An Improved Self-embedding Algorithm: Digital Content Protection against Compression Attacks in Digital Watermarking

Pratheepan Yogarajah[1,⋆], Joan Condell[1], Kevin Curran[1], Abbas Cheddad[2], and Paul McKevit[1]

[1] School of Computing and Intelligent Systems, University of Ulster, UK
[2] Umea Centre for Molecular Medicine (UCMM), Umea Universitet, Sweden

Summary. Lossy compression attacks in digital watermarking are one of the major issues in digital watermarking. Cheddad et al. proposed a robust secured self-embedding method which is resistant to a certain amount of JPEG compression. Our experimental results show that the self-embedding method is resistant to JPEG compression attacks and not resistant to other lossy compression attacks such as Block Truncation Coding (BTC) and Singular Value Decomposition (SVD). Therefore we improved Cheddad et al's. method to give better protection against BTC and SVD compression attacks.

1 Introduction

Protecting digital image content is an important task in image security. To protect the content, digital image watermarking techniques are applied. In watermarking the secret information called the watermark, is invisibly embedded into the host digital image. A general watermarking framework for content protection is presented in [1].

Watermarking techniques can be categorized into two types (spatial and frequency domain) according to the embedding process. Watermarking in the frequency domain is more robust than watermarking in the spatial domain [2], because the watermark information can be spread out over the entire image [3]. Commonly used frequency domain transforms include the Discrete Wavelet Transform (DWT), the Discrete Cosine Transform (DCT) and the Discrete Fourier Transform (DFT). However, DWT [4] has been used in digital image watermarking more frequently due to its excellent spatial localisation and multi-resolution characteristics, which are similar to the theoretical models of Human Visual System (HSV) [5].

⋆ This work was part funded by the Invest NI Proof of Concept (PoC) fund.

R.S. Choraś (Ed.): Image Processing & Communications Challenges 2, AISC 84, pp. 59–66.
springerlink.com

The content of watermarked digital images can be easily attacked by using image processing operations such as lossy compression. Invisible watermarking requires a reasonable robustness against compression attacks. Lossy compression algorithms tend to remove invisible information that can be related to the watermark. Watermark robustness under image compression is an essential issue for image content protection. Therefore, watermarks should combine invisibility and robustness simultaneously.

Recently Cheddad et al [6] proposed a method to protect the digital image itself using a secured robust self-embedding technique. In their method, a halftoned version (black and white image) of the original image is used as watermark. The calculated watermark is embedded in the 2D Haar DWT of the original image and the watermarked image is obtained. Then the Wavelet-based Inverse Halftoning via De-convolution (WInHD) is used on the extracted watermark from the watermarked image to recover the approximation of the original image. This is a blind watermarking scheme as the original image is not needed for the recovery process, see Figure 1.

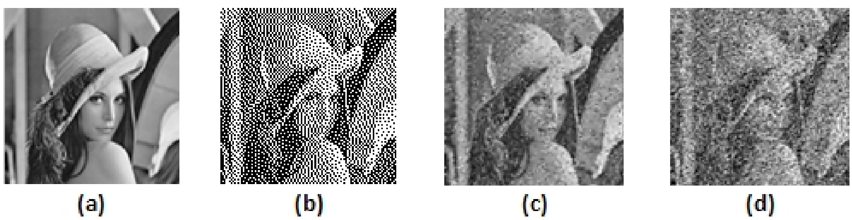

Fig. 1. (a) Original image, (b) halftoned binary image, (c) and (d) are recovered images from JPEG 95% and 85% quality compression attacks respectively

JPEG 2000 is one of the modern lossy compression methods and it is based on DWT. As the Cheddad et al. method is DWT based, it is resistant to JPEG compression attacks to a certain extent. They reported that their method is resilient to JPEG compression up to 80-75% [6]. There is no experimental results shown for other lossy compression techniques, such as Block Truncation Coding (BLC) and Singular Value Decomposition (SVD) etc. Our experimental results show that Cheddad et al's method is not robust to BTC and SVD lossy compression techniques. Therefore we improved Cheddad et al's method and experimental results prove that our method provides better recovery results on BTC and SVD compression attacks.

