

Implementation of a Modified LSB Algorithm Based on Pixel Value Differencing for Image Watermarking

Bhargav Ram M V¹, Suhas J R V², Yashas J³, P Nagaraju⁴U.G. Student, Department of Telecommunication Engineering, R College of Engineering, Bengaluru, India^{1,2,3}Associate Professor, Department of Telecommunication Engineering, R V College of Engineering, Bengaluru, India⁴

ABSTRACT: Watermarking in the spatial domain is an effective way of hiding a data into a digital image. Least Significant Bit (LSB) watermarking algorithm successfully embeds a watermark image onto a host image. But in noisy environments, the correlation coefficient(CC) value of the extracted watermark becomes very low which indicates the algorithm is not suitable in noisy environments. Thus a new algorithm is formulated using the concept of pixel value differencing. The algorithm is imposed on a greyscale image of size 128×128 and a binary watermark of size 64×64 on MATLAB tool. The performance of the algorithm is analysed using PSNR, MSE and CC. Noises like Speckle, Salt and Pepper and Gaussian are considered to evaluate the performance of the algorithm under noisy conditions. The results indicate that the proposed algorithm is more resistant to the above mentioned noises than the LSB watermarking algorithm.

KEYWORDS: LSB algorithm, Pixel value differencing, Threshold PSNR, MSE, CC, Gaussian, Speckle, Salt and Pepper.

I. INTRODUCTION

Watermarking has been a pivotal tool in the era of internet to authenticate the owner of the data. Embedding a watermark image onto a host image directly by manipulating the pixel values is branched out to be the spatial domain technique which includes Least Significant Bit (LSB) watermarking algorithm, Modulo algorithm, Histogram method etc. Embedding a watermark image onto a host image by manipulating the transformed pixel values is the transform domain technique which includes Discrete Wavelet Transform (DWT) watermarking algorithm, Discrete Wavelet Transform-Singular Value Decomposition (DWT-SVD) watermarking algorithm etc. [1].

Spatial domain algorithms are less complex to be implemented since they require a significantly less memory, whereas the transform domain algorithms are complex to be implemented since they require a large memory. But, the transform domain algorithms are very resistant to the noise attacks unlike the spatial domain algorithms [2]. In order to suffice all the requirements of a user, it is desirable to exploit the advantages of both kinds of algorithms and formulate a new algorithm which should be easy to implement and resistant to noise.

In [4], the basics of watermarking and various security criteria are summarized. The quality of the image after various attacks is studied. In [5], embedding the message onto the image has been carried out using the LSB watermarking algorithm. The effects of imposing bits at various positions of the host image are also studied. Watermarking bits are appended at the least significant position of the host pixel, thus manipulating the pixel values directly.

In [6], the third and the fourth LSBs are used to watermark an image. It is comparatively more robust than the classic LSB technique. In this algorithm, two bits are embedded in the third and the fourth LSBs. By adding two bits to one pixel of a host image, it is evident that the embedding capacity is increased.

In [8], difference expansion and LSB embedding are considered for reversible data embedding and authentication. The embedding capacity is drastically increased. Difference between a set of pixels is calculated and the LSB of the difference is considered to embed the watermark. Here, though the manipulation is done directly on the pixel values, a difference variable is considered to implement the changes thus maximizing the capacity.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 4, April 2016

II. METHODOLOGY OF THE ALGORITHM

A. Embedding: Ideally a watermark image should be imposed on a host image in such a way that it can't be visible to the naked eyes after embedding and the information can be retrieved during extraction. Any image can be chosen for the execution, but the standard Lena image is taken for comparison purpose. The chosen watermark is a binary image containing only 0 and 1s.

