WAVELET-BASED IMAGE COMPRESSION ANTI-FORENSICS

Matthew C. Stamm and K. J. Ray Liu

Dept. of Electrical and Computer Engineering, University of Maryland, College Park

ABSTRACT

Because digital images can be modified with relative ease, considerable effort has been spent developing image forensic algorithms capable of tracing an image's processing history. In contrast to this, relatively little consideration has been given to anti-forensic operations designed to mislead forensic techniques. In this paper, we propose an anti-forensic technique capable of removing artifacts indicative of wavelet-based image compression from an image. Our technique operates by adding anti-forensic dither to a previously compressed image's wavelet coefficients so that the anti-forensically modified wavelet coefficient distribution matches a model of the coefficient distribution before compression. Simulation results show that our algorithm is capable of fooling current forensic image compression detection algorithms 100% of the time.

Index Terms— Anti-Forensics, Digital Forensics, Image Compression

1. INTRODUCTION

Within the past decade, a significant amount of research has been performed in the field of digital image forensics. Image forensics seeks to provide information about an image without relying on external descriptors such as metadata tags or extrinsically implanted information such as digital watermarks. Operations have been developed to perform diverse tasks such as detecting evidence of image forgery [1], tracing an image's compression history [2], and determining an image's origin [3]. These operations have proven to be particularly important due to the existence of a wide variety of software that allows users to modify both an image's metadata as well as the image itself. As a result, image forensics have been increasingly used to analyze and verify the authenticity of images used by governmental, legal, scientific, and news media organizations.

While much effort has been spent on the study of image forensics, very little consideration has been given to *anti-forensic* countermeasures designed to deceive image forensic operations. An image manipulator with access to such countermeasures may use them to disguise changes which he or she has made to an image or to falsify an image's origin or processing history. Though few studies of anti-forensic image processing operations have been published, it is quite possible that image forgers familiar with both image forensics and signal processing in general have independently developed anti-forensic operations that they have not made public. As a result, several image forensic techniques may possess unknown vulnerabilities, thus allowing certain image forgeries to remain undetected.

To prevent this scenario, it is imperative that the image forensic community develop and study anti-forensic operations so that forensic examiners may know when the results of forensic tests, particularly those that do not show evidence of image manipulation, can be trusted. The study of image anti-forensics promises additional benefits as well. By examining anti-forensic image processing operations, researchers may develop techniques capable of detecting when anti-forensic countermeasures have been employed. Additionally, anti-forensic operations may be developed to prevent image forensic techniques from being used to reverse engineer proprietary signal processing components within digital cameras.

Recently, we proposed an anti-forensic operation capable of disguising an image's JPEG compression history [4]. Other anti-forensic operations that have been studied include those aimed at disguising evidence of image resizing and rotation [5], artificially synthesizing color filter array artifacts [6], and forging the unique noise pattern introduced into an image by a digital camera's charged coupling device [5].

In this paper, we propose an anti-forensic operation built upon the techniques developed in [4] that is capable of falsifying an image's wavelet compression history. We accomplish this by removing the artifacts which wavelet-based compression schemes introduce into an image's wavelet coefficient histograms. After our anti-forensic operation is applied, an image can be passed off as never-compressed, thereby allowing forensic investigators to be misled about an image's origin and processing history. We evaluate the effectiveness of our anti-forensic operation by testing it against the current state-of-the-art forensic technique for tracing an image's compression history [2].

2. WAVELET-BASED COMPRESSION ARTIFACTS

At present, several wavelet-based image compression schemes exist such as JPEG2000, SPIHT, and the EZW algorithm. Though each scheme performs compression in a different manner, all leave behind similar forensically detectable traces. To understand what these traces are and why they occur, we give a brief overview of how most wavelet-based image compression is performed. We assume that the image undergoing compression is a grayscale image with integer pixel values in the set $\mathbb{P} = \{0, \ldots, 255\}$.

