International Journal of Advance Research in Science and Engineering
Vol. No.4, Special Issue (01), September 2015
www.ijarse.com
IJARSE
ISSN 2319 - 8354

REVERSIBLE ENCRYPTED DATA CONCEALMENT IN ENCRYPTED IMAGES BY RESERVING ROOM APPROACH FOR DATA PROTECTION SYSTEM

Bala Muhammad[1], Bello Musa Yakubu[2]

[1,2] *Computer Science and Engineering Department, Sharda University (India)*

ABSTRACT

Reversible Data Hiding (RDH) gaining the attraction in the world of secured message passing through encrypted images. It is due to the excellent property of recovering the original cover or image after embedded data is extracted while protecting the encrypted image content's confidentiality. The related works may embed data by reversibly vacating room from the encrypted images, which may be subject to some errors on data extraction and/or image restoration. In this paper, we propose a novel method by reserving room before encryption and a watermarking technology with a traditional RDH algorithm, and thus it is easy for the data hider to reversibly embed data in the encrypted image with an authentication to the sender. The data which is more secure and unrecoverable for an external hacker/intruder since it is stored in a random way. The proposed method can get 100% reversibility with data extraction and image recovery. Experiments show that this novel method can embed more than 10 times as large payloads for the same image quality as the earlier methods, such as for PSNR 40dB or above.

Keywords: Reversible Data Hiding, Image Encryption, Random Data Hiding, Self-Embedding.

I. INTRODUCTION

Reversible data hiding in images is a technique, by whichthe original image can be recovered without any loss inquality after the data embedded in an encrypted image isextracted. This technique is widely used in medical imagery,military imagery and forensics, where no distortion of theoriginal cover image is allowed to avoid data loss and for thesmooth message passing. Reversible data hiding wasintroduced in the beginning of 2000's and gained theresearch interest and have emerged in recent years withsome general frameworks.In theoretical aspect a rate-distortion model for Reversibledata hiding [1], proved the rate-distortion bounds for lessmemory and proposed a recursive code construction,however, does not approach the bound. An improvedrecursive code construction [2],[3] for binary covers, whichestablishes the equivalence between data compression andReversible data hiding for binary covers. A more popularmethod, difference expansion[5], where the difference ofeach pixel group is expanded thus the least significant bits ofthe difference are all-zero and can be used for embeddingmessages. Another strategy is histogram shift [6], wherespace is saved for data embedding by shifting the bins of thehistogram of gray values.

The combined differenceexpansion or histogram shift [7]-[11] to residuals of theimage, to achieve better performance. Encryption [12] is aneffective and popular means as it converts the original andmeaningful content

to incomprehensible one. Suppose amedical image database is stored in a data center, and aserver in the data center can embed notations into anencrypted version of a medical image through a RDHtechnique. With the notations, the server can manage theimage or verify its integrity without having the knowledgeof the original content, and thus the patient's privacy isprotected. On the other hand, a doctor, having thecryptographic key, can decrypt and restore the image in areversible manner for the purpose of further diagnosing.Some attempts on RDH in encrypted images have beenmade such as divided the encrypted image into severalblocks. The data extraction and image recovery proceed byfinding which part has been flipped in one block. But theencrypted image [18] should be decrypted first before dataextraction. To separate the data extraction from imagedecryption, emptied out space for data embedding followingthe idea of compressing encrypted images. Compression ofencrypted data can be formulated as source coding with sideinformation at the decoder, in which the typical method is togenerate the compressed data in a lossless manner byexploiting the syndromes of parity-check matrix of channelcodes. In Reversible Data Hiding, only the sender andreceiver are aware of the hidden data and typically if theloaded file falls into the hands of anyone else they wouldn'tretrieve the hidden data. While seeing an encrypted Imagethe first thing that comes to a third person is that, an Imageencrypted and this image is the message that is exchangedbetween the sender and receiver so decryption of Image isdone.

