Proceedings of the World Congress on Engineering and Computer Science 2012 Vol I
WCECS 2012, October 24-26, 2012, San Francisco, USA

Steganography using Odd-even Based Embedding and Compensation Procedure to Restore Histogram

Neeta Nain, Jaideep Singh, Ishan Dayma, Rajesh Meena

Abstract

We present a method to encode a message in a cover message, with odd-even based embedding in the quantized discrete cosine transform domain. Only some part of the image block (8 x 8) (after quantization) will be used for embedding the message bits. And out these only 30% or less coefficients will actually store the message. A random sequence will further be required to choose the 30% coefficients to be used to store the message bits using odd-even based embedding. While selecting coefficient's among low frequency luminance values, a range will be declared using a user provided threshold and a dynamic value obtained during course of the algorithm. Finally a compensation procedure is applied block wise to ensure that stego image histogram should remain close to the original. The distortion caused by embedding is somewhat dependent on how much maximum change is made per embedding by the steganography algorithm. In our experiments, we have used ± 1 embedding (a variant of odd-even based). So during the compensation step, the bin value is repaired using modification of immediate neighbors (immediate left or immediate right) of that bin, satisfying lowest mean square error due to compensation methods. The proposed method of compensation can easily be extended for ±k embedding or for that matter.

1. INTRODUCTION

Steganography is the art and science of communicating in such a way that the very existence of communication is not revealed to a third party. In order to communicate without being detected, the data-hider must obey following conditions.

(A) Perceptual constraint: The perceptual distortion[1] between the original and stego. (B) Image should not be more than a certain maximum amount, D_1, for some perceptual distance measure.

(C) Statistical constraint: The embedding process should not modify the statistics of the host signal more than a very small number, epsilon, for some statistical distance measure[3].

The authors are with Department of Computer Engineering, Malaviya National Institute of Technology, Jaipur, JLN Marg Jaipur- 302017, Rajasthan, India and Government Engineering College Jhalawar, India
(email: neetanain@yahoo.com)

The objective of this project is to investigate stenographic schemes that can provide provable security by achieving zero Kullback-Leibler divergence between the cover and the stego signal distributions, while communicating at high rates.

2. EXISTING TECHNIQUES

Embedding the message data directly into the spatial domain
means it is quite straightforward to detect that embedding has taken place. To counteract this, new methods were developed that embedded the message data in more inconspicuous areas - the most popular being the transform domain.

2.1 Image Processing: Transform Domain Techniques

JPEG compression is a commonly used method for reducing the file size of an image, without reducing the aesthetic qualities enough to become noticeable by the naked eye.. The compression of JPEG images contains several processes:
(A) Converting pixel values to YCbCr: The first step is to convert the RGB color layers of the image into three different components (Y, Cb, and Cr). (B) Down-sampling the chrominance values: The human eye is more sensitive to changes in brightness than to changes in color. This means that it is possible to remove a lot of color information from an image without losing a great deal quality. (C) Transforming values to frequencies: The Discrete Cosine Transform (DCT) is used for JPEG images to transform them into frequencies.

$$F(u,v) = \frac{1}{4}C(u)(v)\left[\sum_{x=0}^{7}\sum_{y=0}^{7} f(x,y)\ \cos\frac{(2x+1)u\pi}{16}\ \sin\frac{(2y+1)v\pi}{16}\right]$$

(1)

Where C(u) C(v)=$\frac{1}{\sqrt{2}}$ for u,v = 0;
C(u) C(v)=1 otherwise

ISBN: 978-988-19251-6-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

(D) Quantization: The aim is to quantize the values that represent the image obtained after above stage. The goal is to eliminate the high frequency (lower-right) values.

(E) Zig-Zag ordering: This stage of JPEG compression[2] reorders the values using a 'zig-zag' type motion so that similar frequencies are grouped together. (F) Lossless Compression: The last process involves the use of two different algorithms. 'Run-Length Encoding' (RLE) compresses the high frequency coefficients and a 'Differential Pulse Code Modulation' (DPCM) compresses the first low frequency coefficient. A Huffman algorithm is then used to compress everything. Finally, the Huffman trees are stored in the JPEG header

We summarize below a few of the existing stego techniques, the common feature among all is the usage of jpeg compression steps summarized above.

