IMPROVED AUTOCORRELATION PEAKS FOR IMAGE WATERMARKING IN THE WAVELET TRANSFORM DOMAIN

Ray Naegle$^1$ and Xiaojun Qi$^2$

$^1$rnaegl01@shepherd.edu
Department of Computer Science, Mathematics, and Engineering
Shepherd University, Shepherdstown, WV 25443-3210

$^2$xqi@cc.usu.edu
Computer Science Department, Utah State University, Logan, UT 84332-4205

ABSTRACT

This paper presents improvements upon an autocorrelation function (ACF)-based watermarking scheme that is resistant to both removal and geometric attacks, proposed by Lee et al. [1], [2]. A host image is processed to give itself periodic autocorrelation (AC). A periodic watermark is then embedded in the perceptually significant blocks of a host image’s wavelet transform coefficients. Using this combination of techniques, we present results detailing improvements over the schemes proposed by Lee et al.

1. INTRODUCTION

In recent years, the Internet has become a staple of modern life. The increased penetration of broadband network access has made it easier for individuals to share information and communicate with one another. At the same time, this increased freedom to share data poses a threat to copyright holders, whose intellectual property can be shared illegally.

Lee et al. [1] describe an ACF-based watermarking scheme in the spatial domain which increases the autocorrelation peaks of a watermarked image. In their method, the cover image is first filtered to isolate the noise which would interfere with the embedded watermark signal.

In another publication, Lee and Lee [2] propose a similar watermarking procedure in the wavelet transform domain. The wavelet transform domain is used by modern image compression algorithms. This allows us to embed different periodic watermark signals with varying strengths, based on the wavelet transform decomposition level and sub-band, which results in increased resistance to removal attacks.

The work described in this paper is based heavily upon the findings of Lee et al. [1], [2]. The proposed method combines the strengths of both of the schemes described. By increasing the autocorrelation of the cover image before embedding the watermark, the resulting peak strength is higher. A watermark is then applied to the image, using properties of the wavelet transform domain to embed the signal with higher strength than in the spatial domain, while keeping visual distortion to a minimum.

By embedding a periodic watermark within an image, the image can later be processed to find it’s AC peaks. These peaks are key to reversing any geometric attack applied to the watermarked image. Intuitively, by increasing the strength of these peaks, the likelihood of a successful attack reversal should increase.

We do not address the reversal of geometric attacks, or the detection of the watermark pattern, as this is covered in great detail by Lee and Lee [2]; instead, we document how combining the techniques described by Lee et al. In both papers results in stronger AC peaks.

It should be noted that all equations used throughout this paper come from the work of Lee et al. [1],[2].

Figure 1. Example AC peaks
2. THE EMBEDDING SCHEME

Two steps are involved in the watermark embedding process. First, we modify the cover image before embedding to ensure a high autocorrelation after removal attacks. An ACF-based approach is limited by the strength of its autocorrelation peaks without them, the geometric attack cannot be reversed, and the watermark cannot be extracted. Second, we will embed the watermark in the wavelet domain of the modified cover image to maintain high invisibility and robustness to removal attacks. The details of these two steps are explained in the following subsections.

2.1 Cover image preprocessing

The watermark signal, a kind of additive noise, can be isolated from the cover image by taking the difference between the cover image and the Weiner filtered image. Most images contain noise to begin with, so we seek to minimize interference between such noise and out watermark. To do this, the noise is isolated from the cover image by

\[ E = I - fI \quad (1), \]

where \( I \) represents the cover image, and \( fI \) is the Weiner filtered image [1]. The resulting noise is segmented into blocks of size \( \frac{M}{2} \times \frac{M}{2} \), and the average block \( r(x, y) \) is calculated by

\[ r(x, y) = \frac{1}{N} \sum_{n=1}^{N} v_n(x, y) \quad (2), \]

where \( v_n(x, y) \) represents the segmented blocks of \( E \). The average block \( r \) is then up-scaled to have size \( M \times M \), to make the changes made here more resistant to removal attacks. For our experiments, we used \( M = 128 \).

