A DWT Based Steganography Approach

EE604 Term Paper

Instructor: Prof. Sumana Gupta

Group No. 1

Group Members

Anirudh Kumar Agrawal, 11907098

Pratik Likhar, 11531

Radhika Ravi, 11553

Introduction

Image Steganography refers to hiding of data inside an image in which the changes should be imperceptible to human eye. A common use of steganography is in watermarking which is used to prevent illegal distribution of content by hiding a signature in the image of the copyright. Watermarking techniques generally have following features:

1. Imperceptibility : Human eye cannot distinguish between the watermarked and original image
2. Security : The watermarked image cannot be copied, deleted or modified by an animus observer
3. Robustness : The watermark still can be extracted out within certain acceptable quality even the image has endured some signal processing or noises before extraction
4. Statistically Undetectable : It is extremely hard or impossible to detect watermark using statistical methods
5. Blind Detection : The extracting have not access to the original image

The methods can be broken down into two main broad categories:

1. Spatial Methods: the processing is applied on the image pixel values directly. This easy and computationally fast, but is affected by signal processing and noises
2. Frequency Domain Methods:the first stepis to transform the image data into frequency domain coefficients by some mathematicaltools (e.g. FFT, DCT, or DWT). Then, accordingto the different data characteristicsgenerated by these transforms, embed thewatermark into the coefficients in frequencydomain. After the watermarked coefficientsare transformed back to spatial domain, theentire embedding procedure is completed.The advantage of this type of watermarking isthe high ability to face some signal processingor noises. However, methods of this type are computationally complex and hence slower.

Algorithm

The frequency domain transform used here is the Haar-DWT which is applied as following

Step 1: At first, scan the pixels from left toright in horizontal direction. Then, perform the addition and subtraction operations on neighboring pixels. Store the sum on the left and the difference on the right. The pixel sums representthe low frequency part (denoted as symbol L) while the pixel differences represent the highfrequency part of the original image (denoted as symbol H).

Step 2: Secondly, scan the pixels from top to bottom in vertical direction. Perform addition
and subtraction operations on neighboring pixels and then store the sum on the top and the difference on the bottom as illustrated in Figure 3. Repeat this operation until all thecolumns are processed. Finally we will obtain 4 sub-bands denoted as LL, HL, LH, and HH respectively.

There are multiple modes based on the relation between image sizes and the text to be embedded

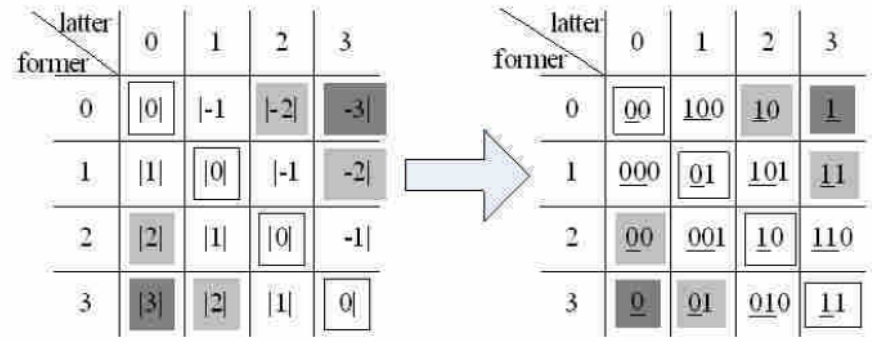
1. Fix Mode

There is a fix embedding capacity requirement n = M x N x (2/4) x 4 bits, following steps are used to achieve the same

Embedding Procedure

Step 1: Apply Discrete Wavelet Transform (DWT) on the grayscale image and obtain the four sub-bands (LL, HL, LH, and HH)
Step 2: Combine every two bits to form a decimal value ranging from 0 to 3. Every two decimal value pair is subtracted to form a sequence of values
Step 3: The embedding process follows the raster scan order the absolute differences values In this case, we embed the absolute difference values at two rightmost LSBs in LH and HL (LH sub-band first, then HL sub-band) and the status values at those in HH and furthermore (the third one, and the fourth one if needed) LSBs in LH and HL sub-bands. Based on Figure 1, the coding on the right Table is designed to record the possible subtraction pairs. The codes underlined are embedded in the third or fourth LSBs in LH and HL sub-bands while the others are embedded in HH (and the first LSB for LH and the second for HL if needed). The embedding positions in HH are just those corresponding positions in LH and HL

