A Block Size Analysis for Blocked Discrete Wavelet Watermarking
Ahmet Şenol¹, Ersin Elbaşi², Kıvanç Dinçer¹, Hayri Sever¹

Abstract—Image watermarking has been an important discipline for claiming ownership, ensuring genuineness of images and fingerprinting of images. An important part of the work in this field is performed in discrete wavelet transform (DWT) domain. The watermark is embedded in coefficient values in one or all of LL, LH, HL, HH bands of first level or higher level (DWT) decompositions. In previous works, a host image is taken as a whole and firstly DWT of the whole image is found and then further level decompositions are obtained, if needed. In this paper, instead of taking DWT of the whole image, we implement the known DWT watermarking algorithm by first dividing host image into smaller blocks, namely 64x64, 32x32, 16x16, 8x8, and then taking DWT of each block independently. In addition, we divide the corresponding watermark into the same/matching number of blocks as the original image blocks. We then embed the corresponding watermark block into bands (LL, LH, HL, HH) of DWT transform of original image blocks. When we analyze the fidelity of watermarked images and similarity values of extracted watermarks for different types of block sizes and attacks, we observed that we get better results in our implementation than the known algorithm for some attack types and the results improve with smaller size blocks.

Index Terms—Block Size, Discrete Wavelet Transform, Image Watermarking, Robust Non-blind Watermarking.

I. INTRODUCTION

Millions of files are downloaded and uploaded every day on the Internet. Majority of the files exchanged on the Internet are image and video files. The ease of access to digital content on the Internet has increased all sorts of misuse including violating copyrights and tampering data without owner’s permission [1]. Content owners are continuously seeking new ways for protecting their copyrighted files.

One of the main methods of protecting copyrighted content is watermarking. Image watermarking is embedding some data, called “watermark”, into the image that does not affect the quality of the image in visually detectable extent, and the watermark is extracted when necessary.

Although a visible image watermark such as a logo can declare the identity of its producer, most watermarks in use are invisible. Image watermarking is commonly used for proving ownership, keeping track of advertisement broadcasts, putting metadata right into the image for indexing/searching and making sure that the image is not changed or disturbed (tamper detection).

There is the term “fragile watermarking”, which is used for proving the image is genuine. The watermark is embedded in such a way that the watermark is lost when the watermarked image is changed to some extent.

A successful watermark algorithm must satisfy following conditions [2]:

Fidelity: Fidelity is a criterion that the watermarked image looks like the original image, i.e. embedded watermark is not perceptible in the watermarked image. You see the original image and watermarked image in Fig.1 and Fig.2, respectively.

Robustness: Robustness is the extent of durability of the watermark embedded in the original image after intentional or unintentional image processing attacks such as compression, cropping, noise, blur or filter. If the purpose is tamper detection, a fragile watermark is needed; i.e., with a slightest change in the watermarked image, watermark must deteriorate. Robustness is directly dependent on the watermarking method and domain. You see a binary image (visual) watermark in Fig.3 to be embedded in the image, and in Fig.4 you see the watermark extracted from a watermarked and attacked image. As seen, the watermark is easily recognizable regardless of the attack.

Fig. 1. Original Image
Fig. 2. Watermarked Image
Fig. 3. Binary Image
Watermark embedded
Fig. 4. Watermark extracted from watermarked and attacked document
II. PREVIOUS WORK IN LITERATURE

Image watermarking began to develop in the spatial domain. Tirkel, Schyndel, and Osborne embedded watermarks of m-sequence to the least significant bit of a 8 bit gray scale image[4]. That spatial domain algorithm had a data load capacity of one bit; that is: the watermark is present or not.

Frequency domain or transform domain methods emerged to replace LSB based methods in the following studies. In 1997 Cox, Kilian, Leighton, and Shamsun [7] used the method of “spread spectrum communication” in which frequency band is very large, the message to be transmitted is spread along the frequency scale such that the signal is very weak in each frequency so that it could not be detected by hostile attacks. They made the analogy with the frequency spectra with the image transformed to frequency domain by DCT, and analogy with watermark and transmitted signal. To place a watermark of length n to an NxN image, firstly NxN DCT of the image is computed and the watermark is put in n highest magnitude coefficients of the transform matrix except the DC (lowest) component. Watermark embedding and detection processes are shown in Fig.5 and Fig.6.

