the host image to produce a watermarked image. When a dispute over the copyright of a suspect image occurs, the image is first processed by watermarking. If the watermark can be retrieved from the suspect image, the copyright can then be confirmed; otherwise, the watermark then serves as the fingerprint and is processed by fingerprinting. If a match in the fingerprint database is found, then the suspect image will be considered a duplicated one. Because the proposed scheme utilizes both watermarking and fingerprinting, it is more robust than those that only adopt watermarking, and it can also obtain the preliminary result more quickly than those that only utilize fingerprinting. The experimental results show that when the watermarked image suffers slight attacks, watermarking alone is enough to identify the copyright. The results also show that when the watermarked image suffers heavy attacks that render watermarking incompetent, fingerprinting can successfully identify the copyright, hence demonstrating the effectiveness of the proposed scheme.

1. Introduction

Many researchers [1–17] have been engaged in finding the solution to protecting copyrights of digital images, which may be duplicated and distributed over the Internet without the authors' permission. Generally speaking, there are two approaches to discovering image copyright infringement. One is watermarking [1–11] and the other is fingerprinting [12–17]. The main idea of watermarking is to embed a piece of information (i.e., watermark) in the host image. If a similar watermark can be retrieved from a suspect image, it is then considered a duplicated one. On the other hand, the principle of fingerprinting is to extract unique features (i.e., fingerprints) from both the host image and the suspect one for comparison. If their fingerprints are similar, the ownership of the image can then be confirmed.

There are some general considerations on the two techniques, including the processing time and robustness. In terms of processing time, watermarking is more efficient because fingerprinting needs extra time to compare the image's fingerprint with those stored in the database. If the database is large, it will be very time consuming. On the other hand, fingerprinting is generally more robust [18, 19] because when a watermarked image suffers some image processing operations that modify the content of the image, the embedded watermark will usually be damaged or even destroyed. On the contrary, since normal image processing does not destroy the features of an image, the fingerprint of the image can therefore be preserved. In summary, fingerprinting is more robust whereas watermarking is more efficient. If the complementary natures of two approaches can be utilized properly, a robust and efficient scheme can then be developed to identify copyrights.

This paper proposes a novel scheme that combines the two techniques to identify copyrights for color images. The proposed scheme generates from the image a fingerprint, which also serves as the watermark. The watermark is then embedded in the host image to produce a watermarked

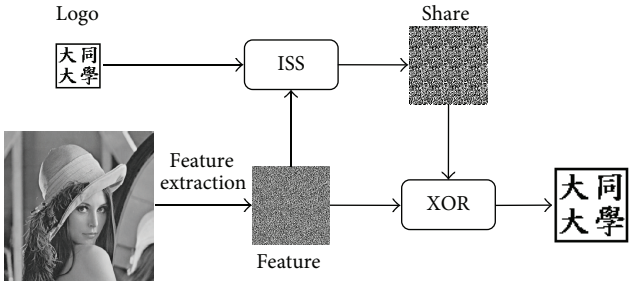

Figure 1: The process of the ISS.

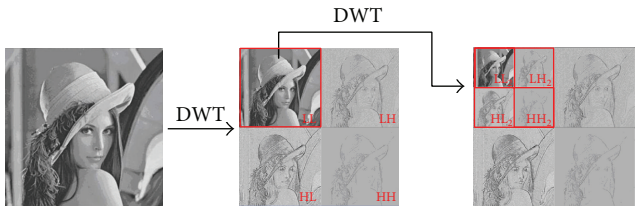

Figure 2: 2D DWT.

image. When there is a dispute over the copyright of a suspect image, the suspect image will first be processed by watermarking, which tries to retrieve the watermark from the suspect image. If the watermark is identified, the copyright is confirmed at this stage; otherwise, the image will then be processed by fingerprinting, which utilizes the retrieved watermark as the fingerprint and compares it with those stored in the database. If a match is found, then the suspect image will be considered a duplicated one.

2. Related Background

The proposed scheme utilizes a special technique called image secret sharing (ISS), whose details can be found in the paper [20] we published in 2008. The following briefly describes the main idea of the technique utilized by the proposed scheme.

The ISS generates a share image from two images. In the proposed scheme, the two images are the logo image and the feature image (as depicted in Figure 1). The logo image can be any identifiable image. The feature image is generated from the input image as follows. First, the input image is split into nonoverlapping 8×8 blocks. Then, the 2D DWT is applied to each block to generate four subbands, LL_2, LH_2, HL_2, and HH_2. An example of 2D DWT is shown in Figure 2. Finally, the coefficients in the LL_2 subband of each DWT block are used to generate the feature image. The ISS then generates a share image from the logo image and feature image. The share image will be used as the fingerprint of the input image by the scheme. The share image also serves as the watermark to be embedded in the host image. The benefit of the ISS scheme is that performing the XOR operation on the feature image and the share image will restore the logo image, which can then be used to identify the copyright.

3. The Proposed Copyright Identification Scheme

The proposed scheme contains two phases: the *fingerprint and watermarked image generation* phase and the *authentication logo detection* phase. The former phase extracts features from the host image, which, along with a logo image, is used to generate the fingerprint. The fingerprint also serves as the watermark, and the phase embeds it in the host image to produce a watermarked image. On the other hand, the latter phase extracts features and retrieves the watermark from the suspect image. The extracted features and the retrieved watermark are utilized to restore the logo image, which is used to identify the copyright. If it fails, the retrieved watermark then serves as the fingerprint and is compared with those in the database to determine if the suspect image is a duplicated one.

The *fingerprint and watermarked image generation* phase (shown in Figure 3) works as follows. In the beginning, *feature extraction* extracts the features of the host image and then *logo scrambling* disarranges the authentication logo to a scrambled logo image. After that, *fingerprint generation* takes as input the extracted features and the scrambled logo to generate the fingerprint. Finally, the fingerprint serves as a watermark and is embedded in the host image, which becomes a watermarked image. The fingerprint is also stored in a database for later use in the next phase.

The *authentication logo detection* phase (shown in Figure 3) checks the watermark first and, if necessary, the fingerprint next. In the beginning, *watermark retrieval* regains the watermark from the suspect image. Next, the features of the suspect image are extracted by *feature extraction*. After that, *logo restoration* takes as input the retrieved watermark (the expected fingerprint of the suspect image) and the extracted features to recover and rearrange the scrambled logo to restore the authentication logo. The phase ends if the accuracy rate of the restored logo determined by *logo comparison* is high enough; otherwise, the process proceeds to retrieve the next available fingerprint from the database and then returns to *logo restoration*, which takes as input the retrieved fingerprint instead of the extracted watermark. The phase restores the logo from the retrieved fingerprint as well as the extracted features and proceeds to *logo comparison*. The looping process continues until the authentication logo is discovered or no fingerprint is available.

