LETTER

Data Hiding in Spatial Color Images on Smartphones by Adaptive R-G-B LSB Replacement

Haeyoung LEE[†a)], ***Member***

SUMMARY This paper presents an adaptive least-significant-bit (LSB) steganography for spatial color images on smartphones. For each red, green, and blue color component, the combinations of All-4bit, One-4bit+Two-2bit, and Two-3bit+One-2bit LSB replacements are proposed for content-adaptivity and natural histograms. The high capacity of 8.4bpp with the average peak signal noise ratio (PSNR) 43.7db and fast processing times on smartphones are also demonstrated

key words: *steganography, adaptive LSB algorithm, spatial color images, data hiding, smartphones*

1. Introduction

Steganography is a technique of concealing data within a cover media which results in a stego media. The counterpart of steganography is steganalysis, which extracts hidden data from within the stego. Image steganography can usually be classified into two categories, according to the types of cover images: bitmap or png images in spatial domain and jpeg images in transform domain. For quality, lossless png images are sometimes preferred than lossy jpeg images.

There are three factors used to evaluate steganography: capacity, imperceptibility, and security [1]. The higher the capacity, the more data which can be embedded in the cover. However, the stego image should also maintain visual and statistical similarity to the cover image to avoid attacks.

Today, smartphones have advanced cameras and can generate color images easily. Steganography on smartphones can also embed various data such as text, voice, or image into the cover [2]. However, smartphones have limited resources. They have less memory and slower CPUs than PCs. Therefore, the memory and processing times of smartphones should also be considered for steganography.

Steganography in spatial domain is considered simpler and faster with higher data capacity than steganography in transform domain [1], [2]. For the performance of spatial image steganography, the data capacity is measured in bits per pixel (bpp). The quality of the stego image is measured by the PSNR value. The histograms of the cover and the stego images demonstrate their statistical similarity.

The least-significant-bit (LSB) algorithm is the most widely used in the spatial domain. This algorithm simply substitutes the least significant bit of pixels in a cover with secret message bits. However, a simple LSB algorithm is vulnerable to various attacks [1], [2].

Many improved methods in grayscale images have been proposed to increase: the capacity [3], security [4], or imperceptibility [3], [5], [6]. Steganography for color images has also been improved by extending the previous grayscale image based techniques [7]–[9] or adding adaptive selections on red, green, or blue (R-G-B) color components [10], [11]. In adaptive LSB steganography for 24bit-color images, 1 to 4bit LSB of each R-G-B 8bit-color component will be adaptively replaced with the secret data bits.

Previous adaptive LSB steganography in color images tried high data capacity of 12bpp: 4-4-4 LSB of R-G-B color components [8], [9] or 4bit LSB of the green and 8 bits of the blue component [10]. However, those fixed LSB replacements generated abnormal peaks in the histogram [9], [10]. Other methods, which replaced adaptively 1 to 4 bit LSB, showed good PSNR values. However, the capacity was relatively low (6.4bpp or 5.1bpp) and may be dependent on the cover image or the size of the concealed message bits [9], [11]. Previous methods also tried complex filtering or sorting, which require significant memory capacity and processing time on smartphones [7]–[11].

Table 1 Adaptive R-G-B LSB replacement with high data capacity. (All-4bit: One-4bit+Two-2bit: Two-3bit+One-2bit = 1 : 1 : 8)

Usage Ratio	Data Capacity	Cases of variously replaced LSBs of red, green and blue color components (R-G-B) in the color image.
10%	12 bpp (All-4bit)	
10%	8 bpp (One-4bit +Two-2bit)	
80%	8 bpp (Two-3bit +One-2bit)	

Manuscript received October 19, 2017.
Manuscript revised March 28, 2018.
Manuscript publicized April 25, 2018.

[†]The author is with Computer Engineering Dept., Hongik University, T703, 94 Wausan-ro Mapo-gu, Seoul, Korea.
a) E-mail: leeh@hongik.ac.kr

DOI: 10.1587/transinf.2017EDL8232

This paper presents a new adaptive LSB steganography technique for spatial color images in png on smartphones. As listed in Table 1, Table 4, Fig. 5, and Fig. 6, the proposed algorithm provides high data capacity while maintaining the level of undetectability: the embedding rate is 8.4bpp and natural histograms with an average PSNR 43.7 are generated. Besides, the average processing time is about 9 seconds for up to 92KBytes on typical smartphones as listed in Table 5 and Table 6.

