Improved Reversible Data Hiding Scheme Based on AMBTC Compression Technique

Shan Sun, Zhaoxia Yin(✉), Jin Tang, and Bin Luo

Key Laboratory of Intelligent Computing and Signal Processing, Ministry of Education, Anhui University, Hefei 230601, People's Republic of China
yinzhaoxia@ahu.edu.cn

Abstract. As the compression technique spread on the Internet, data hiding combining with compression techniques is a hot property in recent years. In this paper, a reversible data hiding (RDH) scheme for Absolute Moment Block Truncation Coding (AMBTC) compressed image is proposed. In the proposed scheme, according to the four kinds of situations which the secret data combine with the bitmap value, we provide some rules to embed data by use the mean value and the absolute moment. Experimental results and analysis demonstrate that, the proposed method can achieve high capacity with low distortion. Besides, the proposed method is very simple and can be easily applied to real-time transmission due to its lower computational complexity.

Keywords: AMBTC · Reversible data hiding · High capacity

1 Introduction

Reversible data hiding (RDH) is a technique which embeds secret data into a cover medium and can extract the embedded data and recover the original medium with lossless. The technique is used in some special applications, which the exact recovery of the original cover medium is required, such as military images, medical images and forensics. The difference expansion (DE) scheme is one of the popular RDH schemes which first compute the error of pixel groups and expand it to adjust additional secret data [1]. The histogram shifting (HS) is another classic method which hides secret data by shifting the histogram of pixel values [2]. And other RDH schemes also have a very good ascension in embedding effect [3–5].

The compressed domain image data hiding schemes embed secret data in encrypted images. As is known to all, Joint Photographic Experts Group (JPEG) [6], Vector Quantization (VQ) [7] and Block Truncation Coding (BTC) [8–11] are the most well-known lossy compression techniques, and the relevant schemes have been proposed. In 1979, BTC was proposed by Delp and Mitchell [12]. Lema and Mitchell improve the BTC method by proposing an Absolute Moment Block Truncation Coding (AMBTC) scheme [13] which compute low mean pixel value, high mean pixel value and bitmap or absolute moment, mean pixel value and bitmap in each block. When receiver get the triple, it can easily reconstructed image block by replacing each '0' of bitmap with the low mean value and each '1' of bitmap with a high mean value. Hong et al. presented a reversible

F. Chen and Y. Luo (Eds.): Industrial IoT 2017, LNICST 202, pp. 111–118, 2017.
DOI: 10.1007/978-3-319-60753-5_12

data hiding method based on AMBTC-compressed images without extra keeping cost [14]. In 2010, Chen et al. [9] proposed a novel scheme to achieve reversible data embedding in the AMBTC compression by interchanging of the two quantization levels accompanied by the bitmap flipping. Ou et al. [15] presented a reversible secret sharing method generated AMBTC-compressed shadows in 2014. And proposed two ways to extract the secret data and achieve decoding according to whether the light-weight computational devices are obtainable. If they are obtainable, the stego image can be completely decoded and recover the original image without error, otherwise, the decoded image is very similar to primitive image. In 2015, Lin et al. [16] utilized the redundancy of the block in the AMBTC-compressed images to decide whether the block can embed the secret data. Then the scheme created four incompatible situations in the embeddable blocks to embed data. Lin et al.'s method used four hiding strategies to deal with four different cases in the embeddable cover blocks. This achieves low image distortion and high payload, but if the to-be-embedded data are continuous 0 or 1 accidentally, the rest blocks can't continue embed. So the embedding capacity is extremely instable. In order to solve this problem, we make improvement based on Lin et al.'s method.

In this paper, we proposed a reversible data hiding method for AMBTC-image base on Lin et al.'s method and have a great improvement in hiding capacity. Section 2 describes the details of the proposed scheme. Section 3 offers the experimental results. Finally, the conclusions are shown in Sect. 4.

2 Proposed Method

This section presents our new data hiding scheme, includes data embedding, data extraction and image recovery.

2.1 Data Embedding

Choose an AMBTC-compressed image as the cover, and each $m \times n$ size block including a $m \times n$ size bitmap, the mean pixel value AVG_i and the absolute moment a. There are four kinds of situations which the to-be-embedded secret data combine with the bitmap value. For example, if the secret data is '0' and the bitmap value is '1', and it is situation 01. According to the number of the case type t, we can embed the secret data by using the following strategies.

If $t = 1$, we will discard the block.

If $t = 2$, it must meet the conditions that all of the secret data are '0' or '1'. What's more, if all of the secret data are '0' and the number of '0' in the bitmap is more than or equal to 2, the number of '1' in the bitmap is more than or equal to 3, and then could embed the data. Here are the embedding rules:

Situation 00: The to-be-embedded data is '0' and the bitmap is '0'. If first time pertains to situation 00, the relevant pixel value is $AVG - a - 1$ in the cover block. If second time pertains to situation 00, the relevant pixel value is $AVG - a$ in the cover block. Else, the relevant pixel value is $AVG - a$ in the cover block.