3.1. Fingerprint and Watermarked Image Generation Phase.

The following paragraphs detail the stages in the *fingerprint* and *watermarked image generation* phase, including *feature extraction, logo scrambling, fingerprint generation,* and *watermark embedding.*

3.1.1. Feature Extraction. The *feature extraction* stage takes a color image as input and then extracts its features. The stage has two substages, *sampling* and *feature generation*. During *sampling*, the stage first transforms the RGB image to the YCbCr color space [21, 22]. Then, it partitions each of the three channels into several nonoverlapping blocks of size 8

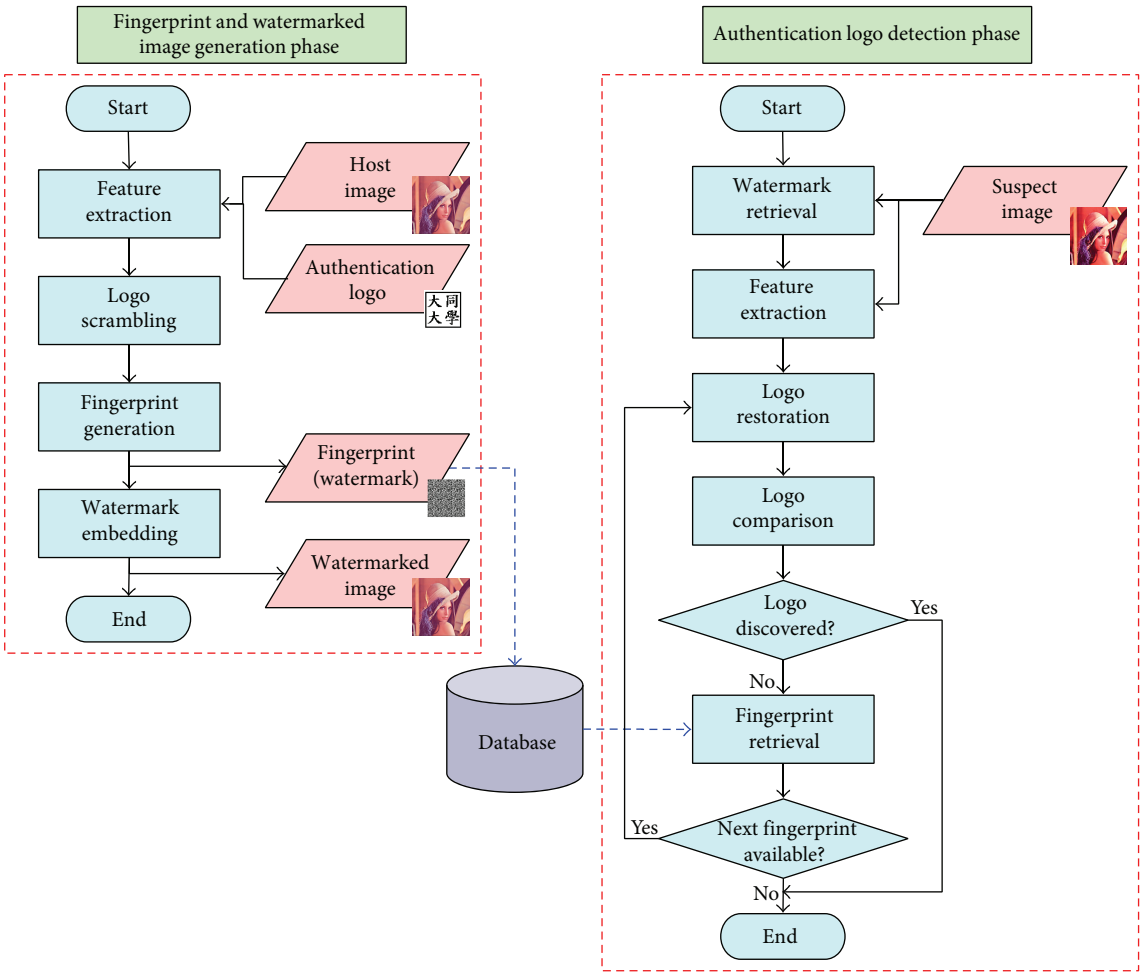

Figure 3: The phases of the proposed scheme.

× 8 (hence, a color image of size $N \times N$ will result in $3 \times N/8 \times N/8$ nonoverlapping blocks). After partition, for each row of the corresponding blocks in the three channels, the stage takes the first four samples from the Y channel, the next two from the Cb, and the last two from the Cr (as depicted in Figure 4) to generate new packed blocks, each of size 8×8.

After *sampling*, the stage enters its second substage, *feature generation*. For each of the packed blocks, the stage applies 2D DWT to the block, resulting in four coefficients in the LL_2 subband. Next, the stage computes the average (denoted by A) of the four coefficients and then obtains a feature type T according to the relationship of the four coefficients and the average A as expressed in (1). Consider

$$T = \begin{cases} 1, & \text{if only one coefficient is smaller than } A. \\ 2, & \text{if only two coefficients are smaller than } A. \\ 3, & \text{if only one coefficient is greater than } A. \\ 4, & \text{if all of the four coefficients are the same and hence all equal } A. \end{cases} \quad (1)$$

According to T, A, and the mapping table shown in Table 1, a feature share (called FT-share) of size 2×2 is determined for each block. The FT-shares represent the features of the input color image. They are assembled to form the feature image.

The steps of the *feature extraction* stage are listed in Algorithm 1.

3.1.2. Logo Scrambling. In order to disperse the intensity of attacks, the proposed scheme adopts Torus automorphism [23] to scramble the authentication logo. The stage uses a predetermined key, k, and the following equation to scramble the logo. Consider

$$\begin{pmatrix} x_t \\ y_t \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ k & k+1 \end{pmatrix} \begin{pmatrix} x_{t-1} \\ y_{t-1} \end{pmatrix} \bmod T, \quad (2)$$

where (x_t, y_t) are the coordinates in state t and T is the coordinate size of the given image. Figure 5 shows an example of the authentication logo scrambled four times with $k = 1$, $T = 128$.

3.1.3. Fingerprint Generation. The proposed scheme uses ISS (mentioned in Section 2) to generate the fingerprint. It determines a FP-share for each FT-share (generated in *feature*

Input: A color image H ($N \times N$).
Output: A feature image FT ($N/8 \times N/8$).
Convert H to the YCbCr color space
Partition each of the Y, Cb, and Cr channels into $N/8 \times N/8$ non-overlapping blocks of size 8×8
For each corresponding block of the three channels
    Take the first 4 samples from the Y channel, the next 2 from the Cb, and last 2 from the Cr as depicted in Figure 4 to form a packed block of size 8×8
End For
Apply 2D-DWT to each packed block to obtain $N/8 \times N/8$ LL_2 blocks of size 2×2
For each of the LL_2 blocks
    Compute the average A of the four coefficients
    Obtain the feature type T according to (1)
    Determine the FT-share according to T, A, and Table 1
End For
Assemble the FT-shares to form the feature image FT

Algorithm 1: Feature extraction.

Table 1: The mapping table.

Feature type T	Mean value position	White logo pixel		FT-share XOR FP-share	Black logo pixel		FT-share XOR FP-share
		FT-share	FP-share		FT-share	FP-share	
1	$a < A < b, c, d$						
	$b < A < a, c, d$						
	$c < A < a, b, d$						
	$d < A < a, b, c$						
2	$a, b \le A < c, d$						
	$c, d \le A < a, b$						
	$a, d \le A < b, c$						
	$b, c \le A < a, d$						
	$a, c \le A < b, d$						
	$b, d \le A < a, c$						
3	$b, c, d \le A < a$						
	$a, c, d \le A < b$						
	$a, b, d \le A < c$						
	$a, b, c \le A < d$						
4	$A = a = b = c = d$						
$\begin{array}{\|c\|c\|}\hline a & b \\ \hline c & d \\ \hline\end{array}$	The coefficients of the LL_2 subband (a is at the top left position, b the top right, c the bottom left, and d the bottom right)						

FT-share: feature share; FP-share: fingerprint share.

Table 2: Partial table of Table 1.