Left table: 4 possible absolute values Right table: Status of subtraction pairs



former \ latter	0	1	2	3
0	|0|	|-1	|-2|	-3|
1	|1|	|0|	|-1	-2|
2	|2|	|1|	|0|	-1|
3	|3|	|2|	|1|	0|

former \ latter	0	1	2	3
0	00	100	10	1
1	000	01	101	11
2	00	001	10	110
3	0	01	010	11

Figure 1

Step 4: After embedding all the message bits we obtain modified DWT coefficients, we perform IDWT on the stego image due to LSB substitution some pixels are not integers and will result in the floating point values, a key matrix was used to store the 4 possible non-integers situations (0.0, 0.25, 0.50, 0.75).

Step 5: The rounded version is the final image F and the key matrix is stored in the example, the ”Description” tag in TIFF format and the “Comment” tag in JPG format

Extraction Procedure

Step 1: Extract the key Matrix from the file tag and transform all the elements of K into 0.0, 0.25, -0.5, -0.25 to K’

Step 2: Obtain H’ matrix by performing DWT transform on E = F + K’

Step 3: Extract the absolute values (0, 1, 2 and 3) from the 2 rightmost LSBs in HH sub-band. According to the value extracted, LSBs of corresponding positions in LH and HL are used to determine the subtraction pair. Base on the mapping rules defined in Figure 1, we can reconstruct 2 values (former and latter) of the decimal sequence. Cue these decimal values in correct order and then expand them to a binary bit stream.

2. Varying Mode

In varying mode number of embedded bits belongs to specific range

Embedding Procedure

Step 1: Apply DWT on the grayscale image and obtain the four sub-bands LL, HL, LH and HH

Step 2: Combine every two bits to form a decimal value ranging from 0 to 3. Every two decimal value pair is subtracted to form a sequence of values. There are only 4 possible absolute values (0, 1, 2, and 3) for the elements in this differential sequence. Record these absolute values in HH by substituting the 2 LSBs of coefficients in HHH with 00, 01, 10, and 11 respectively. However, same absolute value might be consequence of different subtraction pairs. Hence, we need more bits to distinguish the subtraction status. In Figure 1, the right Table coding is designed to record the possible subtraction pairs. The codes underlined are embedded in LH and the others are embedded in HL. Embedding positions in HLH and HHL are just the corresponding positions in HH.

Step 3: The remaining bits of S are embedded at those unused LSBs in LH and then HL bit by bit. For example, if the value embedded in HH is 1, we cannot embed any more message bit at the corresponding position of LH but 1 more bit at the corresponding position in LH.

Step 4: After embedding all message bits, we obtain the slightly modified coefficients matrix H'. By performing the inverse DWT (IDWT) on H', the stego-image E is obtained. Now we employ a “Key matrix” - K to record the 4 possible non-integer situations (0.0, 0.25, 0.5 and 0.75). The rounded pixel values of E are used to show the stego-image. In order to perfectly reconstruct the secret message bits, K is necessary in the extracting procedure

Step 5: The rounded version of E, denoted as F, is then stored in a specific image file format while K is filled in the unused tags (for example, the” Description” tag in TIFF format or the “Comment” tag in JPG format.

Extraction Procedure

Step 1: Extract the key Matrix from the file tag and transform all the elements of K into 0.0, 0.25, -0.5, -0.25 to K’

