Expand Variance Mean Sorting for Reversible Watermarking

Aekawit Kotvicha, Parinya Sanguansat, and M. L. Kulthon Kasemsa

***Abstract*—Prediction Error Expansion is one of Reversible Watermarking (RW) techniques that gives effective results and has been continuously developed over the years. Among those techniques, Sorting Absolute Prediction Error (APE) is an important step used to reduce image distortion; so in this research we focus on developing the technique of sorting APE in ascending order more accurately. We can use local variance values in predicting APE and these values are calculated from means so they are suitable for predicting which areas contains high or low APE values. Consequently, we have improved this parameter by using Expanded Variance Means which can increase efficiency of sorting process and raise PSNR values.**

***Index Terms*—Reversible watermarking, prediction error expansion, variance sorting, lossless data hiding**

I. Introduction

Image Reversible Watermarking is a method to embed data into digital images which can recover both original images and embedded data completely. This field of research has been studied and developed for many years so there are several advanced methods to use in RW. [1],[2],[3],[4] used modulo-arithmetic based additive spread spectrum methods. De Vleeschouwer et al. [5] used the circular interpretation of the bijective transformations of the image histogram. Fridrich et al. [6] losslessly compressed high level bit-plans. Xuan et al. [7] used a histogram modification operation to avoid overflow or underflow problem.

However, the most popular and effective technique, firstly presented by Tian [8], is Different Error Expansion based on bit shifting method. Thodi and Rodriquez's [9],[10] research showed their idea of prediction error to embed data instead of different error and could reduce the values of image distortion. They also utilized a histogram shift method to decrease location map size which helped increasing space to embed more data. Kamstra and Heijimans [11] proposed a method to reduce location map size by sorting pairs according to correlation measures to facilitate compression. Sachnev et al. [12] applied algorithm method using sorting and prediction which greatly reduced location map size. They also divided image pixels into 2 groups "Cross Set (or Cross)" and "Dot Set (or Dot)," and used Cross to predict Dot data and vice versa -- which made the predictor worked more efficiently. This scheme also helped to embed data into low APE regions more precisely. As a result, Sachnev's work gave better output in increasing PSNR values compared with other previous works.

The rest of this paper will discuss in details as follows: section 2 prediction error expansion; section 3 histogram shifting; section 4 expanded variance means sorting; section 5 data embedding and extracting; section 6 experimental results and section 7 conclusion.

Manuscript received May 3, 2012; revised June 4, 2012.

Aekawit Kotvicha and M. L. Kulthon Kasemsa are with Faculty of Information Technology, Rangsit University, Thailand (e-mail: aek_kot@hotmail.com)

Parinya Sanguansat is with Faculty of Engineering and Technology, Panyapiwat Institute of Management, Thailand

II. Prediction Error Expansion

A method of Prediction Error Expansion is to add space into the last bit of data in order to embed more bits into the data. The main idea is to shift bit of data which can obtain by expanding prediction error value of each pixel.

Start by separating image's pixels into 2 sets, Cross and Dot, as seen in Fig. 1.

x	$v_{i-1,j}$	x	o
$v_{i,j-1}$	$u_{i,j}$	$v_{i,j+1}$	x
x	$v_{i+1,j}$	x	o
o	x	o	x

Fig. 1. Cross set and dot set in an image

We predict data in Cross by using Dot in equation (1)

$$u'_{i,j} = round\left(\frac{v_{i,j-1} + v_{i,j+1} + v_{i-1,j} + v_{i+1,j}}{4}\right) \tag{1}$$

Prediction Error (PE) can be calculated by (2).

$$d_{i,j} = u_{i,j} - u'_{i,j} \tag{2}$$

We expand PE to add more space in the last bit so we can embed more information bit $b \in \{0,1\}$. Expanded PE process will be calculated by

$$D_{i,j} = 2d_{i,j} + b \tag{3}$$

Modified image data will be

$$U_{i,j} = u'_{i,j} + D_{i,j} \tag{4}$$

We can decode embedded information and recover original image values using equation (5) and (6).

$$b = U_{i,j} \bmod 2 \tag{5}$$

And

$$u_{i,j} = u'_{i,j} + d_{i,j} \tag{6}$$

where $d_{i,j} = \lfloor D_{i,j} / 2 \rfloor$

III. Histogram Shifting

Histogram Shifting technique is purposed for decreasing image distortion. Start by defining thresholds, Tp and Tn. We use these thresholds for selecting pixels to embed data. PE values, between Tp and Tn, will be expanded while the rest will be shifted as equation (7).

