Adaptive Algorithm Based on Reversible Data Hiding Method for JPEG Images

Hao Zhang[1], Zhaoxia Yin[1,2](✉), Xinpeng Zhang[2], Jianpeng Chen[1],
Ruonan Wang[1], and Bin Luo[1]

[1] Key Laboratory of Intelligent Computing and Signal Processing,
Ministry of Education, Anhui University, Hefei 230601,
People's Republic of China
yinzhaoxia@ahu.edu.cn
[2] School of Communication and Information Engineering,
Shanghai University, Shanghai 200072, People's Republic of China
xzhang@shu.edu.cn

Abstract. This paper presents an adaptive reversible information hiding algorithm that can maintain thee JPEG file sizes by using RLC (Run Length Coding) AC coefficient coded for embedding, the key point is to choose the appropriate number of participation to hide information. By calculating the maximum storage capacity of the image at different system, select the appropriate RLC pairs to rotate and embed data. In the extraction stage, by calculating the sequence of the original RLC pairs status, then consult the mapping relationships between the current sequence and the original RLC pairs sequence, we extract the secret message and recover the original image. Test results proved that the proposed method can improve the rate-distortion performance to some extent.

Keywords: JPEG · Reversible information hiding · Run length coding RLC · Adaptive

1 Introduction

Information Hiding [1–3] is the method of using images, audio, video, text and other data carriers to embed additional information, and does not affect the original carrier. Reversible information hiding (RDH) refers to the receiving terminal can completely recover the original carrier after extracting the embedded information. For some special scenarios, such as telemedicine, military image communication, requiring that the original image must be accurately recovered after data extraction. So far, a large number of RDH method has been proposed. However, most methods are based on spatial image.

Today, JPEG is the most widely used image format in daily life, which makes the study of reversible information hiding method of JPEG images has more practical significance [4]. Redundant information in JPEG images has been compressed seriously, which made JPEG images reversible information hiding research more challenging, because not only the amount of embedded information and the quality of the

J. Wan et al. (Eds.): CloudComp 2016, SPNCE 2016, LNICST 197, pp. 196–203, 2018.
https://doi.org/10.1007/978-3-319-69605-8_18

image with secret information, as two performance indicators, but also the need of controlling the post-embedded JPEG image file size should be focused on.

The quantitative coefficient tables, Huffman tables in the header files of JPEG images can also be used to hide data. For example, in the literature [5], some quantization steps in quantization table are divided by an integer, and the corresponding DCT coefficients are multiplied by the same integer, then the additional data are added on the modified DCT coefficients. This method can obtain high embedding capacity, but with increment of the file size. In the literature [6], data are embedded according to the mapping relationship between the method used Huffman codes and the method didn't use Huffman codes. The literature [7] optimized the mapping relationship in [6]. These two methods are lossless and of course can preserve the file size, but both with relatively low embedding capacity. The embedding capacity in [5] are relatively high, but the file size is not controlled well. In worse cases, the file size increment is up to 10,000 bytes, which is collided with the original purpose of JPEG images that is to compress the image file size for economizing resources. In this paper, we proposed an adaptive algorithm based on a reversible data hiding method for JPEG. With the same length of the secret information, we get the best quality image encryption by determining the best secret information storage format and the most suitable number for the logarithm used in rotating storage secret information RLC, at the same time, we keep the file size and embed data by rotating RLC. In the data extraction stage, by determining the mapping relationships between the states with secret information and the original states, the original image is recovered completely after extracting the secret information completely. Compared with [10], this method performs better in rate-distortion to some extent.

2 Proposed Scheme

In this part, how to hide secret information in the carrier images, receive terminal extract the secret information and recover the original image will be described in detail. By choosing appropriate number of RLC pairs, we can determine the storing format of secret information and the receiver use reverse process to extract information and recover the image. The JPEG encoding [9] and RLC pairs rotation [10] will be described in Sects. 2.1 and 2.2 firstly. Then the procedure of data embedding, data extraction and image recovery are presented in Sects. 2.3 and 2.4.

2.1 The JPEG Encoding

In JPEG encoding, the key that data can be compressed is that DCT coefficients have been quantized. After quantization, many high frequency AC coefficients will be quantized to zero. For the DC coefficients, because there's a great correlation between the adjacent blocks of quantized DC coefficient, so we use the Differential Pulse Code Modulation (DPCM) to the sequence of quantized DC coefficient first. For the AC coefficients, each piece contains many zero-valued coefficients, and many of them are continuous, so we use the Run Length Coding (RLC) to the AC coefficients: first, convert the AC coefficients into one-dimension sequence in Zigzag order, then encode it into a RLC pairs sequence, as formula (1) shown. Finally, we obtain the final code stream by using Huffman encoding.

$$\left\{P_k = \left(r_{k,}\ v_k\right)\right\}_{k=1}^{L} \tag{1}$$

where L is the number of the nonzero AC coefficients in a block, which means the number of RCLP. v_k is the value of the k_{th} nonzero AC coefficient and r_k is the zero run before v_k.

