International Journal of Research Studies in Science, Engineering and Technology
Volume 1, Issue 9, December 2014, PP 55-65
ISSN 2349-4751 (Print) & ISSN 2349-476X (Online)

An Efficient Reversible Data Hiding for High quality Images Using Histogram Modification Method

***D.Bhavani **Ravikiran**

*PG Student, Chalapathi Institute of Technology, Mothadaka, Guntur.
**Assoc Professor, Chalapathi Institute of Technology, Mothadaka,Guntur

Abstract: *A Reversible Data Hiding (RDH) is a technique which can retrieve both the cover image and hidden data without any distortion from the watermarked image. In this paper, a new reversible data hiding scheme is proposed based on two-dimensional difference histogram modification by using difference-pair-mapping (DPM). DPM is a new technique and it is an injective mapping defined on difference-pairs. This technique is a natural extension of expansion embedding and shifting techniques. By using this technique, the image redundancy can be better exploited by DPM and an improved embedding performance is achieved. In addition with DPM, a pixel-pair-selection strategy is also adopted to priory used pixel-pairs located in smooth image regions to embed data. This leads to further enhancement in the embedding performance.*

Keywords: *Difference-pair-mapping (DPM), histogram modification, reversible data hiding (RDH), twodimensional difference-histogram*

1. INTRODUCTION

Reversible data concealing is a data hiding scheme which is widely used for covert communication. In this technique, the secret information is concealed into a cover media by slightly modifying its pixel values and the embedded message as well as the original cover image should be completely recovered from the watermarked image

[1]. RDH is a special type of information hiding and its feasibility is mainly due to the lossless compressibility of natural images. The reversibility in RDH is quite desirable and helpful in some practical applications such as medical image processing [2], multimedia archive management, image trans-coding, and video error-concealment coding, etc. Generally, the performance of a RDH scheme is assessed by the behaviour of capacity-distortion. For a required embedding capacity (EC), to obtain a good marked image quality, one expects to reduce the embedding distortion as much as possible. Many RDH methods have been proposed so far, e.g., the methods based on lossless compression, difference expansion [4], histogram modification [5], [6], prediction-error expansion [9], and integer transform, etc. Among these methods, the histogram-based methods are widely used. The histogram-based methods modify thehistogram in such a way that certain bins are shifted to create vacant space while some other bins are utilized to carry data by filling the vacant space. This type of methods can well control the embedding distortion and provide a sufficient EC. The first histogram-based RDH method is the one proposed by Ni *et al.* in [16]. This method uses peak and minimum points of the pixel-intensity-histogram to embed data. It changes each pixel value at most by 1, and thus a good marked image quality can be obtained. However, its EC is quite low and this method does not work well if the cover image has a flat histogram. To alleviate it, Lee *et al.* [5] proposed to utilize the difference-histogram instead. This novel method exploits the correlation among neighbouring pixels and can embed larger payload with reduced distortion compared with Ni *et al.*'s. Moreover, Lee *et al.*'s method works by modifying the two-dimensional pixel-intensityhistogram by using pixel-pair-mapping (PPM). The PPM is a mapping technique which is an injective mapping defined on pixel-pair. Afterwards, Fallahpour [6] introduced a method by modifying the histogram of prediction-error. Except these aforementioned methods, many other works are also based on histogram by incorporating some strategies such as double-layered embedding [10], [11], embedding-position-selection [9], [10], [12], adaptive embedding [9], contextmodification [22], etc.

The histogram-based RDH methods generally contain two basic steps:

- Histogram generation

- Histogram modification

In the first step, correlation of the local image is simplified to a one-dimensional statistic. Clearly, by this simplification, the redundancy of the image cannot be fully exploited and it only contributes to the second step since a one-dimensional histogram is easy to deal with. Based on this consideration, instead of one-dimensional histogram used in previous RDH methods and to better exploit the image redundancy, a novel Reversible Data Concealing scheme by using a two-dimensional difference-histogram is used. For the proposed method, by considering a pixel-pair and its context, a local image region is projected to a two-dimensional space to obtain a sequence consisting of difference-pairs. Then, a two-dimensional difference-histogram is generated by counting the difference-pairs. Finally, reversible data embedding is implemented according to a specifically designed difference-pair-mapping (DPM). Here, the DPM is an injective mapping defined on difference-pairs, and it is a natural extension of expansion embedding and shifting techniques used in current histogram-based methods. By using the two-dimensional difference-histogram and this specific DPM, compared with the conventional one-dimensional histogram based methods, more pixels are used for carrying data while the number of shifted pixels is reduced as well, and thus an improved embedding performance is achieved. In addition, inspired by the embeddingposition selection techniques introduced in previous works [9], a pixel-pair-selection strategy is adopted in the proposed method to priorly use the pixel-pairs located in smooth image regions to embed data. This may further enhance the embedding performance.