$$D_{i,j} = \begin{cases} 2d_{i,j} + b & \text{if } d_{i,j} \in [T_n, T_p] \\ d_{i,j} + T_p + 1 & \text{if } d_{i,j} > T_p \text{ and } T_p \ge 0 \\ d_{i,j} + T_n & \text{if } d_{i,j} < T_n \text{ and } T_n < 0 \end{cases} \tag{7}$$

We can recover original PE by converging the expanding and shifting formula as seen in equation (8).

$$d_{i,j} = \begin{cases} \lfloor D_{i,j} / 2 \rfloor & \text{if } D_{i,j} \in [2T_n, 2T_p + 1] \\ D_{i,j} - T_p - 1 & \text{if } D_{i,j} > 2T_p + 1 \text{ and } T_p \ge 0 \\ D_{i,j} - T_n & \text{if } D_{i,j} < 2T_n \text{ and } T_n < 0 \end{cases} \tag{8}$$

The threshold values depend on payload sizes. To define an appropriate combination of threshold values, we will select the one which achieves the best PSNR using iterative adjusting [12].

IV. Expand Variance Mean Sorting

One interesting technique used to decrease image distortion is sorting, according to [11], [12], [13] if selecting pixels with low APE values to use in histogram shifting, the output of image distortion will be better than using original sequence pixels. In order to sort pixels, Sachnev et al. [12] used local variance which gave a satisfied result. Local variance can be calculated as in equation (9).

$$\mu_{i,j} = \frac{1}{4}\sum_{k=1}^{4}\left(\Delta v_k - \Delta \bar{v}_k\right)^2 \tag{9}$$

where $\Delta v_1 = |v_{i,j-1} - v_{i-1,j}|$, $\Delta v_2 = |v_{i-1,j} - v_{i,j+1}|$, $\Delta v_3 = |v_{i,j+1} - v_{i+1,j}|$, $\Delta v_4 = |v_{i+1,j} - v_{i,j-1}|$

And $\Delta \bar{v}_k = (\Delta v_1 + \Delta v_2 + \Delta v_3 + \Delta v_4)/4$

We have developed sorting process by averaging local variance of the neighborhood pixels in Cross, this average value is Expanded Variance Mean (EVM) and we use it as parameter in sorting. The parameter will help sorting low variance pixels located in similar variance region and use it to embed first, thus, PSNR is higher. We also select number of pixels to calculate EVM which optimize PSNR. EVM can be calculated by equation (10) and (11).

When ex is even

$$\bar{\mu}^{ex}_{i,j} = \frac{\sum_{l=-\frac{ex}{2}}^{\frac{ex}{2}} \mu_{i+l,j+l} + \sum_{k=1}^{\frac{ex}{2}} \sum_{l=-\frac{ex}{2}}^{\frac{ex}{2}-2k} \left(\mu_{i+l,j+l+2k} + \mu_{i+l+2k,j+l}\right)}{\frac{ex^2}{2} + ex + 1} \tag{10}$$

When ex is odd

$$\bar{\mu}^{ex}_{i,j} = \frac{\sum_{l=-\frac{ex-1}{2}}^{\frac{ex-1}{2}} \mu_{i+l,j+l} + \sum_{k=1}^{\frac{ex-1}{2}} \sum_{l=-\frac{ex+1}{2}}^{\frac{ex+1}{2}-2k} \left(\mu_{i+l,j+l+2k} + \mu_{i+l+2k,j+l}\right)}{\frac{ex^2}{2} + 2ex - \frac{3}{2}} \tag{11}$$

$\mu_{i,j}$ is the variance of the pixel (i,j). The number ex is called a level of variance expansion (VE) which is an index representing the numbers of pixels, n, to be used for calculating EVM which is equal to

$$n = \begin{cases} \frac{ex^2}{2} + ex + 1 & \text{when } ex \text{ is } even \\ \frac{ex^2}{2} + 2ex - \frac{3}{2} & \text{when } ex \text{ is } odd \end{cases} \tag{12}$$

For example: ex=2 means using 5 pixels, ex=3 means using 9 pixels.