Step 2: Obtain H’ matrix by performing DWT transform on E

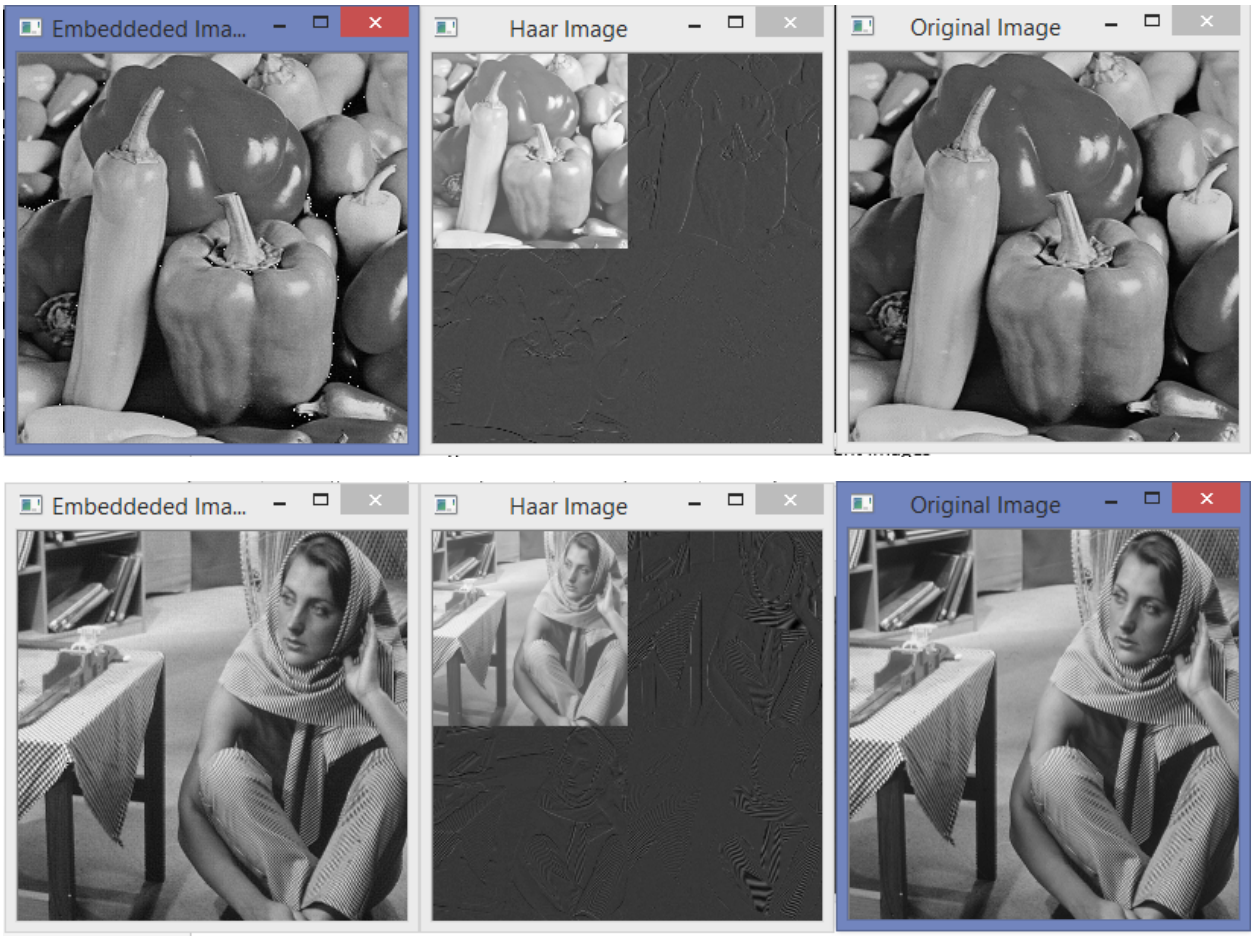
Step 3: Extract the absolute values (0, 1, 2 and 3) from the 2 rightmost LSBs in H'HH sub-band. According to the value extracted, LSBs of corresponding positions in H'LH and H'HL are used to determine the subtraction pair. Based on the mapping rules defined in Table 1, we can reconstruct 2 values (former and latter) of the decimal sequence. Cue these decimal values in correct order and then expand them to a binary bit stream.

Step 4: By extracting some more second LSBs in H'LH and H'HL according to the rule illustrated in Figure 14, we obtain the remaining portion of S. Cascade it with the sequence obtained in step 3, the whole message bit stream S is completely extracted.

