IMPROVED RHOMBUS INTERPOLATION FOR REVERSIBLE WATERMARKING BY DIFFERENCE EXPANSION

Catalin Dragoi, Dinu Coltuc

Electrical Engineering Dept.,
Valahia University of Targoviste, Romania
email: {dragoi,coltuc}@valahia.ro

ABSTRACT

The paper proposes an interpolation error expansion reversible watermarking algorithm. The main novelty of the paper is a modified rhombus interpolation scheme. The four horizontal and vertical neighbors are considered and, depending on their values, the interpolated pixel is computed as the average of the horizontal pixels, of the vertical pixels or of the entire set of four pixels. Experimental results are provided. The proposed scheme outperforms the results obtained by using the average on the four horizontal and vertical neighbors and the ones obtained by using well known predictors as MED or GAP.

Index Terms— Interpolation error expansion, modified rhombus interpolation, MED, GAP

1. INTRODUCTION

Reversible watermarking completely removes the embedded watermark and exactly recovers the original image. So far, several approaches have been developed for reversible watermarking. The most fruitful research direction appears to be the difference expansion (DE) reversible watermarking.

Introduced by Tian, [1], the DE reversible watermarking creates space by expanding two times the difference between pairs of adjacent pixels. Bit by bit, the data and the auxiliary information are further added to the expanded differences and, if no overflow or underflow appears, they are embedded into the image. At detection, the embedded data is extracted from the expanded difference and the original pixels are recovered.

Since the lower the difference to be expanded, the better the results, significant research is devoted to reduce this difference. Thodi et. al, [2], replaced the use of the simple difference between adjacent pixels, with the prediction error obtained for the median edge detector (MED) predictor. MED is a high performance predictor already used in JPEG-LS standard [3]. With MED, the prediction context is composed of the right, lower and lower-diagonal neighbors of a pixel. The predictor tends to select the lower vertical neighbor in cases where a vertical edge exists right to the current location, the right neighbor in cases of a horizontal edge below it, or a linear combination of the context pixels if no edge is detected. Many reversible watermarking schemes are based on MED predictor [2, 4, 5]. Improved results are obtained by using better predictors, as, for instance, the gradient-adjusted predictor [6, 7, 8]. The gradient-adjusted predictor, GAP, is used in CALIC (context-based, adaptive, lossless image coding) algorithm [9]. The GAP predictor is more complex than the MED one. The prediction context is extended from 3 to 7 pixels. Not only the existence of a horizontal/vertical edge is detected, but also its strength (weak, normal or sharp).

Sachnev et al. proposed an improved scheme based on the expansion of the interpolation error [10]. The interpolated pixels is computed as the simple average of the four horizontal and vertical neighbors. The estimation of a pixel as the average on its rhombus neighborhood appears to outperform the prediction provided by MED or GAP. This is due to the fact that one computes the pixel over the entire neighborhood and not only on a part of it.

The simple average does not take into account the presence of details as MED or GAP do. In order to reduce the interpolation error, the paper proposes an adaptive interpolation scheme. The four horizontal and vertical neighbors are considered and, depending on their values, the interpolated pixel is computed as the average of the horizontal pixels, of the vertical pixels or of the entire set of four pixels.

The outline of the paper is as follows. The proposed adaptive interpolation scheme is presented in Section 2. The watermarking scheme is briefly discussed in Section 3. Experimental results are provided in Section 4. Finally, the conclusions are drawn in Section 5.

2. ADAPTIVE PIXEL INTERPOLATION

The interpolation scheme used in [10] considers the rhombus neighborhood. The pixel $x_{i,j}$ is estimated as the average of

This work was supported by the Romanian National Authority for Scientific Research, UEFISCDI, project PN-II-PT-PCCA-2011-3.2-1162.

the four horizontal and vertical neighbors:

$$\hat{x}_{i,j} = \left\lfloor \frac{x_{i-1,j} + x_{i+1,j} + x_{i,j-1} + x_{i,j+1}}{4} \right\rfloor \quad (1)$$

where $\lfloor x \rfloor$ represents the greatest integer less than or equal to x.