We use the EVM, $\bar{\mu}^{EX}$, for sorting such that

$$PSNR\left(T_p, T_n, \bar{\mu}^{EX}\right) = \max_{ex} PSNR\left(T_p, T_n, \bar{\mu}^{ex}\right) \tag{13}$$

The numbers in figure 2 represent the levels of VE used in the algorithm. We can see that different images and different payloads need different levels of VE in order to obtain the highest PSNR. The maximum possible number of ex depends on a size of images. We spend 6 bits in the header to put the level of VE used in the algorithm which we send to the decoder. Hence, in our experiments, we can calculate EVM up to 64 levels which is enough for our test images as can be seen in figure 2. More generally, the maximum level of VE depends on how many bits we define in the header. If we define m bits in the header, the maximum level of VE will be 2^m.

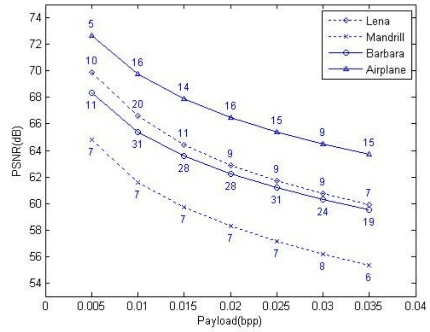


Fig. 2. Payload and PSNR graphs of purposed method applied to four test images with different VE levels.

V. DATA EMBEDING AND EXTRACTING ALGORITHM

One major problem in Reversible Watermarking is overflow and underflow. The problem is caused by modified data higher than 255 or lower than 0 as shown in [12]. Not only Sachnev's method can solve this problem, but also decreases location map size by using double adjustment test [12]. Our proposed algorithms, which are modified from Sachnev's works and based on EVM, are shown as follows;

Embedding Algorithm:

1) Separate image pixels in to Cross and Dot.
2) Preserve the first 40 bits of Cross.
3) Sort the rest of Cross ascending, start from the 41st pixel using EVM.
4) Determine thresholds, Tp and Tn, according to [12].
5) Collect LSB of the first 40 bits and add them into payload
6) Embed payload using histogram shifted method, according to [12].
7) Calculate PSNR.
8) Repeat step 1-7 but increase number of pixels used in calculating EVM (step 3).
9) Select number of pixels in step 8 with optimized PSNR to use in data embedding.

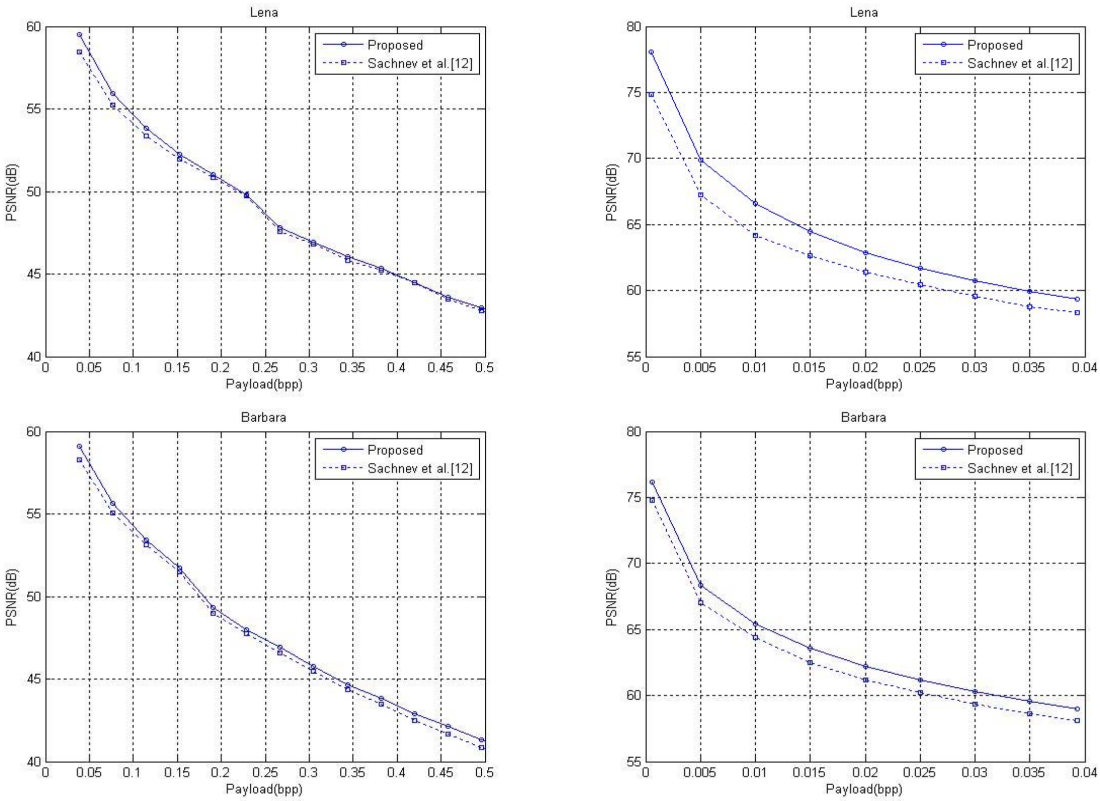
In the 8th step, we repeat embedding algorithm using different numbers of pixels for calculating EVM (step3) in order to obtain the best PSNR value. Then, we put the index *EX*, a level of VE, corresponding to the number of pixels, mention above, into the header. The header involves thresholds, Tp (7 bits), Tn (7 bits), payload size (20 bits) and *EX*, an index of variance expansion (6 bits) which costs 40 bits in Cross. This information is needed for decoding and image recovering.