2. Adaptive R-G-B LSB Steganography for Smartphones

This method proposes a simple and fast delta sorting method. The data capacity is systematically determined by worst case PSNR analysis. Finally, this new content-adaptive embedding method generates natural histograms.

2.1 Hiding and Extracting Data

Previous steganographic methods proposed filtering or sorting every pixel of the image at once and then selecting the needed number of pixels from the processed to embed data. Those methods require additional memory for the 3MN values in the color image of MxN pixels [11]. This paper proposes a simple sorting method, requiring the memory for N values only. The sorting is usually applied to the 4 most-significant-bits (MSB) of pixels. A value of 12 bit, combined three 4bit MSB of RGB components per pixel, is used during sorting, in this proposed method.

In pixel-value differencing (PVD) [3], the difference is calculated in nonoverlapping blocks, with two consecutive pixels. A significant difference value means an edge area. In this paper, the difference value is calculated with two consecutive pixels (i.e., $delta_i = |x_{i+1} - x_i|$) in overlapping blocks. Higher difference values between nonoverlapping blocks in PVD are also considered when using this method. This paper suggests sorting difference values ($delta_i$) in a row to find local edge areas.

The procedure of this new method is illustrated in Fig. 1. For embedding, the inputs are a color cover image and the zipped data to hide. From the size (S) of the data, the number of pixels in a row, P, is calculated. For each row, the following processes are repeated: sort $delta_i$ in descending order, select P pixels, and then replace LSBs of chosen pixels with the zipped data bits. The size S is also embedded in a specific head area. To extract the data from the stego image, first P should be calculated from S in the head area. Then the same sorting and selecting process as embedding is used. This process will be repeated for each row. Gathering extracted LSBs from the chosen pixels will restore the concealed data. Figure 2 demonstrates examples of the adaptive embedding by the proposed delta-sorting. Red dots in the image Lena represent pixels with embedded data.

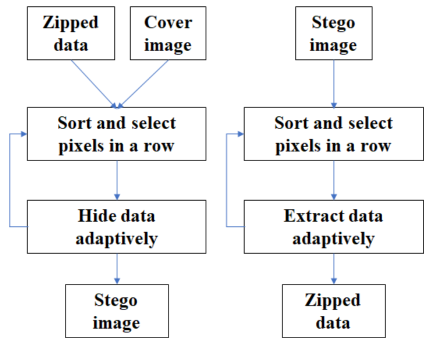

Fig. 1 Procedure of the proposed adaptive R-G-B LSB steganography. The embedding process is presented on the left and the extracting process is presented on the right.

Fig. 2 Examples of adaptive embedding in edge areas by the proposed delta-sorting method in this paper. Red dots are the pixels with concealed data. From left to right, a color image of Lena in 512x512 included 46,080bytes, 64,440 bytes, and 76,800 bytes of concealed data.

2.2 Worst Case PSNR-Based Combination of R-G-B LSB Replacement

The quality of the stego image is measured by PSNR values. The PSNR value can be evaluated by using Eq. (1) and Eq. (2) [12]. The mean-square-error (MSE) in Eq. (1) is used to calculate the differences between the corresponding pixel values, $cover_{i,j}$ in the cover and $stego_{i,j}$ in the stego. For color images, MSE should be calculated for each red, green, and blue color component such as MSE(R), MSE(G), and MSE(B) in Eq. (2). Usually PSNR values of 30db or higher demonstrate decent quality stego images [12].

$$MSE = \frac{1}{NM} \sum_{i=1}^{N} \sum_{j=1}^{M} \left(cover_{i,j} - stego_{i,j}\right)^2 \quad (1)$$

$$PSNR = 10 \times log_{10} \frac{3 \times 255^2}{MSE(R) + MSE(G) + MSE(B)} \quad (2)$$

For the worst case MSE, the maximum difference square between $cover_{i,j}$ and $stego_{i,j}$ is $(2^k - 1)^2$, where k is 1, 2, 3, or 4 replaced bits [6]. With the worst case MSE(R), MSE(G), and MSE(B), the worst case PSNR values for each fixed R-G-B LSB replacement are listed in Table 2. Even one 4bit LSB substitution can decrease the quality of the stego. However, 4-LSB replacement is essential for high data capacity. The combinations of fixed R-G-B LSB replacements have been tried and the worst case PSNR value for each combination is listed in Table 3.

