A NOVEL LOSSY-TO-LOSSLESS WATERMARKING SCHEME FOR JPEG2000 IMAGES

Zhishou Zhang, Qibin Sun, Wai-Choong Wong

Institute for Infocomm Research
21 Heng Mui Keng Terrace, Singapore 119613
{zszhang, qibin, lwong}@i2r.a-star.edu.sg

ABSTRACT

In this paper, we propose a lossy-to-lossless watermarking scheme for JPEG2000/J2K images. The watermarking is incorporated into the procedure of J2K coding in a way that the final watermark coded J2K bitstream can still maintain lossless-to-lossy scalability. To recover the original image, the J2K decoder does not need to extract the embedded watermark. This is achieved by dividing the magnitude bits of the wavelet coefficients into two portions, controlled by the watermarking survival rate (WSR). The upper portion is used to embed watermarks, while the lower portion is modified to compensate for the distortion introduced by the watermarks in the upper portion. Watermark detection can be done either from the upper portion or from the lower portion, based on a presetting threshold. Experimental results show that the J2K coded watermarked images still nearly follow the rate-distortion curve optimized for their original J2K images.

1. INTRODUCTION

Digital watermarking has been well studied in recent years for applications such as ownership rights protection, content integrity verification, etc. Basically, a watermark can be embedded into an image either in a lossy or lossless way. Lossy watermarking means that the embedded watermark impairs the image quality permanently, i.e., it is impossible to recover the original image after watermarking [1, 2, 3]. In contrast, lossless watermarking allows recovery of the original image after watermark extraction [4, 5]. Lossy watermark is usually more robust than lossless watermark because it explores embedding watermarks around edges or other perceptually important parts in the image.

Inspired by the newly issued image standard J2K [6] which integrates lossless and lossy compression into one single bitstream, our objective in this paper is to study a new watermarking scheme which is able to scale from lossy to lossless level. Thus the watermarked J2K bit-stream still owns this unique property. The whole watermarked bit-stream is coded in a lossless manner. If it is transcoded (e.g., truncated or parsed in J2K), the watermark can still be detected from this derived J2K bitstream as long as its transcoded bit-rate is larger than the WSR which is pre-defined at the watermark embedding site.

The WSR is actually used to specify the robustness of the watermark and control the visual quality of watermarked J2K images. The concept of WSR aims to make the watermarking tunable. Such a requirement is very important for security in a pervasive environment which contains many heterogeneous devices with varying computing resources and connected through different channels. Therefore, in such an environment, traditional security solutions which only provide yes/no answer cannot work well because different devices and channels may have different required security levels due to their limited computing power or their targeted applications. For example, sending a thumb-size gray-level image to a small portable device demands less security than sending a full-size color image to a desktop computer. Note that similar concepts have been proposed for other multimedia-related streaming applications [7, 8].

In this paper, we propose a lossy-to-lossless scalable watermarking scheme for J2K image, which satisfy the following requirements:

- After embedding watermark, the scheme generates a single watermarked J2K bitstream that supports lossy-to-lossless decoding.
- The watermarked J2K image is nearly rate-distortion optimized, compared with normal J2K image.
- To recover the original image, the J2K decoder does not need to extract the embedded watermark.

Our proposed scheme is motivated by the scalable lossy-to-lossless audio coder [9], where the most important information is coded in the core layer (i.e., AAC) and the residual error is compensated by the coded information in the enhancement layer. In our scheme, the magnitude bits of the wavelet coefficients are cut into two portions. The watermark is embedded into the upper portion (core), while the lower portion (residual) is modified to compensate for the distortion introduced in the upper portion. Therefore, a decoder can recover the original image without the need to extract the watermark. The cutting point of the magnitude bits is controlled by a user specified WSR. Given the WSR, the cutting points of all wavelet coefficients can be accurately allocated by running EBCOT [6].

This paper is organized as follows: Section 2 gives the system overview; Section 3 and Section 4 describe the detailed watermark embedding and extraction methods respectively. Experimental results are given in Section 5. Finally, Section 6 concludes the paper.

2. SYSTEM OVERVIEW

Fig. 1 illustrates watermark embedding in our proposed scheme. It is integrated with the J2K encoding process. Given the original image, the WSR and the watermark, we divide the magnitude bits of wavelet coefficients into two portions: core and residual. The watermark is embedded into the core portion, while the residual portion is modified to compensate for the distortion introduced by the watermark embedded in core portion. After that, the two portions are concatenated together and sent to J2K encoding, resulting in one single watermarked J2K bitstream that supports lossy-to-lossless decoding.