The embedding consists of the following steps:

1) Calculating the pixel difference: The pixel array of the host image is stored. The absolute difference between the adjacent values is calculated for the whole array. Let the pixel values be

Pixel values: 125 148 245 12 44 78 89

Difference: 23 97 233 32 34 11

The absolute difference between 125 and 148 is 23, thus the value is stored in an array. The same process is carried out for the whole pixel array. If there is 'N' number of pixels, then there can be 'N-1' absolute differences.

2) Choosing a threshold: A threshold value is chosen in order to differentiate the smooth pixel set from the edge pixel set. If the difference is below the chosen threshold value, then the pixel set is called smooth pixel set. If the difference is above the chosen threshold value, then the pixel set is called edge pixel set. The threshold can be any value lesser than the average value of the pixel array. In this paper, we have considered a threshold of 10. Embedding is carried out as follows:

Case1: if the difference is lesser than 10
Embed one watermark bit to the host

Case 2: if the difference is greater than 10
Embed two watermark bits to the host

Thus, if the watermark image is smaller than the host image, there is no need to scan through all the pixel values and calculate the difference. A break point is given at the end so that if the watermark data is finished being scanned, the process halts.

B) Extraction: Ideally, we should make sure that the retrieved watermark is exactly same as the taken watermark. The pixel difference of the embedded image is calculated just like in embedding process.

Extraction is carried out as follows:

Case 1: if the difference is lesser than 10
Extract the last bit

Case 2: if the difference is greater than 10
Extract the last two bits

To demonstrate the logic of the algorithm, let us take a few data. Let the host pixels be {123 125 226 12} and the watermark pixels be {1 0 0 1}. A threshold value of 10 is chosen. The absolute difference between the first two pixels is $|123-125|=2$ which is lesser than the threshold value, thus only one watermark bit is appended in the least significant position of the host pixel. The absolute difference between the next two pixels is $|125-226|=101$, which is higher than the threshold value, thus two watermark bits are appended in the least two significant bits of the host pixel which manipulates the data to be 124. The absolute difference of the next two pixels is $|226-12|=214$ which is greater than the threshold, yet there is only one data is left in the watermark array, thus the last bit is appended in the least significant bit of the host image which makes the value to be 227. Thus the embedded pixel array is {123 124 227 12}.

While extracting, the same logic is applied. The embedded pixel values are {123 124 227 12}. The absolute difference between the first two pixel values is $|123-124|=1$ which is lesser than the threshold value, thus the last bit of the pixel value '1' is extracted and stored. The absolute difference between the next two pixel values is $|124-227|=103$ which is greater than the threshold, thus last two bits of the pixel value '00' is extracted and stored. The absolute difference between the next two pixels is $|227-12|=215$, which is higher than the threshold value. Yet the watermark size is restricted to only four values, thus the last bit '1' is extracted and stored. The extracted watermark is {1 0 0 1} which matches the taken watermark array. This logic is applied for a host pixel size of 128×128 and a watermark pixel size of 64×64 . If the array is traversed till the last pixel, only one bit is appended ultimately as there are no pixels values after the last one.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 4, April 2016

III. HARDWARE IMPLEMENTATION

The algorithm is implemented on TMS320C6713 which operates at 225MHz. The platform used to implement the code on DSK6713 is Code Composer Studio. A host image of size 64*64 and a watermark image of size 64*64 are taken to implement the code on hardware. Various steps involved in the implementation is as follows.

1. The Bitmap images are converted to their respective .dat files which can be written directly to the SDRAM of the processor. This is done because the file handling functions takes a significant amount of time.
2. The embedding and extraction codes are implemented using C language.
3. The code is run in the IRAM of the processor.
4. The data is saved after the execution for the analysis. The .dat file is converted back to the bitmap format. MATLAB is used to calculate the parameters of the algorithm.
5. Various noises are imposed on the embedded image and the noisy images are again dumped on the processor to analyze the performance of the algorithm in noisy environments.

IV. EXPERIMENTAL RESULTS

A grayscale host image of size 128*128 and a binary watermark image of size 64*64 are considered for the algorithm design on MATLAB tool. Several parameters like PSNR, MSE and CC are calculated using standard MATLAB functions.