To increase the autocorrelation of the image, the segmented blocks \( v_n \) are treated as vectors. For each \( v_n \), the up-scaled average block \( R \) is treated as a vector, and modified to have the same length as each \( v_n \), as shown by

\[ R_n = \frac{|v_n| \cdot R}{|R|} \quad (3). \]

The resulting reference vector \( R_n \) is used to calculate the difference vector \( d_n \) for each \( v_n \) as

\[ d_n = R_n - v_n \quad (4). \]

Finally, each block \( v_n \) is modified to be more like \( R_n \) by the following formula:

\[ v'_n(x, y) = v_n(x, y) + \alpha_d \lambda_{dn}(x, y) d_n(x, y) \quad (5), \]

where \( \alpha_d \) is a user-defined weighting factor, and \( \lambda_{dn} \) is a local weighting factor calculated using the Noise Visibility Function (NVF) to control the strength with which the autocorrelated noise \( v'_n \) is mixed with the extracted noise \( v_n \). The local weighting factor \( \lambda_{dn} \) is a sub-block of \( \lambda_d \) that corresponds to \( v_n \). The following formula illustrates that \( \lambda_d \) is defined for the whole image by

\[ \lambda_d = (1 - NVF) \cdot S + NVF \cdot S_1 \quad (6). \]

In (7), \( S \) and \( S_1 \) control the weight given to the autocorrelated noise based on the textured and smooth areas of the cover image. The values \( S = 3 \) and \( S_1 = 1 \) were used as Lee et al. [1] chose. The values for NVF are defined by

\[ NVF(x, y) = \frac{1}{1 + \frac{\sigma_2(x, y)}{\sigma_{max}^2}} \quad (7), \]

where \( \sigma_2(x, y) \) is the local variance of the cover image, \( \sigma_{max}^2 \) is the maximum of the local variance, and \( D \in [50, 100] \).

After \( v'_n \) has been calculated for all \( n \), the segments \( v'_n \) together form the autocorrelated noise \( E' \), which is added back to the filtered image \( fI \), given by

\[ I'(x, y) = fI(x, y) + E'(x, y) \quad (8). \]

2.2 Watermark embedding algorithm

Once the autocorrelated image \( I' \) has been generated, it’s 2-level wavelet transformation is taken. The sub-bands are represented by \( I_j^0 \), which represents the \( \lambda \)-direction sub-band in the \( j \text{th} \) level of the wavelet transformation of \( I' \). The directions are represented by \( \lambda \) as follows:

\[ \theta = 1: \text{horizontal}, 2: \text{diagonal}, 3: \text{vertical}. \]

Two watermark patterns are generated. The watermarks should be of size \( \frac{M}{2} \times \frac{M}{2} \) for the first level sub-bands, and \( \frac{M}{4} \times \frac{M}{4} \) for the second level. The watermarks should be fractional numbers between \([-1, 1]\), and generated by a user key so that the watermark can be generated again for detection.

The watermarks are embedded in the image as shown by

\[ I_j^0 = I_j^0 + \alpha \lambda_j^0(x, y) W_j(x, y) \quad (9), \]

where \( \alpha \) is a global weight, and \( \lambda \) is a local weight. During the embedding step, \( \lambda \) is calculated in much the same way as the noise correlation step, but it is calculated for each sub-band of the wavelet transform. The formula for \( \lambda \) is given as

