International Journal of Software Engineering & Computer Sciences (IJSECS)
ISSN: 2289-8522 ,Volume 1, pp.119-129, February 2015

DOI: http://dx.doi.org/10.15282/ijsecs.1.2015.10.0010

REVERSIBLE WATERMARKING BASED ON SORTING PREDICTION ALGORITHM

Khalid Edris, Jasni Mohamed Zain and Tuty Asmawaty Abdul Kadir,

Faculty of Computer Systems & Software Engineering, University Malaysia Pahang
26300 Gambang, Kuantan, Pahang, Malaysia
Email: Khalod07@yahoo.com

ABSTRACT

Reversible watermarking has drawn a lot of interest in recent years. Sachnev et al proposed reversible watermarking algorithm by combining prediction technology, histogram shifting technology and sorting technology, which has good performance. However, their method is against the characteristics of the human visual system. In this paper, we propose a reversible watermarking algorithm to improve Sachnev et al algorithm by using new sorting method. The performance of the proposed reversible watermarking algorithm is evaluated and compared with Sachnev et al method and other methods. The results indicate that the proposed algorithmhas good performance than Sachnev et al method and can embed data with less distortion.

Keywords: Prediction error; Reversible watermarking; Sorting.

INTRODUCTION

In some certain applications areas, such as military, medicine and law, requirements of the integrity of the original carrier are relatively high, even distortion brought by the watermarking is not allowed, which requires lossless embedding watermark information. Reversible watermarking is also called lossless watermark, the non-distortion watermark and the erasable watermark; it can restore the original carrier without distortion after the watermarking information is extracted.

(Tian, 2003) proposed the Difference Expansion Method when the research of reversible watermarking with large capacity began. The image is divided into pairs of pixels, then the differences and average values are calculated, then the binary form is expanded and the watermark is embedded right after most significant bit. One bit can be embedded in every pixels pair.(Kamstra and Heijmans, 2005) enhanced Tian's method. They sorted pairs according correlation between adjacent pixels, thus reduced location map and improved embedding capacity.(Thodi and Rodriguez, 2007)introduced expansion of prediction error plan, they replaced the difference between the adjacent pixels by a pixel prediction, Later they combined the histogram shift algorithm with prediction technology. (Tai et al., 2009) introduced a reversible method based on histogram modification. They used distribution of pixel differences to achieve large hiding capacity and low distortion.(Tsai et al., 2013)presented a reversible algorithm for grayscale images based on the histogram modification technique. A histogram is constructed from the differences between each pixel and its neighbors. They used a modified histogram shifting algorithm to embed a secret message into the pixels. This algorithm can achieve higher embedding capacity and imperceptible distortion. (Sachnev et al., 2009) combined prediction

technology, histogram shifting technology, sorting technology and put forward sorting prediction scheme, which has the best effect among all the mentioned reversible watermarking algorithms. The basic thought of the above algorithm came from Tian's Difference Expansion Method. By extending the vacant position for the watermark embedding, and in order to reduce the distortion, the watermarks are usually embedded on the smooth pixel block, because the smooth area pixel values are close, so that it can provide smaller pixel difference or predict error value. However, embedding watermark in the smooth area is against the characteristics of the human visual system (HVS), since human visual system is not too sensitive to the change of texture in complex area, but is more sensitive to the change of smooth zone. (Kotvicha et al., 2012) developed the technique of sorting absolute prediction error (APE). They used local variance values in predicting APE. This algorithm decreases image distortion and increases the visual quality. (Afsharizadeh and Mohammadi, 2013) extended Sachnev et al scheme by proposing a new sorting technique to improve the hiding capacity and visual quality. They used a new measure for sorting the cells and achieved good performance.

