DATA HIDING IN JPEG 2000 CODE STREAMS

Wei Liu

Panasonic R&D Center China Co. Ltd.

ABSTRACT

In this research, the issue of integrated data hiding and JPEG 2000 [1, 2] image compression is investigated and a data-hiding scheme is proposed to embed covert messages into JPEG 2000 code streams. With the introduction of visual masking measurement and CSF weighting, visual distortion is very slight even though a large amount of data is embedded. The extraction of hidden data can be performed progressively at the decoder. Experimental results are given finally to show the performance of the proposed scheme.

1. INTRODUCTION

The terminology of data hiding or steganography usually refers to the embedding of covert messages into host media data such as audio, image, or video, which only leads to slight distortion, almost imperceptible to people. A classical example is that of a prisoner communicating with the outside world under the supervision of a prison warden. The hidden data could be of various usages like copyright information, captions, time stamps, or movie subtitles, etc. Generally, there is some difference between data hiding and watermarking in that the latter often indicates a very limited amount of embedded information, whereas the former focuses on the communication of a significant amount of hidden data without perception.

The approaches of still-image steganography are often divided into two categories. The first is to perform data hiding in the spatial or frequency domain without considering the file format, e.g. I. Cox's Spread-Spectrum watermarking [3] scheme. The other is an attempt to hide data in the compressed domain or, say, in the code streams. Compared to the first category, compressed-domain steganography provides higher data-hiding capacity, but is more vulnerable to attacks, e.g. vulnerable to a lower bit-rate recompression. But things may change with the upcoming ISO/ITU-T standard, JPEG 2000.

JPEG 2000 is believed to be a state-of-the-art image compression standard, and provides superior performance to JPEG, including higher coding efficiency, image resolution/quality scalability, better error resilience and spatial random access, and is expected to take the place of JPEG in the near future. Consequently, it is worthwhile to investigate steganography algorithms of hiding secret information inside JPEG 2000 code streams effectively.

This paper is organized as follows. In section 2, we briefly review the Part I of JPEG 2000 standard. Then in section 3, we introduce and evaluate a previous JPEG 2000 steganography algorithm [4] by Po-Chyi Su and C.-C. Jay Kuo. In section 4, we present a new steganography scheme which is supposed to be more robust, progressively extractable, and of high visual fidelity. Simulation results are given in section 5 to show the performance of the new scheme. And, finally, conclusions are drawn in section 6.

2. REVIEW OF JPEG 2000 STANDARD

The JPEG 2000 compression engine is illustrated in block diagram form in Fig. 1. At the encoder, the discrete transform is first applied on the source image data. The transform coefficients are then quantized and entropy coded before being packaged to form the output code stream.

Unlike JPEG baseline, which scales the compression ratio by using a particular Q-table, JPEG 2000 also applies rate control during the process of entropy coding, which is called the Embedded Block Coding with Optimal Truncation (EBCOT) scheme [5]. EBCOT is basically a bit-plane coder with a block-based structure. After DWT and quantization, the quantizing indices in each subband are partitioned into relatively small blocks; that is, the code blocks. Each code block is encoded independently, producing an elementary JPEG 2000 bit-stream. The above process is called the tier-1 coding.

More specifically, in tier-1 coding, an individual code block is encoded one bit-plane at a time from the most significant bit-plane to the least. Each coefficient bit in the bit-plane is coded in only one of the three coding passes, namely the significance propagation (SP), the magnitude refinement (MR), and the cleanup pass. During the SP pass, a bit is coded if its location is not significant (the significance bit has yet to be encountered), but at least one of its eight-connected neighbors is significant. The second pass is the MR pass, which conveys all bits that became significant in a previous bit-plane. And the final pass is the cleanup pass, in which all bits not encoded during the

0-7803-8554-3/04/$20.00 ©2004 IEEE.

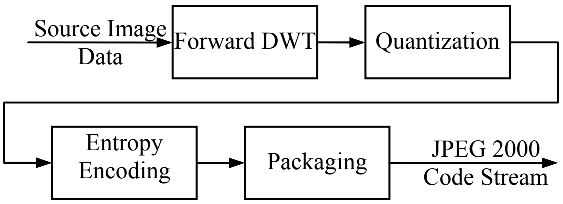

Fig. 1. General block diagram of JPEG 2000 (encoder only).

previous two passes are encoded.

In addition to the above, the binary symbols from the three passes are usually arithmetically coded, but a lazy coding mode is also used to reduce computation cost. According to this mode, after the fourth bit-plane is coded, the SP pass and MR pass in each bit-plane are included as raw (uncompressed) data; that is, the arithmetic coder is bypassed except for the cleanup pass.

