Information Hiding Using Edge Boundaries of Objects

Mehdi Hussain, M. Hussain
Dept. of Computer Science SZABIST
Islamabad, Pakistan.
mehdi141@hotmail.com, mhussain@szabist-isb.edu.pk

Abstract

Information hiding has importance in information security. Generally information hiding is used to protect, authenticate data or for secret communication such as military secret information. It can be achieved in communication protocols, digital multimedia content such as image, text and audio. Currently available information hiding techniques do not pay much attention on stego-object (hidden information in original object) with respect to its original cover (original) object. Both cover and stego-images are drifted in context of edge characteristics of image objects. That is not affordable in some application such as medical domains applications. This paper proposes an information hiding method around the edge boundary of objects in image. The stego-image can be utilized for further processing methods (such as segmentation, identification, and recognition of objects etc). The experimental results show that the stego-image has identical edge boundaries as is in cover-image (using 'Sobel' and 'Canny' edge detection methods), so stego-image can directly be used instead of cover-image for further image processing techniques. Proposed edge based information hiding scheme is secure due to its varying calculated thresholds and has less computational cost.

Keywords: *Information embedding; Edge based data embedding; LSB based embedding; edge boundary data embedding*

1. Introduction

In Information security, data hiding has received a significant attention both in industry and academia. It becomes very common into digital media, such as communication, images, videos, audios, etc. due to tremendous growth of internet. This secret information is used for authentication, bank transactions, credit cards codes, multimedia content copyrights, secret military information, feature tagging (timestamp or control information of application), or important messages passing depends on applications. Two parameters, undetectablilty and embedding capacity have major importance in information hiding. Generally, cover object has to sacrifice its originality due to hide secret information. The distortion due to data hiding is not affordable in some applications like medical images, visual artifacts measuring application etc. We have proposed an edge boundary based information hiding method and highly perceptual transparency as compared with original cover image. For edge computation we explored and used two types of well known edge detection methods, 'Sobel' and 'Canny' with their general/default thresholds parameters. Proposed technique has identical edges of objects in stego and cover-image. Through experimental results, proposed technique has high perceptual transparency with low computational complexity. Stego-image can further be used as an original image for application (segmentation, feature extraction, identification of objects). For example, proposed scheme can be used to store patient data in their medical

images, where it doesn't modify the edges of objects and further image can be used for diagnosis purpose. Its information hiding capacity can easily be increased depending on computed threshold. Thresholds may vary depending on the visual characteristics of image. Moreover, extraction of the secret information is independent of original cover image.

This paper presents the basics of image based steganography. In section 2 has literature review of currently available data hiding methods with their strengths and weakness. In section 3 we have reviewed recent edge based data hiding methods. In section 4 we have described our proposed, pure edge boundary based data embedding method. In section 5 have experimental results of hidden information with different thresholds, its PSNR, MSE and RMSE. In section 6 include the conclusion and future work.

Image Based Steganography

Image steganography contains the following terminology.

- Cover-Image: Original image which is used as a carrier for hidden information.
- Message: Actual information which is used to hide into images. Message could be a plain text or some other image.
- Stego-Image: After embedding message into cover image is known as stego-image.
- Stego-Key: A key is used for embedding or extracting the messages from cover-images and stego-images.

Generally image steganography is method of information hiding into cover-image and generates a stego-image. This stego-image then sent to the other party by known medium, where the third party does not know that this stego-image has hidden message. After receiving stego-image hidden message can simply be extracted with or without stego-key (depending on embedding algorithm) by the receiving end [1]. Basic diagram of image steganography is shown in Fig. 1 without stego-key, where embedding algorithm required a cover image with message for embedding procedure. Output of embedding algorithm is a stego image which simply sent to extracting algorithm, where extracted algorithm unhides the message from stego-image.

