A Novel Steganographic Method based on Edge Detection and Adaptive Multiple Bits Substitution

Keerthi K M
M.Tech Computer Science Student
Thejus Engineering College
Vellarakkad, Thrissur

ABSTRACT
This paper proposes a novel method for grayscale image steganography based on edge detection and adaptive multiple bits substitution. The pixels located in the edge regions usually present more random characteristics than the smooth regions. In proposed method, the Sobel operator is used to compute the gradient magnitude of the pixels of the cover image. As a result, all edges of the cover image, both horizontal and vertical, are fully detected. The sharper edges are adaptively preserved and the weaker edges are suppressed, according to the length of secret data. Therefore, the sharper edges will be used in advance of the weaker edges and the smooth regions for data embedding. Next, the data embedding route is determined using a pseudorandom number generator (PRNG) and multiple bits of secret data are adaptively embedded into *k*- LSBs of the pixels in the route. The value *k* depends on the gradient magnitude of each pixel .The larger the gradient magnitude, the larger the value *k*..

Keywords
The Sobel Operator, Adaptive LSB Substitution.

1. INTRODUCTION
In order to achieve safe and secret communication we need to hide our data from attackers. Data hiding techniques fall into 3 categories

- Cryptography
- Steganography
- Watermarking.

Steganography is the art of hiding information in ways that prevent the detection of hidden messages. Steganography, derived from Greek, literally means "covered writing." It includes a vast array of secret communications methods that conceal the message's very existence. These methods include invisible inks, microdots, character arrangement, digital signatures, covert channels, and spread spectrum communications. Components of Steganographic model are [1]

- Embedded data: message one wishes to secretly send.
- Cover Object: media being used to hide the data.
- StegoObject : product of using cover object to hide the embedded data.

Generally speaking, a steganographic technique is usually evaluated in two aspects [2]:

- Imperceptibility/Stego-image quality: How to preserve the details of the cover image when the secret message is being embedded in so that the differences between the stego-image and the cover image can be perfectly imperceptible to the human eye is the very first problem an ideal steganographic scheme has to face.
- Payload/Hiding capacity: The payload indicates the maximum number of bits that can be hidden with an acceptable resultant stego-image quality.

2. SURVEY
2.1Classification of Steganographic Systems
There are several Steganographic techniques for image file format which are as follows [1]:

I. Spatial domain technique

II. Masking and filtering

III. Transform techniques

IV. Distortion Techniques

2.1.1 Spatial Domain Technique:
There are many versions of spatial steganography, all directly change some bits in the image pixel values in hiding data. Least significant bit (LSB)-based steganography is one of the simplest techniques that hides a secret message in the LSBs of pixel values without introducing many perceptible distortions. To our human eye, changes in the value of the LSB are imperceptible. Embedding of message bits can be done either sequentially or randomly. Least Significant Bit (LSB) replacement, LSB matching, Matrix embedding and Pixel value, differencing are some of the spatial domain techniques.

2.1.2 Masking and Filtering
Before going to explain this technique let us explain two terms masking and filtering in simple words. Masking refers to the act of changing the colour of certain areas of a picture, or transferring these areas onto another background. Masking images entails changing the luminance of the masked area. The smaller the luminance change, the less of a chance that it can be detected. Filtering in image processing is a process that cleans up appearances and allows for selective highlighting of specific information. A filtered image is generated as the center of the mask moves to every pixel in the input image. Mask size determines the degree of smoothing and loss of detail. It transforms pixel intensity values to reveal certain image characteristics.

2.1.3 Transform *Domain Technique*
This is a more complex way of hiding information in an image. The process of embedding data in the frequency domain of a signal is much stronger than embedding principles that operate in the time domain.Embed secret information in a transform space of the signal.Most of the strong steganographic systems today operate within the

transform domain .This have an advantage over LSB techniques as they hide information in areas of the image that are less exposed to compression, cropping, and image processing. Some transform domain techniques do not seem dependent on the image format and they may outrun lossless and loss format conversions.

