Project Report
Digital Image Watermarking
EE381K-Multidimensional Signal Processing
12/5/98

ABSTRACT

This paper discusses a new method of watermarking digital images by using convolutional codes to improve detection. The watermarks are coded, interleaved, and then embedded into the DCT coefficients of an image. Detection is performed by first extracting the DCT coefficients of the watermarked image. Then the sequence is deinterleaved and decoded using the Viterbi Algorithm. This watermarking method is compared to another scheme that does not use coding. The watermarked images are added with noise, compressed, and resized to simulate attempts to remove the watermark. Convolutional codes in watermarks prove to be robust to additive noise and JPEG compression, but vulnerable to image resizing.

1. INTRODUCTION

The recent growth in computer networks, and more specifically, the World Wide Web, has allowed multimedia data such as images to be easily distributed over the Internet. However, many publishers may be reluctant to show their work on the Internet due to a lack of security. Images can be easily duplicated and distributed without the owner's consent. Digital watermarks have been proposed as a way to tackle this issue. A digital watermark is an invisible signature embedded inside an image to show authenticity and ownership. An effective digital watermark should be perceptually invisible to prevent obstruction of the original image. It should be statistically invisible to prevent detection, and it should also be robust to many image manipulations, such as filtering, additive noise, and compression.

Several watermarking techniques have been proposed. Some methods embed the watermark in the spatial domain of the image [1,2]. Other watermarking techniques use transform methods, such as the FFT [3], DCT [4,5], or the Wavelet transform [6] to embed the watermark. Recent developments have also seen the use of the Human Visual System to improve watermark performance [7,8].

With the exception of [9], most watermarking methods do not employ error-correction. This project report proposes a new watermarking scheme using convolutional codes and compares its performance to a method without coding. Performance is based upon its robustness to common image attacks such as additive noise, JPEG compression, and image resizing. Section 2 gives a description of the proposed watermarking method, and Section 3 shows the results. Section 4 explains the potential applications of watermarking, and Section 5 gives a conclusion to this project report.

2. PROPOSED WATERMARKING METHOD

The proposed watermarking method is based on Cox, Killian, Leighton, and Shamoon [4]. Figure 1 shows a block diagram of the proposed watermarking method. The watermark,

$\boldsymbol{W(n)}$ = {0 or 1, 0<n<492} (1)

is a 492-length binary sequence; each element $\boldsymbol{W(n)}$ in the watermark is either 0 or 1. For the experiment, the watermark was generated randomly with an independent, identical distribution. Each element in the watermark could be 0 or 1 with equal probability. The watermark is fed to the input of a convolutional encoder, shown in figure 2. This is a rate ½ convolutional code—each input bit is mapped into two output bits. The output bit depends on the present input and the eight previous inputs. Therefore the convolutional encoder can be viewed as a finite state machine, where each output depends on the input and $2^8 = 256$ different states. A simpler, but less intuitive representation of this convolutional code can be defined by the generator sequences:

$\boldsymbol{G_1}$ = [1 1 1 1 0 1 0 1 1] (2)

$\boldsymbol{G_2}$ = [1 0 1 1 1 0 0 0 1] (3)

Then,

$\boldsymbol{C_1(n)} = [\ \boldsymbol{G_1(n)} \ ** \ \boldsymbol{W(n)}\ ]_2$ (4)

$\boldsymbol{C_2(n)} = [\ \boldsymbol{G_2(n)} \ ** \ \boldsymbol{W(n)}\ ]_2$ (5)

The watermark $\boldsymbol{W(n)}$ is convolved modulo-2 with the generator sequences to produce the output sequences. If L is the length of the generator sequence, and P is the length of the watermark sequence, then the lengths of $\boldsymbol{C_1(n)}$ and $\boldsymbol{C_2(n)}$ are each L+P-1. With L=9 and P=492, each length of $\boldsymbol{C_1(n)}$ and $\boldsymbol{C_2(n)}$ is 500. The coded watermark $\boldsymbol{C(n)}$ is a length 1000 vector taken as:

$\boldsymbol{C(n)} = [\boldsymbol{C_1(0)}\ \boldsymbol{C_2(0)}\ \boldsymbol{C_1(1)}\ \boldsymbol{C_2(2)}\ \ldots\ \boldsymbol{C_1(499)}\ \boldsymbol{C_2(499)}]$ (6)

Before the watermark is embedded into the original image, the coded watermark $\boldsymbol{C(n)}$ is interleaved to prevent possible burst errors—bit errors that occur in consecutive sequences. This interleaved sequence $\boldsymbol{C_i(n)}$ is finally mapped from {0,1} to {-1,+1} to form the output $\boldsymbol{C_m(n)}$. Therefore, $\boldsymbol{C_m(n)}$ is a sequence of 1000 binary numbers taking on the value of –1 or +1.