To extract hiding data and recover the original image, an extracting algorithm is as follows,

Extracting Algorithm:

1) Separate image pixels in to Cross and Dot.
2) Extract header from the first 40 bits of Cross, thus we obtain threshold values, and number of pixels to calculate EVM in the next step.
3) Sort Cross set data, start from the 41st pixel using EVM.
4) Apply extracting algorithm sequentially from step 3, according to [12].
5) Recover LSB values for the first 40 bits of Cross.

For the whole image, after we apply the embedding algorithm to Cross, we will use the same algorithm to Dot. We separate a payload into 2 parts, half will be hided in Cross and the other will be hided in Dot. This will help the predictor performs better than using only one set even we have a small load. On the other hand, when extracting and recovering image data, we will apply an extracting algorithm to Dot to recover original Dot data before decode data and recover Cross.



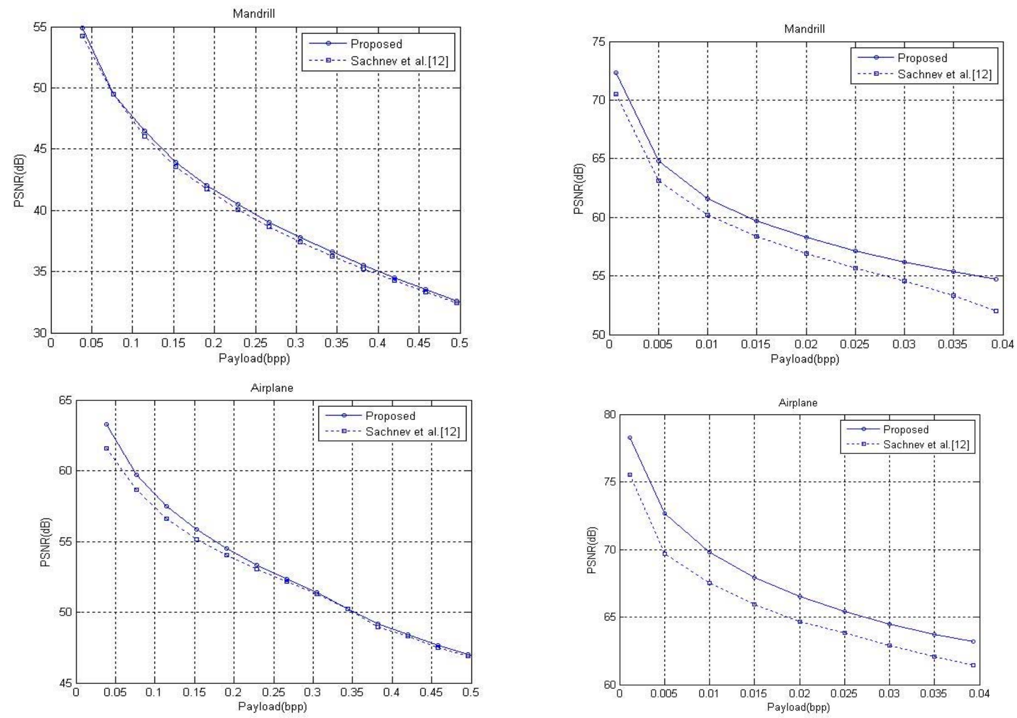


Fig. 3. Payload and PSNR graphs of purposed and sachnev et al. method of four tested images.
left column shows results for 0-0.5 bpp payload. right column of graphs shows results for small payload.