Our method is discussed in Section 2. The secure image encryption algorithm is explained in section 3. Sections 4 and 5 explain the results and provide discussion and conclusions respectively.

2 Our Method

In DWT, an image is decomposed into a set of basis functions namely low frequency band (LL), high-low frequency band (HL), low-high frequency band (LH) and high frequency band (HH), see Figure 2(a). The low frequency band is a lowpass approximation of the original image and includes most energy of the image. The other bands include edge components of horizontal, vertical and diagonal directions at different scales and resolutions respectively.

According to the energy distribution, LL, is the most important. Hence in DWT domain, watermarks should be embedded in the low frequency band [7]. Cheddad et al. also selected the low frequency band of the 1^{st}-level 2D Haar DWT as their embedding area (i.e. LL_1) [6].

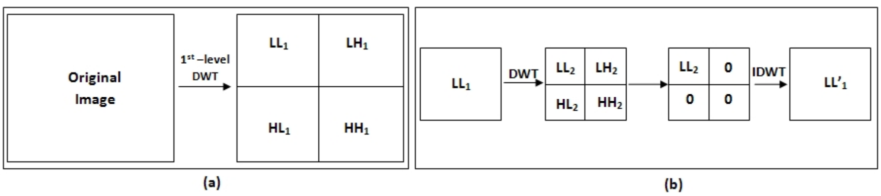

Fig. 2. (a) 1^{st}-level DWT decomposition and (b) LL_1' calculation

Common image processing procedures, which watermarked images may encounter, such as data compression, low-pass filtering and subsampling, tend to change the low frequency coefficients less than high frequency coefficients [7]. Therefore it is necessary to reduce the high frequency coefficient effects to make the watermarked image resistant to compression attacks.

DWT has applications in image processing, where typically the approach is to DWT an image, alter the transform coefficients (by thresholding or zeroing), and inverse DWT to regain an altered image that has been de-noised, or its edges sharpened or blurred. The zeroing high frequency coefficient technique is applied to the digital image watermarking application in [8].

We also applied the zeroing technique to improve the performance of Cheddad et al.'s method against lossy compression attacks. In our method the original LL_1 is further wavelet transformed and then three high frequency bands (LH_2, HL_2 and HH_2, excluding LL_2) are initialised to zeros and inverse wavelet transformed. We then obtain another LL_1' from the process, see Figure 2(b). In our method, this newly calculated LL_1' is used for embedding instead of LL_1.

In our digital image content protection method, first the digital image, that needs to be protected against compressions attacks, is selected (say 256x256 'Lena' grayscale image). Then it is transformed to the DWT domain and LL_1' is generated. The size of LL_1' is half that of the original image. Therefore the original image is resized to be equal to the size of LL_1' and Floyd's [9] error diffusion digital halftoning technique is applied to obtain the watermark (i.e. a halftoned black and white image).

This watermark is embedded in LL'_1 using an encryption algorithm explained in section 3 and the inverse DWT is applied to generate the watermarked image. The embedded watermark is extracted from the compression attacked watermarked image using a decryption algorithm (again see Section 3). Finally the Wavelet-based Inverse Halftoning via De-convolution (WInHD) [10] is applied to the extracted watermark to recover the approximation of the original image.

Embedding the watermark into LL'_1, instead of LL_1, may decrease the watermarked image quality, see Figure 3(b), but the extracted watermark reliability is increased, see Figure 3(d). The BTC and SVD compression attacks are applied to watermarked images. Brief information on BTC and SVD given below.