Situation 01: The to-be-embedded data is '0' and the bitmap is '1'. If first time pertains to situation 01, the relevant pixel value is $AVG + a + 2$ in the cover block. If second time pertains to situation 01, the relevant pixel value is $AVG + a + 1$ in the cover block. If third time pertains to situation 01, the relevant pixel value is $AVG + a$ in the cover block. Else, the relevant pixel value is $AVG + a$ in the cover block.

It's important to note that $AVG - a$ is must greater than 0, and $AVG + a$ is must lesser than 254 in order to prevent overflow.

If $t = 2$ and all of the to-be-embedded data are '1' and the number of '0' in the bitmap is more than or equal to 3, the number of '1' in the bitmap is more than or equal to 3, and then could embed the data. Here are the embedding rules:

Situation 10: The to-be-embedded data is '1' and the bitmap is '0'. If first time pertains to situation 10, the relevant pixel value is $AVG - a - 2$ in the cover block. If second time pertains to situation 10, the relevant pixel value is $AVG - a - 1$ in the cover block. If third time pertains to situation 10, the relevant pixel value is $AVG - a$ in the cover block. Else, the relevant pixel value is $AVG - a$ in the cover block.

Situation 11: The to-be-embedded data is '1' and the bitmap is '1'. If first time pertains to situation 11, the relevant pixel value is $AVG + a + 2$ in the cover block. If second time pertains to situation 11, the relevant pixel value is $AVG + a + 1$ in the cover block. If third time pertains to situation 11, the relevant pixel value is $AVG + a$ in the cover block. Else, the relevant pixel value is $AVG + a$ in the cover block.

It's important to note that $AVG - a$ is must greater than 1, and $AVG + a$ is must lesser than 254 in order to prevent overflow.

If $t = 3$ or 4, it could embed secret data using the following strategies:

Situation 00: The to-be-embedded data is '0' and the bitmap is '0'. The relevant pixel value is $AVG - a$ in the cover block.

Situation 01: The to-be-embedded data is '0' and the bitmap is '1'. The relevant pixel value is $AVG + a$ in the cover block.

Situation 10: The to-be-embedded data is '1' and the bitmap is '0'. The relevant pixel value is $AVG - a - 1$ in the cover block.

Situation 11: The to-be-embedded data is '1' and the bitmap is '1'. The relevant pixel value is $AVG + a + 1$ in the cover block.

It's important to note that $AVG - a$ is must greater than 0, and $AVG + a$ is must lesser than 255 in order to prevent overflow.

2.2 Data Extraction and Image Recovery

Having received the stego image that embeds secret data, the receiver could extract the secret data and recover the original image with lossless. The specific steps are shown below.

Step 1: Scan each $m \times n$ size stego-block. Count the number of the different pixel values *num* in the current block.

Step 2: If $num = 1$ or 2, then go to Step 3. Else if $num = 3$, then go to Step 4. Else if $num = 4$, then go to Step 8. Else if $num = 5$, then go to Step 9. Else, go to Step 10.

Step 3: $num = 1$ or 2 suggested that it is a non-embeddable block, the block same as the original block. If don't scan all blocks, then go to Step 1.

Step 4: $num = 3$, the number is three. Sort the three different values as x_1, x_2 and x_3 from high to low. Calculate $x_1 - x_2$ and $x_2 - x_3$.

Step 5: Compare $(x_1 - x_2)$ with $(x_2 - x_3)$. If $(x_1 - x_2) > (x_2 - x_3)$, then go to Step 6. If $(x_1 - x_2) < (x_2 - x_3)$, then go to Step 7. $(x_1 - x_2)$ could not be equal to $(x_2 - x_3)$ according to our embedding strategies.

Step 6: Because $(x_1 - x_2) > (x_2 - x_3)$, x_2 belongs to Situation 00 and x_3 belongs to Situation 10. According to the parity of x_1 could ensure the case. If the parity of x_1 is the same as x_2, x_1 belongs to Situation 01. We could get $AVG = (x_1 + x_2)/2$, $a = AVG - x_2$. Else, x_1 belongs to Situation 11. We could get $AVG = (x_1 + x_3)/2$, $a = AVG - x_2$. If don't scan all blocks, then go to Step 1.