Figure 1 shows the host image, the embedded image, the watermark image and the extracted watermark. It can be observed that the watermark has been embedded successfully as it is invisible. The extracted watermark matches the taken watermark image satisfying the conditions.



Figure 1: (i) Host image (ii) Watermarked image (iii) Watermark image (iv) Extracted watermark

Table 1 and Table 2 show the performance parameters of the proposed algorithm and the LSB watermarking algorithm. It is evident by a high value of PSNR that the embedded host image is highly perceptible. A CC value nearer to '1' proves that the extracted watermark is impeccable. The parameters of both algorithms match in a noiseless environment.

TABLE 1
PARAMETERS OF THE NEWLY PROPOSED WATERMARKING ALGORITHM

PSNR	56.9709
MSE	0.1306
CC	0.9994

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 4, April 2016

TABLE 2
PARAMETERS OF THE LSB WATERMARKING ALGORITHM

PSNR	51.1311
MSE	0.5012
CC	1

Table 3 shows the comparison between the CC values of the LSB watermarking algorithm and the newly proposed algorithm in presence of the Speckle noise. As it can be seen, even if the variance is as low as 0.1, the CC value reaches a negative value in LSB algorithm, but the algorithm is refined in such a way that the CC value is increased up to 0.5432. Theoretically, as the Variance increases, the CC value should decrease, but in the newly proposed algorithm, the CC value is constant over a wide range of Variance.

TABLE 3
COMPARISON BETWEEN THE CC VALUES IN PRESENCE OF SPECKLE NOISE

Variance	CC(NEW)	CC(LSB)
0.1	0.5432	-0.0052
0.2	0.5441	0.0134
0.3	0.5165	0.0060
0.4	0.5105	-0.0029
0.5	0.5167	-0.0079
0.6	0.5050	-0.0065
0.7	0.5111	-0.0087
0.8	0.5210	-0.0022
0.9	0.4912	-0.0046
1	0.5014	-0.0064

Table 4 and Table 5 represent the variation in the CC values of the LSB watermarking algorithm and the newly proposed algorithm for salt and pepper noise and Gaussian noise. Since LSB is resistant to salt and pepper noise, the CC values of both algorithms are almost same. CC decreases slightly as the intensity of salt and pepper noise increases.

In the Gaussian noise environment, LSB is very fragile which can be inference by very low values of CC. But the newly proposed algorithm has a constant higher value of CC for a wide range of variance of Gaussian noise. For a variance of 0.1, CC is found to be 0.5783 and for a variance of 0.7, CC is found to be 0.5786, thus pointing out the fact that the CC value of the newly proposed algorithm is almost constant.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 4, April 2016

TABLE 4

COMPARISON BETWEEN THE CC VALUES IN PRESENCE OF SALT AND PEPPER NOISE

Intensity	CC(NEW)	CC(LSB)
0.1	0.8791	0.8924
0.2	0.7897	0.7833
0.3	0.7366	0.6871
0.4	0.6728	0.5785
0.5	0.6263	0.4845
0.6	0.6217	0.9843
0.7	0.6199	0.2371
0.8	0.6068	0.2112
0.9	0.6012	0.0980
1	0.5998	-0.060

TABLE 5

COMPARISON BETWEEN THE CC VALUES IN PRESENCE OF GAUSSIAN NOISE

Variance	CC(NEW)	CC(LSB)
0.1	0.5783	0.0028
0.2	0.5721	0.0025
0.3	0.5990	0.0012
0.4	0.6775	0.0079
0.5	0.6225	-0.0043
0.6	0.5221	0.0079
0.7	0.5786	0.0012
0.8	0.5225	-0.0087
0.9	0.6001	-0.0022
1	0.5889	0.0022

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 4, April 2016

Figure 2 shows the embedded images with noises applied and the watermark extracted from the respective embedded images using LSB watermarking algorithm. As it was made clear from the tables, LSB watermarking algorithm can be used if the noise is salt and pepper. It is observed that the extracted watermark is almost same as the taken watermark image, thus the CC value of 0.8820. For Gaussian and Speckle noises, the extracted images are completely corrupt, thus making the LSB watermarking algorithm not suitable under noisy conditions.