The RLC pairs sequence of most image blocks will present an obvious law: with the low-frequency RLC pairs, the zero range r is small and the amplitude v is large; with the high-frequency RLC pairs, the zero range r is large and the amplitude v is small.

2.2 RLC Pairs Rotation

The RLC Pairs Rotation mainly use the two characteristics of the RLC pairs sequence mentioned in Sect. 2.1: with the low-frequency RLC pairs, the zero range r is small and the amplitude v is large; with the high-frequency RLC pairs, the zero range r is large and the amplitude v is small.

Specific operation is: suppose there is a RLC pairs sequence in certain length, donate its last four pairs of continuous sequence $p_1p_2p_3p_4$ as the original sequence state S_0, which S_0 is shown as Table 1. Then S_0 is rotated right once by a pair, generating other three sequence states $S_0 \sim S_3$. Next, S_4, reversed from S_0, is rotated in the same way to generate the RLC sequence states $S_5 \sim S_7$. There will be 8 different sequence states $S_0 \sim S_7$ including original sequence state.

Table 1. The 8 kinds of states of RLC pair

States	Sequences	Data	States	Sequences	Data
S_0	$P_1P_2P_3P_4$	000	S_4	$P_4P_3P_2P_1$	100
S_1	$P_4P_1P_2P_3$	001	S_5	$P_1P_4P_3P_2$	101
S_2	$P_3P_4P_1P_2$	010	S_6	$P_2P_1P_4P_3$	110
S_3	$P_2P_3P_4P_1$	011	S_7	$P_3P_2P_1P_4$	111

For each sequence state $s_i(0 \le i \le 7)$, the value of r, v in the first two RLC pairs $(r_1,\ v_1)$, $(r_2,\ v_2)$ and last two RLC pairs $(r_3,\ v_3)$, $(r_4,\ v_4)$ are put into Eq. (2) [8] to calculate. Calculate α_i and β_i, then subtract α_i from β_i and we have Δ_i. There will be 8 different $\Delta_i(0 \le i \le 8)$ generate by 8 different rotating states. Because of the RLC pairs features mentioned above: the value of $|v_1|$, $|v_2|, r_3$, r_4 will be relatively large, and the value of $|v_3|$, $|v_4|$, r_1, r_2 will be relatively small. In every $\alpha_i(0 \le i \le 7)$ and $\beta_i(0 \le i \le 7)$, the α in initial status will tend to the maximum, and the β in initial status will tend to the minimum, then the Δ_0 in initial status will be the largest among all $\Delta_i(0 \le i \le 7)$. So, when extracting information, confirm the status of $\Delta_{max} = max\{\Delta_i\}_{i=0}^{7}$, then we can extract the secret information and restore the original image successfully.

$$\begin{cases} \alpha_i = (|v_1| + |v_2|) \times (r_3 + r_4 + 2) \\ \beta_i = (|v_3| + |v_4|) \times (r_1 + r_2 + 2) \\ \Delta_i = \alpha_i - \beta_i \end{cases} \tag{2}$$

2.3 Data Embedding

First, we need to determine an appropriate value of *nr*, based on the length of the secret information. Parameter *nr* represents the number of RLC pairs being used to hide in each block, and parameter *na* represents the length of binary bits can be embedded in each block. This method calculates the maximum storage capacity of the image in the quaternary system, octal number system and hexadecimal system (which means rotating 3, 4 and 8 pairs of RLC pairs), and then choose the most suitable system.

When calculating the maximum capacity, the first step in this phase is to pick out the blocks with $L \geq th$. These blocks are defined as *Bc* blocks. (*th* is a parameter to decide which block can be chose to be *th* blocks, and it must be larger than or equal to $nr + 1$ because the first RLC pair of every block will be used to embed mark information and will not be rotating to store secret information. So we can't hide 7 binary number in one block which need 64 RLC pairs rotate) Then we can get the maximum capacity in different *na*.