2.1.4 Distortion Techniques

Distortion techniques need knowledge of the original cover image during the decoding process where the decoder functions to check for differences between the original cover image and the distorted cover image in order to restore the secret message. The encoder adds a sequence of changes to the cover image. So, information is described as being stored by signal distortion .Using this technique, a stegoobject is created by applying a sequence of modifications to the cover image. This sequence of modifications is use to match the secret message required to transmit .The message is encoded at pseudo-randomly chosen pixels. If the stego-image is different from the cover image at the given message pixel, the message bit is a "1." otherwise, the message bit is a "0." The encoder can modify the "1" value pixels in such a manner that the statistical properties of the image are not affected. However, the need for sending the cover image limits the benefits of this technique. In any steganographic technique, the cover image should never be used more than once. If an attacker tampers with the stego-image by cropping, scaling or rotating, the receiver can easily detect it. In some cases, if the message is encoded with error correcting information, the change can even be reversed and the original message can be recovered.

2.2 Disadvantages of Existing Schemes

- All edges, both horizontal and vertical, is not fully exploited.
- Identical edges are not appearing in the cover image and the stego-image.
- The sharper edges have lower priority than the weaker edges to be used.
- The number of bits embedding in the sharper edges is lesser than the ones in the weaker edges.
- Only the secret data, not overhead data, should be embedded in the cover image.

3. STEGANOGRAPHIC METHOD BASED ON EDGE DETECTION AND ADAPTIVE MULTIPLE BITS SUBSTITUTION

This method [2] provides better visual quality than previous techniques as the edges are fully exploited,. The gray level of each pixel in a grayscale image is represented by one byte. Because the same edges should appear in the cover image and the stego-image, Mehran Iranpour [2] divide each pixel into two parts; one part contains *p* bits with lower significance (we call this part *p*-LSBs) and another part includes (8-*p*) bits with higher significance (we call this part (8-*p*)-MSBs), where the value *p* lies in the range [1, 5]. The *p*-LSBs are used for data embedding and the (8-*p*)-MSBs are used for edge detection. Because the (8-*p*)-MSBs do not contribute to data embedding, they remain unchanged and thus the edges appearing in the cover image and the stego-image are identical. Based on his[2] experiments, the maximum value of *p* is selected 5 since embedding more than 5 bits into each edge pixel provides considerable changes and bad visual quality in the stego-image.

3.1 Edge Detection

Edge detection consists of creating a binary image from a gray scale image where non-background pixel values correspond to object boundaries. Edges can be detected with the help of gradient/derivative type operators.The most common kernels used for the gradient edge detector are the Sobel, Roberts Cross and Prewitt operators. We use the Sobel operators for edge detection. Using the Sobel operators, all edges, both horizontal and vertical, will be fully detected. From Fig. 1, derivatives based on the Sobel operator masks for Z_5 are

$$G_x = (z_1 + 2z_2 + z_3) - (z_7 + 2z_8 + z_9)$$
$$G_y = (z_1 + 2z_4 + z_7) - (z_3 + 2z_6 + z_9) \quad (1)$$

Where the *z*'s are the gray levels of the pixels overlapped by the masks at any location in the cover image. After calculating the Gx and Gy for a pixel of the cover image, the gradient magnitude of pixel is computed by

$$g = \left\lfloor \frac{1}{6}(|G_x| + |G_y|) \right\rfloor \quad (2)$$

According to the fact that the number of bits embedding in a selected edge pixel should be dependent on its sharpness, we use *k* bits ($0 \le k \le p$), instead of exactly *p* bits, of *p*-LSBs of the edge pixel for data embedding. The value *k* depends on the sharpness of the edges, the gradient magnitude of pixels in other words, and is computed by

$$k = \left\lfloor \frac{g}{2^{8-p} - 1} * p \right\rfloor \quad (3)$$

3.2 Data Embedding

Embedding scheme consists of four main steps; edge detection, weak edges suppression, route traversing, and data hiding.

