Hindawi
Security and Communication Networks
Volume 2022, Article ID 4542705, 13 pages
https://doi.org/10.1155/2022/4542705

Research Article

Compression Domain Reversible Robust Watermarking Based on Multilayer Embedding

Qianwen Li,[1] **Xiang Wang**,[2] **and Qingqi Pei**[1]

[1]*School of Telecommunications Engineering, Xidian University, Xi'an 710071, China*
[2]*School of Cyber Engineering, Xidian University, Xi'an 710071, China*

Correspondence should be addressed to Xiang Wang; wangxiang@xidian.edu.cn

Received 23 May 2022; Revised 22 June 2022; Accepted 25 July 2022; Published 28 August 2022

Academic Editor: He Li

Copyright © 2022 Qianwen Li et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The integration of data sensing, communication, and computing (SCC) is a requirement for the IoT to provide a high quality of service. However, there are still data security issues in SCC integration, where data are vulnerable to leakage during sensing, communication, and computing, and the leaked data are difficult to be copyrighted and traced to the source of leakage. In this study, we consider the copyright protection problem in multimedia data transmission and propose a robust reversible watermarking algorithm based on multilayer embedding in the compression domain. In the first stage, the robust watermark is embedded into the mid-frequency coefficients, and then, the auxiliary information to revert the robust embedding is embedded into the high-frequency coefficients. To improve robustness and reduce embedding distortion, we propose a coefficient selection method, by which the watermark and the recovery information are embedded in different DCT-quantized coefficients according to the texture complexity. Experiment results indicate that the proposed method performs better than some state-of-the-art RRW methods.

1. Introduction

As one of the strategic emerging industries with important global development, the Internet of Things (IoT) technology is rapidly penetrating into various application areas, from people's daily lives to national strategic industries, and in many ways, also driving the world's development progress. IoT uses communication technologies such as local networks or the Internet to connect sensors [1–3], controllers, machines [4, 5], people, and objects together in new ways, forming a network where people are connected to things and things are connected to things. The integration of data sensing, communication, and computing (SCC) is a requirement for the IoT to provide a high quality of service. However, there are still security issues in SCC integration [6–10], such as node location leakage in data sensing, data leakage in data communication, and authentication. These privacy and security issues have an impact on the development of IoT. Digital watermarking, as a lightweight algorithm, can effectively solve the problems such as data authentication after data leakage. Existing watermarking algorithms can be classified into the robust watermarking algorithm, reversible watermarking algorithm, and robust reversible watermarking algorithm according to the characteristics of watermarking.

Robust watermarking algorithms require high robustness, which means it is possible to extract the correct watermarks when the marked image is attacked [11]. Robust watermarking schemes based on spatial domain mainly utilize some statistical characteristics of the images (e.g., focus or centroid of the image). Because the spatial domain is more intuitive and simple, most of the studies focus on it. However, the transform domain is more robust to compression and quantify attacks. Thus, the transform domain-based method achieved more attention [12, 13].

However, the robust watermark always introduces irreversible distortions to the cover images, which cannot be tolerated in some special applications, such as medical image system, law enforcement, and military image. Reversible watermarking [14] is proposed to overcome this issue, by

which the cover image could be recovered after watermark extraction. There are many relative researches have been proposed so far, mainly including four categories: compression based [15–17], integer transform based [18–21], histogram shifting (HS) based [14, 22–24], and prediction-error expansion (PEE) [25]. Compression-based technology exploits the embedded space by lossless compression of a specific area of the cover image, and then, this area is replaced by the compressed image and the watermark. This kind of method cannot efficiently use the redundant information of the image itself, so its embedding efficiency is low. Different expansion algorithm is one of the representative schemes of integer transform-based researches. This algorithm groups pixels into pairs and the watermarks are embedded into the expansion error of pixel pairs. Later works involve changing pixel pairs into pixel blocks to increase capacity. The histogram translation algorithm embeds watermarking information by changing the histogram formed by an attribute of the host image. This attribute may be only the value of pixels, or it may be the prediction error of the image. Prediction-error expansion (PEE) is one of the representative works. The histogram is constructed by counting the prediction errors of the pixels, and then, a part of the area in the histogram is shifted to create space for embedding the watermark. The method is characterized by large embedded capacity and low complexity of the algorithm. However, all these methods are so fragile that the watermarking cannot be extracted after being attacked.

For solving the poor robustness of reversible watermarking, a new technique named robust reversible watermarking (RRW) is proposed. By RRW, the cover images can be recovered if the marked images do not suffer from attacks, and the robust watermarks are robust against some normal attacks such as image compression, geometric attacks, and unavoidable addition of random noise. As a result, at the decoder side, both the watermarks extraction and image recovery can be achieved in the case of no attacks.

To our knowledge, the RRW methods mainly include two categories: generalized histogram shifting (GHS) and multilayer watermarking (MLW). GHS achieves good robustness by increasing the shifting distance of partial bins in the statistical quantity histogram, by which extra space for tolerance is created around the embedded position. When the image is distorted, the watermark is supposed to be extracted correctly as long as the distortion of the histogram does not exceed the fault-tolerant area. One of the representative results of such methods is histogram rotation (HR) [26] proposed by Bender et al., which is based on the patchwork theory [27] and modulo-256 addition. Their scheme can achieve reversibility and robustness by using a circular interpretation of bijective transformations. However, to avoid the overflow/underflow problems, the so-called salt-and-pepper noise occurs. Later on, Ni et al. [28] improved Zeng et al.'s scheme by utilizing different bit-embedding strategies for groups of pixels with different pixel grayscale value distributions together with error correction codes. Experimental results show this method had good robustness in the absence of salt-and-pepper noise. The other is redundant histogram shifting (RHS) [29], which is based on the traditional histogram shifting and gets more redundancy by expanding the distance of shifting. Experiment results show a good performance of robustness. For this method can be extended to the transform domain to improve robustness, some RRW schemes have been developed based on this framework [30, 31]. However, these methods require a lot of additional information (e.g., location maps) to be transmitted over additional secure channels. Therefore, no longer suitable for many applications.

As for MLW, Coltuc proposed the relevant framework in reference [13], which contains two embedding stages: the robust embedding stage and the reversible embedding stage. As shown in Figure 1, the first is the robust watermarking stage, where we embed the watermark information W by the robust watermarking algorithm. Then, the reversible watermarking algorithm is used to embed additional information RI to recover the original image. Finally, the watermarked image is generated. In the case of no attacks, the robust watermark can be extracted from the marked image, and the decoder exactly recovers the cover image. In the case of attacks, the reversibility is lost, but the robust watermark is also supposed to be extracted. One of the advantages of this framework is that the existing excellent robust and reversible watermarking methods could be directly introduced because of the two independent stages. However, Coltuc et al.'s method has its own shortcomings. The reversible watermark in stage 2 directly consists of the distortion from the first stage. Because the watermark information is too large, a reversible watermark (RW) scheme with a large capacity is required, but this will cause a considerable distortion. In addition, the robust and the reversible watermarking use the same embedding domains; thus, the two stages are not independent and interfere with each other.

As shown above, there are a large number of RRW methods, but few algorithms are proposed specifically for JPEG image. In this study, we propose a new reversible robust watermarking scheme for JPEG images. The reversible watermarks and robust watermarks are embedded into the coefficients of different frequency bands, respectively. Meanwhile, the algorithm preferentially selects the region with complex image texture for robust watermark embedding and the smooth texture region for reversible watermark embedding. That will cause less distortion. Moreover, a new robust watermarking is proposed in the study, which needs less information to invert both the robust and reversible watermarking in the second stage.

The rest of this study is organized as follows: in Section 2, the framework of MLW is introduced in detail. In Section 3, the proposed scheme is presented in detail. Experiment results compared with Zeng et al.'s RRW method are given in Section 4. Finally, conclusions are drawn.

2. Preliminaries

The MLW algorithm [12] proposed by Coltuc et al. is briefly introduced in this section. The framework of MLW includes two embedding stages: reversible embedding stage and robust embedding stage.