Feature type T	Mean value position	White logo pixel		FT-share XOR FP-share	Black logo pixel		FT-share XOR FP-share
		FT-share	FP-share		FT-share	FP-share	
2	$b, c \le A < a, d$						

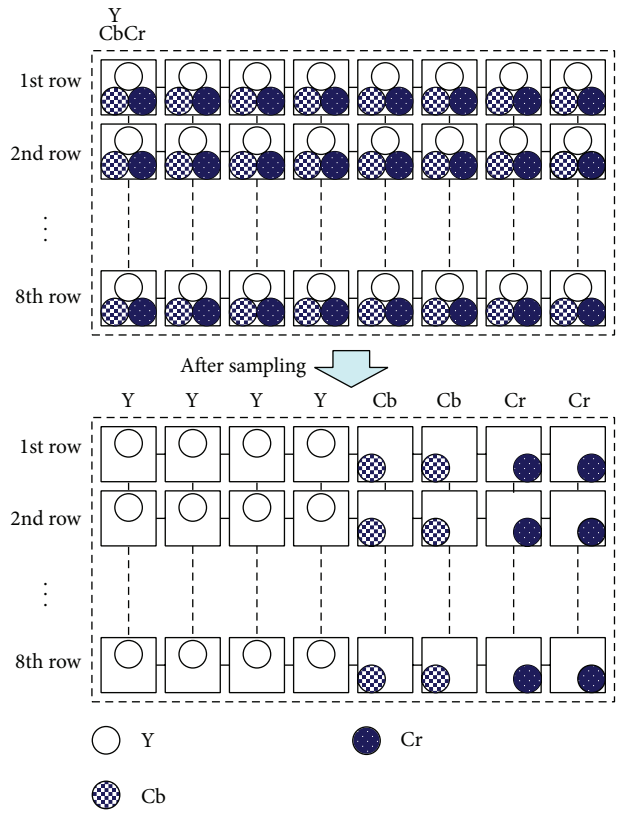

Figure 4: Illustration of *sampling*.

extraction) according to the color of the corresponding pixel in the scrambled logo by looking Table 1 up. For example, if the feature type is 2 and the FT-share is the same as the one in Table 2, then the FP-share will be either one of the two FP-shares in Table 2 according to the color of the corresponding pixel in the scrambled logo. After every FP-share is determined, the stage gathers all the FP-shares to form the fingerprint image. The fingerprint also serves as the watermark in the next stage.

3.1.4. Watermark Embedding. The *watermark embedding* stage uses the resulting fingerprint image as a watermark and then embeds it in the host image. Generally speaking, for the RGB color space, the human visual system is more sensitive to the G channel than to the other two [22–28]. Therefore, the proposed scheme embeds the watermark into the less sensitive R and B channels of the host image. To be more specific, there are four areas: two in the R channel (I_1 and I_2) and two in the B channel (I_3 and I_4), used for *watermark embedding* (see Figure 6). The stage applies the 2D DWT to the R and B channels of the host image and next applies the 1D DWT to the resulting HL_2 and LH_2 to obtain the four blocks, I_1, I_2, I_3, and I_4.

The proposed scheme embeds the watermark into the image by adjusting the coefficients in the I_1 to I_4 blocks according to a predefined M. The value of M affects the robustness and the quality. The larger the M is, the more robust the embedded watermark is, but the worse the quality of the watermarked image is. Figure 7 illustrates the adjustment of the coefficients. If the watermark bit is 1 and the current coefficient is between $M \times (i-1)$ and $M \times i$, then the coefficient will be adjusted to be $M \times (i-1) - M/4$ or $M \times i - M/4$, whichever is closer to the current coefficient. On the other hand, the adjusted coefficient will be $M \times (i-1) + M/4$ or $M \times i + M/4$ if the watermark bit is 0.

The embedding process is described as follows. First, the stage uses the current coefficient c and the predefined M to calculate S, sign, and (r_0, r_1) according to (3), (4), and (5), respectively. In the equations, sign indicates that the c is positive or negative; r_0 is the remainder for watermark bit value 0 and, similarly, r_1 for watermark bit value 1. Then, the stage determines the C_Low and C_High according to the value of the watermark bit. If the value is 0, (6) will be used, otherwise, (7). Finally, the stage adjusts the coefficient c to c' according to (8). Consider

$$S = \frac{|c|}{M}, \tag{3}$$

$$\text{sign} = \begin{cases} 1, & \text{if } c \ge 0, \\ -1, & \text{otherwise}, \end{cases} \tag{4}$$

$$(r_0, r_1) = \begin{cases} \left(\frac{M}{4}, \frac{3M}{4}\right), & \text{if } c \ge 0, \\ \left(\frac{3M}{4}, \frac{M}{4}\right), & \text{otherwise}, \end{cases} \tag{5}$$

$$\begin{aligned} C\_\text{Low} &= \text{sign} \times (S \times M + r_0), \\ C\_\text{High} &= \text{sign} \times ((S + \text{sign}) \times M + r_0), \end{aligned} \tag{6}$$

$$\begin{aligned} C\_\text{Low} &= \text{sign} \times ((S - \text{sign}) \times M + r_1), \\ C\_\text{High} &= \text{sign} \times (S \times M + r_1), \end{aligned} \tag{7}$$

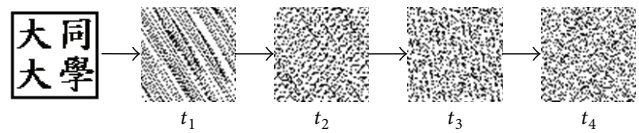

Figure 5: Example of scrambling.

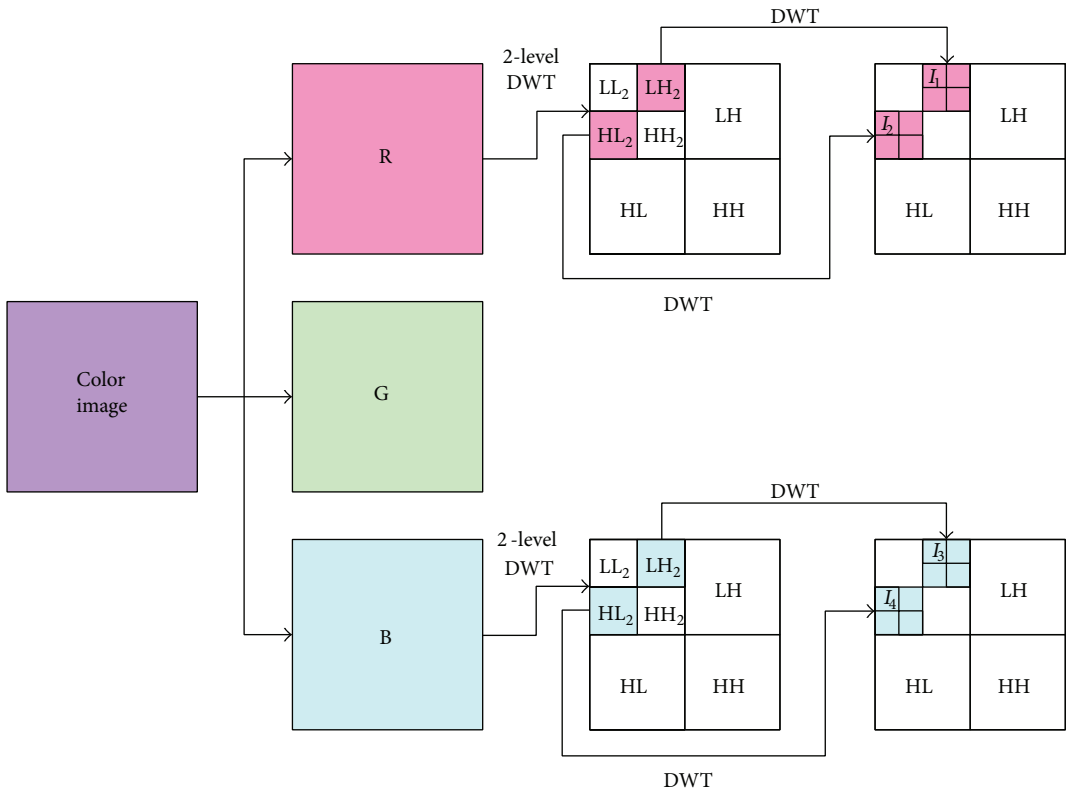

Figure 6: The I_1, I_2, I_3, and I_4 blocks used for *watermark embedding*.

$$c' = \begin{cases} C\_\text{Low}, & \text{if } |C\_\text{Low} - c| \le |C\_\text{High} - c| \\ C\_\text{High}, & \text{otherwise.} \end{cases} \quad (8)$$

The steps of the *watermark embedding stage* are listed in Algorithm 2.

