Hiding of Confidential Data in Spatial Domain Images using Image Interpolation

S. Maria Celestin Vigila[1] and K. Muneeswaran[2]
(Corresponding author: S. Maria Celestin Vigila)

Department of Information Technology, Noorul Islam University[1]
Kumaracoil - 629 180, Tamilnadu, India
Department of Computer Science and Engineering, Mepco Schlenk Engineering College[2]
Sivakasi - 626 005, Tamilnadu, India
(Email:{celesleon, kmuni12}@yahoo.com)

(Received May 28, 2013; revised and accepted Feb. 10 & May 20, 2014)

Abstract

Data hiding is a technique that is used to embed secret data into a cover media. This paper presents the implementation of reversible data hiding in spatial domain images based on neighbor mean image interpolation without impairing the image quality. Here, the cover image and secret bits are extracted from stego image without the need for any additional information. The strength of the proposed method has lower computational complexity, less blurring and greater image resolution. This paper also discusses the performance aspects of the proposed method which is superior in terms of high data embedding capacity and image quality.

Keywords: Image interpolation, payload, reversible data hiding

1 Introduction

In the modern era of high speed internet, multimedia data are represented in digital forms to be transmitted through internet. Since digital media is easily replicated and subject to tampering, protecting content is an important issue. Data hiding schemes have been widely used to protect the content of digital media. Data hiding schemes generally embed important data into the cover media by modifying the pixels to protect the data from illegal peeking or damaging. The concept of data hiding was first suggested by Simmons in 1983 [11]. Data can be hidden in lots of ways. In order to hide secret data, straight data insertion may encode every bit of data in the cover media or it may selectively embed data in noisy areas that describe a smaller amount of attention. Data may also be scattered erratically throughout the cover media. There are a number of ways to hide secret data; the most common methods are the Least Significant Bit (LSB) insertion, masking, filtering and transformations [7].

Reversible data hiding methods enable the exact recovery of the original cover image from the stego image without any distortion upon extraction of the embedded information [6]. Image interpolation tackles the difficulty of generating high resolution images from its low resolution image. The model that is employed to illustrate the relationship between high resolution and low resolution pixels plays a crucial role in the performance of an interpolation algorithm. Conventional linear interpolation methods, derived from space-invariant models, are unsuccessful to capture the hasty evolving statistics around edges and annoying artifacts. Linear interpolation [8] is commonly chosen for computational simplicity not for performance. Hence, reversible data hiding is incredibly important for securing sensitive data in image applications.

With this goal, the implementation of reversible data hiding in spatial domain images based on neighbor mean image interpolation without impairing the image quality is presented in this paper. The cover image and secret text data are extracted from stego image without the need for any additional information. The proposed method can embed a huge amount of secret data whereas maintaining an elevated image quality that of the other data hiding methods.

This paper is organized as follows. Section 2 reviews related works in the area of data hiding. In Section 3, the details of the proposed data embedding method is described. In Section 4, the experimental results are presented and discussed. Finally, the concluding remarks are presented in Section 5.

2 Related Works

In this section, some highlights of the relevant work in the area of data hiding are outlined. A simple and most instinctive scheme for hiding secret data into a digital image is to directly substitute the LSBs of each pixel in the cover image with the bits of secret data [17]. Wang et al. [16] described an optimal re-naming problem for the hidden secret data, and then applied a genetic algorithm for seeking the problem's nearly optimal solution. Thien and Lin [13] suggested a digit by digit data hiding scheme based on modulus function.

Ni et al. [9] proposed a histogram based data hiding scheme. In their scheme, the search for the pair of peak and zero points from the histogram is performed first. The peak point refers to the most often occurring pixel value in the histogram. The zero point stands for the pixel value with zero or minimal occurrences in the histogram. The secret data are embedded by shifting the pixel values located between the peak point and the zero point. Huang et al. [5] proposed a histogram-based scheme which used a multilevel hiding strategy to obtain high capacity and low distortion.