The JPEG 2000 elementary bit-stream has the property that it can be truncated at the end-points of each of its coding passes, and the distortion, incurred when reconstructing from each of these truncated subsets, is estimated and denoted by MSE. The main goal of the EBCOT tier-2 coding is to allocate the truncation points of each code block in an optimal way so as to minimize the total distortion with the targeted bit-rate constraint. After that, the coding passes are packaged into packets and output to form the final code stream.

Once the JPEG 2000 encoder has decided the maximum image quality and resolution, any image quality or resolution can be decompressed by extracting the corresponding bits from the resulting code stream. Additionally, JPEG 2000 code streams support spatial random access. One can retrieve and decompress data from the code stream mapping to the selected spatial regions of an image. In each case discussed above, it is possible to locate, extract, and decode the bits required for the desired image product without having to decode the entire code stream of the image like JPEG baseline. This feature is beneficial to steganography because the process of recompression may fatally ruin the hidden data in the compressed domain. But this feature also brings the challenge that any parts of the code stream may be dropped at the decoder end so that any part of the hidden data may be lost in a particular decompression.

3. PREVIOUS RELATED WORK

Po-Chyi Su and C.-C. Jay Kuo proposed a smart embedding scheme in [4]. They carefully analyzed the challenges of information embedding in JPEG 2000 and suggested that the information should be embedded in the output of the EBCOT tier-2 coding, JPEG 2000 packets, to avoid the two major sources of information loss: quantization and bit-stream truncation. They also suggested that the integrated data hiding and JPEG 2000 encoding scheme should use a lazy coding mode, and the bits of the secret message should be embedded into the raw-coded, MR passes to avoid both any difficulties in modifying the compactly compressed packets and the risk of impacting the significance context of the lower bit-planes of the code block.

The suggested implementation performs lazy-mode JPEG 2000 compression and embeds data *backwardly* from the lastly included, raw coded, MR pass (which has the smallest rate-distortion slope). The embedding procedure is to replace the original pass by a pre-produced bit pattern.

Su's is a powerful scheme in that it achieves a very high embedding capacity for a relatively low cost of computation. However, the backward embedding method has an innate drawback. As we have mentioned, the encoder only decides the maximum resolution and quality of an image. The decoder may extract the required bits for its desired reconstruction. Once it needs an image product of lower pixel-fidelity, the whole code stream must be scaled and some of the least important parts must be dropped. The MR passes selected for data embedding usually have small rate-distortion slopes and are prone to be dropped first. That is, the backward embedding scheme is quite vulnerable to quality scaling. In addition, the backwardly embedded hidden data cannot be extracted progressively by the decoder, which is essential for low bit-rate transmission applications, in which extracting the hidden data progressively without waiting the whole image being downloaded or received is desirable.

4. THE PROPOSED SCHEME

However, Su's scheme is thought-provoking for our further study in designing an alternative. Many of his suggestions are reasonable, like performing lazy-mode JPEG 2000 compression, embedding data in the raw coded MR passes after EBCOT tier-2 coding. But ours is a *forward* embedding method. In our point of view, a secret bit should be embedded as deep into the bit-planes as possible in the consideration of robustness and progressive extraction.

As the embedding goes deeper, the distortion also becomes more severe. Accordingly, the embedding position should be carefully selected to avoid perceptible distortion to human eyes. Thus the pass-wise embedding should be replaced by a point-wise one and the human visual system (HVS) must be considered. Moreover, the embedding in each code block should be *independent* to resist the attack of random spatial access in the decoder.

4.1 Visual Distortion Measurement

Statistically, embedding a bit anywhere into a pass incurs the same distortion in the MSE sense, but study of the HVS indicates that some distortion can be well masked by the

Level	Y(LH, HL, HH)	Cb(LH, HL, HH)	Cr(LH, HL, HH)
1	0.307191, 0.307191, 0.108920	0.097816, 0.097816, 0.031179	0.177435, 0.177435, 0.077130
2	0.861593, 0.861593, 0.742342	0.280068, 0.280068, 0.152290	0.388492, 0.388492, 0.248566
3	1.000000, 1.000000, 1.000000	0.501652, 0.501652, 0.362279	0.598537, 0.598537, 0.470893
4	1.000000, 1.000000, 1.000000	0.689404, 0.689404, 0.579220	0.757626, 0.757626, 0.666951
5	1.000000, 1.000000, 1.000000	0.818766, 0.818766, 0.745875	0.860885, 0.860885, 0.803172

Tab. 1. Sample weighting table for a viewing distance of about 1700 pixels for a five-level decomposition. The LLn subbands (not shown) should be assigned a weight of 1.

original image acting as background [6], which is called the masking effect.