$C_m(n)$ is embedded into the 1000 highest coefficients of $I(u,v)$, the DCT of the original image $i(m,n)$. Embedding in the 1000 highest coefficients of the DCT ensures that the watermark is located in the most significant perceptual components of the image [4]. If the watermark is embedded in less significant components, then it may be possible to eliminate the watermark through compression or other attacks. The 1000 highest DCT coefficients are altered in the following manner:

$\hat{I}(u,v) = I(u,v)[1 + 0.2C_m(n)]$, $0<n<999$, $I(u,v)$ is 1000 highest coefficients

$\hat{I}(u,v) = I(u,v)$, $I(u,v)$ is not 1000 highest coefficients (7)

Because $C_m(n)$ is either -1 or +1 for each n, then the altered DCT coefficient $\hat{I}$ is either 0.8I or 1.2I. Coefficients of $I(u,v)$ that are not the highest 1000 remain unchanged. Taking the inverse DCT of $\hat{I}(u,v)$ gives the watermarked image $\hat{i}(m,n)$.

The non-coding method is similar to the coding method except that watermark does not go through convolutional coding or interleaving. The watermark length is increased to 1000 to match the DCT coefficients. The watermark sequence is mapped from {0,1} to {-1,1} and the DCT coefficients are altered in the same manner as above. Figure 3 shows the original 512 x 512 Mandrill image. Figure 4 displays the watermarked image with coding, and Figure 5 shows the watermarked image without coding. The Power Signal-to-Noise Ratio is computed as

$$PSNR = 10\log_{10}\left(\frac{\sum_m\sum_n [\hat{i}(m,n)]^2}{\sum_m\sum_n [i(m,n)-\hat{i}(m,n)]^2}\right) \quad (8)$$

The watermarked image with convolutional coding acquires a PSNR = 29.142 dB, and the watermarked image without coding obtains a PSNR = 29.121 dB. Both watermarked images appear identical to the original image, and the high PSNR values indicate that these methods are statistically close to the original image.

Figure 6 details the watermark detection method. The DCT of the suspected or corrupted image $s(m,n)$ is taken, and knowledge of the original signal $i(m,n)$ gives the location of the

altered DCT coefficients. The watermark is extracted by first performing the inverse operation of equation (7). Then the sequence is deinterleaved and decoded using the Viterbi algorithm. The Viterbi algorithm is a common method used to decode convolutional codes. It seeks to find a path through the trellis that is the minimum distance for all input/output combinations, given that the coding begins and ends in the all-zero state [10]. This decoding method performs optimal, maximum-likelihood detection given a specific output.

After decoding, the correlation coefficient

$$\rho = \frac{\sum_n W(n)W'(n)}{\sqrt{\sum_n [W(n)]^2}\sqrt{\sum_n [W'(n)]^2}}$$

is computed to determine how closely ***W(n)*** resembles ***W'(n)***. If ***W(n)*** and ***W'(n)*** are identical, then $\rho = 1$.

3. EXPERIMENT RESULTS

Random Gaussian noise was added to the watermarked Mandrill image before detection. The results are summarized in Table 1. As the noise variance increases, the correlation coefficient for both methods decrease. In all three cases, the watermarking scheme with convolutional coding performed better than the method without coding.

$N(\mu,\sigma^2)$	ρ(coding)	ρ(non-coding)
N(0,100)	1.000	0.992
N(0,400)	1.000	0.8896
N(0,900)	0.8415	0.7660

Table 1: Additive Gaussian noise N(u,s), μ = mean, σ^2 = variance, ρ = correlation coefficient

The two watermarked images were compressed under JPEG at rates of 2:1, 4:1, and 8:1. Table 2 summarizes the results. Both coding methods were extremely robust to JPEG compression, as indicated by the high correlation coefficient ρ. This is due to the fact that JPEG compression is based on removing insignificant components in the DCT of an image. Because the watermarks were embedded in the most significant components of the DCT, most of the

watermark remains unchanged. In all cases, the convolutional coding method performed better than the non-coding method. In fact, the coding method was able to reproduce the original watermark exactly.