Most of the work on reversible data hiding focuses on the data embedding/extracting on the plain spatial domain [1, 3, 4, 14]. Tsai et al. [15] developed a reversible data embedding method that merges predictive coding and histogram shifting techniques. Chang et al. [2] suggested a reversible data embedding method to embed secret data in original images based on the edge directed prediction scheme. In this scheme, an embedded pixel value is simplifying along with a predetermined threshold and the difference between the predicted pixel value and its original pixel value.

Peng et al. [10] proposed a reversible data embedding algorithm based on integer transform and adaptive embedding. In this algorithm, the parameter is adaptively chosen in dissimilar blocks in integer transform. In [12], Tai et al. presented a reversible data hiding scheme based on histogram modification using pairs of peak points. Owing to these existing works on data hiding and its popularity, it is proposed to hide the data in spatial domain images using neighbor mean image interpolation.

3 Proposed Method

The proposed method is made up of interpolation, data embedding and data extraction phases. Initially, the proposed method introduces a scaling up and an interpolation technique. The scaling up image focuses on high speed and low complexity that is used as a cover image. In the data embedding phase, the secret data S is taken as input and then, it converts into binary secret bits. After the conversion process, then these secret bits are embedded into the cover image C, and then transmit the stego image to a receiver without impairing the image quality.

At receiving side, the cover image and secret bits are extracted from stego image without the need for any additional information. Then these secret bits are converted into text data. Figure 1 shows the sketch of the proposed data embedding method.

3.1 Interpolation Phase

In the Neighbor Mean Interpolation (NMI) phase uses neighboring pixel values to compute the mean, and after that the computed mean value is introduced into a pixel that has not been assigned yet. In general, high resolution pixels are getting when a neighboring pixel values are referenced in order to calculate a value that is to be assigned, but time complexity is higher when the number of referenced pixel is higher. The scaling up method decides what application to which it should be applied.

$$p'(x,y) = \begin{cases} p(x,y) & if\ x = k.i, y = k.j \\ [p(x,y-1)+p(x,y+1)]/k & \\ & if\ x = k.i, y = k.j+1 \\ [p(x-1,y)+(p'(x+1,y)]/k & \\ & if\ x = k.i+1, y = k.j \\ [p(x-1,y-1)+p'(x-1,y)+ & \\ \quad p'(x,y-1)]/k+1) & otherwise \end{cases} \tag{1}$$

Assume that $p(x,y)$ represents the value of a pixel located at (x,y) in original image and its interpolated pixel in cover image $p'(x,y)$ is computed as Equation (1), where $0 \le y \le x$ and $i,\ j = 0, 1, \cdots, 127$. k stands for a value of scaling up coefficient. In the scaling up process, the cover image is scaled two times more. Therefore, the value of k is 2 for which a cover image can preserve an elevated resolution.

For the pixel $p'(0,0)$ and $p'(2,2)$ are the same value with $p(0,0)$ and $p(2,2)$, respectively. In the case of $x < y$, $p'(0,1)$ is computed as $(p(0,0) + p(0,2))/2$ operation. When $x < y$, $p'(1,0)$ is calculated as $(p(0,0) + p(2,0))/2$. Finally, $p'(1,1)$ is obtained from $(p(0,0) + p'(0,1) + p'(1,0))/3$.

The proposed neighbor mean interpolation method is more efficient, less blurring and greater image resolution than nearest neighbor interpolation and bilinear interpolation methods. So, the resulting image that is produced by the NMI method is used with a cover image.

3.2 Data Embedding Phase

In the data embedding phase, the sequence of data embedding can be in zig-zag, left-to-right and top to bottom directions. Before secret data are embedded, the cover image is partitioned into four-pixel, non-overlapping, consecutive by zig-zag scanning. For every four non-overlapping consecutive pixel values, i.e., $p(x,y)$, $p(x,y+1)$, $p(x+1,y)$ and $p(x+1,y+1)$, the corresponding stego-image pixel values are $p'(x,y)$, $p'(x,y+1)$, $p'(x+1,y)$, and $p'(x+1,y+1)$, respectively. Here, secret data are embedded into three pixels except for $p(x,y)$ pixel. The details of each step are specified as follows.

