

Wavelet-based image compression and content authentication

Jun Tian

Digimarc Corporation, 19801 SW 72nd Avenue, Tualatin, OR 97062, USA

ABSTRACT

In the digital information age, digital content (audio, image, and video) can be easily copied, manipulated, and distributed. Copyright protection and content authentication of digital content has become an urgent problem to content owners and distributors. Digital watermarking has provided a valid solution to this problem. Based on its application scenario, most digital watermarking methods can be divided into two categories: robust watermarking and fragile watermarking. Here, we will concentrate on fragile watermarking of digital images, which is for image content authentication. Our fragile watermarking method is heavily based on the new image compression standard JPEG 2000. We choose a compressed bit stream from JPEG 2000 as the hash of an image, and embed the hash back to the image. The exceptional compression performance of JPEG 2000 solves the tradeoff between small hash size and high hash confidence level. In the authentication stage, the embedded compressed bit stream will be extracted. Then it will be compared with the compressed bit stream of the image to be authenticated. The authentication decision comes from the comparison result. Besides content authentication, we will also show how to employ this watermarking method for hiding one image into another.

Keywords: Image Compression, Wavelet, Content Authentication, Digital Watermarking, Fragile Watermark, Image Hiding, Digimarc

1. INTRODUCTION

Digital watermarking has grown explosively in the last few years. It embeds an invisible (in some cases, visible) mark into digital content for the purpose of copyright protection, content authentication, counterfeit deterrence, multimedia indexing, or broadcast monitoring, etc. The design of watermark message, watermark embedding method, and watermark detector are three major areas in digital watermarking research. For a detailed review of digital watermarking, we refer to¹⁻⁷ etc.

From the application point of review, most digital watermarking methods can be divided into two categories: robust watermarking and fragile watermarking. Robust watermarking is mainly aimed at copyright protection. Here “robust” means the embedded watermark should be very resistant to various signal processing operations. For robust image watermarking, these operations include image compression, scaling, rotation, translation, cropping, and filtering, etc. On the other hand, fragile watermarking is aimed at content authentication. A fragile watermark will be altered or destroyed when the digital content is modified. A fragile watermarking scheme usually creates a hash or fingerprint of a digital content and embeds the hash back into it. The requirement on the hash is that the hash size should be small, as limited by hiding capacity and content quality degradation; also the hash confidence level should be very high, that is, if two digital contents have the same hash, then these two should be the same (at least no audio and/or visual difference). For the embedding method, it should not degrade the audio, image, or video quality, and provide a reasonably large hiding capacity. More importantly, the hash of the watermarked digital content should be the same of the original, unwatermarked digital content. Otherwise at the authentication stage, the authentic digital content will be detected as unauthentic because of the hash mismatch. This will create confusion to the authentication detector. In this paper, we will present a fragile watermarking method of digital images. Our method can be applied to digital audio and video as well.

Image compression and digital image watermarking seem to be fighting against each other. Compression is trying to remove redundance in the image and throw away less important information to minimize the storage size. Watermarking is trying to add additional redundance (with a regular pattern) in the image. However digital watermarking could learn from image compression, especially for the purpose of achieving better watermark embedding and detection while maintaining the image quality. For example, lots of image modeling and human visual modeling

Further author information:

E-mail: jtian@digimarc.com, Phone: 1-503-495-4691, Fax: 1-503-495-4606, Web: <http://www.ece.rice.edu/~juntian/>



Figure 1. Original 512×512 , 8 bpp grayscale Lena.

developed in image compression have been utilized into digital watermarking. Our fragile watermarking method will be heavily based on the new image compression standard JPEG 2000.

Due to its exceptional compression performance, JPEG 2000 gives excellent image quality, both subjectively and objectively, even at a very low bit rate. We choose a compressed bit stream from JPEG 2000 as the hash of an image, and embed the hash back to the image. The compressed bit stream from JPEG 2000 solves the tradeoff between small hash size and high hash confidence level. In the authentication stage, the embedded compressed bit stream will be extracted. It will be compared with the compressed bit stream of the image to be authenticated. If these two match, the image is classified as authentic. If not, one can even decompress the embedded compressed bit stream and use the decompressed image to recover the tampered region.

The paper is organized as follows. We give a brief review of JPEG 2000 in Sect. 2. The digital image authentication method is presented in Sect. 3. Besides image authentication, we also show how to employ this watermarking method for image hiding in Sect. 4. The paper is concluded in Sect. 5.

For simplicity, we will only consider grayscale images in this paper. The generalization to color images is straightforward.