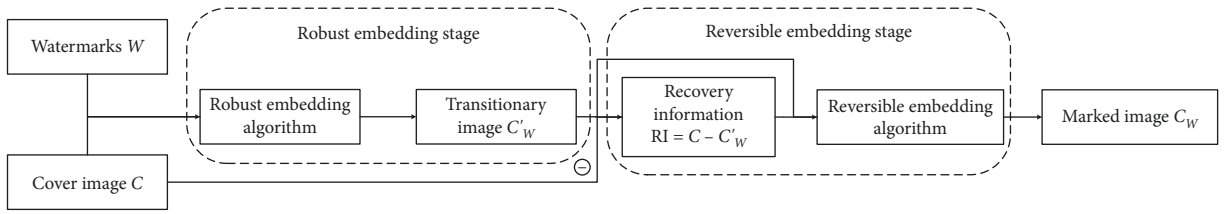

Figure 1: The framework of Coltuc et al.'s RRW method.

As shown in Figure 1, a watermark W is embedded into the cover image C by a robust watermarking method to generate a transitionary image C_W', and then, the recovery information $R_I = C - C_W'$ is hidden in the image C_W' by the reversible watermarking method. In the decoder, as Figure 2 shows, the recovery information R_I can be extracted from C_W as well as recovering the transitionary image C_W'. Then, the original image C can be recovered according to $C = C_W' + R_I$.

2.1. Embedding Procedure

2.1.1. Robust Embedding. The cover image is divided to 8×8 blocks and the DCT coefficients of each block are computed. One bit of watermark can be embedded into two coefficients from the middle-band frequencies of one block. For instance, a bit of "1" is embedded if the difference of these two coefficients is positive and "0" otherwise. Specifically, for two coefficients c_1 and c_2 of a 8×8 block, one bit of watermark w is embedded as follows:

$$(c_1', c_2') = \begin{cases} (c_1, c_2), & \text{if } c_1 - c_2 > 0 \text{ and } w = 1, \\ (c_2, c_1), & \text{if } c_1 - c_2 \le 0 \text{ and } w = 1, \\ (c_1, c_2), & \text{if } c_1 - c_2 \le 0 \text{ and } w = 0, \\ (c_2, c_1), & \text{if } c_1 - c_2 > 0 \text{ and } w = 0, \end{cases} \tag{1}$$

where c_1' and c_2' denote the marked coefficients. Figure 3 shows two examples of the embedding process, and the two framed coefficients are the selected coefficients c_1 and c_2. In the case of $c_1 > c_2$, when embedding bit 0, the positions of two coefficients are swapped; otherwise, the positions remain unchanged. In the case of $c_1 \le c_2$, when embedding bit 0, the positions of two coefficients remain unchanged; otherwise, they are swapped.

2.2. Reversible Embedding. As shown in Figure 1, in this stage, the recovery information $R_I = C - C_W'$, which is the difference between C and C_W', is first generated. Then, R_I is losslessly compressed to get the reversible watermark R. The RRW relies on the reversible watermarking stage's embedding capacity; thus, a reversible watermarking algorithm with high capacity is demanded. Specifically, C_W' is partitioned into pairs of pixels, and each pair (c_1', c_2') is transformed as follows:

$$\begin{cases} c_1'' = (m+1)c_1' - mc_2', \\ c_2'' = (m+1)c_2' - mc_1', \end{cases} \tag{2}$$

where $m > 1$ is a fixed integer to control EC.

The watermark $r \in [-m, m]$ is embedded as follows:

$$(c_{w1}, c_{w2}) \longrightarrow (c_1'' + r, c_2''), \tag{3}$$

and after all bits of the compressed information R are embedded, the marked image is generated.

2.3. Extracting Procedure. In the case of the marked image C_W suffers no distortion, as shown in Figure 2, the reversible extracting procedure is first implemented. The same as the embedding procedure in stage 2, the marked image C_W is partitioned into pairs of pixels, and for each pixel pair, the compressed information R is extracted as follows:

$$r = \left((m+1)c_{w1} + mc_{w2}\right) mod\,(2m+1), \tag{4}$$

and the extracted watermark R is then decompressed to get the recovery information R_I, and the transitionary image C_W' is recovered as follows:

$$\begin{cases} c_1' = \lfloor \dfrac{(m+1)c_{w1} + mc_{w2}}{2m+1} \rfloor, \\ \\ c_2' = \lfloor \dfrac{(m+1)c_{w2} + mc_{w1}}{2m+1} \rfloor, \end{cases} \tag{5}$$

where $\lfloor * \rfloor$ is the floor operation.

After reversible decoding, the watermark W could be extracted from the transitionary image C_W' by a robust extracting process. Specifically, C_W' is divided into 8×8 blocks and the DCT coefficients of each block are computed. For each block, the watermarking bit is extracted according to the difference between the two selected coefficients. If this difference is positive, the embedded information is "1," and otherwise, the embedded information is "0." Finally, the original image is recovered as $C = R_I + C_W'$.

In the case of marked image C_W with distortion, the original image C and the transitionary image C_W' could not be restored, but the watermark W can be extracted from the distorted image directly, as shown in Figure 4.

3. Proposed Method

As aforementioned, Coltuc et al.'s algorithm has its own shortcomings. First, the two embedding stages shared the same embedding domain, which leads to additional distortion. At decoder side actually will face two kinds of noises: the attacks from outside and the interference of the reversible embedding. Second, the spatial domain embedding process is not suitable to be directly introduced to JPEG images.

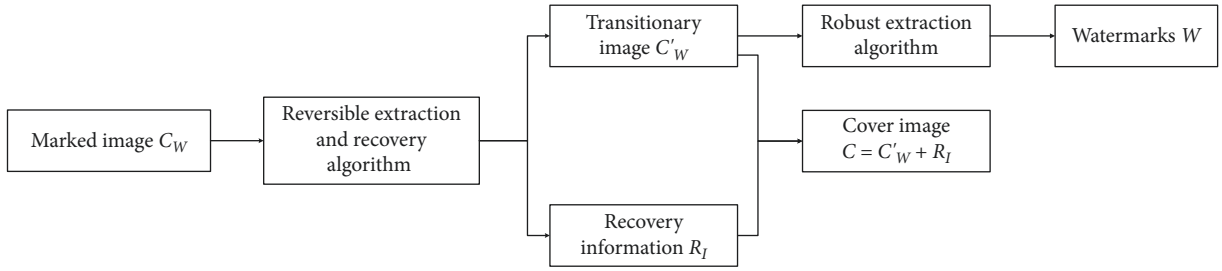

Figure 2: Decoder without distortion.

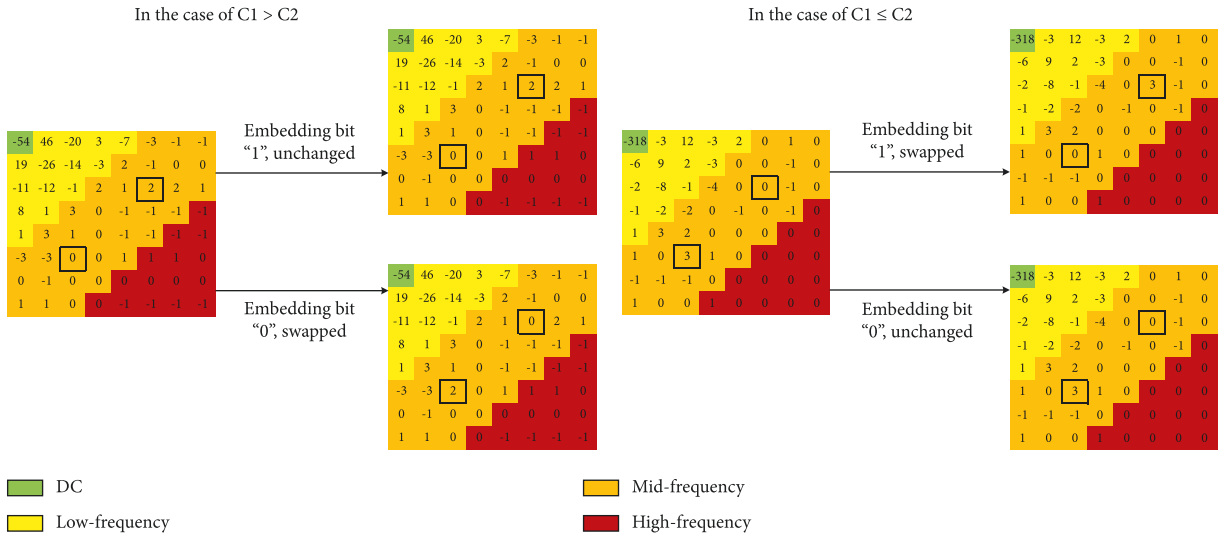

Figure 3: Robust embedding.

