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, Dp, 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)
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[1] divergence between the cover and 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 ± 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 ±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 ≤ 30 qualifies for embedding.
where $p$, the original QDCT term, is mapped to $q$, $±\theta$, denotes the corresponding number obtained from the dither sequence, “$\text{mod}(p,2)$” 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 $p$ belongs to the range $(0,0.5]$ or $[-0.5,0]$, respectively. Let, $\lambda$ be the common hiding fraction for all bins. Let $X(i)$ and $\hat{X}(i)$ denote the elements mapped to the $i$'th bins of $X$ and $\hat{X}$, respectively. Now, assuming an equal number of 0’s and 1’s in the input message that affects the elements in $X(i)$, $\lambda/4$ fraction of coefficients from $X(i)$ gets transferred to both $\hat{X}(i + 1)$ and $\hat{X}(i - 1)$. Also, $\lambda/2$ fraction of coefficients is moved to $\hat{X}(i)$. 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 $\hat{X}(i)$. 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 $(\pm$ in (2)) is positive or negative. By a similar logic, $\lambda/4$ fraction of terms from bins $X(i - 1)$ and $X(i + 1)$ will be shifted to $\hat{X}(i)$. Thus, based on this analysis, the number of terms in $\hat{X}(i)$ is as follows:

$$\begin{align*}
\hat{X}(i) & \approx \frac{\lambda X(i)}{2} + \frac{\lambda X(i-1)}{4} + \frac{\lambda X(i+1)}{4}.
\end{align*}$$

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.

### 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.
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 Figure 3.

![Figure 3: Histogram of embeddable coefficients before and after embedding.](image)

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.

Below is the code to select the quality factor of the image.

```c
(b) Code to select the quality factor
```

```c
if quality == 0 quality = 1;
if quality > 100 quality = 100;
if quality < 50:
    quality = 50
else:
    quality = quality;
end
if quality = 0
    quality = quality + 100
end
```

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.

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

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 steganographic (part of) algorithm
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%. Table 1 below shows few of the statistics obtained during experiments.

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

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


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


[8] S. K. Muttoo, Sudhir Kumar Reader, Department of Computer Science, University of Delhi, India Reader, Rajhans College, University of Delhi, New Delhi, India “DataHiding in JPEG Images”.