Changing Partners Technique in Reversible Steganography

Jae-Hyeon Woo[1)], Weon-Jin Cho[2)]

[1)] Graduate School of Information Management and Security,
Korea University, Korea

[2)] FTTH Industry Forum
Korea Association of Information and Telecommunication, Seoul, Korea

Abstract

Steganography is hiding messages in cover materials like image, audio, etc. The previous studies are limited in the shifting problem of the histogram and capacity. This paper proposes a new technology which overcomes these two problems. Our new technology uses two colors as a pair and chooses either changing to its partner or remaining as they are according as each hiding bit. As a result, we show that the result of this technology shows that the hiding capacity is higher than other methods.

1. Introduction

In the previous method such as LSB modification, there is no need to recover the original cover image because the original image was not so important. They only focused on the hidden message. Recently, it is required to recover the cover image in many applications, for example, verification of the message integrity and signature. When we send an image, we insert the watermarking to the original image and send the modified image. The receiver recovers both the watermarking and the original image, and verifies the integrity and authentication from this information.

There are many reversible steganography or reversible watermarking methods. Chang et al. [1] suggested a side matching and relocation method which hides a secret message in VQ-compressed index. Ni et al. [2] also suggested reversible data hiding which shifts the histogram of the cover image. Tian [4] suggested a difference expansion method. This method doubles the difference of neighbor pixel and hides one bit in that difference. Others suggested efficient methods [5-10].

Let S_1 and S_2 be colors of any pixel (S_1 and S_2 may be colors of pixels, mean value of wavelets, or some other things). When hiding a bit and one pixel has the S_1 color, we change the S_1 color of a pixel into S_2 color, and there must be a mark showing whether S_2 is the original color or converted color from S_1. For example, when one bit is hidden in a number 5 (i.e., 0101_2), 5 remains as 5 when we hide bit 0, or it shifts bits to the left (doubling the value) when we hide bit 1. So the value 5 is converted to 5 or 10. Similarly, 10 can be converted to 10 or 20. In the recovering process, assume that one value is 10, how can we recover the message? Did it come from 5 (hidden bit 1) or from the original 10 (hidden bit 0)? There is no way to determine whether 10 was doubled from 5 or the original 10 without any information. Then, a location map is used to solve such a problem, but the location map size is sometimes too big to guarantee payload capacity. So it is necessary to minimize the location map size.

In this paper, we suggest a new method to minimize the size of recovering information and enlarge hiding capacity.

2. Previous Methods

Chang et al. [1] suggested hiding messages in VQ-Compressed index. It needs to generate state codebooks G_0, G_1 and G_2 from super codebook of the index table. Suppose that only two-side matching, block X's matching side pixels are compared with U-side pixels and L-side pixels in Fig. 1. We compare each block with its upper and left blocks (side matching) and define each codeword group G_0, G_1 and G_2. After moving all X's in G_1 to G_2, X's in G_0 move to G_1 if we hide bit 1 and unchanged if hide bit 0. The secret bit string can be recovered from X in G_0 or G_1. This method can hide the secret string in VQ-Compressed image index. However, the capacity is very low and PSNR is not so good because this method hides one bit in a block size code ($n \times n$ pixel size).

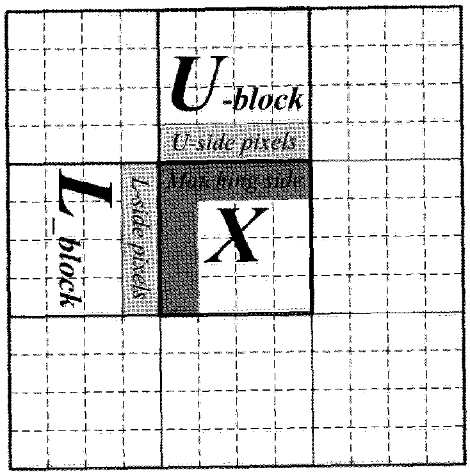
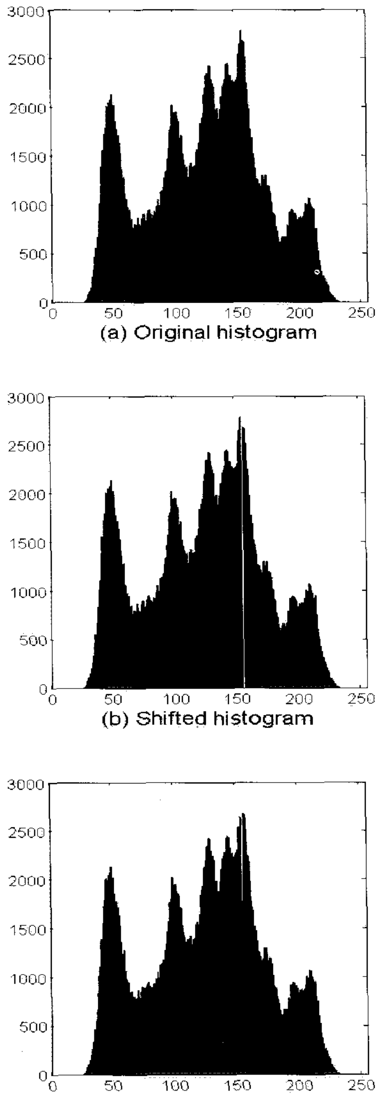


