Journal of Signal and Information Processing, **2013, 4, 259-273**
http://dx.doi.org/10.4236/jsip.2013.43034 Published Online August 2013 (http://www.scirp.org/journal/jsip)

Spatial Image Watermarking by Error-Correction Coding in Gray Codes

Tadahiko Kimoto

Faculty of Science and Engineering, Toyo University, Kawagoe, Japan.
Email: kimoto@toyo.jp

Received May 8th, 2013; revised June 8th, 2013; accepted July 8th, 2013

ABSTRACT

In this paper, error-correction coding (ECC) in Gray codes is considered and its performance in the protecting of spatial image watermarks against lossy data compression is demonstrated. For this purpose, the differences between bit patterns of two Gray codewords are analyzed in detail. On the basis of the properties, a method for encoding watermark bits in the Gray codewords that represent signal levels by a single-error-correcting (SEC) code is developed, which is referred to as the Gray-ECC method in this paper. The two codewords of the SEC code corresponding to respective watermark bits are determined so as to minimize the expected amount of distortion caused by the watermark embedding. The stochastic analyses show that an error-correcting capacity of the Gray-ECC method is superior to that of the ECC in natural binary codes for changes in signal codewords. Experiments of the Gray-ECC method were conducted on 8-bit monochrome images to evaluate both the features of watermarked images and the performance of robustness for image distortion resulting from the JPEG DCT-baseline coding scheme. The results demonstrate that, compared with a conventional averaging-based method, the Gray-ECC method yields watermarked images with less amount of signal distortion and also makes the watermark comparably robust for lossy data compression.

Keywords: Gray Code; Error-Correcting Code; Digital Watermark; Spatial Domain; JPEG DCT-Based Compression

1. Introduction

Digital watermarking is a technique for uniting digital signals and additional data, which are commonly called watermarks, so as to make changes in the signals imperceptible [1]. As for image signals, a watermark is embedded into a source image in a manner such that the resulting degradation of image quality is hardly noticeable. The watermark is to be recovered afterward for verifying the ownership of the image, for instance. From the viewpoint of signal domains to directly modify for expressing watermarks, the watermarking techniques are mainly divided into two categories: spatial-domain-based techniques and transform-domain-based ones.

In transform-domain-based image watermarking, pixel values of a source image, which are expressed in quantized signal levels, are first transformed into frequencies using a signal transformation such as the discrete cosine transform (DCT). These frequencies are, then, modified so as to include a watermark. Lastly, a watermarked image is produced from the modified frequencies by the inverse transformation and re-quantization. The changes in frequencies made by the watermark are inversely transformed to the changes in pixel values and thereby, they are dispersed over the watermarked image in such a way that the image distortion can be less conspicuous. Quantizing errors, however, generally result from the re-quantization of the inverse transforms. These errors disturb the transform domain as a result. Hence, the watermark essentially needs to be embedded with redundancies in the transform domain. These redundancies can also lead to some extent of robustness of the watermarks against signal changes in the spatial domain.

In spatial-domain-based image watermarking, pixel values are directly modified so as to include a watermark. A well-known conventional method for this processing is to replace some bits of pixel values, which are assumed to be expressed in binary notation, with watermark bits. Accurate watermarks can thus be embedded without any such redundancies as that essential to the transform-domain-based watermarking schemes. Such changes in pixel values, however, directly distort the source image. In addition, the watermarked image generally has weak robustness to subsequent signal distortions because even a

small change in a pixel value is likely to cause a significant change in the binary representation [2]. However, spatial-domain watermarks are nevertheless used in a number of applications such as steganography and fragile watermarking for content authentication.

As regards today's digital systems, data are usually compressed for transmission, storage, etc. When a watermarked image is transmitted through an image communication system, the image is usually compressed by, for example, the JPEG coding scheme before transmission. Hence, when the image is recovered at the receiver, it is distorted due to the lossy data compression. Because such lossy data compression is a kind of usual processing in communication systems, if watermarking is carried out before transmission, the watermark must be robust against at least the distortion caused by a data compression method being in use.

Wang *et al*. proposed the pixel surrounding (PS) method as a spatial-domain-based image watermarking method, and reported that the watermarks embedded by the method can survive JPEG lossy compression [3]. The PS method reduces effects of signal distortions on the embedded watermark by averaging pixel values, which is a kind of low pass filtering. On the other hand, error correcting techniques in channel coding such as the Bose-Chaudhuri-Hockquenghem (BCH) codes [4] can be used to improve robustness of spatial-domain watermarks [5]. A number of the related coding methods were studied in the late 1990's [2]. Most of such error-correction coding (ECC) methods have been applied to transform-domain watermarks [6].

With regard to signal bits, a quantized signal level is generally represented by a natural binary code, which is one of equal-length codes. Another equal-length code is the Gray code (also called the reflected binary code). This code has the property that any two adjacent codewords differ in only one bit position. Thus, small changes in a signal level are unlikely to affect all the bits of the codeword [7]. Accordingly, if the number of the changed bits is limited, they can be corrected by using error-correcting codes. In this paper, we consider watermarking by using both the Gray code to express the spatial domain and an error-correcting code to achieve robustness to signal changes. The objective of robustness here is to survive ordinary data compression such as the JPEG lossy coding.

The rest of this paper is organized as follows. Section 2 describes the codeword properties of Gray code. In the section, first, a single bit position where two adjacent codewords differ is given as a function of a signal level. Then, for each difference signal-level, the distribution of bit positions where two codewords just apart by the level from each other differ is demonstrated and compared with that in the case of the natural binary code. After formalizing the watermarking of signal codewords, Section 3 presents a scheme of error-correction coding in the Gray code representation. Based on the properties of the Gray code described in the preceding section, signal codewords involving single-error-correcting codewords are defined. The performance of error correction in the change of signal levels is verified by stochastic analyses. In Section 4, results of some experiments conducted on test images are presented to demonstrate performance of the proposed method, referred to as the Gray-ECC method, by comparison with the PS method. To begin with, the procedures for implementing the Gray-ECC method in image watermarking are defined. In addition, with regard to the PS method being compared, the implementing procedures are explained on referring to [3], and its properties depending on the parameter values are discussed to make a reasonable comparison with the ECC method. Then, both the image quality and the capability of robustness to JPEG lossy compression are evaluated from the experimental results. Section 5 concludes the paper.

2. Properties of Gray Code

2.1. Difference between Two Adjacent Codewords

First of all, we define the notation used in this paper. An m-bit Gray code is transformed from the natural binary code of the same length. Let R_m denote the decimal range of m-bit quantized levels $[0, 2^m - 1]$, and let C_R generally represent an m-bit binary code that consists of 2^m codewords corresponding to respective levels in R_m. The m-bit natural binary codeword $(b_{m-1} \cdots b_1 b_0)$ for a level $v \in R_m$ is just the binary representation of v; that is,

$$v = \sum_{i=0}^{m-1} b_i 2^i \tag{1}$$

where b_i is either 0 or 1 for $i = 0,1,\cdots,m-1$. Let C_{RN} denote the m-bit natural binary code and $C_{RN}(v)$ be the codeword in C_{RN} associated with a level $v \in R_m$. The m-bit Gray codeword $(g_{m-1} \cdots g_1 g_0)$ for a level v is calculated from the $C_{RN}(v)$ by the relation

$$g_i = b_i \oplus b_{i+1} \tag{2}$$

for $i = 0,1,\cdots,m-1$. In Equation (2), $\oplus$ denotes the exclusive OR operation, and b_m that the equation needs is temporary here and supposed to be 0. Let C_{RG} denote the m-bit Gray code and $C_{RG}(v)$ be the codeword in C_{RG} associated with a level $v \in R_m$.