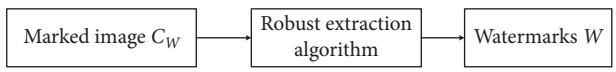

Figure 4: Decoder with distortion.

To this end, a RRW scheme for JPEG images is presented in this section. This scheme focuses on the compressed domain and embeds watermarks in the DCT-quantized coefficients. The proposed method follows the Coltuc et al.'s MLW framework and overcomes some shortcomings.

As Figure 5 shows, in the JPEG compression process, the source image is divided into 8×8 sized nonoverlapping blocks, for the pixels in each block, the DCT will be performed. Then, the DCT coefficients are quantified by using a quantization table, which is the cause of the image loss. After the entropy coding, the JPEG image is formed. The proposed method is applied to the quantized coefficients, and these coefficients are divided into two independent embedding domains, one for robust embedding and the other for reversible embedding.

3.1. Embedding Procedure. The embedding procedure roughly contains three steps: coefficients preprocessing, robust embedding, and reversible embedding. This section will introduce them one by one.

3.2. Stage 1 (Coefficients Selection). The DCT-quantized coefficients of the block B_i are zigzag arranged to generate a vector C_i. As shown in Figure 6, the vector could easily be divide into four regions: the low frequency C_i^l marked in yellow, the medium frequency for robust embedding C_i^{ro} marked in orange, the medium frequency for reversible embedding C_i^{re} marked in red, and the high frequency C_i^h marked in bronzing. The energy is concentrated in the low frequency coefficients C_i^l, which always refers to the part with large scale information (e.g., the background region). Thus, data embedding into low-frequency coefficients can generate visible artifacts. Coefficients in this part will stay unchanged. For the high-frequency region C_i^h, which always represents the small-scale detail information of the images, such as noise, edge, and jump part). The coefficients in this part will also remain unchanged because changing them affects the efficiency of entropy coding. Moreover, we use C_i^h to predict the texture complexity of each block as follows:

$$B_i \in \begin{cases} B^s, & if\ g \le G; \\ B^c, & if\ g > G; \end{cases} \tag{6}$$

where we set $G > 0$ as the threshold to distinguish smooth block B^s and complex block B^c, and $g = \sum_{c_k \in C^h} |c_k|$. is used to predict the texture complexity of each block. Obviously, the larger g is, the more complex block texture, which indicates

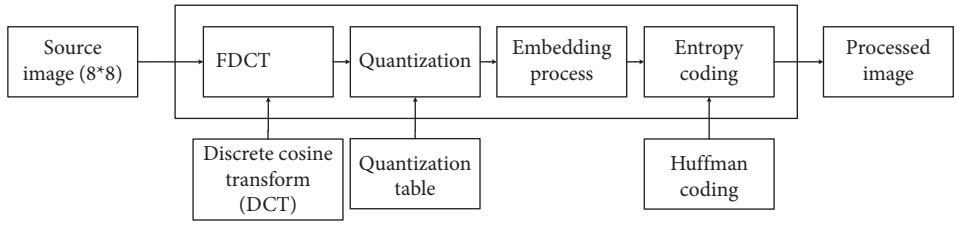

Figure 5: The position of our embedding process throughout the JPEG compress procedure.

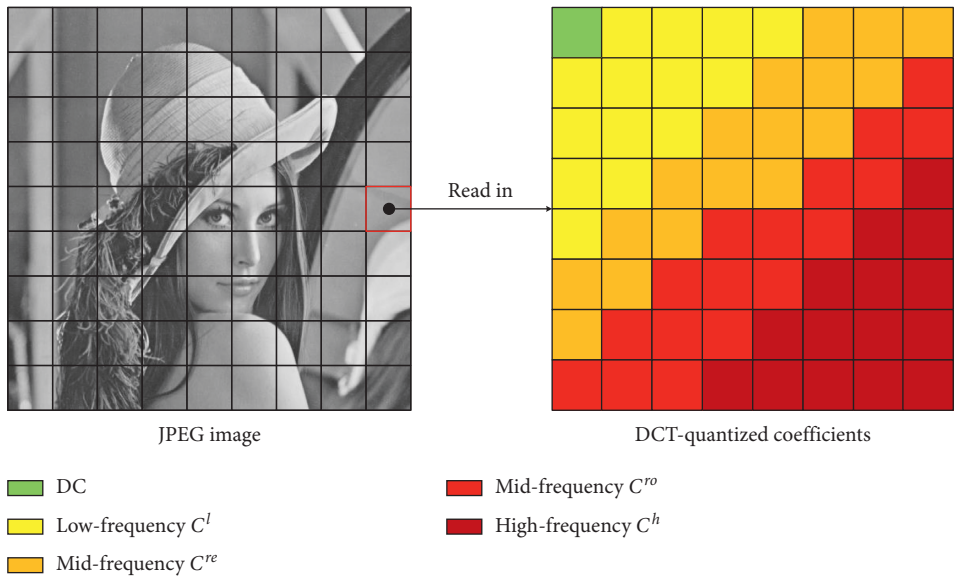

Figure 6: Frequency segmentation in a block.

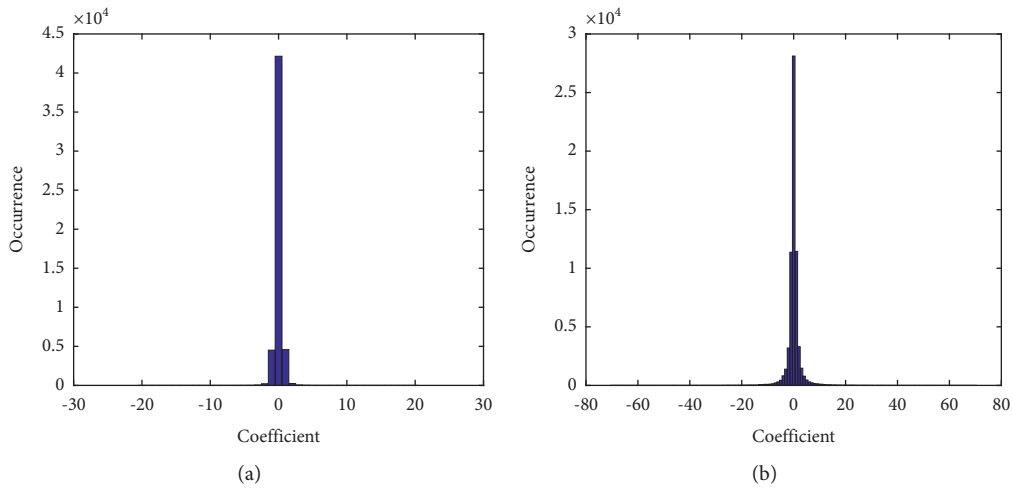

Figure 7: Coefficients histogram of region C^{re} (a) and region C^{ro} (b) for Lena.

the richer block information.The medium frequency coefficients in C^{ro} are robust against many attacks, which are selected to carry the robust watermarks. In addition, because the area with higher textural information usually has higher just noticeable difference (JND), we only use C^{ro} of the blocks from set B^c for robust embedding to ensure the high visual image quality. The reversible watermark in this scheme is embedded by the classical histogram shift (HS). For HS methods, the histogram with higher peak value and small variance produces higher performance. Thus, we use the coefficients of region C^{re} in smooth block B^s to embed reversible watermark, because the high-frequency coefficient in smooth area has a small variance. Figure 7 shows the comparison between the histogram of C^{re} and C^{ro}.

