Exploring LSB Substitution and Pixel-value Differencing for Block-based Adaptive Data Hiding

Yuan-Yu Tsai, Jian-Ting Chen, and Chi-Shiang Chan
(*Corresponding author*: *Chi-Shiang Chan*)

Department of Applied Informatics and Multimedia, Asia University,
No. 500, Lioufeng Rd., Wufeng Dist., Taichung City 41354, Taiwan
(Email: CSChan@asia.edu.tw)

(Received Sep. 30, 2013; revised and accepted Oct. 6, 2013)

Abstract

Khodaei and Faez proposed a new adaptive data hiding technique based on LSB substitution and pixel-value differencing. Their algorithm can embed a large amount of secret data while maintaining acceptable image quality. However, their proposed algorithm only has fixed embedding capacity. In addition, the derivation for three consecutive pixels in the boundary region is poorly manipulated using raster scan order, resulting in inaccurate pixel differences. Finally, an overflow problem may occur for some embedding cases. In this study, we adopt non-overlapping blocks with *m*-by-*n* pixels to address the above problems. The cover image is first partitioned into non-overlapping blocks. The LSB substitution and optimal pixel adjustment process are then employed to embed the secret message into the central pixel of each block. The residual pixels within the same block are with message embedded using a pixel-value differencing scheme. The experimental results show that our proposed algorithm can achieve an adjustable embedding capacity according to the block size. The proposed technique is feasible in adaptive data hiding.

Keywords: Pixel-value Differencing, LSB Substitution, Adaptive Data Hiding, Optimal Pixel Adjustment Process, Block

1 Introduction

Most data hiding schemes [2] embed the same amount of a secret message into each embeddable pixel of the cover images. However, the above action is not wise because not all pixels can accept the same amount of distortion. Generally, pixels located in a complex area can tolerate more distortion than those located in a smooth area. The amount of embedded message for each pixel should be determined by its surrounding image complexity [7, 9].

Pixel-value differencing (PVD) [9], proposed by Wu and Tsai, is an adaptive image data hiding algorithm in the spatial domain. The main concept of PVD is to utilize the difference of two consecutive pixels to decide the amount of secret message. More secret message is embedded into the larger pixel difference because these two pixels are supposed to locate on a complex area. Although Wu and Tsai's algorithm can achieve adaptive data hiding, the limited capacity is an urgent problem. As a result, many improvements [1, 3, 4, 5, 6, 8, 10, 11] are proposed. Among them, integrating LSB substitution into PVD is an effective solution.

To improve embedding capacity, Wu et al. [10] adopted a LSB substitution scheme on the pixel pair located in the smooth area. They first use the difference between two consecutive pixels to obtain their located image complexity. Instead of the PVD scheme, 3-LSB substitution is employed to hide secret message into the pixel pair located in a smooth area. An additional adjusting operation is then performed while the data-embedded pixel pair belongs to the complex area. Consequently, Khodaei and Faez [3] improved Wu et al.'s algorithm by partitioning the cover image into non-overlapping blocks with 'three' consecutive pixels in the raster scan order. They then adopted a LSB substitution scheme and optimal pixel adjustment process (OPAP) to embed the secret message into the central pixel. Thereafter, the difference between the other two pixels and the data-embedded central pixel is used to determine the amount of secret message and the PVD scheme is then performed.

Khodaei and Faez's algorithm achieves satisfactory results, but it can be improved. First, in the data embedding phase, each pixel in the cover image is traversed in the raster scan order and then three consecutive pixels are collected as the same group. However, the pixel traversal scheme may affect the effectiveness of the proposed algorithm. The derivation for three consecutive pixels in the boundary region is also poorly manipulated, resulting in inaccurate pixel differences. Second, the embedding capacity is fixed and cannot be adjusted according to user demand. Third and finally, an overflow problem may occur after data embedding.

This study proposes an adaptive data hiding toward adjustable embedding capacity for grayscale images. Specifically, the proposed algorithm employs dynamic block subdivision on the cover image to address the drawbacks of previous algorithms. Moreover, we also modify Khodaei and Faez's embedding equation to avoid the overflow problem. Experimental results show that this adaptive algorithm can provide an adjustable embedding capacity and that the overflow problem is indeed solved. Our proposed method also solves the problem of pixel traversal order that appeared in the previous algorithm. The proposed technique is feasible in adaptive image data hiding.