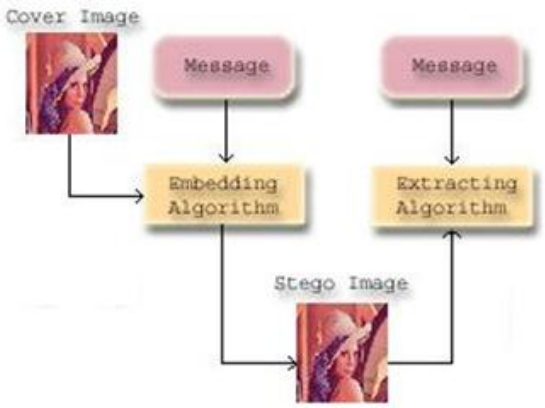


Fig. 1. Image Steganography Overview

2. Literature Review

In [2, 3 and 4] authors have introduced the Lease Significant Bit (LSB) methods. Where up to certain number of least significant bits of each pixel is over write or modify with hidden information bits. The above procedure is also required to unhide the hidden messages from the stego-images. It could be a very risky for information hiding because 'sequential scanning' based techniques [5] can easily be recovered the hidden messages with considering the neighboring pixel intensity variation. To resolve the above problem another method of embedding hidden bits using the random selection of pixels in an image. In this case it required a stego-key used to hide and recover the secret message bits from pixels. So according to [5], it has key management overhead. Synchronization or updating phase of stego-key for both sender and receiver has a major concern. Another method of Stego Color Cycle (SCC) systematic orderly selection of pixels channel Red Green Blue (RGB) of cover–image used to embed the message bits. According to [5] if 'sequential scanning' of pixel can identify some pixels hidden data then all remaining hidden data can easily detectable. In [6] authors introduced a pixel indicator technique, where for embedding data bits, select one channel from RGB and modify the data bits up to 2 LSB. But the selection criteria of method are sequential. Hidden capacity of this method totally depends on the cover image channel bits. In [7] introduced an enhanced scheme of the above method. It utilizes the variable number of bits of selected pixel channel (of RGB) for embedding. It increases the capacity of the scheme presented in [6]. In [8], authors have introduced a data hiding technique where it finds out the dark area of the image and used LSB to embed secret data in it. It converts dark area to binary image and labels each object using 8 pixel connectivity schemes for hiding data bits. In [9] method, it takes the difference of two consecutive pixels of cover-image for computing the size of the hidden data bits. In [10] histogram analysis method hacked the above [9] Pixel Value Differencing (PVD) method. This [9] approach can provide high embedding capacity. To best of our knowledge the authors of above papers have shown the range of PSNR of their stego-image around 50%, but their embedding capacities are high against our proposed method.

3. Edge Based Scheme

Generally, edge based data hiding schemes utilize the edges as well as smooth region of the cover image to store hidden information. These schemes have good perceptual transparency (depending on the embedding capacity) and with good human visual quality. But as comparison with the cover image to stego-image has much variation. Peak Signal to Noise Ratio (PSNR) difference with respect to cover image is too high. In [11], author introduced a high capacity of hidden data utilizing the LSB and hybrid edge detection schemes. Edge detection is based on two types of canny and fuzzy edges detectors. Based on these edge computations it used the LSB substitution to embed the hidden data. To achieve the perceptual transparency [12] has proposed a method for data embedding, where it used the edge pixel itself and its neighboring pixels for hidden data. This method utilized the feature of high variation of pixel intensity area (edges) to store data bits. It just modified the LSBs while keeping the unchanged most significant bits at sharper edges. It is an edge adaptive case of LSB replacement so cover and stego-images have much statistical difference. Modification of edge pixels of stego-image object may not regenerate those identical edges as in cover-image, because stego-image edge pixels are drifted in context of original edges of cover-image

objects. For further image processing techniques (segmentation, identification or etc... which based on edge computations) results are different in cover and stego-image. In [13] author have introduced a new adaptive data hiding method that used the edge area with k- LSB method and the smooth area of the image with PVD method. So [13] provides both larger capacity and high visual quality. On the other side in [14] method takes advantage over [13] method because it has almost same capacity and visual statistical results but with less computational complexity. All of the above edge based data hiding methods modify the actual edges of objects and its stego-image is not able to regenerate the original edges as in cover-image. Our proposed method embeds the data around the edge boundaries of objects to consider retaining the edge boundaries identical as in cover-image.

4. Proposed Method

Proposed method utilizes the edge boundaries of an image as embedded data. For edge computation we used both 'Sobel' and 'Canny' edge detector methods with their default thresholds parameters. Following embedding procedure is tested with both sobel and canny edge detectors and their results discussed in experimental sections. Currently, we fix the horizontal edges direction for data embedding in experimental results. EDGE_LEN denotes the length of any edge. DIF_THRESH denotes the difference threshold value of two pixels.

First, compute the edges of the image using canny edge detector. Now find out the horizontal edges up to certain number of EDGE_LEN from canny edge detected image. Now compute the absolute difference of edge (pixel) with its upper (row-1) boundary pixel that should be greater than DIF_THRESH. If DIF_THRESH condition satisfied then LSBs substitution is used to embed the hidden data bit into the upper boundary of the edge pixel. (DIF_THRESH conditions are available in pseudo code of the embedding procedure and table-1 also shows the embedding cases and its conditions.) After embedding the hidden data bits compute again the canny edge detection of stego-image. If the edges of objects in stego-image is not identical with original cover-image objects edges then update the DIF_THRESH value (with predefine values). Now repeat the above whole procedure, until the stego-image edges are identical to the cover-edge image. Complete procedure can be summarized with following steps edge computation, edge length finding, embedding data and matching the cover-image edges with stego-image edges. Finally, we have a stego-image (having hidden bits, around the horizontal edge of the image). Both EDGE_LEN and DIF_THRESH may be embedded, or known by both parties.