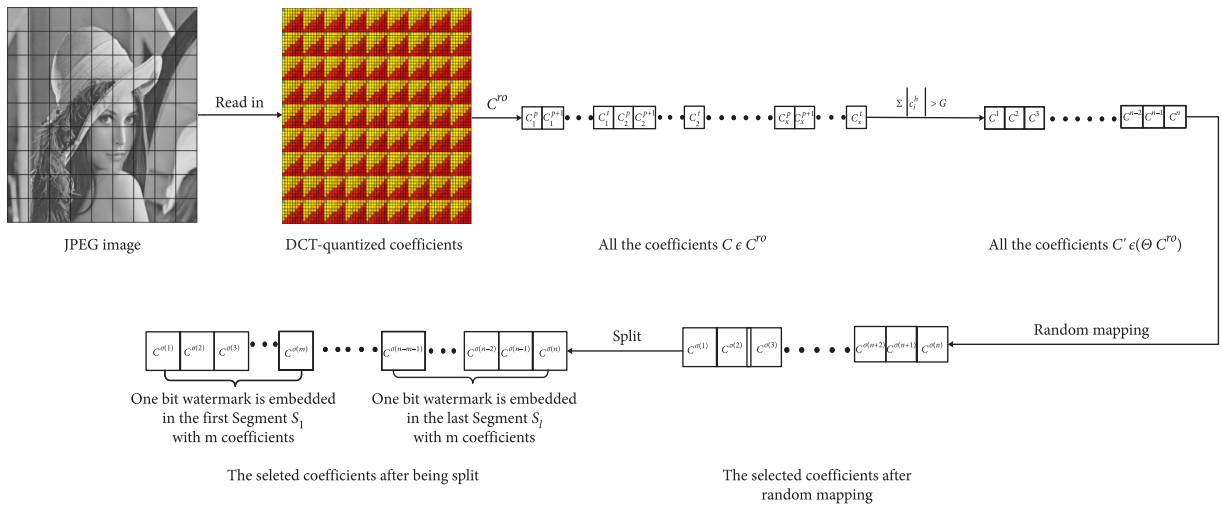

Figure 8: The diagram of robust embedding.

Therefore, the coefficients of medium frequency region C^{re} marked in red in Figure 6 from smooth block B^s are used for reversible embedding.

3.3. Stage 2 (Robust Embedding). The block B_i includes two independent embedding part: a robust embedding part $C_i^{ro} = \{c_i^p, c_i^{p+1}, \ldots, c_i^t\}$. and a reversible embedding part $C_i^{re} = \{c_i^{t+1}, c_i^{t+2}, \ldots, c_i^q\}$., where p, q, and t represent the position of the coefficient in the vector C_i. The robust watermark is embedding into C^{ro} of the complex blocks. We use the set Ω to denote these coefficients: $C_i^{ro} \in \Omega$, $if\ B_i \in B^c$. Suppose there are n coefficients in Ω. We use a random mapping as follows: $\{1, 2, \ldots, n\} \Rightarrow \{\sigma(1), \ldots, \sigma(2), \sigma(n)\}$, and the coefficients are arranged as a random long vector $\{c^{\sigma(1)}, c^{\sigma(2)}, \ldots, c^{\sigma(n)}\}$.. Then, to improve the robustness, we partition the coefficients into l segments S_k with m coefficients according to the robust watermarks length l, where $m = \lfloor n/l \rfloor$, $k \in \{0, \ldots, l-1\}$, and $\{c^{k\times m+1}, c^{k\times m+2}, \ldots, c^{(k+1)\times m}\} \in S_k$. Details are shown in Figure 8. Each coefficients c^i in the kth segment S_k is modified to embed the kth bit watermark $w_k \in [0, 1]$ as follows:

$$c_w^i = \begin{cases} c^i + \delta_k, & \text{if } w_k = 1 \text{ and } c^i \in S_k, \\ c^i - \delta_k, & \text{if } w_k = 0 \text{ and } c^i \in S_k. \end{cases} \tag{7}$$

δ_k is the modification value of each segment, which is calculated by the threshold T as follows:

$$\delta_k = \lfloor \frac{T - \left| \sum c^i \right|}{m} \rfloor, \tag{8}$$

where $\sum = \sum_{i=k\times m+1}^{(k+1)\times m} (k \in \{0, \ldots, l-1\})$. For the reversibility, δ_k in each segment must be recorded. After embedding, the sum of coefficients in each segment is greater than or equal to a threshold T.

It is worth noting that the coefficients will stay unchanged if the sum of the original coefficients of a segment exceeds or equals to the threshold requirement, i.e., the modification $\delta = 0$. After the watermark embedding is completed, the related inverse operation is performed on the coefficients in each segment to restore the position of the coefficients in each block. Afterwards, the robust embedding procedure is completed.

3.4. Stage 3 (Reversible Embedding). In reversible embedding stage, auxiliary information L, which is the information to erase the robust embedding distortion at the decoder side, is as reversible watermark. With the help of the auxiliary information L, the extraction of the watermarks w and recovery of the original image could be achieved. The auxiliary information L consists of δ_i when each bit of robust watermark is embedded:

$$L = \{\delta_1, \delta_2, \cdots, \delta_l\}. \tag{9}$$

The auxiliary information L is losslessly compressed as the reversible watermarks and transformed into a 0–1 sequence $\beta = (\beta_1, \ldots, \beta_l)$ with length l. Then, the reversible watermarks β are embedded in all the coefficients in the C^{re} of smooth block $B_i \in B^s$ by the classical HS-based reversible watermarking method. The set $\Phi = \{C_i^{re}, \text{if } B_i \text{in } B^s\}$. More specifically, the reversible embedding process is performed as follows:

$$c_w^{i'} = \begin{cases} c^i + \beta_j \times (T1 + 1), & \text{if } c^i \in \Phi \text{ and } c^i \in (0, T1], \\ c^i - \beta_j \times (T1 + 1), & \text{if } c^i \in \Phi \text{ and } c^i \in [-T1, 0), \\ c^i + (T1 + 1), & \text{if } c^i \in \Phi \text{ and } c^i > T1, \\ c^i - (T1 + 1), & \text{if } c^i \in \Phi \text{ and } c^i < T1, \end{cases} \tag{10}$$

where $\beta_j \in [0, 1]\,(j = (1, \ldots, l))$ is the reversible watermark, and $T1$ is the threshold controlling the EC (embedding capacity) of HS method.

Noticed that, in consideration of the entropy coding in JPEG compress procedure, the more zero coefficients are, the smaller size of the JPEG image is. Thus, we skip the

situation when $c^i = 0$, which will control the size of the watermarked image. As for the problem of overflow/underflow that most watermarking schemes must face, our algorithm does not need to worry about it because it works in the transform domain of the JPEG image.

After the above operations, all the watermarks are embedded in the coefficients. The last step is to reintegrate the DCT coefficients and rewrite them into the JPEG image.

3.5. Extracting Procedure. As for the decoding side, there are two kinds of cases that should be mentioned. In the case of no attacks, it contains the watermark extraction and image recovery. After preprocessing the DCT-quantized coefficients as mentioned in Section 3.1, the reversible watermarks should firstly be extracted:

$$\beta = \begin{cases} 0, & \text{if } c_w^{i'} \in [-T1, T1] \text{ and } c_w^{i'} \neq 0, \\ 1, & \text{if } c_w^{i'} \in (-2 \times T1 - 1, -T1) \text{ or } c_w^{i'} \in (T1, 2 \times T1 + 1). \end{cases} \tag{11}$$

The auxiliary information L is obtained after the decompression of the reversible watermarks. Meanwhile, the coefficient $c^i \in \Phi$ is recovered by the following equation:

$$c^i = \begin{cases} c_w^{i'} - (T1 + 1), & \text{if } c_w^{i'} \in (T1, +\infty), \\ c_w^{i'} + (T1 + 1), & \text{if } c_w^{i'} \in (-\infty, -T1). \end{cases} \tag{12}$$

According to Section 3.1, the robust watermark is extracted only by the following equation:

$$w = \begin{cases} 0, & \text{if } \sum c_w^i < 0, \\ 1, & \text{if } \sum c_w^i \geq 0. \end{cases} \tag{13}$$

Finally, with help of the auxiliary information L, the coefficient $c_{\sigma(i)}^{ro}$ is recovered by the following equation:

$$c^i = \begin{cases} c_w^i - \delta_k, & \text{if } w = 1, \\ c_w^i + \delta_k, & \text{if } w = 0, \end{cases} \tag{14}$$

where the definition of δ_k is as shown in equation (9). With the related inverse operation, the coefficients can be back to their original positions and the coefficients $c^i \in \Omega$ can be restored.