Step 1: Edge Detection

They use (8-*p*)-MSBs for edge detection, [2] they first shift the cover image to the right by *p* bits. Then using (2) compute the gradient magnitude of all pixels and create a matrix, named *S*, containing these values. It is obvious that the matrix *S* is not a binary matrix. This means we can distinguish between a sharp edge and a weak edge. We will use this fact in the following processes.

Step 2: Weak Edges Suppression

In paper [2] first compute the sharpness threshold *T*. Assume that the length of data is denoted as *m*. He extracts the non-zero values of matrix *S* and creates a vector, named *E*, containing these values. Because the sharper edges should be used first, he sorts the vector *E* in descending order. Assume that the value *k*, which computed by (3), for the i^{th} member of vector *E* is denoted as. We define the variables $n = 0$ and $i = 1$. In an iterative loop, we compute, add it to *n*, and compare *n* with *m*. If *n* is greater than or equal to *m*, we assign the i^{th} member of vector *E* to *T* and stop the loop. Otherwise we do $i = i + 1$ and repeat the loop. After computing the sharpness threshold T, the weak edges are suppressed. In other words, we assign zero to each member of matrix S which is less than the threshold T.

Step 3: Route Traversing

For determining the data embedding route, a pseudorandom number generator (PRNG) is used [2]. Because the sharper edges should have higher priority than the weaker edges and the smooth regions to be used, in this paper [3] first add the coordinates of the edge pixels to the route using the stream generated by PRNG. If the length of data is greater than the capacity of the edge regions in the cover image, he adds the coordinates of the remaining pixels (the pixels located in the weak edge regions or the smooth regions) in the stream to the route. By this strategy, the embedding capacity increases.

Step 4: Data Hiding

Once the embedding route is determined, for each pixel in the route, if the gradient magnitude of pixel in matrix *S* is not zero, *k* is computed by equation (3)**;** otherwise *k* is assigned by 1. Next, he embeds *k* bits of data into *k*-LSBs of the pixel. Finally, when data hiding is finished, he embed the required information for data extraction procedure, i.e. the length of data (*m*) and the sharpness threshold (*T*), into the LSBs of pixels in a predefined region in the stego-image which must not be used for data embedding. It can be easily extended to color images by embedding data into each color plane (R, G and B) independently. In such a way, the embedding capacity will increase.

3.4 Data Extraction

Data extraction procedure is the same as data embedding procedure. They first extract the required information for data extraction [2], i.e. the length of data (*m*) and the sharpness threshold (*T*), from the LSBs of pixels in the predefined region in the stego-image. Then he [3] shifts the stego-image to the right by *p* bits (*p* must be shared between sender and receiver) and compute the gradient magnitude of all pixels by (2) and creates the matrix *S*. Because the (8-*p*)-MSBs are used for edge detection, the matrix *S* obtained in this stage is identical with matrix *S* obtained in data embedding stage. Using the sharpness threshold *T*, the weak edges are suppressed. In other words, we assign zero to each member of matrix *S* which is less than *T*. Next, we determine the data extraction route using the PRNG employed in data embedding procedure and extract *k* bits of secret data, which *k* is computed by equation (3) for the edge pixels or k is assigned by 1 for the smooth regions, from k-LSBs of each pixel in the route until we reach the length of data, i.e. m.

4. SIMULATION MODEL

In this section, he [2] presents some experimental results to show the effectiveness of this proposed method [2]. Some reference images (e.g. Lena,) shows the edges detected using the Sobel operators on Lena. Although the value *p* is almost a large number (*p*=4), many edges have been detected. As seen in Fig. 7(b), the visual quality of the cover image shifted to the right by *p*=4 bits is not too bad and this image is suitable for edge detection. The horizontal edges (matrix G_x), the vertical edges (matrix G_y), and the gradient magnitude of pixels (matrix *S*) are also shown in below Figure .