(A) JSteg**:** JSteg was developed by Derek Upham[8] and it is a transform domain stego-system that sequentially embeds the message bits in the LSB of the JPEG coefficients excluding 0 and 1. There is no key required, so anyone that knows that a stego is made using this system can extract the message.

(B) Outguess: Outguess 0.1 was developed by Niels Provos[9] and improves the embedding algorithm of JSteg by using a Pseudo Random Number Generator (PRNG) in order to get the coefficients randomly. The LSB of the selected nonzero non-one JPEG coefficient is embedded with the message bit. Outguess was improved with a second version (Outguess 0.2)[9] which preserves the first-order statistics of the image by making appropriate corrections after embedding.

3. PROPOSED TECHNIQUE

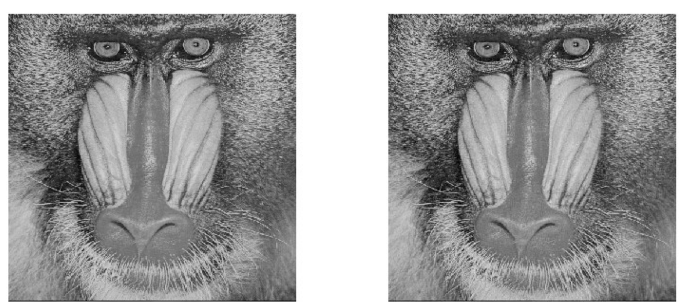
The algorithm we have proposed here is based on two new concepts; where to embed message bits in the coefficients: it makes use of odd-even based embedding and a 1st order restoration procedure to get the histogram back to the level of the histogram of original image. While choosing the coefficients for embedding three different constraints are being placed; one concerns with only using low frequency coefficient of luminance part of image; it makes use of pseudo random number generator (PRNG) in order to choose coefficients; percentage of coefficients to be actually used for hiding message bits and the numeric range of coefficients are required to be specified by the user. Finally a restoration process will be employed on the stego image to minimize the distortion caused by embedding or to achieve zero Kullback-Leibler[] divergence between the cover and the stego signal distributions, while communicating at high rates.

3.1. Statistical Restoration

We have used the principle of statistical restoration, where a certain fraction of the available coefficients are used for hiding while the rest is used to compensate for the changes in the host statistics due to hiding. By avoiding hiding in the low probability regions of the host distribution, we are able to achieve zero Kullback-Liebler[1] divergence between the cover and stego distributions, even while embedding at high rates. The proposed scheme is based on the idea of pixel swapping. The cover pixels are categorized into two streams, one is for embedding and another is for restoration. At the time of embedding value of a pixel (say α, from embedding stream) is changed to β. Now the idea is to find a pixel with value β in compensation stream and change it to α. The problem with this formulation is that at the time of embedding some pixels with value β may get changed to α.

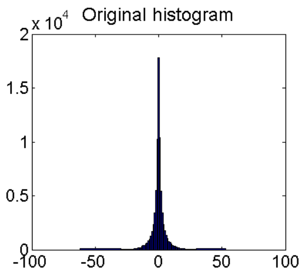
To overcome this problem, at the time of embedding we maintain a record of the pairs of pixel values which get changed into one another. So, after embedding we can get an exact count of the number of pixels which have to be compensated in order to maintain the first order statistics of the cover image. Next important point to be noted is measure of distortion added to the cover due to the compensation procedure. This distortion is somewhat dependent on how much maximum change is made per embedding by the stenographic algorithm. In our experiments, we have used the $\pm$ 1 embedding. So the absolute distortion per pixel due to embedding is at most 1. So during the compensation step, the bin value is repaired using modification of immediate neighbors (immediate left or immediate right) of that bin, satisfying lowest mean square error due to compensation methods. The proposed method of compensation can easily be extended for $\pm k$ embedding or for that matter any kind of embedding procedure either in the spatial or the transform domain. But the amount of noise added due to compensation will increase with the increase in the noise added during the embedding step.

