Reversible Watermark with Large Capacity Based on the Prediction Error Expansion

Kuribayashi, Minoru

Morii, Masakatu

Tanaka, Hatsukazu

(Citation)
IEICE transactions on fundamentals of electronics, communications and computer sciences,E91-A(7):1780-1790

(Issue Date)
2008-07-01

(Resource Type)
journal article

(Version)
Version of Record

PAPER

Reversible Watermark with Large Capacity Based on the Prediction Error Expansion

Minoru KURIBAYASHI[†a)], **Masakatu MORII**[†], ***Members, and*** **Hatsukazu TANAKA**[††], ***Fellow***

SUMMARY A reversible watermark 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 watermark, which contributes to increase the amount of embedded information with less degradation. As one of the drawbacks discovered in the above conventional method is the 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 characteristics. We also exclude the positions such that the modification in the embedding operation cannot increase the capacity but merely degrade the image quality, which can be applicable to the conventional methods.

key words: *reversible watermark, prediction error, location map*

1. Introduction

In a general data-hiding technique [1], an embedding operation causes irreversible degradation to an image. Although the degradation is perceptually slight, it may not be acceptable to some applications such as medical or military images. For the countermeasure, a lossless data-hiding technique, which is called reversible(invertible) watermark, has been developed. In general, in a fragile watermark, some information about an image, e.g. LSB bit-plane, is removed for embedding a watermark. In the reversible watermark, since an original image is completely restored from the watermarked image, it is possible to assure its originality. Although the localization of the altered area is impossible for the reversible watermark, the invertibility will be very attractive for some applications such as medical images, military satellite images, etc. If a digital signature is embedded as the watermark, an intentional alteration of image is detected from an attacked image, and its security is dependent on the applied signature scheme. One of the applications of reversible watermark is to manage medical images binding with patient's private information by embedding the information. In this application, the sharing of medical images among doctors as the resources for research without revealing private information and the protection for the leak of sensitive information are possible. Only a primary doctor can retrieve the embedded information.

Manuscript received June 12, 2007.
Manuscript revised February 8, 2008.
[†]The authors are with the Graduate School of Engineering, Kobe University, Kobe-shi, 657-8501 Japan.
[††]The author is with the Kobe Institute of Computing, Kobe-shi, 650-0001 Japan.
a) E-mail: kminoru@kobe-u.ac.jp
DOI: 10.1093/ietfec/e91–a.7.1780

There are many important things in the reversible watermark technique; for example transparency, secrecy, capacity, computational complexity, etc. Generally, the perceptual quality of a watermarked image is degraded according to the amount of watermark information. In our paper, we focused on not only the capacity, but also the perceptual quality, which is referred as *capacity-distortion behavior*. If watermark information is attached with a compressed original image instead of embedding and the total file size is much smaller than the compressed watermarked image, the availability of reversible watermarking technique will narrow down. Hence, the evaluation of the total file size after compression is considered in our paper.

The reversible watermark technique 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 maintains 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] has presented a difference expansion transform of a pair of pixels, which is identical as the 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 each pair from those pairs that are not expected to cause overflow or underflow. In order to recover the original image, a location map that indicates the modified pairs is embedded as a part of the embedding information after compression. Kamstra et al. [4] improved the capacity for high quality images by exploiting low-pass filtered image to predict the location map. Alattar [5] extended Tian's method to the difference expansion of a vector of several pixels to achieve larger capacity. If the vector consists of n pixels, the capacity is at most $n/(n-1)$ [bpp] (bit per pixel) and the size of location map is reduced to $1/n$ of its original size. However, if n is increased, the embeddable positions are decreased, hence it is difficult to achieve the maximum capacity. Moreover, although the location map is compressed, it does not remove all redundancy contained in it.

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], and to reduce the size of location map effectively. Since the average value of the prediction errors in JPEG-LS 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. For the reduction of the location map, we propose an best composition of a vector suitable for the prediction based expansion method. And we study the local characteristics of the local conditions of pixels, and find that at some pixels the location map is not necessary if special conditions, which are called *absolutely expandable (AE)* and *absolutely changeable (AC)*, are satisfied. From our study of composition of the location map, it is suitable to exclude the information corresponding to AE and AC. Then, the modified location map is efficiently compressed by arithmetic coding. It is remarkable that pixels of AC contribute not on the increase of the capacity, but on the degradation of the image quality. Hence, by excluding such pixels from the embedding position, embedding a small message at low distortion becomes possible. Since such a modification using AE and AC is also applicable to the conventional schemes, we estimate the effects of the modification in our simulation.

The preliminary version of this paper was presented in [7], but its simulation results contain errors which are related not to the basic embedding algorithm, but to the improved algorithm for the composition of the location map. From our computer simulated results, we found that the method in [7] does not enhance the performance of the compression algorithm. Therefore, in this paper we improve the compression ratio of the location map by composing the map with new approaches, which are presented in Sect. 4.

In the next section, the different expansion method is reviewed, and the predictor applied in JPEG-LS and its property are summarized. The basic algorithm of our reversible watermark is shown in Sect. 3. In Sect. 4, the capacity is improved by vectorizing several pixels, and by exploiting the conditions, AE and AC, to reduce the size of location map. The numerical results are described in Sect. 5. Finally, we summarize our conclusions.

2. Preliminary

2.1 Expanding Differences

For embedding a watermark, differences among pixels are applied. Tian [3] uses two neighboring pixels, Alattar [5] generalizes the scheme for neighboring n pixels, and we calculate prediction errors for each pixel using a predictor in JPEG-LS [6]. Tian's algorithm is capable of embedding as high as 0.5 [bpp] because one bit is embedded in the difference of two pixels, and Alattar's one is as high as $(n-1)/n$ [bpp] for each difference of n pixels. On the other hand, our scheme can be at most 1 [bpp] as pixel-wise operation is possible by applying the prediction error instead of differences among pixels. Note that multiple embedding is possible for those methods, the actual capacity becomes more than 1 [bpp]. Although the applied differences are different in those schemes, the basic idea is identical.

