タイトル Title	Reversible Watermark with Large Capacity Using the Predictive Coding
著者 Author(s)	Kuribayashi, Minoru / Tanaka, Hatsukazu / Morii, Masakatu
掲載誌・巻号・ページ Citation	Lecture Notes in Computer Science,3783 - Information and Communications Security:441-453
刊行日 Issue date	2005
資源タイプ Resource Type	Journal Article / 学術雑誌論文
版区分 Resource Version	author
権利 Rights	
DOI	10.1007/11602897_37
JaLCDOI	
URL	http://www.lib.kobe-u.ac.jp/handle_kernel/90000259

Reversible Watermark with Large Capacity Using the Predictive Coding

Minoru Kuribayashi[1], Masakatu Morii[1], and Hatsukazu Tanaka[2]

[1] Department of Electrical and Electronics Engineering
Faculty of Engineering, Kobe University
1-1, Rokkodai-cho, Nada-ku, Kobe, 657-8501 Japan.
{kminoru, mmorii}@kobe-u.ac.jp
[2] Kobe Institute of Computing,
2-2-7, Kanou-cho, Chuo-ku, Kobe, 650-0001 Japan.
tanaka@kic.ac.jp

Abstract. A reversible watermarking algorithm with large capacity has been developed by applying the difference expansion of a generalized integer transform. In this algorithm, a watermark signal is inserted in the LSB of the difference values among pixels. In this paper, we apply the prediction errors calculated by a predictor in JPEG-LS for embedding a watermark signal, which contributes to increase the amount of embedded information with less degradation. As one of the drawback discovered in the above conventional method is a large size of the embedded location map introduced to make it reversible, we decrease the large size of the location map by vectorization, and then modify the composition of the map using the local characteristic in order to enhance the performance of JBIG2.

1 Introduction

In a data-hiding technique[1], the embedding causes irreversible degradation to an image. Although the degradation is perceptually slight, it may not be accepted to some applications such as medical or military images. For the countermeasure, lossless data-hiding techniques, which is called reversible(invertible) watermark, have been developed. The reversible watermark techniques might be classified into two methods. One compresses features of an image and transmits the compressed bit-stream as a part of the embedding information. At the decoding, the embedded information including the compressed bit-stream is extracted, and the original image is restored by replacing the modified features with the decompressed original features. In [2], each pixel is first quantized by a quantization step size L, and appends the embedding information to the compressed quantization noise. Then, the information is added to the quantized image. The scheme tends to be superior when the watermarked image keeps high quality. However, for a large amount of information, the capacity would be inadequate.

The other method uses reversible integer transforms to the spatial domain of an image, and embeds a watermark information in the transformed signal values.

Tian[3] presented a difference expansion transform of pair of pixels, which is haar wavelet transform, to devise a reversible watermark with a large capacity and low degradation. His algorithm divides an image into pairs of pixels, then it inserted one bit into the difference of the pixels of each pair from those pairs that are not expected to causes an overflow or underflow. So as to recover the original image, a location map that indicates the modified pairs is embedded as a part of the embedding information after compression. Heijmans et al.[4] improves the capacity for high quality images by performing low pass filtering to an image to predict a location map. Alattar[5] extends Tian's method to the difference expansion of a vector of several pixels to achieve larger capacity. The algorithm can insert several bits in the difference expansion of each vector of adjacent pixels. As each element in the location map indicates the embedding position of the corresponding pair/vector, the size of the map depends on the number of the pairs/vectors. Even if the location map is compressed, the size is still large, and hence the capacity is restricted in the above schemes. In addition, the positions where a watermark signal is embedded is mainly at a flat region of an image for the property of the algorithm, which causes perceptual degradation.

In this paper, we propose a new technique to embed a large amount of information with less degradation. Our main idea is to embed a watermark in prediction errors calculated by a well-designed predictor in JPEG-LS[6]. Since the average value of the prediction errors is small, our scheme can spread out the positions where a watermark signal can be embedded without causing both overflow and underflow. Such positions are not restricted only in a flat region, but also in a noisy one, and hence the distortions caused by our embedding may be less perceived. One of the drawbacks in the conventional schemes[3]-[5] is the size of the location map even if it is compressed. First, we reduce the size of the map itself by making a vector which treats several pixels as one cluster. And we study the characteristic of the local conditions of pixels, and find that at some pixels the location map is not necessary if a certain condition is satisfied. Such characteristic is exploited to enhance the performance of the compression algorithm. As the results, our scheme achieves very large capacity with less distortions. Since the operation is reversible, multiple embedding is possible and hence the capacity may be increased further.