The above scheme is of very low complexity. If the central pixel is in a uniform region, one expects to have a very good estimate. If the pixel is not in a uniform region, the estimation error can be rather large.

The proposed approach considers the same rhombus neighborhood. If the pixel belongs to a uniform region, the central pixel is estimated as in equation (1). If not, the rhombus is split in two groups: horizontal and vertical neighbors. The homogeneity of each group is checked. Furthermore, the central pixel is considered as belonging to the most homogeneous group and it is estimated as the average of the group. More precisely, let $x_{i,j-1}$, $x_{i,j+1}$ be the horizontal neighbors and let $x_{i-1,j}$, $x_{i+1,j}$ be the vertical ones. The homogeneity of each group is determined by simply computing the distance between the pixels:

$$d_h = |x_{i,j-1} - x_{i,j+1}|\,,\; d_v = |x_{i-1,j} - x_{i+1,j}| \quad (2)$$

The uniformity of the rhombus is decided by computing the distance between the averages of horizontal and vertical groups, D and by checking D against a threshold T_u. Let

$$D = \left| \frac{x_{i,j-1} + x_{i,j+1}}{2} - \frac{x_{i-1,j} + x_{i+1,j}}{2} \right| \quad (3)$$

The predicted value $\hat{x}_{i,j}$ is computed as:

$$\hat{x} = \begin{cases} \left\lfloor \frac{x_{i,j-1}+x_{i,j+1}}{2} \right\rfloor, & \text{if } d_h < d_v,\ D \geq T_u \\ \left\lfloor \frac{x_{i-1,j}+x_{i+1,j}}{2} \right\rfloor, & \text{if } d_h > d_v,\ D \geq T_u \\ \left\lfloor \frac{x_{i-1,j}+x_{i+1,j}+x_{i,j-1}+x_{i,j+1}}{4}, \right\rfloor & \text{otherwise} \end{cases} \quad (4)$$

If D is smaller than the threshold T_u, the neighbors have approximately the same values and the average of all four pixels is expected to produce very good results. The results of equation (4) depends on T_u. The value of T_u can be selected in function of the image at hand. For instance, its value can be easily pinpointed by varying the threshold and by computing the prediction errors for each pixel and selecting the threshold that provides the most prediction errors equal to zero. In our tests, we found $T_u \in [1, 20]$.

Several measures have been tested for D. Equation (3) appeared to provide the best results.

Before going any further a comment should be made. The basic idea behind the predictor is that if, for instance, the horizontal neighbors have values which are closer to each other than the vertical neighbors, then this indicates that there could be a strong connection between these pixels. Using only the horizontal pixels can produce better results than using the entire four pixel context, provided that the pixels are not in an aria with uniform values (indicated by $D < T_u$). Essentially, our approach fits a horizontal or a vertical line over the rhombus neighborhood. The idea of detecting a line and further estimating the pixel according to the existence and the location of such a line is also used by MED and GAP predictors [3, 9].

3. REVERSIBLE WATERMARKING SCHEME

3.1. Marking

The estimation error, p_e, for each pixel $x_{i,j}$ is computed as:

$$p_e = x_{i,j} - \hat{x}_{i,j} \quad (5)$$

A data bit, b, is embedded into $x_{i,j}$ by expanding its prediction error:

$$x'_{i,j} = x_{i,j} + p_e + b \quad (6)$$

where $x'_{i,j}$ represents the new value of $x_{i,j}$ after embeding. The new prediction error for the embedded pixel is:

$$p'_e = x'_{i,j} - \hat{x}_{i,j} = x_{i,j} + p_e + b - \hat{x}_{i,j} = 2p_e + b \quad (7)$$

The pixel values are limited to the spatial domain $[0, L-1]$, where for k bits images, $L = 2^k$. For 8 bit graylevel images, $L = 256$. In order to prevent underflow or overflow errors, the histogram shifting with flag bits DE scheme proposed by Thodi and Rodriguez is used [4].