As shown in Table 2 and Table 3, the lower data capacity does not guarantee the higher PSNR values. The combi-

Table 2 Worst case PSNR by fixed R-G-B LSB replacement.

Number of LSB bits replaced in Red-Green-Blue component (R-G-B)	Capacity (bpp)	PSNR (db)
All-1bit (1-1-1)	3	48.13
All-2bit (2-2-2)	6	38.59
Two-3bit + One-2bit (3-3-2), (3-2-3), (2-3-3)	8	32.61
One-4bit + Two-2bit (4-2-2), (2-4-2), (2-2-4)	8	29.05
All-3bit (3-3-3)	9	31.23
All-4bit (4-4-4)	12	24.61

Table 3 Worst case PSNR by various R-G-B LSB combinations.

Usage Ratio All-4bit (%)	Usage Ratio One-4bit + Two-2bit (%)	Usage Ratio Two-3bit + One-2bit (%)	Data Capacity (bpp)	Worst PSNR (db)
30	0	70	9.2	30.21
0	40	60	8.0	31.18
0	70	30	8.0	30.11
10	20	70	8.4	31.10
10	10	80	8.4	31.45

nation of 10% of All-4bit, 10% of One-4bit+Two-2bit, and 80% of Two-3bit+One-2bit (i.e., 1 : 1 : 8) for R-G-B LSB replacement respectively generates the highest PSNR values of 31.45db with the capacity of 8.4bpp. This combination can produce a better stego image even with more concealed data, than other combinations of lower capacity 8.0 bpp. The actual PSNR values of stego images using this new method are much higher than the worst case PSNR value, listed in Table 4. The average PSNR for tested images is 43.7db.

2.3 Content-Adaptive Embedding

After sorting $delta_i$ in descending order, P pixels (x_i) in a row are selected to hide data bits. For 1 : 1 : 8 embedding of 30byte data, 30bytes in 3 : 3 : 24 will be embedded in 29 pixels of the cover image: First, 3bytes from 2 pixels by All-4bit LSB replacements, and then 3bytes from 3 pixels by One-4bit+Two-2bit LSB replacements, and finally 24bytes from 24 pixels by Two-3bit+One-2bit LSB replacements. In this way, 4bit replacement will be applied to the complex pixels with higher difference values.

For more content-adaptivity, 4bit MSB values of the R-G-B component will determine the various combinations for One-4bit+Two-2bit and Two-3bit+One-2bit (see Table 1). For One-4bit +Two-2bit, a color component with the middle MSB value embeds 4bits in its LSB. For Two-3bit+One-2bit, a component with the largest MSB value embeds 2bits. These combinations generated the best PSNR values when tested.

The smallest replacement (i.e., 2 LSB bits) in the color component having the largest MSB can minimize noticeable color changes in the stego images. For example, abnormal colors are hardly presented in the enlarged stego4 for the part of hair in Lena having a lot of data embedded, as shown in Fig. 3. As a result, various combinations of 4-2-2, 2-4-2, 2-2-4, 2-3-3, 3-2-3, or 3-3-2 LSB replacements are induced and contribute to generating natural distribution histograms.

Previous methods added adaptive LSB matching to improve the quality of the stego image after embedding [11]. For example, 11001000 in a component is changed to

Fig. 3 Enlarged sub-images of Lena: the middle is the cover and the right is the stego4 having a lot of data hidden but no noticeable abnormalities.

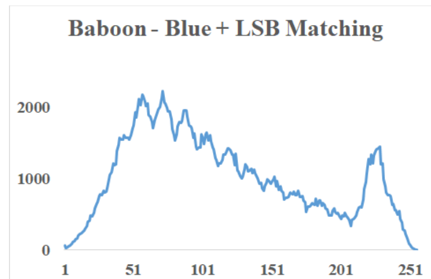

Fig. 4 Example of an abnormal histogram by adding LSB matching (The histogram of the blue component of Baboon's Stego4 with 92,160 bytes).

Table 4 PSNR values of the stego images in db.