A bit position where two adjacent Gray codewords differ depends on the levels that the codewords represent. For a level v ($0 \le v \le 2^m - 2$), consider the codewords

$C_{RN}(v)=(b_{m-1}\cdots b_1b_0)$ and $C_{RG}(v)=(g_{m-1}\cdots g_1g_0)$, and the respective succeeding codewords $C_{RN}(v+1)=(b'_{m-1}\cdots b'_1b'_0)$ and $C_{RG}(v)=(g'_{m-1}\cdots g'_1g'_0)$. Then, the difference between $C_{RG}(v)$ and $C_{RG}(v+1)$ is expressed as the bit sequence $(\gamma_{m-1}\cdots\gamma_1\gamma_0)$ that, by using Equation (2) with the associative and commutative laws of the exclusive OR operation, is given by the relations

$$\gamma_i = g_i \oplus g'_i = (b_i \oplus b'_i)\oplus(b_{i+1}\oplus b'_{i+1}) \tag{3}$$

for $i=0,1,\cdots,m-1$. From another viewpoint, the least significant bits of any $C_{RN}(v)$ can be expressed as $b_Jb_{J-1}\ldots b_0=01\cdots1$; that is, a certain $J\geq 0$ exists, and $b_J=0$ and $b_\ell=1$ for $\ell=J-1,J-2,\ldots,0$. Here, we define $J=0$ if v is even. Note that the corresponding bits in $C_{RN}(v+1)$ are $b'_Jb'_{J-1}\cdots b'_0=10\cdots0$. Then,

$$b_j\oplus b'_j=\begin{cases}1, & 0\leq j\leq J\\ 0, & J+1\leq j\leq m-1\end{cases}. \tag{4}$$

Using this relation in Equation (3) yields

$$\gamma_i=\begin{cases}1, & i=J\\ 0, & i\neq J\end{cases} \tag{5}$$

for $i=0,1,\ldots,m-1$. Thus, $C_{RG}(v)$ and $C_{RG}(v+1)$ differ in only one bit γ_J, and J depends on the bit pattern of $C_{RN}(v)$ or equivalently, on the value of v. Consequently, the bit position J where $C_{RG}(v)$ and $C_{RG}(v+1)$ differ is given by

$$J=\begin{cases}0, & \text{if } v \text{ is even}\\ \max\left\{j \mid v+1\propto 2^j\right\}, & \text{if } v \text{ is odd}\end{cases}. \tag{6}$$

In addition, it is found that $J\geq 1$ if v is odd.

2.2. Difference between Two Codewords

The difference between $C_{RG}(v)$ and $C_{RG}(v+d)$, where d (≥1) is a difference signal-level, is given by the result of accumulating bit changes from $C_{RG}(v)$ through $C_{RG}(v+d)$. Hence, these two codewords differ in those bit positions where the bit inversion occurs once or odd times during the accumulation. For $d\geq 2$, thus, one or more bits differ between the codewords, and the loca- tions of difference bits are determined from the starting level v. Consequently, the distribution of difference bits in the entire m-bit codeword for all v's that satisfy $v+d\in R_m$ is derived using Equation (6).

Let m = 8 as a practical example. For d = 1, 128 even values of 255 v's yield the difference bit g_0's between $C_{RG}(v)$ and $C_{RG}(v+d)$. Out of the 127 possible codeword pairs of odd v, 64 pairs have the difference bit at g_1, 32 pairs at g_2, 16 pairs at g_3, 8 pairs at g_4, 4 pairs at g_5, 2 pairs at g_6, and one pair at g_7. For d = 2, if v is even, then v + 1 is odd, and vice versa. As a result, any codeword pair has a difference bit g_0 and another one; out of 254 possible codeword pairs, 128 pairs have the different g_1, 64 pairs g_2, 32 pairs g_3, 16 pairs g_4, 8 pairs g_5, 4 pairs g_6, and 2 pairs g_7. For d = 3, in the case where v is even, in any one of 127 possible codeword pairs, the g_0's are same because v + 2 is also even, and the bits are different in one of the other seven bit-positions. For the other 126 pairs where v is odd, two codewords differ in three bit-positions composed of g_0, g_1 and one of the other six positions. **Figure 1** illustrates the distributions of the difference bits in $(g_7\cdots g_1g_0)$ for each value of d from 1 to 8, for instance.

2.3. Comparison with Natural Binary Code

As for m-bit natural binary representations, the difference between two codewords is analyzed as below. The number, μ, of bit positions where two adjacent codewords differ ranges from 1 to m. For each μ, there are $2^{m-\mu}$ possible codeword pairs, and any of the pairs differs in the bit positions from 0 through $\mu-1$. For a difference signal-level d = 2, the number, also denoted by μ, of bit positions where two codewords $C_{RN}(v)$ and $C_{RN}(v+2)$ differ ranges from 1 to m − 1. There are $2^{m-\mu}$ possible codeword pairs for each μ, and any of the pairs differs in the bit positions from 1 through μ while the b_0's are always the same. For a difference signal-level d = 3, the number of bit positions where two codewords $C_{RN}(v)$ and $C_{RN}(v+3)$ differ ranges from 2 to m. The difference between the codewords gets more complicated than that for the above d = 2 or d = 1, while the b_0's are always different.

As the above examples demonstrate, for a given difference signal-level d, with regard to the pairs of codewords whose levels are apart by d from each other, the number of bit positions where two Gray codewords differ lies in a narrower range than that of bit positions where two natural binary codewords differ.

3. Error-Correcting in Gray Code

3.1. Formalization of Watermarking Codewords

The process of watermarking signals in code representations can be expressed in a generalized form as below.

1) Embedding process

In addition to the notation used in the preceding section, we define the inverse mapping from C_R to R_m, that

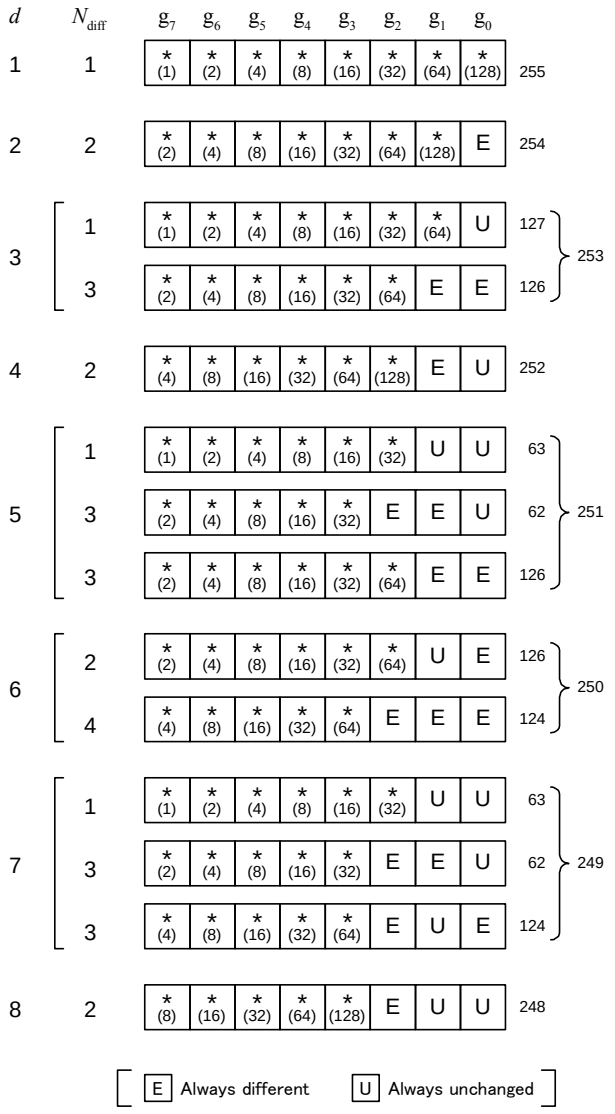

Figure 1. Distributions of difference bits between two 8-bit Gray codewords apart by a difference signal-level d from each other, for $d = 1, 2, \ldots, 8$: N_{diff} denotes the number of difference bits occurring in a codeword; the numbers written in the right side denote the total numbers of codeword pairs to which the bit distributions apply; the number written at each bit position in parentheses denotes the number of codeword pairs which differ at the position; the asterisks indicate those bit positions only one of which differs at the same time.

is, $C_R^{-1}(\boldsymbol{c}) \in R_m$ for $\boldsymbol{c} \in C_R$ by

$$C_R^{-1}(\boldsymbol{c}) = \left\{ v \,\middle|\, \boldsymbol{c} = C_R(v), v \in R_m \right\} \tag{7}$$

where a unique v is determined for each $\boldsymbol{c}$ since the code C_R is a one-to-one mapping. As for a watermarking algorithm to be considered, let us regard both of the embedding of watermarks and the extracting of watermarks as kinds of point operations where the processing for any signal in a signal sequence depends only on the value of that signal. Also, for simplicity, we assume that one watermark bit is embedded in one chosen signal.