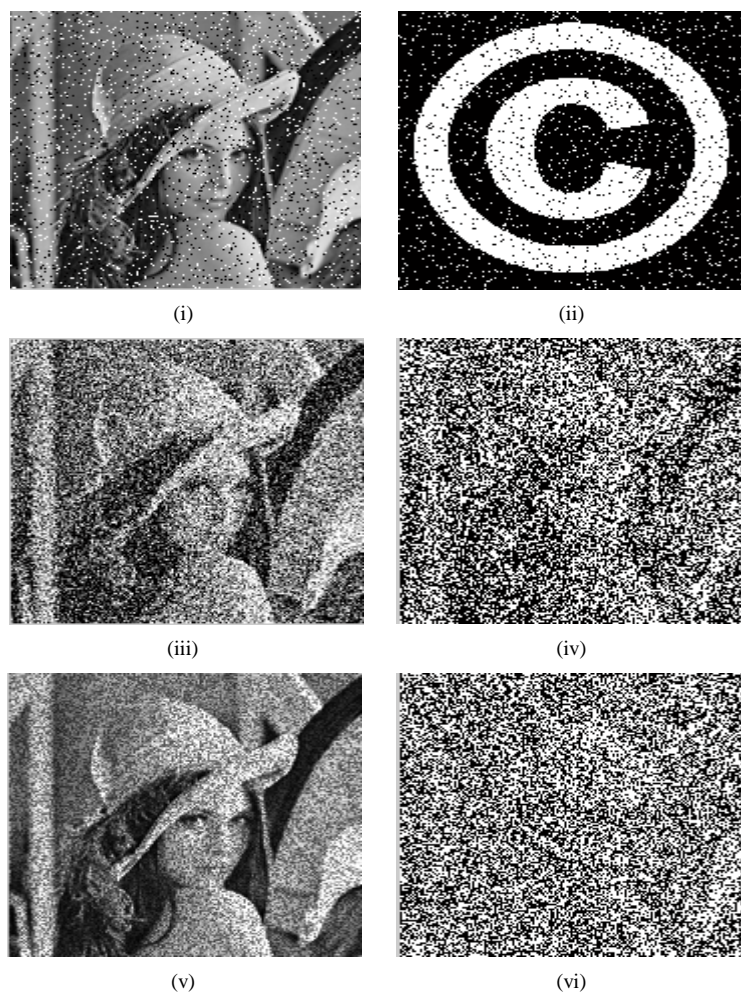


Figure 2: Various attacked watermarked images and their corresponding extracted watermarks using LSB watermarking algorithm (i) Salt and pepper noise intensity 0.1 (ii) Extracted watermark(CC=0.8820) (iii) Gaussian variance 0.1 (iv) Extracted watermark(CC=-0.0622) (v) Speckle variance 0.1 (vi) Extracted watermark(CC=-0.0391)

Figure 3 shows the embedded images with noises applied and the watermark extracted from the respective embedded images using the newly proposed watermarking algorithm. It can be observed from the figure that the algorithm serves well for all three noises. Under salt and pepper conditions, the CC value is 0.8791 and the watermark image can be matched to the taken watermark image. In Gaussian conditions, though the extracted watermark is corrupted, the watermark can be perceived. In Speckle conditions, the CC value is 0.5432 for the variance of 0.1 indicating the fact that though the extracted watermark does not exactly match the taken watermark image, it is sufficient enough to

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 4, April 2016

differentiate the watermark image from the extracted watermark image. The results obtained prove that the fragile watermarking algorithm is improved to get a semi-fragile watermarking algorithm by simple modifications.

