A STUDY ON REVERSIBLE WATERMARKING USING THE MULTI-LEVEL SCHEME FOR MOSAIC IMAGES

Razvan IORDACHE[1], Dinu COLTUC[2]

This paper proposes a multiple layer combinations reversible watermarking scheme. We use a combination of predictors for calculating the new value of central pixel, as you can see in this paper we use a circular predictor, a cross one and also a square predictor, taking in consideration all the possibilities to mark the pixels. Also, an advantage of our method is the one of splitting the image into more than two sets of pixels, we create an advantage because we can cover better the image, and use precise predictors in different situations (we cover all type of situations – when we have shapes that must be recovered in decoding stage, but also when we have flat images). In the end, our results are obtained for mosaic images, we had this type of choice because using this type of test images we can do a division between the three channels of colors and can compute our algorithm much more in depth of details.

Keywords: reversible watermarking, multi-layer embedding, mosaic images, Bayer color filter array

1. Introduction

In reversible watermarking the embedded data is extracted at decoding stage and original image is recovered without any loss of information. Over time increased attention was directed towards the methods based on difference expansion or the ones based on histogram shifting, these being the most used.

The principle is to modify the pixels such that space is created for embedding new bits of information. Space created is usually two times bigger than the prediction error and is in fact a two times multiplication that assures that the least significant bit of the prediction error is set to zero. Through history few methods have been noticed among others. One of them is the median edge detector (MED) of JPEG-LS [3]. This method has been used by [1, 2] and its prediction context is formed with three pixels only, the right, lower and lower-diagonal pixels near the central one that needs to be predicted. It has different cases, if there is a vertical edge near the current location the lower vertical pixel is

[1] PhD student, Department of Applied Electronics, Faculty of Electronic, Telecommunications and Information Technology, University POLITEHNICA of Bucharest, Romania, e-mail: iordache.razvan.87@gmail.com

[2] Prof., Dept. of Applied Electronics, University VALAHIA of Targoviste, Romania, e-mail: dinu_coltuc@yahoo.com

picked, if there is a horizontal edge the right pixels is selected or a linear combination of the three pixels otherwise.

Another known predictor is the gradient-adjusted predictor (GAP), which was used by [4]. It has better results than MED and it works on a local context composed by 7 pixels, assuring a more precise prediction. There is one simplified version, called SGAP, it provides similar results, but the reduced number of pixels is transposed in lower cost. One of the most efficient method is the one proposed by Sachnev [6]. In his method, the predictor computes the average on the rhombus of the four pixels displayed around the central one, one of the advantages is that his predictor is a noncausal predictor while MED, GAP are causal predictors and therefore it uses all the information from the neighborhood around pixel, not only a part like the causal predictors. Many tried to achieve the performance of Sachnev's predictor and a lot of new papers are inspired by his method [7–11].

In this paper, we try to improve the method proposed by Yang et al in paper [12], where they use a Bayer Filter Mosaic (BFM). BFM is a color filter array (CFA) for arranging RGB color filters on a square grid of photosensors. Its particular arrangement of color filters is used in most single-chip digital image sensors used in digital cameras, camcorders, and scanners to create a color image. Our method uses a combination of predictors, using in the end all the pixels around the one that is being marked. Using this type of combination, we can utilize circular predictor (where all pixels are used), the cross (we only use the two vertical and the two horizontal neighbors of central pixel) and square (the two pairs of pixels positioned on the two diagonals that intersects the central pixel).

2. Proposed scheme

In the proposed scheme, we first split the images into the three levels of colors, and we obtain one image for green channel of color, one for blue and another for red. With these three images obtained we then start to mark all the three images after the following algorithm – we consider four sets of pixels in which we use three different predictors, for the first layer we use the circular predictor - $\boldsymbol{circ}_{i,j}$, which is the mean of other eight neighbors of central pixel $\boldsymbol{x}_{i,j}$:

$$\boldsymbol{circ}_{i,j} = \frac{x_{i-1,j-1}+x_{i-1,j}+x_{i-1,j+1}+x_{i,j-1}+x_{i,j+1}+x_{i+1,j-1}+x_{i+1,j}+x_{i+1,j\_1}}{8}, \quad (1)$$

On the second layer, we insert information with the cross predictor - $\boldsymbol{cross}_{i,j}$, which uses the four strong neighbors of the central pixel, two on vertical and other two on horizontal, computing a mean of them:

$$\boldsymbol{cross}_{i,j} = \frac{x_{i-1,j}+x_{i+1,j}+x_{i,j-1}+x_{i,j+1}}{4}, \quad (2)$$

In the following step, we insert information with the square predictor - $square_{i,j}$ which is a combination of the other four neighbors of central pixel, the ones from both diagonals, the predictor calculates their mean and the obtained value is used to mark all the pixels from this set.:

$$square_{i,j} = \frac{x_{i-1,j-1}+x_{i-1,j+1}+x_{i+1,j-1}+x_{i+1,j+1}}{4}, \tag{3}$$

In the end, we conclude with the fourth layer where we use again the circular predictor computer with (1).