This rest of this paper is organized as follows; Section 2 provides a review of Khodaei and Faez's algorithm; Section 3 details the proposed technique; Section 4 presents a discussion on the experimental results; and finally, Section 5 offers a conclusion and directions for future research.

2 Khodaei and Faez's Proposed Algorithm

This section introduces the core technologies of the algorithm proposed by Khodaei and Faez, containing range division, data embedding, and data extraction phases.

The range division phase divides the gray levels [0, 255] into five ranges R_j (j=1,...,5). Each range has its corresponding lower bound l_j and upper bound u_j. The number of embedding bits t_j for each region is determined by the width w_j of each region (See Eq. 1). For example, in Type 2 division, R_1=[0, 7], R_2=[8, 15], R_3=[16, 31], R_4=[32, 63], and R_5=[64, 255]. The embedding bit for the previous four regions is 3, 3, 4, and 5 sequentially. To avoid causing larger distortion, t_5 is set as $\lfloor \log_2 l_5 \rfloor = 6$.

$$t_j = \lfloor \log_2 w_j \rfloor = \lfloor \log_2 (u_j - l_j + 1) \rfloor \tag{1}$$

Next, the data embedding phase first partitions the cover image into non-overlapping blocks B_i with three consecutive pixels in the raster scan order. A k-LSB substitution and OPAP are then adopted to embed the secret message into the central pixel p_{i2} within each block. Consequently, the differences d_{i1} and d_{i2} between the data-embedded central pixel p'_{i2} with two other pixels p_{i1} and p_{i2} are calculated. The number of embedding bits for two other pixels is determined by the located region R_{i1} and R_{i2} illustrated in the range division phase. The stego values of p_{i1} and p_{i2} are then derived by the PVD scheme based on Eq. 2 whereas d'_{ik} is the data-embedded pixel difference. Repeat the above two phases for each block and the stego image can be obtained.

$$p'_{ik} = \begin{cases} p''_{ik} & if \ |p_{ik} - p''_{ik}| < |p_{ik} - p'''_{ik}| \ and \ 0 \le p''_{ik} \le 255 \\ p'''_{ik} & otherwise \end{cases},$$

$$where \ \begin{cases} p''_{ik} = p'_{ic} - d'_{ik} \\ p'''_{ik} = p'_{ic} + d'_{ik} \end{cases}, \ k = 1, \ 2 \tag{2}$$

For data extraction, the stego image is first partitioned into non-overlapping blocks with three consecutive pixels in the raster scan order. First, for each block, their proposed algorithm derives k-bit secret message from the central pixel. Thereafter, the differences between the central pixel with two other pixels are calculated and the located range can be determined according to the difference value. Finally, the secret message can be obtained from the distance between the above calculated difference and the lower bound of the located range.

3 Our Proposed Algorithm

This section presents the proposed adaptive data hiding algorithm. This algorithm also consists of three phases: the range division, data embedding, and data extraction.

The range division phase is the same as Type 2 division in Khodaei and Faez's algorithm. The phase divides the gray levels [0, 255] into five ranges and derives the relative information of each region, including the lower bound, the upper bound, and the number of embedding bits.

Instead of traversing each pixel in the raster scan or inverse-s orders, the data embedding phase employs two parameters, m and n, to indicate the width and height of the block. Consequentially, the cover image with *W-by-H* pixels is divided into non-overlapping blocks, each with *m-by-n* pixels. Further, we dynamically adjust the block size in the boundary region to avoid causing inaccurate pixel differences in the previous algorithm. For example, Figure 1 shows the subdivision results (in bold line) for a cover image with *8-by-8* pixels. The parameter m and n are both set as 3. Obviously, the block subdivision in the boundary region is adjusted according to the number of residual pixels in the boundary region.