2 Preliminary

2.1 Modeling in JPEG-LS

JPEG-LS[6] is the algorithm at the core of the new ISO/ITU standard for lossless and near-lossless compression. The algorithm attains significantly better compression ratios, similar or superior to those obtained with state-of-the art schemes based on arithmetic coding, but at a fraction of the complexity.

Lossless image compression schemes, in general, consist of two distinct and independent components: *modeling* and *coding*. The modeling part is formulated as an inductive inference problem, in which an image is observed sample by sample in some predefined order. In JPEG-LS, the sample points are defined as

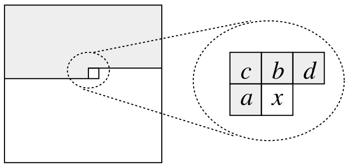


Fig. 1. The sample points of the fixed predictors.

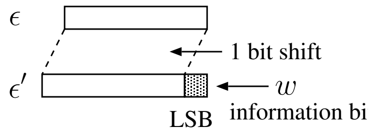


Fig. 2. Expanding of the prediction error to embed a watermark.

a, b, c, and d depicted in Fig.1. And the current sample x is predicted as $\hat{x}$ using the four samples. In the coding part, the predictive error ϵ

$$\epsilon = x - \hat{x} \tag{1}$$

is encoded with an extended family of Golomb-type code[8] which is adaptive symbol-by-symbol coding at very low complexity.

The distribution of the prediction error ϵ is well modeled by a *two-sided geometric distribution* (TSGD)[7] centered at zero. And ϵ is also extremely sensitive for the changes in the previously occurred samples. Note that the change occurred in one sample is propagating to the following every prediction error.

2.2 Expanding

The main idea of our scheme is to utilize the prediction errors calculated by the predictor in JPEG-LS to embed information bits in an image. Since the prediction errors follow TSGD centered at zero, the average value might be small. Based on the characteristic, a lossless embedding could obtain a better performance than that of haar wavelet transform[3][4] and generalized integer transform[5]. In JPEG-LS, only the prediction errors are preserved assisted by entropy coding[8]. Therefore, if the original prediction error can be recovered from a watermarked image, the embedding operation is reversible.

When an information bit w is embedded, a prediction error ϵ is expanded so as to insert w without loss of original information of ϵ. The basic embedding operation is to double ϵ, and to put w on its LSB(See Fig.2).

$$\epsilon' = 2\epsilon + w, \tag{2}$$

It is the same operation as that of the conventional schemes[3]-[5]. Those schemes calculate the difference among neighboring pixels, and embed a watermark information bit if its difference is less than a certain threshold. However, they have a trouble to embed a watermark in a noisy region because most of the differences in such a region are large. Such trouble is also occurred at the embedding in prediction errors, but it may not so serious compared with the conventional ones because the predictor may be able to output relatively small prediction errors for noisy regions. In order to apply the basic embedding operation for the prediction errors, several parameters must be modified carefully. Thodi and Rodrfguez[9] applied a predicted value, but the predictor is not a well-designed one. Although their results show that the capacity is large, their method is not reversible because a location map is not considered.

2.3 Definitions

On the expansion of the differences in the conventional schemes, overflow and underflow of pixel values are avoided by carefully selecting the target pair/vector of pixels. The selection is based on the definitions of *expandable* and *changeable*, and it is performed in the transformed domain. Since the definitions and classifications are essentially same as that of our scheme, we describe the detail of our definitions. The main concern of our scheme is the embedding capacity for each pixel/vector. Tian's algorithm is capable of embedding as high as 1/2 bits/pixel because one bit is embedded in the difference of pair of pixels, and Alattar's one is as high as $(n-1)/n$ bits/pixel for each difference of n pixels. On the other hand, our scheme can be at most 1 bits/pixel as pixel-wise operation is possible by applying the prediction error instead of differences among pixels.

It seems difficult to modify the prediction error using the previously occurred pixels a, b, c, and d because those pixels are also used for the prediction of other pixels. Instead, we modify the current target pixel x, which is easily calculated from Eq.(1).

$$x_e = x + \epsilon + w \tag{3}$$

Here, it must be considered that the pixel value must be in $[0, 255]$, otherwise it causes overflow(more than 255) or underflow(less than 0). In order to control the embedding operation, the following definition is introduced.

