A Reversible Watermark Scheme Combined with Hash Function and Lossless Compression

YongJie Wang[1], Yao Zhao[1], Jeng-Shyang Pan[2], and ShaoWei Weng[1]

[1] Institute of Information Science, Beijing Jiaotong University, P.R. China
[2] National Kaohsiung University of Applied Sciences, Taiwan

Abstract. The paper presents a reversible watermark scheme based on hash function and lossless compression technology. The paper extracts the abstract information of the image as hash code and lossless compresses the LSB(Least Significant Bit) of the image to make some space where the image hash can be completely inserted. It needs to be done without introducing a large distortion to the image. The scheme can achieve the authentication purpose by hash code and restores the original image by retrieving the watermark sequence completely because of the lossless decompression of the LSB of the image.

1 Introduction

In the Internet age, audio, video media, image can be easily modified, copied and transmitted. In some application field, such as the medical image, military image and X-ray image, they are not allowed to be illegal modified and manipulated, even if several bits changed.

The reversible watermarking techniques satisfy those requirements. The reversible watermark is also named as lossless watermark, invertible watermark and erasable watermark, that is to say the reversible watermark can completely restore the original image. Owing to the reversibility, the reversible watermark is applied in the medicine fields, physical fields and so on. The retrieved watermark bits compare with the original watermark bits to detect whether the image is authentic.

In [1], Honsinger proposed a reversible watermarking as an application of image authentication in the patent. He realized the reversibility of the watermarking processing by modulo addition and a robust spatial additive watermark. It seems to be the first report on the field of reversible watermark. Another reversible watermark scheme is presented in [2] by Fridrich. The watermark is embedded in the saved space by lossless compression on the bit-plane.

This reversible watermark scheme is based on hash function and lossless compression. So we will review some relevant knowledge about the hash function and lossless in the next section, then propose our reversible image watermarking scheme and the simulation results in the Sect. 3 and Sect. 4, respectively.

R. Khosla et al. (Eds.): KES 2005, LNAI 3682, pp. 1168–1174, 2005.

2 Relevant Knowledge

This section will introduce two skills applied in the reversible watermark. One is hash function, and another is the lossless compression skill.

2.1 Hash Function

In information security fields, information authentication and digital signature technology develops rapidly. The hash function plays an important role in those applications. In the modern times, the message authentication and digital signatures technology had a great development in the information security. In [3], the hash function is introduced like that. As with the message authentication code, a hash function accepts a variable-size message M as input and produces a fixed-size hash code $H(M)$, sometimes called a message digest, as output. A hash value can be calculated by the formula as follows.

$$h = H(M) \tag{1}$$

The hash code is a function of all the bits of the message and provides an error-detection capability: A change to any bits in the message results in a change to the hash code. So the message receiver can recalculate the hash code to detect the integrality of the information. Usually, the hash function needs not the secret key, but we will use the secret key in this watermark scheme for the security of watermark.

To be useful for message authentication, a hash function H must have the following properties (adapted from a list in [NECH92]):

1) H can be applied to a block of data of any size.
2) H produces a fixed-length output.
3) $H(x)$ is relatively easy to compute for any given x, making both hardware and software implementations practical.
4) For any given code h, it is computationally infeasible to find x such that $H(x) = h$. This is sometimes referred to in the literature as the *one-way* property.
5) For any given block x, it is computationally infeasible to find $y \neq x$ with $H(y) = H(x)$. This is sometimes referred to as *weak collision resistance.*
6) It is computationally infeasible to find any pair (y, x) such that $H(y) = H(x)$. This is sometimes referred to as *strong collision resistance.*

2.2 Lossless Compression

Lossless Compression is also called as reversible compression, distortion-free coding, or noise-free coding and so on. This is a kind of developed technique. We are very familiar with Winzip software which is a good example of lossless compression. Natural images have a high degree of redundancy due to the correlation of pixels with their surrounding pixels. Information theory gives a basis for reducing redundancy, thereby achieving data compression. Accordingly, image data

compression consists of two functions: decorrelation and encoding. Sometimes we also call compression progress as removing redundancy progress. The techniques in which the original image can be completely reproduced are said to be information-preserving or reversible.

3 Proposed Reversible Watermarking Approach

3.1 Embedding and Extracting Process

First map the image to get hash code. Hash function has several selections such as SHA_1 and MD5. Hash code consists of digest information of the original image. In the following section, the paper will authenticate whether the image is modified or not. After getting the hash code, our problem is how to embed the hash code in the original image. Two problems are needed to notice:

1) The method cannot impact the visual quality of watermarked image. PSNR of watermarked image gets to a certain value.
2) The method can restore the original image from watermarked image, that is to say the method can remove the watermark bits and restore the original image.