3.2. Authentication Logo Detection Phase. The phase is activated when a dispute over the copyright of a suspect image occurs. The following details the stages in the phase, including *watermark retrieval* and *logo restoration*.

3.2.1. Watermark Retrieval. The stage regains the watermark from the suspect image. First, the stage applies the DWT to the R and B planes of the suspect image in the same way as that in *watermark embedding* to obtain the embedding blocks, I_1 to I_4 (refer to Figure 6). Then, for each coefficient c of the I_1–I_4 blocks, it obtains the watermark bit w according to

$$w = \begin{cases} 1, & \text{if } \left(c \ge 0,\ c \bmod M \ge \dfrac{M}{2}\right), \\ & \text{or } \left(c < 0,\ |c| \bmod M < \dfrac{M}{2}\right), \\ 0, & \text{otherwise,} \end{cases} \quad (9)$$

where M is the same as that in (3).

Finally, the watermark (which is supposed to be the fingerprint of the image) is restored by assembling every w for each coefficient c in the I_1–I_4 blocks.

Algorithm 3 lists the steps of the *watermark retrieval* stage.

3.2.2. Logo Restoration. The stage restores the authentication logo. As shown in Figure 8, it has four substages: *scrambled*

Input: A color image H ($N \times N$) and a fingerprint image FP ($N/4 \times N/4$).
Output: A watermarked image ($N \times N$).
Apply DWT to the R and B channels of H to obtain the four embedding blocks $I_1, \ldots, I_4$
For each coefficient c of the I_1–I_4 blocks and the corresponding pixel of FP
  Compute the *sign*, S, and (r_0, r_1) by (3), (4), and (5), respectively
  Compute the C_Low and C_High by (6) if watermark bit is 0, or (7) if otherwise
  Adjust the coefficient to c' by (8)
End For
Perform inverse DWT to produce the watermarked image

Algorithm 2: Watermark embedding.

Input: A suspect color image S ($N \times N$).
Output: A fingerprint (watermarked) image FP′ ($N/4 \times N/4$).
Apply DWT to the R and B channels of S and obtain the embedding blocks I_1–I_4
For each coefficient c of the I_1–I_4 blocks
  Obtain w according to (9)
End For
Assemble every watermark bit w to restore the fingerprint image FP′

Algorithm 3: Watermark retrieval.

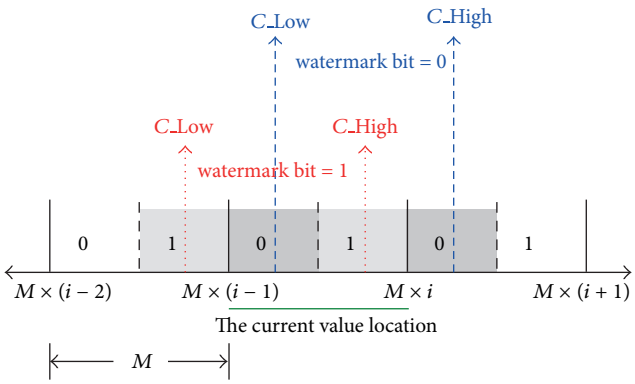

Figure 7: Coefficient adjustment.

logo recovery, *logo unscrambling*, *logo enhancement*, and *logo resizing*.

Scrambled Logo Recovery. After retrieving the fingerprint image (i.e., watermark) by *watermark retrieval* and extracting the feature image by *feature extraction*, the substage performs XOR operation on each pixel of the fingerprint image and the corresponding pixel of the feature image to retrieve the scrambled logo. Because both of the two images are black-and-white, each pixel is either 0 (black) or 1 (white). Therefore, the substage simply performs bitwise XOR operation on the two images and obtains a scrambled logo.

Logo Unscrambling. As mentioned in Section 3.1.2, the logo was scrambled before it is used to generate fingerprint in the *fingerprint and watermarked image generation* phase. The substage adopts (10), which is the inverse equation of (2), to rearrange the scrambled logo and restore the logo. Consider

$$\begin{pmatrix} x_t \\ y_t \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ k & k+1 \end{pmatrix}^{-1} \begin{pmatrix} x_{t-1} \\ y_{t-1} \end{pmatrix} \bmod T. \tag{10}$$

Logo Enhancement. The substage enhances the restored logo by erosion and dilation. Erosion removes pixels on object boundaries in an image and therefore can be used to remove smaller islands in the image; dilation, on the other hand, adds pixels to the boundaries of objects in an image and hence can be used to remove bright areas from the image. The substage performs erosion followed by dilation once, which is illustrated in Figure 9.

Erosion is performed by resetting each of the pixels according to (11). That is, if one of the neighbors is black, then the current pixel will be set to black; otherwise, it will be set to white. Let $L(x, y)$ be the current pixel value to be determined and $L(x+1, y)$, $L(x+1, y+1)$, and $L(x, y+1)$ its neighbors; then

$$L(x, y) = \begin{cases} 0 \text{ (black)}, & \text{if } L(x+1, y) \text{ or } L(x+1, y+1) \\ & \text{or } L(x, y+1) \text{ is black} \\ 1 \text{ (white)}, & \text{otherwise.} \end{cases} \tag{11}$$

Dilation, on the contrary, resets the neighbors of each pixel rather than the pixel itself. If $L(x, y)$ is black; the substage sets all of its neighbors, $L(x+1, y)$, $L(x+1, y+1)$, and $L(x, y+1)$, to black; otherwise, the neighbors remain unchanged.

Logo Resizing. The proposed scheme adopts ISS to retrieve the authentication logo, which causes *pixel expansion* because one logo pixel is mapped to a share of four pixels (mentioned

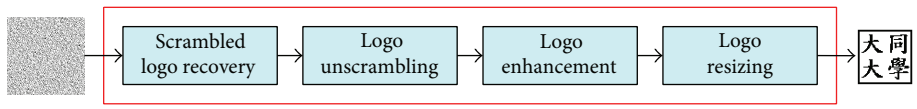

Figure 8: The four substages of *logo restoration*.

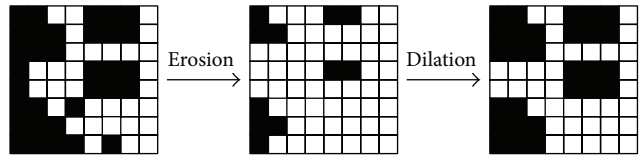

Figure 9: An example of erosion and dilation.

Figure 10: The authentication logo (64 × 64).

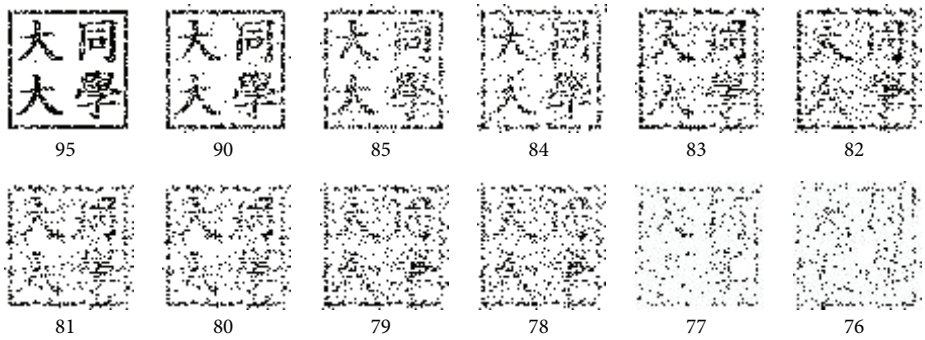

Figure 11: The restored logo with their *AR* values (%).