Fig. 1. Chang et al.'s side matching and relocation method.

Ni et al. [2] showed that they shift the histogram of the color spectrum to make a hole if the image has a zero point. The hole is made near the peak color in the histogram, and the secret messages are hidden in the peak color. Fig. 2 shows both the method and result. If there isn't any zero point, keep the coordinates and color value of minimum point, and regard this minimum color as a hole. However, this method saves all the minimum point coordinates to the header, and the capacity is limited according as the number of zero points.



(a) Original histogram

(b) Shifted histogram

(c) Stegoimage histogram

Fig. 2. Ni et al.'s hiding method

Tian [4] suggested a method to hide messages using the difference. The difference between two close pixels is doubled and one bit is hidden in the doubled difference. It is divided into four disjoint classes of difference values, *EZ*, *EN*, *CN*, and *NC* and hid messages in *EZ*, *EN* and *CN*. But the location map distinguishing *EN* and *CN* is still overloaded.

3. Proposed Method

In one image file, one pixel has a color value, and its neighbor pixels have similar color. The neighbor colors of one pixel have some bound. For example, an image is shown in Fig. 3 (a).

32	54	48	49	48	70	85	101
50	48	61	55	60	47	72	95
71	54	43	59	50	49	63	89
70	64	48	48	40	60	56	48
95	100	75	74	74	74	71	44
102	105	99	100	100	102	98	80
150	171	170	165	167	167	160	150
143	170	171	180	181	181	181	170

(a) Original image

32	54	48_0	49	171_1	70	85	101
50	171_1	61	55	60	47	72	95
71	54	43	59	50	49	63	89
70	64	48_0	171_1	40	60	56	48
95	100	75	74	74	74	71	44
102	105	99	100	100	102	98	80
150	171_0	170	165	167	167	160	150
143	170	48_1	180	181	181	181	170

(b) Stegoimage

Fig. 3. (a) Table of the pixel colors, and (b) table after hiding message string '0110101'

Each number means the gray color value (from 0 to 255). We consider a pair (48, 171) and their right side values. The right side values of 48 are {49, 70, 61, 48, 40}. There is one 48 which has no right side. The right side values of 171 are {170, 180}. The maximum value of the right side of 48 is 70 and its minimum value of the right side of the color 48 is 40. Similarly, the maximum value of the right side of the color 171 is 180 and its minimum right side value is 170. (See Fig. 4).

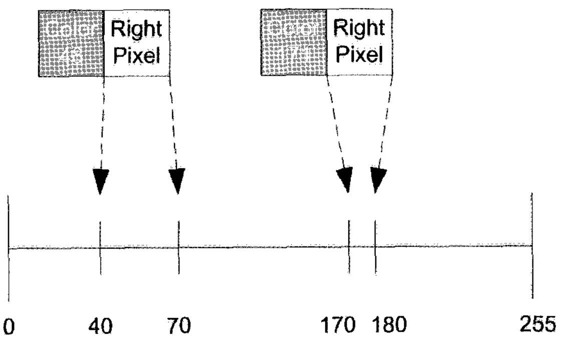


Fig. 4. The right side pixel's Max and Min bounds.