Let d be a given difference among selected pixels and $w \in \{0,1\}$ be a watermark. The embedding operation based on the expansion of differences is given as follows.

$$d_e = 2d + w \tag{1}$$

In the inverse operation, it is sufficient to calculate $d_e/2$ after extracting w ($= d_e \bmod 2$). The pixels are carefully selected such that they remain identifiable after embedding and they do not suffer from overflow or underflow after embedding. In general, each pixel is represented by 8-bit scale of $[0, 255]$, and overflow and underflow mean that the pixel value becomes more than 255 and less than 0, respectively. On the expansion of differences, overflow and underflow of pixel values should be avoided by carefully selecting the target pair/vector of adjacent pixels. In [3] d is the difference of two selected adjacent pixels. On the other hand, for each pixel the difference d_i, $(1 \le i \le m-1)$ from the mean value d_0 of a set of m pixels x_i, $(0 \le i \le m-1)$ is treated in [5], where

$$d_0 = \frac{1}{n}\sum_{i=0}^{m-1} x_i \tag{2}$$

$$d_i = x_i - d_0, \quad (1 \le i \le m-1). \tag{3}$$

In order to control the embedding operation, the following definition is introduced.

Definition 1: The set of m pixels x_i, $(0 \le i \le m-1)$ is said to be difference expandable if, for any $w_i \in \{0,1\}$, $(1 \le i \le m)$, every difference d_i in the set can be modified to $d_{i,e}$ without causing overflow and underflow in the set of pixels.

$$d_{i,e} = 2d_i + w_i, \quad (1 \le i \le m-1) \tag{4}$$

The group of expandable sets is called E_d.

It seems to be sufficient to modify the difference expandable set of pixels to embed a watermark. However, it is difficult to distinguish if the modified set is originated from the expandable set at the extraction step. Hence, the other classification of the set is necessary.

Definition 2: The set of m pixels x_i, $(0 \le i \le m-1)$ is said to be difference changeable if, for any $w_i \in \{0,1\}$, $(1 \le i \le m-1)$, every difference d_i can be modified to $d_{i,c}$ without causing overflow and underflow.

$$d_{i,c} = 2\left\lfloor \frac{d_i}{2} \right\rfloor + w_i, \quad (1 \le i \le m-1) \tag{5}$$

The group of changeable sets is called C_d.

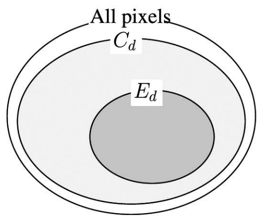


Fig. 1 The organization of the sets E_d and C_d.

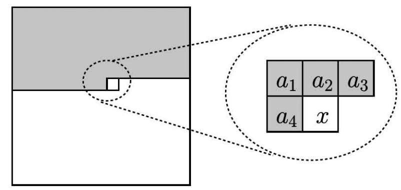


Fig. 2 The sample points of the fixed predictors.

It is noted that E_d is included in C_d, which is illustrated in Fig. 1. In addition, even if the operation of Eqs. (4)/(5) is performed to the set of n pixels, the set is still in C_d because the operation Eq. (5) can be preformed again to the set without overflow and underflow. After such operations, embedding information is inserted in the LSB of the differences d_e and d_c. Hence, it is extracted from the LSB of such differences that are determined C_d at the extraction without any additional information.

To ensure invertibility, the locations of the expanded differences and the original LSBs of d_i in $C_d \setminus E_d$ are appended to a watermark. Because some sets in E_d turn into $C_d \setminus E_d$ after the expanding operation, the locations of the original group E_d or $C_d \setminus E_d$, which is called location map, must be stored.

In general, there is a trade-off between the distortions and the capacity in watermark technique, and it is desirable to control the trade-off depending on applications. It is achieved by introducing a threshold T for the determination of difference expandable or not. When Definition 1 is satisfied and $|d_i| < T$, the pixel is regarded as expandable. Then, the changes caused by the operation of Eq. (4) are restricted less than T, and hence the degradation of the image quality is controlled.

Since the size of the location map is basically very large, it is compressed by any lossless compression algorithm. Let $\boldsymbol{L}$ be the compressed location map, $\boldsymbol{B}$ be the LSBs of d in $C_d \setminus E_d$, and $\boldsymbol{W}$ be a watermark. Then, embedding information $\boldsymbol{w}$ is

$$\boldsymbol{w} = \boldsymbol{head}||\boldsymbol{L}||\boldsymbol{B}||\boldsymbol{W} \tag{6}$$

$$= \{w_t | w_t \in \{0, 1\}, 1 \leq t \leq \sigma\}, \tag{7}$$

where $\boldsymbol{head}$ 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

$$\sigma = len(\boldsymbol{head}) + len(\boldsymbol{L}) + len(\boldsymbol{B}) + len(\boldsymbol{W}) \tag{8}$$

where the function $len(x)$ outputs the bit-length of x. The embedding information $\boldsymbol{w}$ is embedded into all LSBs of differences of pixels in C_d which are selected from non-overlapping adjacent pixels,

$$\sigma = N_{E_d} + N_{C_d \setminus E_d}, \tag{9}$$

where N_{E_d} and $N_{C_d \setminus E_d}$ are the number of pixels in E_d and $C_d \setminus E_d$, respectively.

In the conventional schemes, the calculation of differences is based on integer wavelet transform. The basic method proposed by Tian [3] has been generalized by Alattar [5], and further improved methods were proposed later. Xuan et al. [8] applied a companding technique for the difference expansion method and Kamstra et al. [4] utilized the low-pass image of wavelet transform, which has not been focused on before, in order to estimate locally the expandability of the high-pass image. Thodi and Rodriguez [9] applied a predictor of lossless image compression instead of the wavelet transform. The applied predictor guesses a current pixel x from three sample points a_1, a_2, and a_4 depicted in Fig. 2. Since such a predictor is simple, the performance can be improved by a well-designed predictor like that of JPEG-LS. Moreover, because of its embedding/recovering algorithm, the composition of location map is different from the method in [3]–[5], which is a disadvantage for the adaption of our improved method.