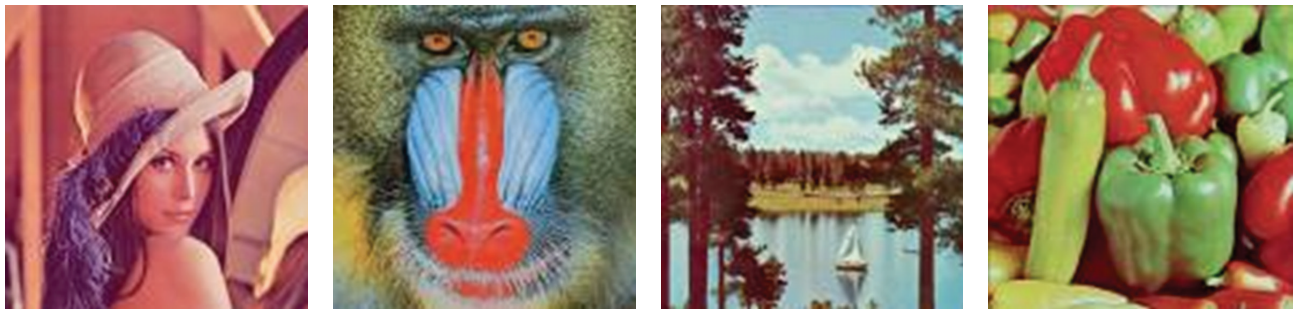

Figure 12: The test images used in the experiments: (a) Lena, (b) Mandrill, (c) Sailboat, and (d) Peppers.

in Section 3.1.3). As a result, the retrieved logo from the previous stage will be larger than the original one. To resize the logo to its original size, the substage partitions the enhanced logo into several blocks of size 2 × 2, each of which is then reduced into one pixel with value $L(x, y)$ according to the following rule:

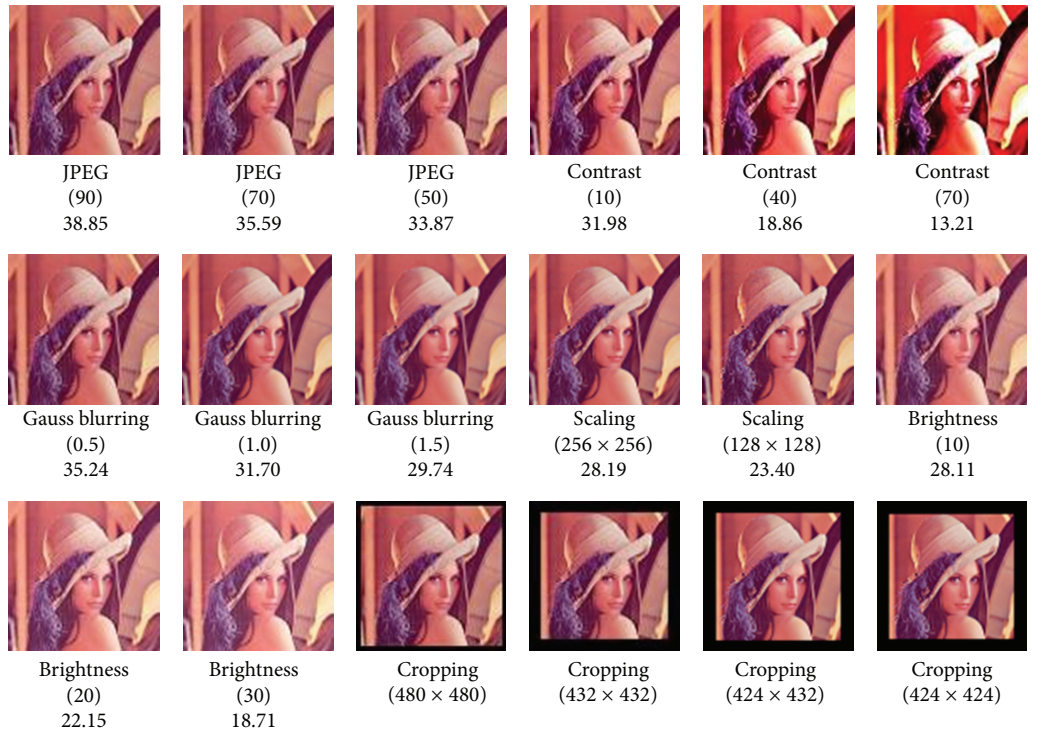

Figure 13: Some examples of "Lena" after different image attacks (PSNRs are listed in the last rows).

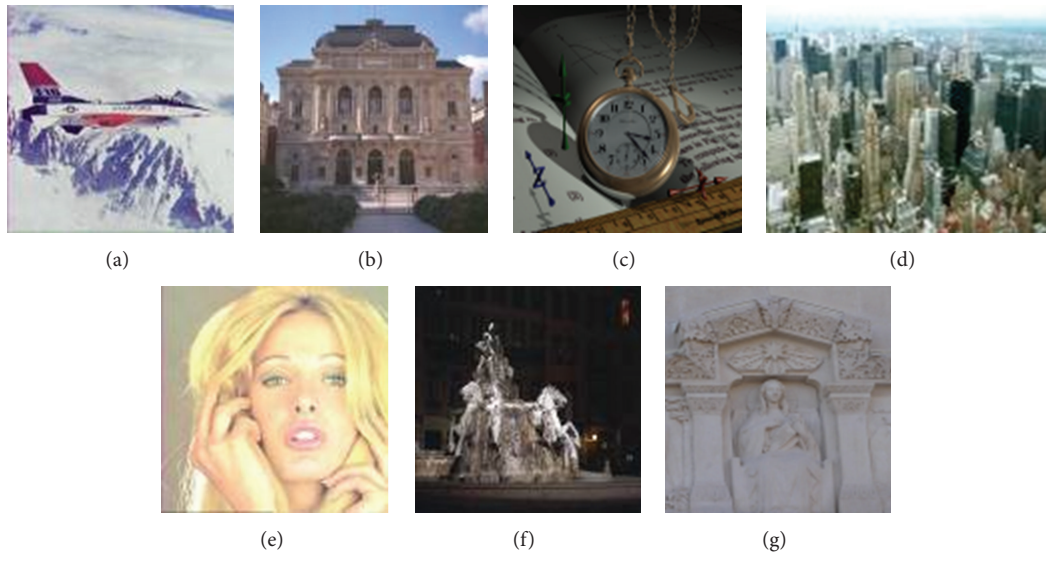

Figure 14: The other seven color images used for testing uniqueness.

$$L(x,y) = \begin{cases} 0\,(\text{black}), & \text{the number of the black pixels} \\ & \text{in each block} \geq 3, \\ 1\,(\text{white}), & \text{otherwise.} \end{cases} \tag{12}$$

4. Experimental Results

Two kinds of experiments were conducted to prove the effectiveness of the proposed scheme. The first experiment shows the robustness of our scheme and the other demonstrates the capability of unique identification. In the experiments, the authentication logo used to generate the watermark (fingerprint) is shown in Figure 10.

Two common measurements used to estimate the robustness of our scheme are described as below.