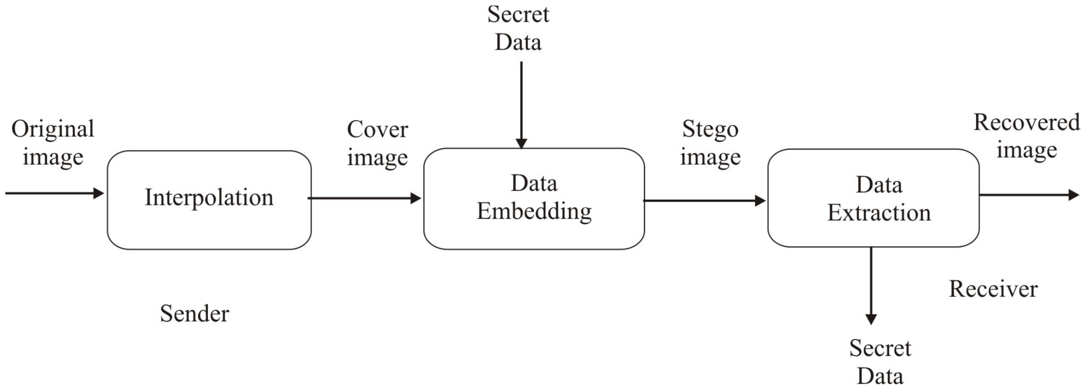


Figure 1: Sketch of proposed data hiding method

1) For every four non-overlapping consecutive pixel values, a difference value d is computed as Equation (2), where $0 \leq x, y \leq 127$ and β, α value is 0 or 1, respectively. When $\beta = \alpha = 0$, the difference value d as zero.

$$d = p(k.x + \beta, k.y + \alpha) - p(k.x, k.y). \tag{2}$$

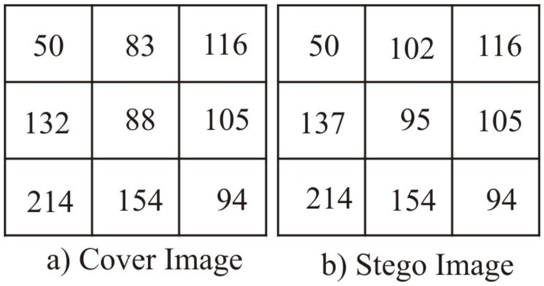
2) Calculate the number of bits, say n, which can be embedded in this pixel is

$$n = \log_2 |d|. \tag{3}$$

3) Select first n bits from the secret message S that is substream $S(n)$ and it is converted to integer value b.

4) Then, a stego image pixel $p'(x, y)$ is computed as follows.

$$p'(x, y) = p(x, y) + b. \tag{4}$$



50	83	116
132	88	105
214	154	94

a) Cover Image

50	102	116
137	95	105
214	154	94

b) Stego Image

Figure 2: Example results of the data embedding procedure

The following serves as a detailed example to depict the data embedding procedure. Figure 2(a) shows a 3∗3 cover image, and Figure 2(b) shows its stego image. Let secret message $S =$ "1001100010100111". In Figure 2(a), pixel $p(0, 0) = 50$ is the starting point for zig-zag scanning. In the proposed method, pixel $p(0, 0)$ is retained. Next, consider the pixel $p(0, 1)$ as 83. First the difference value d is computed by using Equation (2) as $d = 83 - 50 = 33$. Second, by Equation (3) calculate the number of bits embedded in $p(0, 1)$ as $\log_2 \mid d \mid = 5$. Third, select first 5 bits from the secret message S and convert it into integer value $b = 10011_2 = 19$. Therefore, the pixel of stego-image is $p'(0, 1) = 83 + 19 = 102$. Similarly, other pixel values $p(1, 0)$ and $p(1, 1)$ are calculated and tabulated in Table 1.