VI. Experimental Results

We have run several tests on our algorithm using varied 512x512 grayscale standard test images with both low and high variation values e.g. Lena, Barbara, Mandrill, Airplane etc. Figure 3 presents the result of the four images using our scheme comparing to Sachnev's, the graphs show relationship between payload size (bpp) and Peak Signal to Noise Ratio value (PSNR) which indicates image distortion. The left column presents results of the methods for payload up to 0.5 bpp; the right column presents results when the methods applied to a small load.

The results show our scheme's output get higher PSNR values. Suitable number of pixels used for calculating EVM well be varied depending on image's variation and payload size. This developed tool helps sorting APE more accurately and gives better performance in decreasing image distortion, especially in small payloads. However, in low variation images e.g. Lena or Airplane, the gap between 2 graphs will be narrower when payload sizes become bigger as more pixels will be added in data embedding.

VII. Conclusion

This paper shows that Expand Variance Sorting can improve sorting efficiency and use EVM in embedding and extracting can give better sequence of pixels when compare to Sachnev's method. Thus, location map size gets smaller and using lower PE values for histogram shifting. These lead to the result that an image distortion is decreasing. The sorting scheme also raises PSNR values, especially when payloads are small. However, in low variation images e.g. Lena or Airplane, the capacity of this procedure will decrease when payloads get bigger.

References

[1] J. W. Bender, D. Gruhl, N. Morimoto, and A. Lu, "Technique for data hiding," *IBM Syst.Journal*, vol. 35, no. 3, pp. 3131-336, 1996.

[2] M. Barton, "Method and apparatus for embedding authentication information within digital data," U.S. Patent, pp. 646 997, 1997.

[3] B. Macq, "Lossless multi resolution transform for image authenticating watermarking," *EUSIPCO Proceeding. Tampere: EUSIPCO*, pp. 533-536, 2000.

[4] C. W. Honsinger, P. Jones, M. Rabbani, and J. C. Stoffel, "Lossless recovery of an original image containing embedded data," *U.S. Patent 6*, pp. 278-791, 2001.

[5] C. D. Vleeschouver, J. E. Delaigle, and B. Marq, "Circular interpretation of bijective transformation in lossless watermarking for media asset management," *IEEE Trans Multimedia*, vol. 5, no. 1, pp. 97-105, Mar. 2003.

[6] J. Fridrich, M. Goljan, and R. Du, "Lossless data embedding-New paradigm in digital watermarking," *EURASIP Journal Appl. Signal Process.*, vol. 2, pp. 185-196, 2003.

[7] G. Xuan, J. Zhu, J. Chen, Y. Q. Shi, Z. Ni, and W. Su, "Distortionless data hiding based on integer wavelet transform," *IEE Electron. Lett.,* vol. 38, no. 25, pp. 1646-1648, Dec. 2002.

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

[9] D. M. Thodi and J. J. Rodriquez, "Prediction-error based reversible watermarking," in *Proceeding of IEEE Conf. Image Processing*, pp.1549-1552, Oct. 2004.

[10] D. M. Thodi and J. J. Rodriquez, "Expansion embedding techniques for reversible watermarking," *IEEE Trans Journal Image Process.*, vol. 16, no. 3, pp. 721-730, Mar. 2007.

[11] L. H. J. Kamstra and A. M. Heijimans, "Reversible data embedding into images using wavelet and sorting," *IEEE Tran. Image Process.* vol. 14, no. 12, pp. 2082-2090, Dec. 2005.

[12] V. Sachnev, H. J. Kim, J. Nam, S. Suresh, and Y. Q. Shi, "Reversible watermarking algorithm using sorting and prediction," *IEEE Transaction on Circuit System and Video Technology*, vol. 19, no. 7, pp. 989-999, July 2009.

[13] H. J. Hwang, H. J. Kim, V. Sachnev, and S. H. Joo, "Reversible watermarking method using optimal histogram pair shifting based on prediction and sorting," *KSII Transactions on Internet and Information Systems*, vol. 4, no. 4, 2010.