We denote *Maxright*(48) = 70 and *Minright*(48) = 40. We define two colors i and j ($i < j$) as a pair ($i; j$) with no overlap of their neighbors, that is, *Maxright*(i) < *Minright*(j), and the color Pair Set is

$$\text{Pair Set} = \{(i; j) \mid i < j,\ \text{Maxright}(i) < \text{Minright}(j)\}. \qquad (1)$$

We assign the *Maxright*(48) = 70 as *border*. Both pixel 48 and its right side pixel is less than or equal to border 70. Both pixel 171 and its right side pixel are larger than border. In order to hide a bit in the pixel, we scan from the first pixel. When we meet color 48, it is changed into 171 if hide bit 1, otherwise it is not changed. When we meet the color 171, it is changed into 48 if hide bit 1. It is shown in Fig. 3 (b) after hiding a message string '0110101'. Note that no bit is hidden in the marked 48 in Fig. 3 because this pixel has no right side pixel.

The recovering process has the reversed direction, from the last hidden pixel to the first pixel. When we meet the color 48, the bit 1 is extracted and the color 171 is recovered from 48 if its right side pixel is larger than border. The bit 0 is extracted and color 48 remains if its right side pixel isn't larger than border. When we meet color 171, the recovering process can be done similarly.

All colors from 0 to 255 are not always used in one picture. So we can also make a color pair (i, j) even if i or j is not used in the cover image. In that case, we don't have to find neither *Maxright* and *Minright* nor border of i and j.

If (i, j) is an element of the pair set, then we call i and j the member color, and the i and j are partners of each other. When we hide a secret message string, we search the member color from the first pixel of the cover image. We change the member color into its partner color if we hide bit 1 and the color remains if hide bit 0. The number of pairs cannot exceed 128 pairs in an 8-bit gray image.

We consider two modes to define the pair set. The first mode is Maximum Capacity mode that finds the mostly used color i and j satisfying only the condition *Maxright*(i) < *Minright*(j) or *Maxright*(j) < *Minright*(i). The second mode is PSNR mode that finds the closest j from i such that $|i - j|$ is the minimum value. This mode also satisfies the condition of the first mode. Maximum Capacity mode is used when hiding large messages. PSNR mode is used when small message.

When recovering the cover image and extracting the hidden bit, the information about the triple (i, j, and border) is enough. So any location map isn't needed. Consider the case of a pair (i, j) satisfying $i < j$. If a pixel is the member color i and its right pixel color is not larger than the border, the hidden bit 0 is extracted. If its right pixel color is larger than border, the hidden bit 1 is extracted and the original color j can be recovered from i. If the pixel is the member color j and its right pixel color is larger than the border, the hidden bit 0 is extracted. If its right pixel color is not larger than border, the hidden bit 1 is extracted and the original color i can be recovered from j.

If one of i and j is not used as a color in the cover image, we call that color an empty color and we need only information about the pair (i; j) and the empty color value. In this case, recovering is much easier because we don't need to compare with border. Assume that (i, j) is a pair and j is the empty color. If we meet a pixel color i in recovering process, the hidden bit is 0. If we meet a pixel color j, the hidden bit is 1 and the original color i can be recovered from j. Theoretically, an image with more empty colors can hide larger amount of messages because we can make more pairs in such an image. Empty color can be a partner of any color without considering about border.

We must search the member color from the last hidden pixel of the cover image for recovering as the changed pixel C affects when its left pixel L recovered. L compares with C and interprets whether its color should be changed or not. If C is changed, then L can't be interpreted from the changed C correctly. There is no confusion when recovering is done in the inverse direction. Till now we used only the right side direction, but we can also compare with any direction pixel if hiding and recovering order is concerned..

.

4. Experimental Results

Table 1. Expected Maximum Capacity

Images (512-512)	Max Capacity (bits)	Max bpp	Max Pairs
Baboon	145,445	0.555	80
F-16	226,891	0.866	91
Lena	229,393	0.875	109
Peppers	156,025	0.595	74
Tiffany	242,020	0.923	93
Sailboat	191,809	0.732	76

