

Received October 20, 2020, accepted November 20, 2020, date of publication December 24, 2020, date of current version January 12, 2021.

Digital Object Identifier 10.1109/ACCESS.2020.3047270

A Color Image Authentication Scheme With Grayscale Invariance

WIEN HONG¹, JEANNE CHEN¹, PEI-SHIH CHANG¹, JIE WU²,
TUNG-SHOU CHEN¹, AND JASON LIN^{1,3}

¹Department of Computer Science and Information Engineering, National Taichung University of Science and Technology, Taichung 40401, Taiwan

²School of Electrical and Computer Engineering, Nanfang College of Sun Yat-sen University, Guangzhou 510970, China

³International School of Technology and Management, Feng Chia University, Taichung 40724, Taiwan

Corresponding author: Jason Lin (jasonlin@fcu.edu.tw)

ABSTRACT Color image authentication is a method that allows the detection of tampered regions on images. Existing related works evaluated their performance based on visual quality and detection capability of the marked image, but the issue of grayscale invariance was ignored. However, many applications in image processing required the color image to first convert into a grayscale image before any further post-processing such as edge detection, color masking in Photoshop, and the display of e-ink. If the resulting image and the original image are different in their grayscale value, they might produce different outcomes after the post-processing. Therefore, how to maintain unchanged of the grayscale value has become an important issue. This paper presents a grayscale-invariance color image authentication technique. We proposed to embed the authentication code into two of three primary color channels and adjust the remaining one to remedy the distortion of grayscale value. The experimental results showed that, when embedding each four-bit authentication code into two color channels, the visual quality of the marked images has achieved an average 33.26 dB of peak signal-to-noise ratio (PSNR) while providing a satisfactory detectability.

INDEX TERMS Image authentication, grayscale invariance, color image.


I. INTRODUCTION

Digital images have become an inevitable electronic medium nowadays. Various applications such as video streaming, animation, and video conferencing are built based on digital images, which broadly existed in our daily life. Despite the importance of digital images, the ease of tampering with their content has caused a lot of criminal events relevant to confidentiality, integrity, and accessibility of the transmitted images. Therefore, many researchers have developed different approaches for authentication of digital images [1]–[18].

The image authentication technique can be used to verify the integrity of the image and to ascertain whether the image has been tampered with or damaged. Existing image authentication techniques can be roughly divided into two types of approaches. The first type of approaches applied statistical features [1], [2] to verify the images. They utilize the nature of statistical feature in digital images to achieve the goal. If an image has been tampered, it will have different statistical features than the normal one. We can leverage from these

statistical results to verify whether an image has been tampered or not. The second type of approaches is watermark-based authentication schemes [3]–[18]. This type of approaches will first produce a fragile watermark consisting of authentication code generated from some key information of the original image, and then utilize the embedding techniques [19]–[21] to embed the authentication code into the original image to form a marked one. When verifying whether an image has been tampered, we only need to extract the key information from the marked image to generate the authentication code and examine whether it is consistent with the one that embedded in the original image. Although this type of approaches may cause distortions to the visual quality, their detection rate on the tampered regions is much better than approaches using statistical features. Hence, the watermark-based methodology for image authentication research has attracted a lot of attention.

The watermark-based authentication techniques can be applied to images of compressed domain [16]–[18] or spatial domain [3]–[15]. The compressed domain methods embed the authentication codes into the compressed data stream

The associate editor coordinating the review of this manuscript and approving it for publication was Amit Singh .

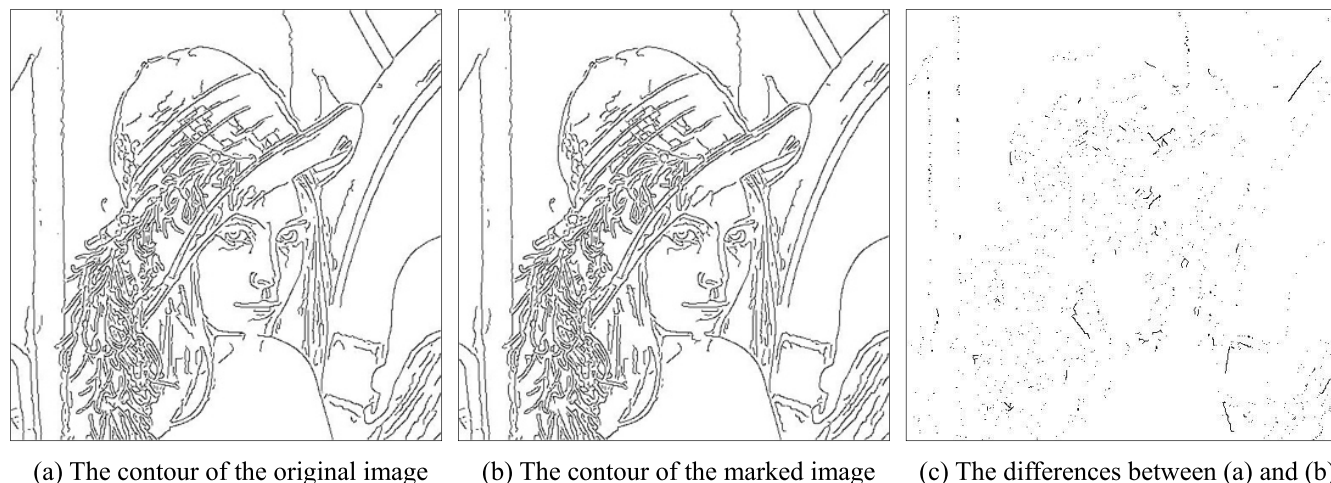


FIGURE 1. The contour differences of Lena image before and after the embedment of authentication code.

while the spatial domain methods embed the authentication code by altering the pixel values. In general, the redundancy of image data will be greatly reduced after image compression. The compressed domain methods will result in a less performance of tamper detection due to the embedding capacity has largely decreased. Also, this implies that larger image blocks are required to store the authentication code, causing an unpleasant block-level detection result. On the contrary, the spatial domain methods have more redundant spaces for embedding the authentication code due to the nature of strong correlations between adjacent pixels. This property allows us to precisely locate the tampered regions when verifying the images.

There are numerous authentication techniques for grayscale images been proposed thus far in spatial domain [3]–[8]. These embedding techniques can be roughly divided into irrecoverable [3]–[5] and recoverable [6]–[8] schemes. The recoverable schemes not only detect the tampered regions, but also recover tampered regions back to its original content. However, this type of scheme requires more extra space to accommodate the recovery information that used to recover the original pixel values. To achieve this goal, image blocks rather than pixels are often used as embedding units to embed both authentication and recovery codes. Therefore, the tamper localization of this type can only be implemented in block-level. Compared with the recoverable schemes, the irrecoverable authentication schemes cannot recover the image to its original version. Nevertheless, they do not require extra spaces for storing the reversible code. Therefore, they obtain a better visual quality of images and the detection precision can be implemented in pixel-wise at best.

Although the above-mentioned authentication techniques [3]–[8] only apply to grayscale images, they can be easily converted to schemes that apply in color images. Each pixel in a color image consists of three primary color channels: red, green, and blue. Therefore, a color image has more

embedding spaces than a grayscale image with one single channel. While most of the earlier works on color image authentication only focus on elevating the image quality and detection rate [9], [10], recent works [11]–[15] have added in some special features such as recoverable on the tampered regions. However, none of them have considered whether the original image and the marked image will produce the same grayscale images.

Many imaging applications require converting the color image into grayscale image before any further post-processing. If the grayscale values of any marked image are different from the ones of its original image, the post-processing of the two images will result differently, and further cause problems in the subsequent processes. For example, image processing applications such as edge detection and color masking using Photoshop required to convert the color image into a grayscale one before proceeding to the next step of processing. Therefore, ensuring the grayscale-invariance between the original and marked images is an important task for such applications since a different grayscale image might produce different results after the subsequent processing.

We demonstrate a simple example as shown in Figure 1. Figure 1(a) is the result of edge detection for the original image of Lena, and Fig. 1(b) is the result that performs the same edge detection after embedding the authentication code into Lena image using Belferdi *et al.*'s method [13]. The differences between the two images are present in Fig. 1(c). The outcomes of Fig. 1(a) and Fig. 1(b) are approximately the same in visual observations, but the differences between the two reveals in Fig. 1(c) like Lena's hair, her face, and the lower edge of her shoulder.

For these reasons, we proposed the first color image authentication technique based on grayscale invariance. The proposed technique not only assures that both the original image and marked image generate the same grayscale image but also maintain a good visual quality of the image. Moreover, on tamper detection, our technique can precisely

mark the tampered regions in pixel-level. To the best of our knowledge, this is the first work presenting the color image authentication with the capability of grayscale invariance.