For selected *na*, choose the minimum nr in this *na*, and $th = nr + 1$. Generally speaking, the current capacity value is the maximum value, but sometimes there will be fluctuations, so after recording the current capacity, expanding nr to get maximum capacity, and choose the maximum of *na* with principle, this time the maximum capacity will be longer than the length of the cipher text, change the value of nr again to make the capacity slightly longer than the length of the cipher text, we choose to reduce nr.

After determining the *nr*, we improve the value of *th* constantly, and finally we get the maximum capacity, which is most closed to the length of the cipher text. This is the *nr* and *th* used in the data embedding. Then we confirm the *Bc* block we need by $L \geq th$.

Second, the last *nr* RLC pairs on each *Bc* block are rotated. For instance, when $nr = 4$, whose rotation states are shown in Table 1.

There are eight sequence we got according to the method mentioned previously. For each $\Delta_i (0 \leq i \leq 2nr - 1)$ we can calculate an unique Δ_{max} and then those blocks with Δ_{max} can be picked out as *Bm* blocks. Among these *Bm* blocks, we use flag-bit 1 to mark the blocks whose Δ_{max} is in the original states $(\Delta_{max} = \Delta_0)$, and these blocks can be chose to embed secret information. Other blocks, meanwhile, are marked with 0 $(\Delta_{max} \neq \Delta_0)$. FB is the array to store these flag-bits in order, and LF is the length of FB. Obviously LF equals to the number of *Bm* blocks.

Third, a data hiding key *K* (which is the seed chosen manually to generate random number) can be used to generate LF positions randomly to decide the *Bc* blocks to hide information. The flag-bits in array FB are used to replace the LSBs of variable length integer (VLI) on the first RLC pairs in these LF blocks. Then the needless data in LF array are converted into sequence L_d. L_d is divided into many units with 4 bits in each unit to embed information.

Fourth, according to Table 1, the blocks we got in the second step are rotated according to d. For example, if $d = 010$, S_0 is rotated to S_2, and is rotated to S_5 if $d = 101$.

2.4 Data Extraction and Image Recovery

In this part, we will elaborate on the specific data extraction and image restoration steps.

First: Get the Bc blocks and Bm blocks in the same way described in Sect. 2.3.

Second: By the same data hiding key K, array FB' can be taken from the Bc blocks, then the embedding blocks containing embedded data can be picked out from the FB' array.

Third, for these embedding blocks, each $\Delta_i (0 \leq i \leq 2nr - 1)$ should be calculated and compared to find the position of Δ_{max}, represented as M. Then M is exactly the decimal form of secret information. For example, as shown in Table 1, if the S_M of a marked block is $P_1P_4P_3P_2$, and the Δ_{max} is in the state of $S_2 : P_1P_2P_3P_4$, then the embedded data is 2 in decimal, and is 010 in binary. If the S_M of a marked block is $P_1P_4P_3P_2$, then the embedded data is 5 in decimal, and is 101 in binary. All the extracted data units in binary are concatenated as L'_d.

Finally, the original LSBs of VLI on the first RLC pairs are obtained from L'_d, and put back to the original position. Then the images are recovered perfectly.

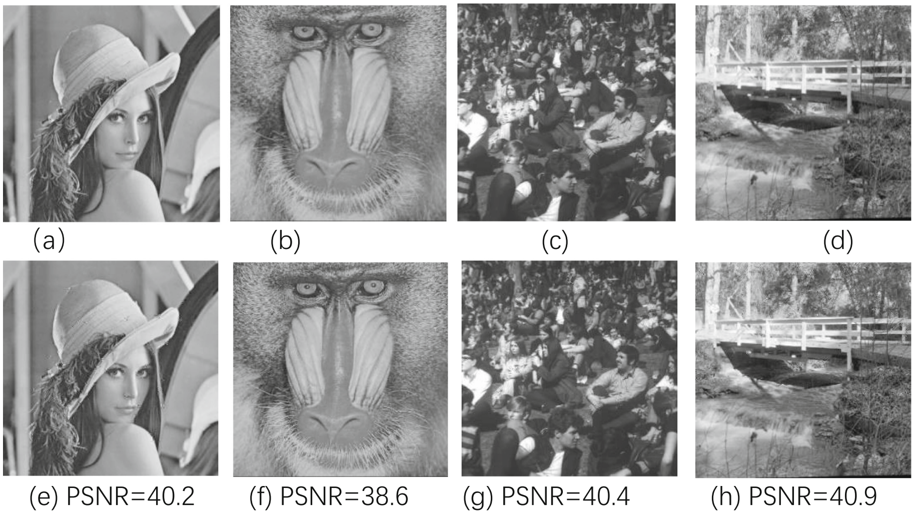
It must be said that, the position change of the RLC pairs would not generate any negative effects on encoding and decoding of JPEG image, for we only change the order of RLC sequence, this will not affect the result of Hoffman coding.

