

# A Robust Digital Image-in-Image Watermarking Algorithm Using the Fast Hadamard Transform

Anthony T.S.Ho, Jun Shen, Soon Hie Tan  
School of Electrical and Electronic Engineering, Nanyang Technological University

## ABSTRACT

In this paper, we propose a robust image-in-image watermarking algorithm based on the fast Hadamard transform (FHT) for the copyright protection of digital images. Most current research makes use of a normally distributed random vector as a watermark and where the watermark can only be detected by cross-correlating the received coefficients with the watermark generated by secret key and then comparing an experimental threshold value. However, the FHT image-in-image method involves a “blind” watermarking process that retrieves the watermark without the need for an original image present.

In the proposed approach, a number of pseudorandom selected  $8 \times 8$  sub-blocks of original image and a watermark image are decomposed into Hadamard coefficients. To increase the invisibility of the watermark, a visual model based on original image characteristics, such as edges and textures are incorporated to determine the watermarking strength factor. All the AC Hadamard coefficients of watermark image is scaled by the watermarking strength factor and inserted into several middle and high frequency AC components of the Hadamard coefficients from the sub-blocks of original image. To further increase the reliability of the watermarking against the common geometric distortions, such as rotation and scaling, a post-processing technique is proposed. Understanding the type of distortion provides a mean to apply a reversal of the attack on the watermarked image, enabling the restoration to the synchronization of the embedding positions.

The performance of the proposed algorithm is evaluated using Stirmark. The experiment uses container image of size  $512 \times 512 \times 8$ bits and the watermark image of size  $64 \times 64 \times 8$ bits. It survives about 60% of all Stirmark attacks. The simplicity of Hadamard transform offers a significant advantage in shorter processing time and ease of hardware implementation than the commonly used DCT and DWT techniques.

**Keywords:** Fast Hadamard Transform, Copyright Protection, Digital Watermarking

## 1. INTRODUCTION

With the advent of the Internet, the online purchasing and distribution of digital images can now be performed relatively easily. Over the past few years, the technology of digital watermarking has emerged as a leading candidate that could solve the problems of legal ownership and content authentications for digital multimedia data.

A great deal of research efforts has been focused on digital image watermarking in recent years. The techniques proposed so far can be divided into two main groups. One is the spatial domain approach. The simplest example is to embed a watermark into the least significant bits (LSBs) of the image pixels [1]. The other is the frequency domain approach. Cox et al. [2] used the spread spectrum communication for digital multimedia watermarking. They embedded a Gaussian distributed sequence into the perceptually most significant frequency components of container image. Hsu and Wu [3] embedded an image watermark into the selectively modified middle frequency of discrete cosine transform (DCT) coefficients of the container image.

The major problem with many watermarking schemes is that they are not very robust against different types of image manipulations or attacks such as the ones found in Stirmark. Moreover, some of these techniques are quite complicated to implement in real-time. In this paper, we propose a fast Hadamard transform (FHT) based watermarking approach. Grayscale image can be used as watermark, which is inserted into Hadamard coefficients of sub-blocks of the original container image.

This paper is organized as follows: the forward and reverse transformation of FHT and the choice of FHT domain is described in Section 2. In Section 3, image embedded watermarking algorithms are discussed. Section 4 discusses the watermark strength factor determination and post-processing techniques. Experiment results under Stirmark attacks on a digital image and the relevant discussions are presented in Section 5. Finally, the conclusion is given in Section 6.

## 2. 2D-HADAMARD TRANSFORM OF IMAGE

The 2D-Hadamard transform has been used extensively in image processing and image compression [4,5]. In this section, we give a brief review of the Hadamard transform representation of image data, which is used in the watermarking embedding and extraction process. The reason for choosing FHT domain is also discussed.

Let  $[U]$  represents the original image and  $[V]$  the transformed image, the 2D-Hadamard transform is given by:

$$[V] = \frac{H_n [U] H_n}{N} \quad (1)$$

where  $H_n$  represents an  $N \times N$  Hadamard matrix,  $N=2^n$ ,  $n=1,2,3,\dots$ , with element values either  $+1$  or  $-1$ . The advantages of Hadamard transform are that the elements of the transform matrix  $H_n$  are simple: they are binary, real numbers and the rows or columns of  $H_n$  are orthogonal. Hence the Hadamard transform matrix has the following property:

$$H_n = H_n^* = H^T = H^{-1} \quad (2)$$

Since  $H_n$  has  $N$  orthogonal rows  $H_n H_n = NI$  ( $I$  is the identity matrix) and  $H_n H_n = NH_n H_n^{-1}$ , thus  $H_n^{-1} = H_n/N$ . The inverse 2D-fast Hadamard transform (IFHT) is given as

$$[U] = H_n^{-1} [V] H_n^* = \frac{H_n [V] H_n}{N} \quad (3)$$

In our watermarking algorithm, the forward and reverse Hadamard transform is applied to the sub-blocks of the original or watermarked images. A two-dimensional FHT of the segmented  $8 \times 8$  blocks is performed by applying a one-dimensional FHT on the rows first and then followed by a 1-D FHT on the columns. In the computer implementation, the  $8 \times 8$  sub-block is first multiplied by Hadamard matrix, and then the resultant matrix is transposed and multiplied by the Hadamard matrix again. The final matrix is the transformed matrix.

For  $N=2$ , the Hadamard matrix,  $H_1$ , is called a core matrix, which is defined as

$$H_1 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad (4)$$

The Hadamard matrix of the order  $n$  is generated in terms of Hadamard matrix of order  $n-1$  using Kronecker product,  $\otimes$ , as

$$H_n = H_{n-1} \otimes H_1 \quad (5)$$

or

$$H_n = \begin{bmatrix} H_{n-1} & H_{n-1} \\ H_{n-1} & -H_{n-1} \end{bmatrix} \quad (6)$$

Since in our algorithm, the processing is carried out based on the  $8 \times 8$  sub-blocks of the whole image, the third order Hadamard transform matrix  $H_3$  is used. By applying (5) or (6),  $H_3$  becomes:

$$H_3 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix} \quad (7)$$

The characteristic of the rows or columns of  $H_3$  is sign transitions, which is defined as a 1 to -1 or -1 to 1 change. In  $H_3$ , the number of transitions for row 1 to row 8 is 0, 7, 3, 4, 1, 6, 2 and 5 according to equation (7). The number of sign changes is referred to as sequency [4]. The concept of sequency is analogous to its Fourier counterpart frequency. Zero sign transitions correspond to DC and a large number of sign transitions correspond to high frequency. In Hadamard matrix  $H_3$ , the elements are not arranged in increasing sequency, such that the number of transitions is 0, 1, 2, 3, 4, 5, 6 and 7. If the order of rows is in ascending of sequency, this transform matrix is called a Walsh transform matrix. A Walsh transform may cause the transformed matrix to have DC value at upper left corner and AC coefficients arrange in Zigzag order from low frequency components to high frequency components. The Walsh transform is not used here because the middle and high frequency AC components, which could be watermarked has shown to be somewhat unreliable and the performance worse than existing DCT watermarking algorithms. On the contrary, the Hadamard transform matrix  $H_3$  has its AC components in Hadamard order. It is possible that in the watermarking process, some of the watermark information can be embedded into the low frequency AC components. This increases the mark reliability and makes it more difficult to attack and remove.

Moreover, Hadamard transform has more useful middle and high frequency bands available, for hiding the watermark, as compared to other high coding gain transforms like DCT and DWT, at high noise environment. It has been shown that transforms including DCT and sub-band transforms are suitable for watermarking when the channel noise is low [6,7]. But low channel noise condition is not usually the case. For low quality JPEG or wavelet compression as well as some linear or nonlinear filtering such as Gaussian filtering and median filtering, the processing noise approaches infinity for the middle and high frequency bands. In these cases, the high gain transform watermarking methods are not very robust. But middle and high frequency Hadamard transform coefficients have components equivalent to where many DCT low-frequency AC coefficients are located. So it is more likely that in high noise environment the Hadamard transform bands would remain and unscathed and immune to the channel noise.

Furthermore, low quality JPEG and wavelet compression could normally affect the high frequency components of DCT or DWT based watermarked image than the high frequency components in the Hadamard bands. Watermark information embedded in these components of DCT and DWT may be removed by these compressions. However, Hadamard transform would survive the compression as some of the watermark energy is embedded into the equivalent low frequency components of DCT and DWT domain. Therefore, decompositions that are not so efficient for compression, such as the Hadamard transform, would provide more immunity to channel noise than decompositions with high coding gain transform [7].

Another advantage of using the FHT is that it has a shorter processing time and its ease of hardware implementation than many commonly used orthogonal transforms. An efficient FHT hardware structure has been reported [8]. Based on the radix-2 algorithm, the structure can be implemented with shift register and controllable adder/subtractors for high processing throughput. This makes it possible for the real-time implementation of a digital watermarking system.