2.2 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 a_1, a_2, a_3, and a_4 depicted in Fig. 2, and the current sample x is predicted as $\hat{x}$ using the four samples and the information concerning the previously predicted results. In the coding part, the prediction error ϵ

$$\epsilon = x - \hat{x} \tag{10}$$

is encoded with an extended family of Golomb-type code [11] 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) [10] centered at zero. And ϵ is also extremely sensitive to the changes in the four samples. Note that the change occurred in one sample is propagating to the following every prediction error.

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 this 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 [11]. Therefore, if the original prediction error can be recovered from a watermarked image, the embedding operation is reversible.

It seems difficult to modify the prediction error with the previously occurring pixels a_1, a_2, a_3, and a_4 because those pixels are also used for the prediction of other pixels. Instead, we modify the current target pixel x, which is easily calculated by the following equations.

$$x_e = x + \epsilon + w \tag{11}$$

$$x_c = x + (\epsilon \bmod 2) + w \tag{12}$$

Here, overflow and underflow of the modified pixel must be considered, and similar to the conventional schemes that consider the trade-off between distortions and capacity, a threshold T should be applied for the determination of expandable or not. In order to control the embedding operation, the following definitions are introduced for the expansion of prediction errors.

Definition 3: The pixel x is said to be prediction error expandable if, for any $w \in \{0, 1\}$, x can be modified to x_e without causing overflow and underflow.

Definition 4: The pixel x is said to be prediction error changeable if, for any $w \in \{0, 1\}$, x can be modified to x_c without causing overflow and underflow.

Each pixel can be classified into two groups according to Definitions 3 and 4. One group contains all prediction error expandable pixels whose prediction errors less than a predefined threshold T, and the other group contains all prediction error changeable pixels. For simplicity, we call them expandable/changeable hereafter, and represent them by the group E and C, respectively.

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 inverse operation for the embedding is shown.

3.1 Embedding Reversible Watermark

The proposed algorithm is composed of two parts for the embedding of a watermark; `Formatting` and `Embedding`. In the *formatting*, we estimate the expandability of the prediction errors calculated in a raster scan order. Then, for the current pixel $x_{i,j}$ if the prediction error $\epsilon_{i,j}$ is in E, it is expanded using Eq. (11). If $\epsilon_{i,j}$ is in $C \setminus E$, its LSB is set to "0," which is based on the operation of Eq. (12). After such an operation, every prediction error, not pixel value, becomes an even number. The embedding operation is simply to add each piece of the embedding information w_t directly to each pixel $x_{i,j} \in C$, which implies the insertion of w_t into the LSB of the formatted prediction error $\bar{\epsilon}_{i,j}$. Note that for the prediction of a current pixel $x_{i,j}$, previously formatted four pixels which are specified in Fig. 2 are used.

`Formatting:` For a given image, every prediction error is set to even number as follows.

1. Set a counter $t = 1$.
2. Calculate a prediction error $\epsilon_{i,j}$ for $x_{i,j}$, and the following operations are performed.
   2-1. Modify $x_{i,j}$ to $\bar{x}_{i,j}$ in order to make the prediction error $\bar{\epsilon}_{i,j}$ to be even number when $x_{i,j} \in C$.
   $$\bar{x}_{i,j} = \begin{cases} x_{i,j} + \epsilon_{i,j} & \textit{if } x_{i,j} \in E \\ x_{i,j} + (\epsilon_{i,j} \bmod 2) & \textit{if } x_{i,j} \in C \setminus E \\ x_{i,j} & \textit{otherwise} \end{cases} \tag{13}$$
   2-2. The location map is produced.
   $$L_{i,j} = \begin{cases} 0 & \textit{if } x_{i,j} \in E \\ 1 & \textit{otherwise} \end{cases} \tag{14}$$
   2-3. If $x_{i,j} \in C \setminus E$, then the LSB of $\epsilon_{i,j}$,
   $$B_t = \epsilon_{i,j} \bmod 2, \tag{15}$$
   is added to $\boldsymbol{B}$ as t-th element.
   2-4. Increment $t = t + 1$ if $x_{i,j} \in C$.
3. Perform step 2 until all pixels are processed.

`Embedding:` For a formatted image, embedding information $\boldsymbol{w}$ which is produced based on Eq. (6) is embedded as follows.

1. Set a counter $t = 1$.
2. Modify $\bar{x}_{i,j}$ to $x'_{i,j}$ using the embedding information bit w_t
   $$x'_{i,j} = \begin{cases} \bar{x}_{i,j} + w_t & \textit{if } x_{i,j} \in C \\ \bar{x}_{i,j} & \textit{otherwise} \end{cases} \tag{16}$$
3. Increment $t = t + 1$ if $x_{i,j} \in C$.
4. If $t \leq \sigma$, go back to the step 2, otherwise quit.

Remark 1: In the step 2 at `formatting`, the modification implies,

$$\bar{\epsilon}_{i,j} = \begin{cases} 2\epsilon_{i,j} & \textit{if } x_{i,j} \in E \\ 2\left\lfloor \dfrac{\epsilon_{i,j}}{2} \right\rfloor & \textit{if } x_{i,j} \in C \setminus E \\ \epsilon_{i,j} & \textit{otherwise.} \end{cases} \tag{17}$$

Hence, each prediction error in C is set to even number.

3.2 Extraction and Recovery

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

On the prediction of JPEG-LS, the scanning order is very important to recover the original image. 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 for the raster scanned pixel such that the former pixels are re-formatted similar to the `formatting`. In this operation, not only the embedding information $\boldsymbol{w}$ is extracted, but also the formatted image and prediction errors are concurrently recovered. For convenience, we call the operation `extraction`.