Step 7: Because $(x_1 - x_2) < (x_2 - x_3)$, x_1 belongs to Situation 11 and x_2 belongs to Situation 01. According to the parity of x_3 could ensure the case. If the parity of x_3 is the same as x_2, x_3 belongs to Situation 00. We could get $AVG = (x_3 + x_2)/2$, $a = AVG - x_3$. Else, x_3 belongs to Situation 10. We could get $AVG = (x_1 + x_3)/2$, $a = AVG - x_3 - 1$. If don't scan all blocks, then go to Step 1.

Step 8: $num = 4$. Sort the four different values as x_1, x_2, x_3 and x_4 from high to low. x_1 belongs to Situation 11, x_2 belongs to Situation 01, x_3 belongs to Situation 00 and x_4 belongs to Situation 10. We could get $AVG = (x_3 + x_2)/2$, $a = AVG - x_3$. If don't scan all blocks, then go to Step 1.

Step 9: $num = 5$. Sort the five different values as x_1, x_2, x_3, x_4 and x_5 from high to low. x_1, x_2, x_3 belongs to Situation 01, x_4 and x_5 belongs to Situation 00. We could get $AVG = (x_3 + x_4)/2$, $a = AVG - x_4$. And all of the data embedded is '0'. We could get the bitmap value according to the position corresponding case. If don't scan all blocks, then go to Step 1.

Step 10: $num = 6$. Sort the six different values as x_1, x_2, x_3, x_4, x_5 and x_6 from high to low. x_1, x_2, x_3 belongs to Situation 11, x_4, x_5, x_6 belongs to Situation 10. We could get $AVG = (x_3 + x_4)/2$, $a = AVG - x_4$. And all of the data embedded is '1'. We could get the bitmap value according to the position corresponding case. If don't scan all blocks, then go to Step 1.

So far, each $m \times n$ size block including a $m \times n$ size bitmap, the mean pixel value AVG_i and the absolute moment a_i. The triple is same as the block's triple of the original AMBTC-compressed image. And we completed the data extraction and image recovery.

3 Experimental Results

In order to evaluate the proposed method, we use six test grayscale images as shown in Fig. 1. There are Lena, Jet, Sailboat, Baboon, Man, Woman. Each of them has the size of 512 × 512. The secret data are generated by employing a pseudo random number generator. We perform several experiments to instruct the proposed method superiorly of hiding capacity (CAP) and stability compared with Lin et al.'s method. In all the experiments, the block of AMBTC-compression size is 4 × 4.

(1) Lena *(2) Jet* *(3) Sailboat*

(4) Baboon *(5) Man* *(6) Woman*

Fig. 1. Test images

In order to prove the proposed method superiorly of the hiding capacity (bits), we test each image by embedding the secret data in ten times using the proposed method and Lin et al.'s method [16]. Table 1 shows average of the hiding capacity (bits) and compare with Lin et al.'s method in ten times test.

Table 1. The PSNR values and CAP and compare with Lin et al.'s method

Image	Lena	Jet	Sailboat	Baboon	Man	Woman
Proposed method CAP (bits)	262101	256173	262072	262141	260045	258891
Lin et al.'s method CAP (bits)	198395	205171	176277	221394	248544	246723
Improvement (bits)	63706	51002	85795	40747	11501	12168

As shown in Table 1, the proposed method performs significantly better than Lin et al.'s method in hiding payload. In order to prove the proposed method superiorly of the embedding stability, we use each image by embedding the secret data in ten times by the proposed method and Lin et al.'s method. Figure 2 shows the hiding capacity (bits) of Lena by using proposed method and Lin et al.'s method. All of the binary data embedded in the first experiment are "0" and in the second experiment are "1". Then in the following eight experiments, the to-be-embedded data are generated by employing a pseudo random number generator which using the same seed in the same experiment ID, ensuring using the same to-be-embedded data in the same experiment ID.

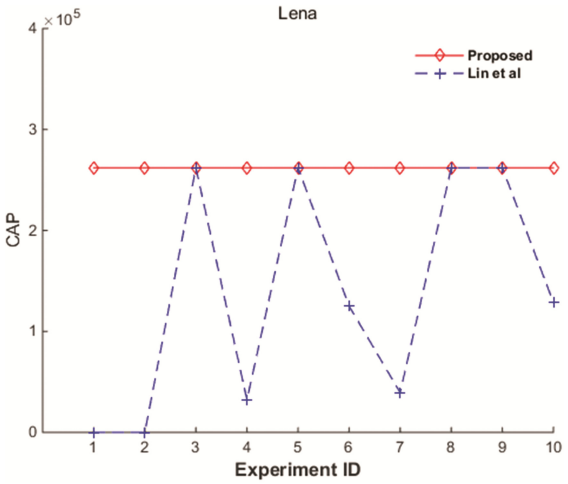

Fig. 2. Experimental results of Lena in ten times (Color figure online)

The horizontal axis represents the experiment index, and the vertical axis represents the hiding capacity in Fig. 2. Red solid curve represents the proposed method hiding capacity in different test and has changed little. Blue dotted curve represents Lin et al.'s method hiding capacity in different test and has almost changed dramatically. And it is obvious that the movements of the blue dotted curve are almost identical. This is because that the defects of this method which can't continue to embed data when the embedded data are continuous 0 or 1 accidentally. But the proposed method is almost unaffected. Compare with the proposed method, the curve with no fluctuations suggest that our method is more settled. Obviously, our proposed scheme can embed much more secret data and the embedding capacity is much more stable.