Suppose that a source signal sequence, X, and a watermark-bit sequence, W, are given. In the embedding process, first, the value v_x of a given signal x in X is mapped to the codeword $\boldsymbol{c}_x$ in C_R; that is,

$$\boldsymbol{c}_x = C_R(v_x). \tag{8}$$

Next, for a given watermark bit $w \in W$, a procedure, named EMBED, performs embedding w into $\boldsymbol{c}_x$; this is expressed in the form of a point operation such that

$$\boldsymbol{c}_y = \text{EMBED}(\boldsymbol{c}_x, w) \tag{9}$$

where $\boldsymbol{c}_y$ is the resulting codeword and $\boldsymbol{c}_y \in C_R$. The signal value, v_y, that $\boldsymbol{c}_y$ represents is given by

$$v_y = C_R^{-1}(\boldsymbol{c}_y), \tag{10}$$

and the source signal x is replaced with the signal of the level v_y. Let Y denote the entire sequence of the replaced signals. Thus, the signal sequence X is changed to Y with the watermark W embedded there. The difference of the signal values v_x and v_y, which is generally referred to as an embedding distortion, degrades the signal quality of Y from X.

2) Extracting process

Assume that, undergoing some signal processing such as data compression, the above sequence Y changes to another one, Z, of different signal values. In the watermark recovering process, a procedure for extracting watermark bits from codewords in C_R is defined as the following inverse process of Equation (9): Referring to the procedure as EXTRACT, we can write it in the form of a point operation; that is, using the notation of Equation (9),

$$w = \text{EXTRACT}(\boldsymbol{c}_y). \tag{11}$$

To extract the watermark from Z using this procedure, first, the value, v_z, of each signal in Z is mapped to the codeword, $\boldsymbol{c}_z$, in C_R; that is, similarly to $\boldsymbol{c}_x$ of Equation (8),

$$\boldsymbol{c}_z = C_R(v_z). \tag{12}$$

Then, the procedure performs on $\boldsymbol{c}_z$ as

$$w' = \text{EXTRACT}(\boldsymbol{c}_z) \tag{13}$$

where w' is the resulting bit.

Robust watermarking requires w' to be equal to w in spite of the change in signal level from v_y to v_z. However, if $v_z \neq v_y$, then obviously, $C_R(v_z) \neq C_R(v_y)$ because of the one-to-one mapping of the code. As a result, the change in codeword is likely to lessen the reliability of a watermark extracted from the resultant codeword.

It is clear from the above relations that the change in

codewords affects directly the extraction of watermark bits. This also implies that for the purpose of recovering watermarks, it is sufficient to recover at least those bits of a codeword which affect the embedded watermark.

3.2. Single-Error-Correcting in Gray Code

When a signal changes by one level, one bit of the corresponding Gray codeword changes. Accordingly, the changed bit can be corrected by means of a single-error-correcting (SEC) code. Then, we employ single-error-correction coding in watermarking to make embedded watermarks completely robust against a change by one level in any signal.

1) Encoding procedure

The processing of encoding watermark bits with a SEC code can be involved in the embedding operation EMBED of Equation (9). Let C_E denote an error-correcting code to be used, which is assumed to be a linear block code in this paper, and n_E be the code length of C_E. For a signal level v_x and a watermark bit w, given the corresponding codeword $\boldsymbol{c}_x = C_R(v_x)$, $\mathrm{EMBED}(\boldsymbol{c}_x, w)$ can be defined by the following two steps. Note that we still use a general notation of C_R as the code that represents signal levels.

(EN.1) Generate a codeword, $\boldsymbol{c}_w$, in C_E corresponding to the bit value of w.

(EN.2) Replace n_E consecutive bits of $\boldsymbol{c}_x$ from a specified bit position k (≥ 0) with all the bits of $\boldsymbol{c}_w$. Here, the bit position 0 designates the least significant bit (LSB). The resulting codeword is equivalent to $\boldsymbol{c}_y$ of Equation (9). Let us define a REPLACE operator to implement the operation of this step in the form

$$\boldsymbol{c}_y = \mathrm{REPLACE}(\boldsymbol{c}_x, \boldsymbol{c}_w, k). \tag{14}$$

In the above procedure, the codeword $\boldsymbol{c}_x$ is changed depending on both $\boldsymbol{c}_w$ and k. The change in codeword immediately affects the signal level and the resultant level v_y is determined. To reduce the embedding distortion, $v_y - v_x$, a code C_E of short codewords is desirable. A binary linear SEC code of the minimum code length is a (3, 1, 3) code consisting of two 3-bit codewords that are apart by the Hamming distance of 3 from each other and that correspond to respective values of an informa- tion bit 0 and 1. A typical (3, 1, 3) code is the so-called (3, 1) repetition code or also known as the general Hamming (3, 1) code [4]. Regarding each watermark bit as the information bit per codeword, we concentrate on a (3, 1, 3) code from now on.

2) Determining signal codewords

Given a source signal and a watermark bit, to reduce the embedding distortion, from those codewords which include the error-correcting codeword corresponding to the watermark bit in the same bit-position, the one that results in the smallest embedding distortion must be chosen as the encoded codeword of the signal. The problem of choosing such codewords can be simplified in the way below.

Suppose that a (3, 1, 3) code $C_E = \{\boldsymbol{c}_{E0}, \boldsymbol{c}_{E1}\}$ where the two codeword $\boldsymbol{c}_{E0}$ and $\boldsymbol{c}_{E1}$ correspond to respective watermark bits of 0 and 1 is given. Given a code C_R for representing signal levels, for each value of watermark bit, putting the corresponding error-correcting codeword into all the codewords of C_R by means of the REPLACE operator with a specified value of k, we obtain a subset-code of C_R, expressed in the form

$$\begin{aligned} &C_{W\,k,b} \\ &\triangleq \left\{\boldsymbol{c}' \mid \boldsymbol{c}' = \mathrm{REPLACE}(\boldsymbol{c}, \boldsymbol{c}_{E\,b}, k), \forall \boldsymbol{c} \in C_R\right\} \end{aligned} \tag{15}$$

for b = 0 and 1 respectively. From each subset-code, a set of all signal levels that the codewords of the subset-code represent is obtained such that

$$U_{k,b} \triangleq \left\{u \mid u = C_R^{-1}(\boldsymbol{c}), \forall \boldsymbol{c} \in C_{W\,k,b}\right\} \tag{16}$$

It follows that $U_{k,0}$ and $U_{k,1}$ are subsets of R_m. Thus, these subsets consist of signal values that each have had a watermark bit embedded by the error-correcting code.