II. RELATED WORK

The methods proposed in [16]–[18] can be summarized asthe framework, "vacating room after encryption (VRAE)",as illustrated in Figure 1(a).In this framework, a contentowner encrypts the original image using a standard cipherwith an encryption key. After producing the encryptedimage, the content owner hands over it to a data hider andthe data hider can embed some auxiliary data into theencrypted image by losslessly vacating some roomaccording to a data hiding key.

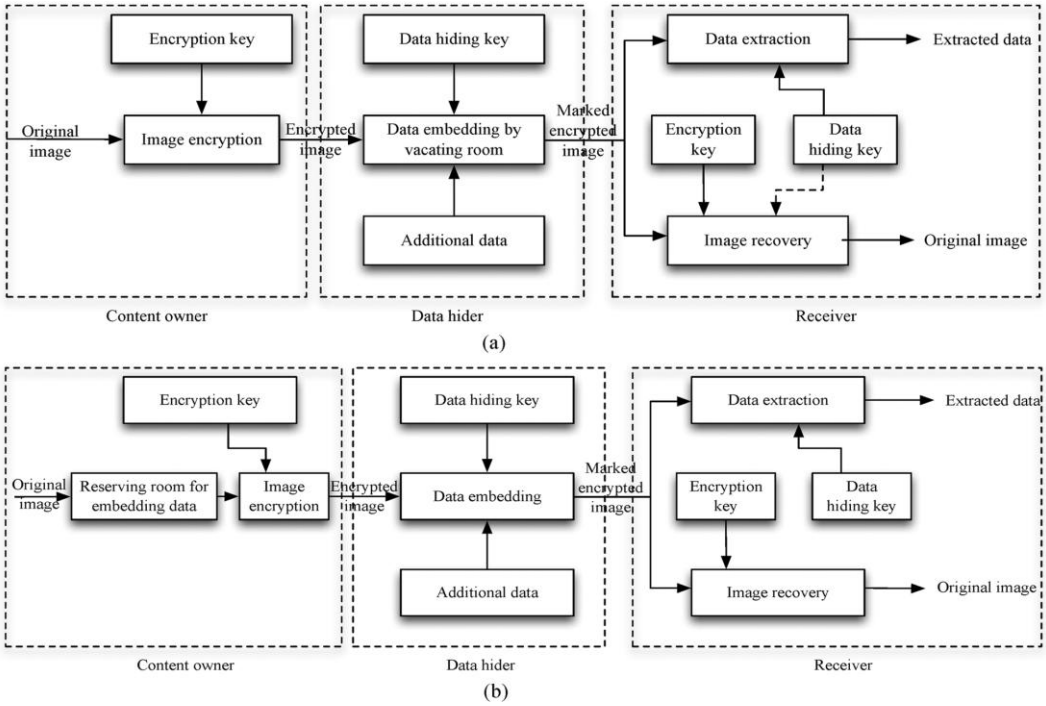

Figure 1: Previous works, (a) Showing without Reserving Room (b) Showing with Reserving Room Before Encryption.

Then a receiver, maybe the content owner himself orunauthorized third party can extract the embedded data withthe data hiding key and further recover the original imagefrom the encrypted version according to the encryption key.In all methods of [16]–[18], the encrypted 8-bit gray-scaleimages are generated by encrypting every bit-planes with astream cipher. The method in [16] segments the encryptedimage into a number of non-overlapping blocks sized by a xa each block is used to carry one additional bit. Additionalbit is considered as 0.To do this, pixels in each lock arepseudo-randomly divided into two sets S1 and S2 accordingto a data hiding key. To have decode or to recover the imagethey gets into trouble while doing mode function whenreached 8 the value coincides. Another risk of defeat of bitextraction and image recovery when divided block isrelatively small (e.g., a=8) or has much fine detailedtextures.