Each algorithm begins by computing the two-dimensional discrete wavelet transform an image, resulting in four subbands of wavelet coefficients denoted by LL, LH, HL, and HH. This process is repeated on the LL subband M times to achieve an M-level wavelet decomposition. In JPEG2000, the image may be first divided into a set of equally sized tiles, each of which separately undergo this process rather than the image as a whole.

Tree-based schemes such as SPIHT and EZW proceed by separating each subband into a series of bit planes, then generating a significance map for each bit plane [7]. This is done by constructing trees of insignificant coefficients which are spatially correlated across different levels of a particular subband. The significance maps are then scanned into a single bitstream beginning with the significance map of most significant bit plane then proceeding downward. Lossy compression is achieved by terminating the bitstream when a particular bit budget is exhausted or equivalently by reading the entire transformed image into a single bitstream, then truncating it to a fixed number of bits. By contrast, other schemes such as JPEG2000

Email: {mcstamm,kjrliu}@umd.edu

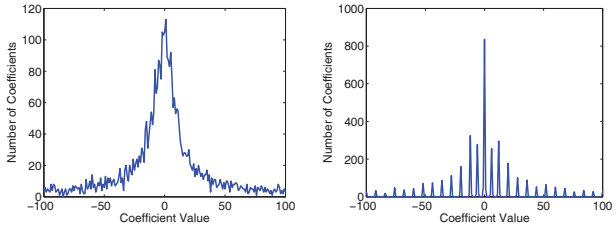

Fig. 1. Left: Histogram of wavelet coefficients from an uncompressed image. Right: Histogram of wavelet coefficients from the same image after SPIHT compression.

achieve compression by simply performing scalar quantization on the coefficients of each of the wavelet subbands, then scanning them into a single bitstream.

Decompression is performed by using the compressed bitstream to construct an approximation of the wavelet transformed image. In the case of tree-based schemes, all bit plane data lost due to the truncation of the uncompressed bitstream is set to zero. For JPEG2000 and similar algorithms, dequantization is performed by multiplying each quantized coefficient value by its corresponding quantization step size. Finally, the two-dimensional inverse discrete wavelet transform is performed on each level of the wavelet decomposition and the resulting pixel values are projected back into $\mathbb{P}$ to recover the decompressed image.

During this process, SPIHT, EZW, and similar schemes introduce compression artifacts into the set of wavelet coefficients by truncating the encoded bitstream. Because the bitstream is comprised of the significance map of each bit plane and order such that the most significant bit plane occurs first followed by bit planes of progressively lower significance, the process of truncating the bitstream and replacing the lost bits with zeros will cause the reconstructed wavelet coefficients to cluster around certain integer values. This effect can be seen in Fig. 1 which shows the histogram of second level HL wavelet coefficients taken from both an uncompressed image and one which has been compressed using the SPIHT algorithm. A similar effect is produced by the quantization and dequantization of wavelet coefficients which occurs in JPEG2000. It is these quantization or bitstream truncation artifacts which are used to detect previous applications of wavelet-based image compression [2].

3. COMPRESSION ARTIFACT REMOVAL

If we wish to create an anti-forensically modified image which exhibits no signs of prior wavelet-based compression, it is clear that we must remove all compression artifacts from the image's wavelet subband coefficient histograms. To do this, we propose an approach similar to the one we introduced in [4]. Namely, we propose estimating the distribution of each subband's wavelet coefficients before compression from the set of compressed coefficients, then adding noise to the set of compressed wavelet coefficients so that the distribution of anti-forensically modified coefficients approximates that of the coefficients before compression. The distribution of the additive noise, which we will refer to hereafter as anti-forensic dither, is chosen to be conditionally dependent upon the coefficient value to which it is being added. This is done to both ensure that the distribution of anti-forensically modified wavelet coefficients matches the estimated distribution of unmodified coefficients and to minimize the image distortion introduced by the anti-forensic dither.