Definition 1. *The pixel x is said to be expandable if, for any $w \in \{0,1\}$, x can be modified to x_e without causing overflow and underflow.*

If a pixel is expandable, it is possible to embed an information bit in the LSB by expanding the prediction error ϵ. However, when one wants to extract an information, it is impossible to find if the pixel was expandable or not before the embedding. By considering the extraction, the prediction errors of the pixels which are not expandable are modified to even number by the following equation.

$$x^\star = \begin{cases} x-1 & if\ \epsilon\ is\ odd \\ x & otherwise. \end{cases} \tag{4}$$

Then the LSB of the prediction error is removed, and the information is embedded with a watermark information in an image so as to be reversible. Here, the overflow and underflow of $x^\star$ must be considered. Therefore, the following definition is also introduced.

Definition 2. *The pixel x is said to be changeable if, for any $w \in \{0,1\}$, x can be modified to x_c without causing overflow and underflow.*

$$x_c = x^\star + w \tag{5}$$

Notice that all expandable pixels are changeable and they are still changeable after the embedding. Based on the characteristic, for both expandable and changeable pixels, information bits are embedded into the LSBs of their prediction errors, which can be extracted from all changeable pixels of the watermarked image. Although the watermark can be extracted, it is impossible to determine if each pixel was expandable or changeable before embedding. Hence, the information about the original conditions of pixels which is called location map is embedded in addition to a watermark. Here, a lossless compression algorithm, such as JBIG2 and an arithmetic compression algorithm, is performed to reduce the size of the map. As the consequence, the embedding information bits are composed of three parts; a compressed location map, LSB of ϵ of changeable pixel, and a watermark information.

In general, there is a trade-off between the distortions and the capacity in watermarking technique, and it is desirable that the trade-off is controlled for applications. It is achieved by introducing a threshold T for the determination of expandable or not. If the absolute value of a prediction error is less than T and Definition 1 is satisfied, the pixel is regarded as expandable. Since the changes caused by the operation at expandable pixels are restricted less than T, the degradation of the quality is controlled.

Each pixel can be classified into three groups according to the Definition 1 and 2. The first group S_1 contains all expandable pixels whose prediction errors less than a predefined threshold T. The second group S_2 contains all changeable pixels that are not in S_1. The third group S_3 contains the rest of the pixels which implies not changeable. Also, let S_4 denote all changeable pixels ($S_4 = S1 \cup S_2$).

3 Proposed Reversible Watermarking Algorithm

In this section, we propose a new reversible watermark scheme using the predictive coding technique in JPEG-LS. A basic algorithm of the reversible operation for the embedding is shown.

3.1 Embedding a Reversible Watermark

The proposed algorithm is composed of two parts for the embedding of a watermark, one is *formatting*, and the other is *embedding*. The summary of the operation is shown below.

Formatting: For a scanned pixel $x_{i,j}, (0 \le i \le N-1, 0 \le j \le M-1)$ by a raster scan order, the following operations are performed.

1. Calculate the prediction error $\epsilon_{i,j}$.
2. Modify $x_{i,j}$ to $\bar{x}_{i,j}$ based on the following three conditions.

$$\bar{x}_{i,j} = \begin{cases} x_{i,j} + \epsilon_{i,j} & if\ x_{i,j} \in S_1 \\ x_{i,j} - 1 & if\ x_{i,j} \in S_2\ and\ \epsilon_{i,j}\ is\ odd \\ x_{i,j} & otherwise \end{cases} \tag{6}$$

The modification implies,

$$\bar{\epsilon}_{i,j} = \begin{cases} 2\epsilon_{i,j} & if\ x_{i,j} \in S_1 \\ \epsilon_{i,j} - 1 & if\ x_{i,j} \in S_2\ and\ \epsilon_{i,j}\ is\ odd \\ \epsilon_{i,j} & otherwise. \end{cases} \tag{7}$$

After the above modification, the prediction error $\bar{\epsilon}_{i,j}$ becomes even number if $x_{i,j} \in S_4$.

3. The location map is set to $L_{i,j} = 0$ if $x_{i,j} \in S_1$, otherwise $L_{i,j} = 1$.
4. If $x_{i,j} \in S_2$, then the LSB of $\epsilon_{i,j}$ is added to a vector $\boldsymbol{B}$ as one element.

Note that for the prediction of a current pixel $x_{i,j}$, the previously formatted four pixels which are specified in Fig.1 are used.

