

PREDICTION-ERROR BASED REVERSIBLE WATERMARKING

Diljith M. Thodi and Jeffrey J. Rodríguez

Dept. of Electrical & Computer Engineering
The University of Arizona
Tucson, Arizona 85721
{diljith, rodriguez}@ece.arizona.edu

ABSTRACT

Reversible watermarking has become a highly desirable subset of fragile watermarking for sensitive digital imagery in application domains such as military and medical because of the ability to embed data with zero loss of host information. This reversibility enables the recovery of the original host content upon verification of the authenticity of the received content. We propose a new reversible watermarking algorithm. The algorithm exploits the correlation inherent among the neighboring pixels in an image region using a predictor. The prediction-error at each location is calculated and, depending on the amount of information to be embedded, locations are selected for embedding. Data embedding is done by expanding the prediction-error values. A compressed location map of the embedded locations is also embedded along with the information bits. Our algorithm exploits the redundancy in the image to achieve very high data embedding rates while keeping the resulting distortion low.

1. INTRODUCTION

The sophisticated processing capabilities of modern digital multimedia systems have made it very easy to alter the content in the digital medium. This has raised questions regarding the trustworthiness of digital content. Fragile watermarking has been proposed as a solution to this problem. In fragile watermarking, a watermark—usually a signature of the medium—is embedded into the medium in such a manner that any alteration of the medium would corrupt the embedded signature. The distortion introduced in the embedding process is called embedding distortion. Although most applications constrain the embedding distortion to a small value, the original contents are usually irreversibly changed. However, certain application domains, such as military and medical, are sensitive to the embedding distortion and prohibit permanent loss of signal fidelity. This highlights the need for lossless or reversible watermarking. Reversible watermarking restores the original (unwatermarked) medium upon verification of authenticity.

Several high-capacity reversible watermarking schemes [1][3][4][5] have been developed during the past several years. Tian's scheme [1] achieves very high data embedding rates, approaching one bit per a pair of pixels. His scheme involves embedding by expanding the difference values of a pair of pixels. Alattar [5] extended the scheme to embedding two bits in differences among a triplet of pixels. The pairing is done on adjacent pixels. The redundancy in an image results in small magnitudes for the difference values of the paired pixels. Embedding a bit in the difference value cause the difference value to be doubled. But since the difference values are small, the embedding distortion remains small.

In this paper we propose a new reversible data embedding algorithm for gray-scale images. Our algorithm embeds the data by expanding the prediction error values. A predictor exploits the redundancy in an image better than the difference expansion transform used in Tian's algorithm. The embedding rate of our scheme approaches one bit per pixel. Multiple embedding can be used if a higher rate is desired. In the next section we explain the how information embedding and extraction is done by prediction-error expansion. In Sect. 3, we describe the procedure for embedding an information bitstream in an image. The decoder is described in Sect. 4. The results of our algorithm are presented in Sect. 5 and the summary of conclusions presented in Sect. 6.

2. PREDICTION-ERROR EXPANSION

Consider a pixel in a gray-scale image with intensity x into which a bit i is to be embedded. A predictor operates on the neighbourhood of the considered pixel and predicts its intensity, \hat{x} . The prediction-error, p_e , is

$$p_e = x - \hat{x} \quad (1)$$

The bit i is embedded by expanding the prediction-error. The binary representation of p_e is shifted left by one bit to create a vacant LSB, into which the bit, i is embedded. Let

the binary representation of p_e be

$$p_e = b_{l-1}b_{l-2}\dots b_0$$

where b_0 is the LSB and l is the bit length of p_e . Expanding the prediction-error and embedding the bit i changes the prediction-error, p_e , to p'_e , where

$$p'_e = b_{l-1}b_{l-2}\dots b_0i = 2p_e + i \quad (2)$$

The modified prediction-error, p'_e , and the predicted intensity, \hat{x} , are combined to obtain the embedded pixel intensity, x' , where

$$x' = \hat{x} + p'_e = x - p_e + p'_e = x + p_e + i \quad (3)$$

The watermark detector extracts the embedded bit, i , and restores the original pixel intensity, x . The intensity of the pixel is predicted using the same predictor as employed by the embedder. The predicted intensity will be \hat{x} (if the neighbourhood has not been altered). The prediction-error at the decoder is $x' - \hat{x} = p'_e$. The embedded bit is the LSB of p'_e .