In every layer, we calculate the error prediction - $e_{i,j}$, error which is the difference between the original value of pixel $x_{i,j}$ and the new estimated valued - $\hat{x}_{i,j}$, computed for every layer:

$$e_{i,j} = x_{i,j} - \hat{x}_{i,j}, \tag{4}$$

The value b can embed information between 0 and 1 bit, this value can be recovered in the process of decoding. So, the watermarked pixel - $X_{i,j}$ is computed with the following relation:

$$X_{i,j} = x_{i,j} + 2 \cdot e_{i,j} + b_{i,j}, \tag{5}$$

The inserted value of information it is correlated with the error prediction, and to control the distortion a threshold T>0 is inserted. If the prediction error is less than the threshold and no overflow or underflow is generated, the pixel is transformed and the information is embedded. The histogram shift is used for classifying the way of inserting information into the image. If the prediction error is not situated between (−T, T) then we shift the histogram and store the location where we have underflow/overflow in a location map:

$$X_{i,j} = \begin{cases} x_{i,j} + T, & if\ e_{i,j} \geq T \\ x_{i,j} - (T-1), & if\ e_{i,j} \leq -T \end{cases} \tag{6}$$

3. Proposed embedding scheme

Storing the location of the pixels that generate overflow/underflow will decrease the capacity of the embedding. Every time when a pixel generates overflow/underflow its location is stored in a location map. This will be helpful at decoding stage, where we will process the marked imaged in the reverse order. In the decoding stage, we have the location map as input argument and read the image in reverse order. So, if in the marking process we start from the upper left corner and modify the image, in the decoding stage we start from the lower right corner of the image and unmark all the pixels, using the predictors in reverse order also – circular first, square, cross and again circular.

4. Experimental results

For this paper the images used were the ones from the KODIM (Kodak images) data base, the selected images are presented below – Kodim03, Kodim07, Kodim08, Kodim11, Kodim22, Kodim23.

Fig. 1 – Test images used from Kodak set, in order Kodim03, Kodim07, Kodim08, Kodim11, Kodim22, Kodim23

We will compare our work with the one form Yang et al papers, but also for a more accurate exemplification we included also the method proposed by Sachnev. First of all, the original image is divided into three other images, and for all three methods we compute the enumerated algorithms.

Table 1

PSNR values for image Kodim03

	Our		Sachnev		Yang	
	Max	Mean	Max	Mean	Max	Mean
Blue channel	43.49	32.21	43.43	32.39	42.45	30.78
Red channel	43.53	31.90	43.46	32.16	42.84	31.94

Table 2

PSNR values for image Kodim07

	Our		Sachnev		Yang	
	Max	Mean	Max	Mean	Max	Mean
Blue channel	43.09	32.35	43.12	33.60	42.36	31.43
Red channel	43.22	30.94	43.20	32.20	42.54	32.45

Table 3

PSNR values for image Kodim08

	Our		Sachnev		Yang	
	Max	Mean	Max	Mean	Max	Mean
Blue channel	39.25	26.88	39.29	27.43	42.61	28.35
Red channel	37.07	25.51	37.13	26.84	39.44	28.51

Our method is represented with a blue line, Sachnev's with green, and Yang's with red (as you can see in Fig. 2)..