The bit-stream of the location map $L_{i,j}$ is compressed by JBIG2. We call the bit-stream of the compressed map $\boldsymbol{L}$. Then, $\boldsymbol{L}$, $\boldsymbol{B}$, and a watermark information bit-stream $\boldsymbol{W}$ are embedded. Combining those bit-streams, the embedding information $\boldsymbol{w}$ is produced.

$$\boldsymbol{w} = \{\boldsymbol{head}||\boldsymbol{L}||\boldsymbol{B}||\boldsymbol{W}\} \tag{8}$$

$$= \{w_t|1 \leq t \leq \sigma\} \tag{9}$$

Where ***head*** is a header file of the embedding information, $||$ means concatenation, and σ is the bit-length of $\boldsymbol{w}$. Here, σ can be represented as

$$\sigma = len(\boldsymbol{head}) + len(\boldsymbol{B}) + len(\boldsymbol{L}) + len(\boldsymbol{W}), \tag{10}$$

$$= N_1 + N_2, \tag{11}$$

where the function $len(x)$ outputs the bit-length of x, and N_1 and N_2 are the number of pixels in S_1 and S_2 respectively.

Embedding: After the above formatting operation, every prediction error, not pixel value, becomes even number. The embedding operation is simply to add w_t directly to each pixel $x_{i,j} \in S_4$ by a raster scan order, which implies to insert w_t into the LSB of the formatted prediction error $\bar{\epsilon}_{i,j}$.

By setting a counter $t = 1$, the following operations are performed repeatedly for each formatted pixel $\bar{x}_{i,j}$.

1. Modify $\bar{x}_{i,j}$ to $x'_{i,j}$ using the embedding information bit w_t

$$x'_{i,j} = \begin{cases} \bar{x}_{i,j} + w_t & if\ x_{i,j} \in S_4 \\ \bar{x}_{i,j} & otherwise \end{cases} \tag{12}$$

2. Increment $t = t + 1$ if $x_{i,j} \in S_4$.
3. If $t \leq \sigma$, go back to the step 1, otherwise quit.

3.2 Extraction and Recovery

On a reversible watermark, an original image is recovered from a watermarked image using an embedded information. Therefore, a watermark is first extracted from a watermarked image, and then the original image is recovered.

Extraction: On the prediction of JPEG-LS, the scanning order is very important to recover the original image, and it is performed by a raster scan order. When the embedded information is extracted, each information bit is extracted by this order. Since each prediction error is calculated from the previously formatted four pixels, the same pixels are required for the prediction at the extraction. Therefore, the extraction is performed by the following steps for each raster scanned pixel.

1. Set a counter $t = 1$.
2. Calculate the prediction error $\epsilon_{i,j}$ for a target pixel $x'_{i,j}$.
3. If $x'_{i,j}$ is changeable, which implies $x_{i,j} \in S4$, then the following operations are performed.
   3-1. Extract the LSB of $\epsilon'_{i,j}$ as t-th embedding information bit w_t.
$$w_t = \epsilon'_{i,j} \pmod 2, \tag{13}$$
   3-2. In order to make the prediction error even number, subtract w_t from $x'_{i,j}$.
$$\bar{x}_{i,j} = x'_{i,j} - w_t \tag{14}$$
   3-3. Store the re-formatted prediction error $\bar{\epsilon}_{i,j}$.
$$\bar{\epsilon}_{i,j} = \epsilon'_{i,j} - w_t \tag{15}$$
   3-4. Increment $t = t + 1$.
4. Perform the above step 2 and step 3 using the re-formatted four pixels until all pixels are checked.

Recovery: If the embedding information $\boldsymbol{w}$ is completely extracted and the original formatted image is recovered, then the original image is recovered using $\boldsymbol{w}$ and each re-formatted prediction error $\bar{\epsilon}_{i,j}$. The procedure is described below.

1. $\boldsymbol{w}$ is divided into three bit-streams, $\boldsymbol{L}$, $\boldsymbol{B}$, and $\boldsymbol{W}$ using the header file $\boldsymbol{head}$ which is predefined bits from the top of $\boldsymbol{w}$.
2. Stretch $\boldsymbol{L}$ to obtain a location map $L_{i,j}$.
3. Using $\boldsymbol{B} = \{B_t | 1 \le t \le len(\boldsymbol{B})\}$, each original pixel $x_{i,j}$ is recovered.

$$x_{i,j} = \begin{cases} \bar{x}_{i,j} - \frac{\bar{\epsilon}_{i,j}}{2} & if\ L_{i,j} = 0 \\ \bar{x}_{i,j} + B_t & if\ L_{i,j} = 1\ and\ x'_{i,j}\ is\ changeable \end{cases} \tag{16}$$