2. PROPOSED METHOD

A. Main Idea

In our method, we first divide the host image into non-overlappingblocks such that each block contains n pixels(e.g., n = 1 for Ni et al.'s method, and n = 2 for Lee etal.'s method). Then, an n-dimensional histogram is generatedby counting the frequency of the pixel-value-array sized n ofeach divided block. Finally, data embedding is implementedby modifying the resulting n-dimensional histogram. Noticethat the pixel-value-array is an element of Zn, we then needto divide Zn into two disjointedsets, one set is used tocarry hidden data by expansion embedding, and the otherset is simply shifted to create vacant spaces to ensure thereversibility. We now present our new approach.

Let S and T be a partition of Zn: S∪T = Zn and S∩T = ∅.Suppose that three functions g : T → Zn, f0 : S → Zn andf1 : S → Zn satisfy the following conditions:C1: The functions g, f0 and f1 are injective.C2: The sets g(T), f0(S) and f1(S) are disjointed with eachother.

Here, g is called "shifting function" and will be used to shiftpixel values, f0 and f1 are called "embedding functions" andwill be used to embed data. More specifically, each blockwith value x ∈ T will be shifted to g(x), and the block withvalue x ∈ S will be expanded to either f0(x) or f1(x) tocarry one data bit. The shifting and embedding functions willgive a HS-based RDH algorithm where the reversibility canbe guaranteed by the conditions C1 and C2.

The underflow/overflow is an inevitable problem of RDH,i.e., for gray-scale image, the shifted and expanded valuesshould be restricted in the range of [0, 255]. To deal with this,the above defined sets T and S need be further processed. Let

$$A_n = \{\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{Z}^n : 0 \leq x_i \leq 255\}$$

be the set of all pixel-value-arrays of length n of gray-scaleimage. We define

$$\begin{aligned} T_s &= A_n \cap g^{-1}(A_n) \\ S_e &= A_n \cap f_0^{-1}(A_n) \cap f_1^{-1}(A_n) \\ T_{u,o} &= A_n \cap T - T_s \\ S_{u,o} &= A_n \cap S - S_e. \end{aligned}$$

The sub-indices “s”, “e” and “u, o” mean “shift”, “embed”and “underflow/overflow”, respectively. Obviously, the foursets Ts , Se, Tu,o and Su,o are disjointed with each other andconstitute a partition of An, i.e.,

$$A_n = T_s \cup S_e \cup T_{u,o} \cup S_{u,o}.$$

Moreover, the sets g(Ts), f0(Se) and f1(Se) are containedin An and the condition C2 ensures that they are alsodisjointed.

By definitions (9)-(12), each block with value x ∈Ts willbe shifted, each block with value x ∈ Se will be expanded tocarry one data bit, and the block with value x ∈Tu,o∪Su,owill remain unchanged since it cannot be shifted or expandeddue to underflow/overflow.

Before further describing our method, we give an examplefor better understanding the definitions above. More exampleswill be given in Section IV. Take n = 1 and for an integera ∈ {1,…, 253}, we define

$$\begin{cases} S = \{a, a+1\} \text{ and } T = \mathbb{Z} - S \\ g(x) = \begin{cases} x-1, \text{ if } x < a \\ x+1, \text{ if } x > a+1 \end{cases} \\ f_0(a) = a, \;\; f_0(a+1) = a+1 \\ f_1(a) = a-1, \;\; f_1(a+1) = a+2. \end{cases}$$

One can verify that (g, f0, f1) satisfy the conditions C1 andC2. Particularly, the sets defined in (9)-(12) can be detailed asfollows: Ts = {1,…, a −1}∪{a+2,…, 254}, Se = {a, a+1},Tu,o = {0, 255}, and Su,o = ∅. We will see that this specificconstruction corresponds the modified version of Ni et al.’smethod introduced in Section II.

We now briefly describe how to reversibly embed data byusing the shifting and embedding functions (see Fig. 2 forillustration). Consider a pixel block of length n whose valueis x ∈ An.

1) If x ∈Ts , the block does not carry any hidden data andits value is simply shifted to g(x) ∈ An. For instance, inthe example (14), the pixel value in Ts = {1,…, a−1}∪{a+2,…, 254} needs to be shifted and the shifted valuebelongs to the set g(Ts) = {0,…, a−2}∪{a+3,…, 255}.

2) If x ∈ Se, the new pixel value is taken as fm(x) ∈Anwhere m ∈ {0, 1} is the corresponding data bit to beembedded. In this situation, one data bit is embeddedinto the block. Since the sets g(Ts), f0(Se) and f1(Se)are disjointed with each other, decoder can distinguishthe blocks used for shifting from those for embeddingdata. As a result, by the condition C1, decoder canrecover the original pixel value and extract the embeddeddata bit. For instance, in the example (14), the pixelvalue in Se = {a, a + 1} is expanded to carry onedata bit, and the expanded value belongs to the setf0(Se) ∪ f1(Se) = {a, a + 1} ∪ {a − 1, a + 2} ={a − 1, a, a + 1, a + 2}.

3) If x ∈Tu,o, we know that g(x) /∈ An. In this case,to prevent underflow/overflow, we do nothing with x.Regarding the example (14), the pixel will not bechanged if its value belongs to Tu,o = {0, 255}.