As shown in the figure 1(b) its holds good enough to havegood recovery capability and data exactly recovered [19].Reduced the loss of bit to the minimum, by reserving thepixels before the image is encrypted. It also allows the datato be extracted before and after the image is beingdecrypted. They too failed to have the security for the datawhat is hidden and also to have an authentication betweenthe sender and the receiver. In the proposed method sincethe data hider is getting a chance of using the encryptedimage, an authentication is produced in terms ofwatermarking. After the data is hidden there is a chance forattackers to get the sequentially hidden data. The proposedmethod explains how to hide data or message in randommanner and also to store a single character in eight differentparts or pixels using ASCII to 8-bit data conversion.

III. PROPOSED METHOD

In the proposed method the pixels in the image is vacatedaccording to the number of data to be hidden, which isdefined as reserving room. The image is then encryptedusing some algorithms. An authentication is added to theimage uses watermarking technology, the watermarkedimage is then send to the data hider to hide the data. Using arandom key a randomization of data is done for eachcharacter which is represented in 8 bits, they are thenrandomly hidden in the vacated space in the image usingreserving room algorithm and send to the receiver. Thereceiver first checks for the authentication which meansrecovering the watermark. Then using the random key thedata is retrieved. Or in other case the image is decrypted andthe data is retrieved.

3.1 Image Encryption

Assume the original image is in uncompressed format and each pixel with gray value falling into [0, 255] is represented by 8 bits.

$$B_{i,j,k} = b_{i,j,k} \oplus r_{i,j,k}$$

where ri,j,k are determined by an encryption key using a standard stream cipher. Then, B,i,j,k are concatenated orderly as the encrypted data. A number of secure stream cipher methods can be used here to ensure that anyone without the encryption key, such as a potential attacker or the data hider, cannot obtain any information about original content from the encrypted data.

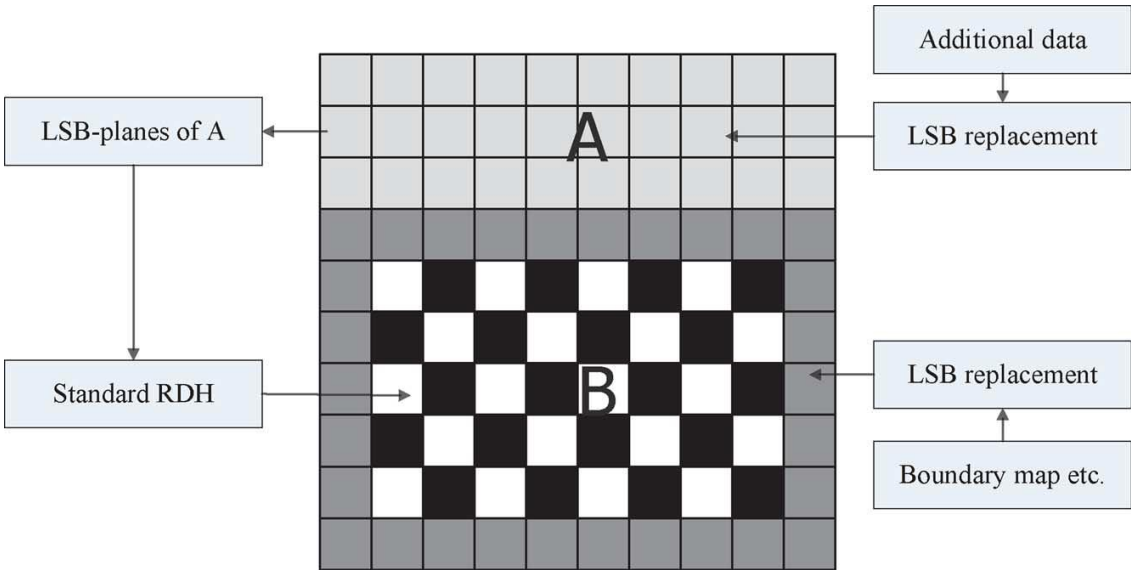

Fig. 2. Illustration of Image Partition and Embedding Process.

3.2 Data Embedding

In the data embedding phase, some parameters are embedded into a small number of encrypted pixels, and the LSB of the other encrypted pixels are compressed to create a space for accommodating the additional data and the original data at the positions occupied by the parameters.