2. A BRIEF REVIEW OF JPEG 2000

The new still image compression standard JPEG 2000⁸⁻¹⁰ is a wavelet-based image compression system.¹¹⁻¹³ It can compress several different types of still images (bi-level, gray-level, color) with different characteristics (natural images, scientific, medical, military imagery, text, and rendered graphics). Its compression performance is a significant improvement over the previous discrete cosine transform (DCT) based JPEG.¹⁴ An illustration is shown in Figs. 1 and 2. Figure 1 is an 8 bit per pixel (bpp) grayscale “Lena” image with size 512×512 . The decompressed image of compression ratio 32:1 by JPEG 2000 is shown in Fig. 2. As it can be seen, even at a low bit rate $\text{bpp} = 0.25$, the visual quality is still very good.



Figure 2. JPEG 2000 compression at ratio 32:1 (bpp = 0.25).

In JPEG 2000, a discrete wavelet transform (DWT)¹⁵⁻¹⁸ first decomposes an image into spatial frequency subbands. Each subband is then partitioned into several blocks. Each block will be coded independently into embedded block bit stream by Embedded Block Coding with Optimized Truncation (EBCOT) algorithm.¹⁹ The EBCOT algorithm quantizes the wavelet coefficients, groups the quantization bits by context modeling, and uses the MQ binary arithmetic coder to further compress the bit stream. Finally the output of MQ coder from all blocks is collected into packets to form the JPEG 2000 bit stream.

In addition, JPEG 2000 provides quality scalability, resolution scalability, and spatial scalability. Especially, a lower bit rate image can be constructed by truncating the bit stream of a higher bit rate. For more details of JPEG 2000, we refer to.^{8-10,19}

3. DIGITAL IMAGE AUTHENTICATION

Image compression is to minimize the storage of digital image while maintaining the image quality. Image watermarking is to embed useful information into image and maintain the image quality as well. Image compression has been utilized into image watermarking mainly on visual modeling and the design of watermark embedding method. Here we present a method in which the watermark message is directly from image compression.

In fragile watermarking, the watermark is designed to detect possible changes in pixel values of digital images. One approach is to create a hash of the image and embed the hash back into the image. The hash size could not be too large since otherwise it will degrade the image quality. However for authenticating image with very high probabilistic confidence, one would desire the hash to be as large as possible. A good hash design for content authentication should provide

Uniqueness: Identical images produce identical hashes.

Capacity: The hash size should be small such that the image quality will not degrade.



Figure 3. Fragile watermarked image.

Invariance: The hash of an image before and after watermark embedding should be the same (or sufficiently close).

Confidence : If two images have the same hash, then these two should be the same (at least no visual difference).

These four requirements are not compatible with one another, especially the capacity and confidence requirements. To solve the tradeoff and take advantage of the compression performance of JPEG 2000, we develop a method which takes the compressed bit stream from JPEG 2000 as the hash of a digital image.

Our method begins with dividing a digital image into $M \times N$ blocks, B_1, B_2, \dots, B_t . For each block B_i ($1 \leq i \leq t$), we discard the least significant bit (LSB) (which is a shift right operation), and compress it using JPEG 2000 at a compression ratio 32:1. The compression ratio can be changed as a tradeoff among hiding capacity, image quality, and confidence level. Note that in the JPEG 2000 compressed bit stream, some header information (such as the size of the image) can be discarded, since the decoder has access to such information (which will be clear in the detection procedure). This will further reduce the length of the bit stream. For added security, the reduced bit stream of JPEG 2000 will be modulated by a pseudo random binary sequence PN . Next we create a hash, H_i , of the modulated JPEG 2000 bit stream S_i . We will embed both the modulated bit stream S_i and its hash H_i from block B_i into other blocks, as follows.

We define four permutation transformations P_1, P_2, P_3, P_4 , based on a secret key K . Each P_l ($l = 1, 2, 3, 4$) maps one block B_i to another block B_j (or equivalently maps the index i to j). We partition the LSBs of B_i into four non-overlapping regions: three regions for storing modulated JPEG 2000 bit streams and one region for storing the hash of modulated JPEG 2000 bit stream. For the modulated JPEG 2000 bit stream S_i , it will be stored three times, into the LSBs of $B_{P_1(i)}, B_{P_2(i)}, B_{P_3(i)}$, respectively. The storing of S_i can be done by a simple LSB replacement. For the hash H_i of the modulated JPEG 2000 bit stream, it will be stored once into the LSB of $B_{P_4(i)}$. This completes the fragile watermark embedding. As an example, we embed a fragile watermark into the ‘‘Lena’’ image of Fig. 1. The resulting watermarked image is shown in Fig. 3.