Presented below are some sample results as shown in Figure 1 where we have applied the secure steganography algorithm on the baboon image. We have embedded 23300 bits in the 512 x 512 image, and have used 19 AC DCT coefficients per 8 x 8 blocks for hiding. The hiding fraction is 30% i.e., only 30% of total available coefficients are embedded with message bits; and out of them only the coefficients where magnitude is $\leq$ 30 qualifies for embedding.

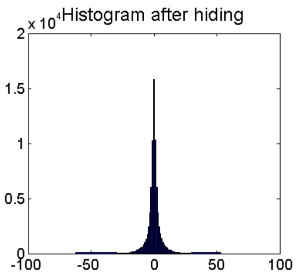


(a) Original image (b) Image hiding 23300bits

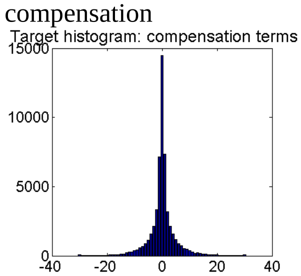
ISBN: 978-988-19251-6-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)



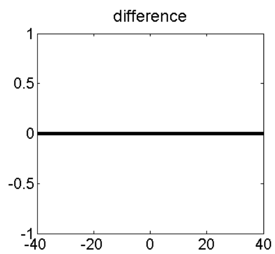
(c) Histogram of AC DCT DCT Coefficients available for hiding



(d) Histogram of AC coefficients after hiding but before compensation



(e)Desired histogram for the compensation coefficients, to ensure zero KL divergence



(d)Difference between the original and composite images, after compensation

Figure1: Histograms obtained of cover image before and after applying embedding algorithm and compensation procedure

3.2. Odd-Even Based Hiding Framework

The luminance part of the image is used for hiding, as this is the most important part of the image from the perspective of human eye and during jpeg compression this part of image is going to remain most intact. We divide the luminance image into 8×8 blocks, perform block-wise DCT, divide element-wise by a certain quality factor matrix and then select a certain frequency band for hiding. The DCT coefficients thus selected are rounded off to produce the quantized DCT (QDCT) based dataset X. For hiding, we use odd/even embedding[5] (a simple version of QIM) to convert the terms to their nearest odd or even integer, depending on whether the input bit is 1 or 0, respectively. Suppose, a QDCT term is 4 and we wish to embed 0 - then the QDCT term gets mapped to the nearest even number, which is 4. For embedding 1, we use a dither sequence, with numbers in the range [-0.5, 0.5] which are produced by a pseudorandom generator, to decide whether to map 4 to 3 or 5.

To embed 1→ q= round (p + 1 − mod (p – ₴, 2)); (2)

To embed 0 → q = round (p + 1 − mod (p + 1 −₴, 2)) (3)

where p, the original QDCT term, is mapped to q, ± ₴, denotes the corresponding number obtained from the dither sequence, “mod(p,2)” is the remainder obtained after dividing p by 2 and “round” denotes the rounding off operation. If p is an even(odd) number and 1(0) is to be embedded, it is mapped to (p - 1) or (p + 1) depending on whether ± belongs to the range (0,0.5] or [-0.5,0], respectively. Let¸ λ be the common hiding fraction for all bins. Let X(i) and Ĥ(i) denote the elements mapped to the i’th bins of X and Ĥ, respectively. Now, assuming an equal number of 0’s and 1’s in the input message that affects the elements in X(i), λ/4 fraction of coefficients from X(i) gets transferred to both Ĥ (i + 1) and Ĥ (I − 1). Also, ¸ 2 fraction of coefficients is moved to Ĥ(i). Explanation – let the value of the input QDCT coefficient be i, an even number, and if the input bit is 0, the output term, obtained using (3), is i itself. Since about half the bits in the input sequence are 0, about, 2 terms in X(i) are moved to Ĥ (i).If the input bit is 1, the output term gets mapped to the nearest odd number, which can be (i − 1) or (i +1), depending on whether the dither value (± in (2)) is positive or negative. By a similar logic, ¸ 4 fraction of terms from bins X(i −1) and X(i + 1) will be shifted to Ĥ(i). Thus, based on this analysis, the number of terms in Ĥ (i) is as follows:

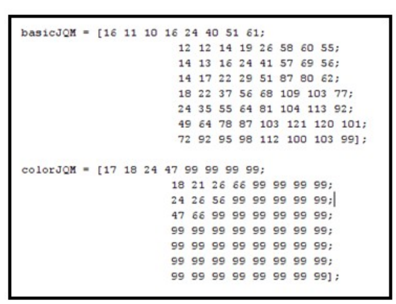
$$B_{\hat{H}}(i) \approx \frac{\lambda Bx(i)}{2} + \frac{\lambda Bx(i-1)}{4} + \frac{\lambda Bx(i+1)}{4}$$

To reiterate, the main assumptions behind this analysis are: the input message has equal number of 0’s and 1’s and the dither values are equally likely to be positive or negative. The assumptions are valid only if both the message and the dither sequence are long enough (minimum image size considered is 256X256)

3.3 Embedding Algorithm

1. As Inputs algorithm takes in the percentage of the of the coefficients to be embedded, quality constant to be used to decide the quality of image after quantization, threshold value to decide the range of coefficient to be used for embedding, number of coefficient per 8X8 block available for encoding.

Here quality constant will decide the matrix to be used during quantization and also the type of matrix will be decided at this step, from two different matrices available; one for color image and other for a gray scale image.



```
basicJQM = [16 11 10 16 24 40 51 61;
            12 12 14 19 26 58 60 55;
            14 13 16 24 41 57 69 56;
            14 17 22 29 51 87 80 62;
            18 22 37 56 68 109 103 77;
            24 35 55 64 81 104 113 92;
            49 64 78 87 103 121 120 101;
            72 92 95 98 112 100 103 99];

colorJQM = [17 18 24 47 99 99 99 99;
            18 21 26 66 99 99 99 99;
            24 26 56 99 99 99 99 99;
            47 66 99 99 99 99 99 99;
            99 99 99 99 99 99 99 99;
            99 99 99 99 99 99 99 99;
            99 99 99 99 99 99 99 99;
            99 99 99 99 99 99 99 99];
```

(a) Two available quantization matrices

ISBN: 978-988-19251-6-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

```
if quality <= 0 quality = 1;
 end
 if quality > 100 quality = 100;
 end

 if quality < 50
    scalefactor = 5000/quality;
 else
    scalefactor = 200 - quality*2;
 end

 if isColor == 0
    JQM = double(uint8((basicJQM*scalefactor+50)/100));
else
    JQM = double(uint8((colorJQM*scalefactor+50)/100));
end
```

(b) Code to select the quality factor

Figure 2: Quantization procedure

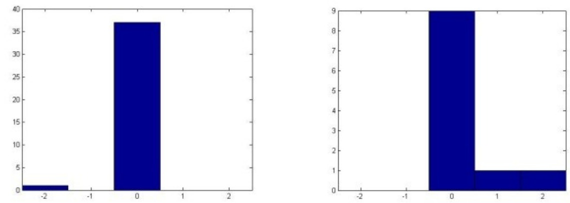
2. Thus we separate the luminance component of the image, as all of our embedding will be done only in luminance part. On dividing above luminance matrix into 8x8 blocks will give the coefficients available to act as holder of message bits, out of which 30% (or as per user input) will be actually be taken into consideration to hide message bits.
3. A matrix will be defined at this stage, let's say by the name 'used', which will exclude dc and higher frequency coefficients from the luminance 8x8 matrix.

```
usedt= used=
    0   1   1   1   1   1   0   0

    1   1   1   1   1   0   0   0

    1   1   1   1   0   0   0   0

    1   1   1   0   0   0   0   0

    1   1   0   0   0   0   0   0

    0   0   0   0   0   0   0   0

    0   0   0   0   0   0   0   0

    0   0   0   0   0   0   0   0
```