Table 1: Example details of the data embedding procedure

Coordinate	$p(x, y)$	d	n	S(n)	b	$p'(x, y)$
(0, 1)	83	33	5	10011	19	102
(1, 0)	132	82	6	000101	5	137
(1, 1)	88	38	5	00111	7	95

3.3 Data Extraction Phase

In the data extraction phase, the cover image and secret bits are obtained by using stego image only without need for any further information. The cover image and the secret data are extracted by using the following step by step procedure.

1) Compute the pixel in cover image $p(x, y)$ from the stego image $p'(x, y)$ using a simple arithmetic expression as defined by Equation (5), where $0 \leq y \leq x$ and $i, j = 0, 1, \cdots, 127$ and k is defined to 2.

$$p(x, y) = \begin{cases} [p'(x, y) + p'(x, y)] \\ \qquad if\ x = k.i, y = k.j \\ [p'(x, y) + p'(x, y + 1)]/k \\ \qquad if\ x = k.i, y = k.j + 1 \\ [p'(x, y) + (p'(x + 1, y)]/k \\ \qquad if\ x = k.i + 1, y = k.j \\ [k.p'(x, y) + p'(x, y + 2)/k + \\ \quad p'(x + 2, y)/k]/(k + 1) \\ \qquad otherwise \end{cases} \tag{5}$$

2) Then, the secret data embedding in $p(x, y)$ is computed as

$$b = p'(x, y) - p(x, y). \quad (6)$$

3) Next, calculate the difference value d using Equation (2) for the three neighboring pixels, excluding $p(0, 0)$; this is because $p'(0, 0)$ does not include secret bits.

4) Finally, the length of hiding is calculated for each secret bit. The integer value b is represented as secret bits based on the hiding length.

Table 2: Example details of the data extraction procedure

Coordinate	$p'(x, y)$	$p(x, y)$	b	d	n	S(n)
(0,1)	102	83	19	33	5	10011
(1,0)	137	132	5	82	6	101
(1,1)	95	88	7	38	5	111

Table 2 shows an example of the data extraction procedure. For example, $p(0, 1)$ is calculated by $(p'(0, 0) + p'(0, 2))/2 = (50 + 116)/2 = 83$ and the secret data embedding in $p(0, 1)$ is $102 - 83 = 19$. $p(1, 0)$ is also calculated by $(p'(0, 0) + p'(2, 0))/2 = (50 + 214)/2 = 132$ and the secret data embedding in $p(1, 0)$ is $137 - 132 = 5$. Finally, $p(1, 1)$ is obtained by $[2 * p'(0, 0) + p'(0, 2)/2 + p'(2, 0)/2]/3 = (2 * 50 + 116/2 + 214/2)/3 = 265/3 = 88$ and secret data embedding in $p'(1, 1)$ is $95 - 88 = 7$. Then calculate the difference value d as 33, 82 and 38 which correspond to $p'(0, 1)$, $p'(1, 0)$ and $p'(1, 1)$. Finally, based on the hiding length n, the integer value b is represented as secret bits as $19 = 10011_2$, $5 = 000101_2$ and $7 = 00111_2$. As a result, the cover image and secret data are extracted from the stego image only.

4 Results and Discussion

In this paper, four standard 512 × 512 gray scale images Lena, Pepper, Airplane and Baboon are considered as cover image for data embedding phase as shown in Figure 3. To evaluate the performances of the proposed method, two indicators have been considered.

Foremost, the histogram of the cover image and stego image are compared. An image histogram shows how pixels in an image are distributed by plotting the number of pixels at each intensity level. The histogram of the stego image is slightly varies from its cover image. So, the image distortion gets reduced in this proposed method. Due to this reason, the hacker cannot find the hidden data, because stego image looks same as original image but the secret data is embedded in stego image. From this, the proposed method is more efficient to hide the secret data by verifies the histogram of the original and stego images as shown in Figure 4.