Cover images	Stego1	Stego2	Stego3	Stego4
Baboon	45.41	44.16	43.19	42.39
Lena	45.35	44.14	43.16	42.37
Pepper	45.14	43.91	42.94	42.12
Pink-flower	45.31	44.08	43.13	42.31
Yellow-flower	45.33	44.11	43.13	42.32
Average	45.31	44.08	43.11	42.30

11001111 when embedding 111 in its LSB. LSB matching will modify 11001111 to 11000111 by flipping 1 to 0 in the 4th LSB. It will decrease the difference between 11001000 in a cover and 1100111 in a stego and increase the PSNR value of the stego. However, as shown in Fig. 4, the forced flipping in specific LSB generated abnormal histograms. Hence, LSB matching is not suggested in this new algorithm.

3. Experimental Results on Smartphones

This new adaptive steganography was implemented in C# in a tool of Xamarine Android and was tested on typical Android smartphones; Samsung Galaxy Note5 and LG V20. As shown in Fig. 6, five color png images were tested as cover images: Baboon, Lena, Pepper, Pink-flower, and Yellow-flower in 512x512. The last two flower images were captured by a smartphone. Text, images, and recorded voice files were zipped to create data files. The four varied sizes of data files are 46,080 bytes (368,640 bits), 64,440 bytes (491,520 bits), 76,800 bytes (644,400 bits), and 92,160 bytes (737,280 bits). The generated stego images are named Stego1, Stego2, Stego3, and Stego4 respectively.

This new method generated high data capacity of 8.4 bpp (see Table 3). The PSNR values of stego images are listed in Table 4. Natural histograms of RGB components of an image Baboon are illustrated in Fig. 5. To demonstrate

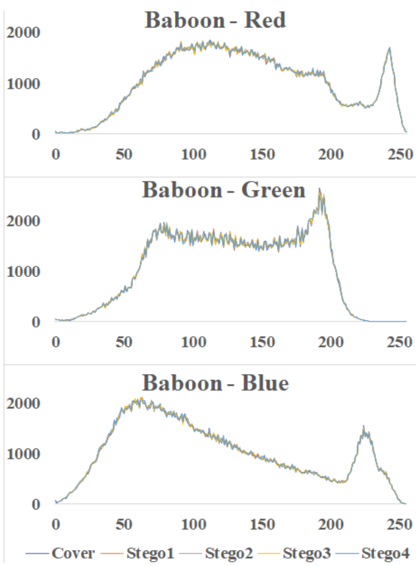

Fig. 5 Histograms of an image Baboon generated by this new method. Histograms of the cover and the stegos are similar to each other.

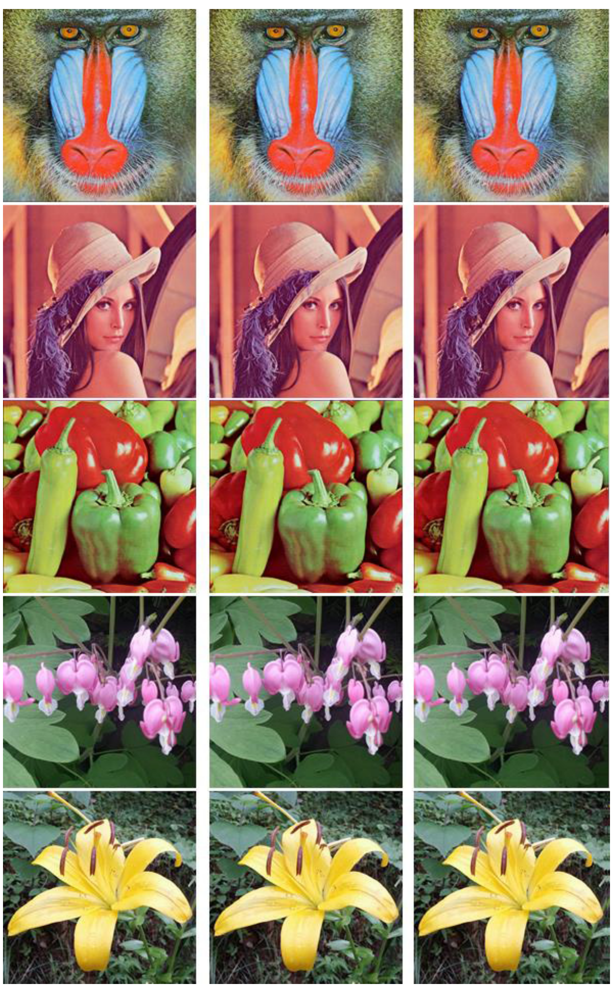

Fig. 6 Color cover images on the left, Stego1's with 368,640bits hidden in the middle, and Stego4's with 737,280bits hidden on the right.