4) If x ∈Su,o, we see either f0(x) /∈ An or f1(x) /∈ An. Inthis case, the block will also remain unchanged. Yet, theset Tu,o∪Su,o may overlap with g(Ts)∪ f0(Se)∪ f1(Se).Then we use a location map to record the locations ofblocks whose values belong to Tu,o∪Su,o. In the example(14), the pixel whose value is 0 or 255 needs to berecorded in the location map.

In summary, with shifting and embedding functions, we canobtain a HS-based RDH algorithm. The detailed embeddingand extraction procedures are given below.We take T Su = Tu,o∪Su,o for simplicity in the followingcontext.

B. Data Embedding

The embedding procedure contains several steps. First, afterdividing the host image into non-overlapping blocks, theblocks are further divided into three parts to get I1, I2 and I3.Then, by using shifting and embedding functions, embed thehidden data into I1 and I3. Next, by using LSB replacement,embed the location map which records the underflow/overflowlocations into I1. Notice that, before replacing LSBs, theoriginal LSBs of I1 should be recorded into a LSB sequence.Finally, embed the LSB sequence into I2 using shifting andembedding functions.

Here, the partition of three parts is to solve the underflow/overflow problem by embedding the location map intothe host image. The part I1 is double embedded to embedfirst the hidden data (using shifting and embedding functions)and then the location map (using LSB replacement).The detailed data embedding procedure is described asbellow.

Step 1: Divide the host image into k non-overlapping blocks{X1,…,Xk } such that each Xi contains n pixels. Assume thevalue of Xi is xi ∈ An.

Step 2: Define the location map LM: LM(i) = 0 if xi ∈Ts∪ Se, and LM(i) = 1 if xi ∈ T Su . Clearly, LM is a binarysequence of length k. Denote k=log2(k + 1), where k·is the ceiling function. Take l = {i : LM(i) = 1} which isthe amount of underflow/overflow blocks. Then we define abinary sequence LMc of length lc = (l + 1)k to record allunderflow/overflow locations.

1) (LMc(1),…, LMc(k)) is the binary representation of l.

2) For each j ∈ {1,…, l}, (LMc(jk +1),…, LMc(jk+k))is the binary representation of i, where i is the j -th indexsuch that LM(i) = 1.

Step 3: Divide the k blocks into three parts to get I1, I2and I3.

1) I1 = {X1,…, Xk1} with k1 = lcn

2) I2 = {Xk1+1,…, Xk1+k2} such that it contains exactlylc embeddable blocks. Here, a block is called “embeddable”if its value belongs to Se.

3) I3 = {Xk1+k2+1,…,Xk } is the set of rest blocks.

Step 4: Embed the hidden data into I1 and I3, i.e., for anyi∈ {1,…, k1, k1 + k2 + 1,…, k}.

1) If xi ∈Ts , replace the value of Xi by g(xi).

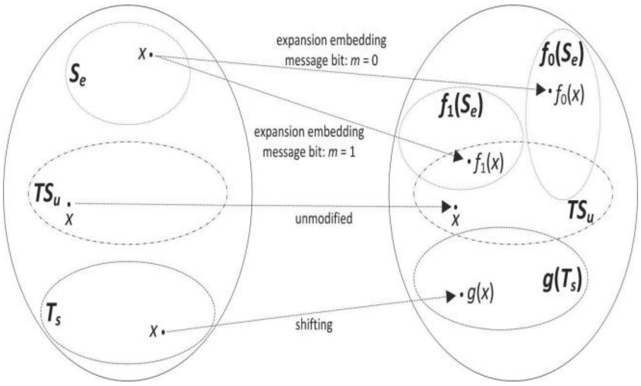
2) If xi ∈ Se, replace the value of Xi by fm(xi), wherem ∈ {0, 1} is the data bit to be embedded.

3) If xi ∈ T Su, we do nothing with Xi .Step 5: Record LSBs of the first lc pixels of I1 to get a

binary sequence SLSB (recall that I1 contains nk1 = nlcn ≥ lcpixels). Then replace these LSBs by the sequence LMc definedin Step 2.

Step 6: Embed the sequence SLSB into I2 in the same way asStep 4. Since the length of SLSB is lc, SLSB can be embeddedexactly into the embeddable blocks of I2. Finally, we get themarked image.

The core of this procedure is Step 4 and the partition inStep 3 is to deal with the underflow/overflow problem.It should be mentioned that there is another commonlyused way to encode the location map: in Step 2, one can getLMc by losslessly compressing LM. However, in HS-basedRDH algorithms, there are usually only a few blocks whichmay cause underflow/overflow. We then prefer to record thoseproblematic locations rather than compressing the locationmap because the latter solution is time-consuming.



C. Data Extraction

The data extraction procedure also contains several steps.First, the same as the data embedding, divide the markedimage blocks into three parts to get I1, I2 and I3. Then,determine the location map by reading LSBs of I1. Next,according to the location map and by using shifting andembedding functions, determine the LSB sequence (definedin Step 5 of data embedding) by extracting data from I2, andthen replace the LSBs of I1 by the extracted LSB sequence.Finally, extract the embedded data from I1 and I3. Notice that,using shifting and embedding functions, the image restorationcan be realized simultaneously with the data extraction.The detailed data extraction procedure is described as bellow.