### 3. WATERMARKING IN FHT DOMAIN

In our technique, the original image information is not necessary at the watermark identification stage. This refers to a "blind" watermarking process [9]. The block diagram of the watermarking system is shown in Figure 1:

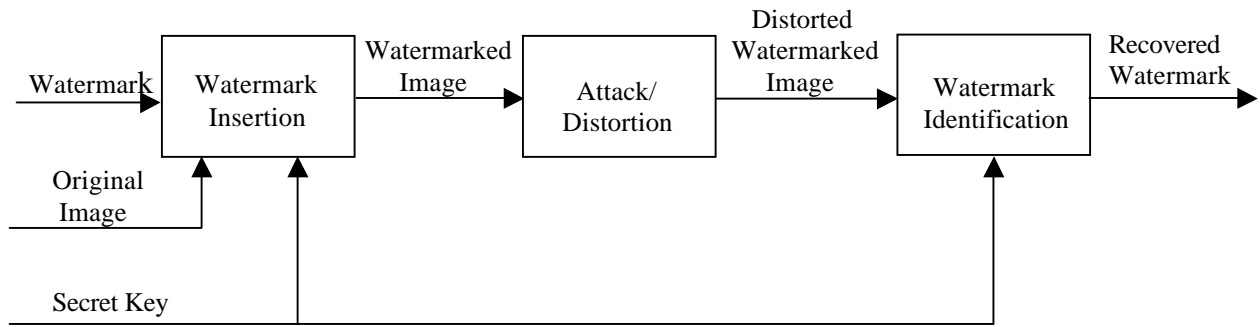


Figure 1: Block diagram of “blind” watermarking system

The proposed watermarking algorithm hides visually recognizable patterns in the container image. In our experiment, a grayscale image is used as the watermark. Copyright information in the form of a trademark or logo can be created as a pattern for this grayscale watermark image.

In the watermarking embedding process, the watermark image,  $w(x,y)$ , is first transformed into FHT coefficients by equation (1). We use a grayscale image of size  $64 \times 64$  as a watermark for testing. Thus in the FHT, Hadamard transform matrix  $H_6$  is used. After transformation,  $64 \times 64$  Hadamard transform coefficients are obtained. The DC component is stored in the hash file and the AC components are used for embedding.

The original digital image,  $f(x,y)$ , is also decomposed into a set of non-overlapped blocks of  $h \times h$ , denoted by  $f_k(x',y')$ ,  $k=0, 1, \dots, K-1$ , where the subscript  $k$  denotes the index of blocks and  $K$  denotes the total number of blocks. In our experiment, a test image of size  $512 \times 512$  and sub-block size of  $8 \times 8$  is used. The algorithm pseudorandomly selects the sub-blocks for watermark insertion using a m-sequence random number generator. The seed of m-sequence and initial state are stored in the hash file. After that, FHT is performed on each selected sub-blocks of original image by equation (1). Since the sub-block size is  $8 \times 8$ , the Hadamard transform matrix used here is  $H_3$ . Thus for each sub-block, an  $8 \times 8$  matrix of Hadamard transform coefficients is obtained.

Similar to the character embedding algorithm, not all the Hadamard transformed matrix is used for watermark insertion. Sixteen middle and high frequency components are used. If the watermark FHT coefficients are denoted by  $m_i$ , the AC components of FHT coefficients of original image sub-blocks before and after inserting watermark are denoted by  $x_i$  and  $x_i^*$  respectively, and  $i \in (0, n]$ , with  $n$  the number of the watermarked coefficients which is 16 in our experiment. The watermark strength factor is denoted by  $\alpha$ . The embedding formula is

$$x_i^* = \alpha m_i \quad (8)$$

The original coefficient  $x_i$  is replaced by  $x_i^*$ . After the watermark insertion, a new  $8 \times 8$  matrix of FHT coefficients of image sub-block is obtained. The IFHT is then applied on the  $8 \times 8$  matrix using equation (3) to obtain the luminance value matrix of the watermarked image sub-block,  $f_k'(x',y')$ . After performing the watermark insertion for all the relevant sub-blocks of the original image, the watermarked image,  $f'(x,y)$ , is obtained. At the same time, the hash file is generated for decoding process. The image-in-image watermark embedding process is shown in Figure 2: (original image lena.bmp marked with watermark image dmt.bmp)