II. PROPOSED METHOD

A color image consists of three primary color channels: red, green, and blue (RGB). Many applications of color image required to convert each RGB pixel into a grayscale value before any further post-processing. The resulting image is usually called the grayscale image. The human visual system has different perceptual intensities on the RGB channels. Hence, when converting a color pixel with colors r , g , and b into a grayscale value gv , instead of taking the average of the RGB values, we apply the following conversion formula [22] to calculate gv :

$$gv = f(r, g, b) = \text{round}(0.299r + 0.587g + 0.114b), \quad (1)$$

where $f(\cdot)$ is the function for color-to-grayscale conversion and the function $\text{round}(x)$ rounds up the value of x to an integer representing the grayscale value gv . Note that in Eq. (1), the green channel g has the largest coefficient because the human visual system is more sensitive to green color than red and blue ones. Therefore, if embedding the authentication code into red and blue channels, we only need to slightly alter the value of green color to ensure the invariance of the grayscale value. According to the above observation, this paper proposes a novel image authentication method to achieve grayscale invariance by embedding the authentication code into the LSBs of red and blue channels, and then alters the value of green channel to complement the changes that impact to the grayscale value.

A. GENERATION AND EMBEDMENT OF AUTHENTICATION CODE

Let $p_i = (r_i, g_i, b_i)$ be the i -th pixel with RGB values r_i , g_i , and b_i of a color image, and the lengths of authentication codes to be embedded into r_i and b_i are ℓ_r and ℓ_b , respectively. Therefore, the total length of authentication code is $\ell_r + \ell_b$. Suppose r_i^m and b_i^m are the decimal value of r_i and b_i with ℓ_r and ℓ_b LSBs setting to zeros, respectively. As a result, we have

$$r_i^m = \text{floor}(r_i/2^{\ell_r}) \times 2^{\ell_r}, \quad (2)$$

and

$$b_i^m = \text{floor}(b_i/2^{\ell_b}) \times 2^{\ell_b}, \quad (3)$$

where $\text{floor}(x)$ is the floor function that takes the closest integer less than x .

To generate the authentication code ac_i of p_i , the grayscale value gv_i of p_i is obtained from Eq. (1) firstly, and then gv_i , r_i^m , b_i^m , and i are hashed using MD5 [23]. The binary output after hashing is complemented with 0s at the end of output until the number of bits is divisible by $2^\eta \times (\ell_r + \ell_b)$, where η is a positive integer. The resulting bit string is then iteratively folded by performing the exclusive-OR operation to

reduce the length of complemented bit string until it reaches to $\ell_r + \ell_b$ bits. The folded result is the authentication code ac_i to be embedded into p_i by using the LSB substitution.

To embed authentication code ac_i into p_i , ac_i is firstly divided into ac_i^r and ac_i^b . Let ac_i^r be the first ℓ_r bits of ac_i , and ac_i^b be the rest ℓ_b bits of ac_i . We can then embed ac_i^r into r_i and ac_i^b into b_i using the equations

$$\hat{r}_i = r_i^m + (ac_i^r)_{10} \quad (4)$$

and

$$\hat{b}_i = b_i^m + (ac_i^b)_{10}, \quad (5)$$

where $(x)_{10}$ represents the decimal value of the binary string x .

After the embedment of ac_i into p_i , two altered values \hat{r}_i and \hat{b}_i of the red and blue channels will be obtained, respectively. Since \hat{r}_i and \hat{b}_i have caused distortions to the grayscale of the original pixel p_i , the original value of the green channel g_i has to be altered by

$$\hat{g}_i = \text{round}((gv_i - 0.299\hat{r}_i - 0.114\hat{b}_i)/0.587) \quad (6)$$

to obtain an adjusted green color value \hat{g}_i such that the resulting grayscale value \hat{g}_i of the i -th marked pixel $\hat{p}_i = (\hat{r}_i, \hat{g}_i, \hat{b}_i)$ will be equal to the grayscale value gv_i of the i -th original pixel $p_i = (r_i, g_i, b_i)$.

Let us further use a simple example to demonstrate the adjustment to achieve grayscale-invariance. Suppose the i -th pixel of the original image is $p_i = (42, 151, 69)$ and the length of the authentication code ac_i is $\ell_r = \ell_b = 4$. Eqs. (2) and (3) are first applied to obtain $r_i^m = 32$ and $b_i^m = 64$. Assume the authentication code created from the input (r_i^m, b_i^m, gv_i, i) is $ac_i = 1001\ 1000_2$. Given that $ac_i^r = 1001_2$ and $ac_i^b = 1000_2$, Eqs. (4) and (5) are applied to get $\hat{r}_i = 41$ and $\hat{b}_i = 72$, respectively. The value of the remaining unchanged green channel g_i will then be adjusted to $\hat{g}_i = 151$ by Eq. (6) to complement the distortions caused by \hat{r}_i and \hat{b}_i . As a result, the marked pixel $\hat{p}_i = (41, 151, 72)$ is the grayscale-invariance alteration of the original pixel $p_i = (42, 151, 69)$ after embedding the eight-bit authentication code $1001\ 1000_2$.

The direct LSB replacement of the authentication code in Eqs. (4) and (5) has assured the value of \hat{r}_i and \hat{b}_i falls within the interval $[0, 255]$. However, the value of \hat{g}_i produced by Eq. (6) is dependent on \hat{r}_i and \hat{b}_i . Eq. (6) is solvable if and only if the result of \hat{g}_i is in the range from 0 to 255. This implies that not only the authentication code has successfully embedded into \hat{r}_i and \hat{b}_i , but also the grayscale invariance is guaranteed by the adjusted \hat{g}_i . That is,

$$gv_i = f(r_i, g_i, b_i) = f(\hat{r}_i, \hat{g}_i, \hat{b}_i) = \hat{g}_i. \quad (7)$$

If Eq. (6) is solvable, p_i is termed a solvable pixel, and the aforementioned method to embed authentication code into a solvable pixel is referred to as the Type I embedment. On the contrary, if \hat{g}_i falls outside of the range $[0, 255]$, then Eq. (6) is considered unsolvable under this circumstance, and

p_i is termed an unsolvable pixel. When \hat{g}_i has no solution, another Type II embedding method is required to ensure that the marked pixel \hat{p}_i remains the same grayscale value as its original pixel p_i .

B. THE EMBEDDING METHOD FOR UNSOLVABLE CASE

Two major factors will lead Eq. (6) to no solution. First of all, an overlong authentication code might cause too much deviation on red and blue channel after the embedding, and hence cannot merely depend on the alteration of green channel to keep the invariance of grayscale value. Second, since the authentication code only embeds in red and blue channels, the burden of adjusting the value to keep grayscale invariance will be taken solely on the remaining green channel, which increases the likelihood of unsolvable cases. Therefore, when unsolvable cases happen, the proposed Type II embedding method sets the length of authentication code to two bits and embedded them into blue channel to avoid the unsolvable cases. The reason of selecting the blue channel is due to its least coefficient in Eq. (1), which allows the alteration to the large extent to achieve the same grayscale value as the original pixel.

To find an alternative pixel $p'_i = (r'_i, g'_i, b'_i)$ to achieve the above-mentioned goal, the red channel r_i and green channel g_i are perturbed within a small value α to collect all possible perturbed pairs (r'_i, g'_i) , as described in Eqs. (9). Use Eq. (10) to set two LSBs of b_i to 00_2 and obtain b_i^* . Notice that b_i^* is also perturbed by an integer z , where $-1 \leq z \leq 1$. By substituting the two-bit authentication code into the LSBs of b_i^* , b_i^* may increase from 0 to 3. Hence, we need to make sure that (r'_i, g'_i, b_i^*) and $(r'_i, g'_i, b_i^* + 3)$ have the same grayscale value gv_i via Eq. (11). The function $\text{hash}(x)$ in Eq. (12) fold the hash code x created from the input via MD5 into a two-bit authentication code ac_i . Finally, ac_i is embedded into b_i^* using Eq. (13) and obtains the embedded result b'_i . Under this circumstance, several p'_i can be found. Although all these p'_i can be used as alternative pixels, the best candidate with the minimum sum of squared deviations to the original color pixel p_i will be selected in order to make the marked pixel visually looked closer to the original one. Overall, the whole process of the embedding method for unsolvable cases can be formulated into an optimization problem described as follows:

$$\text{Minimize: } (r_i - r'_i)^2 + (g_i - g'_i)^2 + (b_i - b'_i)^2 \tag{8}$$

$$\text{Subject to: } |r_i - r'_i| \leq \alpha, |g_i - g'_i| \leq \alpha, \tag{9}$$

$$b_i^* = (\text{floor}(b_i/4) + z) \times 4, -1 \leq z \leq 1 \tag{10}$$

$$gv_i = f(r_i, g_i, b_i) = f(r'_i, g'_i, b_i^*) = f(r'_i, g'_i, b_i^* + 3), \tag{11}$$

$$ac_i = \text{hash}(i, gv_i, r'_i, g'_i, b_i^*), \tag{12}$$

$$b'_i = b_i^* + (ac_i)_{10}, \tag{13}$$

$$0 \leq r'_i \leq 255, 0 \leq g'_i \leq 255, 0 \leq b'_i \leq 255, \tag{14}$$

where $r'_i, g'_i,$ and b'_i are the variables to be optimized. The solution to the above optimization problem is denoted by $\hat{p}_i = (\hat{r}_i, \hat{g}_i, \hat{b}_i)$. Notice that p_i and \hat{p}_i have the same grayscale value and the two-bit authentication code is embedded into the two LSBs of \hat{b}_i .