SACHNEV ALGORITM

Sachnev and other members put up with a comprehensive reversible embedding-watermark algorithm based on prediction and sorting by combining histogram shifting algorithm. Compared with other reversible watermarking algorithms, this algorithm has better effect. The following briefly describe the main contents of Sachnev algorithm:

PREDICTION ALGORITHM

We classify all the pixels of image into two groups: one group for embedding-watermark, the other for predictive-value calculation. In Figure 1, five pixels constitute a mark for information embedding, the middle one among which is for embedding-watermark; other four pixels are used to calculate predictive-value$u'_{i,j}$, which can be calculated through the following formula:

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

Then, to calculate the prediction-error:

$$d_{i,j} = u_{i,j} - u'_{i,j} (2)$$

Then, embed the watermark by expanding prediction-error:

$$D_{i,j} = 2 \times d_{i,j} + b \quad (3)$$

b means 1 bit of information, "0" or "1". $D_{i,j}$is the prediction-error after embedding-watermark.

The pixel-value $U_{i,j}$ after $u_{i,j}$ takes embedding-watermark:

$$U_{i,j} = D_{i,j} + u'_{i,j} (4)$$

After the embedding-watermark, extracting-watermark is the inverse process of embedding-watermark. As at the end of embedding-watermark, the four pixel-values used for predict do not change, thus the predictive-value $u'_{i,j}$ does not change which can be calculated out with the four pixels. The watermark-information *b* and the original pixel-value can be calculated out through the predicted-value $u'_{i,j}$ and the pixel-value $U_{i,j}$ after the embedding watermark.

$$D_{i,j} = U_{i,j} - u'_{i,j} (5)$$

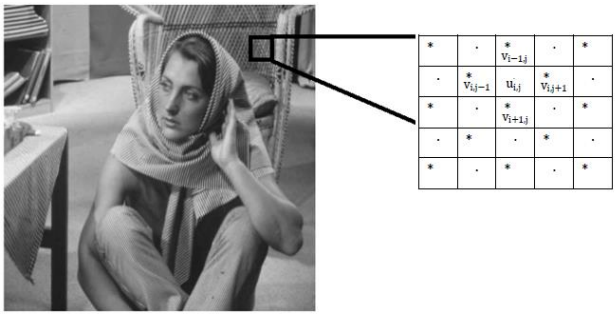

Figure 1. Prediction pattern

The embedded watermark-information can be calculated though the following formula:

$$b = D_{i,j} \bmod 2 \quad (6)$$

The prediction-error before embedding watermark:

$$d_{i,j} = \left\lfloor \frac{D_{i,j}}{2} \right\rfloor \quad (7)$$

The pixel-value of host pixel:

$$u_{i,j} = u'_{i,j} + d_{i,j} (8)$$

Thus extracting-watermark and the recovery of host image have been completed.

HISTOGRAM SHIFTING

The algorithm put up by Sachnev used the Histogram-shifting algorithm proposed by Thodi and

other members. Histogram-shifting algorithm is a more effective reversible embedding-watermark algorithm, which is often used to avoid the problem of value overlapping for the enlargement of value difference. This algorithm uses two threshold values, namely lower limit$T_n$and upper limit value T_p. All the prediction-error value between $[T_n,T_p]$ will be used to enlarge for embedding-watermark. Those prediction-error values outside the interval will be shifted to avoid the problem of value overlapping for the enlargement of value difference. The histogram shift encoding algorithm modifies prediction errors $d_{i,j}$ as follows:

$$D_{i,j} = \begin{cases} 2 \times d_{i,j} + b, \ if\ d_{i,j} \in [T_n, T_p] \\ d_{i,j} + T_p + 1, if\ d_{i,j} > T_p\ and\ T_p \geq 0 \\ d_{i,j} + T_n, if\ d_{i,j} < T_n\ and\ T_n < 0 \end{cases} \quad (9)$$

The watermark can be extracted and the host-image pixel-value can be recovered at the end of extracting-watermark according to the following formula:

$$d_{i,j} = \begin{cases} \left\lfloor \frac{D_{i,j}}{2} \right\rfloor, if\ \ D_{i,j} \in [2T_n, 2T_p + 1] \\ D_{i,j} - T_p - 1, if\ D_{i,j} > 2T_p + 1\ and\ T_p \geq 0 \\ D_{i,j} - T_n, \ if\ D_{i,j} < 2T_n\ and\ T_n < 0 \end{cases} \quad (10)$$

$$b = D_{i,j} \bmod 2, \ D_{i,j} \in [2T_n, 2T_p + 1] \quad (11)$$

The embedded capacity can be controlled by controlling the value of T_n, T_p, but the distortion of the image will be influenced when the capacity is controlled. Therefore, it's very important to choose a suitable threshold limit value.