Results

Fixed Mode

Following pictures are obtained when running the algorithm for fixed mode



The following table shows the PSNR obtained for different images

Images	PSNR(dB)
Lena	40.4420
Peppers	40.8364
Barbara	40.4688
Mandril	40.5798
Boat	40.5428

Varying Mode

Following pictures are obtained when running the algorithm for varying mode

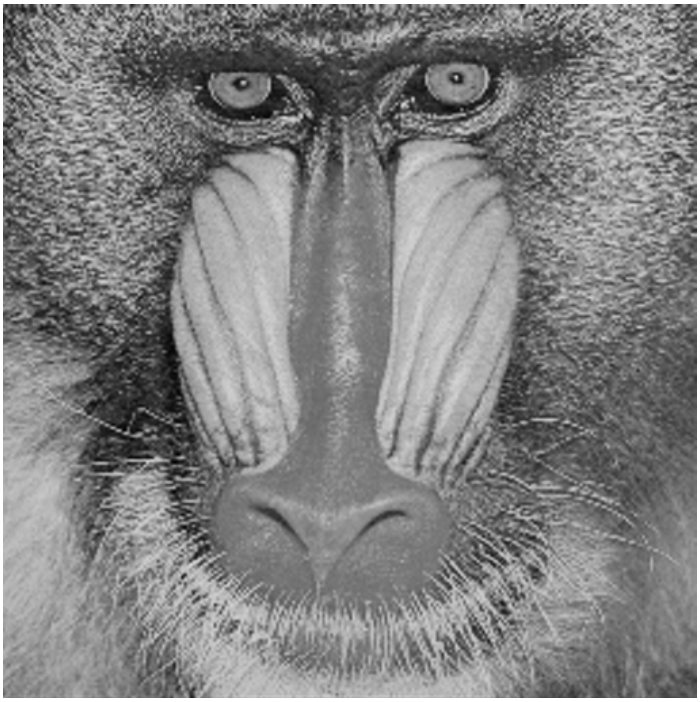
LENA

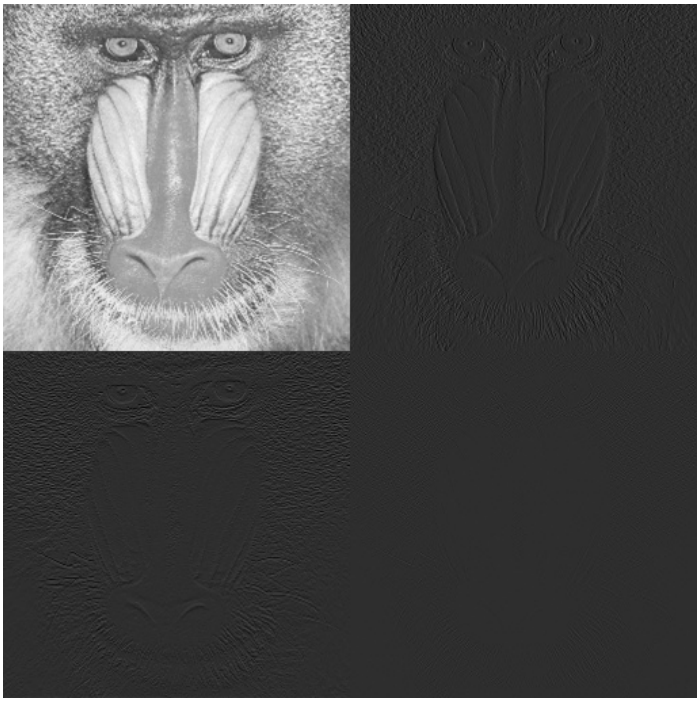
CASE 1: $n \leq (3/4)\ MxN$

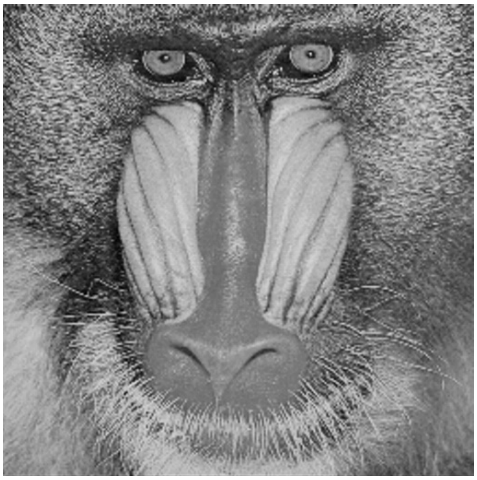
CASE 2: $(3/4)\ MxN \leq n \leq MxN$

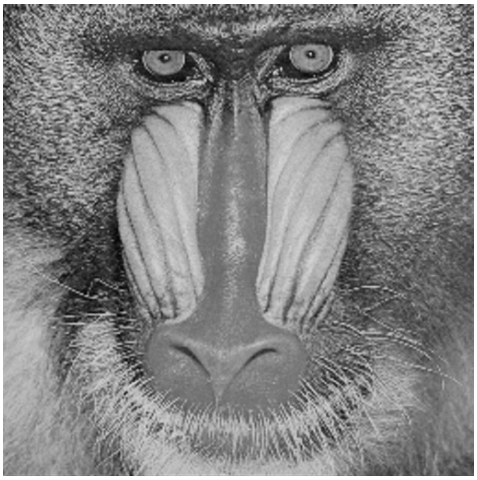
CASE 3: $MxN \leq n \leq (9/8)\ MxN$

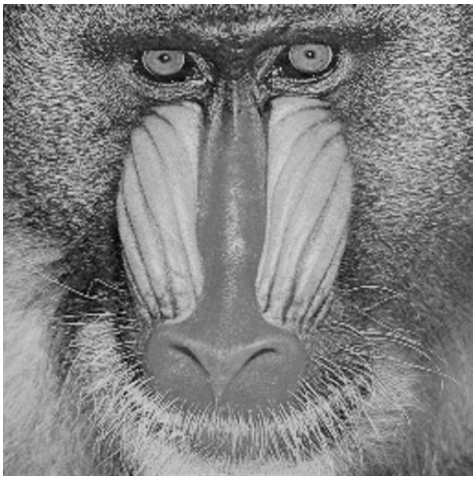
MANDRIL











The following table shows the PSNR obtained for different images

Images	**PSNR(dB)**		
	Case-1	**Case-2**	**Case-3**
Lena	49.2008	46.2553	44.2926
Peppers	49.2477	46.1719	44.3210
Mandril	49.0497	46.2782	44.4409
Barbara	49.0302	46.1832	44.3737
Living Room	49.0077	46.2201	44.4061

To the check the robustness of the algorithm with fixed mode case under external attacks, we tried the following cases

1. Noise

   Noise can be added to the image while transmission, we used Additive Gaussian Noise for simulating the factor based on which following results were obtained

Image	Error
Lena	60.56%
Mandril	60.55%
Barbara	60.58%

Table 1 : Error with Gaussian Noise (sigma = 2)

Image	Error
Lena	69.55 %
Mandril	69.54%
Barbara	69.54%

Table 2: Error with Gaussian Noise (sigma = 5)

   Thus we can see that the procedure does not fare well under addition of noise

2. Rotation

   The image might be rotated by some angle theta, for simplicity we assume that the image when received was rotated by an angle of 90 degrees. Thus, after rotating the image by -90 degrees we run the extraction algorithm once again, the results are recorded in table 3.