$$i = p'_e - 2\lfloor p'_e/2 \rfloor \quad (4)$$

The true pixel intensity is restored after calculating the original prediction-error as

$$p_e = \lfloor p'_e/2 \rfloor \quad (5)$$

$$x = x' - p_e - i \quad (6)$$

Since the grayscale values are restricted to the range [0,255], brute force expansion embedding of some pixels might result in overflow or underflow. This problem is overcome by not considering for prediction-error expansion, any pixel which could potentially cause an overflow/underflow. The locations that can undergo prediction-error expansion without an overflow or an underflow are called *expandable locations*. For a location to be an expandable location, the resultant pixel intensity as given by eq. (2) should be a legal grayscale value, i.e. it must satisfy

$$0 \leq x' \leq 255$$

$$0 \leq x + p_e + i \leq 255 \text{ for } i = 0, 1$$

$$x + p_e \begin{cases} \leq 254 & \text{if } p_e \geq 0 \\ \geq 0 & \text{if } p_e < 0 \end{cases} \quad (7)$$

Any location that satisfies (6) is an expandable location.

Prediction Algorithm: We have used a low complexity predictor with an inherent primitive edge detector. The predictor operates on a three neighbour context. The prediction algorithm and the contexts are as follows

$$\hat{x} = \begin{cases} \max(a, b) & \text{if } c \leq \min(a, b) \\ \min(a, b) & \text{if } c \geq \max(a, b) \\ a + b - c & \text{otherwise} \end{cases} \quad \begin{array}{|c|c|} \hline x & b \\ \hline a & c \\ \hline \end{array}$$

where x is the current pixel, \hat{x} is the predicted value and a, b, c is the context as shown.

The above predictor has been employed in many image compression applications [2], and it has been modified here to suit an anti-causal context. The pixels at the last column and last row have only one pixel in their context. Their prediction is set to the intensity of the single pixel in the context. The pixel at the last column of the last row is not considered for embedding.

3. ENCODER

The encoding of the watermark in the image is done in two stages. Two different embedding schemes are used for the two stages. The first stage embeds the information bits and the auxiliary data using prediction-error expansion technique. In the second stage, a compressed map of the locations embedded in the first stage, is embedded by the technique of LSB embedding.

Consider an $r \times c$ host gray-scale image. Using the prediction algorithm described in the previous section, the prediction error of every location except the last column of the last row is calculated. From the prediction-error and the pixel intensity the expandable locations are identified using eq. (7).

3.1. The First Stage

The first step in this stage is to select a subset from the set of expandable locations for expansion embedding. The next step is to create a location map of the selected locations and to losslessly compress it. The information bitstream and the auxiliary data are then embedded in the selected locations. The compressed location map is embedded in the second stage.

Selection of Locations: Several strategies have been proposed in [1] for the selection of locations to embed from the set of all expandable locations. Even though the strategies suggested in [1] are for embedding by difference expansion, they are suitable for embedding by prediction-error expansion. In this paper we have selected the locations by setting a threshold T on the magnitude of the prediction-errors, and all expandable locations with prediction-error magnitudes less than the threshold are selected for embedding. We denote ξ be the set of all locations that have been selected for expansion embedding and L_ξ as the cardinality of ξ .

Location Map: A two-dimensional ($r \times c$) bitmap (M) is created by assigning a '1' to all locations belonging to ξ and a '0' to all the other locations. The bitmap (M) is compressed using a lossless compression algorithm. We have used JBIG2, the international standard for compression of

bi-level images. See [6] for a detailed discussion of JBIG2. A unique end-of-map indicator (*EOM*) is at the end of the compressed map C . Let L_C be the length of C .

Auxillary Data and Embedding: The maximum information embedding capacity of our scheme is $L_\xi - L_C$. Therefore, the selection of the threshold T should take into consideration the size of the compressed bitstream and the number of bits to be embedded.

Let D be the array in which the information bits to be embedded are stored. The encoder sequences through the image in raster scan order. The locations which do not belong to ξ are left unaltered while locations belonging to ξ are expansion embedded with the corresponding bit from D . The first bit of D is embedded in the first location that belongs to ξ , the second bit in the second location that belongs to ξ , and so on.

