Digital Watermarking in RGB Domain on Weighted ECC

Sergey Bezzateev, Natalia Voloshina, Konstantin Zhidanov
Saint-Petersburg State University of Aerospace Instrumentation
Saint-Petersburg, Russia
bsv@aanet.ru, natali@vu.spb.ru, konstantin.zhidanov@gmail.com

Abstract

Digital watermarking is a technique for embedding imperceptible mark into digital content, with further extraction of this mark. This paper proposes a watermarking method to embed watermark in the spatial domain using error-correcting codes. Image is divided into blocks. Two perfect error-correcting codes are used to embed watermark. Codeword of corresponding perfect code is written into one block to embed one watermark bit.

Index Terms: Digital watermarking, Error-correcting codes, Weighted Hamming metric.

I. INTRODUCTION

Digital watermark is a digital sign embedded into digital image. An effective digital watermark should be imperceptible to keep source image quality. It also should be robust against image manipulations such as adding noise, filtering and wiping image bits. Watermarks could be embedded into the image in the frequency or spatial domain.

Currently, digital watermark (DWM) embedding systems use technologies of error-correction coding to protect watermark from possible errors, caused by various attacks on DWM or by image processing. In this case DWM should be encoded by specified code before embedding. Then, encoded DWM is embedded into source image. In this case the quality of extracted DWM depends on selected code properties, the type of the attack on marked image and embedding method. This paper proposes new approach of using error-correcting codes to embed digital watermark in the spatial domain.

A. Methods

This research is concentrated on DWM in time domain, for example RGB components.

The most widespread method to embed watermark in RGB domain is LSB (least significant bit) [1, 2]. This method erases data in low-order bits (LSB) of color or brightness components and writes watermark bits in cleared area. This process is shown on Fig. 1.

This method has obvious disadvantage – small robustness [6]. There are three common attacks on digital watermark in images: cleaning low-order bits, adding distortions (like blur, or applying median filter), cutting-off rows or columns of pixels. These attacks save initial image quality, but could totally corrupt watermark. First attack completely wipes watermark written to least significant bits, and there is no way to restore it. Second attack could greatly corrupt the watermark, but if the watermark is encoded with error-correction codes it gives us a chance to restore some part of corrupted watermark. Third attack erases bits of embedded data, so it can cause total corruption of watermark.

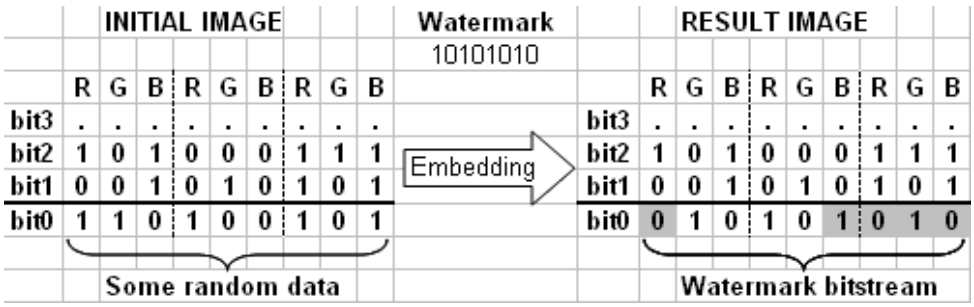


Fig. 1. Embedding watermark into LSB zone. Grey cells mean changed bits

So, the main challenge is to protect the watermark from all these attacks with saving the image quality.

The idea is to use not only least significant bits of the image, but write watermark data to more significant bits. But, it is clear that as more significant bits are modified, more distortions appear in the image. So, the goal is to find the balance between robustness and imperceptibility of watermark.

II. PROPOSED METHODS

A. Basic method of image marking

Original image bits can be looked at as a number of codewords of certain perfect code with some errors [3]. In this case embedding process could be considered as correction errors in original codewords, so that resulting image would contain codewords of certain code. Presence of these codewords will be proof of digital watermark presence. This method was proposed and examined in details in [3].

Since this method uses words of perfect error-correcting code (e.g. Hamming code), number of modified bits per codeword is limited by code distance. During embedding process initial codewords are corrected, so bits of image are changed. These changes cause distortions in image. These distortions depend on number of changed bits and their positions. It is possible to estimate average level of distortions and choose proper code to decrease it to acceptable value.

To increase robustness of watermarking method it is proposed to use not only LSB, but also more significant levels. The example of embedding principle using Hamming (7, 4) code is shown on Fig. 2. In example certain seven bits of each pixel are treated as codeword of length 7. To embed watermark it is need to change specific bits of each codeword, so resulting bits become codewords of Hamming (7, 4) code. Due to properties of Hamming code, the embedding changes no more than one bit per codeword.

This method marks image with chosen code but doesn't embed custom message.