JPEG Compression Ratio	ρ(coding)	ρ(non-coding)
2:1	1.000	0.998
4:1	1.000	0.996
8:1	1.000	0.967

Table 2: JPEG Compression, ρ = correlation coefficient

The watermarked images were resized from 512 x 512, to 384 x 384, 256 x 256, and 128 x 128, corresponding to a 75%, 50%, and 25% reduction. Once the image is reduced, significant information is lost. The images were expanded back to 512 x 512 using a bicubic interpolation, which was provided in MATLAB. Results are shown in Table 3. Both schemes were not particularly robust to image reduction, and the convolutional coding method appears to be more vulnerable to this attack then the non-coding method.

% Reduction	ρ(coding)	ρ(non-coding)
75%	0.566	0.624
50%	0.137	0.286
25%	0.043	0.174

Table 3: Image Resizing, ρ = correlation coefficient

Because the Viterbi Algorithm searches exhaustively for all input/output combinations, the amount of computations can be extremely high when compared to the non-coding method. The average time for watermark detection using the coding method was approximately two minutes; without coding, the detection time was fifteen seconds. The coding method may become a problem if many images need to be detected quickly. Sub-optimal decoding may be needed to lower the number of computations.

4. APPLICATIONS

Many potential applications exist for digital watermarking. Artists and photographers could mark their images to secure ownership rights. Publishing companies who commercially distribute their images could watermark them to prevent unauthorized distribution. Digimarc

Corporation already has a software package that searches the Internet for web pages containing watermarks [11]. Watermarking could also apply to other multimedia data such as audio and video. Compact Discs (CD) and Digital Video Discs (DVD) are extremely susceptible to bootlegging via the Internet. Digital watermarks might take part in diminishing this potential underground market.

5. CONCLUSION

Digital watermarks that employ convolutional coding prove to be more robust to additive noise and JPEG compression when compared to the non-coding method. Surprisingly, the coding method appeared more vulnerable to image resizing. Perhaps more sophisticated method of image restoration are needed to improve detection. Even though the coding method improved the watermark detection under compression and additive noise, there is a significant gain in computation due to the complexity of the Viterbi Algorithm. If detection time is an important factor, then sub-optimal methods may be used to reduce computation with a tradeoff of more errors. Nevertheless, under certain conditions, the watermarking methods that use convolutional coding have the potential to outperform their non-coding counterparts.

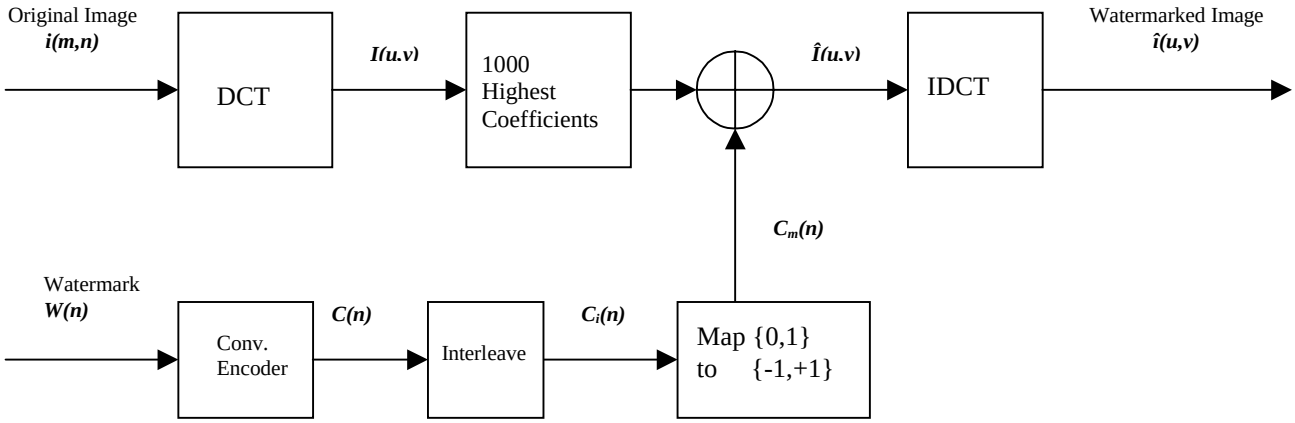

Figure 1: Proposed Watermarking Method--Block Diagram

Input=[...10110101011000000000]
G1 = [1 1 1 1 0 1 0 1 1]
G2 = [1 0 1 1 1 0 0 0 1]