For a given signal level $v_x \in R_m$ and a watermark bit $w \in \{0, 1\}$, we choose such a level, v_y, as closest to v_x from $U_{k,w}$. That is, v_y is determined so that

$$|v_y - v_x| = \min_{v \in U_{k,w}} \{|v - v_x|\}. \tag{17}$$

It can be proved that in the case that the Gray code C_{RG} is used as C_R, the unique v_y exists closest to each v_x in either $U_{k,w}$. Let us denote the choosing operation of Equation (17) by the operator ENCODE in the form

$$v_y = \mathrm{ENCODE}(v_x, w). \tag{18}$$

This operator involves $U_{k,0}$ and $U_{k,1}$ that both have been made from C_R and C_E by Equations (15) and (16). The signals of a sequence X are thus encoded one by one to the single-error-correctable values, which compose the watermarked sequence Y.

3) Determining SEC codewords

Since C_E affects $U_{k,0}$ and $U_{k,1}$ through Equations (15) and (16), embedding distortions also depend on the codewords comprised in C_E. On the assumptions that each level in R_m is equally likely to occur for any signal and also that a watermark bit is a random variable, for a given code $C_E = \{\boldsymbol{c}_{E0}, \boldsymbol{c}_{E1}\}$ and a bit-position parameter k, the amount of embedding distortion can be estimated by calculating the expected value of the mean squared error (MSE) between the source signal levels and the encoded signal levels, defined by

$$\mathrm{MSE} \triangleq \frac{1}{2}\left[\frac{1}{2^m}\sum_{v=0}^{2^m-1}\left\{v-\mathrm{ENCODE}(v,0)\right\}^2 + \frac{1}{2^m}\sum_{v=0}^{2^m-1}\left\{v-\mathrm{ENCODE}(v,1)\right\}^2\right]. \quad (19)$$

In fact, if C_R has a periodic structure in codeword bits, the above ranges of summation can be reduced.

There exist the following four kinds of pair of 3-bit codewords that are apart from each other by Hamming distance of 3: (i) {(000), (111)}, (ii) {(001), (110)}, (iii) {(011), (100)}, and (iv) {(010), (101)}. Under the condition of using the Gray code C_{RG} as C_R, the expected MSE values of Equation (19) were calculated not only for each of the four codeword pairs but also for each bit position k of 0 and 1, respectively. The results are listed in **Table 1**, together with the peak-signal-to-noise ratio (PSNR) values in decibels (dB) defined by

$$\mathrm{PSNR} \triangleq 10\cdot\log_{10}\frac{\left(2^m-1\right)^2}{\mathrm{MSE}}(\mathrm{dB}). \quad (20)$$

On the above assumptions for stochastic analysis, the codes (i) and (iii) have exactly the same MSE values, and so do the codes (ii) and (iv) for both the bit positions. As shown in **Table 1**, the codes (i) and (iii) yield less distortion than the codes (ii) and (iv). On the other hand, as shown later, all the four codeword pairs have almost the same performance in error correction. Consequently, the codeword pair (iv) has been selected as C_E for the Gray code; denoted by C_{EG}, that is, $C_{EG} = \{(010), (101)\}$. We refer to the encoding by the ENCODE operator with C_{EG} and the position parameter $k = 0$ as $\mathrm{ECC_{G0}}$ and that with C_{EG} and $k = 1$ as $\mathrm{ECC_{G1}}$. The encoding with the bit position of $k > 1$ is no longer practical for 8-bit signals due to large embedding distortions.

Table 1. Expected MSEs for (3, 1, 3) SEC codes embedded in Gray codes. (a) Bit position k = 0; (b) Bit position k = 1.

(a)

Pair	$\{c_{E0}, c_{E1}\}$		MSE	PSNR (dB)
(i)	000	111	12.5	37.2
(ii)	001	110	8.5	38.8
(iii)	011	100	12.5	37.2
(iv)	010	101	8.5	38.8

(b)

Pair	$\{c_{E0}, c_{E1}\}$		MSE	PSNR (dB)
(i)	000	111	44.7	31.6
(ii)	001	110	29.7	33.4
(iii)	011	100	44.7	31.6
(iv)	010	101	29.7	33.4

Figure 2 shows the relationship between source signal levels v_x and encoded signal levels v_y of Equation (18) in $\mathrm{ECC_{G0}}$ and $\mathrm{ECC_{G1}}$. The relationship in the interval of 16 source levels and that in the interval of 32 source levels repeat periodically over the 8-bit dynamic range for $\mathrm{ECC_{G0}}$ and $\mathrm{ECC_{G1}}$, respectively. In the period of $\mathrm{ECC_{G1}}$, the stepwise changes differ for the two codewords. With regard to embedding distortions, the magnitude of difference in signal levels caused by $\mathrm{ECC_{G0}}$ is less than or equal to 6, and that caused by $\mathrm{ECC_{G1}}$, less than or equal to 12.

4) Decoding procedure

Given a signal sequence Z that has come from Y described in Section 3.2 (2), the process for decoding signal values to extract watermark bits is described as follows. Assume that both the error-correcting code $C_E = \{\boldsymbol{c}_{E0}, \boldsymbol{c}_{E1}\}$ and the bit-position parameter k that have been used for Y are known to the decoder. Let v_z be the value of a signal z in Z, and let $\boldsymbol{c}_z$ denote $C_R(v_z)$.

(DE.1) Obtain a codeword, $\boldsymbol{e}$, from $\boldsymbol{c}_z$ by separating n_E

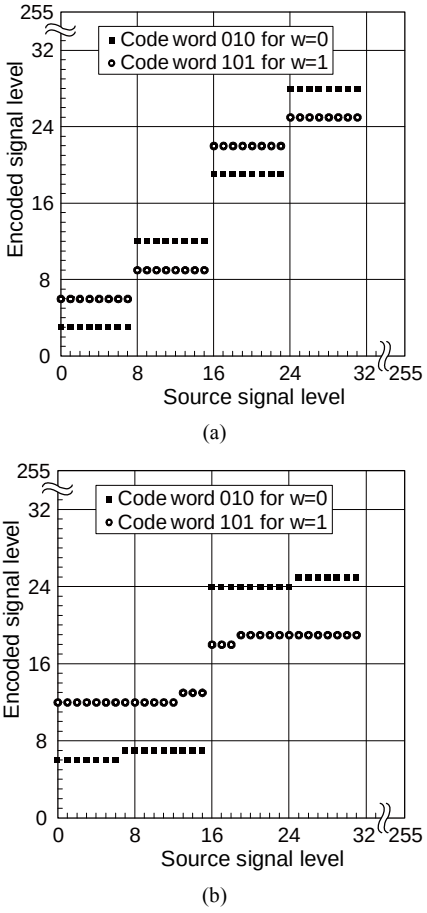

Figure 2. Source level-encoded level relationship of ECC methods. (a) $\mathrm{ECC_{G0}}$; (b) $\mathrm{ECC_{G1}}$.

consecutive bits starting at the bit position k. We define the operator SEPARATE that performs this bit-operation as

$$\boldsymbol{e} = \mathrm{SEPARATE}\left(\boldsymbol{c}_z, k, n_E\right) \tag{21}$$

where actually $n_E = 3$ for the (3, 1, 3) code C_E.

(DE.2) The codeword $\boldsymbol{e}$, which might have been distorted, is supposed to be decoded by using C_E. To implement the decoding, assuming all distorted codewords are equally likely, we use a maximum-likelihood decoding method [4]. By the method, $\boldsymbol{e}$ is mapped to one of the two codewords $\boldsymbol{c}_{E0}$ and $\boldsymbol{c}_{E1}$ such that is closer to $\boldsymbol{e}$ than the other in terms of the Hamming distance. Let $\boldsymbol{e}'$ be the resulting codeword.

(DE.3) The bit value, w', corresponding to $\boldsymbol{e}'$ in C_E is immediately obtained as the watermark bit extracted from v_z.

We express the procedure that performs the above three steps for a given v_z by the operator DECODE of the form

$$w' = \mathrm{DECODE}\left(v_z\right) \tag{22}$$

which is implicitly associated with C_E and k.