The detailed procedure is as follows. According to a data-hiding key, the data-hider pseudo-randomlyselects Np encrypted pixels that will be used to carry the parameters for data hiding. Here, Np is a small positive integer, for example, Np=20. The other (N-Np) encrypted pixels are pseudo-randomly permuted and divided into a number of groups, each of which contains L pixels. The permutation way is also determined by the data-hiding key. For each pixel-group, collect the M least significant bits of the L pixels, and denote them as B (k,1) , B (k,2) …… B(k,M*L) where k is a group index within [1,(N-Np)/L] and M is a positive integer less than 5. The data-hider also generates a matrix G sized (M*L – S) * M*L, which is composed of two parts. The left part is the identity matrix and the right part is pseudo-random binary matrix derived from the data-hiding key.For each group , which is product with the G matrix to form a matrix of size (M * L-S). Which has a sparse bits of size S, in which the data is embedded and arrange the pixels into the original form and repermutated to form a original image.

3.3 Image Decryption

When having an encrypted image containing embedded data, a receiver firstly generates ri,j,k according to the encryption key, and calculates the exclusive-or of the received data and ri,j,k to decrypt the image. We denote the decrypted bits as b1i,j,k . Clearly, the original five most significant bits (MSB) are retrieved correctly. For a certain pixel, if the embedded bit in the block including the pixel is zero and the pixel belongs to S1, or the embedded bit is 1 and the pixel belongs to S0 , the data-hiding does not affect any encrypted bits of the pixel. So, the three decrypted LSB must be same as the original LSB, implying that the decrypted gray value of the

pixel is correct. On the other hand, if the embedded bit in the pixel's block is 0 and the pixel belongs to S0, or the embedded bit is 1 and the pixel belongs to S1.

$$\begin{aligned} b'_{i,j,k} &= r_{i,j,k} \oplus B'_{i,j,k} \\ &= r_{i,j,k} \oplus \overline{B_{i,j,k}} \\ &= r_{i,j,k} \oplus \overline{b_{i,j,k} \oplus r_{i,j,k}} \\ &= \overline{b_{i,j,k}}, \qquad\qquad k = 0,\ 1,\ 2. \end{aligned}$$

That means the three decrypted LSB must be different from the original LSB. In this case:

$$b'_{i,j,k} + b_{i,j,k} = 1,$$

3.4 Data Extraction andImage Recovery

Since data extraction is completely independent from imagedecryption, the order of them implies two different practical applications.

Fig. 3. (a) Original Lena, (b) its encrypted version, (c) encrypted image containing embedded data with embedding rate 0.017 bpp, and (d) directly decrypted version with PSNR 39.0 dB.

1) Case 1: Extracting Data From Encrypted Images:

Tomanage and update personal information of images which areencrypted for protecting clients' privacy, an inferior databasemanager may only get access to the data hiding key and have tomanipulate data in encrypted domain. The order of data extractionbefore image decryption guarantees the feasibility of ourwork in this

case.When the database manager gets the data hiding key, he candecrypt the LSB-planes of**B** and extract the additional data**m**by directly reading the decrypted version. When requesting forupdating information of encrypted images, the database manager,then, updates information through LSB replacement andencrypts updated information according to the data hiding keyall over again. As the whole process is entirely operated on encrypteddomain, it avoids the leakage of original content.

2) Case 2: Extracting Data FromDecrypted Images:

In Case1, both embedding and extraction of the data are manipulated inencrypted domain. On the other hand, there is a different situationthat the user wants to decrypt the image first and extracts thedata from the decrypted image when it is needed. The followingexample is an application for such scenario. Assume Alice outsourcedher images to a cloud server, and the images are encryptedto protect their contents. Into the encrypted images, thecloud server marks the images by embedding some notation,including the identity of the images' owner, the identity of thecloud server and time stamps, to manage the encrypted images.Note that the cloud server has no right to do any permanentdamage to the images. Now an authorized user, Bob who hasbeen shared the encryption key and the data hiding key, downloadedand decrypted the images. Bob hoped to get marked decryptedimages, i.e., decrypted images still including the notation,which can be used to trace the source and history of thedata. The order of image decryption before/without data extractionis perfectly suitable for this case.