3 Experiment Result

The test JPEG pictures are compressed from the standard 512 × 512-pixel gray picture according to different QF (80). In this paper, we try to analysis the performance of this method from the quality of the picture with secret information, the embedding capacity and the change of the picture before and after the information embedding.

First is the quality of the picture with secret information. Figure 1 shows the original pictures and the pictures with secret information of Lena, Barbara, Baboon and Crowd with PSNR of each picture marked below the picture when $QF = 80$ and the length of secret information $ls = 800\,bits$. As is shown in the Fig. 1, the pictures with secret information perform well in visually, and all the picture with secret information can be recovered in this method. Figure 2 shows the comparison of the performance of rate-distortion between the experimental result in the method this paper proposed and the experimental result literature [10] obtained. Figure 3 shows change of file size in different embedding capacity of four different pictures.

Finally, comparing the method in this paper with literature [10], shown in Fig. 2, the method has obvious advantages with lower embedded capacity. Since the cipher text is converted into high system to store information, the total amount of blocks has more influence on picture. When embedded capacity is high, this method can get the



(a) (b) (c) (d)

(e) PSNR=40.2 (f) PSNR=38.6 (g) PSNR=40.4 (h) PSNR=40.9

Fig. 1. (a) Lena (original) (b) Baboon (original) (c) Bridge (original) (d) Crowd (original) (e) Lena (with secret information) (f) Baboon (with secret information) (g) Bridge (with secret information) (h) Crowd (with secret information)

