ITERATIVE ALGORITHM FOR DATA HIDING IN IMAGES

Ankur Datta¹ John J. Leeson ¹

We propose an iterative algorithm for data hiding in images, which actively seeks to minimize the error due to hiding non-native data in images. The method is based on statistical analysis of images (least-cost random walk) and is very robust to changes due to file format conversions or changes due to application of a blurring filter like Gaussian.

1. Introduction

The goal of steganography is to hide the existence of secret communication between the two parties. The message to be transmitted has no relationship to the cover image and the only role the cover image plays is that of a foreign message carrier. The content of the cover image has no value to the sender or the decoder. Non-native data embedding always introduces distortion. Ideally the distortion produced should be as small as possible. Usually various statistical and human perception models are used to test the level of distortion present in an image.

2. Previous Work

This paper falls in the sub-category of cover-regions under steganography. There has been a lot of interesting work done recently in this sub-category. [2] describes the basic cover-region theory. Researchers in ‘Distortion-Free Data Embedding’ (‘DFDE’) [1] used a linear function for data embedding, whose special case is just plain, old LSB. Groups of pixels are selected sequentially and a linear function calculates the associations of the groups. Three types of groups are defined: R group, S group and U group. Bit 1 is assigned to the R group; bit 0 to the S group and the U group is left untouched. There are several problems with this approach. Natural images usually have smooth gradients, which means that sequentially selecting groups will lead to large accumulation in the mismatch between the message bits and the groups, leading to large changes in the image. A linear function also means that the change can be easily modeled and hence can be used by ‘steg-analytic’ techniques to detect such types of data embedding. Further the technique has an inherent assumption of bias towards the R groups in natural images, which leads to automatic errors in an equally probable random set of bit stream.

3. Our Algorithm

1. Take the Input image and calculate the functions $F1$ and $F2$ based on pre-defined function definitions.

¹ School of Electrical Engineering and Computer Science, University of Central Florida, 4000 Central Florida Blvd., Orlando, Florida, USA 32816. {ankur, leeson}@cs.ucf.edu
2. Calculate the association of block of pixels to either of the two groups (‘on’ or ‘off’).

3. A random walk is taken through the image, where the cost is calculated to embed the message. This walk is repeated iteratively for different seeds.

4. After the least-cost random walk over these iterations is found, the message is embedded in the image along with the seed by making changes to the non-conforming blocks.

5. The message on the other side is extracted by calculating the function association and the random walk through the transmitted seed.

![Algorithm for Data Embedding and Extraction](image)

### 4. Function and Pixel Association Calculation

After reading in an image in which the message is to be embedded, we calculate two pre-defined functions on the groups. A 'group' is a rectangular set of image pixels. A group can have varying sizes from 3X3 onwards to an entire image as one group, but it cannot vary within a random walk. We use $\omega$ to represent the area of the rectangle and assume $\omega$ is odd.

Our two functions are $F1$ and $F2$, where $F1$ and $F2$ are defined as follows for a group of $\omega$ pixels. We also define a binary group association function $\lambda$.

$$F1 = (\omega - 1) \cdot \text{Image} \left( \frac{\omega}{2} \right) \quad \text{and} \quad F2 = \sum_{i}^{\omega} \text{Image}(i) - \text{Image} \left( \frac{\omega}{2} \right)$$

$$\lambda = \begin{cases} 'On' & \text{if } F1 > F2 \\ 'Off' & \text{otherwise} \end{cases}$$

These definitions of $F1$ and $F2$ are not strict; in fact any $n^{th}$ power of this original definition can be used.