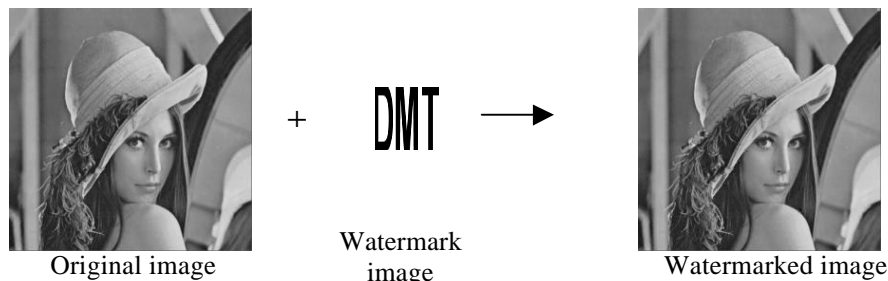


Figure 2: image-in-image watermarking embedding process

The watermarked image is then passed through a channel and possibly distorted by channel noise and external attacks. The watermark is extracted by using the embedding position and the watermark strength factor  $\alpha$  as follows. The received watermarked image is denoted by  $f'(x,y)$ . The information of watermark embedded sub-blocks' position is extracted from the seed of m-sequence and initial state number stored in the hash file. By transforming all the relevant sub-blocks,  $f_k''(x',y')$ , into the FHT domain, we get all the Hadamard transform coefficients embedded with the watermark. Using one of the sub-block FHT coefficients as an example, the watermark is inserted into the bottom right sixteen middle and high frequency components. If these components are denoted by  $x_i^{**}$ , the retrieved watermark FHT coefficients are denoted by  $m_i$ , and  $i \in (0, n]$ , with  $n$  the number of the watermarked coefficients 16. The watermark extraction formula is given as:

$$m_i = \frac{x_i^{**}}{\alpha} \quad (9)$$

All the watermark FHT coefficients are extracted from the sub-blocks of the watermarked image. The AC coefficients together with the DC component stored in hash file are rearranged into a  $64 \times 64$  FHT coefficients matrix. The extracted watermark image,  $w(x,y)$ , is obtained by IFHT of the  $64 \times 64$  Hadamard coefficients matrix using equation (3).

#### 4. WATERMARK STRENGTH FACTOR CONTROL AND POST-PROCESSING TECHNIQUE

Two main criteria of digital watermarking are to minimize data degradation and to increase the robustness of algorithm against external attacks. Minimizing data degradation is equivalent to increase the invisibility. To increase the invisibility of the watermark, the watermarking strength factor is used to control the watermark embedding strength adaptive to the original image characteristics. To increase the robustness of the watermarking system, post-processing techniques are incorporated to compensate the effects of some geometric attacks such as resizing and rotation.

##### 4.1 Watermark strength factor $\alpha$

The determination of the watermark strength factor is based on the original image textures and edges characteristic. It is found that edge information of an image is the most important factor for our perception of the image [10]. In fact, this information is required to be transmitted if the final receiver is the HVS (human visual system) [11]. It is essential to maintain edge integrity so as to preserve the image quality. Smooth areas influence our visual perception together with the edge information. The JND perception thresholds are relatively low as compared to coarse textured regions. For the texture areas, the distortion visibility is low. A coarse textured region has a very high noise-sensitivity level. With this knowledge, we can control the watermarking strength applied to the different areas of the image.

The classification of different areas is based on the Hadamard transformed space energy analysis and Canny edge detection algorithm [12]. The first visual mask model is determined by Hadamard transformed space image energy distribution. The analysis is performed on the FHT coefficients of sub-blocks for watermarking. For coarse texture and outstanding edge areas, signal energy is concentrated in AC components of FHT coefficients. In fine texture areas, most of the energy is concentrated in low AC components and DC components. We use a squared sum of AC components to generate this visual mask,  $mask_1(j,k)$  that can be used to distinguish the fine and coarse texture areas. The second visual mask model is generated by a Canny edge detection algorithm. The Canny edge algorithm was selected because of its ability in detecting weak edges by using two different thresholds. It is applied to each selected sub-blocks of the original image used for watermarking. Counting the number of edge points in each sub-block, we obtain another visual mask,  $mask_2(j,k)$ . This mask is used to determine the coarse texture or outstanding edge in the image block. Large values in this mask indicate that the corresponding block is highly textured. Smaller values indicate that the block contains outstanding edges [13]. We multiply the two mask values and scale them to the proper range, we obtain the watermark strength factor  $\alpha$  as follows:

$$\alpha = \beta * mask_1(j,k) * mask_2(j,k) \quad (10)$$

where  $\beta$  is the scaling factor,  $j$  and  $k$  indicate the positions of the sub-blocks