The image is split into two regions: S (selected) and R (reserved). Region S contains the pixels selected for embedding, while R is a reserved region used to transmit the information needed at the decoding stage as the threshold values and the flag bits. The embedding into the R region proceeds by first recording in a bitstream the least significant bits (LSBs) of the pixels belonging to R and then, by replacing them with the needed information. The substituted bitstream is further embedded into S as auxiliary data. Thus, at detection, the original pixels of R can be recovered.

The watermark bitstream contains the payload (the data that we want to hide in the image), a end-of-payload indicator and the auxiliary data.

Expanding a large estimation error increases the distortion introduced by the watermarking. In order to avoid large distortions, a threshold control scheme is used. Data is embedded into a pixel only if its estimation error is smaller than a threshold T. The watermark size determines the value of T. In order to find T, we start with $T = 1$ and we simulate the embedding process. If there are not enough embeddable pixels, we increment the threshold value until the watermark can be fully embedded. The pixels where $p_e < T$ are collected into the set E (embedded) group.

The pixels with $p_e \geq T$ form the not embedded group, N. These pixels are modified such that, at detection, they provide a larger error than the embedded pixels. In order to avoid the overlap between the estimation errors, the pixels in N are modified as follows:

$$x'_{i,j} = \begin{cases} x_{i,j} + T & \text{if } p_e \geq 0, \\ x_{i,j} - T + 1 & \text{if } p_e < 0. \end{cases} \tag{8}$$

The procedure described by (8) is known as histogram shifting. The pixels of N which cannot be modified by (8) because of underflow or overflow are left unchanged. They form the group U (unmodified). The flag bits are used to distinguish between the pixels that are part of group U and those that belong to E or N. All the pixels of U have values in the range of $[0, T-1) \cup (L-1-T, L-1]$. When the current pixel belongs to U, the "0" flag is inserted in the next embeddable location instead of the watermark bit, the "1" flag is used for every pixel that belongs to $E \cup N$ and has its value in $[0, T-1) \cup (L-1-T, L-1]$. If we have two or more consecutive pixels that can't be altered, the flag bit corresponding to the last pixel is inserted first.

3.2. Two stage marking

The use of raster-scan order for data embedding does not fit with the rhombus pattern used for pixel estimation. After the embedding of pixel $x_{i,j}$, the embedded value $x'_{i,j}$ take part in the estimation of other pixels. To mitigate this effect, the watermarking scheme divides the pixels in the S region into two sets, similar to [10]: the "cross" set and the "dot" set.

All the pixels from the first row of S that belong to an even column are distributed to the cross set, the remaining pixels from this row (that are part of odd columns) belong to the dot set. For the next row, the pixels belonging to an even column are in the dot set, and those of odd columns are in cross set. For the remaining rows, we similarly alternate the distribution of pixels according to their column. The dot and cross sets form a chess board pattern. It can be observed that the neighbors used to estimate the pixels in the cross set belong to the dot set and conversely, the neighbors of the pixels in the dot set are part of the cross set.

The pixels in the cross set are split into the E, N and U groups. The watermark is inserted in the pixels that belong to E using (6). The pixels in N are shifted using (8). After the embedding of all the pixels of the cross set, we repeat the procedure for the pixels of the dot set.

The estimation of the pixels of the cross set is not influenced by the embedding process. By the contrary, the rhombus neighbors of the pixels in the dot set are altered by watermarking the cross set. The estimation of the dot set pixels is not as accurate as the one of the cross set. Nevertheless the results are better than using the raster-scan order.