We use a simple example to demonstrate the embedding process of the unsolvable cases. Suppose $p_i = (0, 0, 6), gv_i = 1, \ell_r = \ell_b = 4$ and $ac_i = 1101\ 1111_2$. The calculation process in Section II.B yields $\hat{r}_i = 13$ and $\hat{b}_i = 15$. Obviously, \hat{g}_i needs to become negative to keep $gv_i = 1$ and makes p_i unsolvable. Therefore, the embedding method described in this section will be applied to remedy the unsolvable situation. We assume $\alpha = 1$ and all pairs of $(r'_i, g'_i) = \{(0, 0), (0, 1), (1, 0), (1, 1)\}$ that satisfy Eqs. (9) and (14) are collected. Note that $(r'_i, g'_i, b_i^*) = \{(0, 0, 8), (0, 1, 0), (0, 1, 4), (1, 0, 4), (1, 1, 0)\}$ satisfy the constraints in Eq. (11). By assuming $ac_i = 10_2$, we have $(r'_i, g'_i, b'_i) = \{(0, 0, 10), (0, 1, 2), (0, 1, 6), (1, 0, 6), (1, 1, 2)\}$. Compared with other pixel combinations, $(0, 1, 6)$ has achieved the minimum result in Eq. (8), so we set $\hat{p}_i = (0, 1, 6)$, which is the marked pixel of $p_i = (0, 0, 6)$ after embedding the two-bit authentication code.

C. THE EMBEDMENT PROCEDURES

During the embedment of authentication code, a list U containing all the grayscale values of first visited unsolvable pixels is built. Any color pixel with grayscale value in U has to use the Type II embedding method described in Section II.B to embed its authentication code. In contrast, if the grayscale value of a color pixel is not in U , then the Type I embedding method is applied to this pixel to embed the authentication codes. The embedding procedures of the proposed method are listed step-by-step as follows:

Input: Color image I , parameters $\alpha, \ell_r,$ and ℓ_b .

Output: Marked color image \hat{I} , a list U containing grayscale values using Type II embedment.

- Step 1: Initialize an empty list U and sequentially scan pixels in I . Let p_i be the scanned pixels.
- Step 2: Calculate the grayscale value gv_i of p_i using Eq. (1). If gv_i is in U , go to Step 3 to perform Type II embedment; otherwise, go to Step 4 to attempt to perform Type I embedment. If no more pixel to be processed, go to Step 5.
- Step 3: Generate a 2-bit authentication code ac_i , and embed ac_i into p_i using the Type II embedment, as described in Section II.B. Go to Step 2 to process the next pixel.
- Step 4: Generate a $(\ell_r + \ell_b)$ -bit ac_i and attempt to embed ac_i into p_i using the Type I embedment, as described in Section II.A. If the attempt is negative (i.e., p_i is unsolvable), then place gv_i in U and go to Step 3. If the attempt is positive, we have obtained the embedded pixel \hat{p}_i and go to Step 2 to process the next pixel.
- Step 5: All pixels have successfully embedded their authentication code, and the marked image $\hat{I} = \{\hat{p}_i\}_{i=1}^N = \{(\hat{r}_i, \hat{g}_i, \hat{b}_i)\}_{i=1}^N$ of the original image is generated.

We use 8-bit to record each grayscale value in U , and U has to be used as side information to authenticate the marked image \tilde{I} , as will be addressed in the next sub-section. Therefore, the size of side information is $8 \times |U|$, where $|U|$ is the length of U . In general, there are only a few grayscale values in U and thus a few bits are enough to record them. Figure 2 shows the schematic diagram of the proposed embedding method.

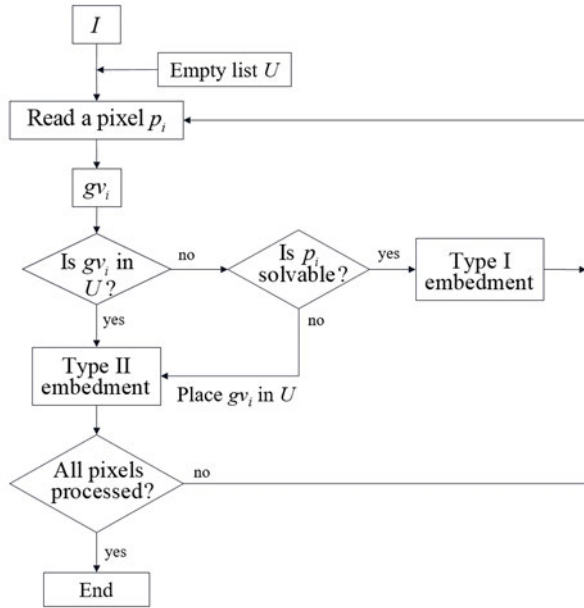


FIGURE 2. Embedding procedures of the proposed method.

D. THE DETECTION OF TAMPERED PIXELS

To detect whether a marked image $\tilde{I} = \{\tilde{p}_i\}_{i=1}^N = \{(\tilde{r}_i, \tilde{g}_i, \tilde{b}_i)\}_{i=1}^N$ has been tampered, we can check whether there is any modification on the extracted authentication code. We require the list U to find out what type of the embedding method a pixel used during embedment. Once the embedding method of a pixel is known, we can extract the authentication code for that pixel using the corresponding extraction algorithm. The tampered detection procedures are listed as follows:

Input: To-be-detected image \tilde{I} , list U , and parameters α , ℓ_r , and ℓ_b .

Output: Detection result of \tilde{I} .

- Step 1: Sequentially scan all pixels $\{\tilde{p}_i\}_{i=1}^N$ in the given to-be-detected image, where $\tilde{p}_i = (\tilde{r}_i, \tilde{g}_i, \tilde{b}_i)$.
- Step 2: If the grayscale value \tilde{g}_{v_i} of the marked pixel \tilde{p}_i is not found in U , then the Type I embedding method was used to embed the authentication code. In this case, \tilde{r}_i^m and \tilde{b}_i^m is obtained using Eqs. (2) and (3), and the $(\ell_r + \ell_b)$ -bit authentication code \tilde{a}_{c_i} is obtained, as described in Section II.A. The embedded authentication code $\tilde{e}_{a_{c_i}}$ can be obtained by concatenating the ℓ_r LSBs of \tilde{r}_i and the ℓ_b LSBs of \tilde{b}_i .

Step 3: If the grayscale value \tilde{g}_{v_i} of the marked pixel \tilde{p}_i is found in U , the authentication code is embedded using the Type II embedding method. In this case, i , \tilde{g}_{v_i} , \tilde{r}_i , \tilde{g}_i , and $\tilde{b}_i^* = \text{floor}(\tilde{b}_i/4) \times 4$ are used to generate the authentication code \tilde{a}_{c_i} . The embedded two-bit authentication code $\tilde{z}_{a_{c_i}}$ can be obtained by extracting two LSBs of \tilde{b}_i .

Step 4: Check whether the generated authentication code \tilde{a}_{c_i} and the embedded authentication code $\tilde{z}_{a_{c_i}}$ are the same. If yes, then the marked pixel has not been tampered. Otherwise, the marked pixel has been tampered.

Step 5: Repeat Steps 2 to 4 until all pixels have been through tamper detection.

III. EXPERIMENTAL RESULTS

In this section, we conduct several experiments to evaluate the performance of the proposed method. The experiments will use different lengths of authentication code on various test images to show the distributions of unsolvable pixels and types of embedment, the image quality after embedment, and the error detection rate of tampered images. A total of eight grayscale images of size 512×512 : Lena, House, Jet, Peppers, Sailboat, Splash, Tiffany, and, Baboon, as shown in Fig. 3, are used as the test images for our experiments. These test images can be obtained from the USC-SIPI image database [24].

We adopt the PSNR metric to evaluate the image quality after embedment, which is calculated as

$$\text{PSNR} = 10 \times \log_{10} \frac{255^2}{\text{MSE}}, \tag{15}$$

where MSE is the mean square error between the original image and the marked image. From the evaluation in Eq. (15), a higher PSNR value represents lower distortion of marked image and vice versa. Moreover, in all the experiments of this section, we set $\alpha = 2$ because this setting already achieved a very satisfactory result.