SORTING ALGORITHM

To ensure distortion of image to the least after embedding-watermark, the order of embedding-watermark in the image becomes very important. Therefore, to take embedding-watermark in order from the up to down or the left to the right for the pixel-blocks which can take embedding-watermark to sort effectively can minimize the distortion of the image. The sorting-method put up by Sachnev and other members calculate a variance yield according to the pixel-value for prediction in each pixel-block, and then sort according to the variance yield. The formula to calculate variance yield is:

$$\mu_{i,j} = \frac{1}{4}\sum_{k=1}^{4}(\Delta v_k - \Delta\bar{v}_k)^2 \quad (12)$$

Where, $\Delta v_1 = \left|v_{i,j-1} - v_{i-1,j}\right|$, $\Delta v_2 = \left|v_{i-1,j} - v_{i,j+1}\right|$, $\Delta v_3 = \left|v_{i,j+1} - v_{i+1,j}\right|$, $\Delta v_4 = \left|v_{i+1,j} - v_{i,j-1}\right|$, $\Delta\overline{v_k} = (\Delta v_1 \ + \Delta v_2 \ + \Delta v_3 \ + \Delta v_4 \ )/4$ (13)

Refer to figure 1, in the above formula, $\mu_{i,j}$can be calculated with the pixel-value for prediction. As the pixel-value for prediction doesnot change after embedding-watermark, the embedding-watermark doesnot change.

PROPOSED ALGORITHM

The sorting-method put up by Sachnev applies $\mu_{i,j}$as asorting-parameter has a good effect. However, the proposed sorting algorithm in this paper comes out with better results than Sanchevalgorithm. The value of $\mu_{i,j}$is the difference variance of the four pixel-values for prediction. The difference variance of these four pixel-blocks can reflect the smooth level of the pixel-block. If $\mu_{i,j}$ is small, it means the pixel-block is smooth. If $\mu_{i,j}$ is big, it means the pixel-block is rich in textures. The core concept of sorting according to the variance value is to calculate the value of prediction-error for prediction,it can be used to compare the smooth level of the pixel-block for prediction, the value of prediction-error in smooth pixel-block is relatively less,but $\mu_{i,j}$ is not the best sorting algorithm for predicting the value of prediction-error.

The sorting-parameter $u'_{i,j}$ used in the proposed algorithm is more accurately and effectively in predicting the value of prediction-error.

In equation (12) $v_k$stands for the four pixel-values for prediction in Figure 1, $\bar{v}_k$ means the mean value of the four pixel-values, while $\mu_{i,j}$ is the variance value of the four pixels.The formula to calculate the prediction-error value is $d_{i,j} = u_{i,j} - u'_{i,j}$, in which $u'_{i,j}$ is the predictive-value of the four pixel-values mentioned above.Prediction-error is achieved through the pixel-value minus the mean value of these four pixels, and $\mu_{i,j}$is the variance of the four pixel-values. Thus $u'_{i,j}$ is suitable to predict the value of prediction-error. There are two key features:

1. The value of $u'_{i,j}$ keeps the same after embedding-watermark.

2. The value of $u'_{i,j}$ can predict the value of prediction-error more accurately.

HVS-BASED EMBEDDING-WATERMARK ALGORITHM

Considering the influence of HVS, people are very sensitive to the change in the even area while not that sensitive to the change in the area rich in textures, therefore, the embedding-watermark in texture area can improve the visual quality greater. However, less distortion of image is brought out after embedding-watermark. As the pixel-values differs greatly in the texture area, the prediction-error value offered is big, thus there is a big image distortion when reversible-watermarking-algorithm is embedded by the method of enlargement. Therefore, we suggest embedding-watermark of the smooth pixel-block in the texture area, choosing the smooth pixel-block in the big texture area to take embedding-watermark first. Thus it can ensure to get a higher visual quality at the same distortion level.