3.3 Capacity

The capacity of our scheme is dependent on the number of pixels in S_1 and the compression ratio of the location map. In our scheme, each embedding information bit w_t is inserted into the LSB of the prediction error of the corresponding pixel in S_4. Notice that for a pixel in S_2, the LSB is just replaced by w_t and the LSB is preserved in $\boldsymbol{B}$, which can not be compressed in theory because of its randomness. $\boldsymbol{B}$ is, therefore, directly embedded and the bit-length $len(\boldsymbol{B})$ must be equal to N_2. As the result, the bit-length of the watermark information is represented as follows using Eq.(10) and Eq.(11).

$$len(\boldsymbol{W}) = N_1 - len(\boldsymbol{head}) - len(\boldsymbol{L}). \tag{17}$$

4 Enhancement of the Performance

For the improvement of the capacity, one simple method is to increase N_1 in Eq.(17) by enlarging a threshold T, but it causes the degradation of an image. In order to increase the capacity without degrading the perceptual quality, we propose two methods to reduce the size of compressed location map, $len(\boldsymbol{L})$, in this section.

4.1 Vectorization

Since the embedding operation is performed for each pixel in the basic scheme, each pixel needs the corresponding location map, which becomes the same size of an image. Although it is compressed, the size may be still large. In order to decrease the location map itself, several pixels are put together into one vector which is judged expandable if all pixels in the vector are in S_1. Such pixels should be selected carefully from an image in our scheme, because the raster scan order must be followed. Generally, there is a strong mutual relation among neighboring pixels, the vectorization can increase the capacity efficiently.

1. For successive m pixels, the formatting operation is performed.
2. If all m pixels are in S_1, a reduced location map is set to $\ell_{i,j/m} = 0$ and go to the next m pixels. Otherwise, $\ell_{i,j/m} = 1$ and performs the step 3 to step 5 using the original m pixels successively.
3. Calculate the prediction error $\epsilon_{i,j}$ based on Eq.(1).
4. Modify $x_{i,j}$ to $\bar{x}_{i,j}$ based on the following two conditions.

$$\bar{x}_{i,j} = \begin{cases} x_{i,j} - 1 & if\ x_{i,j} \in S_4\ and\ \epsilon_{i,j}\ is\ odd \\ x_{i,j} & otherwise \end{cases} \tag{18}$$

5. If $x_{i,j} \in S_4$, then the LSB of $\epsilon_{i,j}$ is added to a vector $\boldsymbol{B}$ as one element.

After the above operations, the produced location map consists of $N \times M/m$ elements. Therefore, the size of the location map becomes $1/m$ compared with the basic one. The vectorization method reduces the size of location map with a little sacrifice of the capacity.

In the vectorized scheme, each vector is classified according to the Definition 1, Definition 2, and a threshold T. The threshold controls the trade-off between the capacity of a watermark information and the distortions caused by the embedding. If at least one pixel in a vector is not in S_1, the vector is regarded as non-expandable. Then, instead of expansion, each pixel in the vector is modified as a changeable or non-changeable ones. It is remarkable that the decoder of the vectorized scheme can apply the same one as the basic scheme because the embedded information is extracted from the LSB of the prediction errors of pixels in S_4. In order to recover the original image, the applied method must be informed, which is easily realized by adding such information to the header file ***head***.

4.2 Composition of Location Map

In order to reduce the size of $\boldsymbol{L}$ more effectively, we reconsider the composition of the map. In the basic scheme, the map is merely produced by putting "0" or "1" symbol to each element if a target pixel is in S_1 or not, and the information is required for the recovery for the original pixel because the operation is dependent on the pixel if it was in S_1 or not. Here, it is remarkable that several pixels are still in S_1 after the embedding. For such pixels, the location map is not required for the recovery operation because they are determined by themselves.

When a pixel x is in S_1 and its prediction error ϵ satisfies an inequality $|\epsilon| < T/2$, the formatted pixel $\bar{x}$ also belongs to a group S_1 for the formatted prediction error $\bar{\epsilon}(= 2\epsilon)$ if x_{ae},

$$x_{ae} = \bar{x} + 2\epsilon + w, \tag{19}$$

$$= x + 3\epsilon + w, \tag{20}$$

dose not cause overflow and underflow, where $w \in \{0, 1\}$. In order to classify such pixels, we define *absolutely expandable*.