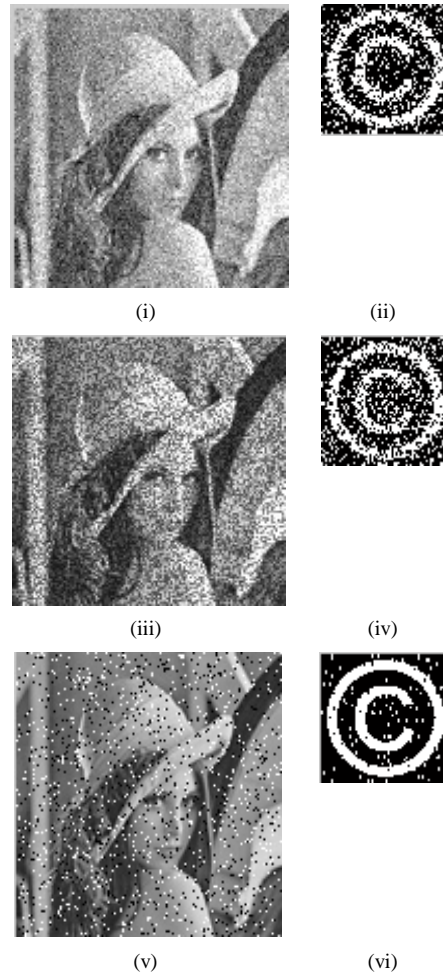


Figure 3 : (i) Watermarked image with Gaussian noise(Mean=0;V=0.1) (ii) Extracted watermark(CC=0.5783) (iii) Watermarked image with Speckle noise(V=0.1) (iv) Extracted watermark(CC=0.5432) (v) Watermarked image with Salt and Pepper noise(V=0.1) (vi) Extracted watermark(CC=0.8791)

Figure 4 shows the embedded host image obtained on CC Studio using DSK6713. The path View-> Display-> Image is used to display the image on the platform.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 4, April 2016

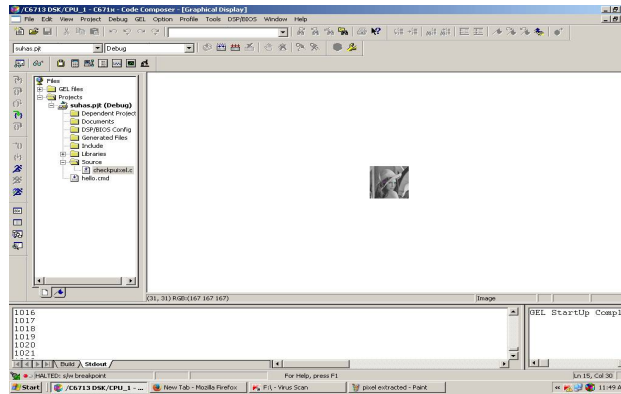


Figure 4: Embedded image on CC Studio.

Figure 5 shows the extracted watermark on CC Studio using DSK6713. The extracted of size 64*64 is displayed on the screen using the above mentioned path.

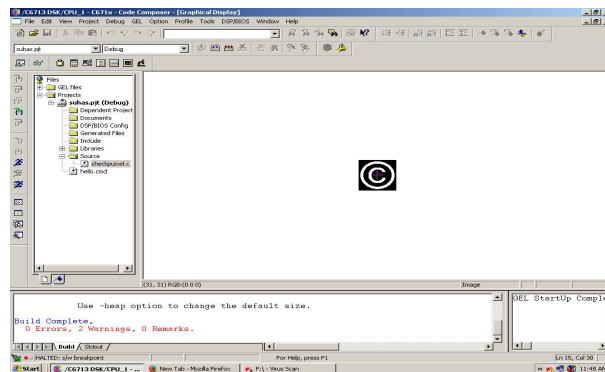


Figure 5: Extracted image on CC Studio

Both images are saved using the path File->Data->Save and converted back to bitmap format. Various parameters are calculated on MATLAB tool to compare the hardware results with the MATLAB results. Table 6 shows the hardware parameters which match the parameters that are obtained from MATLAB tool.