Piva, Barni, Bartolini and Capellini used a similar method [8]. They also embedded the watermark in DCT domain where watermark is a pseudo random sequence of real numbers (PSRN) having mean 0 and variance 1. They differ from work of Cox et al. in that, instead of embedding the watermark in first 1000 DCT coefficients, they skip the first L coefficients and embed the watermark in the following M coefficients. The algorithm is semi-blind so the original image is not used in watermark detection phase. The authors skip the first L coefficients for the watermark to be not affected by high-pass filters such as histogram equalization, gamma correction. In the paper, they used L=25000, M=16000, where M is also the watermark size. The watermark is embedded in middle frequencies so that algorithm is more robust to low-pass and high-pass filters.

Correlation between the original and extracted watermark is computed as:

\[ Z: \text{Correlation} = \frac{(W * V^*)}{M} = \frac{1}{M} \sum_{i=1}^{M} W_i V_i^* \quad (3) \]
where $W$ is original watermark, $V^*$ is the extracted watermark and $M$ is the watermark size.

If $Z > T_z$, where $T_z$ is correlation threshold value, then it is decided that watermark is present. $T_z$ is calculated as:

$$T_z = \frac{\alpha}{2M} \sum_{i=1}^{M} |V^*_i|,$$  \hspace{1cm} (4)  

where $\alpha = 0.2$ in the paper.

Nikolaidis & Pitas claim that correlation detector is not optimal since data distributions of transform coefficients do not have Gaussian probability density function(pdf); instead, generalized Gaussian (GG) pdf is observed in DCT/DWT transform values[9]. They proposed a new watermark detection scheme that uses the parameter values of this pdf and pdf's parameters which are calculated from watermarked data itself. Rahman, Ahmad & Swamy propose to use pdf based on Gauss-Hermit expansion in detection of watermarks embedded in DWT sub band coefficients [10]. They claim that their detection method gives better results because this new pdf statistically matches better to the exact pdf since number of parameters calculated from higher order moments of image coefficients is more than GG or other type of pdf’s.

Tao and Eskicioğlu proposed a non-blind Discrete Wavelet Transform (DWT) based watermarking method[6]. In discrete wavelet transform, image is decomposed into four bands, each of which is half sized in both row and column size, which makes it a quarter of original size. Upper left band is called LL band, in which a coarser version of the original image is placed. The other bands are: on upper right band HL (horizontal detail), on lower left band LH (vertical detail), and lower right band LL (diagonal detail) high pass bands. The LL band in decomposition can be further decomposed to obtain another four bands of a lower level.

A binary image logo watermark is embedded in each four of the decomposition levels of the original image. After applying DWT transform to the image, in each of the bands (LL, LH, HL, HH), watermark embedding is done as in Eq.(5) and extraction is done as in Eq(6).

I: original 2Nx2N image  
W: Nxn binary logo watermark image  
$V_{ij}$: DWT coefficients in the corresponding band  
$V^*_{ij}$: watermarked DWT coefficients  
$V^{**}_{ij}$: watermarked and possibly attacked images DWT coefficients in band $k$, where $k \in \{1,2,3,4\}$.  
$W^*$: extracted binary watermark image

$$V^k_{ij} = V_{ij} + \alpha W_{ij},$$ \hspace{1cm} (5)  

$$W^*_{ij} = (V^{**}_{ij} - V^k_{ij}) / \alpha_k,$$ \hspace{1cm} (6)

Since magnitudes of DWT coefficients are larger in LL band, watermark strength constant $\alpha$ is a larger value than those of LH, HL, HH bands.