3.1. Wavelet Coefficient Distrtibution Model

For SPIHT, EZW, or similar compression schemes, the effect of truncating the encoded bitstream can be modeled as quantization with nonuniform quantization intervals. Let $\{\ldots, q_{-1}, q_0, q_1, \ldots\}$ and $\{\ldots, b_{-1}, b_0, b_1, \ldots\}$ denote the sets of quantized values and quantization boundaries respectively, where $q_0 = 0$, $q_k < q_{k+1}$, and $b_k < b_{k+1}$. A wavelet coefficient Y in a compressed image can be written in terms of its corresponding wavelet coefficient X in an uncompressed image using the equation

$$Y = q_k \quad \text{if } b_k \leq X < b_{k+1}. \tag{1}$$

By contrast, JPEG2000 employs scalar quantization and places the resulting quantized values at the center of each quantization interval rather than at the end with the smallest magnitude. If the quantized values are properly modified, however, scalar quantization becomes a simplification of the quantization rule described in (1). Accordingly, all methods developed to remove compression artifacts from SPIHT, EZW, or similarly compressed images can be simply adapted to accommodate JPEG2000. Because of this, we will only consider tree-based compression schemes for the remainder of this paper.

We model the distribution of an uncompressed image's wavelet coefficients within each subband using the Laplace distribution [8]

$$P(X = x) = \tfrac{\lambda}{2} e^{-\lambda |x|}. \tag{2}$$

Using (1) and (2), the distribution of wavelet coefficients in a compressed image can be written as

$$P(Y = q_k) = \begin{cases} \frac{1}{2}(e^{-\lambda b_k} - e^{-\lambda b_{k+1}}) & \text{if } k \geq 1, \\ 1 - \frac{1}{2}(e^{\lambda b_0} + e^{-\lambda b_1}) & \text{if } k = 0, \\ \frac{1}{2}(e^{\lambda b_{k+1}} - e^{\lambda b_k}) & \text{if } k \leq -1. \end{cases} \tag{3}$$

3.2. Estimation of the Uncompressed Coefficient Distribution

In order to choose the appropriate anti-forensic dither distribution for each subband, we must first estimate the distribution of wavelet coefficients before compression using the coefficients obtained from the compressed image. Since we employ a parameterized model of this distribution, only the parameter λ must be estimated. To accomplish this, we fit the nonzero entries of the histogram of compressed wavelet coefficients within each subband to the function

$$h_k = c e^{-\hat{\lambda}|q_k|} \tag{4}$$

where h_k denotes the histogram value at q_k, and use $\hat{\lambda}$ as our estimate of λ. Theoretically, all wavelet coefficients should take values in the set of quantization values, however, the process of projecting the pixels in the decompressed image back into the set $\mathbb{P}$ slightly perturbs these values. This can be compensated for by simply rounding each wavelet coefficient to the nearest value in the set of quantized coefficients.

The histogram is fitted to the model by solving the following weighted least squares minimization problem

$$\min_{\hat{\lambda}, c} \sum_k h_k (\log h_k - \log c + \hat{\lambda}|q_k|)^2 \tag{5}$$

where the model has been linearized by taking the logarithm of both sides of (4) and the model errors are weighted by h_k, the number of observations of each quantized value. By taking the derivative of the function to be minimized with respect to λ and to c, then setting each derivative equal to zero, the resulting equations can be rearranged into the following matrix equation

$$\begin{bmatrix} \sum_k h_k & \sum_k |q_k| h_k \\ \sum_k |q_k| h_k & \sum_k |q_k|^2 h_k \log h_k \end{bmatrix} \begin{bmatrix} \log c \\ -\hat{\lambda} \end{bmatrix} = \begin{bmatrix} \sum_k h_k \log h_k \\ \sum_k |q_k| h_k \log h_k \end{bmatrix}, \tag{6}$$

which can be solved to obtain a value for $\hat{\lambda}$.

