International Journal of Innovative Computing, Information and Control
Volume **9**, Number **2**, February **2013**

ICIC International ©2013 ISSN 1349-4198
pp. 611–623

PREDICTION ERROR ADJUSTMENT TECHNIQUE FOR HIGH QUALITY REVERSIBLE DATA HIDING

Jeanne Chen[1], Wien Hong[2], Chih-Wei Shiu[3], Der-Chyuan Lou[4]
Chien-Chung Lee[4] and Jiang-Lung Liu[4]

[1]Department of Computer Science and Information Engineering
National Taichung University of Science and Technology
No. 129, Sanmin Rd., Sec. 3, Taichung 404, Taiwan
jeanne@nutc.edu.tw

[2]Department of Information Management
Yu Da University
No. 168, Hsueh-fu Rd., Tanwen Village, Chaochiao Township, Miaoli 361, Taiwan
wienhong@ydu.edu.tw

[3]Department of Computer Science and Engineering
National Chung-Hsing University
No. 250, Guo Kuang Rd., Taichung 402, Taiwan
chihwei.shiu@gmail.com

[4]Department of Electrical Engineering
Chung Cheng Institute of Technology
National Defense University
Tahsi, Taoyuan 33509, Taiwan
{dclou; jlliu}@ccit.edu.tw; lcc5921@gmail.com

Received November 2011; revised May 2012

Abstract. *The original modification of prediction error (MPE) method made use of the calculated difference errors and histogram shifting for reversible data hiding which significantly increased image quality and payload in comparison to other similar methods. However, MPE did not fully exploit the merit of prediction since modified pixels were used for predicting errors. In this paper, we proposed a new backward prediction scheme that significantly increased the precision in predictions of the original MPE. Results showed that payload was increased by 15%, significant high PSNR and significant decline in overhead for maintaining the location map.*
Keywords: Overflow, Underflow, Reversible, Prediction, Cover image, Stego-image, Payload

1. **Introduction.** Steganography is a technique for embedding secret messages in digital media [3, 5, 11]. In some applications, such as military, the reversibility of the cover image and the security of the secret message must be maintained. However, maintaining the quality of the stego-image with high payload is a difficult and challenging task.

Reversible data hiding techniques involve manipulating pixel intensities. Coltuc and Chassery [2] proposed the reverse contrast mapping scheme which involved integer transformation of pairs of pixels to pairs of pixels. At most, only one bit of secret could be hidden in one pixel of a pair which limited payload. Tian [10] proposed a simple reversible one pair one bit hiding which required a location map. Alattar [1] and Kamstra and Heijmans [6] extended Tian's scheme to increase hiding capacity. [1] increased hiding for $k-1$ bits to every k-pixels vector. However, the scheme required a larger location map to record non-embeddable locations. [6] integrated extracted bitstreams with the secret data

before replacing the original bitstreams. The scheme had comparable capacity-distortion behavior at low embedding rate and also required a location map. The location maps are larger in [1, 6, 10] since all possible non-embeddable locations have to be recorded. Ni *et al.* [8] proposed a simple histogram shifting scheme with high stego-image quality. Although the scheme did not require a location map, it had small payloads. Thodi and Rodriguez [9] also proposed a scheme which was based on difference expansion of pixel pair and histogram shifting. They were able to achieve a higher payload. [4, 6, 7] proposed improvements to increase payload/image quality that outperformed [1, 10]. However, there are problems with both methods. Although histogram shifting resulted in higher image quality, it still has low payload. The difference expansion method has higher payload but resulted in high image distortion.

Hong *et al.* [4] proposed the modification prediction error (MPE) method which made use of calculated difference errors from prediction errors of neighboring pixels and histogram shifting to hide data. The proposed MPE significantly increased payload and improved the quality of the stego-image in comparison to other related methods. However, the predictions errors were made with modified neighboring pixel values which resulted in inaccuracies. For example, to predict pixel x, the three neighboring pixels a, b and c would be required (see Figure 1). These neighboring pixels would have been modified to hide one bit of secret data in previous steps since scanning to predict the pixels was in raster order from left to right and top to bottom. The inaccuracies could cause the peak of the error histogram to decline which transposed into decreased payload. The maximum payload size cannot be accurately estimated. All 0 and 255 pixels had to be recorded to prevent overflows and underflows – these increased the overheads to the location map. However, MPE outperformed [6, 9] in terms of image quality.

In this paper, we proposed an improvement to MPE by the backward modification prediction error (BMPE) scheme. BMPE predicts in a backward raster scan where the original neighboring pixels are always used in error prediction. The prediction precision will be higher than MPE which predicts with neighboring pixels that could have been altered previously. Overhead is also lower since only the location of pixels with prediction values 0 and 255 will be recorded. The high prediction precision and low overhead in BMPE will result in high payload and high image quality.

2. **Modification Prediction Errors (MPE).** The modification prediction error (MPE) made use of a predictor mechanism similar to JPEG-LS as a feature base selector. As shown in Figure 1, the neighboring pixels a, b and c were used to predict pixel x. Equations (1) and (2) calculate the prediction value P of x and the difference error value E, respectively. Difference errors were used to generate a histogram where the peak is shifted left or right one gray scale unit as part of the reversible hiding process. More details may be located in [4].

$$P = \begin{cases} \min(a,b) & \text{if } c \geq \max(a,b), \\ \max(a,b) & \text{if } c \leq \min(a,b), \\ a+b-c & \text{otherwise,} \end{cases} \tag{1}$$

$$E = x - P \tag{2}$$

Since prediction error E of x was conducted by raster scan order of the image from left to right and top to bottom, the neighbor pixels a, b and c could have been modified to hide one bit of secret data in previous steps. As seen in Figure 2, pixel x with value 156 was added one by Equations (1) and (2). Figure 2(b) showed the modified pixel x' was increased to 157 which was then used to predict the next pixel, 155, on its right; thereby causing some imprecision.