Table I Psnr Comparison For Three Different Lsb-Plane Choices Under Various Embedding Rates

		PSNR results (dB)							
embedding rate (bpp)		0.005	0.01	0.05	0.1	0.2	0.3	0.4	0.5
Lena	1 LSB-plane	67.16	63.44	55.46	52.33	49.07	45.00	40.65	35.84
	2 LSB-planes	66.48	62.65	54.69	51.55	48.39	45.10	42.56	39.46
	3 LSB-planes	64.41	60.94	52.95	49.96	46.79	43.98	41.91	39.53
Airplane	1 LSB-plane	65.94	63.18	57.02	54.20	50.98	48.26	44.67	40.78
	2 LSB-planes	65.48	62.33	55.91	53.05	49.87	48.10	45.05	42.73
	3 LSB-planes	63.47	60.97	53.87	50.79	47.65	45.79	43.88	42.19
Barbara	1 LSB-plane	65.39	62.56	55.56	51.46	47.68	43.56	39.24	34.80
	2 LSB-planes	65.00	61.85	54.72	50.71	47.25	43.70	40.78	37.53
	3 LSB-planes	63.33	60.50	53.16	49.36	45.98	42.81	40.34	37.58
Baboon	1 LSB-plane	57.49	55.71	50.19	46.17	40.68	35.87	31.16	25.92
	2 LSB-planes	57.43	55.47	49.87	45.92	40.41	36.47	33.08	29.85
	3 LSB-planes	57.10	55.13	49.23	45.40	40.09	36.33	32.96	30.19
Peppers	1 LSB-plane	63.77	61.30	54.17	51.02	46.00	42.08	36.91	——
	2 LSB-planes	63.67	60.53	53.50	50.50	46.16	42.65	39.47	35.76
	3 LSB-planes	62.34	59.54	52.22	49.18	45.43	42.10	39.40	36.87
Boat	1 LSB-plane	67.22	64.13	56.75	52.62	49.10	45.21	41.24	35.99
	2 LSB-planes	66.72	63.26	55.75	51.71	48.40	44.98	42.46	39.98
	3 LSB-planes	64.57	61.34	53.73	50.02	46.71	43.81	41.70	39.46

IV. IMPLEMENTATION ISSUES

The proposed approach will be tested on public availablestandard images, which include "Lena", "Airplane", "Barbara","Baboon", "Peppers" and "Boat" [19]. The size of all imagesis 512 512 8. The objective criteria PSNR is employed toevaluate the quality of marked decrypted image quantitatively.