In practice, this estimate may be biased due to a consequence of tree-based compression which causes the histogram to deviate from our model. Recall that the encoded bitstream consists of a significance map of each bit plane arranged in descending order. Because truncation is unlikely to occur on the boundary between the maps of two different bit planes, several nonzero entries in map of the lowest retained bit plane will be truncated, then set to zero during decompression. As a result, fewer wavelet coefficients will take the values q_1 and q_{-1} and more coefficients will take the value 0 than our model predicts. This also affects the manner in which we add anti-forensic dither to the wavelet coefficients, which we discuss in greater detail in the subsequent section.

To compensate for this effect, we make use of the following iterative processes:

1. Use (6) to estimate $\hat{\lambda}^{(i)}$ and $\hat{c}^{(i)}$ given the current histogram iterate $\hat{h}^{(i)}$.
2. Update the histogram estimate using the equation:

$$\hat{h}_k^{(i)} = \begin{cases} c^{(i)} & \text{if } k = 0, \\ h_k + \frac{1}{2}(h_0 - c^{(i)}) & \text{if } k = \pm 1, \\ h_k & \text{otherwise.} \end{cases} \tag{7}$$

3. Terminate if $\frac{\hat{\lambda}^{(i)} - \hat{\lambda}^{(i-1)}}{\hat{\lambda}^{(i)}} < \tau$, where τ is a user defined threshold. Otherwise, update the iteration number and return to step 1.

The iteration is initialized by setting $h_k^{(0)} = h_k$ for all k and $\hat{\lambda}^{(0)} = 0$. Upon termination, the current value of $\hat{\lambda}^{(i)}$ is assigned to $\hat{\lambda}$.

3.3. Anti-Forensic Dither

Once the parameter $\hat{\lambda}$ has been estimated for a particular subband, the process of removing compression artifacts from that subband's wavelet coefficient histogram (through the addition of anti-forensic dither) can begin. Before we add anti-forensic dither to the set of compressed coefficients, however, we must again address the mismatch between the modeled distribution and actual distribution of compressed coefficients.

Since we cannot predict where in the bitstream truncation will occur, we cannot appropriately adjust our model of the compressed coefficient distribution to account for this behavior. Instead, we first calculate N_e, the number of zero valued wavelet coefficients in excess of what our model predicts using the equation

$$N_e = h_0 - N_s(1 - \tfrac{1}{2}(e^{\hat{\lambda}b_0} + e^{-\hat{\lambda}b_1})), \tag{8}$$

where N_s is the total number of wavelet coefficients in the current subband. After this is done, N_e zero-valued coefficient are chosen at random. Half of these coefficients are changed from 0 to q_1 while the other half are changed from 0 to q_{-1}, resulting in a coefficient distribution that agrees with our model.

After this is performed, anti-forensic dither D is added to each wavelet coefficient, resulting in a set of anti-forensically modified coefficients Z where

$$Z = Y + D. \tag{9}$$

As was mentioned before, the anti-forensic dither's distribution is conditionally dependent on the value of Y. When adding dither to nonzero valued wavelet coefficients, the dither distribution is

$$P(D = d|Y = q_k, k \neq 0) = \begin{cases} \frac{1}{\alpha_k} e^{-\operatorname{sgn}(q_k)\hat{\lambda}d} & \text{if } (b_k - q_k) \leq d < (b_{k+1} - q_k), \\ 0 & \text{otherwise,} \end{cases} \tag{10}$$

where $\alpha_k = \frac{1}{\hat{\lambda}}(e^{-\operatorname{sgn}(q_k)\hat{\lambda}(b_k - q_k)} - e^{-\operatorname{sgn}(q_k)\hat{\lambda}(b_{k+1} - q_k)})$. For coefficients whose value is zero, the distribution of the dither is

$$P(D = d|Y = 0) = \begin{cases} \frac{1}{\alpha_0} e^{-\hat{\lambda}|d|} & \text{if } b_0 > d > b_1, \\ 0 & \text{otherwise,} \end{cases} \tag{11}$$

where $\alpha_0 = \frac{1}{\hat{\lambda}}(2 - e^{-\hat{\lambda}b_1} - e^{\hat{\lambda}b_0})$.