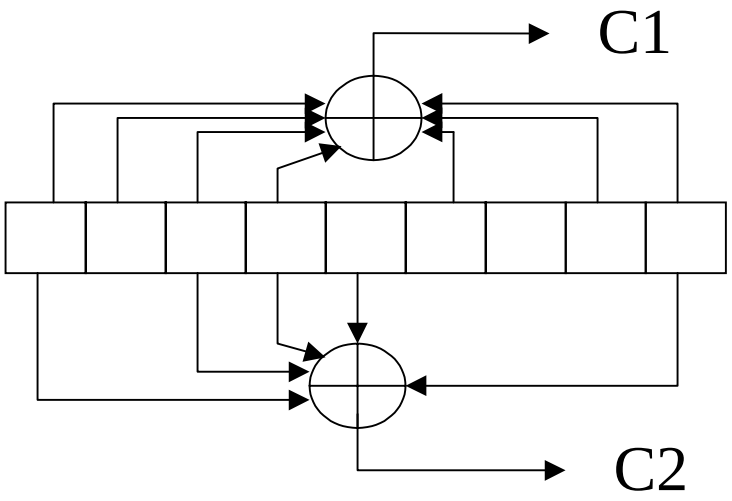

Figure 2: Convolutional Encoder

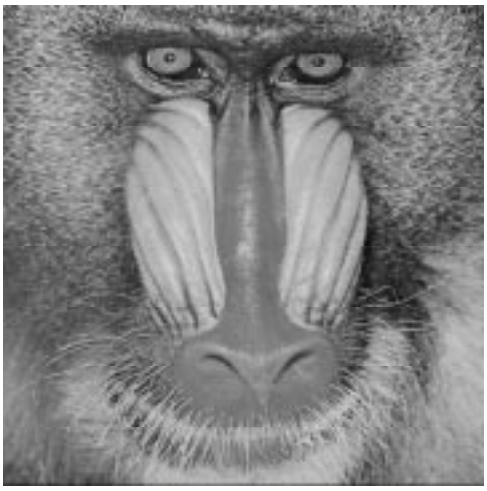

Figure 3: Original Mandrill Image

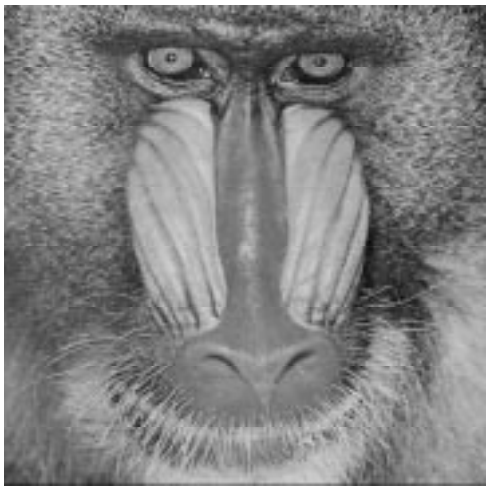

Figure 4: Watermarked Image, Convolutional Coding Method

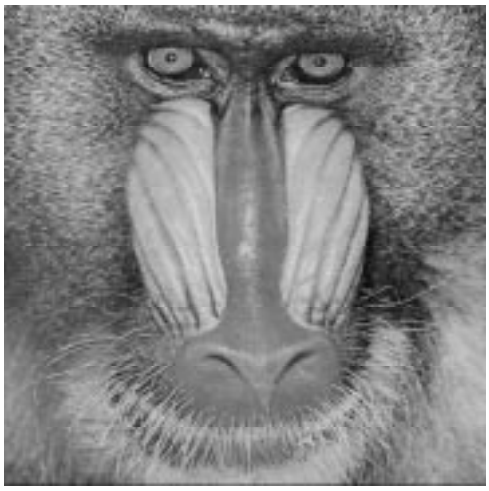

Figure 5: Watermarked Image, Without Convolutional Coding

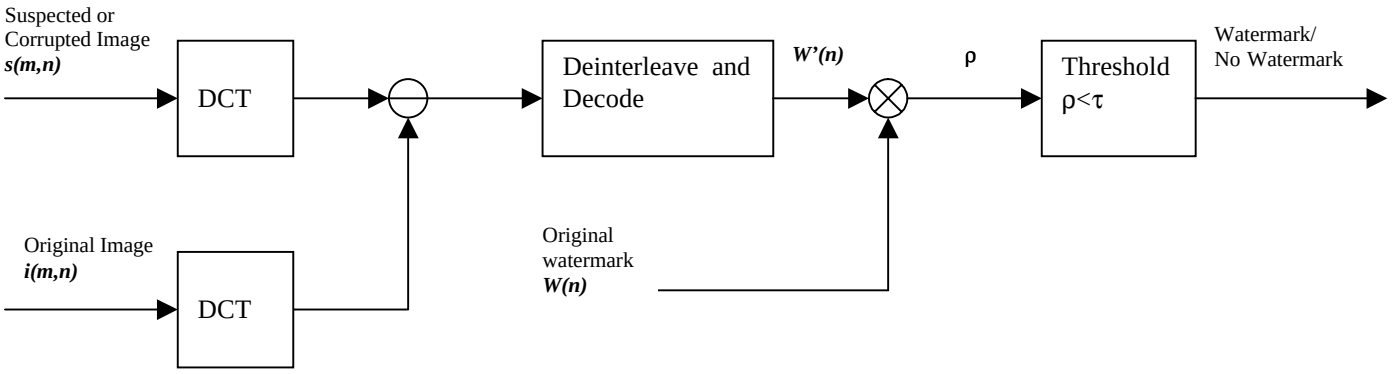

Figure 6: Watermark Detection

REFERENCES

[1] R. Schyndel, A. Tirkel, and C. Osborne, "A Digital Watermark," *Proc. IEEE Int. Conf. on Image Processing*, Nov. 1994, vol. II, pp. 86-90.

[2] I. Pitas, "A Method for Signature Casting on Digital Images," *Proc. IEEE Int. Conf. on Image Processing*, Sept. 1996, vol. III, pp. 215-218.

[3] J. O'Ruanaidh, W. Dowling, and F. Boland, "Phase Watermarking of Digital Images," *Proc. IEEE Int. Conf. on Image Processing*, Sept. 1996, vol. III, pp. 239-242.

[4] I. Cox, J. Killian, F. Leighton, and T. Shamoon, "Secure Spread Spectrum Watermarking for Multimedia Data," *IEEE Transactions on Image Processing*, vol. 6, no. 12, pp. 1673-1687, Dec. 1997.