Divide the image into 64 blocks before embedding-watermark. The images tested in the proposed method are 512×512, therefore, each block contains 64×64 pixels. Then calculate the

variance of each block to sort. To ensure the parameters sorting not to change before or after embedding-watermark, here the predictive pixels only are chosen among each block for calculating the variance of this block:

$$MSE = \frac{1}{2048}\sum_{i=1}^{2048}(v_i - \bar{v})^2 \quad (14)$$

$v_i$in the above formula shows 2048 pixel-value for prediction in each block, and the pixel-value will not change in embedding-watermark, $\bar{v}$ standing for the mean value of 2048 pixel-value. If the *MES* is big, it means it is rich in texture area. If the *MES* is small, it means this block is smoother. Sort these64 blocks according to the *MSE* of each block and give each block a parameter value K_i after sorting.$K_i = i$and$1 \leq K_i \leq 64$. The parameter of the block with most complicated texture is $K_1 = 1$, the parameter of the smoothest block is $K_{64} = 64$, all the pixel-value in a block shares a parameter value K_i.

The prediction-error value of the same level means the image distortion is the same in embedding-watermark. However, these values may lie in the texture area or the smooth area. In this condition, if the pixel in texture area can take embedding-watermark first, it can bring a higher visual quality.

$$FV_{i,j} = u'_{i,j} \times C + K_i \ , C = 64 \quad (15)$$

By revising the sorting-parameter $u'_{i,j}$, $FV_{i,j}$ comes out in the Eq.(15) and finally be used to sort.

In formula 14, *C* is a constant value, meaning 64 here. As the image is divided into 64 blocks in the paper, *C* is the number of blocks after the image being divided and$K_i$is the parameter of each block mentioned above. The main function of C is to identify different sorting space whileK_i is to identify the embedding order of pixel in the same sorting-parameter.

At the end of embedding-watermark, partial additional information, *e.g.* the embedding capacity, should be added onto the watermark in order to extract the watermark correctly. Then the capacity we embedded can end the extracting-watermark when the extracting-watermark extracts. In extracting-watermark, we should divide the image into 64 blocks first and then we calculate the *MSE* of each block. Then we sort each block according to the *MSE* and give each of them a parameter K_i.$K_i = i$and $1 \leq K_i \leq 64$. As only the still embedding-watermark is chosen to calculate *MSE*, the pixel-value for prediction, the *MSE* of each block at the end of extracting-watermark and the value at the embedding-watermark is the same. As the value of $u'_{i,j}$ embedding-watermark doesnot change, the value of $FV_{i,j}$ remains before and after embedding-watermark. Therefore, a same sorting result as that of embedding-watermark end can be achieved through sorting with $FV_{i,j}$ at the end of extracting-watermark, which is the precondition for the normal extracting-watermark.

HVS-BASED REVERSIBLE-WATERMARKING-ALGORITHM

The proposed algorithm mainly states the embedding-watermark and the detail extraction pattern by the major techniques mentioned above.

A. EMBEDDING-WATERMARK

Step 1: Classify all the pixels into two groups, one for embedding-watermark, and the other for prediction. Then divide the image into 64 blocks, the carrier image is 512×512 large, thus the pixel of each block after division is 64×64. Then calculate the *MSE* of each block, and choose the pixel for prediction only when calculating the *MSE*. Sorting of the 64 blocks according to the *MSE* value, and then give each block a parameter value K_i, $K_i = i$, among which the parameter value of the smoothest block is 64 and that of the one with the most textures is $K_1 = 1$.

Step 2: Calculate the predictive-value $u'_{i,j}$, prediction-error $d_{i,j}$, sorting-parameter of each pixel-block $FV_{i,j}$. Then rank the order of all the pixel-blocks according to the calculated value of$FV_{i,j}$. Collect the first *LSB* value of first 34 prediction-errors to be $S_{LBS}$and then constitute the watermark-information as a part of watermark.

Step 3: Find the most suitable threshold limit value T_n, T_paccording to the capacity and the rank result of embedding-watermark. Test the pixel-block of the image according to the pattern of controlling over-flow. Classify the pixel-block into group *A*, *B* and *C*, and then build a location plan (if there is a location plan).