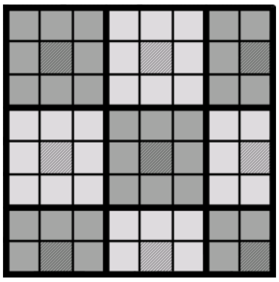

Figure 1. An example of the subdivision results for a cover image with *8-by-8* pixels.

For each block, similar to Khodaei and Faez's algorithm, we then adopt 3-LSB substitution and OPAP to embed the secret message into the central pixel p_{ic} (the hatched region in Figure 1) of each block. Consequently, we calculate the difference between each pixel within the same block and the data-embedded central pixel p'_{ic}. To avoid the overflow problem, we slightly modify the embedding equation shown in Eq. 3. The overflow problem may occur in the previous algorithm when the first condition does not hold and the value of p'''_{ik} is larger than 255.

$$p'_{ik} = \begin{cases} p''_{ik} & if \ |p_{ik} - p''_{ik}| < |p_{ik} - p'''_{ik}| \ and \ 0 \le p''_{ik} \le 255 \\ p''_{ik} & else \ if \ p'''_{ik} > 255 \\ p'''_{ik} & otherwise \end{cases},$$

Figure 2. The visual effects of cover images.

Figure 3. The visual effects of stego images with PSNR value near 30dB.

$$where \ \begin{cases} p''_{ik} = p'_{ic} - d'_{ik} \\ p'''_{ik} = p'_{ic} + d'_{ik} \end{cases}, \ k = 1, \ 2, ..., m \times n \quad (3)$$

For data extraction, the stego image is first partitioned into non-overlapping blocks with *m-by-n* pixels. First, for each block, the proposed algorithm initially derives a 3-bit secret message from the central pixel. Thereafter, the differences between the central pixel and each pixel within the same block are calculated and the located range can be determined according to the calculated difference value. Finally, the secret message can be obtained from the distance between the above calculated difference and the lower bound of the located region.

4 Experimental Results

This section presents the experimental results obtained from eight commonly used grayscale images: “Aerial,” “Baboon,” “Boat,” “F16,” “Goldhill,” “Lena,” “Peppers,” and “Tiffany.” The proposed algorithm was implemented in Matlab programming language on a personal computer with an Intel Xeon E3-1230V2 3.3 GHz processor and 16 GB of memory. Figure 2 shows the visual effect of each

Table 1. The embedding capacity of our algorithm under different block sizes.

Image Name	[3]	Block Size						
		1-by-1	*2-by-2*	*3-by-3*	*4-by-4*	*5-by-5*	*6-by-6*	*7-by-7*
Aerial	839678	786432	861926	879971	911212	927234	947232	957558
Baboon	877073	786432	915125	944952	974571	989536	1003660	1010428
Boat	821226	786432	828646	839258	859171	869137	882608	886140
F16	811591	786432	820944	829609	846117	854646	865987	871447
Goldhill	810949	786432	819961	828930	847041	855272	866791	876317
Lena	806527	786432	808122	812130	826040	833372	843587	849810
Peppers	804382	786432	808670	813598	826288	833233	843538	850871
Tiffany	801402	786432	803202	805750	813096	816668	822510	824387

Table 2. The PSNR value of our algorithm under different block sizes.

Image Name	[3]	Block Size						
		1-by-1	*2-by-2*	*3-by-3*	*4-by-4*	*5-by-5*	*6-by-6*	*7-by-7*
Aerial	32.38	40.70	28.92	27.99	26.47	25.90	25.09	24.82
Baboon	31.22	40.72	28.63	27.73	26.30	26.11	25.77	25.62
Boat	33.76	40.73	33.64	32.77	30.69	30.16	29.19	29.07
F16	35.27	40.72	34.12	33.32	31.53	30.57	29.67	29.08
Goldhill	35.90	40.72	35.20	34.34	32.80	32.34	31.68	31.16
Lena	36.13	40.75	35.64	34.99	32.76	32.38	30.73	30.14
Peppers	33.74	40.72	31.94	31.64	29.60	29.05	27.75	27.42
Tiffany	35.67	40.73	33.81	33.71	32.89	32.00	31.53	31.89

cover image. The embedded secret message is a 0/1 bit string randomly generated. The image quality between the cover image and the stego image was measured using peak signal-to-noise ratio (PSNR), shown in Eq. 3, where MSE (mean squared error) for an *m-by-n* grayscale image is defined as Eq. 4. The experimental results show there is no error in the extracted secret message from the stego images.