Table 5 Processing time for embedding on smartphones in seconds.

Smartphones	Stego1	Stego2	Stego3	Stego4
Galaxy Note5	7.41	7.72	8.31	8.78
V20	9.93	10.45	10.81	11.34
Average	8.67	9.09	9.56	10.06

Table 6 Processing time for extracting on smartphones in seconds.

Smartphones	Stego1	Stego2	Stego3	Stego4
Galaxy Note 5	6.61	6.88	7.24	7.44
V20	9.22	9.56	9.94	10.40
Average	7.92	8.22	8.59	8.92

imperceptibility, cover images and their two stego images are listed in Fig. 6, such as the cover on the left, Stego1 in the middle, and Stego4 on the right. The average processing times, tested on five images with four data sets are listed in Table 5 for embedding, and Table 6 for extracting.

4. Conclusion

This paper introduces a new adaptive LSB steganography for color images in png format on smartphones. Worst case PSNR based LSB replacements are presented for high data capacity while maintaining undetectability. Delta sorting is also proposed to save memory and processing time. Future research will extend this algorithm to jpeg color images and investigate steganalysis against this method.

Acknowledgments

The author would like to thank Weonjoon Choi at SunnyGraphy in Korea for his assistance with C# programming on Android smartphones. This work was supported by the 2017 Hongik University Research Fund.

References

[1] B. Li, J. He, J. Huang, and Y.Q. Shi, "A survey on image steganography and steganalysis," J. Information Hiding and Multimedia Signal Processing, vol.2, no.2, pp.142–172, April 2011.
[2] W. Mazurczyk and L. Caviglione, "Steganography in Modern Smartphones and Mitigation Techniques," IEEE Communication Surveys & Tutorials, vol.17, no.1, pp.334–357, First Quarter 2015.
[3] C.-H. Yang, C.-Y. Weng, S.-J. Wang, and H.-M. Sun, "Adaptive data hiding in edge areas of images with spatial LSB domain systems," IEEE Trans. on Information Forensics and Security, vol.3, no.3, pp.488–497, Sept. 2008.
[4] J. Fridrich, D. Soukal, and M. Goljan, "Maximum likelihood estimation of length of secret message embedded using ±k steganography in spatial domain," Proc. SPIE The Int. Society for Optical Engineering, vol.5681, pp.595–606, March 2005.
[5] W. Luo, F. Huang, and J. Huang, "Edge adaptive image steganography based on LSB matching revisited," IEEE Trans. on Information Forensics and Security, vol.5, no.2, pp.201–214, June 2010.
[6] C.-K. Chan and L.M. Cheng, "Hiding data in images by simple LSB substitution," Pattern Recognition, vol.37, no.3. pp.469–474, March 2004.
[7] L. Li, B. Luo, Q. Li, and X. Fang, "A color images steganography method by multiple embedding strategy based on sobel operator," Proc. 2009 Int. Conf. Multimedia Information Networking and Security, Hubei, China, pp.118–121, Nov. 2009.
[8] A. Majeed, M.L. Mat Kiah, H.T. Madhloom, B.B. Zaidan, and A.A.

Naidan, "Novel approach for high secure and high rate data hidden in the image using image texture analysis," Int. J. Engineering and Technology, vol.1, no.2, pp.63–69, Oct. 2009.
[9] R. English, "Comparison of high capacity steganography techniques," Proc. 2010 Int. Conf. Soft Computing and Pattern Recognition, Paris, France, pp.448–453, Dec. 2010.
[10] M. Juneja and P.S. Sandhu, "Two components based LSB and adaptive LSB steganography based on hybrid feature detection for color images with improved PSNR and capacity," Int. J. Computer Science and Electronics Engineering, vol.1, no.2, pp.345–351, 2013.
[11] J.M. Rad, "Intensity adaptive LSB method applying a revised matching," Int. J. Computer applications, vol.70, no.27, pp.20–26, May 2013.
[12] N.-K. Lim, D.-Y. Kim, and H. Lee, "Interactive progressive image transmission for realtime applications," IEEE Trans. on Consumer Electronics, vol.56, no.4, Nov. 2010.