A Color Image Authentication Method Using Partitioned Palette and Morphological Operations

Chin-Chen CHANG*, Member and Pei-Yu LIN**, Nonmember

SUMMARY Image authentication is applied to protect the integrity of the digital image. Conventional image authentication mechanisms, however, are unfit for the palette-based color images. Palette-based color images such as GIF images are commonly used for media communications. This article proposes a palette-based color image authentication mechanism. This novel scheme can guarantee the essentials of general authentication schemes to protect palette-based color images. Morphological operations are adopted to draw out the tampered area precisely. According to the experimental results, the images embedded with the authentication data still can preserve high image quality; specifically, the new scheme is highly sensitive to altered areas.

key words: image authentication, tamper-proofing, palette, morphological operation

1. Introduction

Due to the rapid advancement of computer technology, most media are now digitized. Traditional photographs can be quickly transformed into digital images using a scanner. Since digital images are represented by a sequence of bits, it is easy to modify digital images by changing the bit values. Hence, protecting the content of digital images is an urgent issue. Until now, engineers have proposed many image watermarking techniques for solving these problems. For different purposes, they have branched into two classifications: robust image watermarking techniques [1], [4], [6], [13], [15], [17] and fragile image watermarking techniques [2], [8]–[11], [14], [16].

Robust image watermarking techniques are used to protect ownership of the digital image. These techniques embed an ownership watermark into the digital image. If an intruder tampers with the watermarked image or announces ownership of the tampered image, the watermark can be extracted from the tampered image. Ownership can be verified by revealing the extracted watermark. Thus, this type of classification focuses on how the watermarking system can preserve the robustness of the image.

In contrast, the purpose of fragile image watermarking techniques is image authentication, that is, to ensure the integrity of the digital image. Whether or not the digital image has been modified is a more considerable point. A fragile image watermarking system must be able to sense tampering of the digital image as well as the area that has been tampered.

Fragile image watermarking techniques can be further divided into digital signature approaches [8], [10], [11], [14] and watermark-based approaches [2], [9], [16]. The main difference lies in whether the authentication data have been embedded into digital media. For digital signature approaches, a set of features is extracted from the digital image as the authentication data. Authentication data are kept as the independent message and can be used to verify a tampered image. Due to its independent property, the authentication data are not modified if the original image is tampered. This type of image authentication mechanism is expected to resist malicious modifications and tolerate modifications caused by image-processing incidents such as JPEG compression or blurring.

The watermark-based approaches also extract a set of features as the authentication data from the digital image and then embed the data into the image. After embedding, the digital image is referred to as an authenticated image. For content authentication, authentication data must be extracted from the tampered image. If parts of the authentication data are lost, the extracted authentication data are used to point out the modified areas. This type of image authentication scheme is used to extract authentication data and to point out the tampered areas after the content of the digital image is modified.

Fragile image watermarking techniques, however, do not fit palette-based color image compression. Palette-based color images such as GIF images are often used for media communications. Tzeng and Tsai [18] propose an authentication method to protect palette images using fragile watermarks and digital signatures. Their method classifies each pixel as embeddable or non-embeddable. The authentication data are embedded into embeddable pixels using a mapping function to obtain an authenticated image. For non-embeddable pixels, an exclusive-OR operation is applied to the authentication data and to the non-embeddable pixels in order to generate an extra digital signature. Their method offers high quality embedded images and protects the integrity of palette images. In addition, this method can efficiently locate the altered area in the authenticated image while it is being tampered. When there are fewer embeddable pixels than non-embeddable pixels, the payload of embeddable authentication data may be reduced, and the size of the digital image is minimized.
signature may increase.

To mitigate these potential problems, this article proposes a novel palette-based color image authentication mechanism. The new scheme consists of two components: a VQ-based spreading technique that can be used to protect the integrity of digital images and morphological operations applied to connect the modified points in order to contour the tampered areas. Comparing of Tzeng and Tsai's method [18], the major advantages of the new scheme are: 1) all pixels are embeddable, thus it needs not to generate the digital signature for the authentication of the protected palette image; 2) it can highlight the tampered areas in the authenticated image by using the morphological operations. Our experimental results demonstrate that the proposed scheme preserves high standards and offers quality protection for palette-based color images.

The rest of this paper is organized as follows. To begin with, we review the VQ-based spreading technique in Sect. 2. Then, we present our method in Sect. 3. In Sect. 4, we offer our experimental results to demonstrate the effectiveness of our new method. Finally, we make conclusions in Sect. 5.