W. Zeng's point-wise extended masking [7] provides a facility for a spatially varying nonlinearity to be applied to wavelet samples prior to quantization. This nonlinearity preserves the sign of each coefficient, but modifies its magnitude according to:

$$z_i = \frac{sign(x_i)\left|x_i\right|^{\alpha}}{1+\left(a\sum_{k\in neighborhood}\left|\hat{x}_k\right|^{\beta}\right)\Big/\left|\Phi_i\right|}\,. \tag{1}$$

In equation (1), x_i is a wavelet coefficient resulting from a wavelet transform which is so normalized that the low-pass analysis filters have a unit DC gain and the high-pass analysis filters have a unit Nyquist gain [8, pp. 433]. Otherwise, e.g. the 5/3 reversible wavelet transform, x_i should be re-normalized.

The parameter α assumes a value between 0 and 1 (typically being 0.7); β assumes a small value (typically being 0.2) to distinguish coefficients around sharp edges from coefficients in a complex region; $\hat{x}_k$ denotes the quantized/dequantized version of neighboring coefficient x_k, using only the first few bits; and $|\Phi_i|$ denotes the size of the neighborhood.

The nonlinearity in (1) is believed to transform the coefficients in a subband to a perceptually uniform domain so that the same modification in z_i will incur the same visual distortion. Hence we can measure the visual distortion incurred by modifying a coefficient bit like

$$\Delta z_i = \left|\left|\breve{x}_i^1\right|^{\alpha} - \left|\breve{x}_i^0\right|^{\alpha}\right|\Big/NM[i], \tag{2}$$

where $NM[i]$ denotes the neighborhood masking effects of x_i, using the information of the first four bit-planes like

$$NM[i] = 1+\left(a\sum_{k\in neighborhood}\left|\hat{x}_k\right|^{\beta}\right)\Big/\left|\Phi_i\right|\cdot \tag{3}$$

And (2) differentiates from (1) in that $\breve{x}_i^1$, $\breve{x}_i^0$ are the quantizing results of x_i, using the first several bits that have been encoded or decoded prior to the current bit, and replacing the current bit by 1 and 0, respectively.

By assuming that $\left|\left|\breve{x}_i^1\right| - \left|\breve{x}_i^0\right|\right| << \left|x_i\right|$, equation (2) can be simplified to:

$$\Delta z_i \approx \alpha\cdot\left|\left|\breve{x}_i^1\right| - \left|\breve{x}_i^0\right|\right|\cdot\left|\breve{x}_i^0\right|^{\alpha-1}\Big/NM[i], \tag{4}$$

and further approximation yields

$$\Delta z_i \approx \alpha\cdot 2^m\Delta_b\cdot\left|\dot{x}_i\right|^{\alpha-1}\Big/NM[i] \tag{5}$$

where m is the current bit-plane depth, $\dot{x}_i$ is the quantized x_i, keeping only its most significant bit.

4.2 Visual Distortion Threshold Decision

After the visual distortion is calculated, we compare Δz_i to a certain threshold. The embedding is feasible if and only if Δz_i is no greater than the threshold. Human eyes are less sensitive to high frequency errors than to low frequency errors. The Contrast Sensitivity Function (CSF) [8, pp. 625-631] characterizes the varying sensitivity of the human visual system to 2D spatial frequencies. The thresholds should therefore be different for each subband, inversely proportional to the weighting factors of CSF:

$$\Delta z_i \le Th_b = Th_0\Big/\sqrt{W_b^{\mathrm{csf}}}\,. \tag{6}$$

The parameter Th_0 in (6) is adjustable for the tradeoff between embedding capacity and visual quality. The CSF weights, $\sqrt{W_b^{\mathrm{csf}}}$ for a medium viewing distance of 1700 pixels, are shown in Tab. 1.

Since the right hand side of (5) is monotone increasing with respect to parameter m, we do not have to calculate Δz_i bit by bit, but the maximum embedding depth, $ED[i]$, is needed:

$$ED[i] = \left\lfloor \log_2\frac{\left|\dot{x}_i^0\right|^{1-\alpha}NM[i]\cdot Th_0}{\alpha\cdot\Delta_b\cdot\sqrt{W_b^{\mathrm{csf}}}}\right\rfloor. \tag{7}$$

$ED[i]$ is determined by the magnitude of x_i and the complexity of its spatial context, and is achievable once x_i becomes significant.

4.3 Summary

In what follows we reexamine the embedding procedure outlined previously. In an individual code block, the neighborhood-masking effect for each coefficient, $NM[i]$, is first calculated based on the information of the first four bit-planes using (3). The maximum embedding depth for each significant coefficient, $ED[.]$, is also initialized using (7). Then a bit-plane scan is performed from the fifth bit-plane: each bit in the MR pass is estimated by $ED[i]$. If it is good for embedding, this bit is replaced by a bit of secret message. Before going to the next bit-plane, *ED*[.] is updated for those coefficients that have become significant in the current bit-plane. The scan will stop once the truncation point is encountered. In the end, the modified MR passes are re-packaged using raw coding.