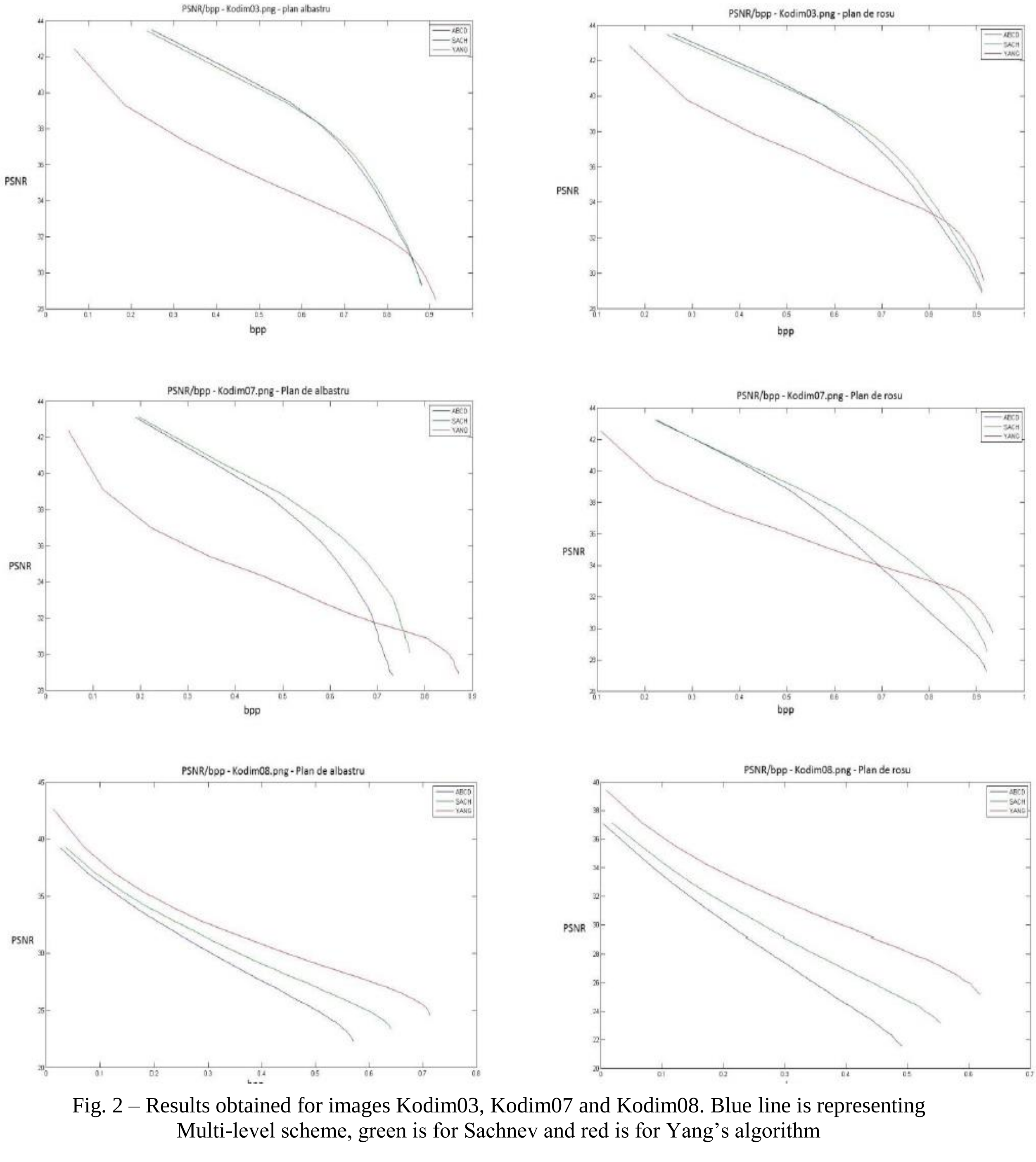

Fig. 2 – Results obtained for images Kodim03, Kodim07 and Kodim08. Blue line is representing Multi-level scheme, green is for Sachnev and red is for Yang's algorithm