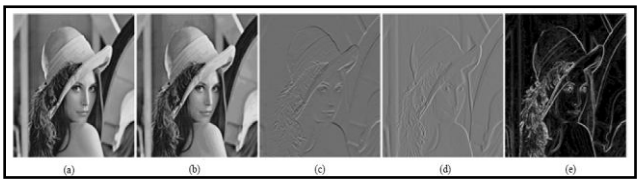

Figure 1 Cover image Right shifted Horizontal Vertical MatrixS

Edges edges p=4bits

These images show that all edges, both horizontal and vertical, are fully exploited in our proposed method. Below figure shows the shifted cover images and the edges (the gradient magnitude of pixels) for some values of *p* on Lena

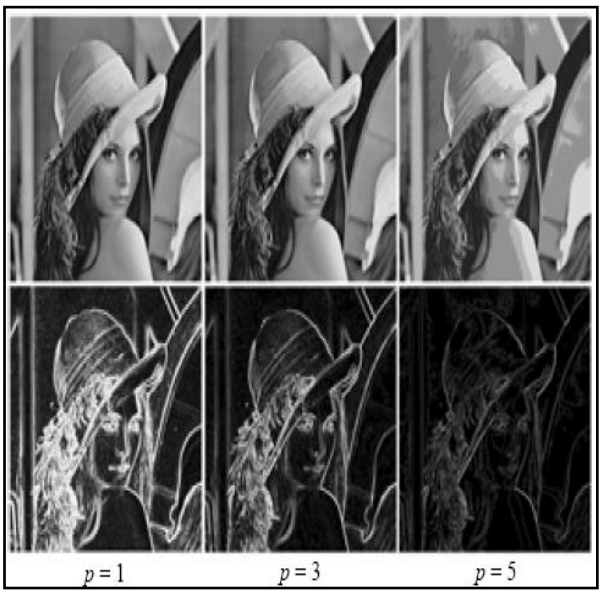

Figure 2 Shifted cover images & edges for some different values of p

As pointed, (8-*p*)-MSBs is used for edge detection. When the value *p* increases, the number of bits of each pixel which is lost during the shift operation will increase. Therefore, the lower bits of each pixel will contribute to edge detection. As a result, the less but sharper edges will appear. The reason that the sharper edges will appear is the fact that the difference between two higher significant bits is more than two lower significant bits. This fact can be seen in edge images in above Fig.As described in weak edges suppression step in data embedding procedure, we select the edges used for embedding according to the length of secret data.

Below Figure depicts the difference between the cover image and the stego-images for some different values of *p* and various payloads (bit per pixel) on Lena.

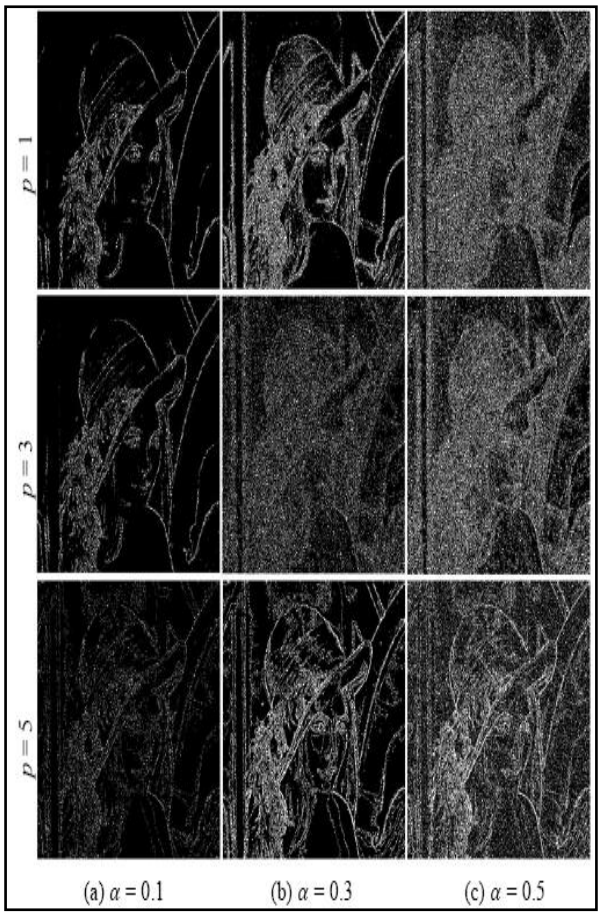

Figure 3 Difference between cover images and stego images for different values of P and various payloads α on Lena