Figure 4. Tampered image, left eye and right eye switched place.

As it can be seen from the above, the hash of one image block B_i (excluding the LSBs) consists of two parts, the modulated JPEG 2000 bit stream S_i and the hash H_i of the modulated JPEG 2000 bit stream. Because the place to store S_i and H_i are in the LSBs of other blocks, the hash of our watermarking method satisfies both the capacity and invariance requirements. The high confidence comes from the exceptional compression performance of JPEG 2000.

The authentication process is parallel to the embedding process. We divide the image to be authenticated into $M \times N$ blocks. For each block B_i ($1 \leq i \leq t$), we partition the LSBs of B_i into four non-overlapping regions, as in the embedding process. Next we discard the LSBs of B_i , and compress B_i using JPEG 2000 at the compression ratio 32:1. Then we modulate the reduced bit stream of JPEG 2000 by the same pseudo random binary sequence PN . The modulated JPEG 2000 bit stream will be compared with three other bit streams. These three bit streams are decoded from the LSBs of blocks $B_{P_1(i)}, B_{P_2(i)}, B_{P_3(i)}$, respectively, where P_1, P_2, P_3 are the permutation transformations from the secret key K . If the bit streams match exactly for all the blocks B_i ($1 \leq i \leq t$), the image will be considered authentic. If (at least) for some block B_i , there is a mismatch between the modulated JPEG 2000 bit stream and (at least) one bit stream from the LSBs of three other blocks, the image is tampered, and a recovery procedure will follow.

When an image is found tampered, there are three possibilities:

1. Some block B_i (excluding its LSBs) has been tampered.
2. The LSBs of some block has been tampered.
3. Both 1 and 2.

The role of the hash H_i of modulated JPEG 2000 bit stream is to provide guidance for recovery. In the authentication process, when there is a mismatch between the modulated JPEG 2000 bit stream of some block B_i

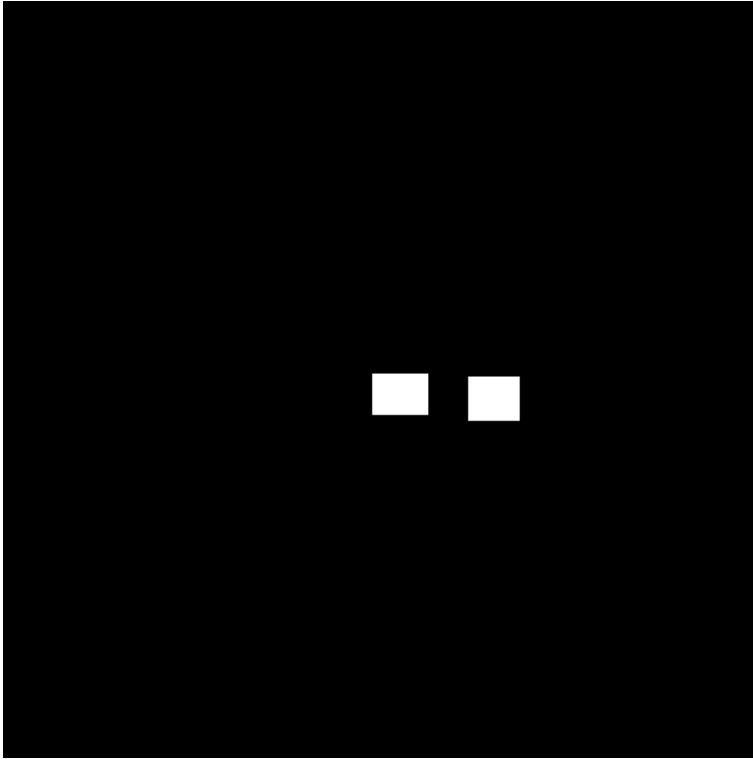


Figure 5. Authentication detection, white denotes detected tampering area.

and (at least) one bit stream from the LSBs of $B_{P_1(i)}, B_{P_2(i)}, B_{P_3(i)}$, we extract H_i from the LSBs of $B_{P_4(i)}$. For convenience, we denote the modulated JPEG 2000 bit stream of B_i as S_0 , and the three bit streams from the LSBs of $B_{P_1(i)}, B_{P_2(i)}, B_{P_3(i)}$ as S_1, S_2, S_3 , respectively. Now if the hash of S_0 matches exactly H_i , then the tampering happens at the LSBs of $B_{P_l(i)}$ ($1 \leq l \leq 3$) when the hash of S_l does not match H_i . If the hash of S_0 does not match H_i , but for some l ($1 \leq l \leq 3$), the hash of S_l matches H_i , then the tampering happens at B_i , and we decompress S_l to reconstruct an approximation at the tampered local area (where there is a bit mismatch between S_0 and S_l) in B_i . If none of the hash of S_0, S_1, S_2, S_3 matches H_i , we combine S_0, S_1, S_2, S_3 to create a new bit stream S' . If S_0 does not match S' , decompress S' to reconstruct an approximation at the tampered local area (where there is a bit mismatch between S_0 and S') in B_i .