A. THE INFLUENCE OF UNSOLVABLE PIXELS

Examine whether a pixel is unsolvable or not is crucial in the proposed method. An unsolvable pixel not only has to be processed using the Type II embedding technique but also forces other pixels with the same grayscale to be embedded using Type II. In the following section, we will investigate the distribution of the unsolvable pixels as well as the distributions of embedment using Type I and Type II.

1) THE DISTRIBUTION OF UNSOLVABLE PIXELS

In this experiment, four images, including Lena, Sailboat, Splash, and Tiffany are used to demonstrate the distribution of unsolvable pixels for attempting the embedment of authentication code using Type I embedding method. For each image, the embedment will attempt on $\ell_r = \ell_b = 2$, $\ell_r = \ell_b = 3$, and $\ell_r = \ell_b = 4$. Figure 4 shows the experimental results after embedment, where black dots are the locations

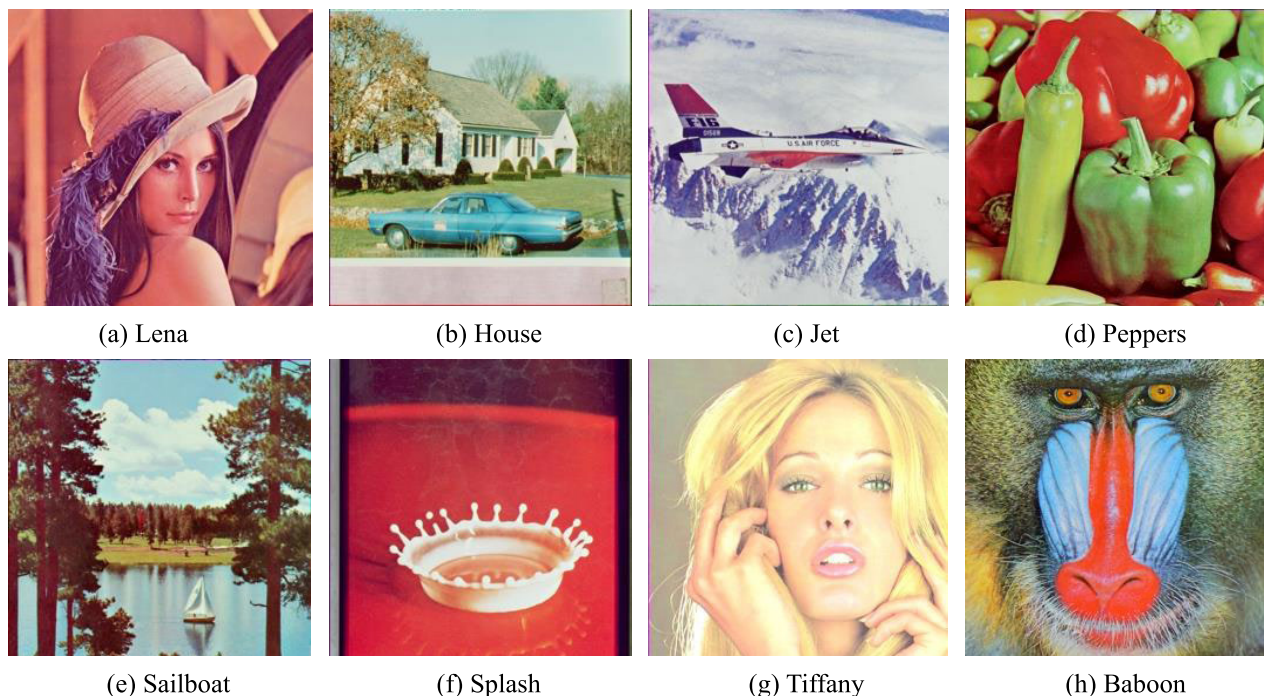


FIGURE 3. Eight 512 × 512 color test images.

TABLE 1. The number of unsolvable cases for different lengths of authentication code.

(l_r, l_b)	Lena	House	Jet	Peppers	Sailboat	Splash	Tiffany	Baboon
(2,2)	27	15	1	3519	124	3827	5617	16
(3,3)	39	22	2	5217	200	5700	9935	41
(4,4)	105	36	9	7340	348	6485	16212	74
(2,3)	30	18	1	4744	142	4284	5814	27
(3,2)	43	20	2	4413	174	5500	9799	34
(2,4)	35	27	2	7253	208	5025	6562	53
(4,2)	100	30	11	5779	283	6278	16185	58

of unsolvable pixels. As seen in Figs. 4(a), (b), and (c), black dots sparsely distribute over the image. These results imply that only a few pixels in the Lena image need to apply the Type II embedment. Furthermore, Fig. 4(c) shows that even embedding a longer authentication code with $l_r = l_b = 4$, most pixels still remain solvable.

The number of unsolvable pixels for the Sailboat image (Figs. 4(d), (e), and (f)) are approximately the same as the Lena image (Figs. 4(a), (b), and (c)). However, the Splash and Tiffany images have produced more unsolvable cases than the Lena and Sailboat images. We observed from the image histograms that the grayscale values for the Splash and Tiffany images are mostly distributed near the extremum 0 and 255 (see Figs. 5(c) and (d)), whereas the grayscale values for the Lena and Sailboat images near the extremum are empty

(see Figs. 5(a) and (b)). This observation has further indicated the fact that the amount of unsolvable cases is proportional to the number of grayscale values near 0 or 255 in which require only a little alteration to go out of the grayscale range.

Table 1 shows the experimental results of all test images in Fig. 3 for their number of unsolvable pixels with various lengths of authentication code. We observed that under the same length of authentication code, the combinations of $l_r > l_b$ are easier to produce unsolvable pixels than the combinations of $l_r < l_b$. This is because the coefficient of the red channel in Eq. (1) is larger than that of the blue one. Therefore, the l_r -bit embedment of authentication code in the red channel has a greater effect on the grayscale value than the l_b -bit embedment in the blue channel, which will make the unsolvable cases more likely to occur. On the other hand,

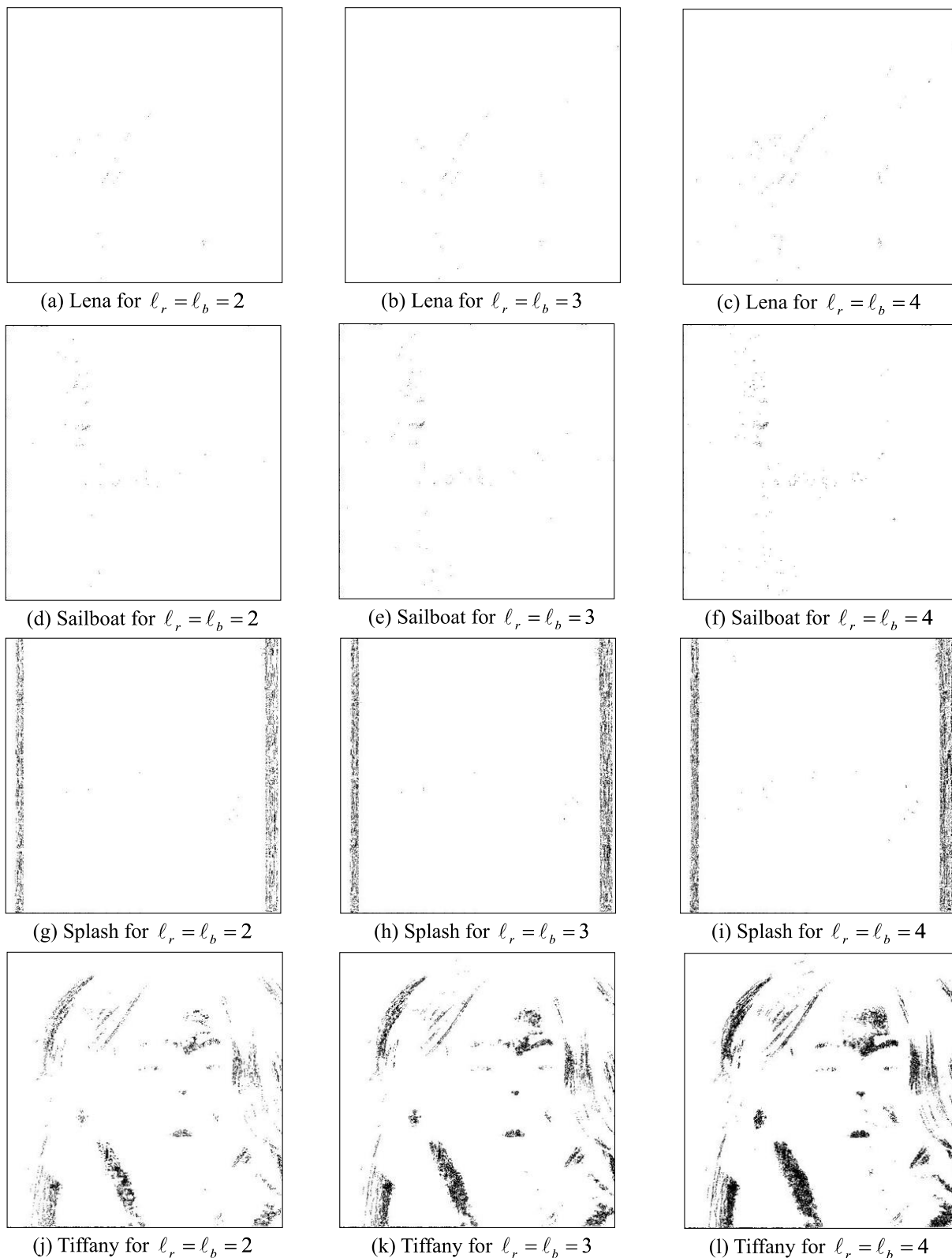


FIGURE 4. The distribution of unsolvable pixels in four test images after the embedment of authentication code with length $l_r = l_b = \{2, 3, 4\}$.