3.3. Decoding

If the embedding is performed in raster scan ordering, row by row, from the upper left to the lower right pixels, the decoding proceeds in reverse ordering, from bottom to top. The first pixel restored to its original value is the last embedded one. The decoder can distinguish between the pixels that belong to E, N or U based on their prediction error, their value and the flag bits:

- If $x'_{i,j} \in [T-1, L-1-T]$:
  - $x'_{i,j} \in E$ if $p'_e \in [-2T+1, 2T]$;
  - $x'_{i,j} \in N$ if $p'_e \notin [-2T+1, 2T]$;
- If $(x'_{i,j} \in [0, T-1) \cup (L-1-T, L-1]$:
  - $x'_{i,j} \in U$ if the last bit decoded is "0";
  - $x'_{i,j} \in E$ if $p'_e \in [-2T+1, 2T]$;
  - $x'_{i,j} \in N$ if $p'_e \notin [-2T+1, 2T]$.

If the pixel is in group E, the embedded bit is extracted from the estimation error:

$$b = p'_e \bmod 2 \tag{9}$$

and then the pixel is restored to its original values:

$$x_{i,j} = x'_{i,j} - \lfloor p'_e \rfloor - b \tag{10}$$

If the pixels is from group N it was not used for embedding, but it was modified. The restored pixel is:

$$xi, j = \begin{cases} x'_{i,j} - T & \text{if } p_e \geq 0, \\ x'_{i,j} + T - 1 & \text{if } p_e < 0. \end{cases} \tag{11}$$

If the pixel is from U, it remains unchanged. We remind that the pixels of U were not altered by the embedding process.

The decoding first extracts the watermark from the dot set. After the pixels of the dot set are restored, the cross set pixels are decoded as well. After the embedded information is extracted from all the pixels, the bitstream is reversed to recover the watermark (the decoding was done in reverse order of the embedding). The payload is separated from the auxiliary data and then the auxiliary data is used to restore the pixels in R to their original values.

4. EXPERIMENTAL RESULTS

We further consider five standard graylevel test images of sizes 512×512, namely *Lena, Mandrill, Jetplane, Barbara, Tiffany* (Fig. 1). These images are extensively used in the reversible watermarking literature.

The proposed adaptive interpolation scheme is first compared with two well-known prediction schemes (MED used

Fig. 1. The test images

by Thodi [4], and GAP used in the CALIC algorithm [9]) and with the interpolation of [10]. The histograms of the estimation errors for the test images *Jetplane* and *Barbara* are presented in Fig. 2. Each histogram is centered on zero. As it appears from Fig. 2, the proposed method provides the best results, i.e., the sharpest estimation error histogram. The simple average of [10] gives better results than the GAP on *Jetplane* and similar results on *Barbara*. The MED predictor appears to give the poorest results. The results for the other three images are rather similar with the ones presented in Fig. 2.

The reversible watermarking scheme of Section 3 using the proposed interpolation is further tested on the images of Fig. 1. The results are compared with the ones obtained by the standard prediction error expansion with histogram shifting and flag bits of [4] for MED and GAP predictors and with the results obtained by the simple average on rhombus [10].

The proposed scheme clearly outperforms the MED predictor on all test images. The average improvement in PSNR considering all five images is of 1.8 dB. The highest average improvement for a single image is of 2.1 dB on *Barbara*. The lowest average improvement for a single image is of 1 dB, obtained on *Mandrill*. We have rather similar results when comparing with the ones obtained for the GAP predictor: an average improvement of 1.4 dB over GAP, the largest difference between the two schemes on a single image was obtained also on *Barbara* (an average of 1.8 dB) and the smallest was on *Mandrill* (an average of 0.7 dB) .

The proposed scheme outperforms the normal interpolation scheme on three out of the five test images. On the other two, *Lena* and *Mandrill*, the results are similar. The average improvement for the five images is 0.28 dB. The best results, an improvement of 0.6 dB, were obtained on *Barbara*.