Step 1: The same as Step 1 of data embedding, dividethe marked image into k non-overlapping blocks {Y1,…,Yk }.Assume the value of Yi is yi∈ An.

Step 2: Firstly, determine the amount of problematic locations,l, by reading LSBs of the first k =log2(k+1) pixels.

Secondly, read LSBs of the first lc = (l + 1)k pixels to getthe sequence LMc. Then we can get the location map LM.Finally, with k1 = lcn, LM, and by identifying embeddableblocks, we can obtain the same partition as defined in Step 3of data embedding.

Step 3: Extract data from I2 and recover original pixelvalues of I2, i.e., for any i∈ {k1 + 1,…, k1 + k2}.

1) If LM(i) = 0 and yi∈ g(Ts), the original pixel valueis g−1(yi) and there is no embedded data.

2) If LM(i) = 0 and yi∈fm(Se) holds for a certainm ∈ {0, 1}, the original pixel value is f −1m (yi) and theembedded data bit is m.

3) If LM(i) = 1, the original pixel value is yi itself andthere is no embedded data.

The sequence SLSB defined in Step 5 of data embedding isextracted in this step.

Step 4: Replace LSBs of the first lc pixels of I1 by SLSB.

Step 5: Extract the embedded hidden data and recoveroriginal pixel values in I1 ∪ I3 in the same way as Step 3.

Finally, the hidden data is extracted and the original image isrestored.

D. Embedding Capacity

According to the data embedding procedure, the EC of ourmethod can be formulated as:

$$\sum_{\mathbf{x}\in S_e} h_n(\mathbf{x}) - \lceil \log_2(k+1) \rceil \sum_{\mathbf{x}\in TS_u} h_n(\mathbf{x}) - \lceil \log_2(k+1) \rceil$$

wherehn is the n-dimensional histogram: hn(x) = {i : xi =x}.
Notice that Ts , Se, T Su constitute a partition of An. Then,

$$\sharp A_n = \sharp T_s + \sharp S_e + \sharp TS_u .$$

Moreover, recall that g(Ts), f0(Se) and f1(Se) are disjointedwith each other and they are contained in An. Hence,

$$\sharp A_n \geq \sharp g(T_s) + \sharp f_0(S_e) + \sharp f_1(S_e).$$

By the condition C1, we have Ts = g(Ts) and Se = f0(Se) = f1(Se). Consequently, by comparing (16) with(17), we see that

$$\sharp T_s + \sharp S_e + \sharp TS_u \geq \sharp T_s + \sharp S_e + \sharp S_e .$$

Hence, T Su ≥ Se holds. Thus, reviewing (15), sincelog2(k +1) 1) is a positive constant larger than 1, to successfully embed data and increase EC, we should carefully design(S, T) and (g, f0, f1) such that hn(x) is large for x ∈ Se whilesmall for x ∈ T Su. That is to say, we should use the mostfrequent bins in the histogram to embed data and meanwhileavoid underflow/overflow.

Simulation Results

a) Designing Window

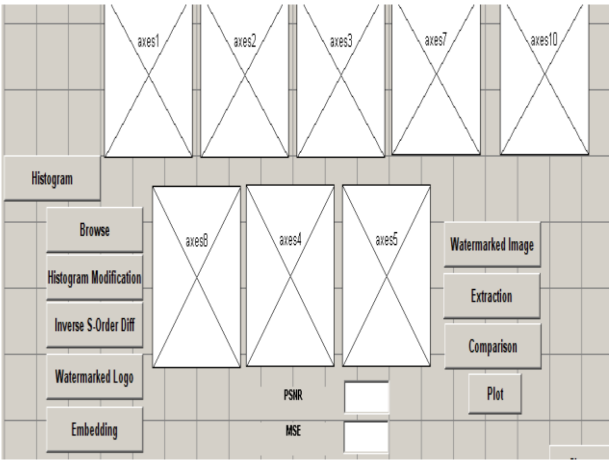


Fig1. *Application Window*

The above figure shows that to create the designing window using MATLAB (GUIDE) Tool box widgets.

b) Application Window.