3.3. Stochastic Analysis of Error-Correcting Performance

Error-correcting performance of $\mathrm{ECC_{G0}}$ and $\mathrm{ECC_{G1}}$ are revealed by stochastic analysis below.

1) Error-correcting capabilities for a single level-change

For each encoded signal level $u \in U_{k,b}$ where k is given and $b \in \{0, 1\}$, given a difference signal-level d satisfying $u + d \in R_m$, whether the result of decoding u + d, that is, $b' = \mathrm{DECODE}\left(u + d\right)$ agrees with b or not is determined. The agreement between the two bits b and b' depends on the difference between the two error-correcting codewords, $\boldsymbol{c}_{Eb}$ of $C_{RG}\left(u\right)$ and $\boldsymbol{a}'$ of $C_{RG}\left(u + d\right)$ that is obtained by

$$\boldsymbol{a}' = \mathrm{SEPARATE}\left(C_{RG}\left(u + d\right), k, 3\right) \tag{23}$$

If the Hamming distance between $\boldsymbol{c}_{Eb}$ and $\boldsymbol{a}'$ is either 0 or 1, then $\boldsymbol{a}'$ is mapped to $\boldsymbol{c}_{Eb}$ in the decoding procedure and thus, b' agrees with b. Otherwise, $\boldsymbol{a}'$ is mapped to the other error-correcting codeword $\boldsymbol{c}_{E\bar{b}}$, where $\bar{b}$ denotes the 1's complement of b, and consequently, b' is regarded as $\bar{b}$.

Because the difference bits between two signal codewords $C_{RG}\left(u\right)$ and $C_{RG}\left(u + d\right)$ depend on the u in $U_{k,b}$, we define a ratio of the correctly decoding of signals for the entire $U_{k,b}$ as follows. First, the whole of those values of $U_{k,b}$ which are allowed to increase by d within R_m is described by a set, $U_{k,b}\left(d\right)$, such that

$$U_{k,b}\left(d\right) \triangleq \left\{u \mid u + d \in R_m, \forall u \in U_{k,b}\right\}. \tag{24}$$

Then, out of the values in $U_{k,b}\left(d\right)$, a set, $T_{k,b}\left(d\right)$, of those values which, after increasing by d, still contain correctly decodable error-correcting codewords is expressed in the form such that

$$T_{k,b}\left(d\right) \triangleq \left\{u \mid h\left(\boldsymbol{c}_{Eb}, \boldsymbol{a}'\right) \le 1, \forall u \in U_{k,b}\left(d\right)\right\} \tag{25}$$

where $\boldsymbol{a}'$ is the same as that of Equation (23) and $h\left(\boldsymbol{x}, \boldsymbol{y}\right)$ denotes the Hamming distance between two codewords $\boldsymbol{x}$ and $\boldsymbol{y}$. Supposing that signal values in $U_{k,b}$ are equally probable, we estimate the probability of the correctly decoding of $U_{k,b}$ for a variable d, denoted by $P_{k,b}\left(d\right)$, by the ratio

$$P_{k,b}\left(d\right) \triangleq \left\langle T_{k,b}\left(d\right)\right\rangle \Big/ \left\langle U_{k,b}\left(d\right)\right\rangle, \tag{26}$$

where $\left\langle S\right\rangle$ denotes the number of elements in a set S, for b = 0 and 1, and k = 0 and 1 according to Section 3.2 (3).

On the additional assumption that watermark bits are random variables, an estimate of the correctly decoding probability of $\mathrm{ECC_{G}}k$, for k = 0 and 1, can be defined as follows. For the definition, $P_{k,b}\left(d\right)$ and $P_{k,b}\left(-d\right)$ of the same d are put together by taking the average of the two values on the assumption that a signal change by d levels and that by $-d$ levels are equally likely to happen to any level in $U_{k,b}$. In fact, because of the symmetrical structure of Gray code, it holds that $U_{k,b}\left(d\right) = U_{k,b}\left(-d\right)$ and $T_{k,b}\left(d\right) = T_{k,b}\left(-d\right)$ for any possible d and each b. Then, the overall probability of correctly decoding by $\mathrm{ECC_{G}}k$ for a variable d, denoted by $P_{Gk}\left(d\right)$, is defined by

$$\begin{aligned} P_{Gk}\left(d\right) \triangleq \big\{&P_{k,0}\left(d\right) + P_{k,0}\left(-d\right) \\ &+ P_{k,1}\left(d\right) + P_{k,1}\left(-d\right)\big\}\big/4 \end{aligned}, \tag{27}$$

where $d > 0$.

As regards $\mathrm{ECC_{G0}}$, **Figure 3(a)** shows the variation of $P_{G0}\left(d\right)$ with d, where let $d \le 32$, for instance. In this case, $P_{0,b}\left(d\right) = P_{0,b}\left(-d\right)$, for b = 0 and 1, and for any possible d, and they have one of three values 0, about 0.5 and 1 if d is so small that $U_{0,b}\left(d\right)$, the denominator of Equation (26), is composed of a lot of elements. As shown in **Figure 3(a)**, $P_{G0}\left(d\right) > 0$ for any d, and $P_{G0}\left(d\right)$ has one of three values about 0.25, about 0.5, and 1. Particularly for $d = 1$, $P_{G0}\left(1\right) = 1$.

For comparison, we define a (3, 1, 3) code, denoted by C_{EN}, to use for the natural binary code C_{RN}. Two codewords in C_{EN} have been determined from MSE values similarly to those in C_{EG}; that is, $C_{EN} = \{(011), (100)\}$. We refer to the encoding by the ENCODE operator with C_{EN} as the natural ECC method, and let $\mathrm{ECC_{N}}k$ denote the natural ECC method with the position parameter k, where practically $k = 0$ or 1. Note that no operation is actually needed for the transformation between m-bit binary signal values and m-bit codewords in C_{RN}. Also, the expected probability of correctly decoding by $\mathrm{ECC_{N}}k$ for a difference signal-level d, denoted by $P_{N\,k}(d)$, is defined similarly to $P_{G\,k}(d)$, for $k = 0$ and 1 respectively. The variation of $P_{N\,0}(d)$ with d is added in **Figure 3(a)** by the broken line. For any possible d, each of $P_{0,\,b}(d)$ and $P_{0,\,b}(-d)$ has either 0 or 1, for $b = 0$ and 1, and consequently, $P_{N\,0}(d)$ has one of three values 0, 0.5 and 1. Particularly for $d = 1$, $P_{N\,0}(1) = 0.5$.

2) Error-correcting capabilities for multiple level-changes

The decoding properties are estimated under the condition that all the changes up to d levels occur in the encoded signals. For the estimation, supposing that all the difference signal-levels in the range from 1 to d are equally likely, we define the accumulative probability of correctly decoding by $\mathrm{ECC_{G}}k$, denoted by $\hat{P}_{G\,k}(d)$ in the form of a function of d, by averaging from $j = 1$ through d; that is,

$$\hat{P}_{G\,k}(d) \triangleq \sum_{j=1}^{d} P_{G\,k}(j) \Big/ d \,, \tag{28}$$

for $k = 0$ and 1 respectively. Similarly to $\hat{P}_{G\,k}(d)$, the accumulative probability of correctly decoding by $\mathrm{ECC_{N}}k$, denoted by $\hat{P}_{N\,k}(d)$, is defined for each $k = 0$ and 1.