The auxillary data is generated from the LSBs of the first L_C locations. For the first L_C locations, after each location has been processed, the LSB of the pixel intensity is appended to D . For an embedded location, the appended bit is the LSB of the modified pixel intensity, whereas for the unaltered location it is the actual LSB. Thus, as the embedding proceeds, the array D grows with the auxillary data. After the whole image has been processed by the first stage, L_ξ bits have been embedded, of which the first $L_\xi - L_C$ bits are the information bits, and the last L_C bits are the auxillary data.

3.2. The Second Stage

The compressed location map C created at the first stage is embedded at the second stage. The embedding is done by replacing the LSBs of the first L_C locations with the compressed bitstream C . The LSBs that have been replaced by this process are the auxillary data that was embedded in the first stage. The resultant image at the end of the LSB embedding is the watermarked image W .

4. DECODER

Let the watermarked image W be the received image at the decoder. The second stage is decoded first to obtain the compressed location map C . Decompressing C gives the location map (M). The location map (M) is used to extract the information bits.

Extracting the Location Map: The LSBs of the received image are collected in raster scan order. The sequence is searched for the end-of-map indicator (*EOM*). The bits from the start until the *EOM* comprise the compressed bitstream C , which is decompressed to obtain the location map (M).

Extracting the Information Bits: The decoding of the information bits is done in reverse raster scan order. The

embedded locations (ξ) are determined from the location map. The decoder starts at the pixel at the next-to-last column of the last row. The pixel at the last column of the last row was not considered for embedding at the encoder. The decoder determines if the current location belongs to ξ from the location map. For a location that belongs to ξ , the prediction error is calculated. From the calculated prediction error, the embedded bit is extracted and the original pixel intensity restored as explained in Sect. 2. The extraction of the embedded bit and restoration of the original pixel intensity are done in tandem. It is important to restore the original pixel intensity before proceeding to decode the next location because the current location could be a context to the next location to be decoded, in which case the original pixel value at the current location is the true context of the next location to be embedded.

Each decoded bit is appended to the previously decoded bit. Thus the decoded bitstream is reversed with respect to the encoded bitstream. The last embedded bit is the first decoded bit. The first several bits of the decoded bitstream are the auxillary data bits in the reverse order. The number of auxillary bits embedded is determined from the compressed location map and is equal to the number of bits in the compressed location map (C). As explained in the previous section, the auxillary bits are the LSBs of the pixel intensities of the first L_C locations that had been overwritten in the second stage of embedding. Therefore, before extracting a bit from any of the first L_C locations, we have to restore their LSBs. To ensure this, as soon as any auxillary bit is extracted, the LSB of the corresponding location is overwritten with it. The first extracted bit replaces the LSB of the L_C th location, the second decoded bit is the LSB of the $(L_C - 1)$ st location and so on.

After the entire image has been decoded we obtain the decoded bitstream and the original image (I). The decoded bitstream is the reversed version of the information bits and the auxillary bits. The decoded bitstream is reversed and the auxillary bits removed by dropping the last L_C bits. The remaining bits are the information bits.

5. RESULTS

The algorithm described in the previous sections was implemented and tested on various standard test images. We also implemented Tian's algorithm in order to compare our results with his results. Multiple embedding was done while testing Tian's algorithm in order to achieve rates above 0.5 bpp. For multiple embedding, the pairing direction was horizontal for the first embedding and vertical for the second embedding. The watermark signal was generated by rounding pseudo-randomly generated numbers, uniformly distributed between 0 and 1. The location map was losslessly compressed by the JBIG2 bi-level compression en-

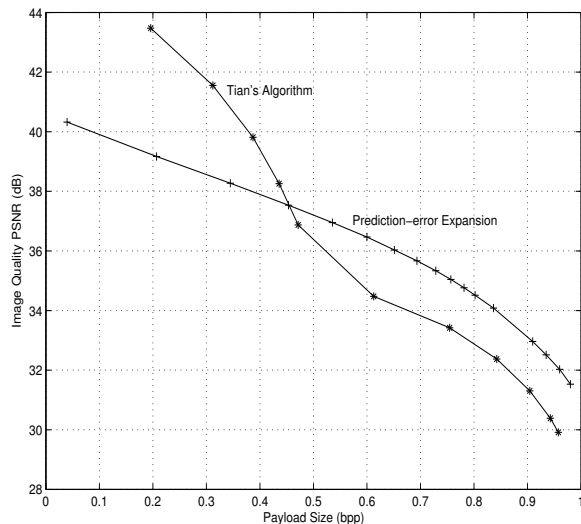


Fig. 1. Embedding capacity vs. image quality for “Lenna”

gine.