TABLE 6
PARAMETERS OF THE HARDWARE IMPLEMENTED ALGORITHM

PSNR	48.1278
MSE	0.542
CC	0.9987

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 4, April 2016

V. CONCLUSION

A semi-fragile watermarking algorithm was formulated to overcome the drawbacks of the LSB algorithm. The performance was analysed using PSNR, MSE and CC on various noises like Salt and Pepper, Gaussian and Speckle. Results showed that the newly proposed algorithm serves better in the noisy conditions. The CC value of the extracted watermark stays constant for a variance range 0-1 by which it is evident that the watermark can be extracted even in the harsh conditions. Computation time depends on the size of the watermark.

Future scope resides in embedding more number of bits based on various pixel averaging values. The algorithm can be manipulated for different threshold values as and when required. The algorithm can be implemented and tested on other platforms for faster computations.

REFERENCES

- [1] Puneet Potdar, Vidyasagar M., Song Han, and Elizabeth Chang. "A survey of digital image watermarking techniques." 3rd IEEE International Conference on Industrial Informatics, 2005.
- [2] A.K. Singh, N. Sharma, M. Dave and A. Mohan, "A novel technique for digital image watermarking in spatial domain(PDGC), 2nd IEEE International Conference on Parallel Distributed and Grid Computing, Solan, pp. 497-501,2012.
- [3] Caldelli, R., Filippini, F. and Becarelli, R, "Reversible watermarking techniques: an overview and a classification", EURASIP Journal on Information Security, 2010.
- [4] D.Chopra, P.Gupta, Gaur Sanjay B.C and A. Gupta, "Lsb Based Digital Image Watermarking For Grayscale Image," IOSR Journal of Computer Engineering (IOSRJCE) vol. 6, no. 1, pp. 36-41,2012.
- [5] Kr Sharma and Rajni , "Analysis of Image Watermarking using Least Significant Bit Algorithm", International Journal of Information Sciences and Techniques (IJIST) Vol.2, No.4, July 2012.
- [6] A. Bamatraf, R. Ibrahim and M. N. B. M. Salleh, "Digital watermarking algorithm using LSB,"International Conference on Computer Applications and Industrial Electronics (ICCAIE), Kuala Lumpur, pp. 155-159, 2010.
- [7] Song, Chunlin, Sud Sudiman, Madjid Merabti and David Llewellyn-Jones, " Analysis of digital image watermark attacks", IEEE Consumer Communications and Networking Conference (CCNC), pp. 1-5, 2010.
- [8] Tian, Jun. "Reversible data embedding using a difference expansion." IEEE Transaction on Circuits Systems, pp. 890-896, 2003.
- [9] Dabas, Pooja, and Kavita Khanna, "A study on spatial and transform domain watermarking techniques", International Journal of Computer Applications ,2013.
- [10] U. Kumari, T. C. Thanuja, J. Vinay, K. N. Bhushan and N. S. Vasanad, "Watermarking using modulo algorithms for gray images", International Conference on Signal and Image Processing (ICSIP), pp. 343-348, 2010.
- [11] Mathai N.J, Kundur D, Sheikholeslami A, "Hardware implementation perspectives of digital video watermarking algorithms", IEEE Transactions on Signal Processing, Volume: 51, Issue:4, pp. 925-938,2003.
- [12] "TMS320C6713 DSK Technical Reference", 506735-0001 Rev. A, May 2003.
- [13] "TMS320C6713 Floating-Point Digital Signal Processor", SPRS186L – December 2001 – Revised November 2005.