Step 4:Embed the watermark and location plan into the carrier image according to the Histogram-shifting-algorithm mentioned. From the 35th pixel-block of embedding-watermark on, those pixel-block in group *A* can have embedding-watermark, those pixel-block in group *B* will not haveembedding-watermark but take some revision and those in group *C* will skip over without any revision.

Step 5: Replace the *LSB* of the first 34 prediction-error after sorting with the 34-bit information with additional information.

Step 6: After the first embedding-watermark, change the roles of the two groups of pixel-block for the second embedding-watermark. Here the embedding-watermark ends.

B. EXTRACTING-WATERMARK AND THERECOVERY OF HOSTIMAGE

Step 1: Classify the pixels into two groups, and then divide the image into 64 blocks. Then calculate the *MSE* of each block. Sort the 64 blocks according to the *MSE* value, and then give each block a parameter value $K_i, K_i = i$, among which the parameter value of the smoothest block is 64 and that of the one with the most textures is $K_1 = 1$.

Step 2: Calculate the predictive-value$u'_{i,j}$, prediction-error$d_{i,j}$,the sorting-parameter of each pixel-block $FV_{i,j}$. Then rank the order of all the pixel-blocks according to the calculated value$FV_{i,j}$.

Step 3: Read the *LSB* value of the first 34 prediction-error according to the rank result, from which the threshold value T_n, T_p, the embedding-watermark capacity *P* come out.

Step 4: Test the pixel-block of the image according to the pattern of controlling overflow and

classify the pixel-blocks into three groups, namely group *A*, group *B* and group *C*. According to the histogram shifting algorithm, to have embedding-watermark from the 35th pixel-block on, those pixel-block in group *A* can have embedding-watermark, those pixel-blocks in group *B* will not have embedding-watermark but take some revision and those in group *C* will skip over without any revision.

Step 5: Extract from the watermark-information and replace the *LSB* of the first 34 prediction-error, recover the value of the first host 34 prediction-error.

Step 6: After the first extracting-watermark, to take the second extracting-watermark. Here the whole extracting-watermark ends.

EXPERIMENTAL RESULTS

The proposed algorithm is compared with the methods of (Sachnevet al, 2009), (Kotvicha et al, 2012), and (Afsharizadeh and Mohammadi, 2013), the experiment in this paper applies 4 experimental images as in figure2, which are 512×512 grayscale images, namely Lena, Airplane, Baboon and Barbara, which is similar to the testing images applied in the above methods.

Figure 2. (a) Lena. (b) Airplane. (c) Baboon. (d) Barbara.

Figure 3, 4, 5, and 6 describe the PSNR effect of the four different images by using the proposed sorting algorithm and the other methods. Comparing with Sachnev's method, the proposed sorting algorithm improve the PSNR because the parameter used in sorting is more effective in predicting the values of prediction error, which improve more the relatively smooth images comparing with the relatively texture images.

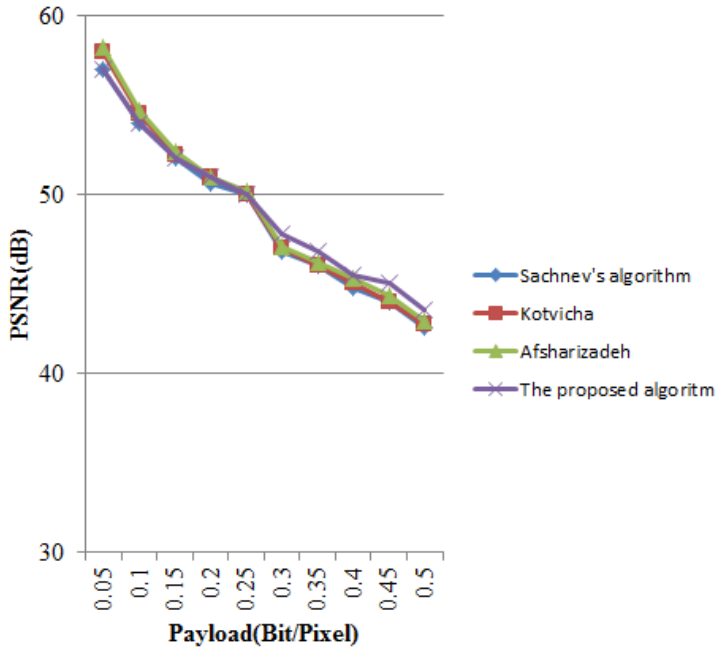

Figure 3. Embedding capacity vs PSNR for the Lena image.