\[ \lambda_j^0(x, y) = L_j \Theta^\theta[(1 - NVF_j^0(x, y)) \cdot S + NVF_j^0(x, y)] \quad (10), \]
where $\Theta^\theta$ and $L_j$ are weighting factors that vary depending on the sub-band and wavelet transform decomposition level. The weighting factor $\Theta^\theta$ is set to 2.5 when $\theta = 2$, otherwise it is set to 1. This gives additional weight to the watermark when being embedded in the diagonal sub-bands of the cover image, as noise is more difficult for humans to perceive in the diagonal sub-bands\[2\]. $L_j$ is a user-defined weighting factor that specifies the weight given to the watermark signal with respect to the wavelet transform decomposition level. In this experiment, we chose $L_1 = 1$, $L_2 = 2.5$. The watermark is embedded less aggressively in the first-level sub-bands because the watermark is more easily to see when embedded in the first-level sub-bands. More weight is given to the watermark in the second-level sub-bands because they are more likely to survive a geometric or removal attack because of their increased low frequency. As before, $S$ and $S_1$ are once again user-defined weights for the textured and smoothed regions respectively. $S$ is set to 5, and $S_1$ is set to 1. This insures that the watermark is embedded more aggressively in areas with more texture.

The values for the NVF are calculated for each sub-band of each decomposition level as given by

$$NVF^\theta_j(x, y) = \frac{1}{1 + \frac{\sigma^2_j(x, y)}{\sigma^2_{j_{\max}}}}$$  (11),

where $\sigma^2_j$ is the local variance of the given sub-band and decomposition level, and $\sigma^2_{j_{\max}}$ is the maximum of $\sigma^2_j$.

### 3. PEAK DETECTION

It is well known that a change to the wavelet transform of an image will be localized to the corresponding area of the image in the spatial domain, so it makes sense that the watermark signal’s periodicity can be extracted from the spatial domain. To do this, the Wiener filter is applied to the marked image, and the resulting filtered image $fI$ is subtracted from the marked image $I$ as given in (1). The resulting signal $E$ contains the periodicity of the watermark.

Since the periodicity of the watermark can be extracted from the spatial domain, any geometric transformation on the image will be visible in the orientation of the autocorrelation peaks of the marked image.

In order to find the autocorrelation peaks, a FFT-based correlation method is used, given by

$$ACF = \frac{\text{IFFT}(\text{FFT}(E) \cdot \text{CONJ} (\text{FFT}(E)))}{|E|^2}$$  (12),

where the operation $\text{CONJ}(x)$ indicates taking the complex conjugate of the operand. The resulting signal $ACF$ will show periodic peaks which can be used to estimate and reverse a geometric attack. As we are only addressing the relative peak strength between several embedding methods, the geometric attack estimation and reversal methods described by Lee et al. [1], [2] are not covered in this paper.

---

4. EXPERIMENTAL RESULTS

For testing, we implemented the ACF-based watermarking scheme in the spatial domain proposed by Lee et al. [1], as well as the scheme in the wavelet transform domain proposed by Lee and Lee [2]. We also implemented the scheme described in this paper. A total of 100 $512 \times 512$ pixel gray scale images were watermarked. Their AC peaks were then found and compared to each other in terms of peak strength.

Figure 2. Extracted AC peaks from a watermarked image

Figure 3. Example images used for the experiment

In order to best illustrate the increased strength of the AC peaks of the watermarked images, we converted each row and column containing peak information into vectors, and found the mean value. We excluded the row and column containing the center peak, which always has a 100% strength value. Figure 4 illustrates the rows and columns used for calculating the mean peak strength. We also shifted the mean vectors to the right or left for ease in reading the AC peaks in figure 5.
We found that combined approach yielded AC peaks with an average strength of 25%. The method proposed by Lee et al. in the spatial domain [1] gave AC peaks with an average strength of 16.2%, and the method proposed by Lee and Lee in the wavelet transform domain [2] gave AC peaks with an average strength of 22.9%.

Future work will determine whether this increased peak strength improves the likelihood of a geometric attack being accurately reversed, and if so, to what degree.

5. ACKNOWLEDGEMENTS

This work is funded in part by grants from the National Science Foundation REU program, as well as the NASA Space Grant Consortium. Ray would like to thank Drs. Xiojun Qi, Reza Mirdamadi, Weidong Liao, Zhujin Wang, and Rajeev Rajaram for their support and guidance.

References