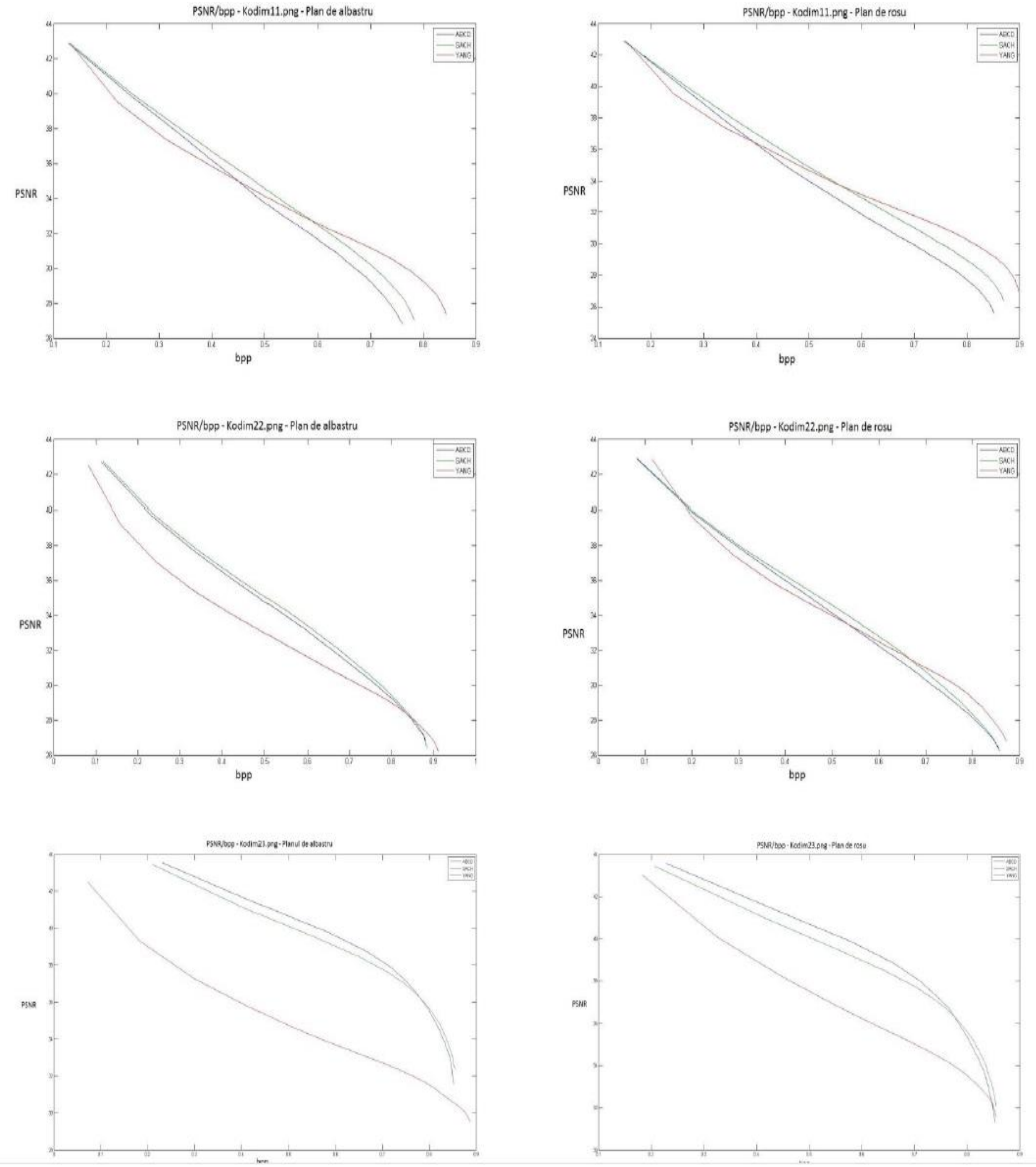

Fig. 3 – Results obtained for images Kodim11, Kodim22 and Kodim23. Blue line is representing Multi-level scheme, green is for Sachnev and red is for Yang's algorithm

The Maximum and Median values for obtained PSNR shows that our method is competitive on images with objects well defined, both channels were marked with success and the method of inserting was more efficient than Yang's (we achieved a higher PSNR value, and the median values computed were in front). On the other hand, for images very detailed with more objects superposed or in images where many edges are implied our method is not so good as Yang's algorithm, but we are very close of Sachnev's work, managing to obtain almost the same value for Maximum PSNR. The method is close to Yang's results as far as we inserted 0.5 – 0.6 bits per pixel, after this value, Yang's work it is gaining advantage, on both channels red or blue. The results can be seen in the Fig. 3, below this paragraph. Other results for test images are shown here and you can see that again for images that have an amount of details (edges, shapes) our method manages to stay close to other two methods (for 0.7-0.8 bits per pixel inserted we have PSNR values close to others), and for other images Yang's algorithm is not so efficient, our work and Sachnev's stay close one of each other, but our method is slightly better. With this algorithm, we manage to top in all the images Sachnev's method, and in comparison with Yang's algorithm in most images we exceed their results, being better on blue and red channels.