After the `Extraction`, the original image is recovered using $\boldsymbol{w}$ and each re-formatted prediction error $\bar{\epsilon}_{i,j}$ by a `Recovery` operation.

`Extraction:` For a watermarked image, embedded information $\boldsymbol{w}$ is extracted as follows.

1. Set a counter $t = 1$.
2. Calculate the prediction error $\epsilon'_{i,j}$ for a current pixel $x'_{i,j}$.
3. If $x'_{i,j}$ is changeable, which implies $x_{i,j} \in C$, then the following operations are performed.

   3-1. Extract the LSB of $\epsilon'_{i,j}$ as t-th embedding information bit w_t.

   $$w_t = \epsilon'_{i,j} \pmod 2, \tag{18}$$

   3-2. Recover the formatted pixel $\bar{x}_{i,j}$ from $x'_{i,j}$.

   $$\bar{x}_{i,j} = x'_{i,j} - w_t \tag{19}$$

   3-3. Store the re-formatted prediction error $\bar{\epsilon}_{i,j}$.

   $$\bar{\epsilon}_{i,j} = \epsilon'_{i,j} - w_t \tag{20}$$

   3-4. Increment $t = t + 1$.
4. Perform the above step 2 and step 3 until all pixels are checked.

`Recovery:` Using the extracted information $\boldsymbol{w}$, the formatted image, and re-formatted prediction errors $\bar{\epsilon}_{i,j}$, the original image is recovered as follows.

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 $\boldsymbol{B} = \{B_t | B_t \in \{0, 1\}, 1 \le t \le len(\boldsymbol{B})\}$, each original pixel $x_{i,j}$ is recovered.

$$x_{i,j} = \begin{cases} \bar{x}_{i,j} - \dfrac{\bar{\epsilon}_{i,j}}{2} & if\ L_{i,j} = 0 \\ \bar{x}_{i,j} + B_t & if\ L_{i,j} = 1\ and\ x'_{i,j} \in C \end{cases} \tag{21}$$

3.3 Capacity

The capacity of our scheme is dependent on the number of pixels in E and the compression ratio of the location map. Although prediction error is used in our scheme, the embedding information can be represented by $\boldsymbol{w}$ in Eq. (6) and its length is σ in Eq. (8), which is also rewritten by

$$\sigma = N_E + N_{C\setminus E}, \tag{22}$$

where N_E and $N_{C\setminus E}$ are the number of pixels in E and $C \setminus E$, respectively.

In our scheme, each embedding information bit w_t is inserted into the LSB of the prediction error of the corresponding pixel in C. Notice that for a pixel in $C \setminus E$, 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 $N_{C\setminus E}$. As a result, the bit-length of the watermark information is represented using Eq. (22) and Eq. (9) as follows.

$$len(\boldsymbol{W}) = N_E - len(\boldsymbol{head}) - len(\boldsymbol{L}). \tag{23}$$

4. Improvement of the Capacity

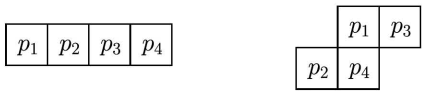
For the improvement of the capacity, one simple method is to increase N_E in Eq. (23) by enlarging a threshold T, but it causes the degradation of an image. In order to increase the capacity without degrading the perceptual quality, we propose two methods to reduce the size of compressed location map, $len(\boldsymbol{L})$, 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 still be large. In order to decrease the location map, several pixels are put together into one vector which is judged expandable if all pixels in the vector are in E. 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. One simple method for vectorization is to select successive pixels in a raster scan order, and in such a case the definitions of E and C are extended as follows.

Definition 5: The set of m pixels is said to be prediction error expandable if, for any $w_i \in \{0, 1\}$, $(1 \le i \le m)$, every pixels in the set can be expanded by the operation in Eq. (11) without causing overflow and underflow.

Definition 6: The set of m pixels is said to be prediction error changeable if, for any $w_i \in \{0, 1\}$, $(1 \le i \le m - 1)$, every pixels in the set can be changed by the operation in Eq. (12) without causing overflow and underflow.



(a) raster scan order (b) zigzag order

Fig. 3 Selection of 4 pixels for one vector.

The embedding procedure with vectorization is summarized as follows.

1. For successive m pixels in a raster scan order, the formatting operation is performed.
2. If a set of m pixels is in E, 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. (10).
4. Modify $x_{i,j}$ to $\bar{x}_{i,j}$ for the successive m pixels.

$$\bar{x}_{i,j} = \begin{cases} x_{i,j} + (\epsilon_{i,j} \bmod 2) & \text{if } x_{i,j} \in C \\ x_{i,j} & \text{otherwise} \end{cases} \tag{24}$$

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

After the above operations, the size of the produced location map is reduced m times from its original size. The vectorization method reduces the size of location map with a little sacrifice of the capacity. Because some pixels in a vector which is judged changeable are expandable, the number of embeddable positions are decreased by vectorization.

On the above simple method, a vector is composed of successive m pixels in a line. Considering the correlation among pixels, however, the selection is not suitable because the first pixel becomes far from the final pixel according to the increase of m. For example, the selected pixels in the above manner is shown in Fig. 3(a) for $m = 4$. On the four pixels p_1, p_2, p_3, p_4 that lie in a row, the distance between p_1 and p_4 is 3 pixels. For a better selection without loss of prediction in JPEG-LS, one candidate of the templates for $m = 4$ is depicted in Fig. 3(b), where the order of the selection is followed in the subscription number. In this template, the distance between pixels is less than $\sqrt{5}$ pixels, hence the mutual relation among them is stronger than the simple method. In order to evaluate the characteristic, their capacity is tested in our simulation, which is discussed in the following section.