Interpretation of DIF_THRESH condition from table-1 is as follows. Upper Pixel denotes the pixel of above (row-1) current edge pixel. Data Bit denotes the message bit to hide. Edge Pixel denotes the current edge pixel value. 'x' denotes the don't care value. From table-1, first row of table-1 shows that if Upper Pixel value is even and Data Bit is '0', then don't modify the Upper Pixel. Second row of table-1, if Upper Pixel value is odd and Data Bit is '0', so have to check first condition (Upper Pixel < Edge Pixel) if satisfied then subtract -1 from the Upper Pixel value of current Edge Pixel, otherwise add +1 into Upper Pixel. Further table-1 rows are also interpreted like above.

Now in extracting phase, take the Sobel/Canny mask filter of stego-image and generate the stego-edge binary image. Find the stego-image edges, where edge length should be greater or equal to EDGE_LEN. If this condition is satisfied then we take the absolute difference of all upper pixels (of edge) with edge pixels itself e.g.

$$\text{Diff} = | \text{Upper Pixel} - \text{Edge Pixel} | \qquad (1)$$

TABLE 1 EMBEDDING CASES

Upper Pixel	Data Bit	Upper Pixel < Edge Pixel	Upper Pixel > Edge Pixel
Even	0	x	x
Odd	0	-1	+1
Even	1	-1	+1
Odd	1	x	x

x: do not change. *Upper Pixel*: Upper pixel of edge. *-1*: Subtract -1 from upper pixel. *+1*: Add +1 in upper pixel.

If 'Diff' value is greater than DIF_THRESH then simply take a least significant bit of upper pixel as an extracted bit. It would be 0 or 1. Otherwise skip upper pixel and move to next upper pixel for 'Diff' calculation (1). Repeat the extracting process until reach to EDGE_LEN. Next find other edge area where stego-image edge pixels are equal or greater than EDGE_LEN, and start again extracting procedure for this edge of stego-image. The pseudo code of embedding and extracting procedure is as follows:-

4.1 Embedding Procedure

Step 1: Compute Sobel/Canny mask filter of cover-image as CoverImgEdge binary image.

Step 2: Copy the cover-image into the stego-image.

Step 3: Find the horizontal edge length equal to EDGE_LEN in CoverImgEdge binary image.

Step 4: IF (Upper pixel value of horizontal edge is LESS than edge pixel value) AND (upper pixel do not belongs to any other edge pixels) AND (Difference of upper pixels and edge pixel values are GREATER than DIF_THRESH)

THEN

IF (Upper pixel == Odd AND Hidden Bit == 0)

Subtract 1 from stego-image upper pixel.

Else IF (Upper pixel == Even AND Hidden Bit ==1)

Subtract 1 from stego-image upper pixel.

Else

Do not update the value of upper pixel.

END

END

IF (Upper pixel value of horizontal edge is GREATER than edge pixel value) AND (Upper pixel do not belong to edge pixel) AND (Difference of upper pixel and edge pixel values are LESS than DIF_THRESH)

THEN

IF (Upper pixel == Odd AND Hidden Bit == 0).

Add 1 in stego-image upper pixel

Else IF (Upper pixel == Even AND Hidden Bit ==1)

Add 1 in stego-image upper pixel

Else

Do not update the value.

END

END

Step 5: Now take again Sobel/Canny mask filter of Stego-image as StegoImgEdge binary image.

IF (StegoImgEdge NOT Equal CoverImgEdge Images)

THEN

Update the DIF_THRESH + constant and repeat from step 2:

END

Step 6: Stego-image contain the hidden bits of message.

4.2 Extracting Procedure

Step 1: Compute Sobel/Canny mask filter of stego-image as StegoImgEdge binary image.

Step2: Find the pixels of horizontal edge length equal to EDGE_LEN in StegoImgEdge binary image.

Step 3: Compute the Diff (equation 1) value from the edge pixel of stego-image.

Step 4: IF 'Diff' greater or equal to DIF_THRESH than take a least significant bit of upper pixel of stego-image. Repeats step 3 until edge reach to the EDGE_LEN.

Step 5: Repeat step 2 for all edges of StegoImgEdge binary image and store the least significant bits to a buffer which indicate the uncover message.