By choosing the anti-forensic dither distributions in this manner, we are able to ensure that the distribution of anti-forensically modified wavelet coefficients matches the modeled distribution of coefficients before compression, provided that $\hat{\lambda} = \lambda$. This can be seen by using (3), (10), and (11) to determine the expression for the probability distribution of Z:

$$\begin{aligned} P(Z = z) &= \sum_k P(Z = z|Y = q_k) P(Y = q_k) \\ &= \sum_{k \leq -1} \tfrac{1}{\alpha_k} e^{\lambda(z - q_k)} \tfrac{1}{2}(e^{\lambda b_{k+1}} - e^{\lambda b_k}) \, \mathbb{1}(b_k \leq z < q_{k+1}) \\ &\quad + \tfrac{1}{\alpha_0} e^{-\lambda|z|}(1 - \tfrac{1}{2}(e^{\lambda b_0} + e^{-\lambda b_1})) \, \mathbb{1}(b_0 \leq z < b_1) \\ &\quad + \sum_{k \geq 1} \tfrac{1}{\alpha_k} e^{-\lambda(z - q_k)} \tfrac{1}{2}(e^{-\lambda b_k} - e^{-\lambda b_{k+1}}) \, \mathbb{1}(b_k \leq z < q_{k+1}) \\ &= \tfrac{\lambda}{2} e^{-\lambda|z|}, \end{aligned} \tag{12}$$

where $\mathbb{1}(\cdot)$ denotes the indicator function.

An additional benefit of this choice of dither distributions is that it allows bounds to be placed on the difference between a wavelet coefficient's value before and after it has undergone anti-forensic modification. Since the support of the anti-forensic dither's distribution is dependent upon the length of the interval between the compressed wavelet coefficient's value and the value either immediately large or smaller than it, the following bounds can be placed

$$|Y - Z| \leq \begin{cases} b_{k+1} - b_k & \text{if } k \neq 1, \\ b_2 - b_{-1} & \text{if } k = 0. \end{cases} \tag{13}$$

Furthermore, since the wavelet coefficient's value before compression must lie within the same interval that the anti-forensically modified coefficient does, the bounds expressed in (13) apply to the quantity $|X - Z|$ as well.

4. SIMULATIONS AND RESULTS

Fig. 2 shows the Lena image compressed using the SPIHT algorithm at a bit rate of 3 bits per pixel both before and after we have applied anti-forensic dither to its wavelet coefficients. By examining these two images, we can see that very little visual distortion is introduced by our anti-forensic algorithm. Additionally, the PSNR between the two images is 46.64dB. Fig. 3 shows the wavelet coefficient histograms corresponding to the fourth level *HH* subband of a four level wavelet decomposition of the Lena image before SPIHT compression, after compression, and after anti-forensic dither has been added to the compressed image. In this figure, we can clearly observe that histogram of anti-forensically modified wavelet coefficients is free from compression artifacts. This example demonstrates that after anti-forensic dither is added to an image's wavelet coefficients, the image can be passed off as having never undergone wavelet-based compression.

In addition to the example above, we performed a larger scale test to verify our proposed algorithm's ability to disguise previous applications of wavelet-based image compression. We built a

Fig. 2. Left: An image compressed using the SPIHT algorithm at a bit rate of 3 bits per pixel before the use of entropy coding. Right: The same image after anti-forensic dither has been applied to its wavelet coefficients.