The improvement provided by the proposed scheme is due to the interpolation of edge pixels. Compared with the simple average, the proposed scheme provides a better estimation of the pixels located on vertical and horizontal fragments of edges. Obviously, such vertical/horizontal fragments of edges

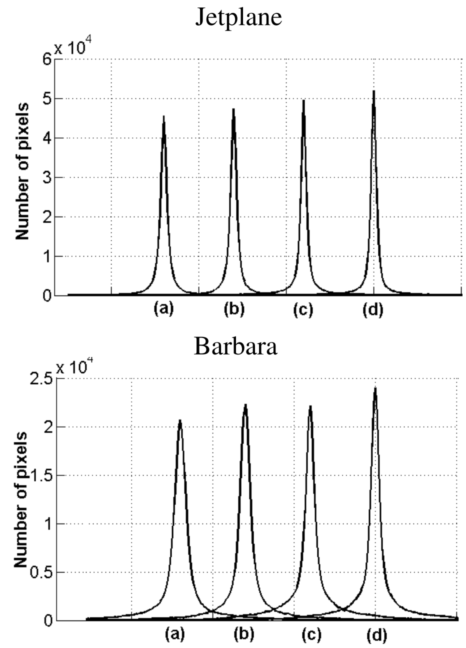


Fig. 2. Estimation error histograms for: MED (a), GAP (b), normal interpolation (c) and the proposed scheme (d).

can be also found in certain textures. For uniform regions, pure diagonal edges (45 or 135 degrees) and fine grain textures the results are similar.

5. CONCLUSIONS

An adaptive interpolation scheme on rhombus neighborhood was proposed. The interpolated value is computed not only as the average of the four pixels as for the classical interpolation, but also as the average of the horizontal or vertical pair of pixels. The proposed scheme appears to provide lower estimation error than the classical interpolation on rhombus or the ones provided by nonlinear predictors as MED and GAP.

The proposed scheme is of interest in reversible watermarking. The experimental results obtained so far are promising. The results on five classical test images show that the proposed scheme outperforms the prediction provided by MED, GAP and normal interpolation with an average value of 1.8 dB, 1.4 dB and 0.28 dB, respectively.

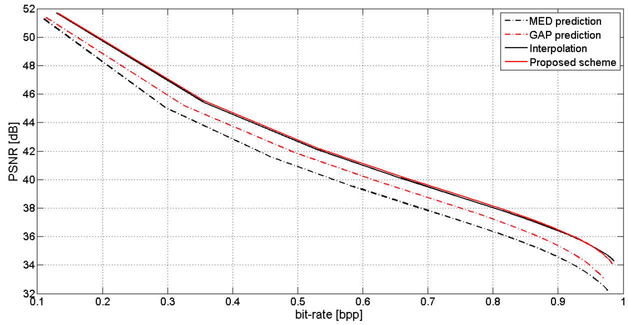
6. REFERENCES

[1] J. Tian, "Reversible Data Embedding Using a Difference Expansion", *IEEE Trans. on Circuits and Systems for Videotechnology*, vol. 13, no. 8, pp. 890–896, 2003.

[2] D. M. Thodi, J. J. Rodriguez, "Prediction-error based reversible watermarking", *ICIP'04*, pp. 1549-1552, 2004.

[3] M. Weinberger, G. Seroussi, G. Sapiro, "The LOCO-I lossless image compression algorithm: Principles and

Lena



standardization into JPEG-LS", *IEEE Trans. on Image Processing*, vol. 9, no. 8, pp. 1309–1324, 2000.

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

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

[6] M Fallahpour, "Reversible image data hiding based on gradient adjusted prediction", *IEICE Electronics Express*, vol. 5, no. 20, pp. 870-876, 2008.

[7] M. Chen, Z. Chen, X. Zeng, Z. Xiong, "Reversible Data Hiding Using Additive Prediction-Error Expansion", *Proc. of the 11th ACM Workshop on Multimedia and Security*, pp. 19-24, 2009.

[8] D. Coltuc, "Improved Embedding for Prediction Based Reversible Watermarking", *IEEE Trans. on Information Forensics and Security*, vol. 6, no. 3, p. 873-882, 2011.