To satisfy the first point, the paper selects the lower bit-planes to make space for hash code. It is very difficult for the LSB of image to resist some attacks. For the second point, the paper compresses the lowest bit-plane to make space for hash code. The hash functions are typically used for digital signatures to authenticate the message being sent so that the recipient can verify that the message is authentic and that it came from the right person.

We show the flow chart of the embedding process and the detecting process in Fig. 1 and Fig. 2 as follows respectively.

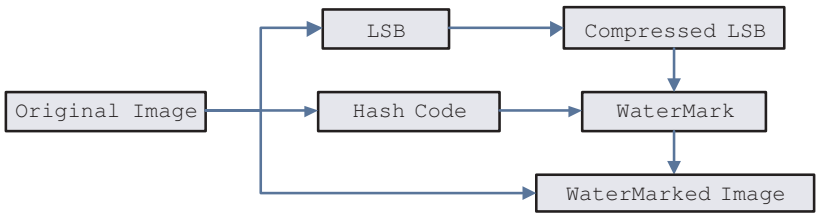

Fig. 1. Embedding Flow Chart.

There are a lot of methods for lossless compress such as arithmetic coding, Huffman coding, Run Length encoding and LZ78 compression algorithm based on the codebooks. By lossless compressing the image (or its feature), we can make some space where the image hash could be inserted. We propose to use the lowest bit-plane that, after lossless compression, provides enough space for the image hash, but bit-planes only include the symbols '0' and '1'. If applied

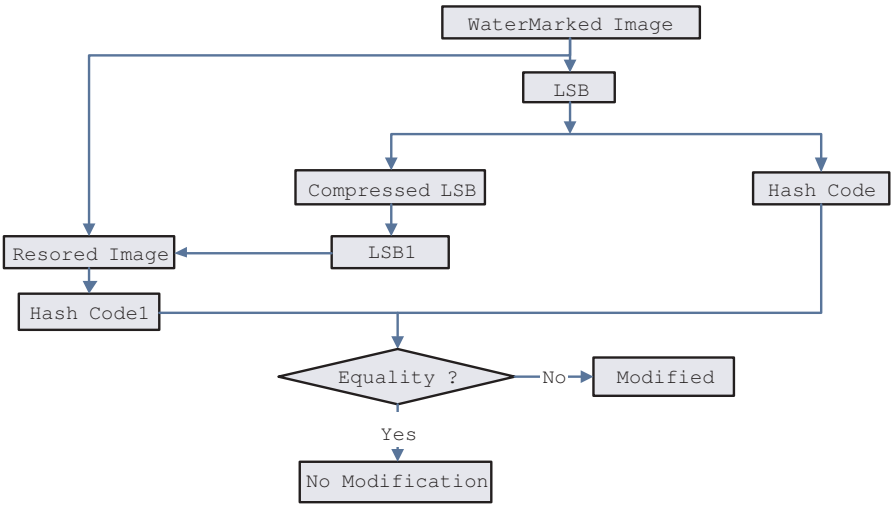

Fig. 2. Detecting Flow Chart.

to compress bit-planes, Huffman coding is of no good effect. The paper applies the Huffman coding and run-length coding to compress bit-planes. Run-length coding (which also can be found as part of the JPEG standard) statistically encodes the symbols '0' and '1' of the bit-planes to get a bit-string, and then Huffman coding encodes the bit-string to compress the bit-planes of the image. In compression process, we make use of the binary property of the bit-planes.

The process is described as follows:

The paper assumes the first symbol of the scan line is '0'. Start the coding by the run-length of the all the symbol '0'.The coder then codes the next new symbol '1' and its run-length repeatedly until the end of the scan is reached. If the first symbol of the scan line is '1', the encoder inserts a codeword for the run-length of zeros. For example, a binary bit-string "0000111000111111" coded as "5 3 3 6", "11110000011111"is coded as "4 5 5". After that we perform the Huffman algorithm on the result of the runlength processing. Then the Huffman code will be the part of the watermark. The encode and decode flow chart are proposed in Fig. 3 and Fig. 4.

4 Experimental Results

The test image is 256-grayscale image "Lena" with size 256×256 shown in Fig. 5. Hash function applies bit-by-bit exclusive-OR(XOR) the algorithm. The length of hash code is 8bit. The probability which the modified image couldn't be detected is 2^{-8}. The improved method is to select the SHA_1 or MD5 algorithm or lengthen the length of hash code. The compression algorithm is adopted as the above Huffman coding and run-length coding. Embed the hash code in the LSB of the image to get good visual quality. Fig. 6 is the watermarked figure.