Second, the Peak Signal-to-Noise Ratio (PSNR) is evaluated to compare visual quality between the cover image and the stego-image. The PSNR value can be computed by the following equation:

$$PSNR(dB) = 10 \times \log \frac{(2^n - 1)^2}{MSE}. \quad (7)$$

Where n represents the number of bits per pixel, and MSE (Mean Square Error) can be computed as follows:

$$MSE = \frac{1}{u \times v} \sum_{0}^{u-1} \sum_{0}^{v-1} (E_{xy} - C_{xy})^2. \quad (8)$$

The parameters u and v represent the height and width of the image. The notations C_{xy} and E_{xy} represent the cover image and the stego image pixel value in position (x, y) respectively. If the distortion between the cover image and the stego image is small, the PSNR value is large. Thus, a larger PSNR value means that the quality of the stego image is better than the smaller one. Generally, with human vision alone, it is hard to differentiate a stego image from its original image when the PSNR value is greater than 30 dB.

Table 3 gives the values of PSNR calculated for the four cover images with different payload size are stated by the formula specified in Equation (7).

Figure 3: The Four Cover Images

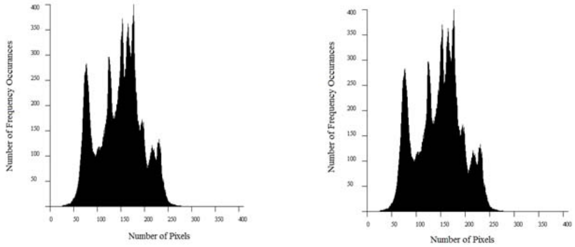


Figure 4: Histogram of original image and stego image

Here we include the results for the 512 × 512, 8 bits per pixel (bpp) grayscale "Lena". The embedded payload size

Table 3: PSNR values for the four cover images with different payload size

Cover image	Payload Size (bits)					
(512 x 512)	2520	36216	42264	56840	71400	85960
Lena	66.7	61.93	54.19	51.56	49.94	48.76
Peppers	55.29	54.44	53.23	51.02	49.55	48.46
Airplane	63.28	59.54	52.82	50.76	49.37	48.32
Baboon	56.96	55.76	51.7	50.03	48.83	47.89

and its PSNR of embedded ”Lena” images are specified in Table 4. It is also very apparent that the payload size increases, the PSNR value decreases. But the image quality is still preserved.

Table 4: Embedded payload size vs. PSNR of embedded ”Lena” image

Payload Size (bits)	2520	36216	42264	56840	71400	85960
PSNR (dB)	66.7	61.93	54.19	51.56	49.94	48.76

Based on these aspects, it is observed that the proposed method is good in terms of lower computational complexity, less blurring, high data embedding capacity and greater image quality.

5 Conclusion

The proposed data hiding method is a good candidate for providing security over confidential data. Since this method uses neighbor mean interpolation for generating the cover image, so it provides less blurring and degradation in cover image. And also it uses reversible data hiding method, so that original image can be extracted without impairing the image quality. Due to these reasons, hacker can’t predict the hidden data. Hence, it is robust method for providing data security across networks. The histogram of original and stego images are compared and it is clear that there is very less degradation between the original and stego image. The performance and quality of the cover image and the stego image are computed and analyzed in terms of PSNR value. With some performance improvement in the computations it can be made useful in real time day to day transaction security related with e-commerce and other online transaction.

Acknowledgments

The authors are grateful to the principal and management of Noorul Islam University and Mepco Schlenk Engineering College for extending all the facilities and constant encouragement for carrying out this research work. The authors gratefully acknowledge the anonymous reviewers for their valuable comments.

References

[1] M. U. Celik, G. Sharma, A. M. Tekalp, and E. Saber, “Lossless generalized LSB data embedding”, *IEEE Transactions on Image Processing*, vol. 14, no. 2, pp. 253–266, 2005.