2. Related Works

In this section, we briefly introduce the concept of Vector Quantization (VQ) [3]. Then, we go over the VQ-based spreading technique proposed by Jo and Kim [5].

2.1 Vector Quantization

VQ is a technique used to compress digital images. A codebook which consists of a set of codewords plays an important role in VQ technique. Each codeword refers to a corresponding index so that we can use the index to reach the codeword. All codewords come from training several candidate image blocks by the LBG algorithm [7]. In the encoding phase, an original image is divided into non-overlapping blocks with the same size. Therefore, we can find the most similar codewords for a sequence of non-overlapping blocks. By recording the corresponding indices for codewords, the image has been compressed by VQ technique. In the decoding phase, because of holding a sequence of indices, we can reconstruct the image by looking codewords up in the codebook according to those indices. We draw out the flowchart of VQ technique in Fig. 1.

2.2 VQ-Based Spreading Technique

In 2002, Jo and Kim [5] proposed a VQ-based spreading technique and applied it on image watermarks. The VQ-based spreading technique divides codewords of a codebook into three groups: \( G_{-1} \), \( G_{0} \), and \( G_{1} \). Each codeword exactly belongs to one of three groups. That we use a codeword belonging to \( G_{1} \) or \( G_{0} \) to encode a block means we embed a bit with value 1 or 0 in the block. Once we use a codeword in \( G_{-1} \), it indicates that we can’t embed any bit in this block.

For each input block, if its closest codeword doesn’t belong to \( G_{-1} \), then there must exist some similar codewords either in \( G_{1} \) or \( G_{0} \). Therefore, we can pick out a similar codeword from \( G_{1} \) or \( G_{0} \) according to the embedded value 1 or 0.

We continue to introduce how to divide codewords into three groups [5]. First of all, we shall decide a threshold \( T \) according to experiments. Furthermore, all codewords in the codebook are set to be unprocessed ones. We then randomly select a codeword \( c_{wi} \) from unprocessed codewords. For \( c_{wi} \), we find two closest codewords \( c_{wj} \) and \( c_{w'j} \) from the codebook. Here, we use the symbol \( D(c_{wx}, c_{wy}) \) to represent the Euclidean distance between two codewords \( c_{wx} \) and \( c_{wy} \).

In the second step, we have to calculate the distance values of \( D(c_{wi}, c_{wj}) \) and \( D(c_{wi}, c_{w'j}) \), respectively. According to the distance values and threshold \( T \), there exist some different cases. The first case is that both of \( D(c_{wi}, c_{wj}) \) and \( D(c_{wi}, c_{w'j}) \) are less than \( T \). In this case, we randomly pick up \( c_{wj} \) or \( c_{w'j} \) to form a pair referred to \( c_{wi} \). Assume that the chosen codeword is \( c_{wj} \). Then we can get a codeword pair \((c_{wi}, c_{wj})\). The second case is that only one distance value is less than \( T \). If \( D(c_{wi}, c_{w'j}) < T \), we can obtain a codeword pair \((c_{wi}, c_{w'j})\). In case that both two distance values are larger than \( T \), the codeword \( c_{wi} \) is assigned to \( G_{-1} \).

In the third step, for each codeword pair \((c_{wi}, c_{wj})\) or \((c_{wi}, c_{w'j})\), we randomly assign \( c_{wi} \) to \( G_{1} \) or \( G_{0} \). The other codeword \( c_{wj} \) or \( c_{w'j} \) is assigned to the opposite group. After going through three steps, all codewords in the codebook are divided into three groups.

We subsequently introduce the watermark embedded procedure. For each input block, we have to find the nearest codeword from the codebook. If the selected codeword belongs to \( G_{0} \) (or \( G_{1} \)) and the bit value of watermark is also 0 (or 1), then we take the index of the selected codeword as an output. If the found codeword belongs to \( G_{0} \) (or \( G_{1} \)) but the bit value of watermark is 1 (or 0), then we have to find a nearest codeword from the opposite group and take the index of found codeword as an output. In case that the found codeword belongs to \( G_{-1} \), then we take the index of the found codeword as an output.

After going through the watermark embedded procedure, the watermark has been embedded into VQ indices.
Note that groups \( \{G_{-1}, G_0, G_1\} \) must be kept secret for security reason.

3. The Proposed Scheme

This section describes the proposed method and the VQ-based spreading technique. Since VQ-based grayscale image compression and palette-based color image compression are similar, it is easy to utilize this technique [5] for palette-based color image compression.