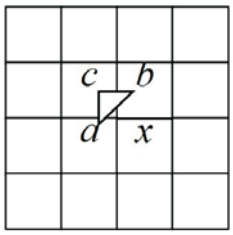


FIGURE 1. JPEG-LS predictor

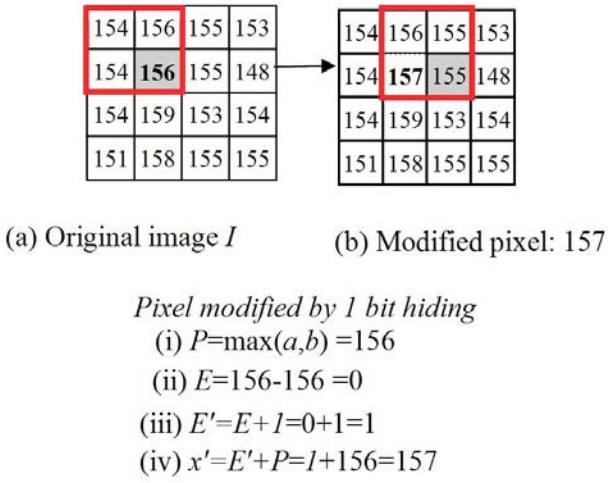


FIGURE 2. Modified pixel used to predict P

In order to avoid the error propagation from modified pixels a, b and c (as seen in Figure 2(b), modified pixel 157 from previous could result in inaccuracy), the original pixel values should be used in predicting P. Therefore, if the original pixel 156 (see window in Figure 2(a)) was used in Figure 2(b) for predicting pixel 155 in the next window, the prediction should be accurate. That is in Figure 2(b) if prediction was made with original pixels 156 (modified value was 157), 155 and 156, the prediction value would be 155 and the prediction error is 0. In our proposed BMPE, the backward modification is used to ensure that prediction is made with the original neighboring pixels: a, b and c.

3. **Backward Modification Prediction Error (BMPE).** In the proposed backward modification prediction error (BMPE) scheme, prediction was made with the original a, b and c pixels. The predictor mechanism used in MPE was from left to right and top to bottom order (see Figure 3(a)). On the contrary, the proposed BMPE predictions are by backward order from right to left and bottom to top, as illustrated in Figure 3(b).

The backward mechanism made sure that original neighboring pixels were used in the predictions. The prediction error is accurate by using the original pixels; therefore, the peak is higher. Furthermore, the maximum payload can be estimated before embedding. A simple trial test was performed on Lena to calculate the predictions and error values for both MPE and BMPE. From the test, BMPE always has a higher peak. A higher peak meant that more prediction errors were concentrated together; the prediction is precise and the payload may be increased.

FIGURE 3. Scan orders for MPE vs. BMPE

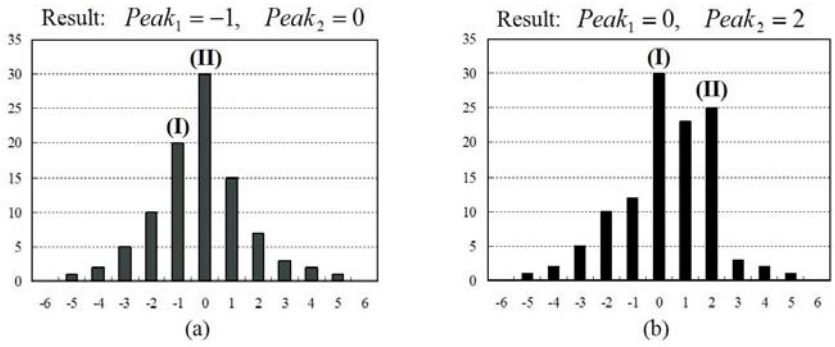


FIGURE 4. Prediction error histogram for with peaks at $Peak_1$ and $Peak_2$

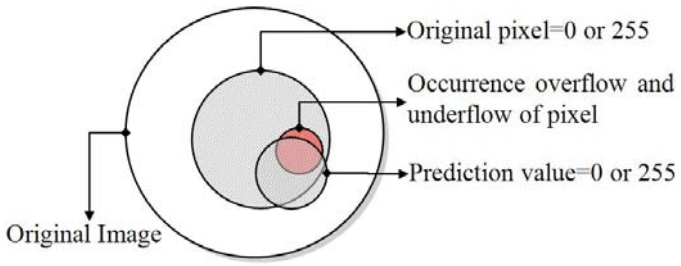


FIGURE 5. Overflow and underflow relationship between MPE and BMPE

As shown in the example in Figure 4, in the proposed scheme embedding would be in the highest two peaks $Peak_1$ and $Peak_2$. The two peaks are considered as suitable for high payload hiding. The embedding rule required that $Peak_1 < Peak_2$.

Two additional rules were included: (1) the first row and column are not to be modified similar to the JPEG-LS predictor; that is, the row and column are needed to obtain the original prediction parameters; (2) to avoid the overflow (> 255) and underflow (< 0). When prediction is 0 or 255, no embedding is done. However, if overflow or underflow occurred after embedding then the location of the stego-pixel would be recorded in the location map.