there are also some rare cases existed that longer authentication code reduces the occurrence of unsolvable pixels.

For example, the embedment of authentication code for the two-color combinations (3,3) and (3,2) in Lena image. This is

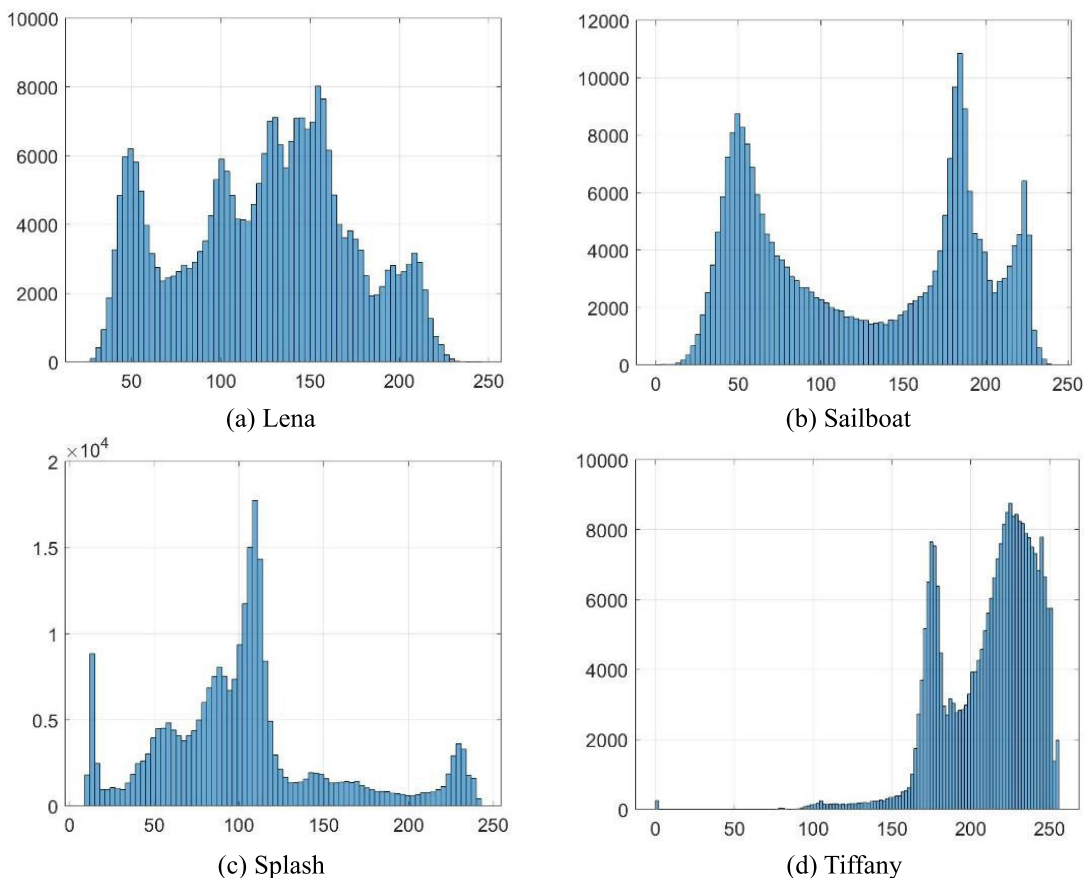


FIGURE 5. The grayscale histogram for the four test images.

possible because different lengths of authentication code can affect the generation of the hash code, and might further make the generated hash code close to its original corresponding pixel. Under such circumstance, the number of unsolvable pixels might drop in a longer authentication code.

2) THE DISTRIBUTION OF TYPE I AND TYPE II EMBEDMENT

When embedding the authentication code, the application of embedding types only considers whether the grayscale value of the pixel is in the list U , but does not consider whether it is an unsolvable pixel. Pixels with and without grayscale value in U will be applied Type II and Type I embedment, respectively. Therefore, it is possible that a pixel is solvable, but still applies the Type II method to embed the authentication code. Figure 6 shows the distributions of unsolvable pixels (Figs. 6(a), (b), (c)), and the pixels that are applied Type II embedment (Figs. 6(d), (e), (f)) for images Jet, Sailboat, and Tiffany with $\ell_r = \ell_b = 4$. As seen from Fig. 6, a small number of unsolvable pixels can turn a plenty number of solvable pixels to be processed with Type II embedment. For example, the Jet image only has nine unsolvable pixels (See Table 1); however, a total of 254 pixels are embedded with the Type II method. This situation happens more apparent in the Tiffany image, where 16,212 unsolvable pixels are observed,

and 57,208 pixels are embedded with the Type II method. The reason is that a significant number of color pixels of the Tiffany image with grayscale values are near the saturated values.

Table 2 presents the number of grayscale values in U , which is denoted by $|U|$, and the number of pixels that are processed with Type II embedment, which is denoted by N_2 . Note that our method can also apply the length combinations of $\ell_r \neq \ell_b$, but for simplicity, only the cases of $\ell_r = \ell_b$ are shown in the current discussion.

Table 2 reveals that even if $\ell_r = \ell_b = 2$, where shorter authentication code is embedded, test images can still produce unsolvable cases. When $\ell_r = \ell_b = 4$, we observe that the Splash image have covered approximately one-fourth of the grayscale values to be processed using Type II embedment. This shows the importance of having a perturbed value α used in Type II embedment to search for an optimal solution to remedy the unsolvable case.

B. IMAGE QUALITY COMPARISONS

While embedding the authentication code, we also need to ensure a good visual quality of the marked image after the embedment. Table 3 shows the PSNRs of marked images in several length combinations of $\ell_r = \{2, 3, 4\}$ and