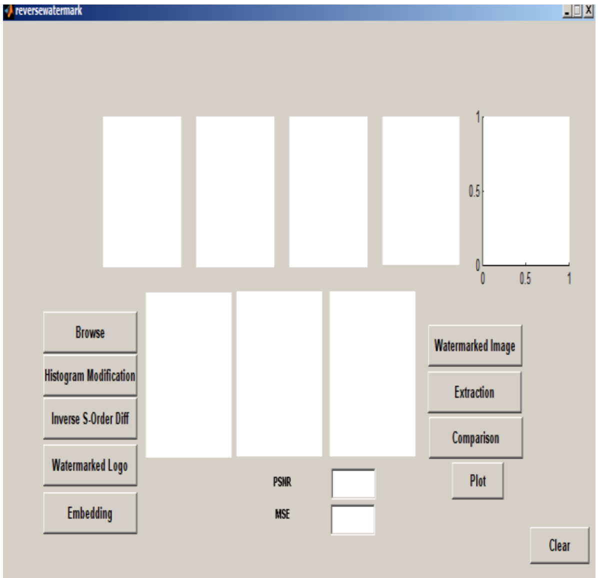
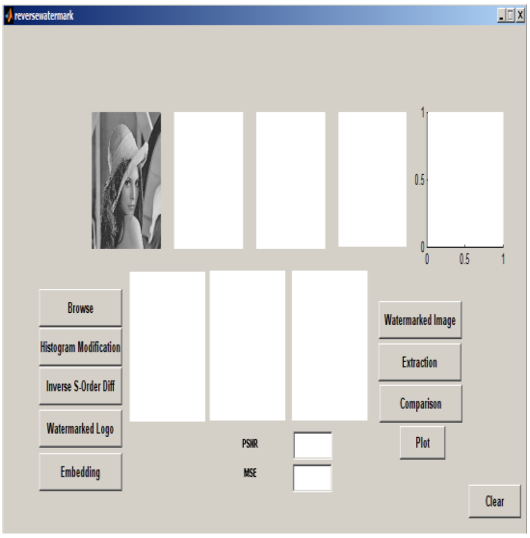


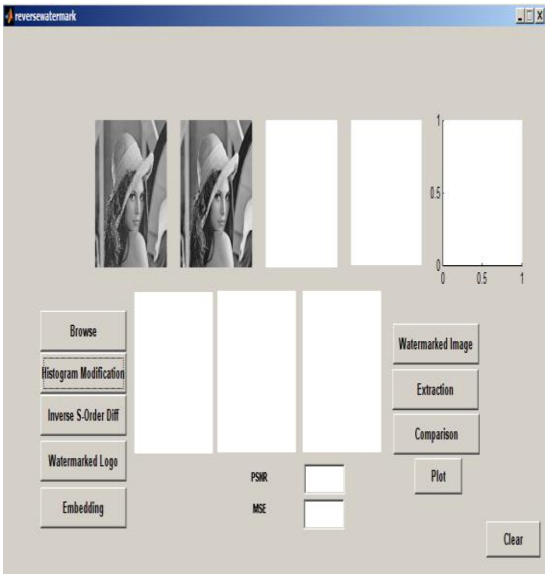
Fig2. *Application Window*

The above figure shows that application window. The designing window to generate this one defiantly