The new scheme consists of an authenticating procedure and a detecting procedure. The first procedure embeds the authentication data into palette-based color image using the VQ-based spreading technique, while the second procedure explains how authorized users verify the tampered color image. The details of these two procedures are presented in the following sections.

3.1 Authenticating Procedure

The authenticating procedure (Fig.2) consists of three phases: palette partition, which divides the palette colors into two groups; authentication data generation, which produces a sequence of bits as the authentication data; and authentication embedding, which embeds the authentication data into the color indices.

Palette partition phase

In the first phase, we divide all the colors in the palette into two groups: \( G_0 \) and \( G_1 \). The partition algorithm is described as follows.

Step 1: Mark all colors in the palette as unprocessed colors.
Step 2: Select one color \( c_i \) from unprocessed colors randomly and mark \( c_i \) as a processed color.
Step 3: Find the closest color to \( c_i \) from unprocessed colors. Assume that \( c_j \) is the closest one. Then, mark \( c_j \) as a processed color.
Step 4: Assign \( c_i \) to \( G_1 \) (or \( G_0 \)) randomly and assign \( c_j \) to the opposite group.
Step 5: Check whether there exist any unprocessed colors or not. If so, go to Step 2.

In the palette partition phase, each color in the palette is assigned to \( G_0 \) or \( G_1 \); these partitions must be kept secret. Note that we do not generate group \( G_{-1} \), since this may cause an ambiguity problem in the verifying procedure. For instance, assume that there exists an authenticated color \( x \) belonging to group \( G_{-1} \). Once the authenticated image has been tampered, \( x \) may fall into \( G_0 \) or \( G_1 \). Subsequently, we may obtain an extra bit 0 or 1 in the verifying procedure. That is, the length of the extracted authentication data is not the same as the length of embedded authentication data. This ambiguity may lead to misjudgment of the tampering detection in the verifying procedure. Therefore, we only divide the colors of the palette into the partition \( \{G_0, G_1\} \).

Authentication data generation phase

The following steps outline how to generate a sequence of authentication data.

Step 1: Select a seed \( SK \) as the secret key.
Step 2: Apply a pseudo random number generator \( PRNG \) with the secret key \( SK \) to generate the authentication data \( S \).

Note that the produced authentication data are meaningless. If the authentication data are meaningful, such as public logos, then an attacker can obtain the partition \( \{G_0, G_1\} \) by analyzing the logo information and the authenticated image. Revealing the partition groups makes the whole system unsafe; this is the drawback of Jo and Kim’s method [5]. However, it is indispensable to make a logo of the company public when the mechanism is proposed for watermarking. Thus, the authentication data are meaningless for tamper-proofing.

Authentication embedding phase

The final phase embeds the authentication data into the palette-based color image for tampering proof. The authentication embedding phase is summarized in the follows.

Step 1: Set the symbol \( i \) equal to 0 and the symbol \( N \) equal to the image size.
Step 2: Input a color pixel \( x_i \), where \( 0 \leq i \leq N \), from a color image.
Step 3: Input a secret bit \( s_i \) from the authentication data \( S \).
Step 4: Check the value of \( s_i \). If \( s_i = 0 \), find the closest color to \( x_i \) from \( G_0 \); otherwise, find the closest color to \( x_i \) from \( G_1 \). Take the found index as the output.
Step 5: Check the value of \( i \). If \( i \neq N \), set \( i = i + 1 \) and go to Step 2.

Finally, we can obtain the color indices to form the authenticated palette-based color image.
3.2 Detecting Procedure

An authorized user is allowed to validate the embedded authentication data and to indicate which areas have been modified. We divide the detecting procedure into two parts: authentication extraction and tampering location. First, we extract the authentication data from the tampered palette-based color image. Since the tampered color pixel may fall into the group to which the authenticated pixel belongs, some modified color pixels may be judged as non-modified. Therefore, in the second part, morphological operations [12] are performed to correct errors (see Sects. 3.2.1 and 3.2.2 for details).

3.2.1 Authentication Extraction

In this subsection, we describe how an authorized user extracts the embedded authentication data with the secret information \( \{G_0, G_1\} \) and the secret key \( SK \) (see Fig. 3).

**Step 1:** Set symbol \( i \) equal to 0 and symbol \( N \) equal to the size of an indexed image.

**Step 2:** Input an color index \( x_i \) from a tampered image, where \( 0 \leq i \leq N \).

**Step 3:** Check the group that \( x_i \) belongs to. If \( x_i \) belongs to \( G_0 \), then output one bit with value 0; otherwise, output one bit with value 1.