In MPE, record must be kept in the location map for all pixels with values 0 or 255. As shown in Figure 5, the smallest circle is the amount of overflow and underflow that actually occurs. When prediction values were either 0 or 255, a substantial amount actually occurred within the smallest circle while a significant amount lies on the pixels that did not have the same prediction range. In BMPE, only pixels with prediction values 0 or 255 will be recorded in the location map. A small amount of underflow and underflow

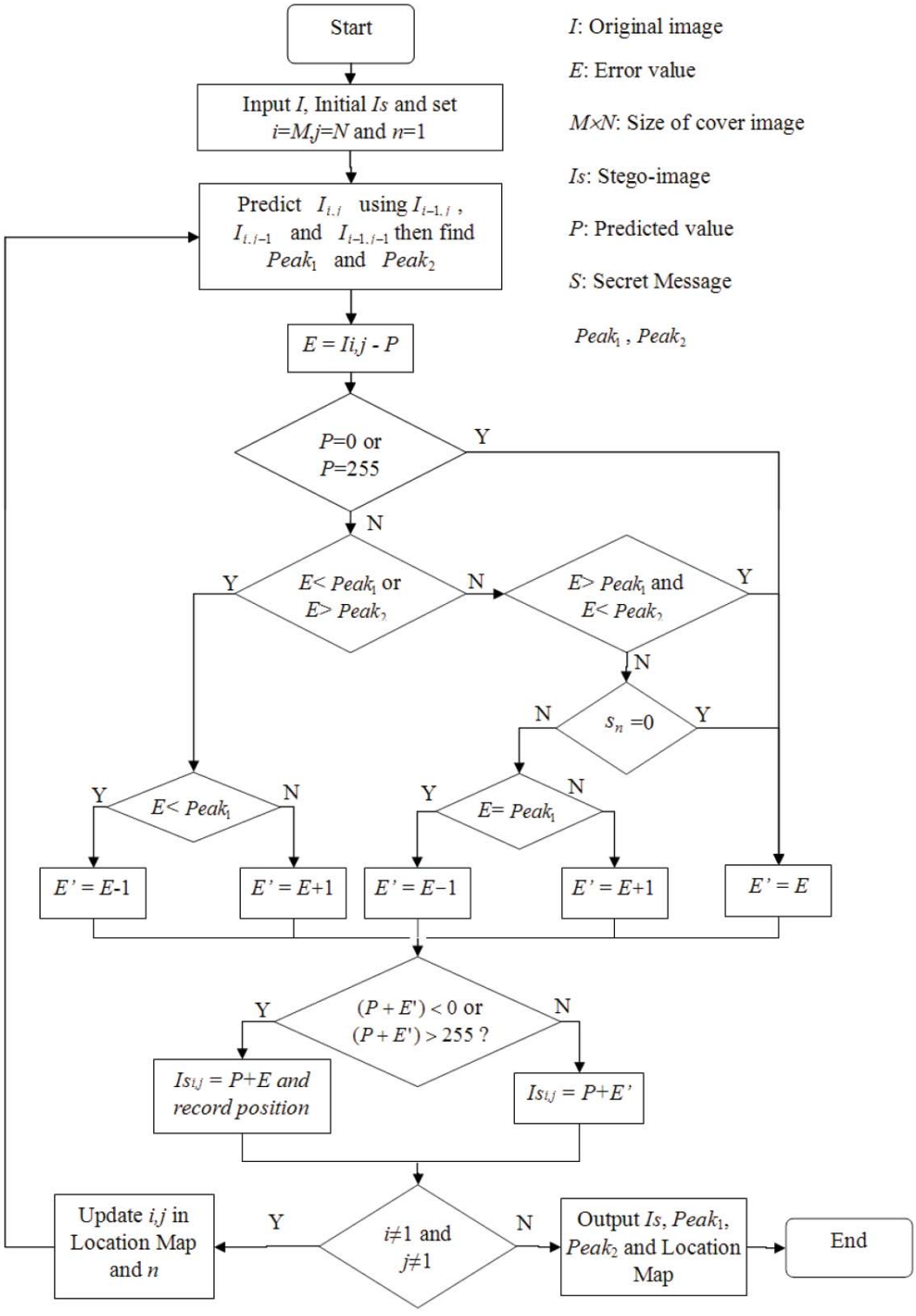


FIGURE 6. Flowchart for embedding

were not recorded but from experience the amount was insignificant.

(1) Flow for embedding

Let the cover image be I and the stego-image be Is. The first row and column of Is remained unmodified and are the same as those in I. The secret S will be encrypted using RSA or DES. Suppose $S = \{s_i | s_i \in \{0, 1\}\}$ where $i = 1, 2, \ldots, z$ and z is the length of

the secret. The cover image I is then transformed to prediction error domain. From the prediction error histogram, the two peaks would be selected for hiding one bit of secret data. Detailed description of the algorithm for the embedding process is given below and the flowchart is illustrated in Figure 6.

Input: Cover Image I, Secret message S.
Output: Stego Image Is, location map, $Peak_1$ and $Peak_2$.

Step 1: Initialized $i = M$ and $j = N$, where i and j are pixel coordinates of I with M width and N height; and, $n = 0$, n is a position in the secret message. From the cover image I find the two highest peaks. We assume $Peak_1 < Peak_2$.
Step 2: Use $I_{i-1,j}, I_{i,j-1}$ and $I_{i-1,j-1}$ to find prediction value P for $I_{i,j}$ using Equation (1). Then calculate the prediction error E as in Equation (2) where the pixel value of $I_{i,j}$ is x.
Step 3: Underflow or overflow condition: If $P = 0$ or 255 then $E' = E$ (without embedding) and goto Step 9.
Step 4: Error values on either side of peaks: If $E < Peak_1$ or $E > Peak_2$ then begin histogram shifting; otherwise goto Step 6.
Step 5: Histogram shifting: If $E < Peak_1$ then $E' = E - 1$; otherwise $E' = E + 1$ (with embedding). Goto Step 9. $Peak_1 < Peak_2$.
Step 6: Error value is within peaks: If $E > Peak_1$ and $E < Peak_2$ then $E' = E$, goto Step 9.
Step 7: Embedding '0' bit: If $S_n = 0$ then $E' = E$ and goto Step 9; otherwise goto Step 8.
Step 8: If $E = Peak_1$ and secret bit is '1' then $E' = E - 1$; otherwise $E' = E + 1$.
Step 9: In the case of overflow, 0 or underflow, 255. If $(P + E') < 0$ or $(P + E') > 255$ then $Is_{i,j} = P + E$ and record the position (i, j) in the location map; otherwise $Is_{i,j} = P + E'$.
Step 10: If $i \neq 1$ and $j \neq 1$ then update i, j (in Location Map) and n; repeat Steps 2-10; otherwise output Is, $Peak_1$ and $Peak_2$ and location map.