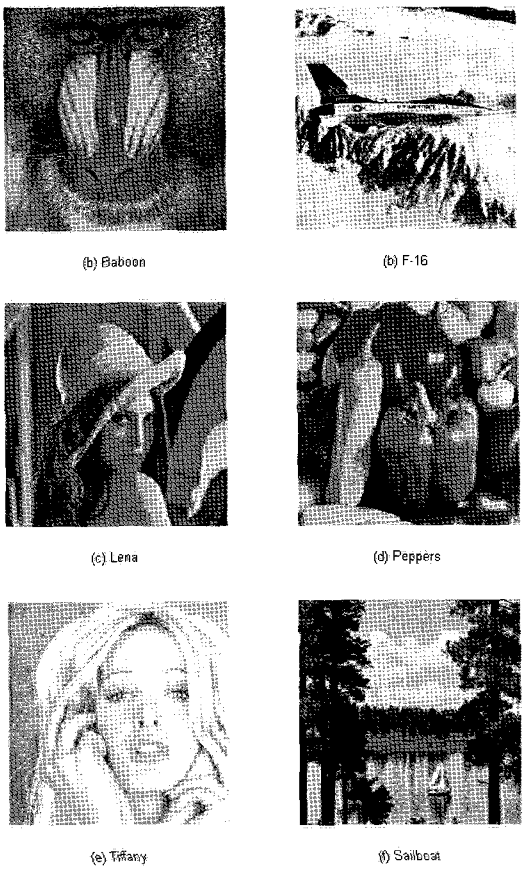


Fig. 5. Sample pictures

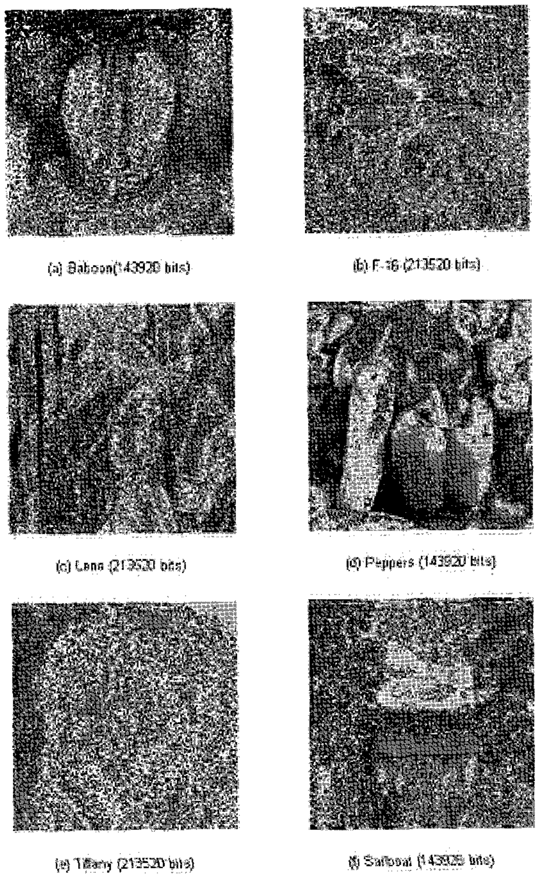


Fig. 6. Result of Maximum Capacity mode.

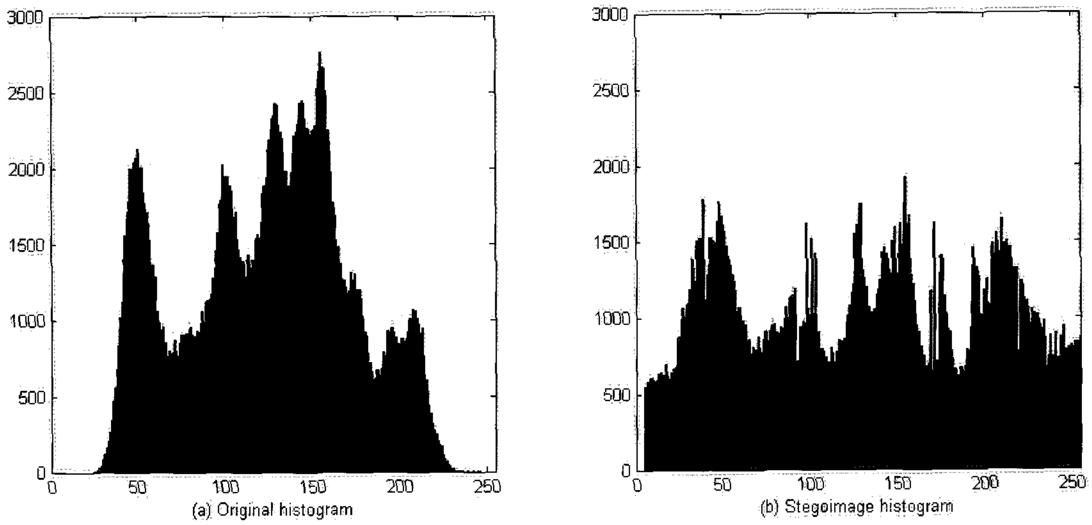


Fig. 7. Histograms of Lena image before and after hiding with Maximum Capacity mode.