Based on the binary function, we obtain the association of each $\omega$ set of pixels or each group as being either on or off. On represents bit 1 and off represents bit 0.
5. Least Cost Random Walk

After finding the pixel associations from the previous step, we proceed to minimize the cost of embedding the foreign message into the image. We minimize the cost of embedding by minimizing the association of 'on' group with '0' bit and of 'off' groups with '1' bit. For this purpose we use a random number generator and use various seeds to come up with a random walk through the image and calculate the error function for each walk. The error function decides the number of iterations; if it’s decreasing at a fast pace then we keep iterating otherwise we stop. The least cost of embedding is the absolute minima over this error space.

\[
\delta = \sum_{i} \text{Group}_{original} - \text{Group}_{desired}
\]

Error function (3)

Where,

\[
\text{Group}_{desired} = \begin{cases} 
\text{'on'} & \text{if Group}_{original} = \text{'off'} \\
\text{'off'} & \text{if Group}_{original} = \text{'on'} 
\end{cases}
\]

(4)

![Fig. 2. Iteration Number Vs. Error function](image)

After finding the least cost random walk, we embed the message in the image. There will be instances where the current group association will not match with the bit of the message, in those situations; we will change the group pixel values according to the error function definition. Since the final random walk error function \( \delta \) is least out of all the iterations, the magnitude of image distortion would be very small. The seed is embedded at a predetermined place in the image, first 5 groups.

6. Extraction

The stego-image thus formed in the previous step, can be transmitted over the internet to the receiver and the message can be recovered on the other end by reversing the methodology just described above. The seed would be extracted first and then based on the seed the same random walk will be generated again and hence the message bits can be extracted by calculating the group association of the pixels in the image.
We define two error comparison statistics: MEE and SMEER. MEE stands for Message Embedding Error and SMEER stands for Signal to Message Embedding Error Ratio.

**Message Embedding Error (MEE)**

\[
\alpha = \sum_{i}^{row} \sum_{j}^{col} |Image_{original} - Image_{embedded}|
\]  

**Signal to Message Embedding Error Ratio (SMEER)**

\[
\beta = 10 \times \log_{10}(\frac{(255^2)}{MEE})
\]

### Table 1. Comparison between our approach and the ‘DFDE’ [1] approach

<table>
<thead>
<tr>
<th>Image Name</th>
<th>Message Size (3X3 group)</th>
<th>MEE -Iterative approach</th>
<th>MEE -‘DFDE’ approach</th>
<th>% in Error Reduction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blue_hills</td>
<td>1000</td>
<td>898</td>
<td>27790</td>
<td>96.7 %</td>
</tr>
<tr>
<td>Cliff_cloud</td>
<td>1000</td>
<td>3428</td>
<td>26973</td>
<td>87.2 %</td>
</tr>
<tr>
<td>Sand</td>
<td>1000</td>
<td>5663</td>
<td>25164</td>
<td>77.4 %</td>
</tr>
<tr>
<td>Iceberg</td>
<td>1000</td>
<td>2014</td>
<td>26703</td>
<td>92.4 %</td>
</tr>
</tbody>
</table>

Following are the comparison plots of our approach and the ‘DFDE’ approach.
8. Non-Linear Vs. Linear Modeling

Not only does our method achieve a very low-cost of embedding foreign data into the image, it's also harder to detect. This benefit comes from the fact that the original function model is non-linear. The changes introduced by this model are non-linear and hence are not easily noticeable or detectable and almost impossible to model.

9. Robustness

Our approach to data embedding is robust to transformations like image format conversion and blurring. We incorporate this robustness in the image by first modifying the binary decision function and then embedding the data in a regular way:

\[ \Omega = \begin{cases} 'On' & \text{if } F I - \mu > F 2 \\ 'Off' & \text{otherwise} \end{cases} \]  

(7)

By using this constant \( \mu \), we scale the image so that other non-linear transformations have little or no effect at all on the group associations.

10. Conclusion

In this paper, we presented a novel approach to data embedding in the images, which is both low-cost and hard to detect. We accomplish this by iteratively finding a least-cost random walk through the image and using a non-linear function to embed the message. Non-linearity assures that the function characteristics are hard to model and consequently are hard to detect.

11. REFERENCES
