ISSN 2250-3153

Image Steganography using Modified DWT Algorithm for Embedding and Extracting data.

Kaipa Rajeswara Reddy,

* Vasireddy Venkata Institute of Technology

Abstract- **In this paper, we propose a modified version of the Haar-DWT algorithm for steganography which embeds data with very minimal change in the quality of the embedded image.**

Index Terms- DWT(Discrete Wavelet Transform), LSB(Least Significant Bit), IDWT(Inverse Discrete Wavelet Transform)

I. Introduction

Image Steganography is a technique used to hide data in an image that can be used for secure data exchange. One of the famous algorithms that are used for image steganography is DWT(Discrete Wavelet Transform) Implementation of DWT is tricky and has many issues such as decimal and negative values which cannot be stored in an image. To solve this there are many proposed solutions with some drawbacks. In this paper, we explore one such solution to address the implementation issues of DWT.

1) Introduction
2) DWT Algorithm
3) New Approach
4) Handling Edge Cases
5) Conclusion
6) References

II. DWT Algorithm

One of the techniques to store the data in an Image while maintaining very little or no visible change in an Image is the LSB(Least Significant Technique) which stores the data in the last bit of an Image. But one can easily get the data just by looking into the image bits. This is solved by DWT using frequency coefficients. One of the DWT techniques is the Haar-DWT which is the simplest DWT.

A. HARR-DWT

In this method, 4 pixels are used to generate 4 coefficients. The 4 pixels are selected either by taking squares of 2X2 or in any other way such that each pixel is only taken once. The formulas that are used to calculate the coefficients d1, d2, d3, and d4 from the pixels p1, p2, p3, and p4 are as follows.

$d1 = (p1+p2+p3+p4)$
$d2 = (p1-p2+p3-p4)$
$d3 = (p1+p2+p3-p4)$
$d4 = (p1-p2-p3+p4)$

The process of calculating the coefficients is known as DWT. The data is embedded on these coefficients rather the pixels using techniques such as LSB. After embedding the data the coefficients are used to generate the pixel values to form the image this step is known as Inverse DWT. The new pixels P1, P2, P3, and P4 are calculated from the coefficients D1, D2, D3, and D4 using the formulas

$P1 = (D1+D2+D3+D4)/4$
$P2 = (D1-D2+D3-D4)/4$
$P3 = (D1+D2+D3-D4)/4$
$P4 = (D1-D2-D3+D4)/4$

Decoding is done by following the DWT step and extracting the coefficients from the embedded image and the required data is present in the LSB of the coefficients.

B. Problem with the above approach

A Black and white image pixel values range from 0 to 255 Integer values. As we can see in the above image on calculating the new pixels P1, P2, P3, and P4 there is a clear chance of getting a fraction value. If we ignore the fraction value and just store the decimal value we cannot get the same coefficients from the pixels when extracting the data. This leads to errors in the extracted data, so we need to store the fraction values. Another issue we have is the possibility of a negative value that cannot be stored in an image. Ignoring the sign will lead to false data in extracting process. There is also a chance of exceeding the limit of 255.

III. New Approach

A. Existing Approach

In order to solve the discussed issues there are many techniques and tricks such as storing the decimal values separately in the form of the description of the Image or dividing the pixel value by 2 since the range of possible values goes from -256 to 256 dividing by 2 doubles the range. But it affects the

ISSN 2250-3153

quality of the embedded image significantly. Storing decimal values required additional memory. In order to address the said issues effectively we propose the following solution.

B. New Approach

In this approach in order to solve the possibility of negative values we use a different set of formulas as follows.

d1 = p1 + p3 + p4 - 2*p2
d2 = p2 + p4 + p1 - 2*p3
d3 = p3 + p1 + p2 - 2*p4
d4 = p4 + p2 + p3 - 2*p1

The above set of formulae is used for calculating the coefficients. After embedding the data the following formulas perform Inverse DWT to generate an Image from new coefficients D1, D2, D3, and D4.

P1 = (D1 + D2 + D3) / 3
P2 = (D2 + D3 + D4) / 3
P3 = (D3 + D4 + D1) / 3
P4 = (D4 + D1 + D2) / 3

When performing LSB the pixel value is changed by at most 1 which is not visible to the naked eye. since the Inverse DWT formulas don't contain a negative sign we can safely rule out the possibility of a negative value. We still have the fraction values as we are dividing the sum by 3. To solve this issue we use a simple trick to store the fraction data within the pixel itself.