The watermark strength factor  $\alpha$  can be controlled according to the texture areas. Coarse textured areas are watermarked with higher strength and outstanding edge areas and smooth areas are watermarked with less strength. In this way, the invisibility of watermarked image can be improved. The strength level for image lenna.bmp sub-blocks are shown in figure 3:

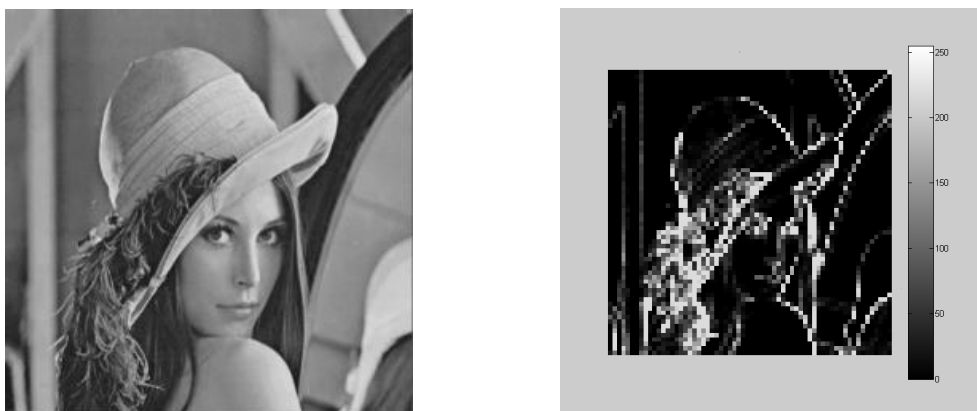


Figure 3: original lenna image and the strength level for each 8x8 sub-block

In above figure, the light areas represent coarse texture areas that are watermarked with higher strength than the dark areas that represent smooth areas that are watermarked with lower strength.

#### 4.2 Post-processing to increase the robustness

Another important criterion for a watermarking system is its robustness. It is very difficult to have the same watermark that could survive all different types of attacks by a dedicated watermarking algorithm. However, the robustness of a watermark algorithm can be further enhanced using post-processing techniques. To overcome some geometrical distortions such as rotation, translation and resizing, we propose the incorporation of posteriori estimation of the geometrical transformation and the application of the reverse transform before recovering the watermark.

In case of resizing attacks, the original image size, say  $m \times n$ , is resized to  $m' \times n'$  ( $m' < m$  &  $n' < n$ ). If the information is directly extracted from the resized image, the position of the hidden bits, stored originally in the hash-file, may now be totally misaligned and becomes incorrect. To solve this problem, we resize the image back to its original size. The original image size information is stored in the hash file. Based on this information, the attacked image  $m' \times n'$  is resized back to  $m \times n$ . Although the image pixel value may have been changed by interpolation, the position of the hidden bits can still be retrieved and corrected.

In order to retrieve the information bits correctly from a rotated watermarked image, a post-processing method to detect and rotate the attacked image back to its original orientation is proposed. Detection of the rotation angle is a crucial step in this method. In our experiment, we use log-polar mapping (LPM) of the image to trace the rotation angle [14]. The image data is normally represented by Cartesian coordinates. However, the log-polar transformation is a conformal mapping of the points on the Cartesian plane  $(x,y)$  to the points in the log-polar plane  $(\zeta, \eta)$ . The conformal mapping is described by the following equations:

$$\zeta = \log \sqrt{x^2 + y^2} \quad (11)$$

$$\eta = a \tan\left(\frac{y}{x}\right) \quad (12)$$

According to the property of the LPM, any angle rotation in the Cartesian plane will result in the shifting in the  $\eta$  axis of the log-polar plane. Since the linear shifting of the data can be measured easily and accurately, the exact angle rotated in

the attacks can be determined. Figure 4 shows the case of applying LPM to the watermarked image after it has been rotated by 30°:

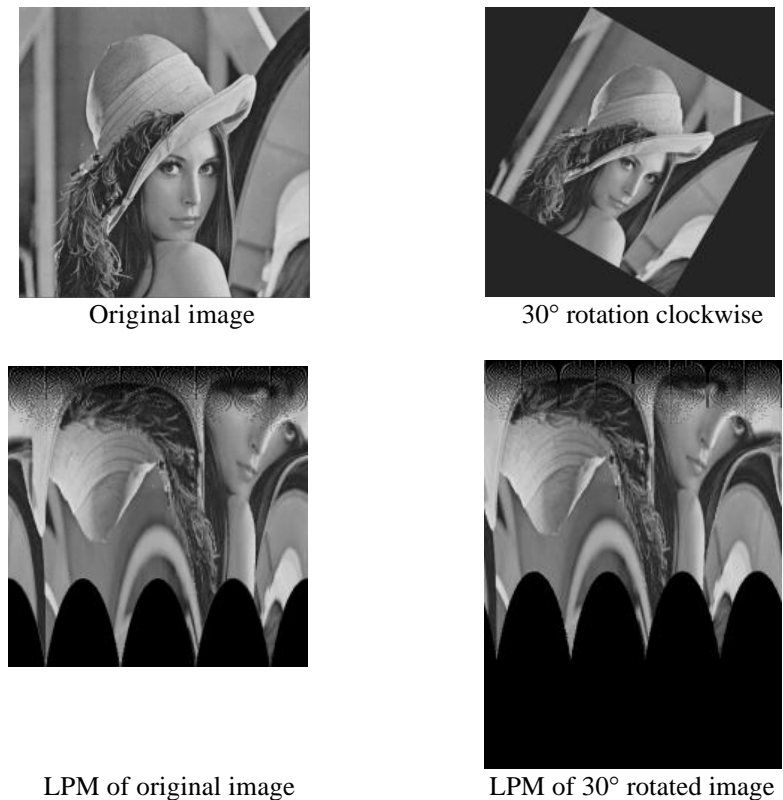


Figure 4: LPM of the lenna image

The detection of rotation angle requires a reference image to compare with the attacked image. The original watermarked image would be an ideal candidate. However, this may result in a large overhead associated with the watermarking system. Our experiment shows that a sub-block of 16×16 LPM data from the original watermarked image (512×512) is sufficient to detect the rotation angle with accuracy to approximately one degree. The reference LPM data is then stored in the hash file. In the watermarking receiver, a cross-correlation is performed between the 16×16 LPM data from the original watermarked image and the LPM data of the attacked watermarked image. The maximum value of the correlation coefficient in the  $\theta$  axis indicates the angle rotated. With this information, a reverse rotation is performed on the attacked image. The watermark embedding position is then synchronized.

## 5. EXPERIMENT RESULTS AND DISCUSSIONS

The experiment for the watermarking system is performed using the MATLAB 6 platform. Two container test images consist of lenna.bmp cameraman.bmp are selected for the experiment. All these images are of sizes of 512×512×8bit, grayscale intensity images. For the robustness test, we use the Stirmark benchmarking software that consists approximately 90 different types of image manipulations or attacks (Stirmark 3.1) [15-17]. The Stirmark attacks destroy the synchronization required in information retrieval via applying some small random geometrical transforms in such a way that the distortion is not visually perceivable. Some Stirmark attack examples such as cropping, changing aspect ratio, rotation and JPEG compression are shown in figure 5:

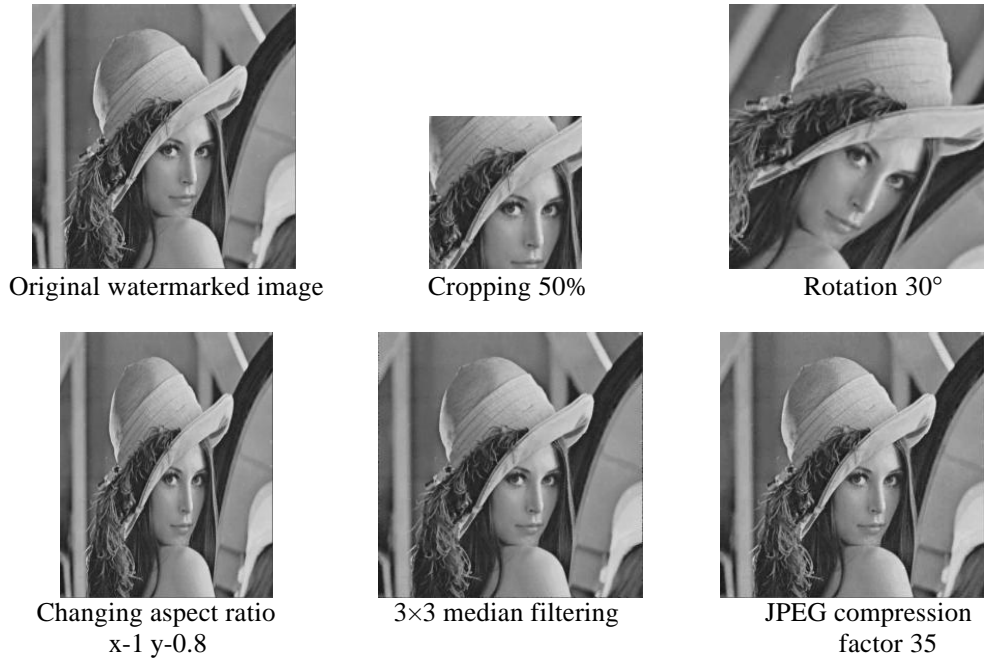


Figure 5: Stirmark attack examples on watermarked image lenna.bmp

For our algorithm, a maximum image size of  $256 \times 256 \times 8$  bit can be hidden into the container digital image of size  $512 \times 512 \times 8$  bit. In the experiment, a grayscale image dmt.bmp with size  $64 \times 64 \times 8$  bit is used. The original and watermarked image examples are shown in figure 6: (lenna.bmp marked with image dmt.bmp)

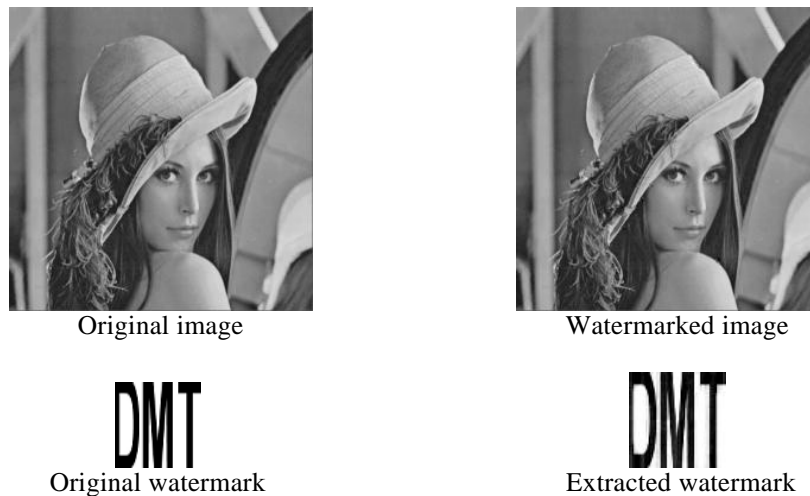


Figure 6: original and watermarked images for image embedded algorithm

Results show that there were no perceptually visible degradations on the watermarked images. The extracted watermark is also highly correlated with the original watermark with correlation factor 0.989. Results using benchmarking software Stirmark are shown in Table 1: (lenna.bmp is marked with image dmt.bmp)







| Image operations                  | Extracted watermark   | Correlation |
|-----------------------------------|---|-------------|
| Sharpening 3×3                    |  | 0.9573      |
| 1 rows 1 column removed           |  | 0.9866      |
| Frequency Mode Laplacian removal  |  | 0.9580      |
| Scaling 0.75                      |  | 0.9354      |
| JPEG Compression of factor 30     |  | 0.8688      |
| Change aspect ratio x_1.00_y_1.20 |  | 0.8199      |

Table 1: Results of some Stirmark tests for image-in-image embedding algorithm

The image embedded FHT domain watermarking algorithm was able to survive up to 60% against the Stirmark attacks. This algorithm was robust under jitter attacks. Cropping attacks up to 50% of the watermarked image could be resisted. Incorporated with post-processing, the algorithm correctly retrieved the watermark under up scaling to double of the image, downscaling to 75% of watermarked image, changing of aspect ratios either in the x-axis or y-axis and small angle rotation attacks. The proposed algorithm was able to resist frequency mode Laplacian removal (FMLR) and 3×3 sharpening attacks. It survived some level of JPEG compression, up to a compression factor of 30. However, It failed for large cropping at 75. The algorithm did not perform well against large factor of scaling-down attacks for factor 0.5. Even with some post-processing applied against this kind of attacks, the scaling-down attacks introduced a significant amount of degradation due to averaging of the original pixel values. It also performed relatively poorly against 3×3 Gaussian filtering, 2×2 median filtering and low factor JPEG compression (factor < 30).