We present the results for 512×512 gray-scale Lenna and Barbara images. The results are shown in Fig. 1 and Fig. 2 respectively.

Comparing our results with Tian’s difference expansion algorithm [1] for the Lenna image (Fig. 1) we find that Tian’s algorithm outperforms our algorithm at low embedding rates, but the performance catches up at around 0.5 bpp, and our algorithm outperforms Tian’s at high embedding rates. At the embedding rate of 0.6 bpp, the PSNR of the image watermarked by our algorithm is better than the image watermarked by Tian’s algorithm by about 0.8dB, and increases to about 2dB at embedding rates approaching 1 bpp. Compared with the modified difference expansion algorithm [5], our algorithm performs much better at high embedding rates, and as good as theirs at low embedding rates. Our algorithm performs better than the G-LSB [4] scheme at embedding rates higher than 0.25 bpp. At an embedding rate of 0.5bpp, the image quality of the image watermarked by prediction-error expansion is better than the image watermarked by G-LSB embedding by about 2dB. The same trend is observed for the Barbara image (Fig. 2), where our algorithm outperforms Tian’s algorithm as the embedding rate is increased.

6. CONCLUSION

We have presented a high-capacity reversible data hiding algorithm based on prediction-error expansion. The algorithm achieves embedding rates close to one bit per pixel without multiple embedding. The quality of the watermarked image is better than that of competing schemes for moderate-to-high embedding rates, whereas Tian’s scheme is better

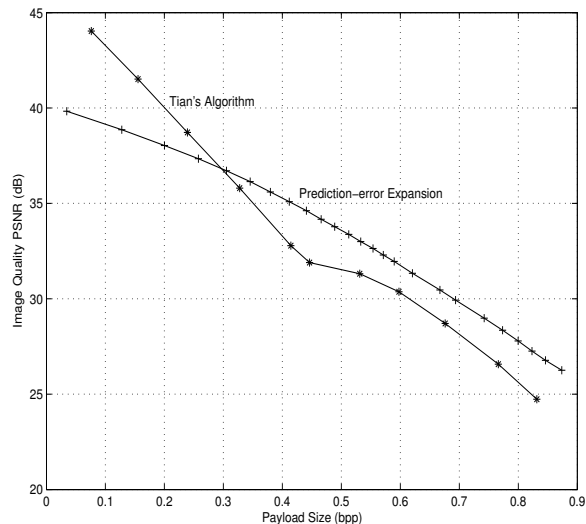


Fig. 2. Embedding capacity vs. image quality for “Barbara”

at low embedding rates. The decoder can extract the embedded watermark and restore the original unwatermarked image, pixel for pixel.

7. REFERENCES

- [1] J. Tian, “Reversible Data Embedding Using a Difference Expansion,” *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 13, no. 8, Aug. 2003, pp. 890-896.
- [2] M. Weinberger, G. Seroussi, and G. Sapiro, “LOCO-I: A Low Complexity, Context-Based, Lossless Image Compression Algorithm,” in *Proc. of the IEEE Data Compression Conf.* 1996, pp. 140-149.
- [3] M. Goljan, J. Fridrich, and R. Du, “Distortion-Free Data Embedding for Images” in *4th Information Hiding Workshop*, 2001, pp. 27-41.
- [4] M.U. Celik, G. Sharma, A.M. Tekalp, and E. Saber, “Reversible Data Hiding,” in *Proc. of the Intl. Conf. on Image Processing*, 2002, vol. II, pp. 157-160.
- [5] A.M. Alattar, “Reversible Watermarking Using Difference Expansion of Triplets,” in *Proc of the Intl. Conf. on Image Processing*, 2003, vol. I, pp. 501-504.
- [6] P.G. Howard, F. Kossentini, B.Martins, S. Forchhammer, and W.J. Rucklidge, “The Emerging JBIG2 Standard,” in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 8, no. 7, pp. 838-848, 1998.