To solve the fraction issue we first divide the pixel value by 3 before calculating the coefficients. The division is a floor division. let p1, p2, p3, p4 be the pixels and np1, np2, np3, np4 be the pixel values after the division by 3. i.e np1=p1/3 and so on. Since we have pixels divided by 3 we can safely remove the division by 3 in the IDWT step. But we still need to calculate the coefficients while extracting the data which requires the pixel values to be the same when calculated by the IDWT formula. In order to save the fractional value we remove the division by 3 from IDWT and recalculate them to the divided version when calculating the coefficients.

1. Embedding step

np1 = ⌊p1/3⌋
np2 = ⌊p2/3⌋
np3 = ⌊p3/3⌋
np4 = ⌊p4/3⌋

d1 = np1 + np3 + np4 - 2*np2
d2 = np2 + np4 + np1 - 2*np3
d3 = np3 + np1 + np2 - 2*np4
d4 = np4 + np2 + np3 - 2*np1

P1 = (D1 + D2 + D3)
P2 = (D2 + D3 + D4)
P3 = (D3 + D4 + D1)
P4 = (D4 + D1 + D2)

2. Extracting step

nP1 = P1/3
nP2 = P2/3
nP3 = P3/3
nP4 = p4/3

d1 = np1 + np3 + np4 - 2*np2
d2 = np2 + np4 + np1 - 2*np3
d3 = np3 + np1 + np2 - 2*np4
d4 = np4 + np2 + np3 - 2*np1

Since the floor division by 3 only takes away of utmost 2 pixels it is negligible and not visible to the naked eye.

IV. Handling Edge Case

In the IDWT formulas, Since we are adding the values there might be a chance of overflow. This can be seen in the example when p1 = 255, p2 = 255, p3 = 255, and p4 = 0. Then the new pixels will be np1 = 85, np2 = 85, np3 = 85 and np4 = 85. Then the coefficients will be d1 = 0, d2 = 0, d3 = 255 and d4 = 0. When data is embedded using LSB, the coefficients might become d1 = 1, d2 = 1, d3 = 255, and d4 = 1. In such cases, the value exceeds the limit. To solve this issue we can do a border correction step which includes subtracting the new pixel values by 2 which only changes the value by at most 7 pixels which doesn't affect the quality of the embedded image.

V. CONCLUSION

A modified version of the DWT algorithm for steganography is proposed for resolving implementation issues and maintaining the newly produced image quality. Formulas are modified along with handling edge cases to store decimal values with out any additional storage. Implementation of the new approach is made easier by making sure the pixel values will be in the range of the required values.

Appendix

Appendixes, if needed, appear before the acknowledgment.

Acknowledgment

The preferred spelling of the word "acknowledgment" in American English is without an "e" after the "g." Use the singular heading even if you have many acknowledgments.

References

[1] G. O. Young, "Synthetic structure of industrial plastics (Book style with paper title and editor)," in *Plastics*, 2nd ed. vol. 3, J. Peters, Ed. New York: McGraw-Hill, 1964, pp. 15–64.
[2] W.-K. Chen, *Linear Networks and Systems* (Book style). Belmont, CA: Wadsworth, 1993, pp. 123–135.
[3] H. Poor, *An Introduction to Signal Detection and Estimation*. New York: Springer-Verlag, 1985, ch. 4.
[4] B. Smith, "An approach to graphs of linear forms (Unpublished work style)," unpublished.
[5] E. H. Miller, "A note on reflector arrays (Periodical style—Accepted for publication)," *IEEE Trans. Antennas Propagat.*, to be published.

ISSN 2250-3153

[6] J. Wang, "Fundamentals of erbium-doped fiber amplifiers arrays (Periodical style—Submitted for publication)," *IEEE J. Quantum Electron.*, submitted for publication.

AUTHORS

First Author – Author name, qualifications, associated institute (if any) and email address.
Second Author – Author name, qualifications, associated institute (if any) and email address.

Third Author – Author name, qualifications, associated institute (if any) and email address.

Correspondence Author – Author name, email address, alternate email address (if any), contact number.