In the vectorized scheme, each vector is classified according to Definition 1 and 2, and a threshold T. The threshold controls the trade-off between the capacity of watermark information and the distortions caused by embedding. If at least one pixel in a vector is not in E, the vector is regarded as non-expandable. Then, instead of expansion, each pixel in the vector is modified as a changeable or non-changeable one. 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 C. 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 E or not, and the information is required for the recovery of the original pixel because the operation is dependent on the pixel if it was in E or not. Since the location map reflects the characteristic of an original image, it contains redundancy, which is effectively compressed by JBIG2. Such a characteristic is also exploited in the conventional schemes [3]–[5]. However, we found that the location map contains further redundancy such that several parts of $L_{i,j}$ are not necessary at the recovery operation.

For example, several pixels in E are still in the same group 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 E and its prediction error ϵ satisfies an inequality $|\epsilon| < T/2$, the formatted pixel $\bar{x}$ also belongs to E for the formatted prediction error $\bar{\epsilon}(= 2\epsilon)$ for any $w \in \{0, 1\}$ if x_{ae},

$$x_{ae} = \bar{x} + 2\epsilon + w, \tag{25}$$
$$= x + 3\epsilon + w, \tag{26}$$

does not cause overflow and underflow.

Although some sets in E turn into $C \setminus E$ after the expanding operation, the others are still in E. Such sets do not require the location map at the extraction. Some sets of watermarked image which are in $C \setminus E$ are also determined by themselves that they are originated from $C \setminus E$ if $|\epsilon| \geq 2T$ because of the following reason. When a set belongs to E, the difference $|\epsilon|$ is less than T. It means that after the expanding operation, the expanded difference $|\epsilon_e|$ must be less than $2T$. Therefore, if $|\epsilon| \geq 2T$, such a set must be in C, which is also held at the recovery operation.

In order to classify such pixels, we defined *absolutely expandable* and *absolutely changeable* as follows.

Definition 7: The set of m pixels x_i, $(0 \leq i \leq m-1)$ is said to be absolutely expandable if the following two conditions are satisfied.

1. Every expanded prediction error $\epsilon_{i,e}$ is still expandable and $|\epsilon_i| < T/2$.
2. Each prediction error ϵ_i is calculated using the previously formatted four pixels without loss of prediction in JPEG-LS. If some of the four pixels are in the set of m pixels, their prediction errors are expanded by doubling the values.

The group of absolutely expandable sets is called AE.

Definition 8: The set of m pixels x_i, $(0 \leq i \leq m-1)$ is said to be absolutely changeable if the following two conditions are satisfied.

1. Every ϵ_i is changeable and $|\epsilon_i| \geq 2T$ is satisfied.

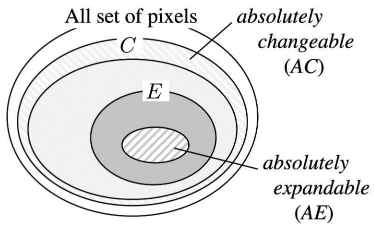


Fig. 4 The organization of the sets E, C, AE, and AC.

2. Each prediction error ϵ_i is calculated using the previously formatted four pixels without loss of prediction in JPEG-LS.

The group of absolutely changeable sets is called AC.

Figure 4 illustrates the organization of the pixel groups in an image. The location map corresponding to the positions of *absolutely expandable* and *absolutely changeable* is not required to recover the original image. Therefore, such redundancy is contained in the original location map.

For the reduction of the size of location map, the information about AE and AC is modified uniformly in [7], which is aimed to enhance the performance of the compression algorithm JBIG2. In the scheme, the location map is composed by the following rules,

$$L_{i,j} = \begin{cases} * & \text{if } x_{i,j} \in AE \cup AC \\ 0 & \text{if } x_{i,j} \in E \setminus AE \\ 1 & \text{otherwise}, \end{cases} \tag{27}$$

where the symbol "*" indicates "0" or "1" dependent on the contexts. And if the symbol is "*," then $L_{i,j}$ is uniformly changed to "0" or "1" because such an operation may enhance the performance of JBIG2. Although the simulation results in [7] showed a better performance compared with the Alattar's method [5], by carefully examining the condition in the simulation, the capacity is not so improved.

Different from the above operation, we exclude the information about AE and AC from the original location map, which implies to extract the essential information from the location map. Such information is concerning a group $C \setminus (AC \cup AE)$.

By introducing the Definition 7 and 8, several pixels in E and $C \setminus E$ can be classified into extra groups, AE and AC. Although each pixel $x_{i,j}$ has a corresponding location map $L_{i,j}$ in the basic scheme, the map in pixels judged AE and AC can be omitted, which contributes on the reduction of the size of $len(\boldsymbol{B})$. Since the modified location map is not a binary image, the adaption of JBIG2 is difficult. Instead, we compress the modified map by LZ77 type compression algorithm. It is noted that the number of pixels in AE is increased if a threshold T is increased, but that of AC is decreased, hence there is a trade-off.

Remark 2: The definitions of absolutely expandable and absolutely changeable are easily applied for the conventional schemes. The adaption of such a definition will improve the capacity-distortion behavior for the difference expansion methods.

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 AE or AC before a recovery operation is performed. If $\bar{x}_{i,j}$ is expandable and $|\bar{\epsilon}_{i,j}|$ is less than T, $x_{i,j}$ must be absolutely expandable. And if $|\bar{\epsilon}_{i,j}|$ is more than $2T$, then $x_{i,j}$ must be absolutely changeable. For such cases, the recovered location map from the extracted $\boldsymbol{w}$ is not required for the recovery of the current pixel $\bar{x}_{i,j}$. 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***.

4.3 Selection of Embedding Positions

The distortions caused by the embedding operation is basically not decreased by the reduction of the size of location map, thus, it is difficult to achieve a watermarked image preserving high quality. One of the factors of the distortions is to modify the sets in $C \setminus E$ based on Eq. (5). It is desirable not to modify the sets in $C \setminus E$ since it contributes not the increase of capacity, but on the decrease of the image quality. However, considering the invertibility such a modification seems to be required.