Definition 3. *The pixel x is said to be absolutely expandable if, for any $w \in \{0, 1\}$, x can be modified to x_{ae} without causing overflow and underflow, and $|\epsilon| < T/2$ is satisfied.*

For a pixel $x' \in S_2$, there are two possible candidates for the group to which the original pixel x belonged, namely S_1 and S_2. There are also several pixels in S_2 that the original group can be determined by themselves if those prediction error satisfy an inequality $|\epsilon| > 2T$ because of the following reason. When a pixel x belongs to a group S_1, the prediction error ϵ is less than T. It means that after the formatting operation, the modified prediction error $\bar{\epsilon}(= 2\epsilon)$ must be less than $2T$. Therefore, if $|\epsilon| > 2T$, such a pixel must be in S_2, which is also hold in the recovery operation. For the classification, we define *absolutely changeable* as follows.

Definition 4. *The pixel x is said to be absolutely changeable if x is changeable and $|\epsilon| > 2T$ is satisfied.*

By introducing the Definition 3 and 4, several pixels in S_1 and S_2 can be belongs to an extra group S_5 which contains absolutely expandable pixels and absolutely changeable pixels. Although each pixel $x_{i,j}$ has a corresponding location map $L_{i,j}$ in the basic scheme, the map in pixels in S_5 can be omitted, which contributes on the reduction of the size of $len(\boldsymbol{B})$. One simple method for the reduction is merely to remove the corresponding information of the map. Considering the compression algorithm of JBIG2, however, the map should be two-dimensional and hence it causes problem. Instead, we manage to compose the location map in order to enhance the performance of JBIG2.

First, we compose the location map by the following rules.

$$L_{i,j} = \begin{cases} * & if\ x_{i,j} \in S_5 \\ 0 & if\ x_{i,j} \in S_1\ except\ x_{i,j} \in S_5 \\ 1 & otherwise \end{cases} \tag{21}$$

Where the symbol “*” indicates “0” or “1” dependent on the contexts. It is certain that higher compression ratio can be achieved by adaptively setting the symbol “*”. In this paper, for the adaptive modification of the map, the following method is applied. We merely set all the symbols to “0” or “1”, compress the two types of the modified map, and adopt the better one. Such selection is very simple, but it contributes greatly for the reduction of the compressed data size.

If the location map is composed by the above procedure, each pixel $\bar{x}_{i,j}$ must be checked whether $x_{i,j}$ is in S_5 before a recovery operation is performed. If $\bar{x}_{i,j}$ is expandable for its prediction error $\bar{\epsilon}_{i,j}$ which is less than T, $x_{i,j}$ must be absolutely expandable. And if $|\bar{\epsilon}_{i,j}|$ is more than T, $x_{i,j}$ must be absolutely changeable. For such cases, the recovered location map from the extracted $\boldsymbol{w}$ is not referred for the recovery of the pixels. Notice that a threshold T is necessary to judge a pixel absolutely expandable/changeable. Therefore, such information should be added to the header file ***head***.

5 Experimental Results

We have implemented our algorithm and estimated the capacity and distortions with the basic scheme and the enhanced scheme. The images used in the evaluation are “lena”, “baboon”, “fruits”, and “F16” with RGB color of 512×512 pixels. We tested the algorithm for each RGB color components respectively with the same threshold T. In the following simulation, the capacity is calculated by omitting the size of ***head*** as it is negligibly small (It may be less than 100 bits).

The capacity obtained with various vector size m for the image “lena” is plotted against the PSNR in Fig.3. This figure reveals that the capacity is increased according to the increase of the vector size m. The similar results are obtained from other images. To achieve a large capacity, the vectorization seems one method for the improvement of the basic scheme. Next, the capacity of the enhanced scheme which modifies the composition of the location map is shown in Fig.4. It is clear that the capacity is improved at all range compared with the basic scheme and its vectorized scheme. It is remarkable that the largest capacity can be achieved for a basic non-vectorized scheme even if it can not make a space to embed a watermark when the value of PSNR is high. If one wants to get better performance for overall range, we recommend to use $m = 4$. In the rest of this section, the results are obtained for the constant parameter $m = 4$.

The effects of the compression in the modified location map is numerically estimated. Figure 5 shows the size of the compressed location map for both the basic scheme and the enhanced scheme. From the result, the location map is well compressed by JBIG2 when the the map is adaptively composed. Since our method simply assigns “0” or “1” symbol to the location map based on the new conditions *absolutely expandable* and *absolutely changeable*, a further optimization can be achieved by composing the map more adaptively considering the applied compression algorithm, which is our future work. For the evaluation of other images, the capacity of the enhanced scheme is shown in Fig.6. The

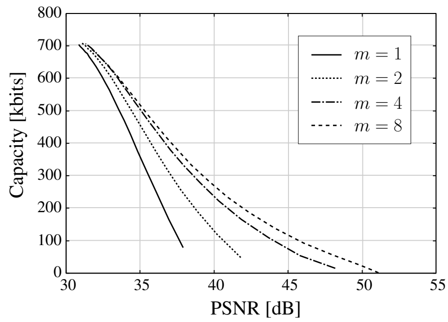


Fig. 3. The comparison of the performance for the number of element of the composed vector for an image "lena".