V. EXPERIMENTAL ANALYSIS AND RESULTS

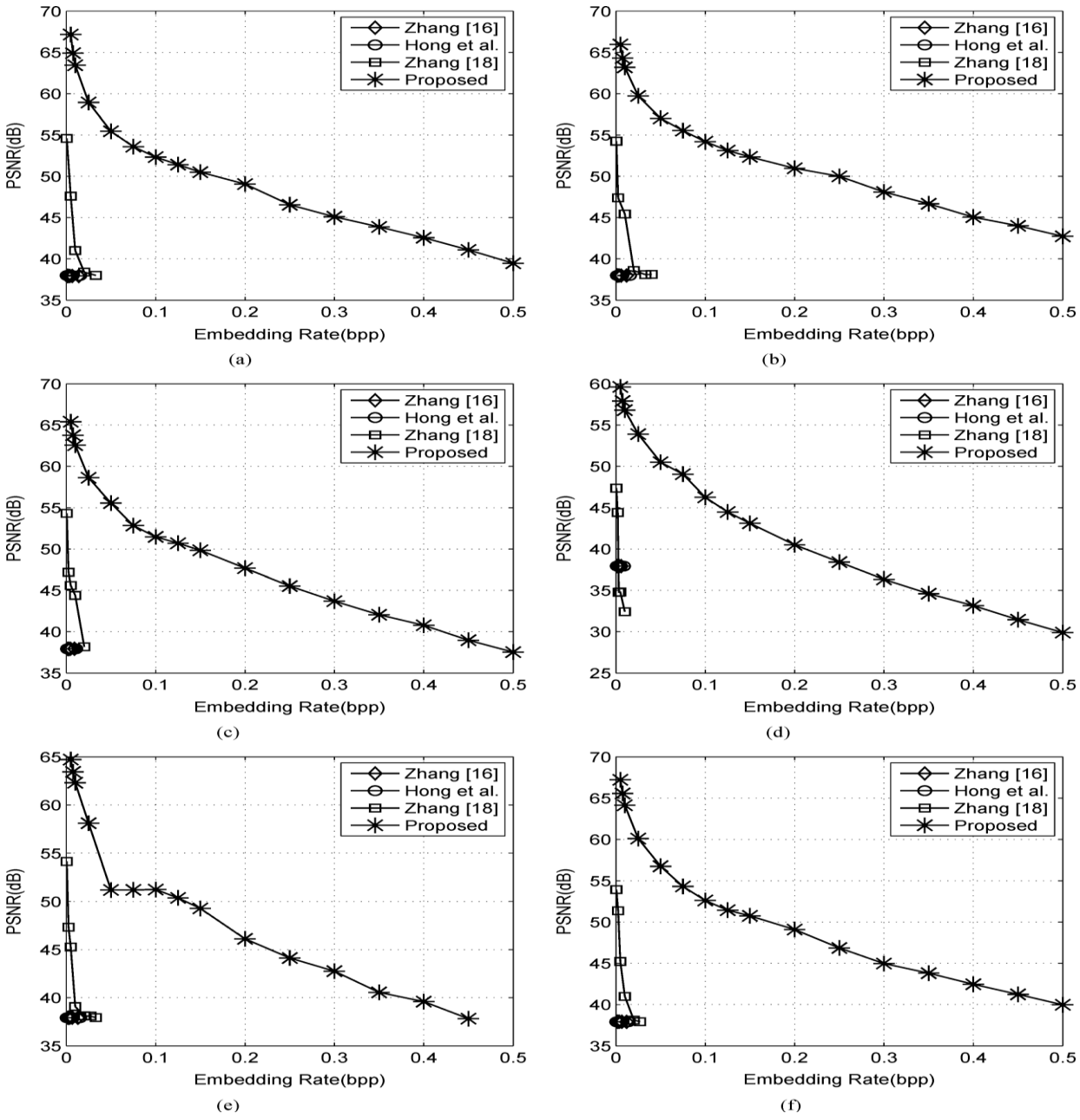

Fig. 4. PSNR Comparison with the Methods of Zhang [16], Hong [17] and Zhang [18]. (a) Lena, (b) Airplane, (c) Barbara, (d) Baboon, (e) Peppers, (f) Boat.

VI. CONCLUSION

In this paper, a novel scheme for separable reversible data hiding in encrypted image is proposed, which consists of image encryption, data embedding and data-extraction/image recovery phases. In the first phase, the content

owner encrypts the original uncompressed image using an encryption key. Although a data-hider does not know the original content, he can compress the least significant bits of the encrypted image using a data-hiding key to create a sparse space to accommodatethe additional data. With an encrypted image containingadditional data, the receiver may extract the additional data using only the data-hiding key, or obtain an image similar to the original one using only the encryption key. When the receiver has both of the keys, he can extract the additional data and recover the original content without anyerror by exploiting the spatial correlation in natural image if the amount of additional data is not too large. If the lossless compression method is used for the encrypted image containing embedded data, the additional data can be still extracted and the original content can be also recovered since the lossless compression does not change the content of the encrypted image containing embedded data. However, the lossy compression method in compatible with encrypted images generated by pixel permutation is not suitable here since the encryption isperformed by bit-XOR operation. In the future, a comprehensive combination of imageencryption and data hiding compatible with lossy compression deserves further investigation.