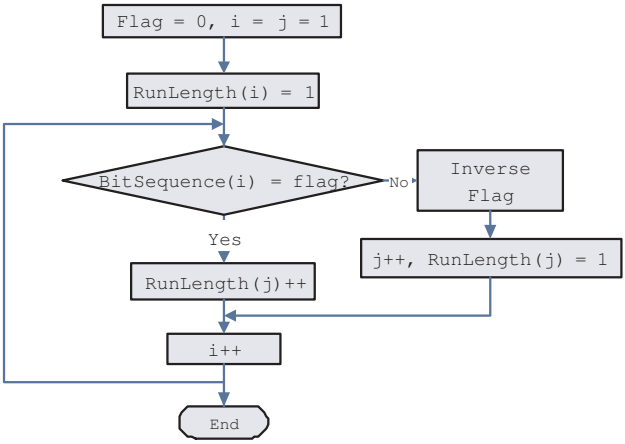

Fig. 3. Encode of Runlenth.

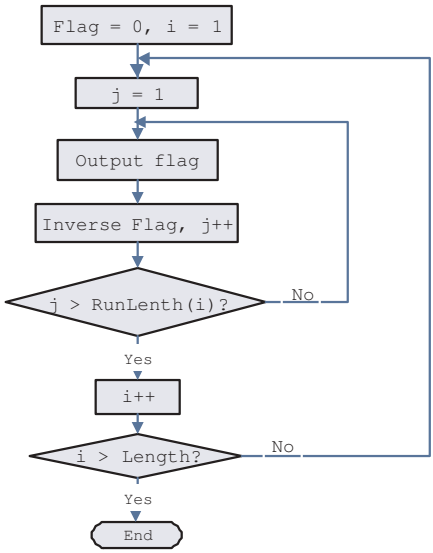

Fig. 4. Decode of Runlenth.

Fig. 5. Original.

Fig. 6. Watermarked.

Fig. 7. Restored.

First without any attacks, the paper retrieves the watermark bits and get the Hash code and compressed LSB. Decompress the LSB and get the restored image(See Fig. 7). Hash functions are typically used to authenticate whether the watermarked image is modified or not. If hash code is not changed, the paper compares the original image with the restored image bit by bit, and finds whether they are totally the same which proves the feasibility of the hash code.

Then we perform a series of attack on the embedded image, seeing the figure Fig. 8 to Fig. 12. These distortions are so trivial that we can not find the modification place by our eyes, and the watermark scheme can detect the modification very sensitivity.

Fig. 8. Adds Gaussian white noise with zero mean and 2^{-4} variance.

Fig. 9. Gaussian low-pass filter of size [3 3] with standard deviation 0.5(positive).

Fig. 10. Modified random pixels with random value.

Fig. 11. Rotation by small angel (0.1°).

Fig. 12. Scaling by 1.01 times.

5 Conclusion

In a word, this arithmetic can satisfy two characteristics of reversible watermark.

1) The original image can be recovered;
2) It is sensitive for the modification of image.

At the same time, it does not cause obvious distortions of the image after embedded the watermark. It can obtain the imperceptibility of vision just like other watermarking scheme. There are two advantages of this method:

1) Arithmetic is simple and easy to implement by software or hardware;
2) The used arithmetic is mature, security and robustness;

Thus this arithmetic has the good feasibility in practicality. For example, this reversible watermark technology can be applied to the digital camera of prison. The prison can take photo for the prisoner and keep it in the archives. It does image authentication when the prison need to use these images so that she or he can find whether these images have been modified lawlessly. It is good for the management of the prison. Another application of this reversible watermark scheme is the protection of the CT image in hospital. So the foreground of this reversible watermark theme is optimistic very much. However, this arithmetic has its own limitations. First, it is hard to confirm where the modification happens. Second, it does not refer to the correlation of the whole image, but only uses the low effective bit-plane. We need to improve this watermark theme in order to overcome these limitations. At the same time the self-recoverability of the reversible watermark will be considered. It means the watermarked imaged can recover itself against some little distortions. Then the feasibility of application will be upgraded after these improvements what will be deal with in the next research.

References

1. C. W. Honsinger, P. Jones, M. Rabbani, and J. C. Stoffel :Lossless Recovery of an Original Image Containing Embedded Data. US patent, No 77102/E-D (1999)
2. J.Fridrich,J.Goljan,and R.Du: Invertible Authentication. In Proceedings of SPIE, Security and Watermarking of multimedia Content, San Jose (2001)
3. William Stallings. (ed.): Cryptography and Network Security: Principles and Practice. Prentice-Hall (2002)