(1) Peak Signal to Noise Ratio (PSNR). The measurement to estimate the color image quality after image processing is a

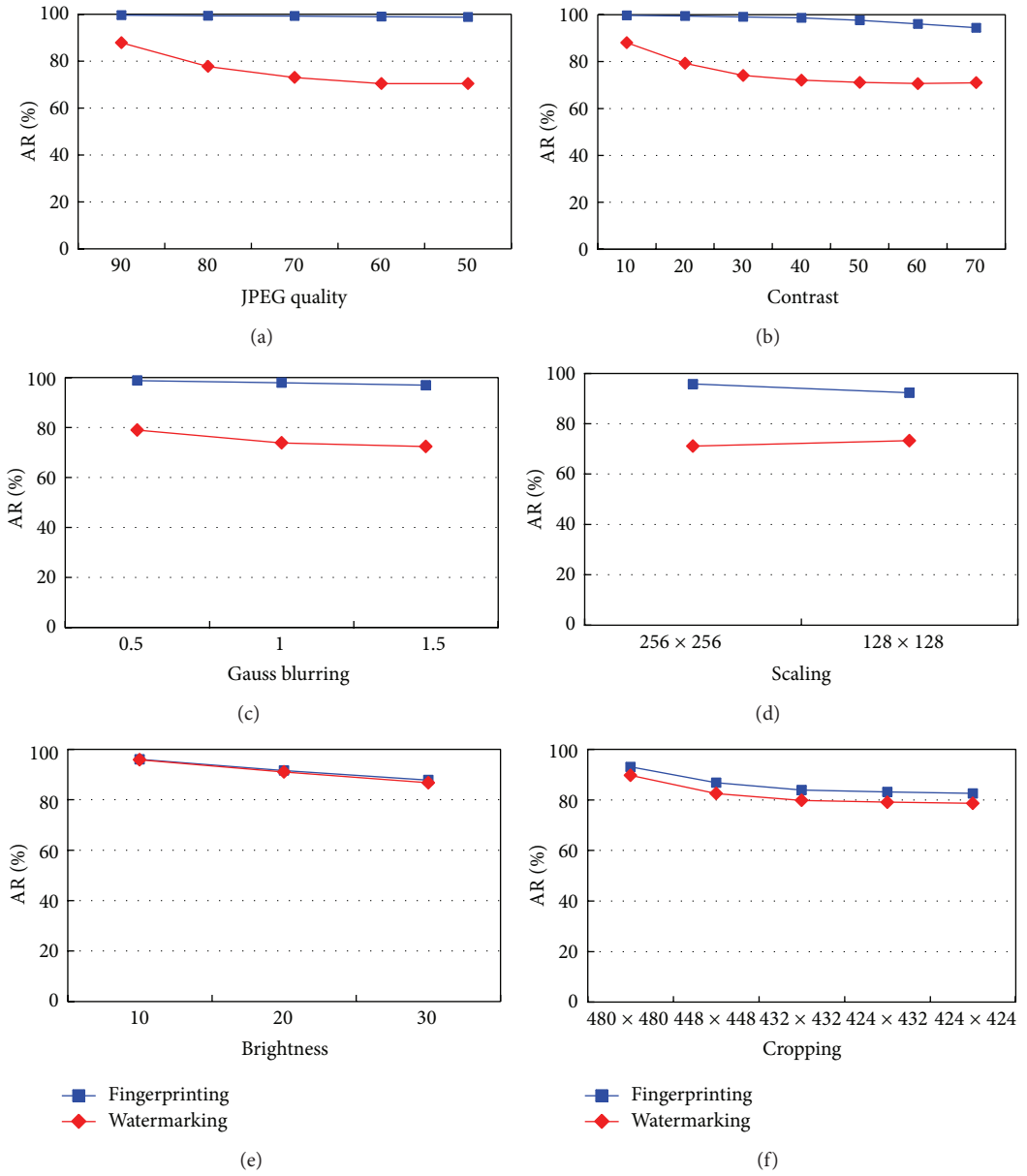

FIGURE 15: The average *AR* values of test images: (a) JPEG quality, (b) contrast, (c) Gauss blurring, (d) scaling, (e) brightness, and (f) cropping.

variant version of normal PSNR [29]. The variant PSNR listed below does not consider the influence of the green channel because the channel is not modified by our scheme. Consider

$$\text{PSNR} = 10\log_{10}\frac{255^2}{(\text{MSE}(R) + \text{MSE}(B))/2}\,\text{dB}, \quad (13)$$

where MSE is the *mean square error* between the original image and the modified image, which is defined as follows:

$$\text{MSE} = \frac{\sum_{i=1}^{N}\sum_{j=1}^{N}\left(x_{ij} - x'_{ij}\right)^2}{N^2}, \quad (14)$$

where x_{ij} represents the original pixel value and x'_{ij} denotes the modified pixel value.

According to the definition of PSNR, the higher the value is, the better the quality of the modified image is. Generally, if the PSNR is greater than 30 dB, the quality of the modified image is acceptable.

(2) Accuracy Rate (AR). The measurement shown below is used to evaluate the correctness of the logo after it has been restored. Consider

$$\text{AR}(\%) = \frac{\text{CP}}{\text{NP}} \times 100, \quad (15)$$

where NP is the number of pixels in the original logo and CP is the number of correct pixels obtained by comparing the pixels of the original logo with the corresponding ones of the restored logo. Figure 11 shows the restored logos with

Table 3: The resulting AR values of the test images.

Attacks		AR (%) (Watermarking/fingerprinting)				
		Lena	Mandrill	Sailboat	Peppers	Average
None		99.93/99.93	99.76/99.76	99.98/99.98	99.63/99.63	**99.83/99.83**
JPEG	(90)	88.31/99.71	86.84/99.44	88.04/99.88	88.16/99.49	**87.84/99.63**
	(80)	79.76/99.73	76.17/99.02	77.29/99.58	77.91/99.12	**77.78/99.36**
	(70)	74.58/99.46	72.12/98.73	72.61/99.54	73.05/99.17	**73.09/99.23**
	(60)	71.58/99.63	70.14/98.1	70.19/99.44	69.95/98.83	**70.47/99.00**
	(50)	71.48/99.41	70.09/97.68	70.43/99.24	69.8/98.88	**70.45/98.80**
Contrast	(10)	92.6/99.85	80.71/99.78	88.62/99.95	90.31/99.27	**88.06/99.71**
	(20)	82.86/99.76	73.12/99.56	79.27/99.78	81.81/98.68	**79.27/99.45**
	(30)	75.85/99.46	70.61/99.07	75.22/99.63	74.63/98.07	**74.08/99.06**
	(40)	72.61/99.27	70.83/97.92	72.31/99.19	72.51/98.14	**72.07/98.63**
	(50)	71.7/98.93	70.41/96.19	71.19/98.75	71.09/96.58	**71.10/97.61**
	(60)	70.39/97.68	70.65/92.7	70.95/98.1	70.53/95.8	**70.63/96.07**
	(70)	70.8/94.65	70.65/90.31	71.12/97.71	71.36/95	**70.98/94.42**
Gauss blurring	(0.5)	83.64/99.41	73.44/97.29	78.17/99.17	80.54/99.05	**78.95/98.73**
	(1.0)	76.1/98.88	70.92/95.78	74.17/98.44	73.88/98.34	**73.77/97.86**
	(1.5)	73.63/98.17	70.95/94.34	72.44/97.29	72.34/97.73	**72.34/96.88**
Scaling	w: 256, h: 256	71.66/97.92	70.75/92.53	71.22/96.83	70.73/95.8	**71.09/95.77**
	w: 128, h: 128	72.83/96.02	73.41/86.04	73.49/94.04	73.41/92.99	**73.29/92.27**
Brightness	(10)	96.66/96.78	94.41/94.92	97.85/97.85	94.46/94.46	**95.85/96.00**
	(20)	90.65/91.48	89.06/90.7	95.9/95.9	88.33/88.33	**90.99/91.60**
	(30)	85.72/87.62	83.98/86.43	93.55/93.58	83.69/83.64	**86.74/87.82**
Cropping	w: 480, h: 480	89.62/92.92	88.99/92.46	91.48/95.04	88.77/91.87	**89.72/93.07**
	w: 448, h: 448	82.71/86.82	81.71/85.67	83.79/88.89	82.03/85.82	**82.56/86.80**
	w: 432, h: 432	79.88/83.08	79.25/83.01	81.13/86.23	79/83.3	**79.82/83.91**
	w: 424, h: 432	78.88/81.76	78.49/81.96	80.52/85.86	78.59/83.06	**79.12/83.16**
	w: 424, h: 424	78.22/81.2	78/81.49	80.25/85.33	78.1/82.42	**78.64/82.61**