We used six sample images as the cover image in Fig. 5. As mentioned, there are two modes of hiding, Maximum Capacity mode and PSNR mode. PSNR mode is used for a small message hiding. In Maximum Capacity mode, we can hide as much as table 1. This capacity is enhanced

The result of Maximum Capacity mode is shown in Fig. 6 and the result of PSNR mode is shown in Fig. 8. The histograms of each mode are shown in Fig.7 and Fig. 9, respectively.

Chang[1]'s method also compares the block with neighbor block, but hiding one bit changes one block and thus the bpp (bit per pixel) is too small, but the proposed method hides one bit in one pixel and the bpp is much larger than that of Chang's. In the Ni[2]'s method, they moved one color to another in histogram, too. But one peak color matches with only a shifted hole color. Therefore, the capacity is as much as the heights of peak

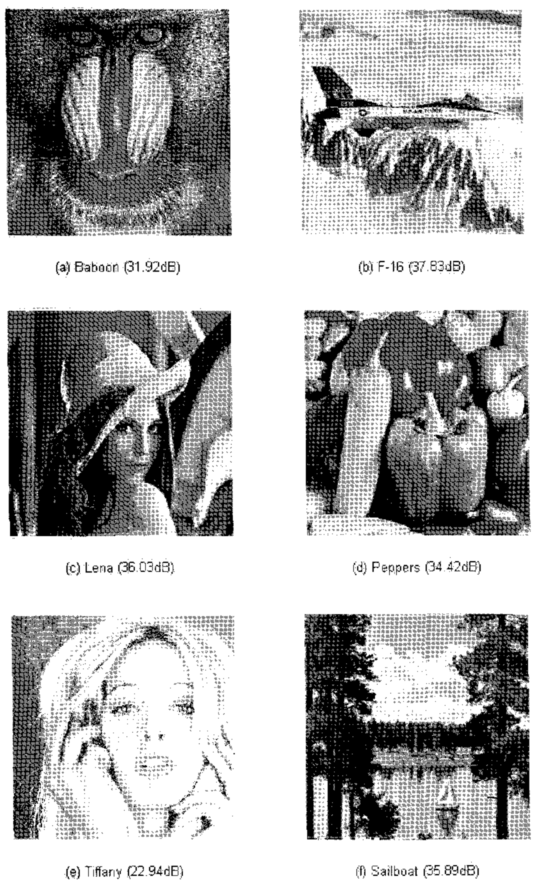


Fig. 8. Result of PSNR mode hiding 10,928 bits.

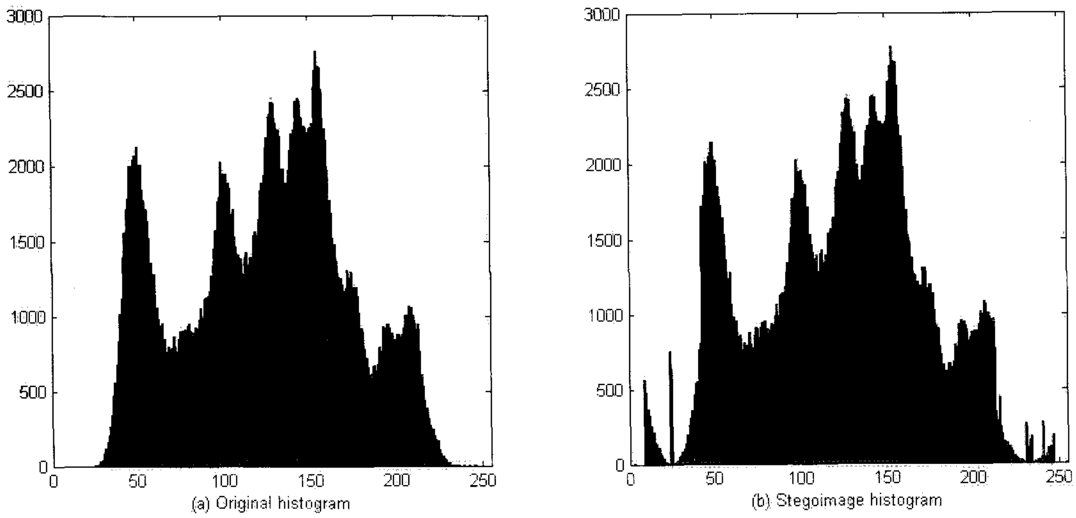


Fig. 9. Histograms of Lena image before and after hiding with PSNR mode.