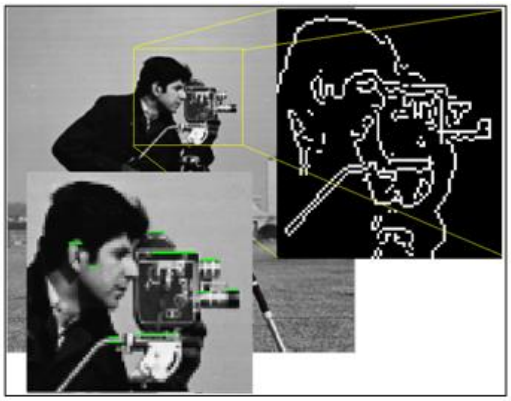


Fig. 2. Shows Hidden Areas Around Edge Boundaries

Every image computes its own DIF_THRESH depending on its texture during data embedding phase. It is strength of this proposed technique; it increases the complexity to

uncover the hidden information. Both cover and stego images have same edges characteristic even after embedding hidden data into stego-image. Advantage of this scheme it can iteratively repeat data embedding until fulfill the required identical edges of stego-image with cover-image, just modifying its thresholds (EDGE_LEN, DIF_THRESH). Fig. 2 shows the zoom view of cameraman image. It identifies area of edges and boundaries of edges with hidden data. Green color shows the hidden data area around the boundaries of edges.

5. Experimental Results

The experimental results presented in this section describe the performance of our proposed technique. For edge detection we use two canny and sobel well known edge detection methods. To conduct our experiments, we have tested our scheme over more then 50 standard images of different resolutions including some of them with 256x256 grayscale images, "cameraman", ''Tiffany", ''Lena", ''Baboon" and etc. These test images are shown in table-2.

Generally, stego-image quality is considered from two aspects. First, we use the Peak Signal-to-Noise Ratio (PSNR) measurement to evaluate the difference between the stego and cover images. Second, we compare the quality of the stego image with the cover image as seen by the Human Visual System (HVS). Mean Square Error (MSE) is between the cover and stego images. For a cover image width and height are m and n, where I denote the cover-image and K denotes the stego-image MSE is defined as:

$$MSE = \frac{1}{m\,n}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j) - K(i,j)]^2 \quad (2)$$

The general PSNR formula is defined as:

$$PSNR = 10 \cdot \log_{10}\left(\frac{MAX_I^2}{MSE}\right) \quad (3)$$

The maximum value of a pixel in grayscale image is 255. A higher PSNR indicates that the quality of the stego image is batter and more similar to the cover image. Table-3 shows the results of proposed method with canny edge detector method. It shows the different images with their DIF_THRESH, EDGE_LEN, Covert Bits and PSNR parameters. For example 'Cameraman' stego-image hides 798 bits of hidden data and has DIFF_THRESH 48, EDGE_LEN 4 and its PSNR is 78%, where table-5 shows that both cover and stego-images have identical canny edges of different images. Table-4 shows the proposed method results with sobel edge detector with all above parameters. Through experimental results table-3 and 4 shows that proposed method with canny edge detection has quite good embedding capacity. Our proposed method is more useful in those applications which use canny based scheme (for further processing).

6. Conclusion

This paper has introduced a method of image steganography, hiding data into image while retaining image objects edges as in cover-image. This technique is targeting to generate a stego-image which can be further used as cover-image in different applications (segmentation of objects or etc). For edge detection most common canny and sobel methods are used. Proposed method embeds the data bits to the edge boundary of stego-image objects, while both cover and stego-image have identical edges, because its edge pixel values are not

modified. Proposed technique is hard to recover the hidden message due to its threshold which varies depending on image itself. Future works can improve its capacity, and multiple edge direction (vertical, diagonal or etc) can be further explored to hide data with high PSNR.

TABLE 2 IMAGE DATA SET FOR EXPERIMENTS

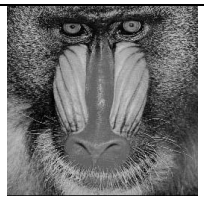
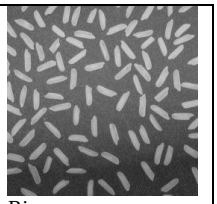
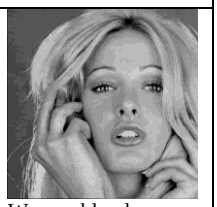
 Cameraman	 Baboon	 Rice
 Lena	 Woman	 Woman blonde
 Crowd	 Pirate	 Living room