**Step 4:** Check the value of \( i \). If \( i \neq N \), set \( i = i + 1 \) and go to Step 2.

After the extracting procedure, we can extract the authentication data \( S' \) from the tampered image.

3.2.2 Tampering Location

Subsequently, we describe how to locate the modified area. We first produce an original authentication data \( S \) by feeding \( SK \) into the PRNG. Using \( S \) and \( S' \), we can detect the modified area by locating the difference location panel \( P \) as follows:

\[
P = S \oplus S',
\]

where \( \oplus \) is an exclusive-OR operator and \( P \) indicates the tampered area (see Fig. 4).

If the output value of the exclusive-OR operator is "1", then the color pixel has been modified. If the output value of the exclusive-OR operator is "0", however, this does not necessarily mean that the color pixel is unmodified. For the second pixel of the modified area, both the authentication data and the extracted data are "0" in the same position. This means that this modified color pixel and the original pixel fall into the same group; this situation is indispensable. Here, it is difficult to directly determine the modified area from the difference location panel \( P \). Therefore, we adopt morphological operations [12] to locate the modified area.

Morphological operations are used to eliminate the isolated noise and link the contour of the object. Observing the difference location panel \( P \) in Fig. 4, we find that the modified area possesses a high density of bits with values of "1". As a result, we can apply the dilation operator and the erosion operator to contour the modified area by connecting the bit with value "1". Note that the executing order of these two operators will cause different results. In this paper, we first execute the dilation operator followed by the erosion operator to achieve our goal.

**Dilation operator phase**

The dilation operator phase creates a 3×3 window that slides from top to bottom and from left to right in difference location panel \( P \) to dilate the bit with value "1". The value of the middle bit in the sliding window is determined by whether there exists any bit with value "1" in this sliding window. We use the difference location panel in Fig. 4 to perform the dilation operator demonstrated in Fig. 5. For instance, the sliding window masks the first bit of the difference location panel \( P \) at the beginning. Since all bit values in the sliding window are "0", the first bit in the difference location panel
with dilation $P_d$ is set as “0”. When the sliding window moves to $(1, 1)$, we set the bit value to $(1, 1)$ in the difference location panel with dilation $P_d$ as “1”, since there exists one bit with value “1” (see Fig. 5).

**Erosion operator phase**

The erosion operator phase is performed on the difference location panel with dilation $P_d$. The process is nearly identical to the dilation operator phase, except that the value of the current bit is set to “1” when all bits in the sliding window are “1”. If there exists any bit with value “0” in the sliding windows, then the value of the current bit is set to “0” (see Fig. 6).

After performing the morphological operations, we indicate the bits with value “1”. According to the connection of the bits with value “1”, we can locate the modified areas.

4. Experimental Results

In this section, we demonstrate the experimental results and evaluate the performance of the novel scheme. The peak-signal to noise rate ($PSNR$) is used to measure the image quality of the authenticated and tampered images. The $PSNR$ formula is described as follows:

$$PSNR = 10 \log_{10} \left( \frac{255^2}{MS\:E} \right) \text{dB}. \quad (2)$$

The mean square error ($MS\:E$) of an image with $H \times W$ pixels is defined as

$$MS\:E = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} (o_{ij} - \overline{o}_{ij})^2, \quad (3)$$

where $o_{ij}$ is the original pixel value and $\overline{o}_{ij}$ is the processed pixel value. The test images used in the simulations are shown in Fig. 7. The size of test images is set to $512 \times 512$ pixels.

For any image authentication mechanism, the quality of the authenticated images and the ability to detect tampered areas are major concerns. After performing the authenticating procedure, we obtain the authenticated images shown in Fig. 8. From the view of human visual perception, it is difficult to distinguish the differences between the original color images and the authenticated color images. In particular, according to the $PSNR$ values of the authenticated images in Fig. 8, we can detect only slight distortion of the images produced by the novel scheme.

Subsequently, we show the ability of the new method to detect the tampered area. We intrude one plane into a smooth area and one plane into a complex area of the Jet image to make it tampering-proof (see Fig. 9(a)). Figure 9(b) displays the detected pixels from the tampered image; the white pixels indicate tampering. The detecting procedure cannot completely determine altered areas, since the tampered color pixels may fall into the group to which the authenticated pixels belong; that is, some modified color pixels are judged as non-modified areas. The morphological operations are then used to improve this phenomenon. Figure 9(c) shows the enhanced image; the white areas indicate distinct
alteration. The results show that the proposed scheme can precisely locate the two intruded planes.