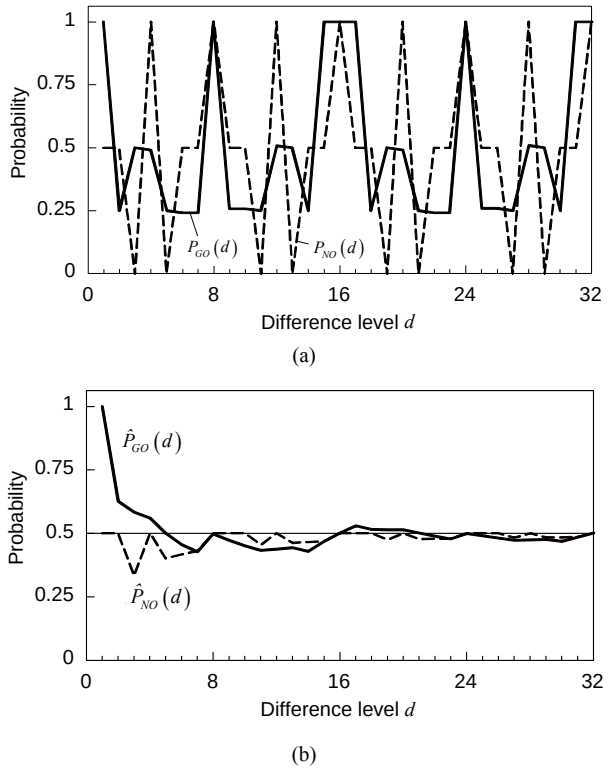

Figure 3. Performance of Gray-ECC in comparison with natural ECC for codeword position of 0 (k = 0). (a) Correctly decoding probabilities; (b) Accumulative probabilities.

The variation of $\hat{P}_{G\,0}(d)$ with d is shown in **Figure 3(b)**, where let $d \le 32$, with the variation of $\hat{P}_{N\,0}(d)$ depicted for comparison by the broken line. $\hat{P}_{G\,0}(d)$ is found to be greater than 0.5 for $d \le 4$. Hence, $\mathrm{ECC_{G0}}$ is expected to achieve the error correction in this range of d. As the figure shows, with increasing d, $\hat{P}_{G\,0}(d)$ gets close to 0.5. In contrast, $\hat{P}_{N\,0}(d) \le 0.5$ for any d. This result indicates that $\mathrm{ECC_{N0}}$ has no capacity of correcting errors for those signals which have undergone change by various levels.

Effects of selected codewords in C_{EG} on the error-correcting performance have been also evaluated. For this purpose, by replacing the codeword pair in C_{EG} with each one of the three other codeword pairs (i), (ii) and (iii) previously mentioned in Section 3.2 (3), both the corresponding $P_{G\,0}(d)$ and $\hat{P}_{G\,0}(d)$ were calculated. In comparison among the four codeword-pairs, the variations of these probabilities with d have been found to be almost the same. Consequently, the four codeword-pairs have no difference in error-correcting performance.

3) Bit position of error-correcting codewords

For the bit position $k = 1$, the performance of $\mathrm{ECC_{G1}}$ has been evaluated. The variation of with d is shown in **Figure 4(a)**, compared with that of $P_{N\,1}(d)$. In $\mathrm{ECC_{G1}}$, $P_{1,\,b}(d) = P_{1,\,b}(-d)$ for $b = 0$ and 1, and for any possible d; they have one of five values 0, about 1/4, about 1/2, about 3/4 and 1. As **Figure 4(a)** illustrates, accordingly, $P_{G\,1}(d)$ has one of five values about 1/4, about 3/8, about 1/2, about 5/8 and 1. Particularly for $d = 2$ as well as $d = 1$, $P_{G\,1}(d) = 1$. In addition, the variations of $\hat{P}_{G\,1}(d)$ and $\hat{P}_{N\,1}(d)$ with d are both depicted in **Figure 4(b)**. As this figure shows, $\hat{P}_{G\,1}(d)$ becomes greater than 0.5 for $d \le 11$. This range of d is much wider than that in $\mathrm{ECC_{G0}}$. **Figure 4** also indicates that the fact $P_{N\,1}(1) = 3/4$ results in that $\hat{P}_{N\,1}(d) > 0.5$ for $d \le 4$.

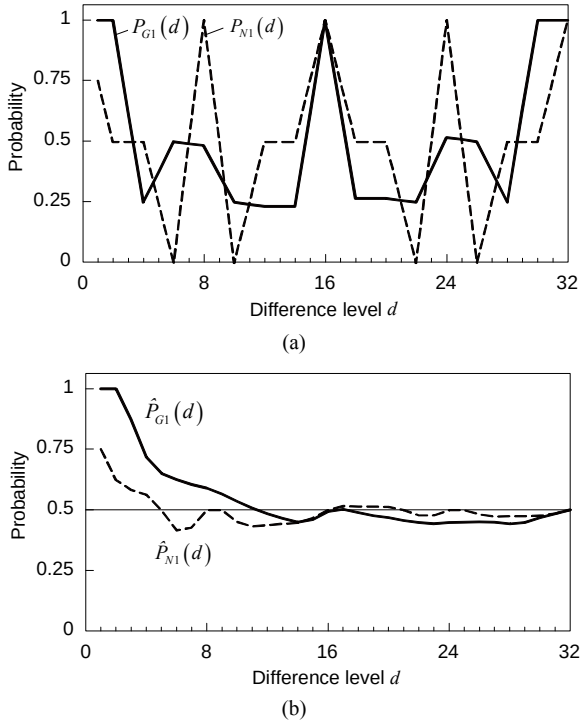

Figure 4. Performance of Gray-ECC in comparison with natural ECC for codeword position of 1 (k = 1). (a) Correctly decoding probabilities; (b) Accumulative probabilities.

Hence, $\mathrm{ECC_{N1}}$ is barely expected to achieve the error correction in this range of d.

From the above stochastic analyses, the advantages of the Gray code over the natural binary code in the ECC method are summarized as follows:

a) For any d, $P_{Gk}(d) > 0$ while $P_{Nk}(d) = 0$ for some of d's.

b) For a given k, $P_{Gk}(d) > P_{Nk}(d)$ for some of d's, in particular for $d = 1$.

These advantages result from the limited distribution of bit positions where two codewords differ as described in Section 2.2. Because of the advantages, $\mathrm{ECC_{Gk}}$ achieves the capability of correcting "errors", namely signal-level changes particularly if the changes are limited in a small range.

4. Experiments

4.1. Watermarking Procedures

The procedures for implementing the ECC method, which means both the Gray-ECC and the natural ECC by generalization, is introduced in this section. Source images are assumed to be monochrome images or signals of one channel of color images, and a signal level of each pixel is represented by an m-bit code in natural binary notation. Using a general notation, for an image f, let $f(x, y)$ denote a pixel, or equivalently a pixel value at the coordinates (x, y) on the image plane. A watermark is a sequence of bits, and the bits are supposed to be embedded into a source image one by one in order.

1) Embedding procedure

Given a source image, f_X, of the size N_X by N_Y pixels and a watermark-bit sequence $W = (w_0, w_1, \cdots, w_{N_W-1})$, where N_W is the number of watermark bits $(N_W \le N_X N_Y)$ and $w_i \in \{0, 1\}$ for $i = 0, 1, \cdots, N_W - 1$, the embedding procedure performs according to the following steps:

(WE.1) Choose N_W points (x_i, y_i), $i = 0, 1, \cdots, N_W - 1$, out of the coordinates of f_X in a manner such that all the N_W points are situated separately on the image plane. Let the sequence of the ordered points be represented by K; that is, $K = \{(x_0, y_0), (x_1, y_1), \cdots, (x_{N_W-1}, y_{N_W-1})\}$.

(WE.2) According to K, obtain a pixel sequence, X, from f_X; that is, $X = (\boldsymbol{x}_0, \boldsymbol{x}_1, \cdots, \boldsymbol{x}_{N_W-1})$ where $\boldsymbol{x}_i = f_X(x_i, y_i)$ for $i = 0, 1, \cdots, N_W - 1$.

(WE.3) For each pixel $\boldsymbol{x}_i$ in X, $i = 0, 1, \cdots, N_W - 1$, encode the pixel value with a given watermark bit w_i using the operator ENCODE of Equation (18), so that

$$v_i = \mathrm{ENCODE}(f_X(x_i, y_i), w_i). \tag{29}$$

Then, replace the value of $\boldsymbol{x}_i$ with v_i. Expressing the resultant pixel by $\boldsymbol{y}_i$, we thus obtain the sequence $Y = (\boldsymbol{y}_0, \boldsymbol{y}_1, \cdots, \boldsymbol{y}_{N_W-1})$.