colors in the histogram for one pair(peak point, hole). However, our method can hide more than Ni's method because it hides in both partner colors without any shift. This method is also better than other methods using the location map. The header containing the recovering information is only about 50 to 300 bytes depending on the size of hidden messages.

5. Conclusion

This paper contributes to reducing the large location map to short header and enlarging the capacity of hidden message. Only some color pair information can stand for the location map. Future work is to improve the method of making pairs to obtain better PSNR as converting the colors to the closest color. This method can be implemented to military service, medical center, and government image processing with a huge hidden message.

6. Acknowledgment

This work was in part supported by ITRC, Korea University.

References

1. Chang, C. C., Lin, C. Y.: Reversible Steganography for VQ-Compressed Images Using Side Matching and Relocation. *IEEE Transactions on Information Forensics and Security*, vol. 1, no. 4, (2006) 493-501.
2. Ni, Z., Shi, Y. Q., Ansari, N., Su, W.: Reversible Data Hiding. *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 3, (2006) 354-362.
3. Xuan, G., Yao, Q., Yang, C., Gao, J., Chai, P., Shi, Y. Q., Ni, Z.: Lossless Data Hiding Using Histogram Shifting Method Based on Integer Wavelets. *Lecture Notes in Computer Science*, vol. 4283, (2006) 323-332.
4. Tian, J.: Reversible Data Embedding Using a Difference Expansion. *IEEE Transactions on Circuits and Systems for Video Technology*. vol. 13, (2003) 890-896.
5. Kamstra, L., Jeijmans, H.: Reversible Data Embedding Into Images Using Wavelet Techniques and Sorting, *IEEE Transactions on Image Processing*, vol. 14, (2005) 2082-2090.
6. Fridrich, J., Goljan, M., Du, R.: Invertible Authentication. *Proceedings of the SPIE Photonics West, Security and Watermarking of Multimedia Contents III*, San Jose, California, vol. 3971, (2001) 197-208.
7. Fridrich, J., Goljan, M., Lisonek, P.,

Soukal, D.: Writing on Wet Paper. *IEEE Transactions on Signal Processing*, vol. 53, (2005), 3923-3935.

8. Alattar, A. M.: Reversible Watermark Using Difference Expansion of Triplets. *Proceedings of the International Conference on Image Processing*, vol. 1, (2003) 501-504
9. Alattar, A. M.: Reversible Watermark Using Difference Expansion of Quads. *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, (2004) 377-380
10. Alattar, A. M.: Reversible Watermark Using the Difference Expansion of a Generalized Integer Transform. *IEEE Transactions on Image Processing*, vol. 13, no. 8, (2004) 1147-1156.

Biography

우 재 현

1991. 2. 부산 동래고등학교 졸업
1996. 2. 서울대학교 기계공학과 졸업 (공학사, BSME)
2002. 2. 서울대학교 전기컴퓨터 공학부 졸업(공학석사, MSEE)
2007. 3. 2. 고려대학교 정보경영전문대학원 박사과정 입학
1996.3 ~ 1998.6 항만단 소대장 (ROTC #34)
2002.3 ~ 2004.3 우리은행 E-business사업단 근무
2004.4 ~ 2006.3 삼성SDS 정보보안센터 근무
2007.3 ~ 고려대학교 정보경영공학전문대학원 박사과정

조 원 진(Cho, Weon-jin)

2002, 연세대학교 공학대학원, 공학석사
2007, 광운대학교 경영정보학 박사과정
1998~2005, 정보통신부 근무
2005~현재, 한국정보통신산업협회(FTTH산업협의회 사무국장)
주요관심사항: 광대역통합망(BcN), 광가입자망(FTTH), 가입자망, 컴퓨터통신망, 컴퓨터네트워크, 무선매쉬시스템 등