In the case of the marked image is distorted, the original image could not be recovered. However, the watermark is expected to be extracted according to equation (14) owning to the robustness of statistical property.

3.6. Procedures of Embedding and Extraction. The embedding procedure is described as follows:

Step 1. The JPEG image I is read in and each JPEG compression unit is treated as a processing block B_i.

Step 2. Zigzagly arranging each block to generate a vector C_i. Each vector is divided into 4 regions $(C^l, C^{ro}, C^{re}, C^h)$ as mentioned before. Then, using the coefficients in region C^h to predict the texture complexity of each block as equation (6).

Step 3. Picking out all the coefficients in the region C^{ro} of complex blocks, then integrating them into a long vector. After random mapping, the vector is then split into separated segments.

Step 4. As shown in equation (8), the robust watermarks w will be embedded in each separated segment in turn. As $w = 1$, the coefficients in the segment are modified so that its sum is greater than the threshold T; otherwise, it is less than the threshold $-T$.

Step 5. Noticed that the following auxiliary information should be recorded as reversible watermark: the value of the modification of each segment δ_k.

Step 6. The reversible watermark will be embedded in the region C^{re} of smooth blocks $B_i \in \Phi$ by the traditional HS method as shown in equation (11).

Step 7. After robust embedding and reversible embedding, reintegrate the DCT coefficients and rewrite them into the JPEG image. The watermarked image is obtained.

The extraction scheme is the inverse of the embedding scheme as follows:

Step 1. Referring to steps 1, 2, and 3 in embedding procedure, the regions C^{ro} and C^{re} are obtained.

Step 2. The reversible watermark should be extracted first. Using the traditional HS extraction and recovery method, auxiliary information can be obtained and coefficients in the region C^{re} of smooth blocks can be restored.

Step 3. Referring to step 4 in the embedding procedure, the robust watermark can be extracted by judging the sign of the sum of the coefficients in the segment, as shown in equation (13).

Step 4. According to auxiliary information, coefficients in the region C^{ro} of complex blocks are recovered by the related inverse operation.

Step 5. Finally, reintegrate the DCT coefficients and rewrite them into the JPEG image. The watermarked is obtained and the original image is restored.

It is worth mentioning that the original image may not be recovered if the watermarked image is attacked. However, the watermark is also supposed to be extracted.

4. Experimental Results and Discussion

This section presents experimental results for the proposed scheme. The proposed method is evaluated with embedding capacity from 200 bits to 1000 bits with an interval of 100 bits. Notice that, Coltuc et al.'s method does not perform well in JPEG images, which is explained earlier. Thus, the performance of the proposed scheme will be compared with Zeng et al.'s RHS method implemented in the compressed domain. For Zeng et al.'s method, the block size is fixed to 8×8 in consideration of JPEG procedure, and only the

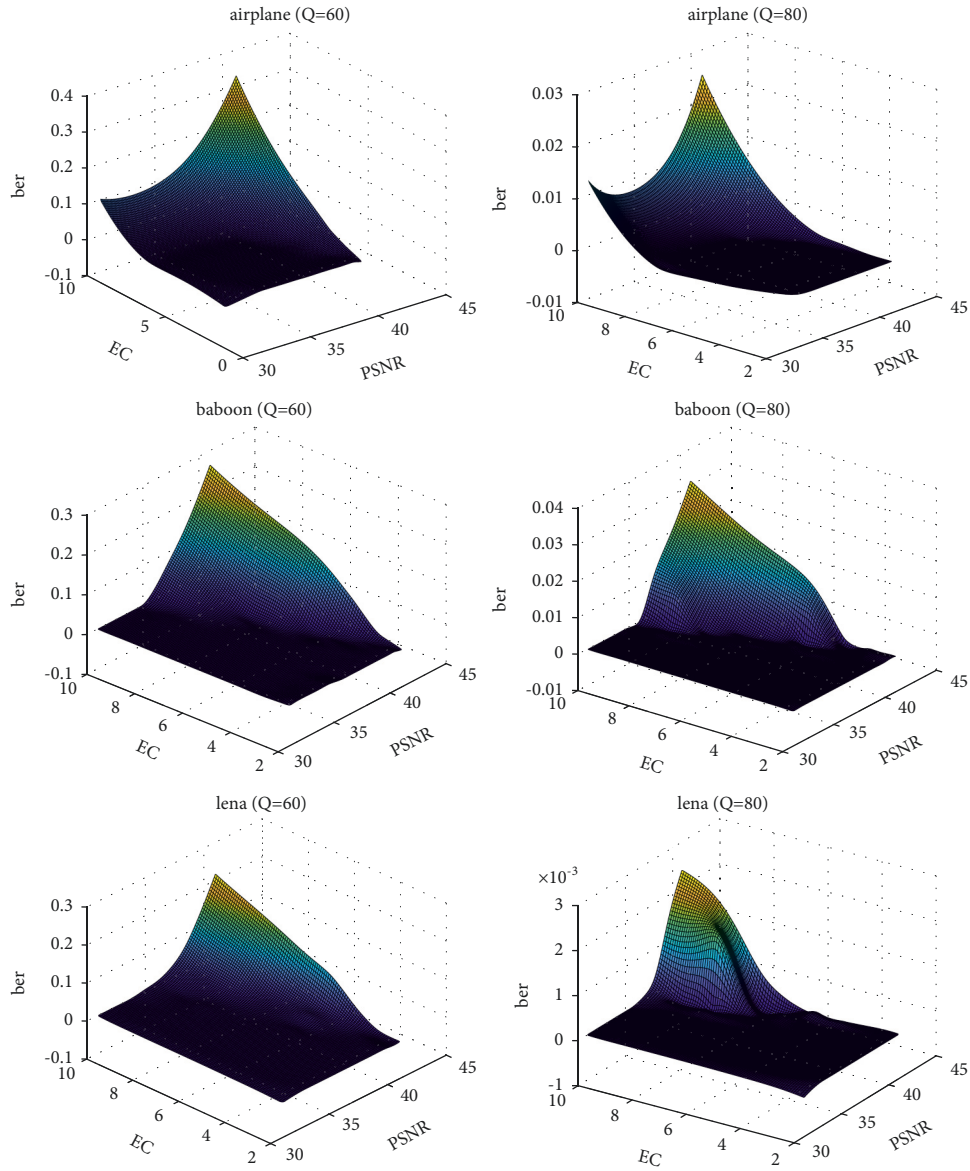

Figure 9: Trend of BER with the changes of the EC and PSNR.

coefficients in the region C^{ro} are used for the sake of fair comparison. We implement the method by Matlab and three 512×512 sized JPEG images (lena, baboon, and airplane) serve as test images, which are compressed with a quality factor of 95%. We change the embedding intensities to obtain different BERs and PSNRs both for the proposed method and Zeng et al.'s method. We change the variables G and T to control the PSNR of the proposed method. The threshold $T1$, which controls the EC in reversible embedding stage, is set as a fixed number ($T1 = 1$), because it has little effect on the image.