[2] C. C. Chang, C. C. Lin, and Y. H. Chen, “Reversible data embedding scheme using differences between original and predicted pixel values”, *IET Information Security*, vol. 2, no. 2, pp. 35–46, 2008.

[3] S. F. Chiou, I. E. Liao, M. S. Hwang, “A capacity-enhanced reversible data hiding scheme based on SMVQ”, *Imaging Science Journal*, vol. 59, no. 1, pp. 17–24, 2011.

[4] W. Hong, T. S. Chen., Y. P. Chang, and C. W. Shiu, “A high capacity reversible data hiding scheme using orthogonal projection and prediction error modification”, *Signal Processing*, vol. 90, no. 11, pp. 2911–2922, 2010.

[5] L. C. Huang, L. Y. Tseng, and M. S. Hwang, “The study on data hiding in medical images”, *International Journal of Network Security*, vol. 14, no. 6, pp. 301–309, 2012.

[6] L. C. Huang, L. Y. Tseng, M. S. Hwang, “A reversible data hiding method by histogram shifting in high quality medical images”, *Journal of Systems and Software*, vol. 86, no. 3, pp. 716–727, 2013.

[7] N. F. Johnson, S. Jajodia, “Exploring steganography: seeing the unseen”, *IEEE Computing*, vol. 31, no. 2, pp. 26–34, 1998.

[8] X. Li, M. T. Orchard, “New edge-directed interpolation”, *IEEE Transactions on Image Processing*, vol. 10, no. 10, pp. 1521–1527, 2001.

[9] Z. Ni, Y. Q. Shi, N. Ansari, and W. Su, “Reversible data hiding”, *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 3, pp. 354–362, 2006.

[10] F. Peng, L. Xiaolong, and B. Yang, “Adaptive reversible data hiding scheme based on integer transform”, *Signal Processing*, vol. 92, no. 1, pp. 54–62, 2012.

[11] G. J. Simmons, “The prisoners’ problem and the subliminal channel”, in *CRYPTO’83*, pp. 51–67, 1983.

[12] W. L. Tai, C. M. Yeh, and C. C. Chang, “Reversible data hiding based on histogram modification of pixel differences”, *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 6, pp. 906–910, 2009.

[13] C. C. Thien, J. C. Lin, “A simple and high-hiding capacity method for hiding digit-by-digit data in images based on modulus function”, *Pattern Recognition*, vol. 36, pp. 2875–2881, 2003.

[14] J. Tian, “Reversible data embedding using a difference expansion”, *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 8, pp. 890–896, 2003.

[15] P. Tsai, Y. C. Hu, and H. L. Yeh, "Reversible image hiding scheme using predictive coding and histogram shifting", *Signal Processing*, vol. 89, no. 6, pp.1129–1143, 2009.

[16] R. Z. Wang, C. F. Lin, and J. C. Lin, "Image hiding by optimal LSB substitution and genetic algorithm", *Pattern Recognition*, vol. 34, no. 3, pp. 671–683, 2001.

[17] N. I. Wu and M. S. Hwang, "Data hiding: Current status and key issues", *International Journal of Network Security*, vol. 4, no. 1, pp. 1–9, 2007.

S. Maria Celestin Vigila completed her B.E. in Computer Science and Engineering in 1996 and M.E. in Computer Science and Engineering in 1999. She completed her Ph.D. in the area of data security from Anna University, Chennai. She is currently Associate Professor in the Department of Information Technology, Noorul Islam University, Kumaracoil and member of ISTE and IET. She is the reviewer for quite a few peer reviewed international journals. Her research interest includes cryptography and network security, wireless networks and information hiding.

K. Muneeswaran is Professor and Head in the Department of Computer Science and Engineering, Mepco Schlenk Engineering College, Sivakasi. His area of interest includes image analysis, computer networks, neural networks, security, grid and cloud computing. He contributed to many funded research projects. Also, he is the reviewer for quite a few peer reviewed international journals.