(2) Flow for extraction and recovery

The flow for the extraction and recovery procedure is as shown in Figure 7. The extraction process is similar to embedding but in the reverse order. The embedded data is extracted in the order as illustrated in Figure 3(a). This is to ensure that the neighboring values are restored before the prediction step. The extraction-recovery algorithm is described below.

Input: Stego-image Is, Location Map, $Peak_1$ and $Peak_2$.
Output: Restore Image Ir, secret message Sr.

Step 1: Initialize $i = 1$, $j = 1$; i and j are pixel coordinates.
Step 2: $Is_{i-1,j}$, $Is_{i,j-1}$ and $Is_{i-1,j}$ are used to predict P; calculate difference error E and $E = Is_{i,j} - P$.
Step 3: If $P = 0$ or 255 then $E' = E$ then goto Step 11.
Step 4: If i, j is in location map, $E' = E$ then goto Step 11.
Step 5: If $E = Peak_1$ or $E = Peak_2$ then set $S = '0'||S$, $E' = E$.
Step 6: If $E = Peak_1 - 1$ then $S = '1'||S$, $E' = E + 1$.
Step 7: If $E = Peak_2 + 1$ then $S = '1'||S$, $E' = E - 1$.
Step 8: If $E < Peak_1 - 1$ then $E' = E + 1$.
Step 9: If $E > Peak_2 + 1$ then $E' = E - 1$.
Step 10: If $E > Peak_1$ or $E < Peak_2$ then $E' = E$.
Step 11: $Ir_{i,j} = P + E'$ and if $i \neq M$ and $j \neq N$ then Update i, j and repeat Steps 2-11; otherwise output Ir and Sr.

4. **Experimental Results and Analysis.** Figure 8 shows six 8-bits 512×512 grayscale images used in the experimental tests. The images were chosen from the University of Southern California Signal and Image Processing Institute Image Database [12]. Images Lena, F-16, Boat, Peppers, Sailboat and Baboon were chosen for their varying complexness. In the selection, Lena is considered the simplest image with smooth background while Baboon is the most complex image. As seen in Figure 9, the prediction histogram peak in Lena is significantly higher than that in Baboon. In the proposed BMPE scheme, the smooth areas (areas with lower variance) are the most desirable locations for embedding since more data can be embedded while providing fewer modifications. The binary secret message is also randomly scrambled for increasing security before being embedded.