In the embedding operation, the embedding information $\boldsymbol{w}$ is inserted in the LSBs of the differences of all set in C. Since C is stable against the embedding operation, the embedding positions in a watermarked image is uniformly determined. It is why the sets in $C \setminus E$ are modified in the embedding operation. Here, it is remarkable that the sets in AC can be excluded from C without loss of invertibility. Because both the embedding and the extraction operations can determine if a set in C is absolutely changeable. By excluding such sets from the modification like Eqs. (5) and (12), the distortions in the corresponding positions in the conventional scheme can be removed, which improves the quality of the watermarked image.

Let N_{AC} be the number of sets in AC and P_w be the energy of the distortions caused by embedding. In the conventional scheme, LSBs of differences in AC, which are not changed in our method, are changed randomly by the embedding operation. Thus, the expected number of changes is $N_{AC}/2$. Since the amount of changes in the difference at each difference is only 1, its energy is also 1. Hence the expected energy values of our improved method is given as follows.

$$E(P'_w) = P_w - \frac{N_{AC}}{2} \tag{28}$$

There is a trade-off between N_{AC} and a threshold T. If T becomes large, N_{AC} is decreased because of the inequity $|\epsilon| \geq 2T$ in the Definition 8. Here, along with the size of T, the number of sets in E is increased and the amount of changes in an image exponentially increased. It is clear that the increase ratio of P_w is much larger than the decrease ratio of N_{AC}. Therefore, the improved method effectively works while T is small. The improved method is also applicable to the conventional difference expansion methods.

5. Experimental Results

We have implemented our algorithm and estimated the capacity and distortions using the RGB color images, "Lena," "baboon," "fruits," and "F16" with 512×512 pixels. We tested the algorithm for each RGB color components (the maximum capacity is 3 [bbp]) to evaluate the capacity-distortion behavior. Although the modified values of pixels are related to a threshold T, we evaluate the performance from the viewpoint of capacity-distortion behavior instead of showing the value of 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. 5, where the vector is selected in a raster scan order template shown in Fig. 3(a). 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. 6. It is clear that the capacity is improved at all ranges 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. It is noted that there is a trade-off between the size of m and the capacity. The increase of m results in the reduction of the size of location map by a factor of m, but the number of sets in E is decreased because all pixels in the set must be determined prediction error expandable. Moreover, the computational complexity of the determination is increased by a factor of m. If one wants to get better performance for overall range, we recommend to use $m = 4$ because of the trade-off. In the rest of this section, the results are obtained for the constant parameter $m = 4$.

In order to estimate the improvement of capacity of the proposed three methods; one removes the information about AE and AC from the original location map, one excludes the set in AC, and the other uses zigzag order for the embedding operation. Figure 7 shows the results, where "gzip," "AC," and "zigzag" stand for the three methods, respectively. For the comparison of the the size of location map, $len(\boldsymbol{L})$ rational to $len(\boldsymbol{W})$ is estimated. $len(\boldsymbol{L})/len(\boldsymbol{W})$ becomes larger according to the increase of PSNR, which is because of the capacity-distortion behavior, and it is better to keep the size small. As shown in Fig. 8, the combination of our three methods outperforms the other improved methods.

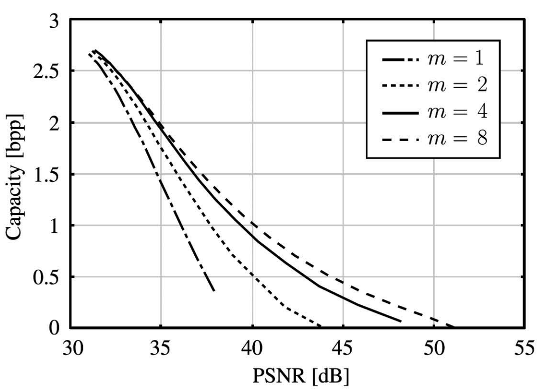


Fig. 5 The comparison of the performance for the number of element of the composed vector for an image "lena."

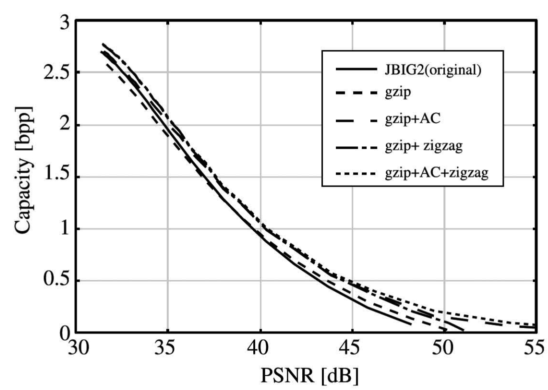


Fig. 7 The capacity-distortion behavior of prediction error based scheme for an image "lena."

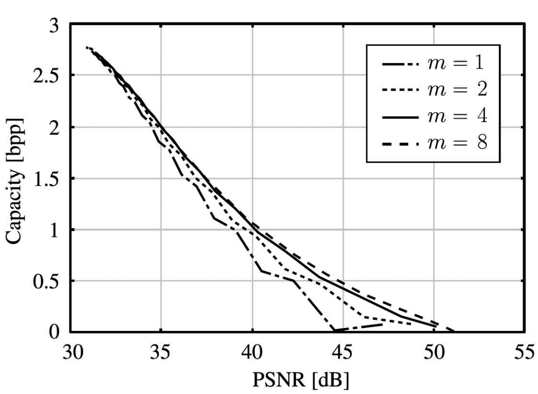


Fig. 6 The capacity improved by the modification of a location map for an image "lena."

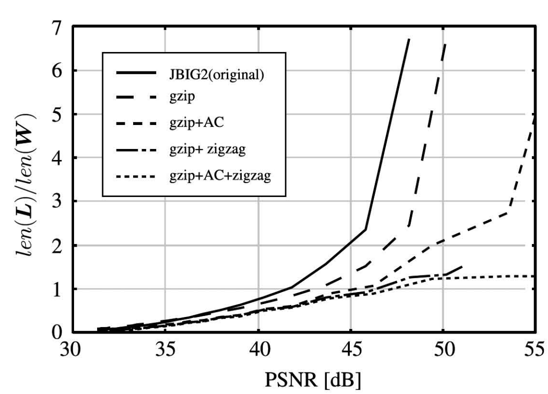


Fig. 8 $len(\boldsymbol{L})/len(\boldsymbol{W})$ versus PSNR of prediction error based scheme for an image "lena."