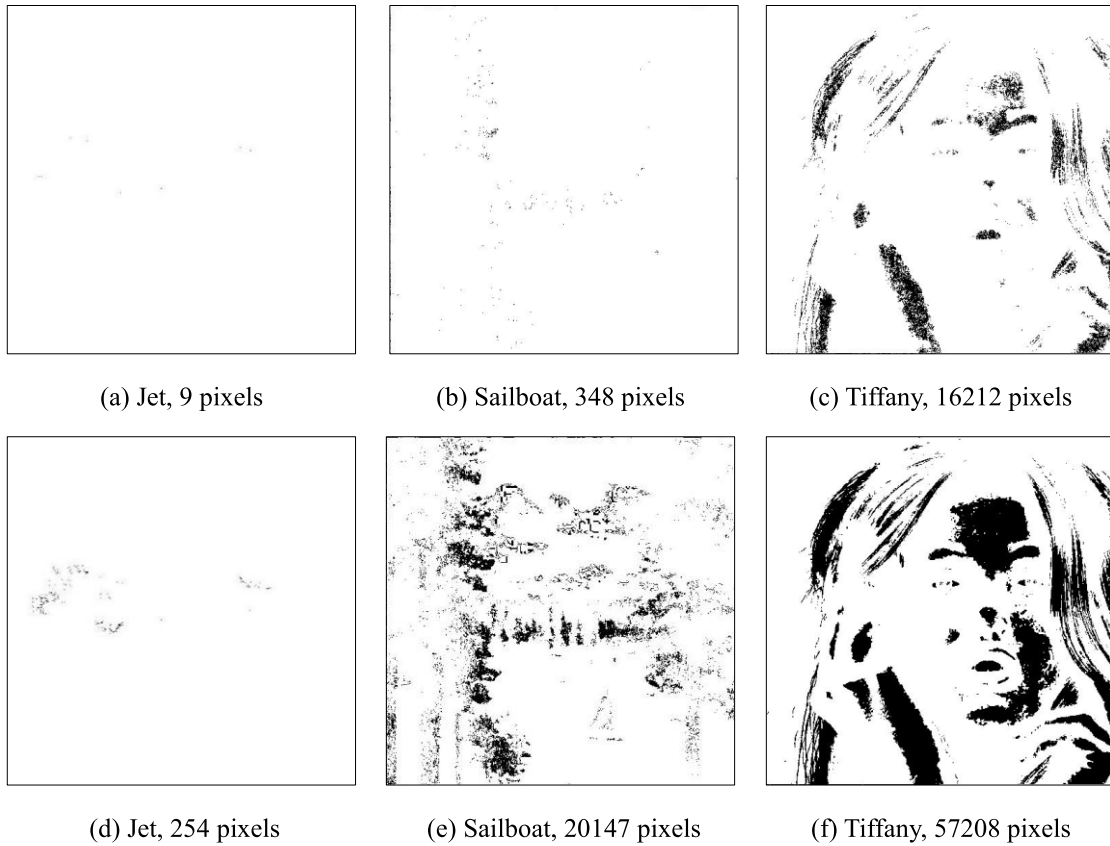


FIGURE 6. Distributions of unsolvable pixels ((a)-(c)) and pixels that are applied Type II embedding ((d)-(f)).

TABLE 2. U and N_2 for various lengths of authentication code.

Images	$(\ell_r, \ell_b) = (2, 2)$		$(\ell_r, \ell_b) = (3, 3)$		$(\ell_r, \ell_b) = (4, 4)$	
	$ U $	N_2	$ U $	N_2	$ U $	N_2
Lena	10	413	10	498	19	3677
House	13	1527	15	4082	20	4220
Jet	1	16	3	66	5	254
Peppers	41	26569	44	27551	46	29007
Sailboat	27	6137	35	10771	46	20147
Splash	63	54098	64	52636	71	58307
Tiffany	28	37074	28	37175	33	57208
Baboon	5	147	7	197	10	309

$\ell_b = \{2, 3, 4\}$. A longer authentication code will cause less PSNR value, as shown in Table 3. For example, the averaged PSNR has dropped significantly from 45.25 dB to 33.25 dB when doubles the embedding capacity from $\ell_r = \ell_b = 2$ to $\ell_r = \ell_b = 4$. In addition, a more unbalanced length of authentication code could introduce larger distortion on the colors. For example, a fixed-length combination $(\ell_r, \ell_b) = (3, 3)$ has a better averaged PSNR of 39.26 dB than the other two unbalanced-length combinations $(\ell_r, \ell_b) = (2, 4)$ and

$(\ell_r, \ell_b) = (4, 2)$ with averaged PSNR of 36.44 dB and 35.63 dB, respectively. Note that no matter how (ℓ_r, ℓ_b) are chosen, the grayscale versions of marked color images are always identical to that of the original color image.

Figures 7 (a)-(h) show the marked image with $\ell_r = \ell_b = 4$, i.e., each color pixel is embedded with 8-bit authentication code at most. As seen from these figures, the marked images still preserve good image quality, and all of them are visually indistinguishable with the original images.

TABLE 3. Image quality comparisons (PSNR in dB).

(ℓ_r, ℓ_b)	Lena	House	Jet	Peppers	Sailboat	Splash	Tiffany	Baboon	Average
(2,2)	45.20	45.23	45.19	45.38	45.18	45.50	45.16	45.18	45.25
(2,3)	41.42	41.34	41.50	41.72	41.47	42.06	41.84	41.37	41.59
(2,4)	36.18	36.16	36.04	36.69	36.25	37.15	36.95	36.16	36.44
(3,2)	40.88	40.83	41.18	41.17	41.02	41.59	40.84	40.86	41.04
(3,3)	39.04	39.14	39.18	39.42	39.30	39.79	39.18	39.05	39.26
(3,4)	35.18	35.33	34.97	35.83	35.54	36.13	35.85	35.29	35.51
(4,2)	35.15	35.46	35.17	35.87	35.77	36.43	35.81	35.39	35.63
(4,3)	34.58	34.95	34.66	35.25	35.20	35.69	35.17	34.80	35.03
(4,4)	32.80	32.98	32.94	33.62	33.35	33.74	33.65	32.96	33.25

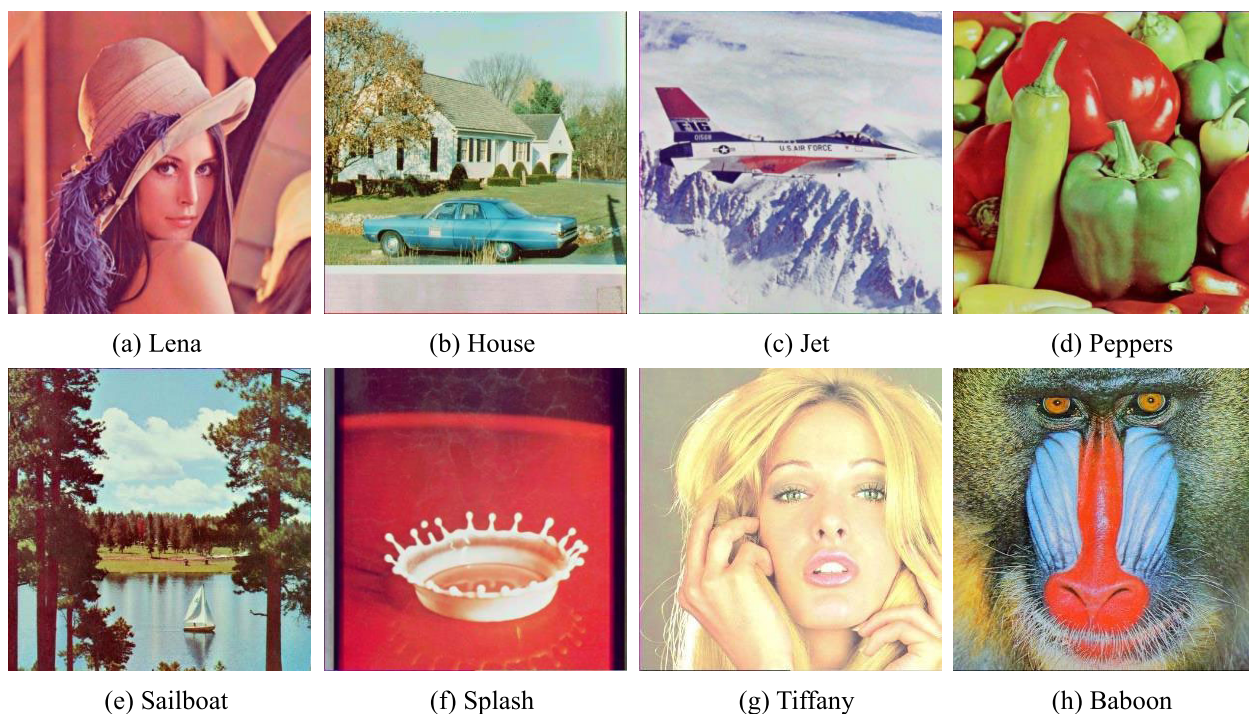


FIGURE 7. Marked color test images with $\ell_r = \ell_b = 4$.

It is interesting to note that, if the length of the authentication code is too long, it may cause obvious distortions to the perceptions of human visual system. For instance, Fig. 8 shows the results of Lena and Baboon images after embedding longer authentication code ($\ell_r = \ell_b = \{5, 6, 7\}$). We can actually see the images in Figs. 8(c) and (f) are seriously blurred after the embedment of authentication code with $\ell_r = \ell_b = 7$. Fortunately, an overlong authentication code is unnecessary in most of the real applications. In our method, setting $\ell_r = \ell_b = 4$ already produced a very satisfactory result, as will be represented in the next sub-section.