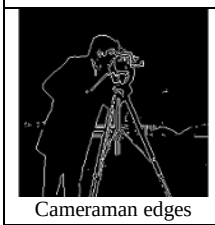
TABLE 3 CANNY EDGE RESULTS WITH PROPOSED METHOD

Image	DIF_THRESH	EDGE_LEN	Covert Bits	PSNR
Cameraman	48	4	798	78
Baboon	96	4	749	86
Rice	32	4	363	76
Lena	16	4	156	80
Woman	48	4	1005	85
Woman blonde	96	4	1187	84
Crowd	128	4	1089	82
Pirate	112	4	1299	97
Living room	112	4	3822	89

TABLE 4 SOBEL EDGE RESULTS WITH PROPOSED METHOD

Image	DIF_THRESH	EDGE_LEN	Covert Bits	PSNR
Cameraman	16	4	240	78
Baboon	80	4	151	81
Rice	32	4	384	75
Lena	16	4	58	83
Woman	80	4	312	96
Woman blonde	128	4	293	90
Crowd	64	4	487	82
Pirate	112	4	374	92
Living room	64	4	1755	80

TABLE 5 CANNY EDGE IMAGES

 Cameraman	 Lena	 Woman blonde
 Cameraman edges	 Lena edges	 Woman blonde edge

References

[1] N. Johnson and S. Jajodia, “Exploring steganography: seeing the unseen”, IEEE Computer, 1998, pp. 26-34.
[2] G.C. Kessler, "An Overview of Steganography for the Computer Forensics Examiner", Forensic Science Communications, Vol. 6, No. 3, July 2004.
[3] D. Artz, "Digital Steganography: Hiding Data within Data", IEEE Internet Computing: Spotlight, pages 75-80, May-June 2001.
[4] K. Bailey, K. Curran, "An Evaluation of Image Based Steganography Methods", Multimedia Tools & Applications, Vol. 30, No. 1, pages 55-88, July 2006.
[5] S. Venkatraman, A. Abraham, M. Paprzycki, "Significance of Steganography on Data Security", International Conference on Information Technology: Coding and Computing (ITCC'04), 5-7 April 2004.
[6] Adnan Gutub, Mahmoud Ankeer, Muhammad Abu- Ghalioun, Abdulrahman Shaheen, and Aleem Alvi, “Pixel indicator high capacity technique for RGB image based Steganography”, WoSPA 2008 – 5th IEEE International Workshop on Signal Processing and its Applications, University of Sharjah, Sharjah, U.A.E. 18 – 20 March 2008.
[7] Mohammad Tanvir Parvez, Adnan Abdul-Aziz Gutub, "RGB Intensity Based Variable-Bits Image Steganography," apscc, pp.1322-1327, 2008 IEEE Asia-Pacific Services Computing Conference, 2008.
[8] H.Motameni, M.Norouzi, M.Jahandar and A.Hatami, "Labeling Method in Steganography," World Academy of Science, Engineering and Technology, France. 2007.
[9] D. C. Wu and W. H. Tsai, “A steganographic method for images by pixel-value differencing", Pattern Recognition Letters, vol. 24, no. 9-10, pp. 1613-1626, 2003.
[10] H.C. Wu, N.I Wu, C.S. Tsai and M.S. Hwang, “Image Steganographic scheme based on pixel-value differencing and LSB replacement methods”,VISP(152), No. 5, October 2005.
[11] Chen, W. J., Chang, C. C. and Le, T. H. N., (2010): "High Payload Steganography Mechanism Using Hybrid Edge Detector," Expert Systems with Applications (ESWA 2010), Vol. 37, No. 4, Apr. 2010, pp. 3292-3301.
[12] Kathryn Hempstalk, "Hiding Behind Corners: Using Edges in Images for Better Steganography", Proceedings of the Computing Women's Congress, Hamilton, New Zealand, 11- 19 February 2006.
[13] Cheng-Hsing Yang, Chi-Yao Weng, Shiuh-Jeng Wang, Member, IEEE, and Hung-Min Sun, “Adaptive Data Hiding in Edge Areas of Images with Spatial LSB Domain Systems”, IEEE Transactions on Information Forensics and Security, VOL. 3, NO. 3, pp. 488-497, September 2008.
[14] Hussain, M. Hussain, M, “Pixel intensity based high capacity data embedding method”, IEEE International Conference Information and Emerging Technologies (ICIET), Pakistan, June 2010.

Author

Mehdi Hussain completed B.S (Computer Science) from Islamia University of Bahawalpur and currently doing M.S (Computer Science) at SZABIST Islamabad (Pakistan), also working as senior software engineer in Private Software House. Research areas of interest are Image steganography, Image Compression, Information Security and so on.