(WE.4) Substitute $\boldsymbol{y}_i$ in Y for the pixel at (x_i, y_i) in the image f_X for $i = 0, 1, \cdots, N_W - 1$. Thereby the watermarked image, f_Y, is achieved.

The sequence K is supposed to be associated with each f_Y, and it must be protected from unauthorized use. We here refer to the ratio of N_W to the number of pixels in f_X as an embedding pixel ratio ρ; that is,

$$\rho \triangleq N_W / (N_X N_Y). \tag{30}$$

2) Extracting procedure

Given an image, f_Z, that is assumed to be derived from the above f_Y, the procedure for extracting watermarks from f_Z is as follows:

(WD.1) Using the same K as that has been used in the embedding procedure for f_X, obtain a pixel sequence, Z, from f_Z; that is, $Z = (\boldsymbol{z}_0, \boldsymbol{z}_1, \cdots, \boldsymbol{z}_{N_W-1})$ where $\boldsymbol{z}_i = f_Z(x_i, y_i)$ for $(x_i, y_i) \in K$, $i = 0, 1, \cdots, N_W - 1$.

(WD.2) For each pixel $\boldsymbol{z}_i$ of Z, $i = 0, 1, \cdots, N_W - 1$, decode a bit, w'_i, from the pixel value using the operator DECODE of Equation (22). We thereby obtain the bit

sequence $W' = \left(w'_0, w'_1, \cdots, w'_{N_W-1}\right)$.

4.2. Pixel Surrounding Method

The performance of the proposed ECC method has been compared with that of the conventional pixel surrounding (PS) method [3]. The procedures of the PS method are outlined below. In addition, our analysis will reveal its essential properties in terms of embedding distortions and robustness.

1) Embedding procedure

Given a source image f_X and a sequence of N_W watermark bits, the embedding procedure is as follows:

(PE.1) As in the step (WE.1) of the ECC method, determine a sequence of N_W points in f_X. Let K denote the sequence, and the same notation as that in the ECC procedure is used here.

(PE.2) According to K, obtain a sequence, X, of N_W pixels from f_X.

(PE.3) Foreach $\boldsymbol{x}_i$ in X, calculate the mean value, a_i, of the eight pixels surrounding $\boldsymbol{x}_i$ in the neighborhood of 3 by 3 pixels. Thus, a sequence $A = \left(a_0, a_1, \cdots, a_{N_W-1}\right)$ is obtainedfor all $i = 0, 1, \cdots, N_W - 1$.

(PE.4) For the ith watermark bit $w_i \in \{0, 1\}$, replace the signal value of $\boldsymbol{x}_i$ with $\hat{a}_i$ such that

$$\hat{a}_i = \begin{cases} a_i - \delta, & \text{if } w_i = 0 \\ a_i + \delta, & \text{otherwise} \end{cases} \tag{31}$$

for $i = 0, 1, \cdots, N_W - 1$, where δ is a specified positive value of signal level. Letting $\boldsymbol{x}'_i$ denote the resultant pixel, we thereby obtain a pixel sequence $X' = \left(\boldsymbol{x}'_0, \boldsymbol{x}'_1, \cdots, \boldsymbol{x}'_{N_W-1}\right)$.

(PE.5) Substitute $\boldsymbol{x}'_i$ in X' for the pixel at $\left(x_i, y_i\right)$ in the image f_X. Let $f_Y^{(P)}$ be the resulting image, so that $f_Y^{(P)}\left(x_i, y_i\right) = \hat{a}_i$ for $i = 0, 1, \cdots, N_W - 1$.

2) Extracting procedure

The procedure for extracting watermarks from a given image, $f_Z^{(P)}$, that is supposed to come from the above $f_Y^{(P)}$ is as follows:

(PD.1) Using K that has been determined in the step (PE.1), obtain a pixel sequence, Z, from $f_Z^{(P)}$; that is, $Z = \left(\boldsymbol{z}_0, \boldsymbol{z}_1, \cdots, \boldsymbol{z}_{N_W-1}\right)$ where $\boldsymbol{z}_i = f_Z^{(P)}\left(x_i, y_i\right)$ for $\left(x_i, y_i\right) \in K$, $i = 0, 1, \cdots, N_W - 1$.

(PD.2) For each $\boldsymbol{z}_i$ in Z, calculate the mean value, a''_i, in the neighborhood of $\boldsymbol{z}_i$ in the same way as in the step (PE.3). Then, a sequence $A'' = \left(a''_0, a''_1, \cdots, a''_{N_W-1}\right)$ is obtained.

(PD. 3) By comparing the signal value $f_Z^{(P)}\left(x_i, y_i\right)$ with a''_i, a bit value, w''_i, is determined in a manner such that

$$w''_i = \begin{cases} 0, & \text{if } f_Z^{(P)}\left(x_i, y_i\right) < a''_i \\ 1, & \text{otherwise} \end{cases} \tag{32}$$

for $i = 0, 1, \cdots, N_W - 1$. Thus, the bit sequence $W'' = \left(w''_0, w''_1, \cdots, w''_{N_W-1}\right)$ is obtained as a watermark extracted from $f_Z^{(P)}$.

3) Embedding distortions

Embedding distortions caused by the PS method are signal changes at the embedding points. To estimate the amount of the distortion, we use the measurement of MSE between the two images f_X and $f_Y^{(P)}$, expressing it by MSE_{PS}. The general definition of MSE between two images f_1 and f_2 of the same size of N_H by N_V pixels is

$$\text{MSE} \triangleq \frac{1}{N_H N_V} \sum_x \sum_y \left\{f_2(x, y) - f_1(x, y)\right\}^2 \tag{33}$$

where the summation is taken over the entire image area. Replacing f_1 and f_2 in Equation (33) with the respective images of f_X and $f_Y^{(P)}$ yields directly the form of MSE_{PS}. Based on the definition, the value of MSE_{PS} for a given δ is approximately proportional to the embedding ratio. Using MSE_{PS} instead of MSE in Equation (20), we can also obtain the value of PSNR.

At each of the embedding points in $f_Y^{(P)}$, the pixel has a value different from the average of its neighborhood pixels by δ according to Equation (31). Then, the pixel value can be regarded as the sum of the spatial average and a noise signal that is randomly chosen between δ and $(-\delta)$ on the assumption that the watermark bits are random variables. Hence, the image $f_Y^{(P)}$ gets closer to a result of spatial averaging with decreasing δ, and accordingly, it looks more blurring, particularly when the embedding pixel ratio is large and close to 1.

4) Robustness

In the step (PE.3) of the above embedding procedure, the neighborhood averaging is carried out using source signal levels of f_X and hence, the resulting average at each point is independent of the situations of any other points of K. In addition, the steps (PE.4) and (PE.5) are implemented by point operations. Accordingly, the steps from (PE.2) through (PE.5) can be accomplished by a parallel processing with the 3 by 3 operator applied to f_X at each point of K. The pixels of $f_Y^{(P)}$ are consequently generated from f_X independently of each other.

On the other hand, in the extracting procedure, the step (PD.2) involves the calculation in the neighborhood in $f_Z^{(P)}$ around each point of K. This spatial operation makes the mean value calculated at each point dependent

on the situations of the other points of K as described below. Suppose that a point of K, say, (x_j, y_j), has its neighbors including another point of K, say, (x_ℓ, y_ℓ). Also, suppose a mean value, a'_j, calculated in the neighborhood around (x_j, y_j) in $f_Y^{(P)}$. Then, a'_j is no longer equal to a_j obtained in the step (PE.3) due to the difference between the level of $\boldsymbol{x}'_\ell = f_Y^{(P)}(x_\ell, y_\ell)$ and $\boldsymbol{x}_\ell = f_X(x_\ell, y_\ell)$.