Visual quality of the watermarked and attacked images is measured using Peak-Signal-to-Noise-Ratio (PSNR) values.

$$PSNR = 20 \log_{10}(255/RMSE)$$ \hspace{1cm} (7)

RMSE is square root of mean squared error between original and distorted images

$$RMSE = \sqrt{\frac{\sum_{i,j}(I_{ij} - I'_{ij})^2}{(N x N)}}$$ \hspace{1cm} (8)

Many DWT-based watermarking schemes were developed thereafter[11]–[14]. Dugad, Ratakonda and Narendra proposed [11] a method similar to that of Tao and Eskicioğlu’s work[6]. Proposed work was also semi-blind DWT-based watermarking method, in which watermark is a binary image of the same size as the original image, watermark is embedded in LH, HL and HH bands. The watermark is embedded in only the coefficients higher than a given threshold $T_1$. Adding watermark only to higher-level coefficients means embedding watermark in only the edges in an image. In the watermark detection phase, coefficients that are above a second threshold $T_2$, where $T_2$ is far above $T_1$, are chosen. Also while calculating the correlation threshold that is used to decide whether watermark is present or not, value is divided by $2M$ instead of $3M$ of Piva et al.’s work[8]. Authors state that since higher level coefficients are preserved despite much degradation, correlation value will be higher than that of [8]. This could give their method more fidelity by increasing deciding threshold value.

$$T_z = \frac{\alpha}{2M} \sum_{i=1}^{M} |V^*_i|,$$ \hspace{1cm} (9)

Lai & Tsai embedded watermark into the singular values of the singular value decomposition (SVD) that is applied to the DWT bands of the original image[15].

Proposed Work: The proposed work pursues the method of Tao and Eskicioğlu [6] that it is a DWT-based non-blind algorithm that embeds the watermark in all bands (LL, LH, HL, HH) of DWT transform. The difference is that, instead of taking DWT of whole image and embedding the whole watermark in first level decomposition bands, the original image is split into 64x64 blocks, and DWT of each block is taken separately. Watermark binary image is also split into 32x32 blocks, and this 32x32 part of watermark is embedded in first level DWT decomposition bands of 64x64 original image block. In each image block, a watermark block that is half size both in length and width of image block is embedded in first level decomposition LL, LH, HL, HH bands.

Similarly, the algorithm is run with the original image split into 32x32, 16x16, 8x8 blocks, obtained watermarked images and their PSNR values, also saved the extracted watermarks and their quality ratios.
PSNR values between the original image and watermarked image are calculated to compute the fidelity performance of the algorithm which measures to what extent the watermarked image looks like the original one. PSNR is described in Eq (7).

\[
SR = \frac{S}{S + D} \quad (10)
\]

S: number of matching pixels in compared images
D: number of different pixels in compared images

Watermark Embedding and Extraction Algorithm for 64x64 image blocks is given below. In the same way, algorithm can be run for 32x32, 16x16, 8x8 sized blocks by changing block size values.

Watermark Embedding Algorithm:

1. Divide Original Image in 64x64 equal parts, Ipo
2. Divide watermark binary image W into 32x32 parts Wp
3. For each 64x64 part of original image block Ipo
   a. (Ipt_LL, Ipt_LH, Ipt_HL, Ipt_HH) = DWT(Ipo), transform
   b. Embed 32x32 corresponding watermark part Wp into each of Ipt_LL, Ipt_LH, Ipt_HL, Ipt_HH
      i) \( Ipt_{LL}'_{ij} = Ipt_{LL}_{ij} + \alpha Wp_{ij} \) where \( \alpha = 8 \) for LL band, \( \alpha = 2 \) for other bands
   c. Ipw = IDWT(Ipt_LL', Ipt_LH', Ipt_HL', Ipt_HH') , Obtain watermarked image part Ipw
   d. Combine watermarked image blocks Ipw's to obtain watermarked image Iw

Watermark Extraction Algorithm:

1. Divide watermarked and possibly attacked Image in 64x64 equal parts, Ip*
2. Divide original Image in 64x64 equal parts, Ipo
   (Ipt_LL, Ipt_LH, Ipt_HL, Ipt_HH) = DWT(Ipo)
4. For each 64x64 block of watermarked and original image block Ip*, Ipo
   Wp_LL* = (Ipt_LL*- Ipo_LL)/coll; where Wp_LL*: Extracted Watermark part values in LL band, coll=8 for LL band.
   Wp_LH* = (Ipt_LH*- Ipo_LH)/coll; Ipt_LH*, Ipt_HL*, Ipt_HH' ) , Obtain watermarked image part Ipw
   Wp_HH* = (Ipt_HH*- Ipo_HH)/coll; Where Wp_HH*: Extracted Watermark part as binary image in HH band
   Wp_LHb*= Wp_LH* > 0.5;
   Wp_HLb*= Wp_HL* > 0.5;
   Wp_HHb*= Wp_HH* > 0.5;
5. Combine extracted watermark parts Wp*'s to obtain extracted watermark W* for each band LL, LH, HL, HH.