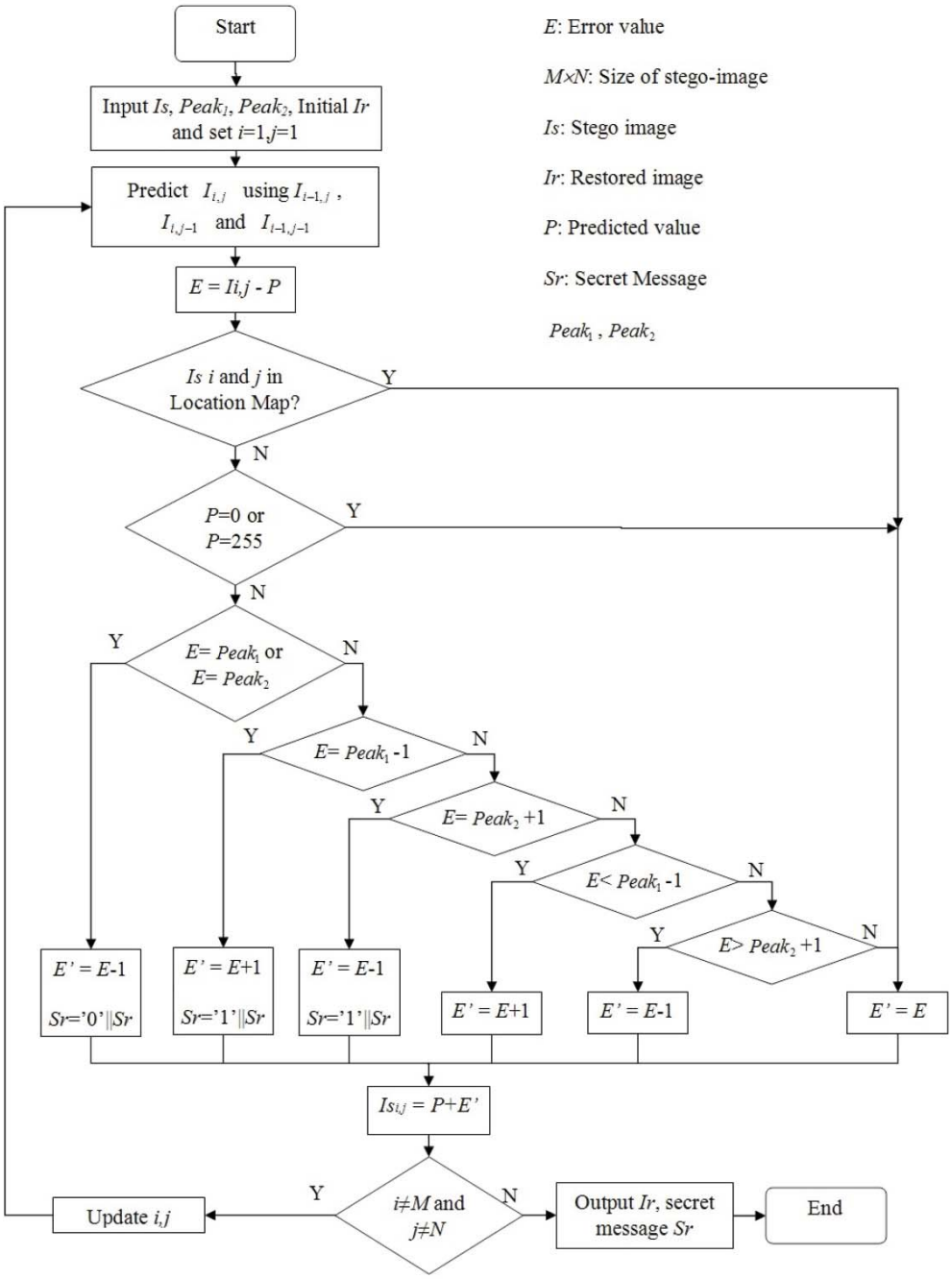


FIGURE 7. Flowchart for extraction and recovery

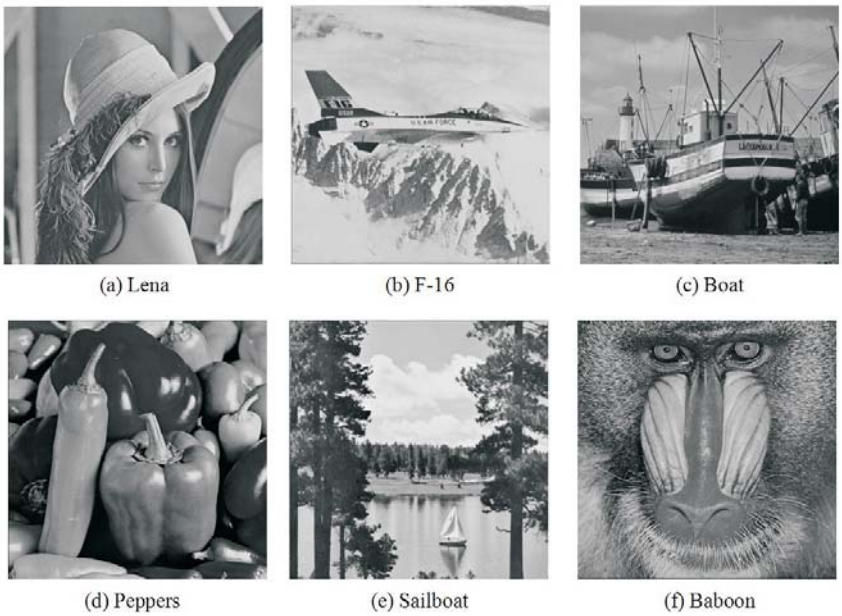


FIGURE 8. Cover images used for embedding

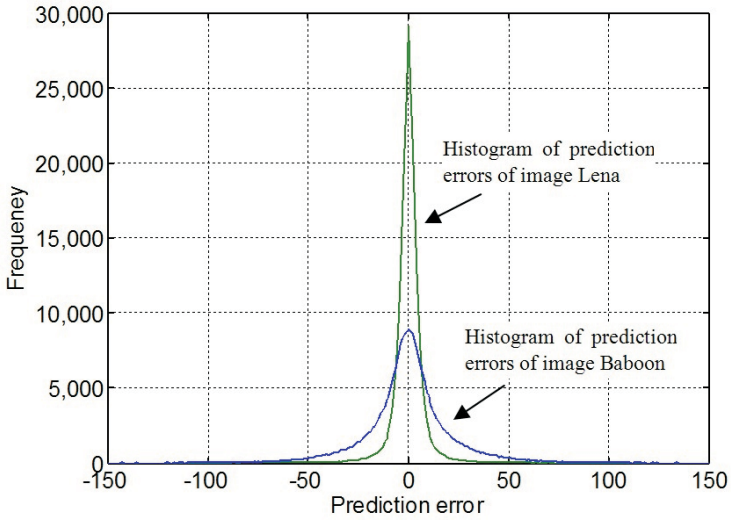


FIGURE 9. Histograms of prediction errors for images Lena and Baboon

Also, only predicted pixels causing overflows and underflows were recorded in the location map; that is $\lceil \log_2 h \rceil + \lceil \log_2 w \rceil$ bits are to be recorded where h and w are the respective height and width of the image. This is a decrease in overhead.

(1) Difference error image

Ni *et al.*'s [8] histogram shifting is adopted in the proposed method during embedding. Therefore, each error value is modified one bit. The bitmap difference between the stego and cover images is as shown in Figure 10; black is 0 bit (no difference) while white is 1 bit (difference $+/-1$).