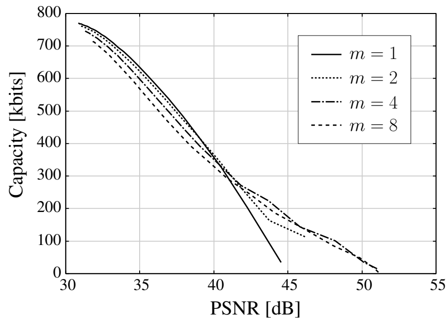


Fig. 4. The capacity improved by the modification of a location map for an image "lena".

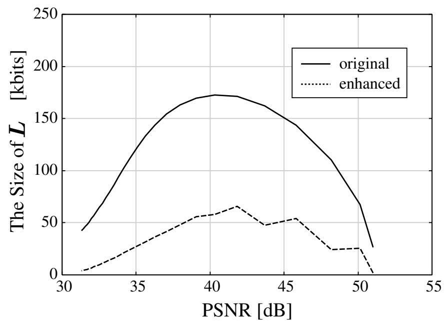


Fig. 5. The comparison with the compressed data size of a location map for an image "lena".

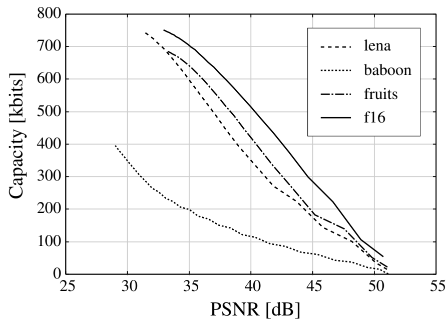


Fig. 6. The capacity of embedding information versus PSNR.

results reveal that the capacity is variable for images. Although the performance of the predictor in JPEG-LS at a flat region is superior to noisy region in a image, a watermark is spread all over the image. The effects caused by the modification of the location map is numerically estimated, and the amount of increased capacity is shown in Fig.7. The contribution in the increase of the capacity is dynamically changed for each image and each PSNR because of the characteristic of the compression algorithm of JBIG2.

When a watermark is embedded, a kind of sharpening effects is appeared and the effects grow stronger for the increase of the amount of watermark information. For the numerical evaluation of the image quality, the relation between PSNR and threshold T is shown in Fig.8. The size of threshold implies the amount of maximum changes caused by the embedding. As the images "lena", "fruits", and "f16" contains a lot of flat regions, the prediction errors are distributed in a small range, and hence the curve is slowly decreased and reached

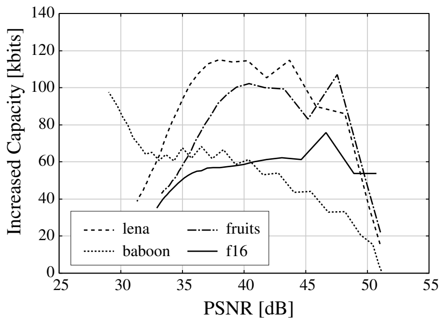


Fig. 7. The improved capacity by reducing the size of the location map.

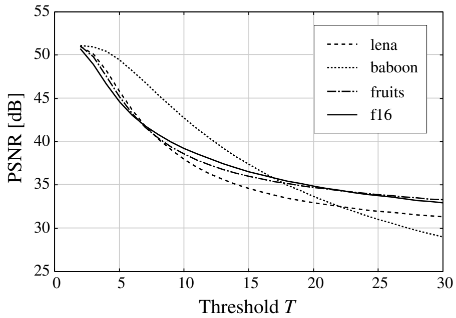


Fig. 8. The relation between PSNR and threshold T.

its lower band for rather small T. On the other hand, as most parts of the image "baboon" are noisy, the curve becomes rapidly down.