We illustrate the authentication and recovery process by Figs. 4, 5, and 6. Figure 4 is a tampered image of Fig. 3, where we switch “Lena”’s left and right eyes. The authentication detector finds the tampering, and locates the tampering area, as shown in white area in Fig. 5. The recovered image is shown in Fig. 6.

4. IMAGE HIDING

Besides content authentication, we now show how to employ our fragile watermarking method to hide one image into another. There already exist research in the literature which utilizes compression ideas for image hiding. For example, Ding, Yan, and Qi²⁰ presented an image hiding technology based on Tangram transform (which is a generalized form of iterated function systems) and Conway’s game. As the same as for digital watermarking, image modeling and human visual modeling from compression can be used in image hiding. In addition, for image hiding, one only needs to hide a compressed image, which can fit in a low hiding capacity, and provide a better security, without sacrificing the hidden image quality.

We start with an original secret image S and an original public image P . The goal is to hide S in P , with the requirement that after the hiding, the image quality of P will not degrade, and the decoded secret image will have no visual difference from S . To employ our fragile watermarking method for image hiding, first we compress



Figure 6. Tampering recovered image.

the secret image S by JPEG 2000. The compression rate will depend on the image size ratio of S and the public image P . Next we embed the JPEG 2000 compressed bit stream of S into P . This embedding could be done by LSB replacement, (which implies an 8:1 JPEG 2000 compression, if S and P have the same size), as in Sect. 3. For security and robustness, we use a sorting order embedding instead.

The sorting order embedder modifies the pixel values in a small block to embed one message bit (“0” or “1”). First it sorts all pixel values in a block. For example, in a 2×2 block, it could be $a_{1,2} > a_{2,2} > a_{2,1} > a_{1,1}$, or $a_{1,1} = a_{2,1} > a_{2,2} > a_{1,2}$, or other sorting orders. It can be found that there are 81 possible different sorting orders in a 2×2 block. We partition all possible sorting orders into two groups G_0 and G_1 . If the message bit to be embedded is “0”, and the current sorting order is in G_0 , no action needed; if the message bit is “0”, and the sorting order is in G_1 , then we modify these four pixel values so that the new sorting order will be in G_0 . The message bit “1” will be embedded similarly. All bits in the JPEG 2000 compressed bit stream of S will be embedded into P by the sorting order embedding, which results a reconstructed public image R . The reconstructed public image R appears no visual difference from the original public image P .

To extract the secret image S from R , we again sort all pixel values in each small block. Based on whether the sorting order is in G_0 or G_1 , we decode one message bit (“0” or “1”). After going through all small blocks in R , we obtain a bit stream. Next we decode the bit stream using JPEG 2000 decoder. This will give us a decompressed image of S .

We include an example to illustration the image hiding procedure. We take the “Lena” image in Fig. 1 as the original secret image. The original public image is shown in Fig. 7, which is an 8 bpp grayscale “Barbara” image with size 512×512 . We compress the “Lena” image by JPEG 2000 with a compression ratio 32:1 and hide the compressed bit stream into the “Barbara” image by the sorting order embedding on 2×2 blocks. The resulting reconstructed public image is shown in Fig. 8. From the reconstructed public image Fig. 8, we can decode the secret “Lena” image and show it in Fig. 9.



Figure 7. Original public image: 512×512 , 8 bpp grayscale Barbara.

5. CONCLUSIONS

In this paper, we present a fragile watermarking method for image content authentication. For embedding, we choose a compressed bit stream from JPEG 2000 as the hash of an image, and embed the hash back to the image. In the authentication stage, the embedded compressed bit stream will be extracted, and compared with the compressed bit stream of the image to be authenticated. If these two match exactly, the image is classified as authentic. If not, one can even decompress the embedded compressed bit stream and use the decompressed image to recover the tampered region.

The compression idea can be used to other types of watermark messages as well. For example, if the watermark is a binary message, one could compress the binary message by entropy coding to reduce its size (thus increase the hiding capacity). If an error correction coding (ECC) is applied to the binary message before watermark embedding, then the compression will be employed before ECC.