4. A pseudorandom number generator will again be used to randomize 'used' matrix across the 8x8 block.

```
sel =

  Columns 1 through 16

     0     1     1     0     0     0     0     0     0     1
     0     0     0     0     0     0

  Columns 17 through 32

     0     1     0     0     1     1     1     0     0     0
     1     0     0     0     0     1

  Columns 33 through 38

     0     0     0     1     0     1
```

Above figure shows the 'sel' matrix which in later part of the algorithm, along with 'used' matrix will select the coefficients for embedding.
5. Another constraint in selecting coefficients for embedding is that the coefficient should be in the user defined range, in our experiments we have taken it as <=30.
6. Finally the coefficients we are left with after applying above specified constraints will be used for embedding. The embedding will be odd-even based embedding technique as explained above.
7. Finally to make sure the stego image does not produce a visibly different histogram in comparison to the histogram of the original image, we require here 'Statistical Restoration method'. In stegnographic (part of) algorithm of our experiments, we have used the ± 1 embedding. So the absolute distortion per pixel due to embedding is at most 1
8. Compensation procedure: (a) here we will make use of the two list of coefficients we have maintained during the time of embedding, one list contains coefficients before embedding and other one contains coefficients after embedding. The histograms obtained from these two lists of coefficients are shown in Figure3.

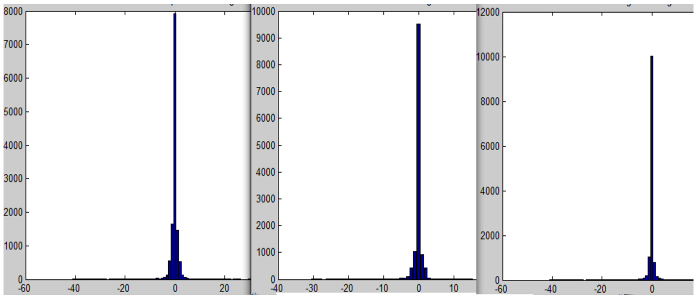


(a) Before embedding (b) After embedding

Figure3: Histogram of embeddable coefficients before and after embedding.

A difference of above two histograms will give us an idea of distortion incurred because of our embedding. Here we are also required to maintain a list of compensating coefficients (ones left after allocating the coefficients asked by the user for embedding) which will later be used for compensating the difference caused by embedding.
(c) A maximum and minimum limit of coefficients obtained using the available embeddable coefficients will give a range of coefficients for which compensation procedure will be applied and further processing will depend on the difference of number of each coefficients.
(d) While taking difference between above shown two histograms, the 'before embedding' histogram will be on the left side and the compensation procedure will be initiated only for the positive values.
(e) A replacement value (equal to (minimum coefficient – 1) initially) will be used for compensation, which will gradually increase as we move to next coefficient.
(d) Compare the current coefficient under consideration or the replacement value with a sorted list of coefficient available for compensation procedure, if the difference between does not exceed permissible limit do a replacement , here the number of replacements made in the compensation coefficient list will depend upon the difference in numbers of this individual coefficient before and after embedding.
(e) Above procedure will reintroduce these lost coefficients back into the blocks. The place occupied by these newly substituted coefficients will help us to reduce the increased number of coefficients by changing them to ones which were in majority earlier.

ISBN: 978-988-19251-6-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)



(a) Uncompensated histogram (b) final histogram (c) Original histogram

Figure 4: Above figures shows the histograms of the image at three different stages, comparing (c) and (a) we could see how decreased number of '0' coefficients are brought back close to normal by substituting compensation coefficients with '0' and how it helps to bring the tower of other distorted coefficients closed to original (as these were the substituted coefficients used for compensation).