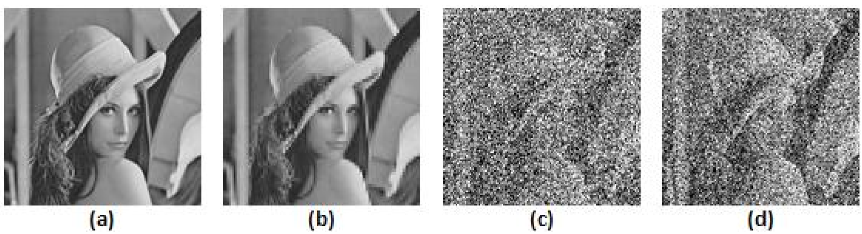

Fig. 3. (a) watermarked image using Cheddad et al's method, (b) watermarked image using our method, (c) recovered image from 4x4 blocks BTC compression attack using Cheddad et al's method and (d) recovered image from 4x4 blocks BTC compression attack using our method

Block Truncation Coding (BTC) is a lossy image compression technique. It divides the original images into blocks and then uses a quantizer to reduce the number of grey levels in each block while maintaining the same mean and standard deviation [11]. Sub blocks of 4x4 pixels allow compression of about 25%. Larger blocks allow greater compression however quality also reduces with the increase in block size due to the nature of the algorithm.

Singular Value Decomposition (SVD) is one of the most useful tools of linear algebra. It is a factorization and approximation technique which effectively reduces any matrix into a smaller invertible and square matrix. Using (SVD) for image compression can be a very useful tool to save storage space [12, 13].

3 Image Encryption Algorithm

This algorithm is explained based on [6] and the encryption algorithm is fully described in [14]. A hash function is more formally defined as the mapping of bit strings of an arbitrary finite length to strings of fixed length [15]. Here we attempt to extend SHA-1 (the terminology and functions used as building blocks to form SHA-1 are described in the US Secure Hash Algorithm 1, [16]) to encrypt digital 2D data. The introduction of Fast Fourier Transform (FFT) forms together with the output of SHA-1 a strong image encryption setting. Let the key bit stream be $\lambda_{k,l}$ where the subscripts k and l denote the width and height after resizing the key's bit stream respectively, i.e., 8, $M * N$, where M, N are the plain image's dimensions.

The FFT will operate on the DCT transform of $\lambda_{k,l}$ subject to Eq. 2.

$$f(u,v) = \frac{1}{N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} F(x,y) e^{-2\pi i(xu+yv)/N} \qquad (1)$$

where $F(x,y) = DCT(\lambda_{k,l})$ satisfying Eq. (2). Note that for the transformation at the FFT and DCT levels we do not utilise all of the coefficients. Rather, we impose the following rule, which generates at the end a binary random-like map. Given the output of Eq. 1 we can derive the binary map straightforwardly as:

$$Map(x,y) = \begin{cases} 1 & iff \quad f(u,v) > 0 \\ 0 & otherwise \end{cases} \qquad (2)$$

This map takes the positive coefficients of the imaginary part to form the ON pixels in the map. Since the coefficients are omitted the reconstruction of the password phrase is impossible, hence the name Irreversible Fast Fourier Transform (IrFFT). In other words, it is a one way function which accepts initially a user password. This map finally is XORed with the binary version of each colour component separately. The core idea here is to transform these changes into the spatial domain where we can apply 2D-DCT and 2D-FFT that introduce the aforementioned sensitivity to the two dimensional space. As such, images can be easily encoded securely with password protection.

The watermark images are securely embedded using the encryption algorithm explained above. The decryption technique is also similar to the encryption algorithm and can be referred in [14].

4 Results and Discussion

In this section, we illustrate and evaluate the performance of the proposed method against JPEG, BTC and SVD compression attacks on grayscale images. Here we present experimental results using the image 'Lena' (256x256 pixels, grayscale). The 'Lena' image is shown in Figure 1(a).

Then the watermarked images are generated using Cheddad et al. and our method, see Figure 3(a) and (b). These watermarked images are attacked by JPEG, BTC and SVD compression techniques. The binary watermarks are extracted from attacked watermarked images and the approximation of the original image is recovered.