Watermark detection is the reverse process of watermark embedding. Similarly, the magnitude bits of wavelet coefficients are divided into two portions. We compute the correlation between the watermark and either the upper portion or the lower portion of the coefficients, which is then compared with a threshold to decide whether the watermark is present. In addition, a decoder can recover the original image without the need to extract the embedded watermark, as described in Section 4.

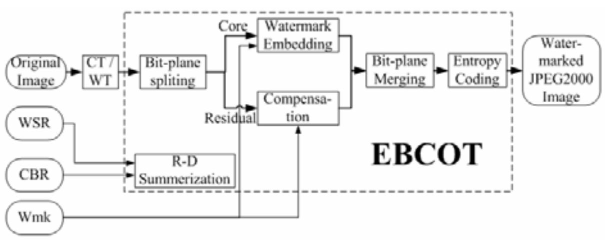

Fig. 1. Diagram of watermark embedding

3. WATERMARK EMBEDDING

We assume the watermark is a random noise sequence of real numbers $w_i \in W$, where $i = 1,2,\ldots,M$, and M is the length of the watermark sequence. It is normal distributed with zero mean and unit variance. We adopt the spread spectrum watermarking technique, though other watermarking techniques may also be applicable.

Firstly, using 5x3 reversible wavelet transform [6], we decompose the original image N times to obtain multi-resolution representations or subbands of the image: HL_n, LH_n, HH_n and LL_N ($n = 1,2,\ldots,N$). HL_n and LH_n subands are used to embed watermarks.

Secondly, WSR is used to search for the optimized truncation points by running the J2K EBCOT process [6]. For the same WSR value, the corresponding truncation point might be different for different codeblocks. If necessary, the truncation point is adjusted to stay at the bit plane boundary. After that, the truncation point is used as the dividing point to split magnitude bits of wavelet coefficients. All magnitude bits above the truncation point form the upper portion (U bits), while the rest form the lower portion (L bits). In particular, each wavelet coefficient I_i is divided into two numbers u_i and d_i such that

$$I_i = u_i \times 2^L + d_i, \quad sign(I_i) = sign(u_i) = sign(d_i)$$

Thirdly, we compute the value ∂_i that should be added to u_i in order to embed the watermark.

$$\partial_i = \begin{cases} \lfloor \alpha \times |u_i| \times w_i + 0.5 \rfloor & if \quad w_i \geq 0 \\ \lfloor \alpha \times |u_i| \times w_i - 0.5 \rfloor & if \quad w_i < 0 \end{cases}$$

where α is a fixed scaling factor between 0 and 1 (α = 0.2 in our experiment). By using $|u_i|$, a visual masking factor is implicitly utilized. For instance, for larger coefficients, the watermark will be embedded with more energy to enhance watermark robustness without degrading perceptual quality. However, adding ∂_i to u_i may cause overflow or change of sign for u_i, which may affect the J2K coding efficiency. So, we need to shape it to a new value ∂_i' that does not cause any overflow or change of sign. The ∂_i' is an integer with the largest magnitude that satisfies the following conditions:

$$|\partial_i'| \leq |\partial_i|, \quad sign(\partial_i) = sign(\partial_i'),$$

$$|u_i + \partial_i'| \leq 2^U - 1, \; sign(u_i + \partial_i') = sign(I_i),$$

$$|d_i - \partial_i' \times 2^L| \leq 2^{L+\lambda} - 1, \; sign(d_i - \partial_i' \times 2^L) = sign(I_i)$$

Once the value of ∂_i' is decided, we can embed the watermark w_i to u_i and obtain $u_i' = u_i + \partial_i'$. On the other hand, the introduced distortion is compensated by modifying d_i, such that $d_i' = d_i - \partial_i' \times 2^L$. After that, u_i' has U magnitude bits and d_i' has $L + \lambda$ magnitude bits, where λ is the number of extra bits to accommodate overflow of $d_i^{'}$. Lastly, the sign bit of I_i, U magnitude bits of $u_i^{'}$ and $L+\lambda$ magnitude bits of $d_i^{'}$ are concatenated as one coefficient $I_i^{'} = u_i^{'} \times 2^{L+\lambda} + d_i^{'}$, which is then sent to EBCOT for encoding. The final output is the J2K coded watermarked image.

4. WATERMARK DETECTION

Given a watermarked J2K image, the detector still needs two other inputs for the watermark detection: WSR and the watermark. WSR is used to determine the dividing point for each coefficient from the J2K decoding process, for recovering the original image and correctly correlating watermarks.