C. THE DETECTION OF TAMPERED IMAGES

A good authentication scheme should be capable of detecting malicious tampering, and the false negative and the false positive rates should be as small as possible. In this subsection, we conduct some tampering on the marked images of Lena, Jet, Tiffany, and Baboon to evaluate the detectability of our proposed method. Figure 9(a) is the tampered Lena image where a flower is placed on Lena’s hat. Figure 9(b) puts an extra aircraft in the Jet image. Figure 9(c) adds a pair of sunglasses on Tiffany’s eyes. Figure 9(d) embeds the MSB of Pepper image into the second MSB of the Baboon image. All four marked images are originally embedded with

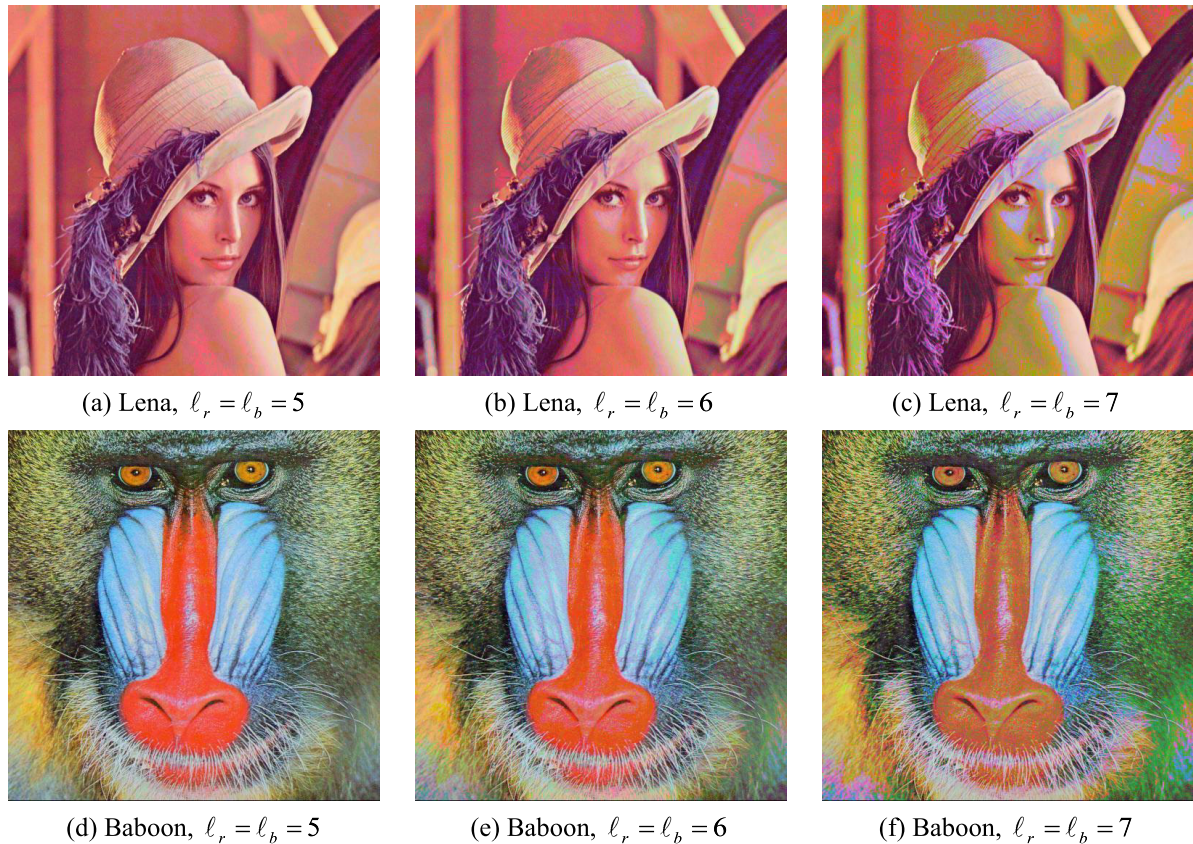


FIGURE 8. The impact of image quality for overlong authentication code.

TABLE 4. Details of detection results for the four test images.

Images	Lena	Jet	Tiffany	Baboon
Number of tampered pixels	9894	17056	21607	262144
Percentage of Tampering	3.77%	6.51%	8.24%	100%
Number of true positive (TP)	9843	16946	20529	260610
Number of true negative (TN)	252250	245088	240537	0
Number of false positive (FP)	51	110	1078	1534
Number of false negative (FN)	0	0	0	0
Detection rate	99.48%	99.36%	95.01%	99.41%

eight-bit authentication code with $l_r = l_b = 4$. The detection results of tampered regions are presented in a binary mask (Figs. 9(e)-(h)).

As seen from the figures, all tampered regions in the marked images are successfully detected in pixel-level. The sparse white dots appeared in the tampered regions are due to the hash collision, in which the extracted authentication code happens to be the same as the one that re-generated. When eight-bit authentication code is embedded into

a color pixel, the hash collision rate is approximately $1/256 \approx 0.39\%$.

Table 4 shows the statistical results of the tamper detections in Fig. 9, and the detection rate γ is calculated by

$$\gamma = \frac{\text{\# of true positive pixels}}{\text{\# of true positive pixels} + \text{\# of false positive pixels}} \quad (16)$$

TABLE 5. Comparisons of related works in various aspects.

Method	Authentication code generation	Embedding unit (pixel)	Length of authentication code	PSNR	Emphasize
[10]	Random Variables	4×4	4 bits	41.4 dB	Robust to Attacks
[11]	Variance and positions of blocks	4×4	32 bits	44.0 dB	Recoverable
[12]	Hash function	2×2	16 bits	44.6 dB	Recoverable
[13]	Hash function	2×2	2 bits	40.3 dB	Recoverable
[14]	Mean of image blocks	2×2 or 2×4	8 bits	44.2-51.1 dB	Recoverable
[15]	Hash function	4×4	4 bits	44.6 dB	Recoverable
Proposed	Hash function	1×1	4-8 bits	32.8-44.5 dB	Grayscale invariance

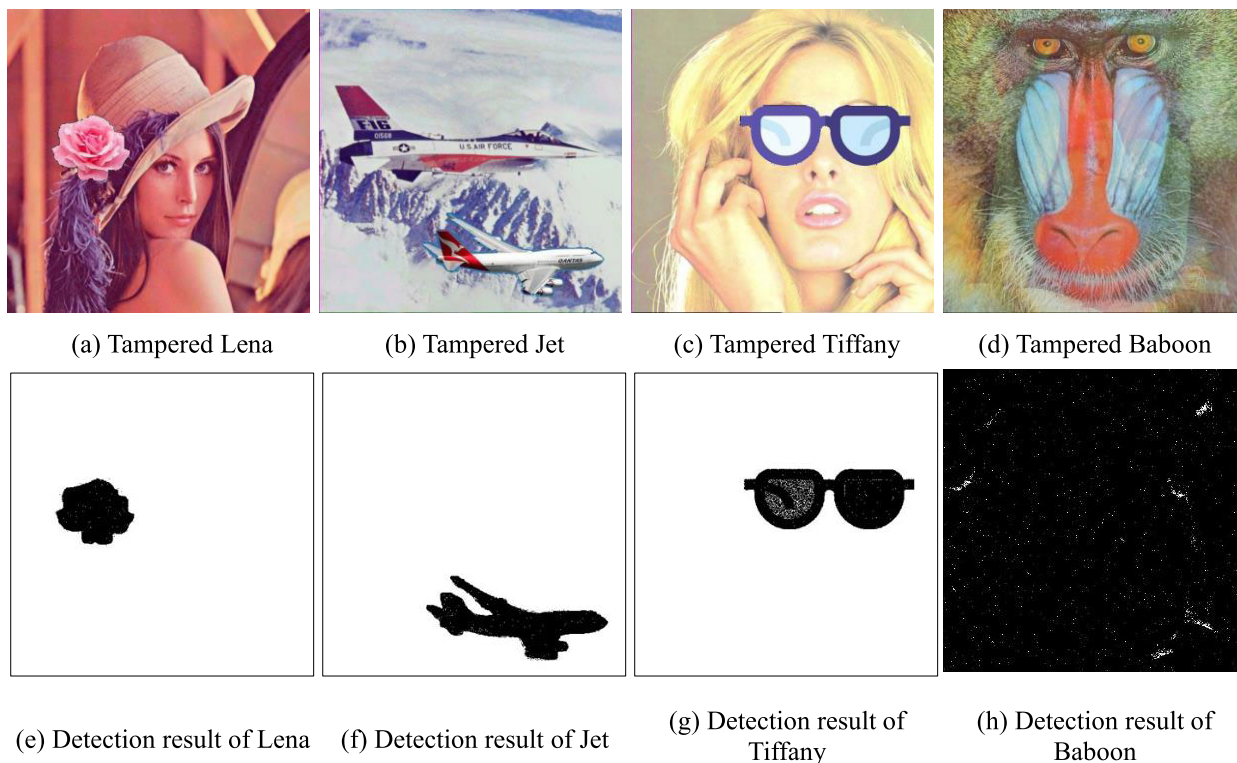


FIGURE 9. Tampered images and their detection results.

Note that Tiffany image has a lower detection rate (95.01%) than other test images due to it has more pixels in unsolvable cases during the embedment of authentication code. Also, the lengths of the authentication code for unsolvable pixels are shorter than those for solvable pixels, which could lead to the misdetection of tampered regions.