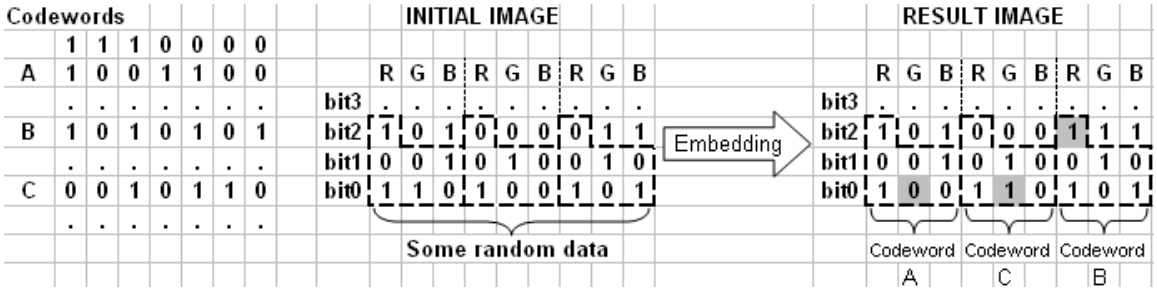


Fig. 2. Fingerprinting with Hamming (7, 4) code. Grey cells mean changed bits

B. Improved method of image marking

Hamming code is not the best solution for using in this method, because it corrects errors uniformly over all codeword length. So, distortions are added randomly in all bits of image, despite of their significance. On other hand, perfect codes in weighted Hamming metric [4], [5] correct errors non-uniformly, according to their parameters. So, choosing codes in weighted Hamming metrics allows to concentrate errors in less significant bit planes, as described in [3]. As result, using of weighted code makes less distortions in image comparing to Hamming code.

C. Method of embedding watermark message

In practice, most watermark applications require adding some information to media data (like owner name, vendor and so on), so previously described method should be improved.

Basic idea is to use two codes. One code shall stand for “1”, another one shall stand for “0”. These two codes should not have any common codewords except the zero codeword. Embedding process is the same as described in subsection B. But it uses two codes to embed watermark bits. As a result, each pixel will contain one bit of watermark message. To read watermark it is need to read codewords sequentially and refer them to one of codes (“0” or “1”).

Wiping LSB zone will erase part of codeword, but another part, which lies in more significant bits, will remain intact. And, as proposed method uses error-correcting codes, it is possible to read original data (“0” or “1”) despite errors. So, proposed method increases robustness of watermark against attacks of first and second type comparing to standard LSB method. But this method still doesn’t have resistance against third type attacks.

D. Proposed method of embedding watermark message

For further increasing of watermark robustness it is proposed to divide image into adjacent pixel blocks. One bit of watermark message will be written into all pixels of corresponding block. The size of blocks is adjustable parameter of embedding. The more size of block will be chosen, the more robust watermark will be, and the less data will be embedded into that image. Each block has size of MxN pixels. Codeword of codes “0” and “1” are written in image blocks marked as “0” and “1” respectively.

The example of dividing image into blocks is shown on Fig.3. Test image “Lena” is divided on blocks of size 20x20 pixels. Corresponding watermark bit value is written in each block.

Watermark extraction is performed in two steps. First step is the same as described in subsection C. On second step the decision on current block value (“0” or “1”) should be done. Even in case of errors in some codewords, watermark can be read from corrupted image by joint analyzing of all codewords of the same block. So, proposed method could significantly increase robustness in comparison to simple LSB method.

A. Example of proposed method

In example, standard color test image “lena.bmp” was used. Blocks of size 20x20 pixels were chosen. Two weighted codes of length 7 were used to embed watermark message, and codeword structure was the same as on Fig.2. So, three levels of bit significance were used. They were named as bit plane 0, bit plane 1 and bit plane 2