Besides image authentication, we also show how to employ our watermarking method for image hiding. We compress a secret image S by JPEG 2000, and embed the JPEG 2000 compressed bit stream into a public image by either LSB replacement, or sorting order embedding for better security and robustness.

In general, we believe that image compression has provided lots of image modeling and human visual modeling which digital watermarking can take advantage of. Digital watermarking should incorporate image compression models to achieve both image security and image fidelity.

ACKNOWLEDGMENTS

The author would like to thank Adnan Alattar, Hugh Brunk, Steve Decker, Kevin Jones, Joel Meyer, Burt Perry, Geoff Rhoads, and Ravi Sharma of Digimarc Corporation, Mehmet U. Celik of Electrical and Computer Engineering Department, University of Rochester, and Wei Ding of Institute of Computing Technology, Chinese Academy of Science, for helpful discussion and valuable comments.



Figure 8. Reconstructed public image.

REFERENCES

1. A. M. Alattar, "Smart images using Digimarc's watermarking technology," in *Security and Watermarking of Multimedia Contents II*, P. W. Wong and E. J. Delp, eds., vol. 3791 of *Proc. SPIE*, pp. 264–273, 2000.
2. V. Cappellini, M. Barni, and F. Bartolini, eds., *Signal Processing, special issue on information theoretic aspects of digital watermarking*, vol. 81, June 2001.
3. F. Hartung and M. Kutter, "Multimedia watermarking techniques," *Proceedings of The IEEE* **87**, pp. 1079–1107, July 1999.
4. S. Katzenbeisser and F. A. P. Petitcolas, eds., *Information hiding techniques for steganography and digital watermarking*, Artech House, 2000.
5. G. C. Langelaar, I. Setyawan, and R. L. Lagendijk, "Watermarking digital image and video data," *IEEE Signal Processing Magazine* **17**, pp. 20–46, Sept. 2000.
6. F. A. P. Petitcolas, R. J. Anderson, and M. G. Kuhn, "Information hiding - a survey," *Proceedings of The IEEE* **87**, pp. 1062–1078, July 1999.
7. M. D. Swanson, M. Kobyashi, and A. H. Tewfik, "Multimedia data-embedding and watermarking technologies," *Proceedings of The IEEE* **86**, pp. 1064–1087, June 1998.
8. M. J. Gormish, D. Lee, and M. W. Marcellin, "JPEG 2000: Overview, architecture, and applications," in *Proceedings of the International Conference on Image Processing*, vol. 2, pp. 29–32, Sept. 2000.
9. D. Taubman, E. Ordentlich, M. Weinberger, G. Seroussi, I. Ueno, and F. Ono, "Embedded block coding in JPEG 2000," in *Proceedings of the International Conference on Image Processing*, vol. 2, pp. 33–36, Sept. 2000.
10. W. Zeng, S. Daly, and S. Lei, "Point-wise extended visual masking for JPEG-2000 image compression," in *Proceedings of the International Conference on Image Processing*, vol. 1, pp. 657–660, Sept. 2000.
11. A. Said and W. A. Pearlman, "A new fast and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans. Circ. Syst. Video Tech.* **6**, pp. 243–250, June 1996.



Figure 9. Decoded secret image.

12. J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Trans. Signal Processing* **41**, pp. 3445–3462, Dec. 1993.
13. P. Topiwala, ed., *Wavelet Image and Video Compression*, Klumer, 1998.
14. G. Wallace, "The JPEG still picture compression standard," *Communications of the ACM* **34**(4), pp. 30–44, 1991.
15. C. S. Burrus, R. A. Gopinath, and H. Guo, *Introduction to Wavelets and Wavelet Transforms*, Prentice Hall, Englewood Cliffs, NJ, 1997.
16. I. Daubechies, *Ten Lectures on Wavelets*, SIAM, Philadelphia, PA, 1992.
17. S. Mallat, "A theory of multiresolution signal decomposition: the wavelet representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **11**(7), pp. 675–693, 1989.
18. H. L. Resnikoff and R. O. Wells, Jr., *Wavelet Analysis and the Scalable Structure of Information*, Springer-Verlag, New York, 1998.
19. D. Taubman, "High performance scalable image compression with EBCOT," *IEEE Transactions on Image Processing* **9**, pp. 1158–1170, July 2000.
20. W. Ding, W. Yan, and D. Qi, "A novel digital image hiding technology based on tangram and conway's game," in *Proceedings of the International Conference on Image Processing*, vol. 1, pp. 601–604, Sept. 2000.