4. EXPERIMENT AND RESULTS

The percent of coefficients can be successfully (success here obviously depends on the distortion in histogram being revert back to an acceptable level) used per block of the available coefficients for hiding greatly varies with the size of the image under embedding. For a jpg image of size: 256x173, 9KB it is as low as 3%, and for another jpg image (size: 512x512, 153KB) it goes as high as 40%. Table1 below shows few of the statistics obtained during experiments.

Table1: PSNR of the embedded image.

Image	Max. % Hidden able	Data embedded (in bits)	PSNR
256x173 , 9KB	4	509	42.074
512x512, 153KB	59	45801	26.422
512x512, 153KB	10	7764	27.502
512x512 , 145KB	45	38820	26.412
256x256, 68.7KB	38	7380	32.777
256X256, 62.1KB	35	6276	33.632
1280X960, 128KB	10	36315	35.609
1600X1200, 193KB	10	56786	37.508
2560X1600, 381KB	7	84868	40.200

The PSNR[6] is the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation. The signal in this case is the original data, and the noise is the error introduced by compression. Typical values for the PSNR in lossy image and video compression are between 30 and 50 dB, where higher is better. Through our algorithms we have achieved a PSNR value in the range as high as 40 – 42 compared to the average [6] PSNR value for jsteg which remains around 32[8].

5. CONCLUSION

Here, we have demonstrated a method to compute the maximum hiding fraction and hiding rate for odd-even based hiding for quantized DCT coefficients such that the hiding remains undetectable after first order statistical restoration. From a steganalyst's perspective, we have looked at first order histograms of individual frequency streams belonging to both original and the cover image. We have done a complete analysis, using diverse type of images as input to the proposed algorithm. The first order statistical restoration works satisfactorily only until it is fed with enough number of blocks, else the data embedding capacity on per block basis dramatically drops down. On the other hand if enough number of blocks are present then the percentage of data embedding capacity also increases instantly. An experimental description of the above conclusion can be inferred from Table-1.

References

[1] K. Solanki, K. Sullivan, U. Madhow, B. S. Manjunath and S. Chandrasekaran,"Statistical Restoration for Robust and Secure Stegnography" Proc. IEEE International Conference on Image Processing, Genova, Italy, Sep. 2005.

[2] Ricardo L. de Queiroz, Member, IEEE "Processing JPEG-Compressed Images and Documents"

[3] K. Solanki, K. Sullivan, U. Madhow, B. S. Manjunath and S. Chandrasekaran, "Probably Secure Steganography: Achieving Zero K-L Divergence using Statistical Restoration" Proc. IEEE International Conference on Image Processing 2006 (ICIP06), Atlanta, GA USA, Oct.2006.

[4] 'A. Sarkar, K. Sullivan and B. S. Manjunath, "Stegnography Capacity Estimation for the Statistical Proc, SPIE- Security, Stegnography, and Watermarking of Multimedia Contents (x), San Jose, California, Jan. 2008. Restoration Framework". Restoration of the Second Order Dependencies for Improved Security" Proc. 32nd IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Honolulu, Hawaii, Apr. 2007.

[5] A. Sarkar, K. Solanki, U. Madhow, S. Chandrasekaran and B. S. Manjunath "Secure Steganography: Statistical Estimating Steganographic Capacity for Odd-Even Based Embedding and its Use in Individual Compensation" Proc. IEEE International Conference on Image Processing (ICIP), San Antonio, TX, Sep. 2007.

[6] T. N. Pappas and R. J. Safranek, "Perceptual criteria for image quality evaluation," in Handbook of Image and Video Processing, A. C. Bovik, Ed. New York: Academic, May 2000.

[7] A. M. Eskicioglu and P. S. Fisher, "Image quality measures and their performance," IEEE Trans. Commun., vol. 43, pp. 2959–2965, Dec. 1995.

[8] S. K. Muttoo, Sushil Kumar Reader, Department of Computer Science, University of Delhi, India Reader, Rajdhani College, University of Delhi, New Delhi ,India "Data Hiding in JPEG Images".

[9] N Provos - Proc. of UM ACM Computer Security Seminar Series. is, 1999 "OutGuess-practical Steganography".

ISBN: 978-988-19251-6-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)