4.1. Robustness against JPEG Recompression. The proposed scheme is first evaluated the robustness against JPEG recompression. The bit error rate (BER) is adopted to estimate the robustness:

$$ber = \frac{l_e}{l_w}, \tag{15}$$

where l_e and l_w are the number of error bits and watermark bits, respectively. The trend of BER with the changes of the EC and PSNR for different images is shown in Figure 9. As mentioned before, to generate different PSNRs, the variables controlling the embedding intensity are varied within a certain range. Specifically, T is varying from 30 to 130 and the scope of G is based on the complexity of the image. As the axis of the EC is fixed, the BER increases with the growth of PSNR. It is because the embedding intensity decreases with the PSNR increasing, resulting in lower robustness. Furthermore, as the axis of the PSNR is fixed, the BER increases with the growth of EC. This is because the robustness of the algorithm decreases as the embedding

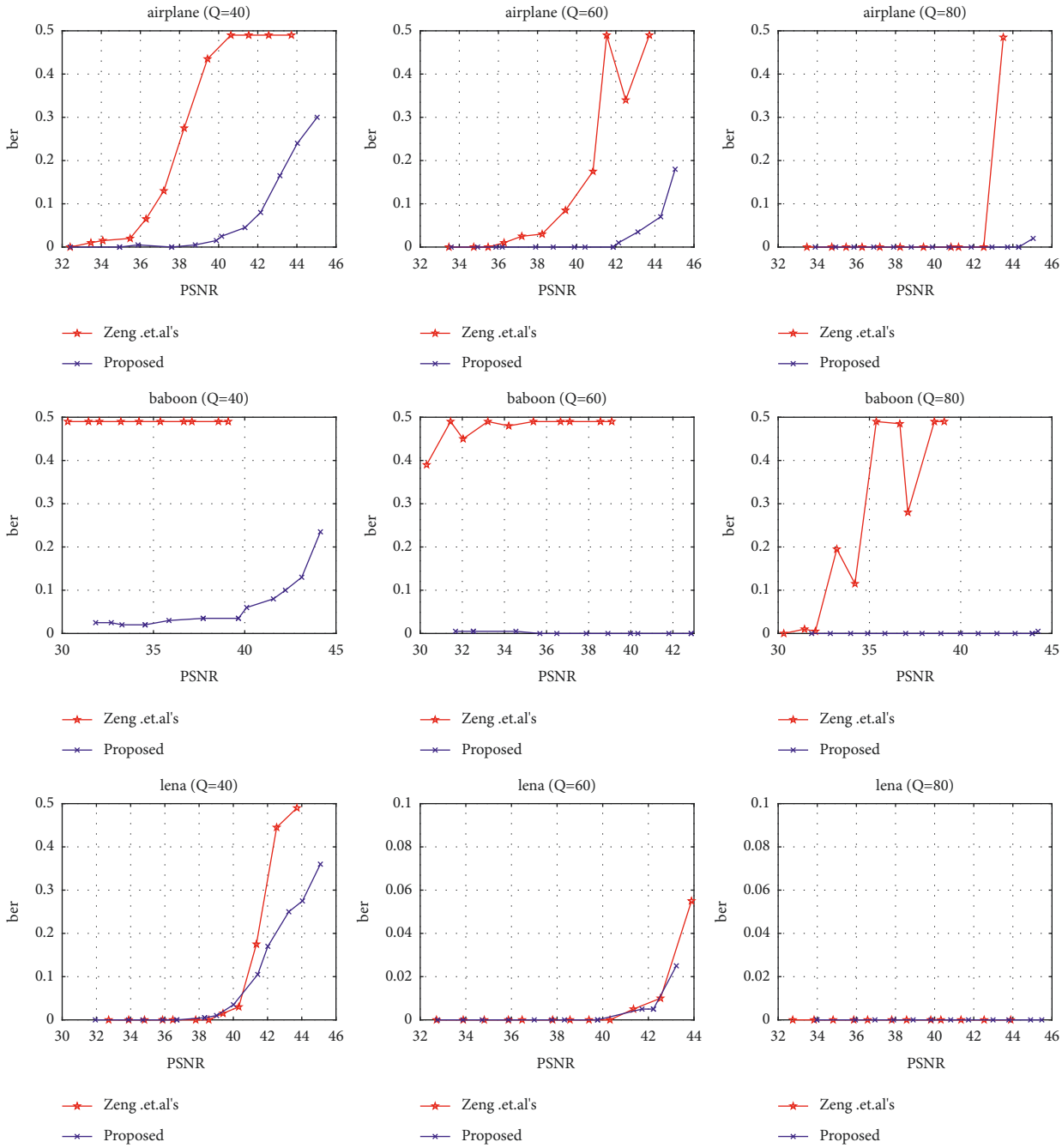

Figure 10: Comparison under different PSNRs at EC is 200 bits.

capacity EC increases when the PSNR is fixed, which leads to an increase in BER.

Figure 10 shows the comparison performance with BER vs. Figure 11PSNR at the EC of 200 bits and 500 bits. The proposed method outperforms the Zeng et al.'s method as the PSNR increases for the test images in most situations, especially for the textural image (baboon). In the case of the 80% JPEG compression, the proposed method has no error bits for 200 and 500 bits. As for the compression is to 60%, the correctness still can arrive at 100% only controlling the PSNR below 40 for the EC of 200 bits, and the corresponding PSNR might be lower when the EC is 500 bits. Even at 40% JPEG compressing, the proposed method also could keep the error rate low when the PSNR is below 36 for the EC of 200 bits. For Zeng et al.'s method, the BER quickly increases to around 50% when the PSNR arrives at a critical point, which means all the bits are wrong. The performance of Zeng et al.'s embedding scheme depends on the distribution of the histogram determined by the variance. The smaller the variance values, the better the performance of the data hiding scheme. Therefore, the performance for textural images is not satisfactory when using Zeng et al.'s method.

Referring to Figure 12, the comparisons at different JPEG Figure 13quality factors are presented. In this

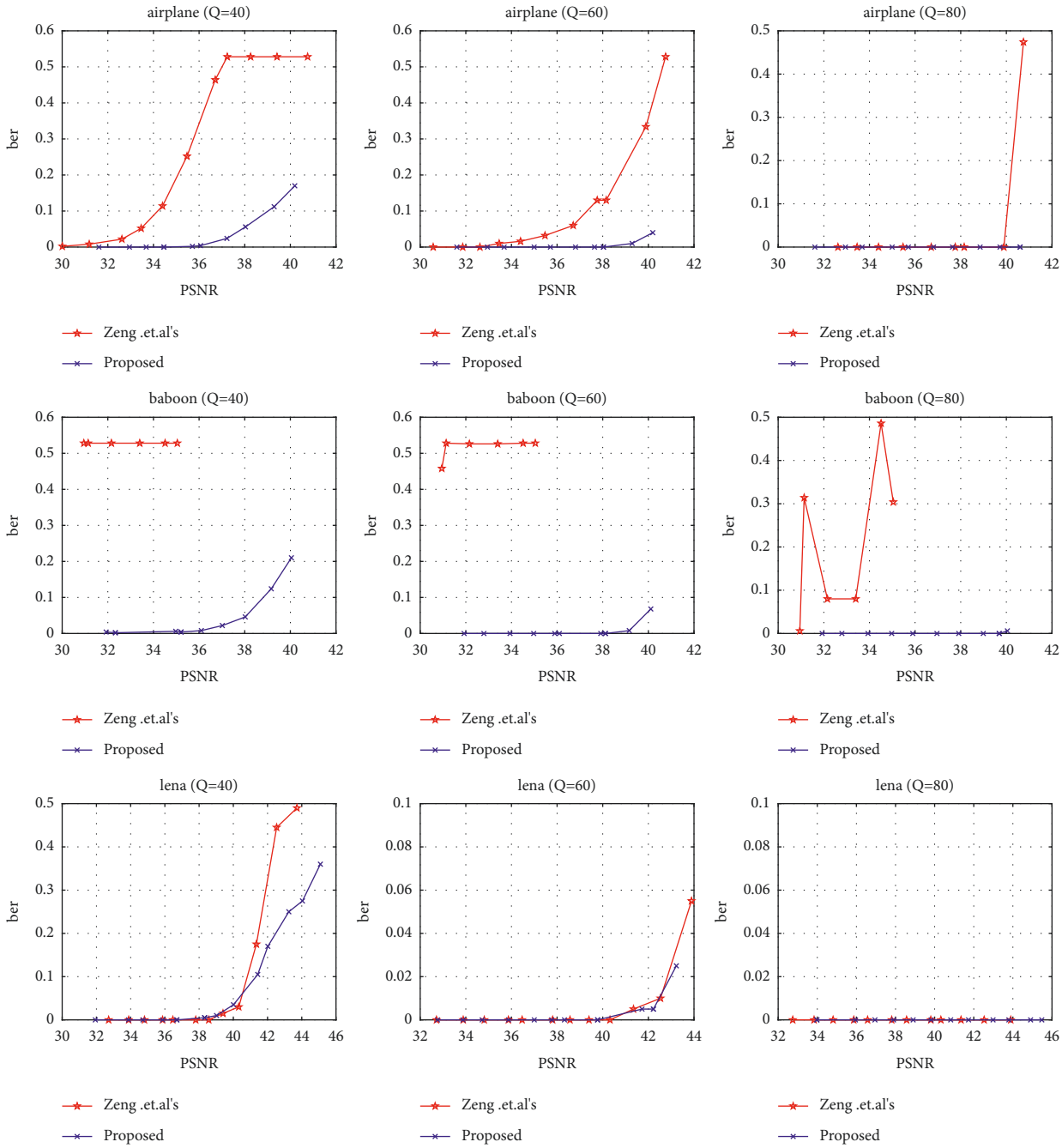

Figure 11: Comparison under different PSNRs at EC is 500 bits.