Fig. 9 Original image.

Fig. 10 Watermarked image (PSNR 31.3 [dB], capacity 2.79 [bpp]).

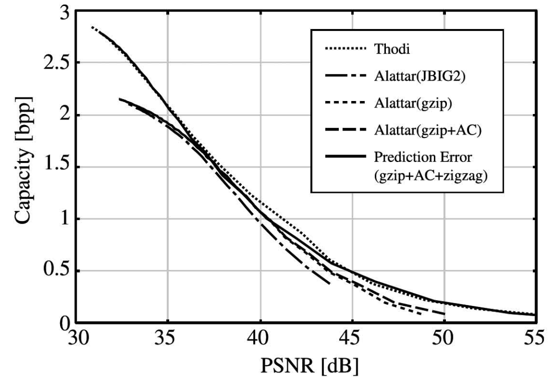


Fig. 11 The comparison of the capacity-distortion behavior for an image "lena."

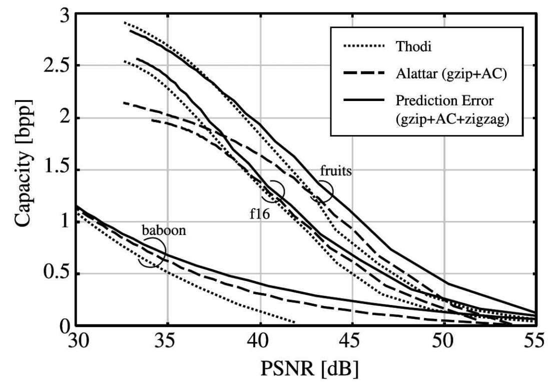


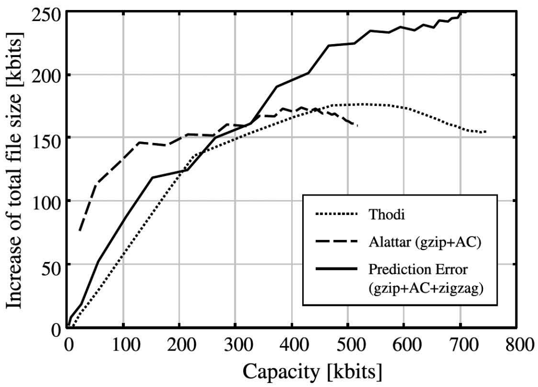
Fig. 12 The capacity-distortion behavior for each images.

When a watermark is embedded, a kind of sharpening effects appears and the effects grow stronger for the increase of the amount of watermark information. For the evaluation of the perceptual degradation, an original image "lena" and the watermarked image are shown in Fig. 9 and Fig. 10, respectively. By comparing these images, the perceptual degradation is not so serious.

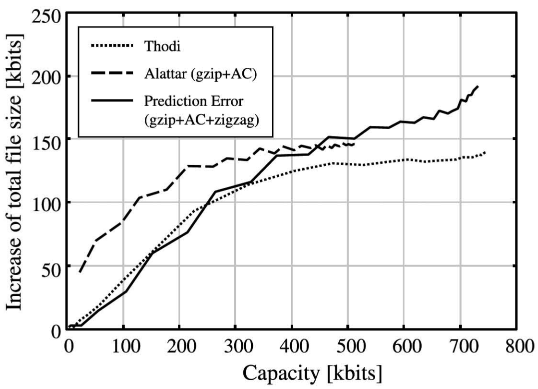
The improved methods that reduce the redundancy from the location map and the embedding positions are also applicable to Alattar's scheme [5] under the following conditions. A spatial quad was assembled from 2×2 adjacent pixels in the same color component, and the algorithm are applied to each color component independently. On the Alattar's scheme, the original embedding procedure that a location map is compressed by JBIG2 is performed for vectors of 2×2 adjacent pixels. By applying our improved method to Alattar's scheme, the vectors are classified into $AE \cup AC$, $E \setminus AE$, and the others, and are compressed by "gzip." For the classification of the case such that the set in AC is excluded, we represent those method by "Alattar(JBIG2)," "Alattar(gzip)," "Alattar(gzip+AC)," respectively.

For the comparison with the conventional schemes, the performance of our improved methods, Alattar's method [5], and Thodi's method [9] is shown in Fig. 11, where the tested image is "lena." It indicates that the capacity at low distortions is not dramatically increased for Alattar's scheme. Although the performance of Thodi's method is similar to that of our method, the comparison of the performance for other images which results are plotted in Fig. 12 show the superiority of our method. When a noisy image like "baboon" is used, the performance at low distortions is not so improved. It is because of the difficulty of prediction in noisy regions. In spite of the characteristic, our scheme can embed more information than the improved Alattar's method. The performance of the other two standard images also outperforms the improved Alattar's method. Although Thodi's scheme partially shows better score for images "lena" and "fruits," the capacity at low distortion is lower than our scheme.

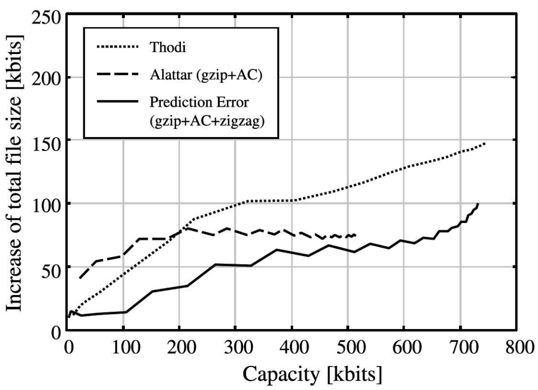
One of the important properties in reversible watermarking is the total file size after the lossless compression. Since the file size of raw file format is very large, it is not realistic to transmit the file directly. In order to preserve the embedded information, only lossless compression algorithm is applicable. Here, we calculate the file size of the compressed and watermarked image, and compare it with that of an compressed original image and watermark information that is attached outside of the image. For the evaluation, we apply three standard lossless compression algorithms,



(a) JPEG-LS



(b) JPEG2000



(c) PNG

Fig. 13 The increase of total file size after lossless compression; JPEG-LS, JPEG2000, and PNG.