$$PSNR = 10log_{10}(255^2/MSE) \quad (3)$$

$$MSE = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}\left|C_{i,j} - S_{i,j}\right|^2 \quad (4)$$

This section first presents the experimental results of the proposed algorithm under different block size, including its embedding capacity and the PSNR value of the different input images. We also show the visual effects of the stego image with the PSNR value near 30dB for comparison. Second, this section also presents the embedding capacity and image quality under different block shapes to demonstrate that pixel traversal is a not wise scheme. Finally, this section illustrates the superiority of our proposed algorithm over previous adaptive data hiding algorithm to demonstrate the feasibility of the proposed method.

Table 1 and 2 shows the results of the embedding capacity and the image quality under different block sizes. The embedding capacity for each test image increases with increasing block size. Obviously, the image quality is decreasing with increasing block size. When the block size is degraded to *1-by-1*, the experimental results are the same as the result of 3-LSB substitution and OPAP. The embedding capacity in our proposed algorithm can be adjusted by the block size. Figure 3 shows the visual effects of the stego image with the PSNR value near 30dB.

This section presents the embedding capacity and image distortion of the test images under different block shapes. From Table 3, we observe the embedding capacity may be influenced by the gradient of the input image. For example, the vertical block subdivision is suitable for some images and the horizontal block subdivision is suitable for other images. However, this problem is not solved in previous algorithm. Khodaei and Faez use only raster scan order to traverse each pixel. Furthermore, the derivation for three consecutive pixels in the boundary region is poorly manipulated, resulting in inaccurate pixel differences. Instead, our block-based adaptive data hiding scheme can effectively address this problem.

Table 3. The embedding capacity of our algorithm under different block shapes.

Image Name	Block Shape					
	1-by-3	*1-by-5*	*3-by-1*	*3-by-5*	*5-by-1*	*5-by-3*
Aerial	839414	875612	850987	903987	890922	910014
Baboon	876063	922605	907264	969746	948851	972629
Boat	820440	839877	812685	855054	835323	855921
F16	811055	828260	812451	843243	831110	844305
Goldhill	810144	827052	810884	841964	830012	845338
Lena	805911	824013	797572	827670	808802	820039
Peppers	804059	817724	801360	825532	812952	823101
Tiffany	800674	809608	795387	813622	802508	810523

Table 4. The PSNR value of our algorithm under different block shapes.

Image Name	Block Shape					
	1-by-3	*1-by-5*	*3-by-1*	*3-by-5*	*5-by-1*	*5-by-3*
Aerial	32.61	29.09	29.50	26.84	27.45	26.68
Baboon	31.32	28.83	28.61	26.78	27.12	26.60
Boat	33.86	31.61	36.34	31.28	32.91	30.98
F16	35.48	32.89	35.16	31.93	32.37	31.36
Goldhill	35.99	33.94	36.57	33.15	34.45	33.43
Lena	36.06	33.13	37.62	32.76	35.44	33.81
Peppers	32.89	30.19	35.46	29.75	32.70	30.27
Tiffany	36.19	33.63	35.58	32.18	34.74	33.32

Finally, this section illustrates the superiority of our proposed algorithm with previous adaptive data hiding algorithms proposed by Khodaei and Faez. First, our embedding capacity can be adjusted by the block size. Users can properly determine the block size by considering the embedding capacity and image quality. Second, we dynamically modify the block shape to address the problem of pixel grouping in the boundary region. Finally, we slightly modify the equation for data embedding to solve the overflow problem caused by previous algorithm. The above advantages actually demonstrate the feasibility of our proposed algorithm.

5 Conclusion and Future Work

This study proposes an adaptive data hiding algorithm based on LSB substitution and PVD for gray-scale images. The main point of this algorithm is to use a dynamic block subdivision to replace pixel traversal. Thus, the results becomes weakly relying on the gradient of the input image. Furthermore, the pixels in the boundary region can be manipulated, allowing accurate pixel differences to be derived for data embedding. Finally and most importantly, the embedding capacity and image quality in our proposed algorithm can be adjusted according to the block size. This study demonstrates the feasibility of this technique for adaptive data hiding with the support of experimental results. In the future, we will integrate the image interpolation scheme and then perform pixel difference calculations to consider both the embedding capacity and image quality.