For quantitative evaluation, the PSNR (Peak Signal-to-Noise Ratio) is introduced to evaluate the performance between the original image and recovered image. The PSNR is defined as follows:

$$\mathbf{PSNR} = 10log_{10}(\frac{255^2}{\mathbf{MSE}})dB \tag{3}$$

$$\mathbf{MSE} = \sum_{i=1}^{n}\sum_{j=1}^{m}\frac{(a_{i,j} - b_{i,j})^2}{n * m} \tag{4}$$

where $m * n$ is the image size, $a_{i,j}$ and $b_{i,j}$ are the corresponding pixel values of cover and recovered images.

4.1 JPEG Compression Attack

From Figure 4 we see that both methods perform similarly against JPEG compression quality factors 75%, 85%, 90% and 95%.

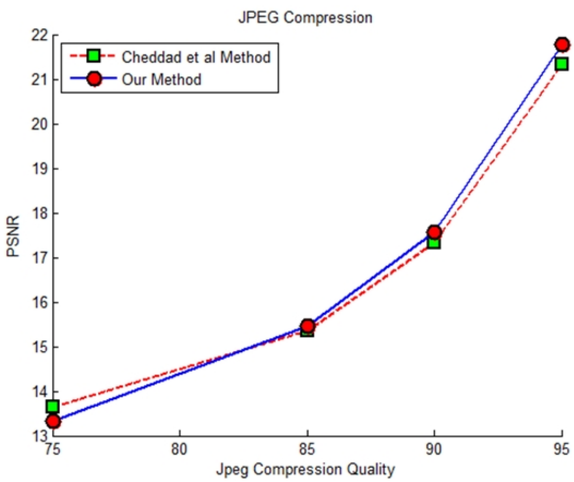

Fig. 4. Results of JPEG compression attack on 'Lena' image

4.2 BTC Compression Attack

Here 2x2, 4x4, 8x8 and 16x16 blocks are selected for experiments. When we apply 8x8 and 16x16 blocks BTC attacks on the watermarked image, the watermarked image becomes severely corrupted intensity of pixels. Therefore we could not see much difference in performance with 8x8 and 16x16 blocks BTC compression. Our method performed better when 2x2 and 4x4 blocks BTC compressions were applied to the watermarked images, see Figure 5.