Assume we have obtained the sign-magnitude representation of each coefficient I_i^*, it is then divided into two numbers u_i^* and d_i^* such that $I_i^* = u_i^* \times 2^{L+\lambda} + d_i^*$ and $sign(I_i^*) = sign(u_i^*) = sign(d_i^*)$. Then we calculate the correlation z between u_i^* and w_i as follows

$$z = \frac{1}{M}\sum_{i=1}^{M} u_i^* \times w_i$$

By comparing the correlation z with a predefined threshold Thr_z [10], the detector can decide whether the watermark sequence W is present in the image.

The above discussion is for watermark detection, where only u_i^* is used (the watermark detection from d_i^* can be derived in a similar way). To recover the original image, we have to use both u_i^* and d_i^*. Suppose there is no introduced distortion, i.e., $u_i^* = u_i^{'}$ and $d_i^* = d_i^{'}$, the recovered coefficient I_i'' can be computed as follows,

$$\begin{aligned} I_i'' &= u_i^* \times 2^L + d_i^* = u_i^{'} \times 2^L + d_i^{'} \\ &= (u_i + \partial_i') \times 2^L + (d_i - \partial_i' \times 2^L) \\ &= u_i \times 2^L + d_i = I_i \end{aligned}$$

Note that WSR is used to recover the original image and correctly correlate the watermarks by accurately allocating the dividing points for each coefficient. Therefore, whether we can get the exact same dividing points as those done at the embedding site plays a key role in our scheme. We observed that actually EBCOT works very well in accurately allocating the dividing points by testing various types of images, assuming no distortion is introduced into the coded watermarked bitstream. Even if there was an exception where EBCOT failed in accurately allocating those points, we could solve this problem by inserting some special markers (>0xFF8F) into the bitstream to flag them. For the cases where some distortion exist, i.e., $u_i^* \neq u_i^{'}$ and $d_i^* \neq d_i^{'}$, though we cannot recover the original image, the watermark can still be detected. In addition, the watermark embedded in upper portion and lower portion can still cancel each other to a certain extent, thereby achieving better image quality.

5. EXPERIMENTAL RESULTS

In order to evaluate the proposed scheme, we have implemented the scheme and measured PSNR, detection response and file size of the watermarked J2K image.

Firstly, we test the compatibility of our scheme with the standard J2K decoder, i.e., using the standard J2K decoder to decode our coded watermarked J2K image. Fig. 2 shows the original "Café" image (gray level, 1024x1280 pixels, 8 bits/pixel) and that decoded by a standard J2K decoder. The image is decomposed into 4 resolution levels, and Subband HL_2 is used for watermark embedding. The measured PSNR is about 40dB.

Fig. 3 compares the PSNR of the normal J2K image and the watermarked J2K images (WSR = 1.5bpp and 3.5bpp) when they are truncated to different rates. It shows that the PSNR curve of the watermarked images closely follows that of the normal J2K images. The only exception is at the point when truncation rate approaches the WSR, where the image quality is lower than the normal J2K image. This is due to the absence of compensation from the lower portion, as all compensated bit planes have been completely truncated. Therefore, the watermarked J2K image is nearly rate-distortion optimized, as compared with the normal J2K images.

Fig. 4 compares the detection response (or correlation value) of the watermarked J2K images (WSR = 1.5bpp and 3.5bpp) when they are truncated to different rates. When the truncation bit rate is greater than 0.2, the detection response is always above the corresponding threshold, which indicates the presence of the claimed watermark. When the truncation rate is very small (<0.2bpp), the detection response drops drastically, thereby being unable to detect the watermark. Another interesting point is that, the detection response for watermark with WSR=1.5bpp drops much slower than that with WSR = 3.5bpp. That is to say, a watermark embedded with a smaller WSR is more robust.

Fig. 5 shows the detection responses when the watermarked image is detected against 200 different watermark sequences, one of which (50^{th}) was indeed embedded in the image. As seen from the graph, it can effectively detect the right watermark using the computed threshold value. The detection response when WSR= 1.5bpp is smaller than that when WSR=3.5bpp. This is because u_i is smaller when WSR is smaller, thus, the watermark is embedded with less energy, resulting in smaller detection response.

Fig. 6 compares the size of the normal J2K image and the size of the watermarked J2K images with different WSRs. Our experiment shows that the size of watermarked J2K images is about 2%~4% larger than the size of the normal J2K images.