4 Conclusion

In this paper, a reversible data hiding in encrypted AMBTC-compressed image is proposed. With the combination of the bitmap and secret data, creates four situations. According to the four situations, a quite simple calculation method be provided. We make the improvement when the to-be-embedded data are continuous "0" or "1" that

Lin et al.'s method can't embed data. So the proposed method makes use of the redundant space of encrypted AMBTC-compressed image without any additional information. In addition, the stego image is not easily discovered by attacker, because it looks the same as the common image. When receiver gets the image with the secret data, he can extract the secret data and recover the original image. Experimental results and analyses demonstrate that compared with prior works, the proposed method improve the embedding capacity, and also enhance the stability of embedding.

Acknowledgments. This research work is partly supported by the National Natural Science Foundation of China (61502009, 61671018, 61472002), China Postdoctoral Science Foundation (2016M591650), Anhui Provincial Natural Science Foundation (1508085SQF216), Key Program for Excellent Young Talents in Colleges and Universities of Anhui Province (gxyqZD2016011), Quality Engineering Program for Colleges and Universities in Anhui Province (2015jyxm042) and Undergraduates Training Foundation of Anhui University (J10118515631, J18520229).

References

1. Tian, J.: Reversible data embedding using a difference expansion. IEEE Trans. Circuits Syst. Video Technol. **13**(8), 890–896 (2003)
2. Ni, Z., Shi, Y., Ansari, N., et al.: Reversible data hiding. IEEE Trans. Circuits Syst. Video Technol. **16**(3), 354–362 (2006)
3. Luo, L., et al.: Reversible image watermarking using interpolation technique. IEEE Trans. Inf. Forensics Secur. **5**(1), 187–193 (2010)
4. Li, X.L., Yang, B., Zeng, T.Y.: Efficient reversible watermarking based on adaptive prediction-error expansion and pixel selection. IEEE Trans. Image Process. **20**(12), 3524–3533 (2011)
5. Zhang, X.: Reversible data hiding with optimal value transfer. IEEE Trans. Multimed. **15**(2), 316–325 (2013)
6. Qian, Z., Zhang, X.: Improved anti-forensics of JPEG compression. J. Syst. Softw. **91**(4), 100–108 (2014)
7. Chang, C.C., Nguyena, T.Y., Lin, C.C.: A novel VQ-based reversible data hiding scheme by using hybrid encoding strategies. J. Syst. Softw. **86**(2), 389–402 (2013)
8. Chuang, J.C., Chang, C.C.: Using a simple and fast image compression algorithm to hide secret information. Int. J. Comput. Appl. **28**(4), 329–333 (2006)
9. Chen, J., Hong, W., Chen, T.S., Shiu, C.W.: Steganography for BTC compressed images using no distortion technique. Imaging Sci. J. **58**(4), 177–185 (2010)
10. Hong, W., Chen, J., Chen, T.S., Shiu, C.W.: Steganography for block truncation coding compressed images using hybrid embedding scheme. Int. J. Innov. Comput. Inf. Control **7**(2), 1–11 (2011)
11. Ou, D., Sun, W.: High payload image steganography with minimum distortion based on absolute moment block truncation coding. Multimed. Tools Appl. **74**(21), 9117–9139 (2015)
12. Bai, J., Chang, C.C.: A high payload steganographic scheme for compressed images with hamming code. Int. J. Netw. Secur. **18**(6), 1122–1129 (2016)
13. Delp, E.J., Mitchell, O.R.: Image compression using block truncation coding. IEEE Trans. Commun. **27**(9), 1335–1342 (1979)
14. Hong, W., Chen, T.S., Shiu, C.W.: Lossless steganography for AMBTC-compressed images. Int Congr Image Signal Process **2**, 13–17 (2008)

15. Ou, D., Sun, W.: Reversible AMBTC-based secret sharing scheme with abilities of two decryptions. J. Vis. Commun. Image Represent. **25**(5), 1222–1239 (2014)
16. Lin, C.C., Liu, X.L., Tai, W.L., Yuan, S.M.: A novel reversible data hiding scheme based on AMBTC compression technique. Multimed. Tools Appl. **74**, 3823–3842 (2015)