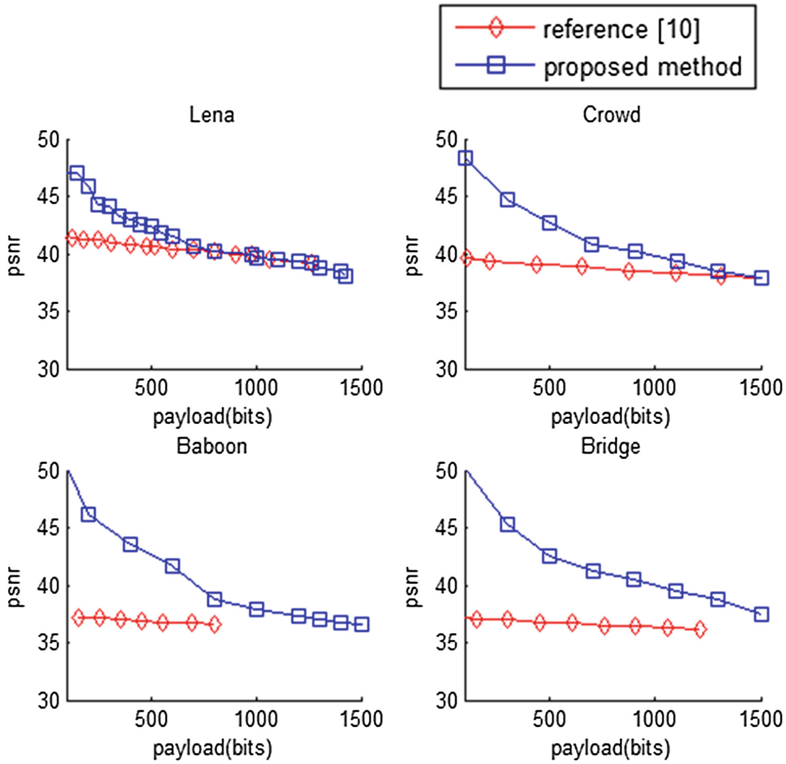


Fig. 2. The comparison of the performance of rate-distortion

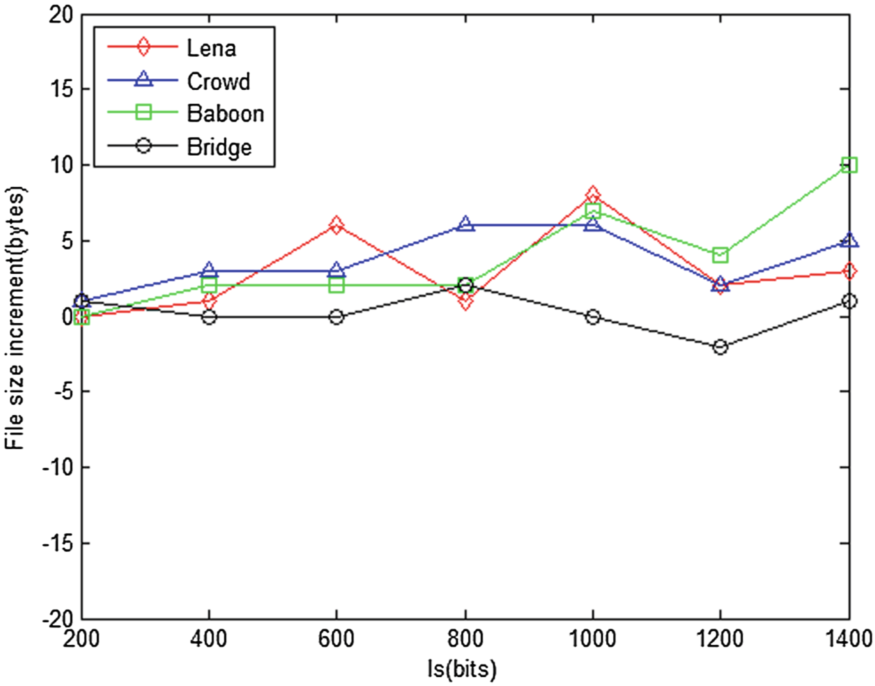


Fig. 3. Change of file size in different embedding capacity of different pictures

same result as literature [10] does, while it can get larger capacity, for the method in this paper can choose more RLC pairs to rotate.

The greatest advantage of this method is that it can keep the size gap between the original picture and the picture with secret information. As is shown in Fig. 3, with different secret information length of different picture, the size change of these pictures is close to zero. It is because in the embedding stage, only the order of the RLC pair is changed, there is no influence on the Huffman coding unit. With the change of flag-bit in the recoding stage, the size change cannot be control to zero, but the change is so small that can be ignored.

At the same time, the security of the picture can be ensured. In the recovery stage, without the hiding key, we cannot got the position of the hiding blocks, even though someone knew the way of extracting, he still cannot got the secret information in the picture.

4 Conclusion

This paper gives a reversible data hiding method based on the adaptive algorithm which can keep the size of the result file in an appropriate range. By converting the binary information into the secret message in appropriate system and embedding it into a series of blocks in the picture. The key point of this method is to find the maximum capacity of storage in different system first, then choose the best number of RLC pairs used to embed information. When the system and the number of the pairs need to be rotate is determined, raise the requirement of the RLC pairs in these blocks can used to embed the secret information gradually to decrease the number of *Bc* blocks to decrease the capacity of embedding and make it approach the final result. Finally, choose appropriate parameters and finish the secret information hiding.

Acknowledgments. This research work is supported by National Science Fund for Distinguished Young Scholars under Grant No. 61525203, National Natural Science Foundation of China (61502009, 61472235, 61572308).

References

1. Zhang, X., Wang, S.: Efficient steganographic embedding by exploiting modification direction. IEEE Commun. Lett. **10**(11), 781–783 (2006)
2. Filler, T., Judas, J., Fridrich, J.: Minimizing additive distortion in steganography using syndrome-trellis codes. IEEE Trans. Inf. Forensics Secur. **6**(3), 920–935 (2011)
3. Atawneh, S., Almomani, A., Sumari, P.: Steganography in digital images: common approaches and tools. IETE Tech. Rev. **30**(4), 344–358 (2013)
4. Wang, H., Laio, C.: JPEG images authentication with discrimination of tampers on the image content or watermark. IETE Tech. Rev. **27**(3), 244–251 (2010)
5. Wang, K., Lu, Z.-M., Hu, Y.-J.: A high capacity lossless data hiding scheme for JPEG images. J. Syst. Softw. **86**(7), 1965–1975 (2013)
6. Mobasseri, B.G., Berger, R.J., Marcinak, M.P., NaikRaikar, Y.J.: Data embedding in JPEG bitstream by code mapping. IEEE Trans. Image Process. **19**(4), 958–966 (2010)
7. Qian, Z., Zhang, X.: Lossless data hiding in JPEG bitstream. J. Syst. Softw. **85**(2), 309–313 (2012)
8. Ong, S., Wong, K.: Rotational based rewritable data hiding in JPEG. In: Visual Communications and Image Processing (VCIP), pp. 1–6. IEEE (2013)
9. Wallace, G.K.: The JPEG still picture compression standard. Commun. ACM **34**(4), 30–44 (1991)
10. Long, J., Yin, Z., Lv, J., Zhang, X.: Rotation based reversible data hiding for JPEG images. IETE Tech. Rev. **33**(6), 607–614 (2016). https://doi.org/10.1080/02564602.2015.1132014