Acknowledgments

We thank Dr. Timothy Williams for his assistance in improving the clarity of this article. The authors also thank the anonymous reviewers for their constructive comments. This work was supported by the National Science Council under the grant numbers NSC 101-2221-E-468-026 and NSC 102-2221-E-468-025.

References

[1] W. Hong, T. S. Chen, and C. W. Luo, "Data embedding using pixel value differencing and diamond encoding with multiple-base notational system," *Journal of Systems and Software*, Vol. 85, pp. 1166-1175, 2012.

[2] L. C. Huang, L. Y. Tseng, and M. S. Hwang, "The study of data hiding in medical images," *International Journal of Network Security*, Vol. 14, pp. 301-309, 2012.

[3] M. Khodaei and K. Faez, "New adaptive steganographic method using least-significant-bit substitution and pixel-value differencing," *IET Image Processing*, Vol. 6, pp. 677-686, 2012.

[4] Y. P. Lee, J. C. Lee, W. K. Chen, K. C. Chang, I. J. Su, C. P. Chang, "High-payload image hiding with quality recovery using tri-way pixel-value differencing," *Information Sciences*, Vol. 191, pp. 214-225, 2012.

[5] X. Liao, Q. Wen, Z. Zhao, and J. Zhang, "A novel steganographic method with four-pixel differencing and modulus function," *Fundamenta Informaticae*, Vol. 118, pp. 281-289, 2012.

[6] H. C. Lu, Y. P. Chu, and M. S. Hwang, "A new steganographic method of the pixel-value differencing," *The Journal of Imaging Science and Technology*, Vol. 50, pp. 424-426, 2006.

[7] Y. Y. Tsai, "An adaptive steganographic algorithm for 3D polygonal models using vertex decimation," *Multimedia Tools and Applications*, in press, DOI: 10.1007/s11042-012-1135-8.

[8] C. M. Wang, N. I. Wu, C. S. Tsai and M. S.Hwang, "A high quality steganographic method with pixel-value differencing and modulus function," *The Journal of Systems and Software*, Vol. 81, pp. 150-158, 2008.

[9] D. C. Wu and W. H. Tsai, "A steganographic method for images by pixel-value differencing," *Pattern Recognition Letters*, Vol. 24, 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," *IEE Proceedings of Vision, Image and Signal Processing*, Vol. 152, pp. 611-615, 2005.

[11] N. I. Wu, K. C. Wu, and C. M. Wang, "Exploring pixel-value differencing and base decomposition for low distortion data embedding," *Applied Soft Computing*, Vol. 12, pp. 942-960, 2012.

Yuan-Yu Tsai received the B.S. degree in Department of Computer Science and Information Engineering from National Central University, Taiwan, in 2000, and the Ph.D. degree in Institute of Computer Science from National Chung Hsing University, Taiwan, in 2006. He is currently an assistant professor at the Department of Applied Informatics and Multimedia, Asia University, Taichung, Taiwan. His research interests include computer graphics and information hiding algorithms for three-dimensional models and images. He is a member of the ACM and the IEEE Computer Society.

Jian-Ting Chen is currently a Master student. He received his B.S. degree in Applied Informatics and Multimedia from Asia University, Taiwan, in 2012. His research interests include information hiding algorithms for images.

Chi-Shiang Chan was born in Taiwan in 1975. He received the B.S. degree in computer science in 1999 from the National Cheng-Chi University and the M.S. degree in computer science and information engineering in 2001 from the National Chung Cheng University. He received his Ph.D. degree in computer engineering in 2005 also from the National Chung Cheng University. From 2007 to 2010, he has worked as an assistant professor with the Department of Information Science and Applications, Asia University. He is currently an associate professor with the Department of Applied Informatics and Multimedia, Asia University. His research interests include image and signal processing, image compression, information hiding, and data engineering.