Image	Error
Lena	0.29 %
Mandril	0.29 %
Barbara	0.29 %

Table 3: Error with rotation of image by 90 degrees

Image	Error
Lena	73.78 %
Mandril	73.56 %
Barbara	73.45 %

Table 4: Error with rotation angle of the image 63 degrees

   Above results lead us to believe if the angle of rotation is an integer multiple of 90 degrees we can extract the message reasonably, but in other cases we fail to do so.

Discussion

Haar Transform was chosen because of the computational simplicity and the fact that we can accurately reproduce the message using the keymatrix which would have been difficult using other transform. Also note that the LH represents horizontal edges, HL represents vertical edges and HH sub-band represents the diagonal edges. Hence the good performance of the algorithm can be attributed to the fact that is only changes the values of the edges.

In terms of the capacity of the given method for an image of size 256 x 256 we are able to embedded a message of size 256 * 256 * (2/4) * 4 = 131072 bits using 256 *256 * (2 /4) + 256 * 256 * (2/4) *4 = 163840 bits in the image, while the total no. of bits in the original image is 256*256*8 = 524288. The space requirement of the process is nearly = 0.3125.

A note on the security issue, an observer cannot successfully decrypt the message without knowing the 'Key Matrix' and even if the person know the key matrix he also needs to know the mapping rules which differ from case to case. Hence making the method secure.

Conclusion

The proposed steganography technique was implemented and its correctness was verified. Further tests on the stability of the algorithm under external attacks give the conclusion that the algorithm