It was also not so effective against minor random geometric transforms, such as shearing and general linear transforms, like the case in characters embedded watermarking technique. In general, the image-in-image algorithm performed worse than the characters embedded technique, because significantly more information was hidden in the container image that made it more susceptible to channel noise and external attacks. This algorithm was very efficient in term of processing time for both characters and image embedded FHT domain watermarking techniques. It took approximately 2 seconds for the embedding process and approximately 1 second for extraction using a MATLAB 6 platform running on a Pentium III 400 MHz PC system. The simplicity of the FHT also offered an advantage over the commonly used DCT and DWT techniques, in terms of ease of hardware implementation.

## 6. CONCLUSION

The project is to develop robust digital watermarking techniques to be incorporated in a digital rights management system for digital images. This report has presented a hybrid watermarking technique for embedding characters or grayscale image watermark into a container digital image based on the FHT for copyright protection. Although Hadamard transform has lower energy compaction capability than other high gain transforms, it was found to be more suitable for watermarking application because the watermark energy could spread over the whole workspace and more likely to survive under various attacks compared to other high gain transform domain such as DCT.

The experimental results showed that the proposed method is robust against most of the Stirmark attacks and Checkmark non-geometric attacks. It was also shown to be more robust than DCT algorithm under the same malicious attacks. Moreover, the Hadamard transform offers a significant advantage in shorter processing time and ease of hardware implementation than commonly used DCT and DWT techniques.

## REFERENCES

- [1] R. G. van Schyndel, A. Z. Tirkel, and C. F. Osborne, "A digital watermark," Proc. IEEE Int. Conf. Image Processing, vol. 2, pp. 86–90, 1994.
- [2] I. J. Cox, J. Kilian, F. T. Leighton, and T. Shamoan, "Secure spread spectrum watermarking for multimedia," IEEE Trans. Image Processing, vol.6, pp. 1673–1687, Dec. 1997.
- [3] C.-T. Hsu and J.-L. Wu, "Hidden digital watermarks in images," IEEE Trans. Image Processing, vol. 8, pp. 58–68, Jan. 1999.
- [4] E. H. Hall, Computer Image Processing and Recognition, New York: Academic Press, 1979.
- [5] R. C. Gonzalez and P. Wintz, Digital Image Processing, Reading, MA: Addison-Wesley, 1977.
- [6] A. N. Akansu and R. A. Haddad, Multiresolution Signal Decomposition: Transforms, Subbands and Wavelets, Academic Press Inc., 1992.
- [7] M. Ramkumar, "Data Hiding in Multimedia – Theory and Applications", PhD Thesis, Jan.2000
- [8] Guoan Bi and B. G. Evans, "Hardware Structure for Walsh-Hadamard transforms," Electronic Letters, Vol.34, No.21, pp. 2005-2006, Oct. 1998
- [9] S. Katzenbeisser and A. P. F. Petitcolas, Information Hiding Techniques for Steganography and Digital Watermarking, Artech House, 2000
- [10] M. S. Kankanhalli, Rajmohan, and K. R. Ramakrishnan, "Content Based Watermarking of Images," ACM Multimedia 98 - Electronic Proceedings, Sept 14-16, 1998
- [11] L. Torres and M. Kunt, Video Coding, The Second Generation Approach. Kluwer Academic Publishers, 1995.
- [12] J. Canny, "A Computational Approach to Edge Detection," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 8, No. 6, Nov 1986
- [13] D. Taskovski, S. Bogdanova and M. Bogdanov, "A Low Resolution Content Based Watermarking of Image in Wavelet Domain", Image and Signal Processing and Analysis, 2001. ISPA 2001. Proceedings of the 2nd International Symposium on , 2001 Page(s):604-608
- [14] E. L. Schwartz, "Spatial Mapping in the Primate Sensory Projection: Analytic Structure and Relevance to Perception," Biological Cybernetics, vol. 25, pp. 181-194, 1977.
- [15] F. A. P. Petitcolas and R. J. Anderson, "Evaluation of Copyright Marking Systems," in IEEE Multimedia Systems, Florence, Italy, 7-11 Jun. 1999, pp. 574-579
- [16] F. A. P. Petitcolas, R. J. Anderson, and M. G. Kuhn, "Attacks on Copyright Marking Systems," in Proceedings of the Second International Workshop on Information Hiding, vol. 1525 of Lecture Notes in Computer Science, Springer,1998, pp 218-238.
- [17] F. A. P. Petitcolas and M. G. Kuhn, Stirmark.  
<http://www.cl.cam.ac.uk/~fapp2/watermarking/benchmark/>, 1999