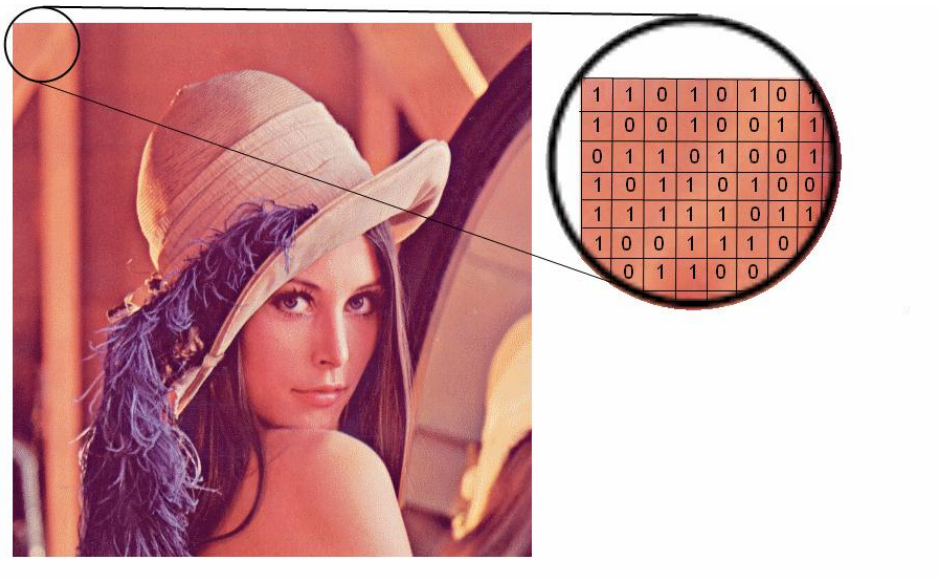


Fig. 3. Dividing image into blocks and assigning watermark bits to each block

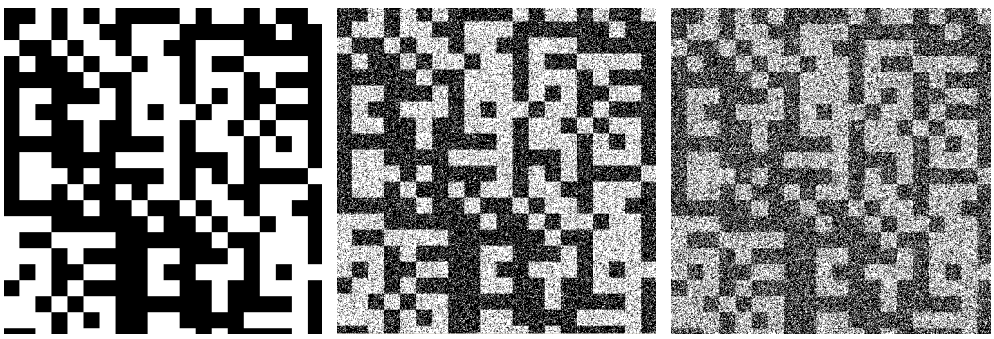
respectively (number of plain matches the order of bit in byte). Proposed method, described in subsection D was realized. Distortion of image with embedded watermark was estimated by PSNR and for number of test images was not more than -44 dB. Three types of attacks were performed. Parameters of attack were chosen so that PSNR is not more then -42 dB.

First attack was made by clearing LSB plane, and watermark message was restored precisely.

As an example of second attack such manipulations as median filtering and adding uniform noise were made.

Third attack was made by cutting off groups of pixel rows and columns of width 5

Results of watermark extraction are shown on Fig. 4. White pixels mean that corresponding codewords are treated as “0”, black pixels mean that codewords treated as “1”, and grey pixels mean that codeword cannot be recognized clearly. Fig. 4a shows watermark which extracted from uncorrupted watermarked image. Fig.4b shows watermark which extracted from image with all bits in bit plane 0 corrupted. Fig.4c shows watermark which extracted from image with all bits in bit plane 0 and half bits in bit plane 1 corrupted. It is obvious, that cutting off pixel rows and columns cannot severely damage such watermark.



a b c

Fig. 4. Extract watermarks

III. CONCLUSION

New method of digital image watermarking is proposed. Experiments, performed on standard images show that proposed method embeds robust watermark, which could be extracted after certain attacks with acceptable quality. Proposed method allows embedding watermark in digital image with imperceptible distortions and good level of robustness. Further research could be pointed on searching for parameters of codes and codeword, which will be optimal for embedding. Also optimal algorithm for reading extracted watermark bits is required.

REFERENCES

[1] Hengfu Yang, Xingming Sun and Guang Sun, "A Semi-Fragile Watermarking Algorithm using Adaptive Least Significant Bit Substitution", *Information Technology Journal*, 2010, vol.9, pp. 20-26.

[2] Todor Todorov, "Improving the Watermarking Process with Usage of Block Error-Correcting Codes Thierry Berger", *Serdica Journal of Computing*, Vol. 2, No 2, (2008), pp. 163-180.

[3] Sergey Bezzateev, Natalia Voloshina, Victor Minchenkov, "Special Class of (L,G) Codes for Watermark Protection in DRM", *Eighth International Conference on Computer Science and Information Technologies*, Yerevan, Armenia, 2011, pp. 225–228.

[4] Sergey Bezzateev, Natalia Voloshina, Konstantin Zhidanov, "Special Class of Codes for Steganographic systems", *Collection of papers Academic journal*, Novosibirsk, Russia, 2012, pp. 112–118.

[5] Sergey Bezzateev, Natalia Voloshina, Konstantin Zhidanov, "Steganographic Method on Weighted Container", *Proceedings 2012 XIII International Symposium on Problems of Redundancy in Information and Control Systems*, Saint-Petersburg, 2012, pp. 10-12.

[6] Arezoo Yadollahpour, Hossein Miar Naimi, "Attack on LSB Steganography in Color and Grayscale Images Using Autocorrelation Coefficients", *European Journal of Scientific Research* Vol.31 No.2 (2009), pp.172-183.