different AR values. As can been seen, restored logos with AR higher than 81% (the ones in the upper row) still can be visually recognized whereas the restored logo with AR equal to 76% is hard to identify. However, according to the description in Section 4.2, if AR is higher than 75%, the scheme still can identify the copyrights of the image.

4.1. Robustness Experiments. The experiments proved our scheme is robust to different kinds of attacks. The test images used in the experiment, including "Lena," "Mandrill," "Sailboat," and "Peppers," are shown in Figure 12. The commercial image processing software "Adobe Image Photoshop CS" was used to simulate several kinds of image attacks, some of which for "Lena" are shown in Figure 13.

The experimental results of the test images are shown in Table 3. As mentioned above, if AR is less than 75%, the scheme cannot identify the copyrights of the image. Table 3 shows that our scheme failed to verify the copyrights for the images suffering the heavier attacks of JPEG, contrast, Gauss blurring, and scaling in *watermark verification*. Nevertheless, the duplications of those images can all be determined in *fingerprint verification*. In summary, our scheme can identify the copyrights of the suspect images under moderate attacks in *watermark verification* and determine the duplications of those suffering the heavy attacks in *fingerprint verification*.

Moreover, there were 100 attacks in total and 53 of them resulted in an AR value higher than 75% in *watermarking verification*. That is to say, the copyrights of the 53% of the attacked images can be successfully identified in *watermark verification*, and hence only 47% of them need *fingerprint verification*.

4.2. Uniqueness Experiments. The experiment showed that our scheme has the capability of unique identification and is able to distinguish a copyrighted image from different ones. The four copyrighted images (with embedded watermarks) in Figure 12 along with seven unwatermarked images (Figures 14(a)–14(g)) were processed by our scheme to identify the copyright. The stored fingerprints of the watermarked images were used in *fingerprint verification* to restore the logos for all of the images.

The results shown in Table 4 demonstrated the extraordinary unique identification capability of our scheme. It can be clearly seen that all the restored logos except the ones of the copyrighted images (those on the rightmost side) are unrecognizable, which proves that our scheme is actually able to distinguish a copyrighted image from different ones. It will

Table 4: Experimental results of the uniqueness experiment.

(a) Lena

	Mandrill	Sailboat	Peppers	F-16	Opera	Watch	New York	Tiffany	Terraux	Fourviere	Lena
Watermark											
AR (%)	73.07	73.69	73.07	69.80	70.39	70.09	71.34	70.26	69.78	70.26	99.93
Fingerprint											
AR (%)	59.06	56.03	49.24	62.72	59.62	44.34	62.35	60.40	45.65	63.21	99.93

(b) Mandrill

	Lena	Sailboat	Peppers	F-16	Opera	Watch	New York	Tiffany	Terraux	Fourviere	Mandrill
Watermark											
AR (%)	73.74	73.69	73.07	72.73	73.52	73.79	72.64	74.35	73.82	72.2	99.76
Fingerprint											
AR (%)	58.2	57.81	62.08	57.84	59.45	60.03	59.64	54.35	59.01	60.38	99.76

(c) Sailboat

	Lena	Mandrill	Peppers	F-16	Opera	Watch	New York	Tiffany	Terraux	Fourviere	Sailboat
Watermark											
AR (%)	73.74	73.07	73.07	72.73	73.52	73.79	72.64	74.35	73.82	72.2	99.98
Fingerprint											
AR (%)	54.32	59.06	53.44	57.91	55.1	52.37	56.47	55.37	53.78	59.4	99.98

(d) Peppers

	Lena	Mandrill	Sailboat	F-16	Opera	Watch	New York	Tiffany	Terraux	Fourviere	Peppers
Watermark											
AR (%)	73.74	73.07	73.69	72.73	73.52	73.79	72.64	74.35	73.82	72.2	99.63
Fingerprint											
AR (%)	47.02	60.13	46.08	55.59	58.69	49.29	53.26	52.8	54.66	55.77	99.63

not mistakenly identify the copyrights of an unwatermarked image.

Moreover, the resulting AR values of all the unwatermarked images are all lower than 75%. Hence, it is reasonable for our scheme to confirm the copyright when AR is higher than 75%.

4.3. Discussion and Comparison. The fingerprint extracted from an image is more robust than the watermark embedded in the image. This can be seen from Figure 15, which shows the average AR values of the images suffering different attacks for watermarking and fingerprinting. The AR values for fingerprinting are all higher than those for watermarking.

Table 5: The processing time (ms) for watermark extraction and logo restoration and comparison.

Stage	Lena	Mandrill	Sailboat	Peppers	Average
Watermark extraction	218	209	200	213	210
Logo restoration and comparison	20	29	39	28	29
Watermark extraction + logo restoration and comparison	238	238	239	241	239

Table 6: Comparison of our combined scheme and the other two pure schemes.

	Pure watermarking scheme	Pure fingerprinting scheme	Our combined scheme
Robustness	△	◯	◯
Efficiency	◯	△	◯

*◯: good; △: fair.

When the image undergoes image processing operations that heavily damage the embedded watermark, the extracted fingerprint still can survive the attacks. However, because linear comparison is a computationally intensive process, fingerprinting is costly in time if there are many fingerprints in the database.

Table 5 shows the processing time for *watermark extraction* and *logo restoration and comparison* in our scheme. The experiments were carried out on a computer equipped with the following hardware and software:

CPU: 3.16 GHz Intel(R) Xeon(R) CPU E3120,

RAM: 4 GB,

OS: Windows 7,

Computing language and environment: MATLAB.

As Table 5 shows, it takes about 0.21 seconds to extract the watermark from the input image and 0.029 seconds to restore and compare the logo. Therefore, the total processing time T_p (in seconds) of our scheme can be expressed by

$$T_p = 0.24 + 0.029 \times n, \tag{16}$$

where n represents the number of the fingerprints retrieved from the database. If the copyright can be identified in *watermark verification*, n is 0. That is, our scheme only needs 0.24 seconds in the best case. Otherwise, our scheme needs additional 0.029 seconds for each retrieved fingerprint in the database.

Because our scheme combines watermarking and fingerprinting techniques, it can be as efficient as pure watermarking schemes and also as robust as pure watermarking schemes. A pure watermarking scheme is very efficient because it only needs to make one watermark comparison to verify the copyright. However, it may not be as robust as a pure fingerprinting scheme when dealing with images that have suffered heavy attacks. Table 6 shows the comparison of our combined scheme and the other two pure schemes. If the copyright of the input image can be identified by *watermarking verification*, our scheme can be as efficient as pure watermarking schemes. If *watermarking verification* fails to identify the copyright, our scheme is still able to determine duplication in *fingerprinting verification*, which makes our scheme as robust as pure fingerprinting schemes.