We use another test image to demonstrate the ability of the new scheme to detect tampering. Figure 10 (a) presents the tampered image. We alter the complex area (hair) and the smooth area (shoulder) of the authentication image. The white pixels in Fig. 10 (b) demonstrate the detected result (Fig. 10 (a)). The new scheme is capable of detecting the tampered region in both the smooth and complex areas.

Moreover, Fig. 10 (c) precisely indicates the detected result after morphological operations.

In the next experiment, we illustrate the sensitivity of the novel scheme. After ranging out areas of the authenticated image, Gaussian blurring is used to blur the selected range. To show the superiority of our method, we select a complex image, Pepper, as the test image. For the authenticated image, we blur the text areas to obtain the tampered image (Fig. 11 (a)). From the results, we cannot observe any alteration in the tampered image, although it is indeed modified.

Figure 11 (b) shows the final result after detecting the modified pixels, indicated in white. The new scheme is able to verify any slight alteration in both the smooth area and the edge area. Figure 11 (c) shows the advantage of the novel scheme with the morphological operations. According to the marked white pixels, our scheme can precisely locate the blurred areas.

For a smooth test image, Tiffany, we blur text areas to obtain the tampered image (Fig. 12 (a)). The tampered areas are slight and imperceptible. The white pixels in Fig. 12 (b) indicate the tampered results from the detecting procedure. Figure 12 (c) presents the emphatic resultant adapting morphological operations. This demonstrates that the new scheme is capable of detecting slight and insignificant alterations in both smooth and complex images. Due to the high sensibility of the proposed scheme, we can hide the message by ranging out the text areas and blurring them. To extract the hiding message, the receiver only needs to proceed through the detecting procedure.

According to Figs. 9 (b), 10 (b), 11 (b), and 12 (b), the
Fig. 11 Detection results of a complex image (Pepper): (a) tampered image, (b) resulting image without morphological operations, (c) resulting image with morphological operations.

Fig. 12 Detection results of a smooth image (Tiffany): (a) tampered image, (b) resulting image without morphological operations, (c) resulting image with morphological operations.

new scheme can effectively detect the altered regions of tampered images. Considering the worst case, the density of white pixels may be low after the detecting procedure. To enhancing this, we extend the window size of the morphological operations to cover the tampered pixels sensitively; this allows the tampered regions to be highlighted more easily. However, a larger window size may cause non-modified pixels near the tampered regions to be regarded as altered. According to the experimental results, a $3 \times 3$ sliding window is suitable for general images, balancing benefits with the tradeoff.

5. Conclusions

Engineers have proposed many image authentication mechanisms to protect the contents of digital gray-level or color images, but few of these approaches are designed for palette-based color images. Since palette-based color images such as GIF images are commonly used for media communications, protecting the content of these images is an important issue. We propose a novel authentication mechanism for palette-based color images that applies the VQ-based spreading technique and morphological operations to draw out the tampered area precisely. The experimental results show that the authenticated color images still possess high quality. Thus, the new method can be practically applied in the real world.

References


Chin-Chen Chang received his BS degree in applied mathematics in 1977 and his MS degree in computer and decision sciences in 1979, both from the National Tsing Hua University, Hsinchu, Taiwan. He received his Ph.D in computer engineering in 1982 from the National Chiao Tung University, Hsinchu, Taiwan. During the academic years of 1980–1983, he was on the faculty of the Department of Computer Engineering at the National Chiao Tung University. From 1983–1989, he was on the faculty of the Institute of Applied Mathematics, National Chung Hsing University, Taichung, Taiwan. From 1989 to 2004, he has worked as a professor in the Institute of Computer Science and Information Engineering at National Chung Cheng University, Chiayi, Taiwan. Since 2005, he has worked as a professor in the Department of Information Engineering and Computer Science at Feng Chia University, Taichung, Taiwan. Dr. Chang is a Fellow of IEEE, a Fellow of IEE and a member of the Chinese Language Computer Society, the Chinese Institute of Engineers of the Republic of China, and the Phi Tau Phi Society of the Republic of China. His research interests include computer cryptography, data engineering, and image compression.

Pei-Yu Lin received the MS degree in computer science and information engineering from National Chung Cheng University, Chiayi, Taiwan in 2004. She is currently pursuing her Ph.D degree in computer science and information engineering from National Chung Cheng University, Chiayi, Taiwan. Her current research interests include digital watermarking, image protection, data mining, and information security.