comparison, the PSNR is set to around 34 dB. It is obvious that the proposed method performs much better. For Zeng et al.'s method, the correct rate dropped quickly with the quality factor decrease, especially for the textural image baboon. This can be explained by comparing the threshold at decoding side: T and G for Zeng et al.'s method and 0 for the proposed method. Table 1 lists partial sum of the coefficients and partial arithmetic average difference of the coefficients with EC of 200 bits for image Lena when the PSNR is around 34. Here, S_o, S_w, and S_c represent the sum of the coefficients before embedding, after embedding and after JPEG compression with the 80% quality factor, and A_o, A_w, and A_c respectively represent the arithmetic average difference of the coefficients. We can see that even at 80% JPEG compression, S_c and A_c reduce quickly, which means the threshold at decoding side must be well designed for Zeng et al.'s method. Table 2 lists BER in different thresholds in the 80% quality factor compression with EC of 200 bits for image Lena, and the PSNR is also set to around 34. It could be found that for Zeng et al.'s method, the high accuracy could

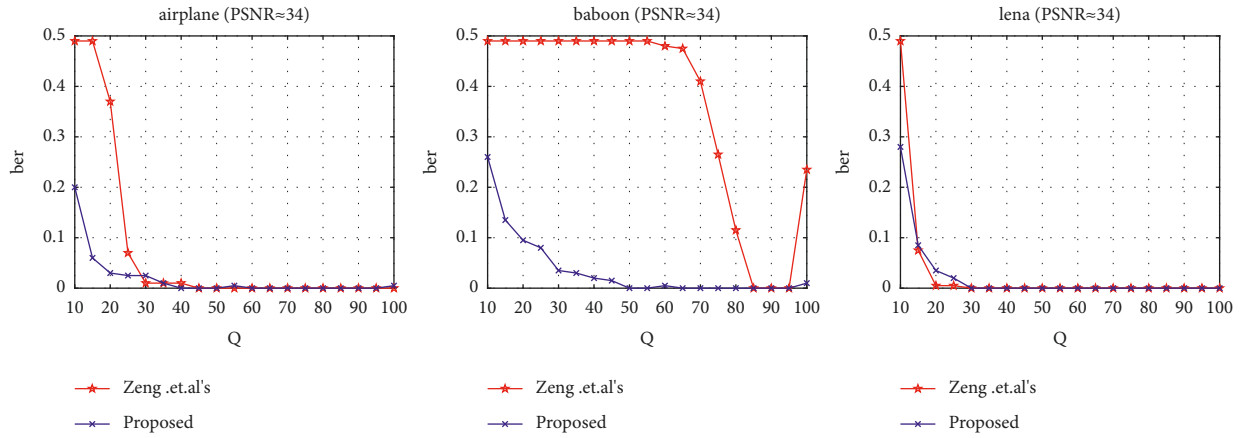

Figure 12: Comparison under different quality factors at EC is 200 bits.

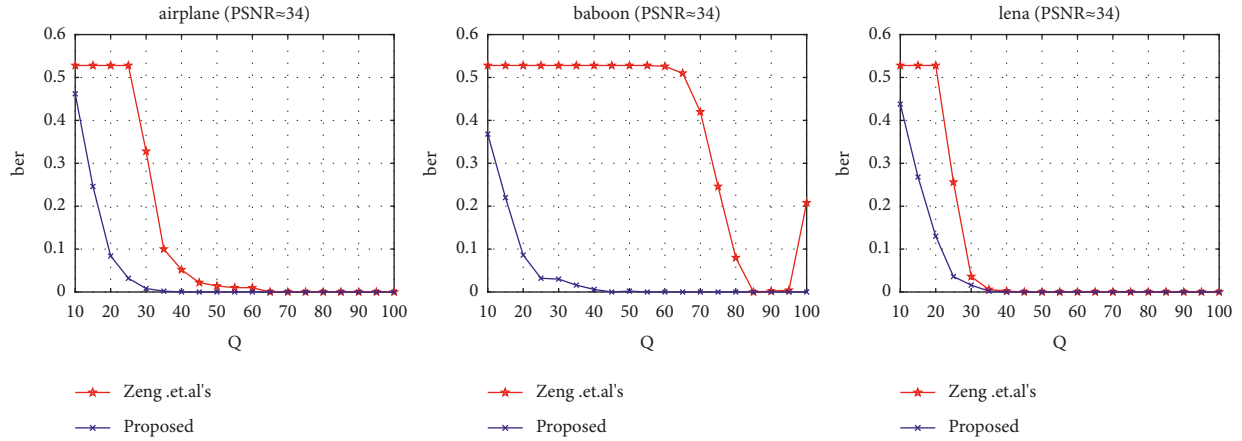

Figure 13: Comparison under different quality factors at EC is 500 bits.

Table 1: Sum of the coefficients and arithmetic average difference of the coefficients in the 80% quality factor compression.

EC = 200 bits						EC = 500 bits					
The proposed method			RHS			The proposed method			RHS		
S_o	S_w	S_c	A_o	A_w	A_c	S_o	S_w	S_c	A_o	A_w	A_c
−5	49	19	−7	111	28	−36	−92	−20	7	7	0
37	48	11	−1	−105	−24	32	−100	−23	−1	−1	0
22	−50	−11	−1	−1	0	−10	50	16	−1	−65	−16
−16	47	6	−1	−105	−23	28	76	15	−1	−65	−15
19	46	2	0	104	26	20	96	21	0	64	16
−102	−102	−29	−1	−1	0	−9	49	15	−1	−65	−16
−95	−95	−27	2	2	0	44	−84	−21	2	2	0
54	54	18	−3	−107	−25	9	67	15	−3	−67	−16

Table 2: BER in different thresholds with EC of 200 bits.

The proposed method			RHS		
G	δ	BER	G	T	BER
4	7	0.000	96	60	0.490
5	8	0.005	104	60	0.490
6	10	0.005	112	56	0.490
7	10	0.005	120	48	0.490
7	11	0.005	128	40	0.015
8	8	0.000	136	32	0.000
8	9	0.000	144	24	0.000

achieve only by setting an appropriate threshold. However, there is no such problem for the proposed method because the threshold is always 0.

4.2. Robustness against Other Attacks. This section introduces the watermarking robustness against other attacks, including AWGN with standard deviation σ_n, Gaussian filter with standard deviation σ_n equal to 1.1, median filter, and Wiener filter. Tables 3 and 4 list the BER results of image Lena for the different σ_n and different sizes of filtering mask,

Table 3: Resistance to other attacks at EC of 200 bits.

Method	AWGN (σ_n)				Gaussian filter			Median filter			Wiener filter
	10	20	30	40	$3 * 3$	$5 * 5$	$7 * 7$	$3 * 3$	$5 * 5$	$7 * 7$	$7 * 7$
Zeng et al.'s	0	0	0	0	0.490	0.490	0.490	0.490	0.490	0.490	0.150
Proposed	0	0	0	0	0.055	0.160	0.175	0.120	0.460	0.480	0.015

Table 4: Resistance to other attacks at EC of 500 bits.