JPEG-LS, JPEG2000, and PNG, which results are shown in Fig. 13. The figure confirms that our scheme outperforms the Alattar's one, and the reversible embedding operation performs well for the lossless compressions, JPEG2000 and PNG. Since our predictor is that of JPEG-LS, the increase of total file size becomes more than 0. Although the result of JPEG-LS is lower compared with Thodi's method, it is certain that the other predictor is applied for the reversible embedding operation in our scheme, the file size will be less than the conventional ones.

Since the reversible watermark technique is a kind of fragile watermark, one of the important applications is the detection of alteration. In the current form, the embedded information is easily extracted from the watermarked image. If the embedding information is encrypted using secure cryptographic tools, the secrecy of the watermark can be protected. Regretfully, it is not possible to apply for the steganography because the statistical distribution of elements such as differences of pixels and prediction errors, are modified for embedding.

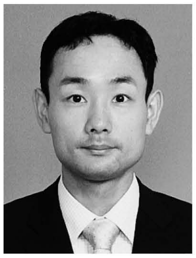
6. Conclusion

In this paper, a reversible watermark with a very large capacity based on the prediction errors calculated in JPEG-LS has been proposed. In order to improve the capacity, we have introduced a vector which is composed of several pixels and the size of the location map is decreased effectively exploiting the characteristic of a pixel/vector. Then, considering the properties of the predictor, the best composition of the vector is designed when the number of pixels is 4. The main contribution is to introduce the new definitions, *absolutely expandable* and *absolutely changeable*, which make it possible to exclude the redundancy of location map. In addition, we have proposed an improved method that keeps the distortion low when embedding small messages by excluding the sets of pixels which are determined *absolutely changeable*. Such improved methods can be applicable to the reversible watermark schemes based on difference expansion. The performance is evaluated in our simulation, and its results show that our scheme outperforms the improved Alattar's one.

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.
[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] L. Kamstra and H. Heijmans, "Reversible data embedding into images using wavelet techniques and sorting," IEEE Trans. Image Process., vol.14, no.12, pp.2082–2090, 2006.
[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] M. Kuribayashi, M. Morii, and H. Tanaka, "Reversible watermark with large capacity using the predictive coding," Proc. ICICS'05, LNCS 3783, pp.441–453, Springer-Verlag, 2005.
[8] G. Xuan, C. Yang, Y. Zhen, Y.Q. Shi, and Z. Ni, "Reversible data hiding using integer wavelet transform and companding technique," Proc. IWDW 2004, LNCS 3304, pp.115–124, Springer-Verlag, 2005.
[9] D.M. Thodi and J.J. Rodriguez, "Expansion embedding techniques for reversible watermarking," IEEE Trans. Image Process., vol.16, no.3, pp.723–730, 2007.
[10] A. Netravali and J.O. Limb, "Picture coding: A review," Proc. IEEE, vol.68, pp.366–406, 1980.

[11] S.W. Golomb, "Run-length encodings," IEEE Trans. Inf. Theory, vol.IT-12, no.3, pp.399–401, 1966.



Minoru Kuribayashi received the B.E., M.E., and D.E degrees from Kobe University, Kobe, Japan, in 1999, 2001, and 2004 respectively. From 2002 to 2007, he was a Research Associate in the Department of Electrical and Electronics Engineering, Kobe University. Since 2007, he has been a Assistant Professor at Division of Electrical and Electronics Engineering, Kobe University. His research interests are in digital watermark, information security, cryptography, and coding theory. He is a member of IEEE.

Masakatu Morii received the B.E. degree in electrical engineering and the M.E. degree in electronics engineering from Saga University, Saga, Japan, and the D.E. degree in communication engineering from Osaka University, Osaka, Japan, in 1983, 1985, and 1989, respectively. From 1989 to 1990 he was an Instructor in the Department of Electronics and Information Science, Kyoto Institute of Technology. From 1990 to 1995 he was an Associate Professor at the Department of Computer Science, Faculty of Engineering at Ehime University. From 1995 to 2005, he was a Professor at Department of Intelligent Systems and Information Science, Faculty of Engineering, University of Tokushima. Since 2005, he has been a Professor at the Department of Electrical and Electronics Engineering, Kobe University. His research interests are in error correcting codes, cryptography, discrete mathematics and computer networks. He is a member of the IEEE, the Japan Society for Industrial and Applied Mathematics, the Information Processing Society of Japan and the Society of Information Theory and Its Applications.

Hatsukazu Tanaka received the B.E. degree from Kobe University, Kobe, Japan in 1964, the M.E. degree in 1966, and the D.E. degree in 1969, both from Osaka University, Osaka, Japan. He joined the Faculty of Engineering, University of Osaka Prefecture as a Research Associate in 1969. In 1973 he was appointed as an Associate Professor in the Department of Electrical Engineering, Kobe University. From 1988 through 2004 he was a Professor in the Department of Electrical and Electronics Engineering, Kobe University. Since 2005 he has been a Professor Emeritus at Kobe University and the President of Kobe Institute of Computing (Graduate School of Information Technology). From 1980 through 1981 he was a member of the Communication Group of the University of Toronto, Toronto, Ontario, Canada, as a Visiting Scientist. His main work is on the basic theory of Information Engineering such as Information Theory, Coding Theory, Cryptography and Information Security, Image Processing, etc. Dr. Tanaka is a Life Fellow of IEEE, and a member of IACR.