As seen in the figure, most of the white bits are concentrated on the edges. The reason for this was that it is difficult to predict the pixel values of edges. In histogram shifting, most of these values will not lie in the peak areas. Therefore, shifting (embedding secret bits) occurred mainly in peaks of non-edges. In the proposed BMPE, embedment occurred

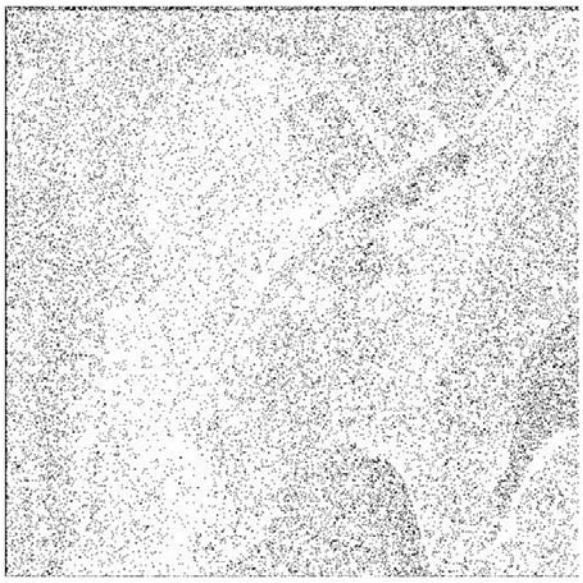


FIGURE 10. Bit difference between the stego and cover image

TABLE 1. Relationship between the payloads and peaks of BMPE and MPE

Images (512×512)	Np	MPE		BMPE	
		Payload (bits)	Percent	Payload (bits)	Percent
Lena	53,834	46,860	87%	53,834	100%
F-16	81,556	66,095	81%	81,556	100%
Boat	33,125	29,299	88%	33,125	100%
Peppers	33,002	27,965	85%	33,002	100%
Sailboat	29,996	26,396	88%	29,996	100%
Baboon	17,695	17,044	96%	17,695	100%
Average	41,535	29,422	84%	41,535	100%

only on the two highest peaks: $Peak_1$ and $Peak_2$. The error values in the peaks were mostly from the smooth areas; these areas were considered to have accurate predictions. Hiding one bit in these areas will shift the histogram one location. On spatial domain, the embedment is relatively imperceptible in comparison to those on the edges.

(2) Payload comparison between MPE vs. BMPE methods

Table 1 shows the payload comparison between MPE and BMPE for embedding in the peaks of the error values. Equation (3) calculates the payload percentile *percent*, where Np is the number of error values for the peaks: $Peak_1$ and $Peak_2$.

$$Percent = \frac{Payload}{Np} \tag{3}$$

The prediction errors of MPE and BMPE can be statistically calculated for $Peak_1$ and $Peak_2$. The calculated total, Np, is the maximum possible payload of MPE and BMPE. The difference between Np and the actual payload can be calculated so as to compare the difference from the payload of MPE. As seen in Table 1, BMPE has the same amount of hiding the size of Np. Comparatively, MPE had a lower payload. The reason was that the original neighboring pixels a, b and c were used in BMPE for predicting while MPE predicts from possible modified neighboring pixels.