[5] S. Craver, N. Memon, B. Yeo, and M. Yeung, "Resolving Rightful Ownerships with Invisible Watermarking Techniques: Limitations, Attacks, and Implications," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 4, pp. 573-586, May 1998.

[6] X. Xia, C. Boncelet, and G. Arce, "A Multiresolution Watermark for Digital Images," *Proc. IEEE Int. Conf. on Image Processing*, Oct. 1997, vol I, pp. 548-551.

[7] F. Bartolini, M. Barni, V. Cappellini, and A. Piva, "Mask Building for Perceptually Hiding Frequency Embedded Watermarks," *Proc. IEEE Int. Conf. on Image Processing*, Oct. 1998, vol. I, pp. 450-454.

[8] J. Delaigle, C. De Vleeschouwer, and B. Macq, "Psychovisual Approach to Digital Picture Watermarking," *Journal of Electronic Imaging*, vol.7, no. 3, pp. 628-640, July 1998.

[9] L. Marvel, C. Retter, and C. Boncelet, "Hiding Information in Images," *Proc. IEEE Int. Conf. on Image Processing*, vol. II, Oct. 1998, pp. 396-398.

[10] J.G. Proakis, and MK. Salehi, *Communication Systems Engineering*, Prentice-Hall, Inc., ISBN 0-13-158932-6, 1994.

[11] http://www.digimarc.com