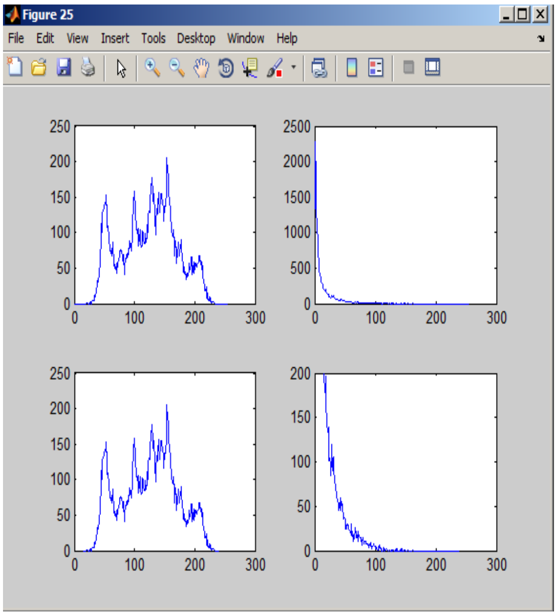
c) Original Image



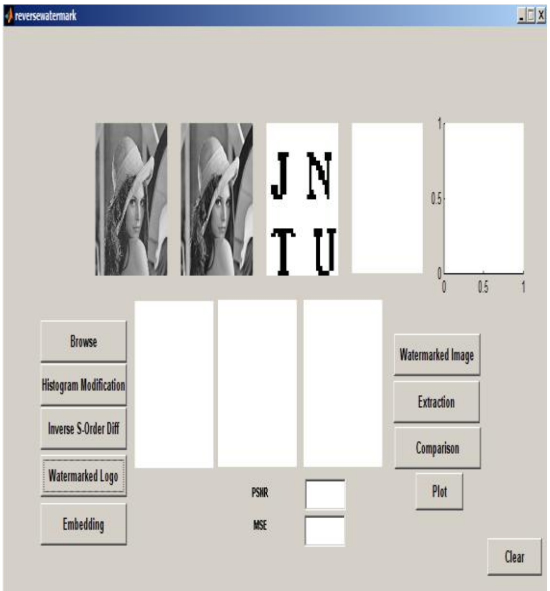
d) Histogram Modification



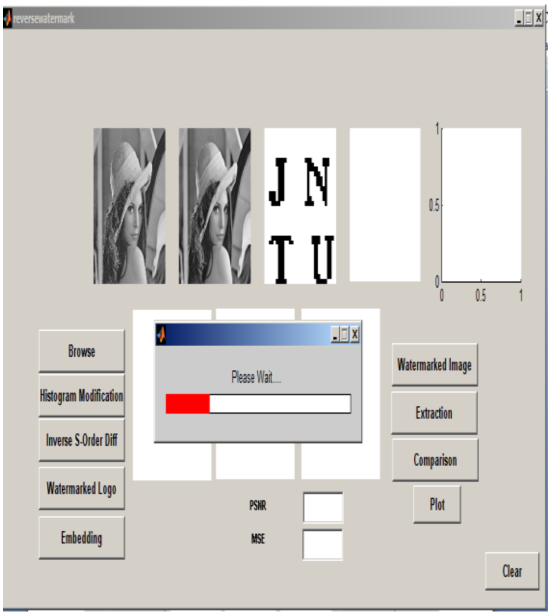
e) Inverse S-Order Difference Image



f) Watermarked Logo Image



g) Embedded Process

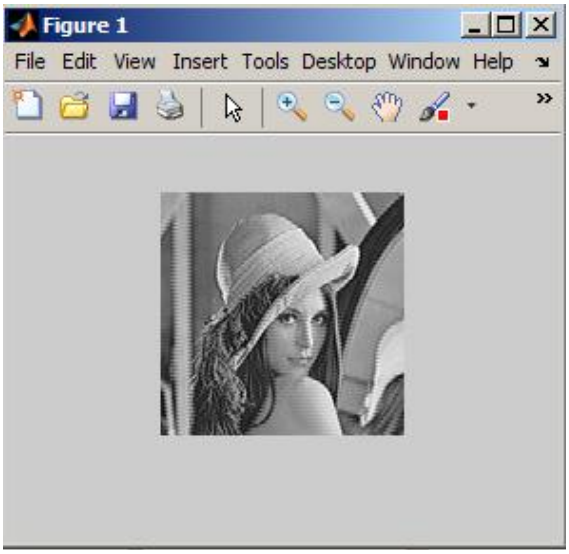


Please Wait.....

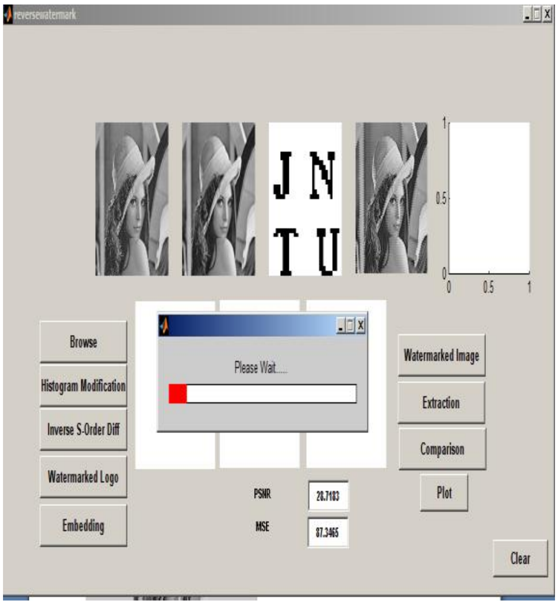
PSNR

MSE

h) Embedded Image or Watermarked Image



i) Extraction process P

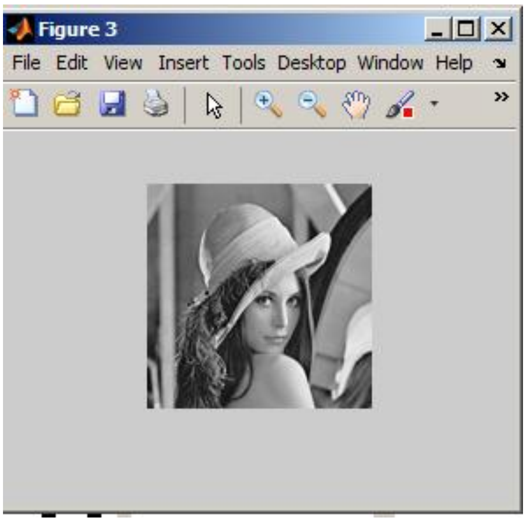


Please Wait.....

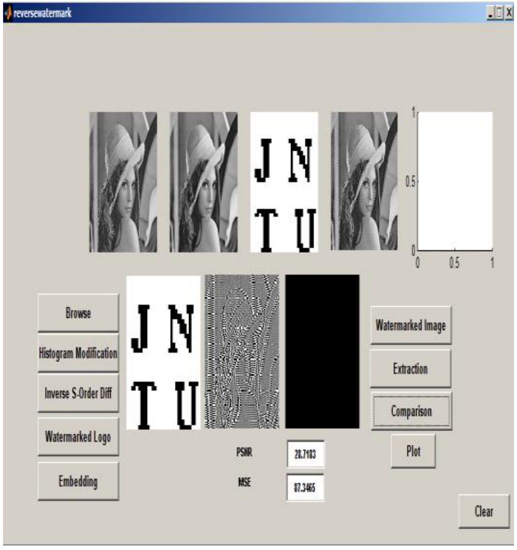
PSNR 28.7183

MSE 87.3465

j) Extracted Image



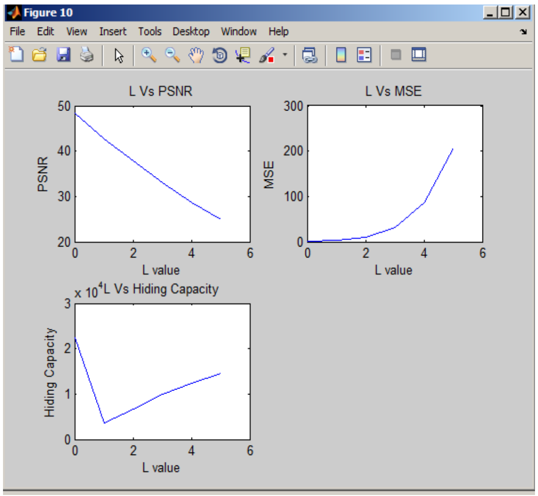
k) Validations (PSNR & MSE)