5. SIMULATION RESULTS

Our simulation is based on JasPer, open-source software for JPEG 2000 Part I coding. Four well-known gray-level images, Boat, F16, Lena, and Peppers, with the same size of 512x512 pixels, are used as test images. A five-level irreversible DWT is performed on them, and we assume that the images will be encoded with full resolution and a maximum bit-rate of 2.0bpp. The secret messages are embedded into the code streams during compression. The decoder extracts the required bits from the code streams to obtain different image products.

Firstly, we are interested in the data-hiding capacity of the proposed steganography scheme. We set the visual distortion threshold (Th_0) to 0.5, and examine the payload of the embedded data at different decoded bit-rates (Tab. 2). The results show that the capacity increases swiftly as the decoder receives more data; and the PSNR of the decoded image also increases because the later received SP and cleanup passes still contribute to the image quality. The results also demonstrate that the hidden data may be progressively extracted by the decoder even if a small portion of data has been received.

Subjective visual distortion is another concerned issue. However it is unfair to simply compare the proposed scheme with Su's since Su's has already aims at the minimal distortion. Thus we design a "progressively backward embedding" scheme that provides similar robustness. That is, we divide a whole JPEG 2000 bit-stream into multiple layers every 0.2bpp, and perform backward embedding in each layer. The number of embedded bits in each layer is exactly equal to the proposed scheme. The testing result for Lena image is shown in Fig. 2. We notice that in the proposed scheme, most of the error energy is distributed in the visually complex areas (like hair) and well masked by the image backgrounds, whereas in the "progressively backward embedding", the error energy is obviously dispersed, thus reduce the transparency of watermarks.

6. CONCLUSIONS

We have proposed an integrated data embedding and image compression scheme to hide secret messages into JPEG 2000 code streams. Its high embedding capacity, progressive extractability and visual transparency have been proven by the experiments.

Bit-rate		0.5	1.0	1.5	2.0
Boat	Hidden Bits	3100	7051	18173	22995
	PSNR (dB)	32.79	35.41	36.82	38.24
F16	Hidden Bits	4093	13916	40762	40762
	PSNR (dB)	35.49	38.26	39.30	39.91
Lena	Hidden Bits	4076	11179	28618	30550
	PSNR (dB)	36.09	38.08	39.20	39.92
Peppers	Hidden Bits	4264	10451	10451	31125
	PSNR (dB)	34.58	36.28	37.61	38.49

Tab. 2. Number of embedded bits and PSNR of the steganographed images vs. different decoded bit-rats.

(a) (b) (c)

Fig. 2. (a) JPEG 2000 encoded Lena image at 2.0bpp; (b) error image with the proposed scheme; (c) error image with the "progressively backward embedding" scheme. The amplitudes have been amplified for display purpose.

REFERENCES

[1] JPEG 2000 Part I Final Committee Draft Version 1.0 (ISO/IEC 15444-1), ISO/IEC JTC1/SC29 WG1, Mar. 2000.

[2] JPEG 2000 Part II Final Committee Draft Version 1.0 (ISO/IEC 15444-2), ISO/IEC JTC1/SC29/WG 1, Dec. 2000.

[3] I. J. Cox, M. L. Miller, and A. L. McKellips, "Watermarking as communications with side information," *Proc. of the IEEE, Special Issue on Identification and Protection of Multimedia Information*, vol. 87, no. 7, pp. 1127–1141, Jul. 1999.

[4] Po-Chyi Su and C.-C. J. Kuo, "Information embedding in JPEG-2000 compressed images", *Proc. of the International Symposium on Circuits and Systems*, Volume 3, May 2003

[5] D. Taubman, "High Performance Scalable Image Compression with EBCOT", *IEEE Trans. on Image Processing*, Vol. 9, pp. 1158-1170, July 2000.

[6] M. Nadenau, S. Winkler, D. Alleysson and M. Kunt, "Human Vision Models for Perceptually Optimized Image Processing – A Review", *Proceedings of the IEEE*, Sep. 2000

[7] W. Zeng, S. Daly and S. Lei, "Point-wise Extended Visual Masking for JPEG-2000 Image Compression", *IEEE International Conference on Image Processing 2000*, Sep. 2000.

[8] D. Taubman, and M. Marcellin, *JPEG 2000 Image Compression Fundamentals, Standards and Practice*, published by Kluwer Academic Publishers, 101 Philip Drive, Assinippi Park, Norwell, Massachusetts, USA, 2002.