D. COMPARISON WITH OTHER METHODS ON VARIOUS ASPECTS

Sections III.B to III.C have compared the performance of the proposed method in terms of image quality and tamper detectability. In this section, we compare the proposed method with other related works, including Al-Otum’s [10], Huang and Jiang’s [11], Lin *et al.*’s [12], Chen *et al.*’s [13],

Belferdi *et al.*'s [14], and Molina-Gracia *et al.*'s [15] methods in various aspects, as shown in Table 5.

Table 5 shows that in addition to localization of tampered regions, recent authentication techniques for color images mainly focus on elevating the robustness to malicious attacks [10], or adding the recoverability feature to the proposed algorithms [11]–[15]. However, none of the existing work addresses the issues of grayscale invariance. Unlike methods [10]–[15] that localize the tamper regions in block-level, our method emphasizes the importance of grayscale invariance, and the tamper localization can be performed in pixel-level. The length of authentication code is adjustable in the proposed method, while the marked image quality can achieve a satisfactory result (32.8–44.5 dB).

IV. CONCLUSION

This paper presents a novel color image authentication scheme capable of grayscale-invariance for each pixel. When embedding the authentication code, the algorithm applies two methods to deal with the solvable and unsolvable cases of pixels, respectively. For solvable pixels, the authentication code will directly replace the LSBs of red and blue channels. As for unsolvable pixels, a perturb value is introduced to coordinate the values of the RGB channels and applied the LSB embedment in blue channel. Experimental results have shown the proposed method ensures a good image quality and it is able to detect the tampered regions at pixel level. Moreover, the generated marked image is guaranteed to have the same grayscale image as the original image.

REFERENCES

- [1] D. M. Uliyan, H. A. Jalab, A. W. A. Wahab, P. Shivakumara, and S. Sadeghi, "A novel forged blurred region detection system for image forensic applications," *Expert Syst. Appl.*, vol. 64, pp. 1–10, Dec. 2016.
- [2] J. Li, X. Li, B. Yang, and X. Sun, "Segmentation-based image copy-move forgery detection scheme," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 3, pp. 507–518, Mar. 2015.
- [3] C. S. Hsu and S. F. Tu, "Probability-based tampering detection scheme for digital images," *Opt. Commun.*, vol. 283, no. 9, pp. 1737–1743, 2010.
- [4] S. Trivedy and A. K. Pal, "A logistic map-based fragile watermarking scheme of digital images with tamper detection," *Iranian J. Sci. Technol., Trans. Electr. Eng.*, vol. 41, no. 2, pp. 103–113, Jun. 2017.
- [5] S. Prasad and A. K. Pal, "A secure fragile watermarking scheme for protecting integrity of digital images," *Iranian J. Sci. Technol., Trans. Electr. Eng.*, vol. 44, no. 2, pp. 703–727, Jun. 2020.
- [6] C.-S. Hsu and S.-F. Tu, "Image tamper detection and recovery using adaptive embedding rules," *Measurement*, vol. 88, pp. 287–296, Jun. 2016.
- [7] C. Qin, P. Ji, X. Zhang, J. Dong, and J. Wang, "Fragile image watermarking with pixel-wise recovery based on overlapping embedding strategy," *Signal Process.*, vol. 138, pp. 280–293, Sep. 2017.
- [8] A. Abdelhakim, H. I. Saleh, and M. Abdelhakim, "Fragile watermarking for image tamper detection and localization with effective recovery capability using K-means clustering," *Multimedia Tools Appl.*, vol. 78, no. 22, pp. 32523–32563, Nov. 2019.
- [9] P. Sutthiwan, Y.-Q. Shi, J. Dong, T. Tan, and T.-T. Ng, "New developments in color image tampering detection," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2010, pp. 3064–3067.
- [10] H. M. Al-Otum, "Color image authentication using a zone-corrected error-monitoring quantization-based watermarking technique," *Opt. Eng.*, vol. 55, no. 8, Aug. 2016, Art. no. 083103.
- [11] S. C. Huang and C. F. Jiang, "A color image authentication and recovery method using block truncation code embedding," *J. Mar. Sci. Technol.*, vol. 20, no. 1, pp. 49–55, 2012.

- [12] C.-H. Lin, T.-H. Chen, and C.-W. Chiu, "Color image authentication with tamper detection and remedy based on BCH and bayer pattern," *Displays*, vol. 34, no. 1, pp. 59–68, Jan. 2013.
- [13] C.-H. Chen, Y.-L. Tang, and W.-S. Hsieh, "Color image authentication and recovery via adaptive encoding," *Math. Problems Eng.*, vol. 2014, Aug. 2014, Art. no. 350753, doi: 10.1155/2014/350753.
- [14] W. Belferdi, A. Behloul, and L. Noui, "A Bayer pattern-based fragile watermarking scheme for color image tamper detection and restoration," *Multidimensional Syst. Signal Process.*, vol. 30, no. 3, pp. 1093–1112, Jul. 2019.
- [15] J. Molina-Garcia, B. P. Garcia-Salgado, V. Ponomaryov, R. Reyes-Reyes, S. Sadovnychiy, and C. Cruz-Ramos, "An effective fragile watermarking scheme for color image tampering detection and self-recovery," *Signal Process., Image Commun.*, vol. 81, Feb. 2020, Art. no. 115725, doi: 10.1016/j.image.2019.115725.
- [16] W. Hong, X. Zhou, D.-C. Lou, X. Huang, and C. Peng, "Detectability improved tamper detection scheme for absolute moment block truncation coding compressed images," *Symmetry*, vol. 10, no. 8, p. 318, Aug. 2018, doi: 10.3390/sym10080318.
- [17] K. Wattanachote, T. K. Shih, W.-L. Chang, and H.-H. Chang, "Tamper detection of JPEG image due to seam modifications," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 12, pp. 2477–2491, Dec. 2015.
- [18] W. Hong, X. Zhou, and D.-C. Lou, "A recoverable AMBTC authentication scheme using similarity embedding strategy," *PLoS ONE*, vol. 14, no. 2, Feb. 2019, Art. no. e0212802.
- [19] W. Hong, "Adaptive image data hiding in edges using patched reference table and pair-wise embedding technique," *Inf. Sci.*, vol. 221, pp. 473–489, Feb. 2013.
- [20] W. Hong and T.-S. Chen, "A novel data embedding method using adaptive pixel pair matching," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 1, pp. 176–184, Feb. 2012.
- [21] C.-K. Chan and L. M. Cheng, "Hiding data in images by simple LSB substitution," *Pattern Recognit.*, vol. 37, no. 3, pp. 469–474, Mar. 2004.
- [22] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 4th ed. London, U.K.: Pearson, 2017.
- [23] A. G. Konheim, *Computer Security and Cryptography*. Hoboken, NJ, USA: Wiley, 2007.
- [24] *SIPi Image Database*. Accessed: Oct. 20, 2020. [Online]. Available: <http://sipi.usc.edu/database/>



WIEN HONG received the M.S. and Ph.D. degrees from the State University of New York at Buffalo, USA, in 1994 and 1997, respectively. He is currently a Professor with the Department of Computer Science and Information Engineering, National Taichung University of Science and Technology, Taiwan. His research interests include steganography, digital watermarking, and image compression.



JEANNE CHEN currently works as a Professor with the Department of Computer Science and Information Engineering, National Taichung University of Science and Technology, Taiwan. Her research interests include data hiding, image cryptosystems, image compression, biomedical imaging, and multimedia design.



PEI-SHIH CHANG is currently an Undergraduate Student Researcher with the Department of Computer Science and Information Engineering, National Taichung University of Science and Technology, Taiwan. His research interests include image compression and steganography.



JIE WU has been a Senior Researcher and an Engineer with the School of Electrical and Computer Engineering, Nanfang College of Sun Yat-sen University, since 2019. She has incorporated several major projects on digital signal processing and applications of embedded system. Her research interests include development of single-chip microcomputer, image compression, data hiding, and image authentication.



TUNG-SHOU CHEN received the B.S. and Ph.D. degrees in computer science and information engineering from National Chiao Tung University, in 1986 and 1992, respectively. From 1994 to 1997, he was with the Faculty of the Department of Information Management, National Chin-Yi Institute of Technology, Taichung, Taiwan. From 1998 to 2000, he was the Dean of Student Affairs and a Professor with the Department of Computer Science and Information Management, Providence University, Taiwan. Since August 2000, he has been with the Department of Computer Science and Information Engineering, National Taichung University of Science and Technology, Taichung, where he is currently a Professor and the University Vice President. He is an IET Fellow. His research interests include image processing, data mining, information security, and bioinformatics.



JASON LIN received the Ph.D. degree in computer engineering from Texas A&M University, College Station, TX, USA, in 2020. He currently works as an Assistant Professor with the International School of Technology and Management and a Jointly Appointed Assistant Professor with the Department of Information Engineering and Computer Science, Feng Chia University, Taichung, Taiwan. His research interests include multimedia processing, artificial intelligence, and quantum cryptography.

• • •