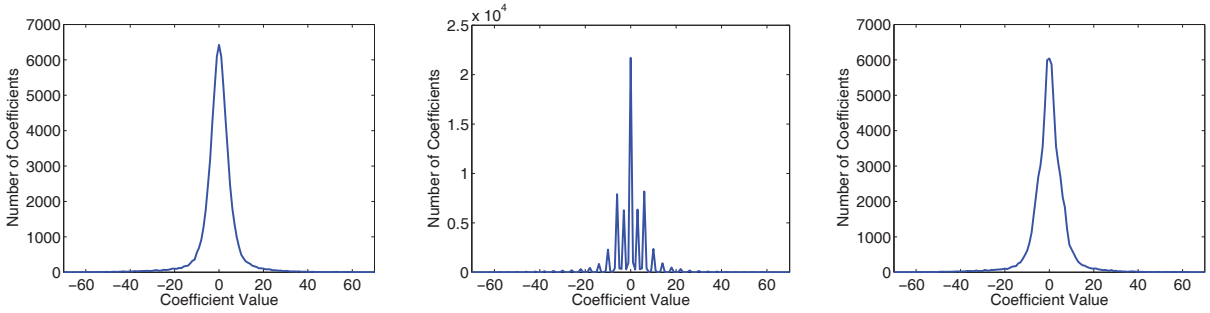

Fig. 3. Histogram of wavelet coefficients from the fourth level *HH* subband of a four level wavelet decomposition of the image shown in Fig. 2 (left), the same image after SPIHT compression (center), and the compressed image after anti-forensic dither has been applied (right).

database of test images by converting the 244 images within the Uncompressed Colour Image Database [9] to grayscale, compressing these images at a bit rate of 2 bits per pixel using the SPIHT algorithm, then applying anti-forensic dither to each compressed image's wavelet coefficients to obtain a set of anti-forensically modified images. We then used the algorithm proposed in [2] to test for evidence of SPIHT compression. This algorithm was trained using the sets of never-compressed and SPIHT compressed grayscale images, resulting in a classification rule able to correctly classify 99.6% of the SPIHT compressed images within the training set without misclassifying any of the never-compressed images. When this algorithm was used to classify the set of anti-forensically modified images, it classified every image in the testing set as never-compresed. This corresponds to a 100% success rate for our proposed anti-forensic algorithm.

5. CONCLUSIONS

In this paper, we have proposed an anti-forensic technique capable of removing the forensically significant artifacts left by wavelet-based image compression schemes. Our technique operates by adding anti-forensic dither to the wavelet coefficients of a compressed image so that the distribution of anti-forensically modified coefficients matches a model of the coefficients before compression. Simulation results show that this technique is able to fool forensic algorithms designed to detect previous applications of wavelet-based compression 100% of the time.

6. REFERENCES

[1] A.C. Popescu and H. Farid, "Statistical tools for digital forensics," in *6th International Workshop on Information Hiding*, Toronto, Canada, 2004.

[2] W. S. Lin, S. K. Tjoa, H. V. Zhao, and K. J. Ray Liu, "Digital image source coder forensics via intrinsic fingerprints," *IEEE Trans. Information Forensics and Security*, vol. 4, no. 3, pp. 460–475, Sept. 2009.

[3] J. Lukas, J. Fridrich, and M. Goljan, "Digital camera identification from sensor noise," *IEEE Trans. on Information Forensics and Security*, vol. 1, no. 2, pp. 205–214, 2006.

[4] M. C. Stamm, S. K. Tjoa, W. S. Lin, and K.J.R. Liu, "Anti-forensics of JPEG compression," in *Proc. ICASSP*, Dallas, Texas, USA, Mar. 2010.

[5] T. Gloe, M. Kirchner, A. Winkler, and R. Böhme, "Can we trust digital image forensics?," in *15th International Conference on Multimedia*, 2007, pp. 78–86.

[6] M. Kirchner and R. Böhme, "Synthesis of color filter array pattern in digital images," in *Proc. SPIE-IS&T Electronic Imaging: Media Forensics and Security*, Feb. 2009, vol. 7254.

[7] A. Said and W.A. Pearlman, "A new fast and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 6, pp. 243–250, June 1996.

[8] J. Li and R.M. Gray, "Text and picture segmentation by the distribution analysis of wavelet coefficients," in *Proc. ICIP*, Chicago, Illinois, USA, Oct. 1998, pp. 790 – 794.

[9] G. Schaefer and M. Stich, "UCID: an uncompressed color image database," in *Proc. SPIE: Storage and Retrieval Methods and Applications for Multimedia*, 2003, vol. 5307, pp. 472–480.