REFERENCES

[1] M. Johnson, P. Ishwar, V. M. Prabhakaran, D. Schonberg, and K.Ramchandran, "On compressing encrypted data," IEEE Trans. Signal Process., vol. 52, no. 10, pp. 2992–3006, Oct. 2004.

[2] W. Zhang, B. Chen, and N. Yu, "Capacity-approaching codes forreversible data hiding," in Proc 13th Information Hiding (IH'2011),LNCS 6958, 2011, pp. 255–269, Springer-Verlag.

[3] W. Zhang, B. Chen, and N. Yu, "Improving various reversible datahiding schemes via optimal codes for binary covers," IEEE Trans.Image Process., vol. 21, no. 6, pp. 2991–3003, Jun. 2012.

[4] J. Fridrich and M. Goljan, "Lossless data embedding for all image formats,"in Proc. SPIE Proc. Photonics West, Electronic Imaging, Securityand Watermarking of Multimedia Contents, San Jose, CA, USA,Jan. 2002, vol. 4675, pp. 572–583.

[5] J. Tian, "Reversible data embedding using a difference expansion,"IEEE Trans. Circuits Syst. Video Technol., vol. 13, no. 8, pp. 890–896,Aug. 2003.

[6] Z. Ni, Y. Shi, N. Ansari, and S. Wei, "Reversible data hiding," IEEETrans. Circuits Syst. Video Technol., vol. 16, no. 3, pp. 354–362, Mar.2006.

[7] D.M. Thodi and J. J. Rodriguez, "Expansion embedding techniques forreversible watermarking," IEEE Trans. Image Process., vol. 16, no. 3,pp. 721–730, Mar. 2007.

[8] X. L. Li, B. Yang, and T. Y. Zeng, "Efficient reversible watermarkingbased on adaptive prediction-error expansion and pixel selection,"IEEE Trans. Image Process., vol. 20, no. 12, pp. 3524–3533, Dec.2011.

[9] P. Tsai, Y. C. Hu, and H. L. Yeh, "Reversible image hiding schemeusing predictive coding and histogram shifting," Signal Process., vol.89, pp. 1129–1143, 2009.

[10] L. Luo et al., "Reversible imagewatermarking using interpolation technique,"IEEE Trans. Inf. Forensics Security, vol. 5, no. 1, pp. 187–193,Mar. 2010.

[11] V. Sachnev, H. J. Kim, J. Nam, S. Suresh, and Y.-Q. Shi, "Reversiblewatermarking algorithm using sorting and prediction," IEEE Trans.Circuits Syst. Video Technol., vol. 19, no. 7, pp. 989–999, Jul. 2009.

[12] X. Zhang, "Reversible data hiding in encrypted image," IEEE Signal Process. Lett., vol. 18, no. 4, pp. 255–258, Apr. 2011.

[13] J. Tian, "Reversible data embedding using a difference expansion,"IEEE Trans. Circuits Syst. Video Technol., vol. 13, no. 8, pp. 890–896, Aug. 2003.

[14] Z. Ni, Y.-Q. Shi, N. Ansari, and W. Su, "Reversible data hiding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 3, pp. 354–362, Mar. 2006.

[15] M. U. Celik, G. Sharma, A. M. Tekalp, and E. Saber, "Lossless generalized- LSB data embedding," *IEEE Trans. Image Process.*, vol. 14, no. 2, pp. 253–266, Feb. 2005.

[16] 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 Process.*, vol. 90, pp. 2911–2922, 2010.

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

[18] A. Mayache, T. Eude, and H. Cherifi, "A comparison of image quality models and metrics based on human visual sensitivity," in *Proc. Int. Conf. Image Processing (ICIP'98)*, Chicago, IL, 1998, vol. 3, pp. 409–413. [19] Z. Wang and A. C. Bovik, "A universal image quality index," *IEEE Signal Process. Lett.*, vol. 9, no. 1, pp. 81–84, Jan. 2002.

[19] Miscelaneous Gray Level Images [Online]. Available: http://decsai.ugr.es/cvg/dbimagenes/g512.php.