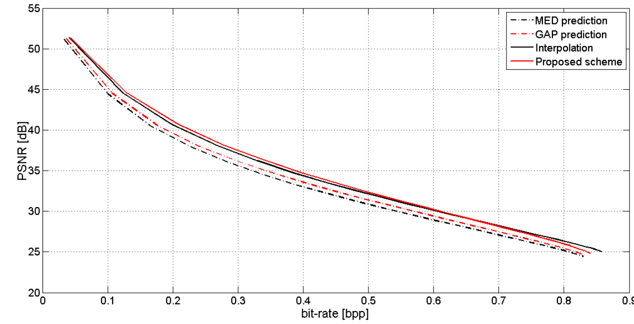
[9] X. Wu and N. Memon, "Context-based, adaptive, lossless image coding", *IEEE Trans. on Communications*, Vol. 45, No. 4, pp. 437 ? 444, 1997.

[10] V. Sachnev, H. J. Kim, J. Nam, S. Suresh, Y. Q. Shi "Reversible Watermarking Algorithm Using Sorting and Prediction", *IEEE Trans. on Circuits and Systems for Videotechnology*, vol. 19, no. 7, pp. 989-999, 2009.

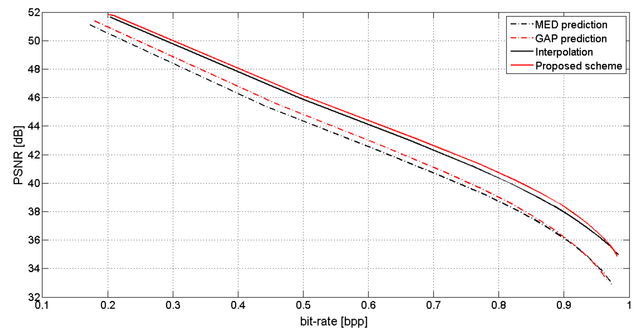
[11] M. Chen, Z. Chen, X. Zeng, Z. Xiong, "Reversible Image Watermarking Based on Full Context Prediction", *Proceedings of the International Conference on Image Processing, ICIP'09*, pp. 4253-4256, 2009.

[12] L. Luo, Z. Chen, M. Chen, X. Zeng, Z. Xiong, "Reversible Image Watermarking Using Interpolation Technique", *IEEE Trans. on Information, Forensics and Security*, vol. 5, no. 1, pp. 187-193, 2010.

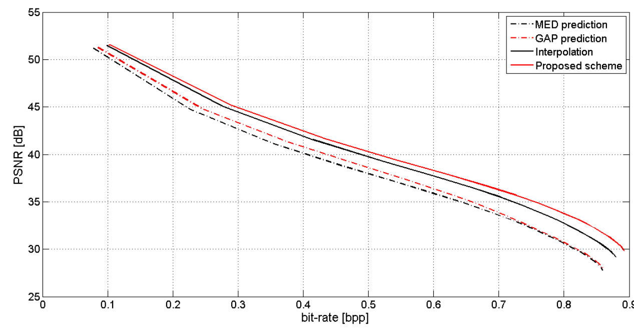
Mandrill



Jetplane



Barbara



Tiffany

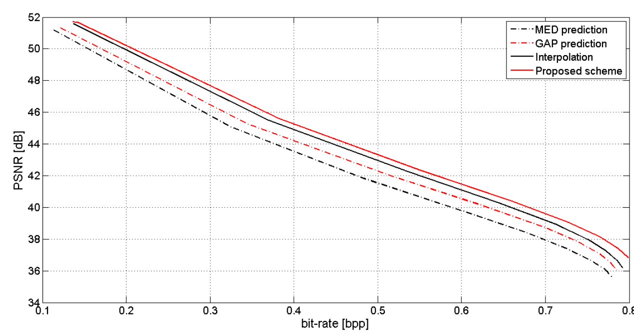


Fig. 3. Experimental reversible watermarking results for schemes using the proposed adaptive interpolation scheme, the simple average, MED and GAP.