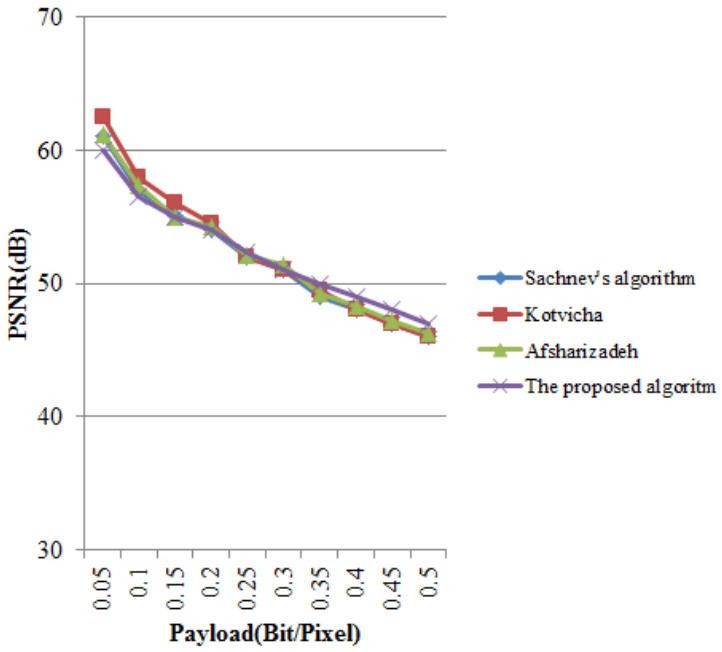

Figure 4. Embedding capacity vs PSNR for the Airplane image.

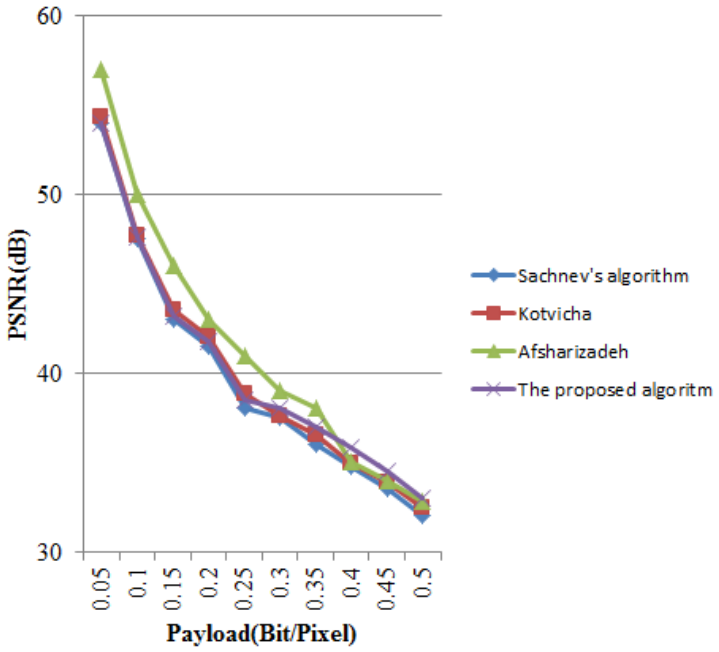

Figure 5. Embedding capacity vs PSNR for the Baboon image.