5. Conclusion

This paper presented a copyright identification scheme that takes advantage of the complementary nature of digital watermarking and fingerprinting. The experimental results showed that when the watermarked image suffers moderate attacks, *watermarking verification* alone is enough to identify the copyright, and there is no need for *fingerprinting verification*. In other words, the proposed scheme can identify the copyright efficiently in this situation. On the other hand, the experimental results also showed that when the watermarked image suffers heavy attacks that render *watermarking verification* incompetent, *fingerprinting verification*, although more time consuming, can successfully determine the duplication, hence demonstrating the robustness of the proposed scheme.

One distinguishing characteristic of the proposed scheme is that it does not need a separate watermark for *watermarking verification* and a separate fingerprint for *fingerprinting verification*. The proposed scheme extracts features from the input image to generate the fingerprint, which also serves as the watermark. Hence, only one piece of information is needed for both *watermarking* and *fingerprinting verifications*.

To further improve the scheme, retrieving the stored fingerprint image from the database to restore the correct authentication logo more quickly is worth studying. When there are more than one fingerprint image in the database, the original fingerprint of the host image must be correctly retrieved; otherwise, the correct authentication logo cannot be restored. Retrieving every stored fingerprint image to restore an authentication logo for comparison is very time consuming. The scheme should provide a more efficient way that is able to find the proper one in less time, which is the future work of the research.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

Financial support of this study by Tatung University, Taipei, Taiwan, under Grant B100-I07-036 is gratefully acknowledged.

References

[1] G. Coatrieux, H. Huang, H. Shu, and L. Luo, "A watermarking-based medical image integrity control system and an image

moment signature for tampering characterization," *IEEE Journal of Biomedical and Health Informatics*, vol. 17, no. 6, pp. 1057–1067, 2013.

[2] K.-C. Liu, "Colour image watermarking for tamper proofing and pattern-based recovery," *IET Image Processing*, vol. 6, no. 5, pp. 445–454, 2012.

[3] Y.-L. Chen and C.-T. Hsu, "Detecting recompression of JPEG images via periodicity analysis of compression artifacts for tampering detection," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 2, pp. 396–406, 2011.

[4] P. -C. Su, Y. -C. Chang, and C. -Y. Wu, "Geometrically resilient digital image watermarking by using interest point extraction and extended pilot signals," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 12, pp. 1897–1908, 2013.

[5] A. Piper and R. Safavi-Naini, "Scalable fragile watermarking for image authentication," *IET Information Security*, vol. 7, no. 4, pp. 300–311, 2013.

[6] M. Pawlak and Y. Xin, "Robust image watermarking: an invariant domain approach," in *Proceedings of the IEEE Canadian Conference on Electrical and Computer Engineering*, vol. 2, pp. 885–888, May 2002.

[7] W.-H. Lin, S.-J. Horng, T.-W. Kao, P. Fan, C.-L. Lee, and Y. Pan, "An efficient watermarking method based on significant difference of wavelet coefficient quantization," *IEEE Transactions on Multimedia*, vol. 10, no. 5, pp. 746–757, 2008.

[8] I.-K. Kong and C.-M. Pun, "Digital image watermarking with blind detection for copyright verification," in *Proceedings of the 1st International Congress on Image and Signal Processing (CISP '08)*, vol. 1, pp. 504–508, May 2008.

[9] L. A. Elrefaey, M. E. Allam, H. A. Kader, and M. Selim, "Robust blind image-adaptive watermarking," in *Proceedings of the 25th National Radio Science Conference (NRSC '08)*, March 2008.

[10] M.-H. Lee and O.-J. Kwon, "Color image watermarking based on DS-CDMA using Hadamard kernel," in *Proceedings of the 10th International Conference on Advanced Communication Technology*, pp. 1592–1597, February 2008.

[11] N.-Y. Lee and C.-C. Wang, "Yet another wavelet watermarking scheme for copyright protection," in *Proceedings of the 9th IEEE International Conference on E-Commerce Technology*, pp. 421–424, July 2007.

[12] H.-M. Sun, C.-J. Hong, and C.-H. Chen, "A new approach to feature-based copyright protection of images," in *Proceedings of the 3rd International Conference on Information Technology: Research and Education (ITRE '05)*, pp. 233–237, June 2005.

[13] C.-C. Chang and J.-C. Chuang, "An image intellectual property protection scheme for gray-level images using visual secret sharing strategy," *Pattern Recognition Letters*, vol. 23, no. 8, pp. 931–941, 2002.

[14] C.-S. Lu, C.-Y. Hsut, S.-W. Sun, and P.-C. Chang, "Robust mesh-based hashing for copy detection and tracing of images," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME '04)*, vol. 1, pp. 731–734, June 2004.

[15] J.-S. Lee and K.-S. Yoon, "The system integration of DRM and fingerprinting," in *Proceedings of the 8th International Conference Advanced Communication Technology (ICACT '06)*, vol. 3, pp. 2180–2183, February 2006.

[16] S.-H. Yang and C.-F. Chen, "Robust image hashing based on SPIHT," in *Proceedings of the 3rd International Conference on Information Technology: Research and Education (ITRE '05)*, pp. 110–114, June 2005.

[17] F. Ahmed and M. Y. Siyal, "A secure and robust DCT-based hashing scheme for image authentication," in *Proceedings of the 10th IEEE Singapore International Conference on Communication Systems (ICCS '06)*, pp. 1–6, October 2006.

[18] J. S. Seo, J. Haitsma, T. Kalker, and C. D. Yoo, "A robust image fingerprinting system using the Radon transform," *Signal Processing: Image Communication*, vol. 19, no. 4, pp. 325–339, 2004.

[19] P. C. Vila, *Content-based audio search: from fingerprinting to semantic audio retrieval [Ph.D. thesis]*, University Pompeu Fabra, Barcelona, Spain, 2007.

[20] S.-L. Hsieh, I.-J. Tsai, B.-Y. Huang, and J.-J. Jian, "Protecting copyrights of color images using a watermarking scheme based on secret sharing and wavelet transform," *Journal of Multimedia*, vol. 3, no. 4, pp. 42–49, 2008.

[21] V. Nabiyev and A. Günay, "Towards a biometric purpose image filter according to skin detection," in *Proceedings of the Second International Conference on Problems of Cybernetics and Informatics*, 2008.

[22] C. Poynton, "Frequently asked questions about color," 2014, http://www.poynton.com/ColorFAQ.html.

[23] G. Voyatzis and I. Pitas, "Applications of torus automorphisms in image watermarking," in *Proceedings of International Conference on Image Processing*, vol. 3, pp. 237–240, 1996.

[24] G. Voyatzis and I. Pitas, "Chaotic mixing of digital images and applications to watermarking," in *Proceeding of European Conference on Multimedia Applications, Services and Techniques (ECMAST '96)*, vol. 2, May 1996.

[25] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, Prentice Hall, New York, NY, USA, 2nd edition, 2002.

[26] E. J. Stollnitz, T. D. DeRose, and D. H. Salestin, "Wavelets for computer graphics: a primer, part 1," *IEEE Computer Graphics and Applications*, vol. 15, no. 3, pp. 76–84, 1995.

[27] KeyLawk, "Colors: the human eye is most sensitive to GREEN," 2014, http://keylawk.blogspot.com/2007/10/colors-human-eye-is-most-sensitive-to.html.

[28] S. Zhao, "Wavelength of maximum human visual sensitivity," 2014, http://hypertextbook.com/facts/2007/SusanZhao.shtml.

[29] I. Nasir, W. Ying, and J. Jianmin, "Novel multiple spatial watermarking technique in color images," in *International Conference on Information Technology: New Generations (ITNG '08)*, pp. 777–782, April 2008.