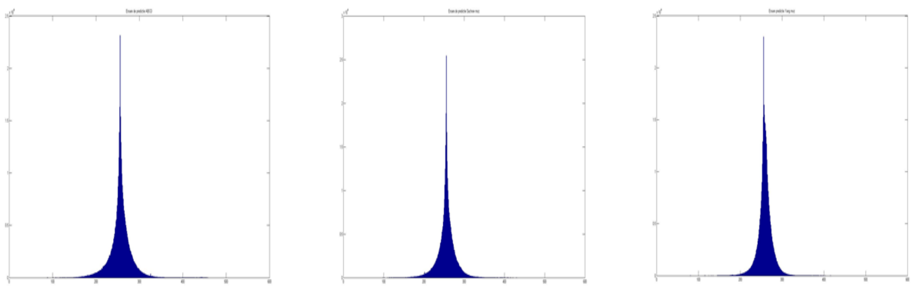

Fig. 4 – Prediction error computer for our method, Sachnev's method and Yang

Also, the prediction error obtained by us is better than the one used by Yang et all in their paper, but both methods are surpassed by Sachnev's prediction error. The error prediction is computed from the mosaic image marked with all the three methods. After we mark all the three channels (green, red, blue) we recomputed the mosaic image and for a threshold T prediction error was generated. As you can see for Kodim23 the results of all three methods are printed below, with Sachnev in front, but with our method over Yang's algorithm.

6. Conclusions

We tried with this paper to compete with other works in the field of the reversible watermarking for mosaic images, and that is why one of the works that was introduced for comparison is the one of Yang et al. We managed to demonstrate that we can compete with his method, and also can surpass Sachnev's robust algorithm. Our method is better than the other two ones because it uses more than one predictor and that is why it is more adaptable on the test image, we use all the pixels around the central one, and also we split the image into four layers of pixels, so the problem is threated in very depth details. In future works we want to try to improve this algorithm and to adapt it to the multi-bit reversible watermarking, because we think that keeping this way of inserting information, we can keep the error prediction high, and the PSNR can improve also.

Acknowledgements

Part of the research presented in this paper was supported by the Sectorial Operational Program Human Resources Development, 2014-2015, of the Ministry of Labor, Family and Social Protection through a Financial Agreement between University POLITEHNICA of Bucharest and AM POSDRU Romania, project ID 132395.

R E F E R E N C E S

[1] *D. M. Thodi* and *J. J. Rodriguez*, "Expansion Embedding Techniques for Reversible Watermarking", IEEE Trans. Image Process., vol. 15, pp. 721–729, 2007.

[2] *Y. Hu, H.-K. Lee* and *J. Li*, "DE-based reversible data hiding with improved overflow location map", IEEE Trans. Circuits Syst. Video Technol., vol. 19, no. 2, pp. 250–260, 2009.

[3] *M. Weinberger, G. Seroussi* and *G. Sapiro*, "The LOCO-I lossless image compression algorithm: Principles and standardization into JPEG-LS", IEEE Trans. on Image Process., vol. 9, no. 8, pp. 1309–1324, 2000.

[4] *X. Wu and N. Memon*, "Context-based, adaptive, lossless image coding", IEEE Trans. on Commun., vol. 45, no. 4, pp. 437-444, 1997.

[5] *D. Coltuc*, "Improved Embedding for Prediction Based Reversible Watermarking", IEEE Trans. Inf. Forensics Security, vol. 6, no. 3, 2011.

[6] *V. Sachnev, H. J. Kim, J. Nam, S. Suresh* and *Y. Q. Shi*, "Reversible Watermarking Algorithm Using Sorting and Prediction", IEEE Trans. Circuits Syst. Video Technol., vol. 19, pp. 989–999, 2009.

[7] *B. Ou, X. Li, Y. Zhao*, and *R. Ni*, "Reversible data hiding based on PDE predictor", Journal of Systems and Software, vol. 86, no. 10, pp. 2700-2709, 2012.

[8] *I.-C. Dragoi, D. Coltuc*, "Improved Rhombus Interpolation for Reversible Watermarking by Difference Expansion", Proc. 20th European Conf. on Signal. Process., EUSIPCO'2012, pp. 1688–1692, 2012.

[9] *W. Zhang, X. Hu, X. Li,* and *N. Yu*, "Recursive Histogram Modification: Establishing Equivalency between Reversible Data Hiding and Lossless Data Compression", IEEE Trans. on Image Process., vol. 22, no. 7, pp. 2775-2785, 2013.

[10] *X. Zhang*, "Reversible Data Hiding with Optimal Value Transfer", IEEE Trans. on Multimedia., vol. 15, no. 2, pp. 316-325, 2013.

[11] *B. Ou, X. Li, Y. Zhao, R. Ni* and *Y. Shi*, "Pairwise prediction-error expansion for efficient reversible data hiding", IEEE Trans. on Image Process., vol. 22, no.12, pp. 5010 - 5021, 2013.

[12] *Wei-Jen Yang, Kuo-Liang Chung* and *Hong-Yuan Mark Liao*, "Efficient reversible data hiding for color filter array images", Inform. Sciences (2011).