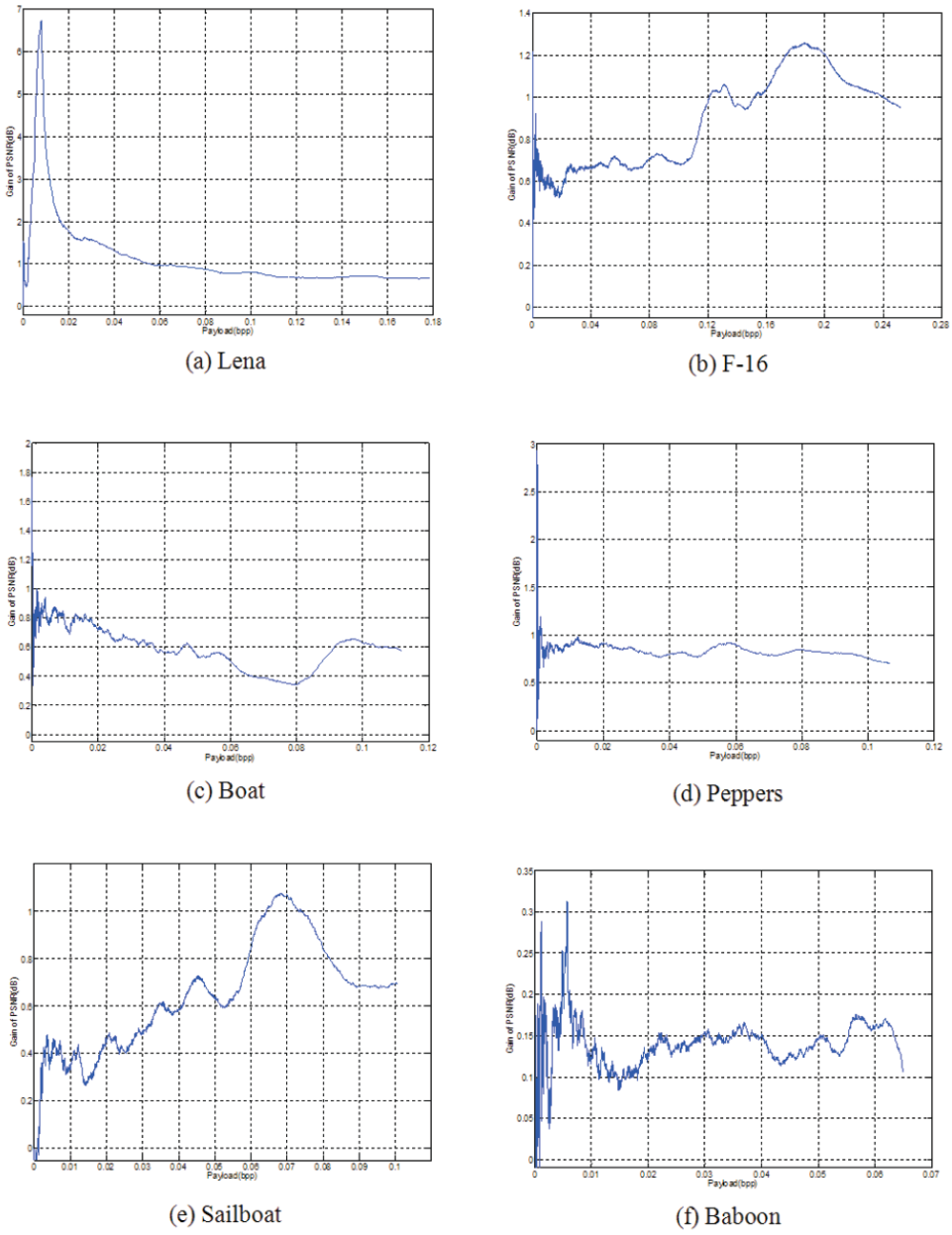


FIGURE 11. Comparing payload differences between BMPE and MPE

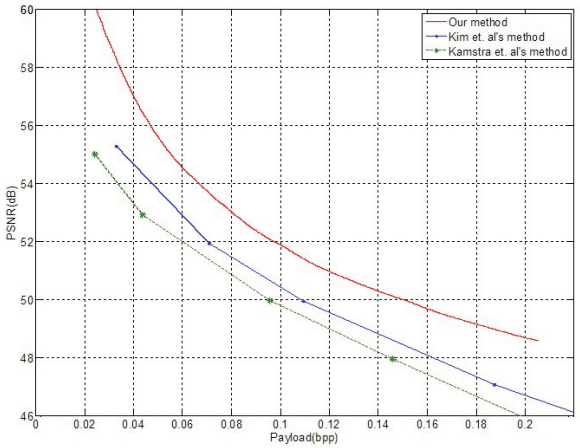
Figure 11 shows BMPE to have higher payloads and PSNRs than MPE. Compared with the other images, Lena has the highest PSNR difference at 6.75 dB and a high payload at 0.08 bpp. Comparatively, Baboon is the most complex image and had a smaller increase in PSNR of 0.11 dB with a payload of 0.07 bpp. Images characterized with more smooth areas such as Lena and F-16 can hide more data and maintain relative quality in the stego-image. As seen in Figure 11(f), Baboon had lower payloads and at under 0.0005 bpp had lower performance than MPE. The reason was that Baboon is a comparably complex image. However, the PSNR for BMPE was relatively higher than MPE.

(3) Payload vs quality (PSNR)

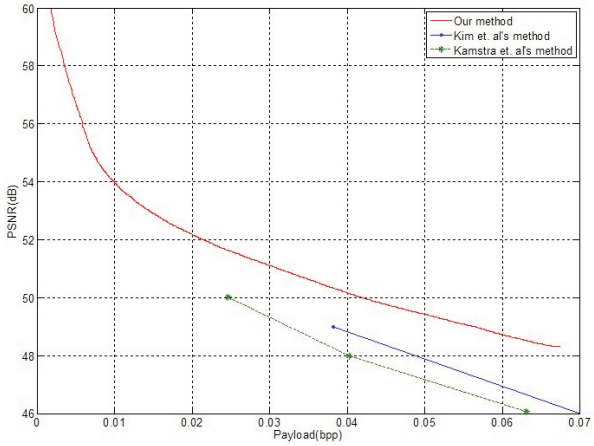
In another comparison on payload difference between BMPE and two other methods, namely, Kim *et al.* [7] and Kamstra *et al.* [6], BMPE showed significantly higher PSNR and payload than both [6] and [7] (see Figure 12). The payloads used in the comparisons were minus the size of location map. Both Lena and Baboon maintained more than 48.13

TABLE 2. Size of location map after feature selection

Images	Location map size (Kbits) Before	After	Improvement (Kbits)
a	874.08	95.61	778.47
b	2.61	1.33	1.28
c	501.94	57.17	444.77
d	2.60	1.33	1.27
e	65.14	10.98	54.16
f	58.93	47.77	11.16
Avg.	250.88	35.70	215.19



(a) Lena



(b) Baboon

FIGURE 12. Comparing payload differences for (a) Lena, (b) Baboon

dB for the highest payload.

(4) Underflow and overflow omission

In the proposed scheme, prediction values 0 and 255 were not used for embedding since the action might cause either an underflow or overflow. High payload is not affected since the peak rarely falls on either of these two locations. For experimental tests on the location map, four different images, Figures 13(a) and 13(b) were chosen for their brightness and two others, Figures 13(e) and 13(f), for darker gradient. Figures 13(c)