PSNR 28.7183

MSE 87.3465

l) Measurements (Plotting)



3. CONCLUSION

In this paper, by revisiting existing algorithms, a generalframework to construct HS-based RDH is proposed. Accordingto our framework, to obtain a RDH algorithm, one justneeds to define the shifting and embedding functions. Thiswork will facilitate the design of RDH. Furthermore, byincorporating our framework with the PEE and pixel selectiontechniques, two novel RDH algorithms are also introduced.These algorithms can achieve a better performance comparedwith the state-of-the-art works. So the proposed framework hasa potential to provide excellent RDH algorithms. However,thought the proposed framework may design different RDHalgorithms, it has also limitations. Some HS-based algorithmssuch as the one based on adaptive embedding [20] and thelocation-map-free methods [24], [36] cannot be derived by theproposed framework.

In future, to push forward the capacity-distortion behaviorof RDH, more meaningful shifting and embedding functionsare expected. Moreover, since the typical two-dimensionalhistogram based methods [10], [12], [33] are special casesof the proposed framework, a direct question is, based ontwo-dimensional histogram (i.e., by taking n = 2 in ourframework), how to determine the optimized shifting andembedding functions such that the embedding distortion isminimized for a given EC. This is also an interesting directionfor future work.

REFERENCES

[1] G. Coatrieux, C. L. Guillou, J. M. Cauvin, and C. Roux, “Reversiblewatermarking for knowledge digest embedding and reliability controlin medical images,” IEEE Trans. Inf. Technol. Biomed., vol. 13, no. 2,pp. 158–165, Mar. 2009.

[2] S. Lee, C. D. Yoo, and T. Kalker, “Reversible image watermarkingbased on integer-to-integer wavelet transform,” IEEE Trans. Inf. Forens.Security, vol. 2, no. 3, pp. 321–330, Sep. 2007.

[3] R. Li, O. C. Au, C. K. M. Yuk, S. Yip, and T. Chan, “Enhanced imagetrans-coding using reversible data hiding,” in Proc. IEEE Int. Symp.Circuits Syst., May 2007, pp. 1273–1276.

[4] R. Caldelli, F. Filippini, and R. Becarelli, “Reversible watermarkingtechniques: An overview and a classification,” Eur. Assoc. SignalProcess. J. Inf. Security, vol. 2010, no. 2, pp. 1–19, 2010.

[5] J. Fridrich, M. Goljan, and R. Du, “Lossless data embedding-newparadigm in digital watermarking,” Eur. Assoc. Signal Process. J. Appl.Signal Process., vol. 2002, no. 2, pp. 185–196, Feb. 2002.

[6] T. Kalker and F. M. J. Willems, “Capacity bounds and constructions forreversible data hiding,” Security Watermarking Multimedia Contents V,vol. 5020, pp. 604–611, Jun. 2003.

[7] M. U. Celik, G. Sharma, A. M. Tekalp, and E. Saber, “Losslessgeneralized-LSB data embedding,” IEEE Trans. Image Process., vol. 14,no. 2, pp. 253–266, Feb. 2005.

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

[9] L. Kamstra and H. J. A. M. Heijmans, “Reversible data embeddinginto images using wavelet techniques and sorting,” IEEE Trans. ImageProcess., vol. 14, no. 12, pp. 2082–2090, Dec. 2005.

[10] 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.

[11] Z. Ni, Y. Q. Shi, N. Ansari, and W. Su, “Reversible data hiding,” IEEETrans. Circuits Syst. Video Technol., vol. 16, no. 3, pp. 354–362, Mar.2006.

[12] S. K. Lee, Y. H. Suh, and Y. S. Ho, “Reversible image authenticationbased on watermarking,” in Proc. IEEE Int. Conf. Multimedia Expo, Jul.2006, pp. 1321–1324.

[13] W. Hong, T. S. Chen, and C. W. Shiu, “Reversible data hiding for highquality images using modification of prediction errors,” J. Syst. Softw.,vol. 82, no. 11, pp. 1833–1842, Nov. 2009.

[14] D. Coltuc and J. M. Chassery, “Very fast watermarking by reversiblecontrast mapping,” IEEE Signal Process. Lett., vol. 14, no. 4,pp. 255–258, Apr. 2007.

[15] X. Wang, X. Li, B. Yang, and Z. Guo, “Efficient generalized integertransform for reversible watermarking,” IEEE Signal Process. Lett.,vol. 17, no. 6, pp. 567–570, Jun. 2010.