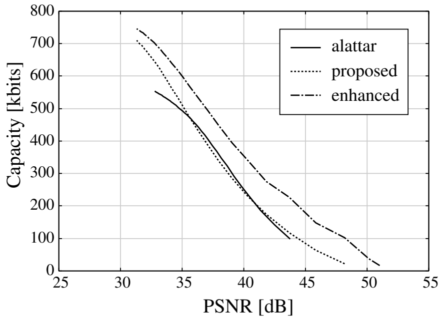
We also compare the performance of our scheme with that of Alattar[5] which achieves a large capacity at the state-of-art schemes. In the work, largest capacity can be obtained when three bits are embedded in each vector of four pixels, which is similar to our scheme as a better performance is obtained for $m = 4$. Under such conditions, the comparisons are shown in Fig. 9. The results clarify that our enhanced scheme is superior to the conventional one for all test images.

6 Conclusion

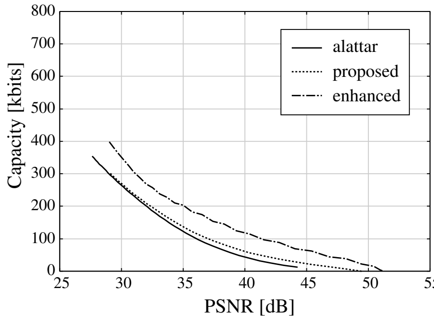
In this paper, a reversible watermark with a very large capacity based on the prediction errors calculated in JPEG-LS has been proposed. Since the prediction errors follow TSGD model centered at zero, the distortions caused by our embedding is kept small. In addition, a watermark is spread all over the image in our scheme, though the conventional schemes embed mainly in the flat region. Those properties contribute to the improvement of the perceptual quality. In order to improve the capacity, we compose a vector from several pixels and the size of location map is decreased effectively exploiting the characteristic of a pixel/vector. From our simulation results, a better performance can be obtained when successive four pixels are treated as a vector. Our future work is to produce a location map adaptively considering the applied compression algorithm, and to try other predictive coding for our technique.

References

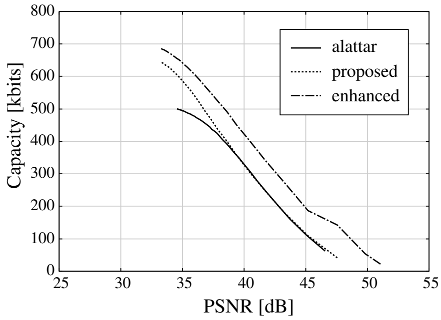
1. S. Katzenbeisser and F. A. P. Petitcolas, *Information hiding techniques for steganography and digital watermarking.* Artech house publishers, Jan. 2000.
2. M. U. Celik, G. Sharma, A. M. Tekalp, and E. Saber, "Lossless generalized-LSB data embedding," *IEEE Trans. Image Process.,* vol.14, no.2, pp.253-266, 2005.



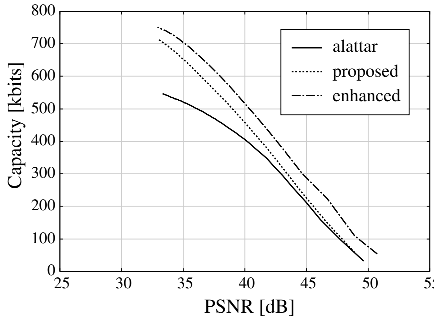
(a) lena



(b) baboon



(c) fruits



(d) f16

Fig. 9. The comparison of the capacity, where the vector is composed from 4 pixels ($m = 4$).

3. J. Tian, "Reversible data embedding using a difference expansion," *IEEE Trans. Circuits Syst. Video Technol.,* vol.13, no.9, pp.890-896, 2003.
4. H. Heijmans and L. Kamstra, "Reversible data embedding based on the haar wavelet decomposition," *Proc. of DICTA2003*, pp.5-14, 2005.
5. A. M. Alattar, "Reversible watermark using the difference expansion of a generalized integer transform," *IEEE Trans. Image Process.,* vol.13, no.8, pp.1147-1156, 2004.
6. M. J. Weinberger, G. Seroussi, and G. Sapiro, "The LOCO-I lossless image compression algorithm: Principles and standardization into JPEG-LS," *IEEE Trans. Image Process.,* vol.9, no.8, pp.1309-1324, 2000.
7. A. Netravali and J. O. Limb, "Picture coding: a review," *Proc. of IEEE,* vol.68, pp.366-406, 1980.
8. S. W. Golomb, "Run-length encodings," *IEEE Trans. Inform. Theory,* vol.IT-12, pp.399-401, 1966.
9. D. M. Thodi and J. J. Rodrfguez, "Reversible watermark by prediction-error expansion," Proc. of 6th IEEE Southwest Symposium on Image Analysis and Interpretation, pp.21-25, 2004.