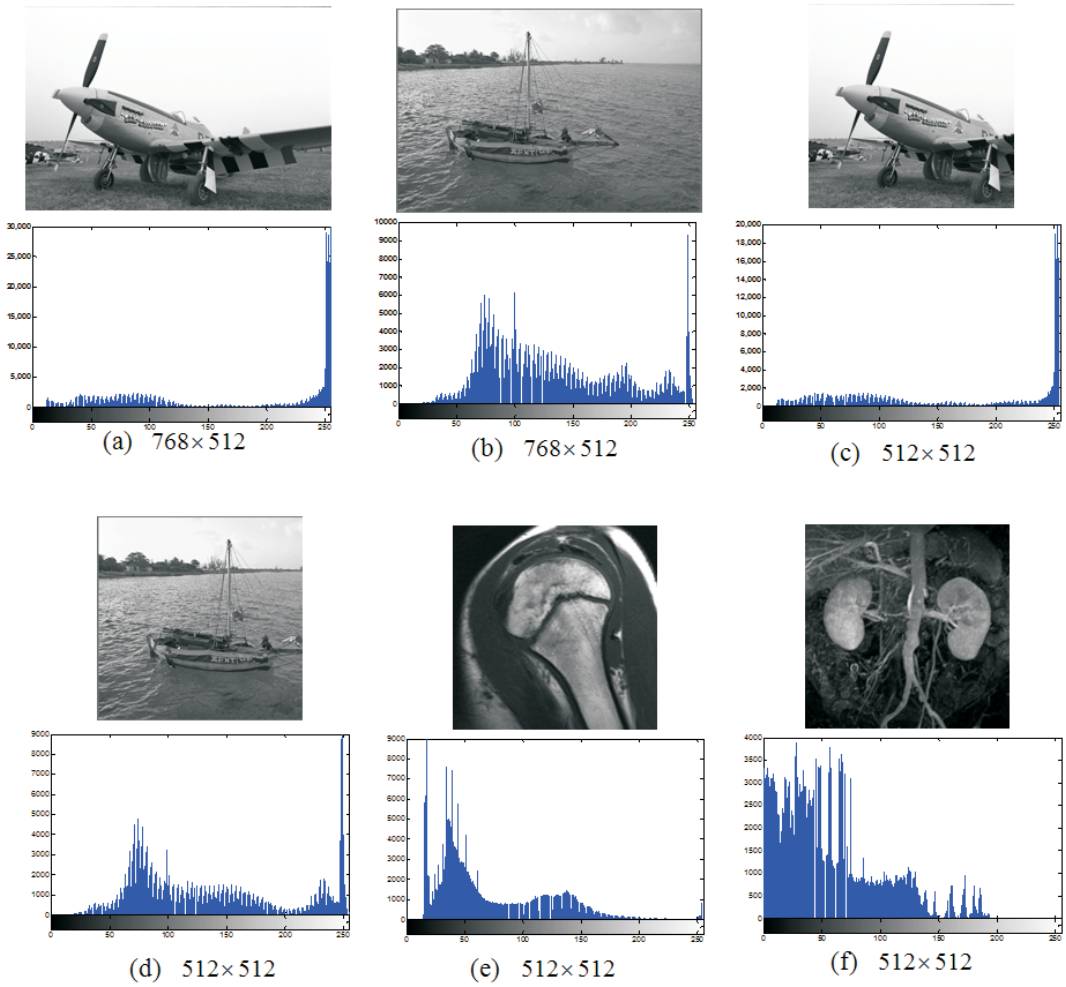


FIGURE 13. Image gradient and corresponding histograms

and 13(d) were sized down versions of Figures 13(a) and 13(b); that is from 768×512 to 512×512, respectively, so that a fair comparison may be made with Figures 13(e) and 13(f). The histograms of the images showed that the peaks for the two brighter images tend to locate towards the right side and for the two darker images, towards the left. Most images with well-distributed gradients have peaks on both sides which would require extra overheads to keep records on the locations.

As seen in Table 2, after the overflow and underflow were omitted as embedding candidates the location map size was reduced by an average of 215.19 Kbits. Also, Figures 13(a) and 13(c) had the largest reduction since a large portion of the background is the smooth bright colored sky.

5. **Conclusions.** The proposed BMPE scheme ensured accuracy for the error values by using only the original neighboring pixels by backward propagation. Results showed that significantly higher payload was achieved while quality was maintained with PSNRs of more than 48 dB. By omitting underflow and overflow from the embedding step for prediction values 0 and 255, overhead for the location map was significantly reduced. In BMPE, embedment was concentrated on the smooth areas. Edges were avoided which had the effect of maintaining the spatial qualities of the stego-images.

REFERENCES

[1] A. M. Alattar, Reversible watermark using the difference expansion of a generalized integer transform, *IEEE Transactions on Image Processing*, vol.13, no.8, pp.1147-1156, 2004.

[2] D. Coltuc and J. M. Chassery, Very fast watermarking by reversible contrast mapping, *IEEE Signal Processing Letters*, vol.14, no.4, pp.255-258, 2007.

[3] C.-C. Chang, Y.-H. Chen and T. D. Kieu, A watermarking technique using synonym substitution for integrity protection of XML documents, *ICIC Express Letters*, vol.4, no.1, pp.89-94, 2010.

[4] W. Hong, T. S. Chen and C. W. Shiu, Reversible data hiding based on histogram shifting of prediction errors, *Journal of Software and Software*, vol.82, pp.1833-1842, 2009.

[5] W. Hong and T. S. Chen, A novel data embedding method using adptive pixel pair matching, *IEEE Transactions on Information Forensics and Security*, vol.7, no.1, pp.176-184, 2012.

[6] L. Kamstra and H. J. A. M. Heijmans, Reversible data embedding into images using wavelet techniques and sorting, *IEEE Transactions on Image Processing*, vol.14, no.12, pp.2082-2090, 2005.

[7] H. J. Kim, V. Sachnev, Y. Q. Shi, J. Nam and H. G. Choo, A novel difference expansion transform for reversible data embedding, *IEEE Transactions on Information Forensics and Security*, vol.3, no.3, pp.456-465, 2008.

[8] Z. Ni, Y. Q. Shi, N. Ansari and W. Su, Reversible data hiding, *IEEE Transactions on Circuits and Systems for Video Technology*, vol.16, no.3, pp.354-362, 2006.

[9] D. M. Thodi and J. J. Rodriguez, Expansion embedding techniques for reversible watermarking, *IEEE Transactions on Image Processing*, vol.16, no.3, pp.721-730, 2007.

[10] J. Tian, Reversible data embedding using a difference expansion, *IEEE Transactions on Circuits and Systems for Video Technology*, vol.13, no.8, pp.890-896, 2003.

[11] H. Wang and S. Wang, Cyber warfare-steganography vs. steganalysis, *Communications of the ACM*, vol.47, no.10, pp.76-82, 2004.

[12] *The USC-SIPI Image Database*, http://sipi.usc.edu/database/, 2011.