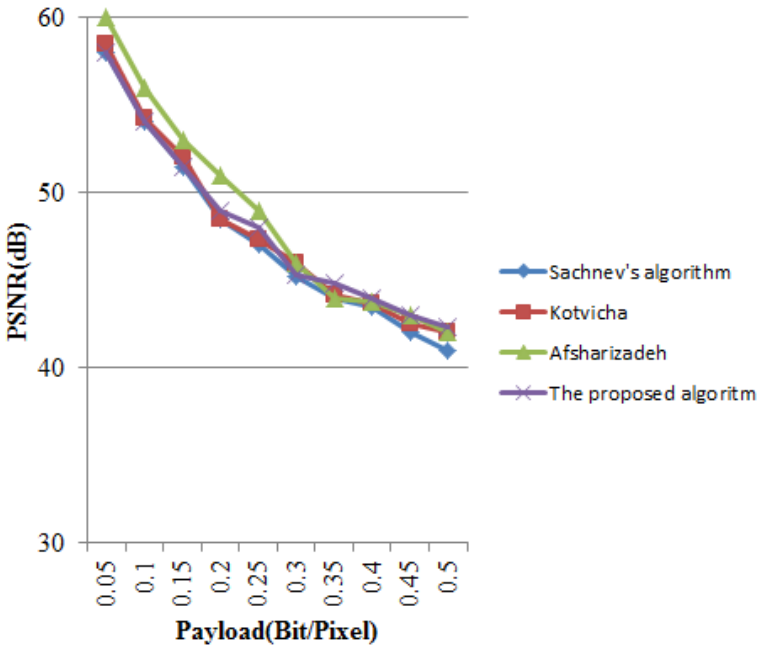

Figure 6. Embedding capacity vs PSNR for the Barbara image.

(Kotvicha et al, 2012), and (Afsharizadeh and Mohammadi, 2013) methods showed good performance when the payload is small, while the proposed algorithm has better performance when the payload is big.

Experimental results show that the proposed algorithm by this paper enjoys efficiency, effectiveness and feasibility in the treatment of the issue of the sorting of prediction error.

CONCLUSION

This paper proposed a new reversible watermarking algorithm based on Sachnevalgorithm. The experimental results show that the new proposed algorithm is more effective than Sachnevalgorithm and other algorithms when the payload is big, because the sorting-parameter $u'_{i,j}$ used in the proposed algorithm is more accurately and effectively in predicting the value of prediction- error. In this algorithm we suggested embedding-watermark of the smooth pixel-block in the texture area, choosing the smooth pixel-block in the big texture area to take embedding-watermark first. Thus it can ensure to get a higher visual quality. After ranking all the pixel-blocks based on the calculated value of $FV_{i,j}$ we embed the watermark and location plan into the carrier image according to the Histogram-shifting-algorithm.

REFERENCES

Afsharizadeh, M. and Mohammadi, M. 2013.A Reversible Watermarking Prediction Based Scheme using a New Sorting Technique. IEEE Transactions on Information Security and Cryptology, 1-5.

Kamstra,L. H. J. andHeijmans A. 2005.Reversible data embedding into images using wavelet techniques and sorting. IEEE Transactions on Image Processing, 14(12):2082–2090.

Kotvicha, A, Sanguansat, P. and Kasemsa, M. L. K. 2012.Expand Variance Mean Sorting for Reversible Watermarking. International Journal of Computer and Communication Engineering, 1(3).

Tai, W. L., Yeh, C. M., and Chen C. 2009. Reversible Data Hiding Based on Histogram Modification of Pixel Differences. IEEE Transactions on Circuits and Systems for Video Technology, 19(6): 906-910.

Thodi, D. M., and Rodriguez J. J. 2007.Expansion embedding techniques for reversible watermarking. Image Processing. IEEE Transactions on Image processing, 16(3):721-730.

Tian, J. 2003.Reversible data embedding using a difference expansion. IEEE Transactions on Circuits and Systems for Video Technology, 13(8): 890-896.

Tsai, Y. Y., Tsai, D. S., and Liu C. L. 2013.Reversible data hiding scheme based on neighboring pixel differences. Digital Signal Processing, 23(3):919–927.

Sachnev, V., Kim, H. J., Nam, J., Suresh, S., and Shi, Y. Q. 2009.Reversible watermarking algorithm using sorting and prediction.IEEE Transactions on Circuits and Systems for Video Technology, vol. 19, no. 7, 989-999.