[16] F. Peng, X. Li, and B. Yang, “Adaptive reversible data hiding schemebased on integer transform,” Signal Process., vol. 92, no. 1, pp. 54–62,Jan. 2012.

[17] D. Coltuc, “Low distortion transform for reversible watermarking,” IEEETrans. Image Process., vol. 21, no. 1, pp. 412–417, Jan. 2012.

[18] A. M. Alattar, “Reversible watermark using the difference expansion ofa generalized integer transform,” IEEE Trans. Image Process., vol. 13,no. 8, pp. 1147–1156, Aug. 2004.

[19] Y. Hu, H. K. Lee, and J. Li, “DE-based reversible data hiding withimproved overflow location map,” IEEE Trans. Circuits Syst. VideoTechnol., vol. 19, no. 2, pp. 250–260, Feb. 2009.

[20] X. Li, B. Yang, and T. Zeng, “Efficient reversible watermarking basedon adaptive prediction-error expansion and pixel selection,” IEEE Trans.Image Process., vol. 20, no. 12, pp. 3524–3533, Dec. 2011.

[21] J. Zhou and O. C. Au, “Determining the capacity parameters in PEEbasedreversible image watermarking,” IEEE Signal Process. Lett.,vol. 19, no. 5, pp. 287–290, May 2012.

[22] H.-T. Wu and J. Huang, “Reversible image watermarking on predictionerrors by efficient histogram modification,” Signal Process., vol. 92,no. 12, pp. 3000–3009, Dec. 2012.

[23] G. Xuan, Y. Q. Shi, P. Chai, X. Cui, Z. Ni, and X. Tong, “Optimumhistogram pair based image lossless data embedding,” in Proc. Int.Workshop Digit. Watermarking Ser. Springer Lect. Notes Comput. Sci.,2007, pp. 264–278.

[24] M. Fujiyoshi, S. Sato, H. L. Jin, and H. Kiya, “A location-map freereversible data hiding method using block-based single parameter,” inProc. IEEE Int. Conf. Image Process., vol. 3, Oct. 2007, pp. 257–260.

[25] M. Fallahpour, “Reversible image data hiding based on gradient adjustedprediction,” Inst. Electron. Inf. Commun. Eng. Electron. Exp., vol. 5,no. 20, pp. 870–876, Oct. 2008.

[26] K. S. Kim, M. J. Lee, H. Y. Lee, and H. K. Lee, “Reversible data hidingexploiting spatial correlation between sub-sampled images,” PatternRecognit., vol. 42, no. 11, pp. 3083–3096, Nov. 2009.

[27] P. Tsai, Y. C. Hu, and H. L. Yeh, “Reversible image hiding scheme usingpredictive coding and histogram shifting,” Signal Process., vol. 89, no. 6,pp. 1129–1143, Jun. 2009.

[28] W. L. Tai, C. M. Yeh, and C. C. Chang, “Reversible data hiding basedon histogram modification of pixel differences,” IEEE Trans. CircuitsSyst. Video Technol., vol. 19, no. 6, pp. 906–910, Jun. 2009.

[29] L. Luo, Z. Chen, M. Chen, X. Zeng, and Z. Xiong, “Reversible imagewatermarking using interpolation technique,” IEEE Trans. Inf. Forens.Security, vol. 5, no. 1, pp. 187–193, Mar. 2010.

[30] W. Hong, T. S. Chen, Y. P. Chang, and C. W. Shiu, “A high capacityreversible data hiding scheme using orthogonal projection and predictionerror modification,” Signal Process., vol. 90, no. 11, pp. 2911–2922,Nov. 2010.

[31] X. Gao, L. An, Y. Yuan, D. Tao, and X. Li, “Lossless data embeddingusing generalized statistical quantity histogram,” IEEE Trans. CircuitsSyst. Video Technol., vol. 21, no. 8, pp. 1061–1070, Aug. 2011.

[32] V. Sachnev, H. J. Kim, J. Nam, S. Suresh, and Y. Q. Shi,“Reversible watermarking algorithm using sorting and prediction,” IEEETrans. Circuits Syst. Video Technol., vol. 19, no. 7, pp. 989–999,Jul. 2009.

[33] S. Weng, Y. Zhao, J. S. Pan, and R. Ni, “Reversible watermarking basedon invariability and adjustment on pixel pairs,” IEEE Signal Process.Lett., vol. 15, no. 11, pp. 721–724, Nov. 2008.

[34] M. J. Weinberger, G. Seroussi, and G. Sapiro, “The LOCO-I losslessimage compression algorithm: Principles and standardization into JPEGLS,”IEEE Trans. Image Process., vol. 9, no. 8, pp. 1309–1324,Aug. 2000.

[35] W. Hong, “Adaptive reversible data hiding method based on errorenergy control and histogram shifting,” Opt. Commun., vol. 285, no. 2,pp. 101–108, 2012.

[36] M. Fujiyoshi, T. Tsuneyoshi, and H. Kiya, “A parameter memorizationfreelossless data hiding method with flexible payload size,” Inst.Electron. Inf. Commun. Eng. Electron. Exp., vol. 7, no. 23,pp. 1702–1708, Nov. 2010.