For the lower payload, only the sharper edges were used. By increasing the length of data, more edges & the smooth regions have been used. Next figure shows the 5 stego-images of Lena for *p*=1, 2,3,4,5

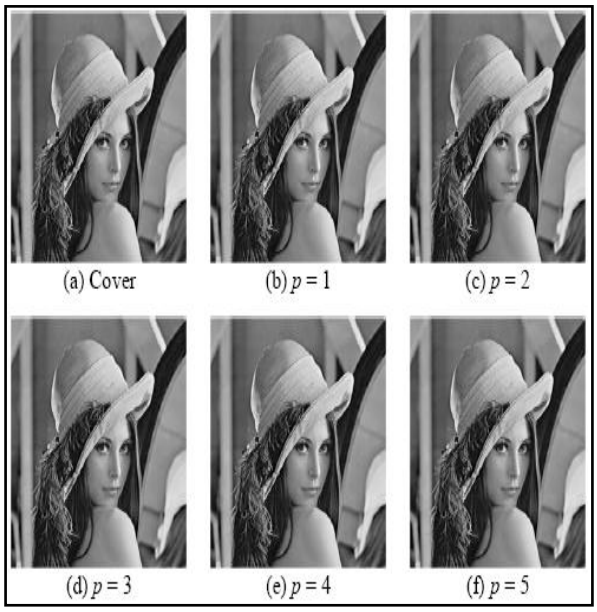

Figure 4 (a) Cover image (b-f) Stego images for some different values of p with maximum capacity

For each value *p*, we embedded the maximum capacity of data to show the maximum distortion. Obviously, the stegoimages created by the proposed method [2] have excellent visual quality.

5. CONCLUSION

The proposed method is an adaptive multiple bits substitution steganographic method for hiding data in the edge regions of grayscale images. The regions located in the edges present more complicated statistical features and thus it is more difficult to observe changes in the edges than those in smooth regions. Because the edges are fully exploited, the proposed method provides better visual quality than previous techniques. We use some bits of pixels for data embedding and the remaining bits for edge detection. By this strategy, the same edges appear in the cover image and the stego-image. Find a sharpness threshold according to the length of secret data, and suppress the weak edges using this threshold.

Next, determine the data embedding route using a PRNG and adaptively embed multiple bits of secret data into the LSBs of the pixels in the route. No overhead data are produced in this method. The experimental results on 8000 images demonstrate the effectiveness of proposed method in resistance to RS steganalysis with better visual quality and higher embedding capacity.

6. REFERENCES

[1] Pratap Chandra Mandal, "Modern Steganographic technique: A survey" IJCSET, 2012.

[2] Mehran Iranpour., "A Novel Steganographic Method Based on Edge Detection and Adaptive Multiple Bits Substitution", IEEE transaction 2013.

[3] G.T. Shrivakshan, Dr. C. Chandrasekar "A Comparison of various Edge Detection Techniques used in Image Processing" IJCSI, 2012.

[4] Sneha Arora, Sanyam Anand "A Proposed Method for Image Steganography Using Edge Detection" International Journal of Engineering Sciences, 2013.

[5] Junji Shikata, Tsutomu Matsumoto, "Unconditionally Secure Steganography against Active Attacks" IEEE transactions on information theory, vol. 54, no. 6, June 2008.

[6] Nitin Jain, Sachin Meshram, Shikha Dubey "Image Steganography Using LSB and Edge – Detection Technique" International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-2, Issue-3, July 2012.

[7] AlaaA.Jabbar, Altaay, Melaka, ShahrinbinSahib, Mazdak Zamani, "An Introduction to Image Steganography Techniques", 2012 International Conference on Advanced Computer Science Applications and Technologies.

[8] Babloo Saha and Shuchi Sharma, "Steganographic Techniques of Data Hiding using Digital Images", Defence Science Journal, Vol. 62, No. 1, January 2012, pp. 11-18,2012.