Since DWT coefficient values are larger in LL bands than in other three bands, watermark strength \( \alpha \) is 8 for LL band, 2 for the other three bands.

III. EXPERIMENTS

The algorithm was run by dividing the image into 512x512 (whole image), 64x64, 32x32, 16x16, 8x8 image blocks, taking DWT of each block separately and embedding the watermark part into each of the first level DWT decomposition bands LL, LH, HL, HH. Watermarked images were saved for each case. Then most common attacks were applied to watermarked images. Those attacks were: Jpeg compression by %75, %50, %25 image quality, 3x3 blur filter, Gaussian noise attack with mean zero and variance 0.001, scale-rescale by 0.5 and 2, histogram equalization, intensity adjustment from range \([0 0.8]\) to \([0 1]\), gamma correction, re-watermarking. The original image used is seen in Fig.1. The binary watermark logo image is an image that has letters BC next to each other as seen in Fig.3. The second watermark embedded for attacking is seen in Fig.7. Watermarked and attacked images were also saved. The watermarked and attacked images and their corresponding PSNR values are seen in Fig.8 for the case when original image is divided into 8x8 image blocks. Only the watermarked images that are subjected to compression attacks are given.

The watermark extraction phase is run for each image block size case. The extracted watermarks and their similarity ratios are given in Fig.9. Again, only watermarks extracted from LL and LH bands for compression, rotate, scale, crop and re-watermark attacks are given. For each type of attacked image, extracted watermark from first DWT decomposition band of that attacked image is listed in the same column, as rows correspond to 64x64, 32x32, 16x16, 8x8 block size cases as seen in Fig.9. The extracted watermark’s similarity ratio (extracted watermark quality) increases as the image is divided into smaller size blocks before embedding the watermark. The coarser watermarks are extracted when the
image is divided into 64x64 parts, and the best results are obtained when the image is divided into 8x8 parts. The similarity ratios for each type of attack and for each size of blocks can be seen in the tabular form in Table I. Tao & Eskicioglu results are given as columns in Table I next to 8x8 block size result column. In this work we applied the same algorithm where Tao applied using same cover image and watermark. Experimental results show that our blocked watermarking method gives better results in embedding and in extraction phases after some common attacks than Tao algorithm. We do not compare directly with Tao’s numerical results because we do not know some parameters such as scaling factor.

When we look at the algorithm in image fidelity aspect, although algorithm applied with smaller blocks gives much better extracted watermarks, it has no drawbacks compared to ones with larger blocks in fidelity aspect. PSNR values are all the same for every different block size. The only type of attack that PSNR values differ is the Gaussian Noise attack. This is because every time Gaussian attack is called, a noise that is formed by completely different values is applied on the watermarked image; so different PSNR values are reasonable.

We reached better results when we increased watermark strength $\alpha =12$ for LL band, $\alpha =5$ for LH, HL, HH bands, but inserted watermark began to be noticeable and image began to deteriorate. Also instead of comparing extracted watermark values with 0.5 when transforming it to binary image, we can compare with 0.45 or 0.4 to get a better-extracted image.

We must also mention a drawback that although the best results are obtained with 8x8 block size, only one level of DWT decomposition is possible for this block size. The second drawback is the extra CPU time needed as smaller block sizes are chosen. CPU time vs. block size graphs are given in fig.10 for embedding and extraction phases. For real time systems, execution time may be a problem but it could be tolerable in other type of systems.