Method	AWGN (σ_n)				Gaussian filter			Median filter			Wiener filter
	10	20	30	40	$3 * 3$	$5 * 5$	$7 * 7$	$3 * 3$	$5 * 5$	$7 * 7$	$7 * 7$
Zeng et al.'s	0	0	0	0	0.528	0.528	0.528	0.528	0.528	0.528	0.528
Proposed	0	0	0	0	0.132	0.270	0.270	0.214	0.492	0.482	0.026

and the PSNR is around 34 dB. Under the AWGN with different σ_n, both the two methods perform well, because it is just a simple additive noise. But as for other complicated attacks, the proposed scheme performs better than Zeng et al.'s. For the Gaussian filter and median filter, Zeng et al.'s method does not have any correct rate, our method can still maintain a certain accuracy. Even under multiplicative noise attacks, the proposed method still can yield a stable performance.

5. Conclusion

In this study, a reversible robust watermarking scheme is proposed. Different from the previous RRW methods, watermarks are embedded into the compression domain in our scheme. The reversible watermarks and robust watermarks are embedded into the coefficients of different frequency bands respectively. Meanwhile, the algorithm preferentially selects the region with complex image texture for robust watermark embedding and the smooth texture region for reversible watermark embedding. That will cause less distortion. The message can be extracted and the cover image could be recovered only if the marked image has not been attacked; otherwise, the message still could be extracted. Experiment results indicate that the proposed method can yield a quite good performance from enhancing the capacity to robustness, especially for the textural images.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Key Research and Development Program of China (2020YFB1807500) and in part by the Fundamental Research Funds for the Central Universities (XJS210107).

References

[1] D. Zhai, C. Wang, R. Zhang, H. Cao, and F. R. Yu, "Energy-saving deployment optimization and resource management for UAV-assisted wireless sensor networks with NOMA," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 6, pp. 6609–6623, 2022.

[2] D. Zhai, H. Li, X. Tang, R. Zhang, and H. Cao, "Joint position optimization, user association, and resource allocation for load balancing in UAV-assisted wireless networks," *Digital Communications and Networks*, 2022.

[3] J. Du, F. R. Yu, G. Lu, J. Wang, J. Jiang, and X. Chu, "MEC-Assisted immersive vr video streaming over terahertz wireless networks: a deep reinforcement learning approach," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 9517–9529, 2020.

[4] L. Liu, M. Zhao, M. Yu, M. A. Jan, D. Lan, and A. Taherkordi, "Mobility-aware multi-hop task offloading for autonomous driving in vehicular edge computing and networks," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–14, 2022.

[5] S. Mao, L. Liu, N. Zhang et al., "Reconfigurable intelligent surface-assisted secure mobile edge computing networks," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 6, pp. 6647–6660, 2022.

[6] H. Guo, Y. Li, and S. Jajodia, "Chaining watermarks for detecting malicious modifications to streaming data," *Information Sciences*, vol. 177, no. 1, pp. 281–298, 2007.

[7] J. Sun, W. Wang, L. Kou et al., "A data authentication scheme for UAV ad hoc network communication," *The Journal of Supercomputing*, vol. 76, no. 6, pp. 4041–4056, 2020.

[8] T. K. Al-Shayea, C. X. Mavromoustakis, J. M. Batalla et al., *Medical Image Watermarking in Four Levels Decomposition of DWT Using Multiple Wavelets in IoT Emergence[M]Convergence of Artificial Intelligence and the Internet of Things*, Springer, Cham, 2020.

[9] J. Feng, L. Liu, Q. Pei, and K. Li, "Min-max cost optimization for efficient hierarchical federated learning in wireless edge networks," *IEEE Transactions on Parallel and Distributed Systems*, p. 1, 2022.

[10] J. Feng, W. Zhang, Q. Pei, J. Wu, and X. Lin, "Heterogeneous computation and resource allocation for wireless powered federated edge learning systems," *IEEE Transactions on Communications*, vol. 70, no. 5, pp. 3220–3233, May 2022.

[11] R. Thabit and B. Ee Khoo, "A new robust reversible watermarking method in the transform domain, the 8th international conference on robotic, vision," *Signal Processing & Power Applications*, 168, 161.

[12] D. Coltuc and J.-M. Chassery, "Distortion-free robust watermarking: a case study," *Security, Steganography, and Watermarking of Multimedia Contents IX*, vol. 6505, p. 65051N, 2007.

[13] D. Coltuc, "Towards distortion-free robust image authentication," *Journal of Physics: Conference Series*, vol. 77, no. 1, Article ID 012005, 2007.

[14] Z. Ni, Y.-Q. Shi, N. Ansari, and W. Su, "Reversible data hiding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 3, pp. 354–362, 2006.

[15] M. U. Celik, G. Sharma, A. M. Tekalp, and E. Saber, "Lossless generalized-LSB data embedding," *IEEE Transactions on Image Processing*, vol. 14, no. 2, pp. 253–266, 2005.

[16] M. U. Celik, G. Sharma, and A. M. Tekalp, "Lossless watermarking for image authentication: a new framework and an implementation," *IEEE Transactions on Image Processing*, vol. 15, no. 4, pp. 1042–1049, 2006.

[17] C.-C. Chang and T. D. Kieu, "A reversible data hiding scheme using complementary embedding strategy," *Information Sciences*, vol. 180, no. 16, pp. 3045–3058, 2010.

[18] A. M. Alattar, "Reversible watermark using the difference expansion of a generalized integer transform," *IEEE Transactions on Image Processing*, vol. 13, no. 8, pp. 1147–1156, 2004.

[19] C. Wang, X. Li, and B. Yang, "High capacity reversible image watermarking based on integer transform," *Image Processing*, vol. 217–220, 2010.

[20] F. Peng, X. Li, and B. Yang, "Adaptive reversible data hiding scheme based on integer transform," *Signal Processing*, vol. 92, no. 1, pp. 54–62, 2012.

[21] S. Weng and J. S. Pan, "Integer transform based reversible watermarking incorporating block selection," *Journal of Visual Communication and Image Representation*, vol. 35, pp. 25–35, 2016.

[22] X. Li, J. Li, B. Li, and B. Yang, "High-fidelity reversible data hiding scheme based on pixel-value-ordering and prediction -error expansion," *Signal Processing*, vol. 93, no. 1, pp. 198–205, 2013.

[23] D. M. Thodi, J. J. Rodriguez, and J. J. Rodríguez, "Expansion embedding techniques for reversible watermarking," *IEEE Transactions on Image Processing*, vol. 16, no. 3, pp. 721–730, 2007.

[24] C.-C. Chang, C.-C. Lin, and Y.-H. Chen, "Reversible data-embedding scheme using differences between original and predicted pixel values," *IET Information Security*, vol. 2, no. 2, pp. 35–46, 2008.

[25] D. Coltuc, "Improved embedding for prediction-based reversible watermarking," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 3, pp. 873–882, 2011.

[26] C. De Vleeschouwer, J.-F. Delaigle, and B. Macq, "Circular interpretation of bijective transformations in lossless watermarking for media asset management," *IEEE Transactions on Multimedia*, vol. 5, no. 1, pp. 97–105, 2003.

[27] W. Bender, D. Gruhl, N. Morimoto, and A. Lu, "Techniques for data hiding," *IBM Systems Journal*, vol. 35, no. 3.4, pp. 313–336, 1996.

[28] Z. Ni, Y. Q. Shi, N. Ansari, W. Su, Q. Sun, and X. Lin, "Robust lossless image data hiding designed for semi-fragile image authentication," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 4, pp. 497–509, 2008.

[29] X. T. Zeng, L. D. Ping, and X. Z. Pan, "A lossless robust data hiding scheme," *Pattern Recognition*, vol. 43, no. 4, pp. 1656–1667, 2010.

[30] X. Gao, L. An, Y. Yuan, D. Tao, and X. Li, "Lossless data embedding using generalized statistical quantity histogram," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, no. 8, pp. 1061–1070, 2011.

[31] L. An, X. Gao, Y. Yuan, D. Tao, C. Deng, and F. Ji, "Content adaptive reliable robust lossless data embedding," *Neurocomputing*, vol. 79, pp. 1–11, 2012.