5. CONCLUSION

In this paper, we have presented a lossy-to-lossless watermarking scheme for J2K images. Our contribution includes: 1.) Generating a single J2K coded watermarked bitstream that supports lossy-to-lossless decoding; 2.) The original image can be recovered without the need to extract the watermark. 3.) Quality of protection and watermark robustness level is quantitatively controlled by a parameter called the watermark survival rate (WSR). Our experiments confirmed that the watermarked J2K image is nearly rate-distortion optimized, the decoder can recover the original image, the watermark can be detected effectively, and it is robust against code stream truncation.

6. REFERENCES

[1] P.-C. Sun, H.-J. M. Wang and C.-C. Jay Kuo, “An Integrated Approach to Image Watermarking and JPEG-2000 Compression”, Journal of VLSI Signal Processing, vol 27, pp.35-53, 2001

[2] R. Grosbois, T. Ebrahimi, “Watermarking in the JPEG2000 domain”, MMSP2001, October 3-5, 2001

[3] W. Zeng and S. Lei, “Transform Domain Perceptual Watermarking with Scalable Visual Detection – A Proposal for JPEG2000”, ISO/IEC JTC/SC29/WG1 N759, March 1998

[4] M. Goljan, J. Fridrich and R. Du, “Distortion-Free Data Embedding”, in 4th Information Hiding Workshop, Aprial 2001

[5] Z. Ni, Y. Q. Shi, N. Ansari, W. Su, Q. Sun and X. Lin, “Robust Lossless Image Data Hiding”, submitted to ICME 2004

[6] “JPEG2000 Part I Final Draft International Standard”, ISO/IEC JTC 1/SC29/WG1 N1876, August 2000

[7] Q. Sun, S.-F. Chang, M. Kurato and M. Suto, “A Quantitative Semi-Fragile JPEG2000 Image Authentication System”, ICIP 2002

[8] C. S. Ong, K. Nahrstedt and W. Yuan, “Quality of Protection for Mobile Multimedia Applications”, ICME 2003

[9] R. Yu, X. Lin, S. Rahardja and C. C. Ko, “A Scalable Lossy To Lossless Audio Coder For MPEG-4 Lossless Audio Coding”, ICASSP 2004

[10] H. Inoue, A. Miyazaki, T. Katsura, “An Image Watermarking Method Based On The Wavelet Transform”, ICIP, 1999

(a) Original image (b) watermarked image, WSR=1.5bpp

Fig. 2 Original and Wmk image decoded by standard decoder

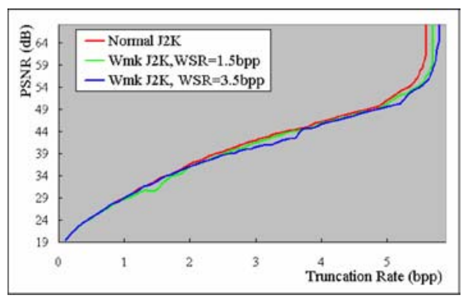

Fig. 3 Image PSNR Vs. Truncation Rate

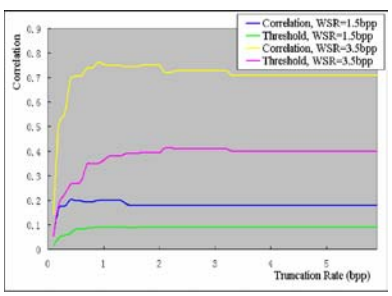

Fig. 4 Correlation Vs. Truncation Rates

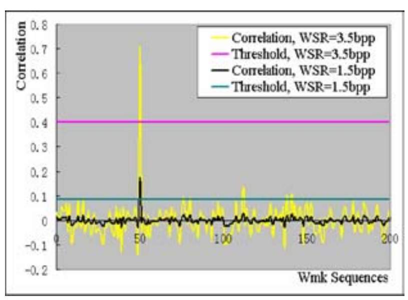

Fig. 5 Correlation Vs. different watermarked sequences

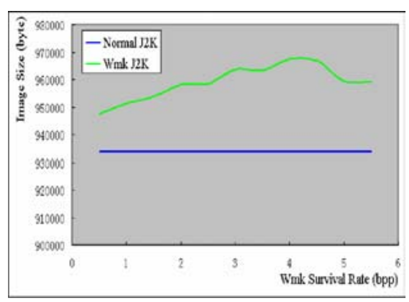

Fig. 6 Original J2K file size Vs. watermarked J2K file size for various WSRs