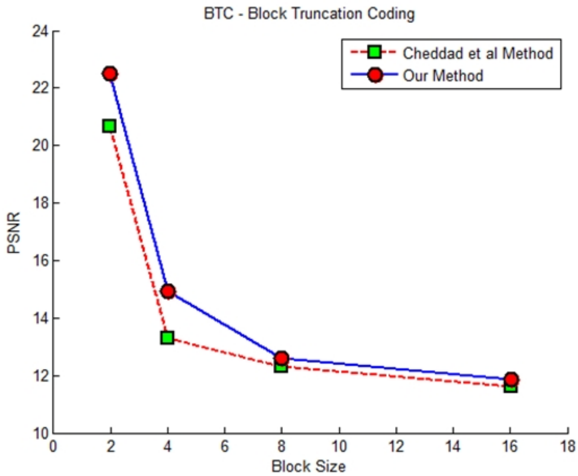

Fig. 5. Results of BTC compression attack on ‘Lena’ image

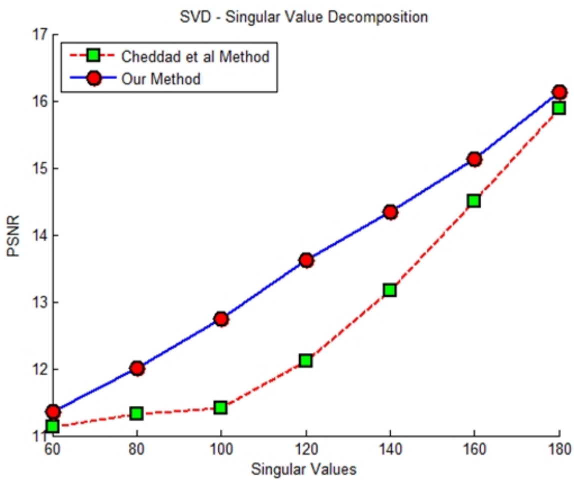

Fig. 6. Results of SVD compression attack on ‘Lena’ image

4.3 SVD Compression Attack

Based on Figure 6, we can see that when 60 and 180 singular values are used for SVD compression attacks, the performances are similar for both methods. When 60 singular values are used, the recovered images from both methods are very noisy. At the same time, when 180 singular values are used, the quality of the recovered images from both methods are more similar. When the singular values between 60 and 180 are used, our method outperformed Cheddad et al’s method.

5 Conclusion

Lossy compression attacks in digital watermarking are one of the major issues when sending digital images over the internet. In this paper, we improved Cheddad et al's self-embedding method to make it resistant to lossy compression attacks such as BTC and SVD. Our experimental results show evidence that the original content of the digital image can be recovered to a certain extent even though the watermarked image is attacked by lossy compression such as JPEG, BTC and SVD. We only experimented with grayscale images. Future work will involve making our method more applicable to a broader range of images, in particular colour images.

References

1. Voyatzis, G., Pitas, I.: Protecting digital-image copyrights: a framework. IEEE Computer Graphics and Applications 19, 18–24 (1999)
2. Bender, W., Gruhl, D., Morimoto, N., Lu, A.: Techniques for data hiding. IBM Systems Journal 35(3&4), 313–336 (1996)
3. Cox, I.J., Killian, J., Leighton, T., Shammon, T.: Secure spread spectrum for multimedia. IEEE Transactions on Image Processing 6(12), 1673–1687 (1997)
4. Vetterli, M., Kovacevic, J.: Wavelet and Subband Coding. Prentice-Hall, Englewood Cliffs (1995)
5. Wolfgang, R.B., Podichuk, C.I., Delp, E.J.: Perceptual watermarks for digital images and video. Proceedings of the IEEE 87(7), 1108–1126 (1999)
6. Cheddad, A., Condell, J., Curran, K., McKevitt, P.: A secure and improved self-embedding algorithm to combat digital document forgery. Signal Processing 89(12), 2324–2332 (2009)
7. Daren, H., Jiufen, L., Jiwu, H., Hongmei, L.: A DWT-based image watermarking algorithm. In: IEEE Int. Conf. Multimedia and Expo., pp. 429–432 (2001)
8. Joo, S., Suh, Y., Shin, J., Kitkuchi, H.: A New Robust Watermark Embedding into Wavelet DC Components. ETRI Journal 24(5), 401–404 (2002)
9. Floyd, R.W., Steinberg, L.: An adaptive algorithm for spatial grayscale. In: Proc. Soc. Inf. Disp., vol. 12, pp. 55–77 (1976)
10. Neelamani, R., Nowak, R., Baraniuk, R.G.: WInHD: wavelet-based inverse halftoning via deconvolution. submitted to IEEE Trans. Image Process. for publication (2002)
11. Chanda, B., Dutta Majumder, D.: Digital Image Processing and Analysis. Prentice-Hall, Englewood Cliffs (2000)
12. Richards, D., Abrahamsen, A.: Image compression using singular value decomposition. linear algebra applications (2001)
13. Prasantha, H.S., Shashidhara, H.L., Balasubramanya Murthy, K.N.: Image Compression using SVD. In: Proc. of International Conference on Computational Intelligence and Multimedia Applications, pp. 143–145 (2007)
14. Cheddad, A., Condell, J., Curran, K., McKevitt, P.: A Hash-based Image Encryption Algorithm. Opt. Comm. Elsevier Science. 283(6), 879–893 (2010)
15. Wang, Y., Liao, X., Xiao, D., Wong, K.: One-way hash function construction based on 2D coupled map lattices. Inf. Sci. 178(5), 1391–1406 (2008)
16. US Secure Hash Algorithm 1 (2001), http://www.faqs.org/rfcs/rfc3174