IV. CONCLUSION

In Tao paper [6], binary pattern has been embedded in four bands of Discrete Wavelet Transform (LL, LH, HL and HH). Test results show that embedding the watermark in lower frequencies is robust to a group of attacks, and embedding the watermark in higher frequencies is robust to another set of attacks. Only for re-watermarking and collusion attacks, the watermarks extracted from all four bands are identical.

In this research work instead of using Tao’s algorithm, we divided cover image into 512x512, 64x64, 32x32, 16x16 and 8x8 pixel blocks. We embedded binary watermark into each block using first level decomposition of DWT. Experimental results show that our method blocked watermarking gives better results in embedding and in extraction phases after some common attacks than Tao’s algorithm. The drawback of using smaller blocks is that computational overhead increases compared to using larger image sub blocks. We conclude that, blocked watermarking can be used successfully in cases where one can tolerate computational overhead.

REFERENCES

Fig. 8. Original, Watermarked and Attacked Images for 8x8 image block sizes and embedded watermark.

Fig. 9. Extracted watermarks from the bands LL, LH, HH for the cases images divided into 64x64, 32x32, 16x16, 8x8 image blocks.

Table I. Similarity ratios for extracted watermarks from LL and LH bands for attack types.

<table>
<thead>
<tr>
<th>Attack Type</th>
<th>512x512</th>
<th>64x64</th>
<th>32x32</th>
<th>16x16</th>
<th>8x8</th>
<th>Tao&amp;Esk</th>
</tr>
</thead>
<tbody>
<tr>
<td>JPEG 75</td>
<td>0.898</td>
<td>0.898</td>
<td>0.905</td>
<td>0.912</td>
<td>0.916</td>
<td>0.92</td>
</tr>
<tr>
<td>JPEG 50</td>
<td>0.803</td>
<td>0.803</td>
<td>0.817</td>
<td>0.83</td>
<td>0.838</td>
<td>0.84</td>
</tr>
<tr>
<td>JPEG 25</td>
<td>0.713</td>
<td>0.713</td>
<td>0.734</td>
<td>0.754</td>
<td>0.764</td>
<td>0.747</td>
</tr>
<tr>
<td>FILTER</td>
<td>0.772</td>
<td>0.772</td>
<td>0.793</td>
<td>0.805</td>
<td>0.815</td>
<td>0.822</td>
</tr>
<tr>
<td>GAUSS</td>
<td>0.685</td>
<td>0.688</td>
<td>0.709</td>
<td>0.727</td>
<td>0.737</td>
<td>0.717</td>
</tr>
<tr>
<td>SCALE-RESCALE</td>
<td>0.75</td>
<td>0.75</td>
<td>0.773</td>
<td>0.795</td>
<td>0.807</td>
<td>0.7795</td>
</tr>
<tr>
<td>HISTOGRAM EQ.</td>
<td>0.627</td>
<td>0.627</td>
<td>0.664</td>
<td>0.713</td>
<td>0.713</td>
<td>0.438</td>
</tr>
<tr>
<td>INTENSITY ADJ</td>
<td>0.801</td>
<td>0.801</td>
<td>0.867</td>
<td>0.932</td>
<td>0.966</td>
<td>0.197</td>
</tr>
<tr>
<td>GAMMA</td>
<td>0.199</td>
<td>0.199</td>
<td>0.199</td>
<td>0.199</td>
<td>0.199</td>
<td>0.803</td>
</tr>
<tr>
<td>ROTATE-REROTATE</td>
<td>0.766</td>
<td>0.766</td>
<td>0.776</td>
<td>0.784</td>
<td>0.789</td>
<td>0.91</td>
</tr>
<tr>
<td>CROP</td>
<td>0.586</td>
<td>0.586</td>
<td>0.586</td>
<td>0.586</td>
<td>0.586</td>
<td>0.996</td>
</tr>
<tr>
<td>REWATERMARK</td>
<td>0.864</td>
<td>0.864</td>
<td>0.915</td>
<td>0.952</td>
<td>0.977</td>
<td>0.905</td>
</tr>
<tr>
<td>TOTAL</td>
<td>7.878</td>
<td>7.881</td>
<td>8.149</td>
<td>8.39</td>
<td>8.526</td>
<td>8.0655</td>
</tr>
</tbody>
</table>

Fig. 9. Continued