The difference between the mean values a_j and a'_j has a certain effect on the performance of watermark extraction. The condition for extracting correct watermark bits from $f_Y^{(P)}$ is given from Equation (32) as $(-\delta) < \alpha_j \le \delta$ where $\alpha_j = a'_j - a_j$. This interval of α_j just depends on δ. If $|\alpha_j|$ is larger than δ, either a watermark bit of 1 or that of 0 is unable to recover, according to whether α_j is positive or negative. Hence, all the watermark bits would be hard to recover even from the watermarked and still unchanged image without any limitation of the situations of embedding points. The condition sufficient for correctly recovering all the watermark bits is that every point of K is situated outside the 3 by 3-pixel neighborhoods of any other points of K. Note that satisfying this condition makes the embedding pixel ratio ρ unable to exceed 1/9.

As for a processed and probably changed image $f_Z^{(P)}$, let us express the pixel values and the mean values calculated in the step (PD.2) by the relations

$$\beta_i \triangleq f_Z^{(P)}(x_i, y_i) - f_Y^{(P)}(x_i, y_i) \tag{34}$$

and

$$\alpha'_i \triangleq a''_i - a_i \tag{35}$$

respectively, for $i = 0, 1, \cdots, N_W - 1$. Substituting these expressions in Equation (32), we find that if $(-\delta) \le \varepsilon_i < \delta$ where $\varepsilon_i \triangleq \beta_i - \alpha'_i$, the watermark bit of either value is to be correctly extracted from $f_Z^{(P)}$. If $\varepsilon_i < (-\delta)$, the watermark bit of 1 is lost, and otherwise, the watermark bit of 0 is lost.

4.3. Experimental Methods

The experimental procedure that was carried out to evaluate the ECC method is described here. Some 8-bit monochrome images of 256 by 256 pixels, such as commonly used "Lena," were used as source images in the experiment. The embedding of watermark bits was carried out for all the pixels of a source image so that the performance with the upper limit of the embedding pixel ratio, that is, $\rho = 1$ could be evaluated. Note that the watermarking is actually to be implemented with $\rho \ll 1$ to hide where watermarks are embedded. As an instance of watermark, W, a sequence of 256 × 256 bits was generated using a pseudo-random number generator with a computer, and thus, $W = (w_0, w_1, \cdots, w_{N_W-1})$ where $N_W = 256 \times 256$.

The procedure applied to each source image, f_X, consists of the following steps:

(EX.1) Embed W into f_X by the ECC method. Let f_Y denote the resulting image. Then, measure a PSNR value between f_X and f_Y.

Here, $\mathrm{ECC_{G0}}$ and $\mathrm{ECC_{G1}}$ were used as the Gray-ECC method, and also, $\mathrm{ECC_{N0}}$ and $\mathrm{ECC_{N1}}$ were used for comparison. Furthermore, the PS method was implemented for f_X with $\rho = 1$ and various values of δ. Thus the respective f_Ys were produced from each f_X by these methods.

(EX.2) Compress f_Y using the JPEG DCT-baseline system, and decompress the resultant data to reconstruct an image, f_Z. Then, measure a PSNR value between f_Y and f_Z to estimate the amount of distortion caused by the JPEG lossy compression.

To obtain various amount of distortion, the JPEG coding performance was varied by multiplying the entire luminance-quantization matrix that is specified as the recommended one in [8] by a constant and thus, several f_Zs with the various distortions were produced for each f_Y.

(EX.3) Extract a watermark-bit sequence from f_Z using the decoding procedure corresponding to each embedding method. Let the resulting bit-sequence be $W' = (w'_0, w'_1, \cdots, w'_{N_W-1})$.

We define a watermark-bit correct ratio, R_C, of f_Z as the ratio of correct watermark bits among those extracted from f_Z. Using the above W and W', it can be calculated by

$$R_C = 1 - \frac{\sum_{i=0}^{N_W-1} |w'_i - w_i|}{N_W}. \tag{36}$$

R_C of f_Y can be defined similarly from a watermark-bit sequence extracted from f_Y.

As described in the step (EX.1), all the methods are compared under the condition that the embedding pixel ratio $\rho = 1$. As for other values of ρ, due to the point operation in these methods, generally, MSE values between f_X and f_Y defined by Equation (33) vary almost in proportion to ρ of f_Y for each f_X.

4.4. Experimental Results and Discussion

1) Quality of watermarked images

Table 2(a) lists PSNR values of the watermarked images f_Y obtained by applying each of the ECC methods to four different source images f_X: "Lena," "Aerial,"

Table 2. Embedding distortions by ECC methods. (a) PSNR (dB); (b) WSNR (dB).

(a)

Image	ECC_{G0}	ECC_{G1}	ECC_{N0}	ECC_{N1}
Lena	38.8	33.4	40.7	35.6
Aerial	38.8	33.3	40.7	35.7
Peppers	38.8	33.4	40.7	35.6
Parrots	38.8	33.2	40.7	35.6
Average	38.8	33.3	40.7	35.6

(b)

Image	ECC_{G0}	ECC_{G1}	ECC_{N0}	ECC_{N1}
Lena	42.5	37.1	44.7	39.2
Aerial	42.6	37.0	44.7	39.4
Peppers	42.4	36.8	44.5	39.0
Parrots	42.2	36.7	44.3	39.1
Average	42.4	36.9	44.6	39.2

"Peppers", and "Parrots". Compared among these images, the PSNR values given by each method are almost the same; the PSNR values given by ECC_{G0} and those given by ECC_{G1} agree well with the respective values obtained by the stochastic analysis (previously shown in **Table 1**), and so do the PSNR values given by ECC_{N0} and ECC_{N1}. On the average, PSNR values of the results of ECC_{G1} are about 5.5 dB smaller than those of the results of ECC_{G0} for any test image; PSNRs of the results of ECC_{Gb} are about 2 dB smaller than those of the results of ECC_{Nb} for $b = 0$ and 1. Note that R_C of f_Y by any ECC method is equal to 1.

In addition, weighted signal-to-noise ratio (WSNR) values were calculated from the difference between f_Y and f_X. In the calculation, the viewing distance of 13 inches and the image resolution of 200 pixels per inch were assumed. **Table 2(b)** lists the WSNR values of the results of each ECC method for the four test images. On the average in the ECC methods, each watermarked image has a WSNR value 3.7 dB larger than the PSNR value.

With regard to the PS method, **Table 3(a)** shows PSNR values and R_Cs of f_Y obtained by the experiment using the four test images with $\rho = 1$ and various values of δ; **Table 3(b)** shows those values for $\delta = 3$ and some values of ρ; **Table 3(c)** shows WSNR values for $\rho = 1$ and some δs. The results for $\rho = 1$ in **Table 3(a)** demonstrate that the PSNR values are substantially small and more than 10 dB smaller than those of ECC_{G0} even for $\delta = 1$. Increasing δ decreases the PSNR values whereas achieving large values of R_C requires

Table 3. Watermarking results of PS method. (a) PSNR (dB) and R_C with ρ = 1; (b) PSNR (dB) and R_C with δ = 3; (c) WSNR (dB) with ρ = 1.

(a)

	Lena		Aerial		Peppers		Parrots	
δ	PSNR	R_C	PSNR	R_C	PSNR	R_C	PSNR	R_C
1	27.5	0.72	25.3	0.60	28.5	0.76	27.8	0.83
3	27.2	0.88	25.1	0.76	28.2	0.89	27.5	0.93
5	26.7	0.93	24.8	0.85	27.5	0.93	27.0	0.96
10	24.9	0.97	23.5	0.95	25.3	0.97	25.0	0.98
15	22.9	0.99	21.9	0.98	23.2	0.99	22.9	0.99

(b)

	Lena		Aerial		Peppers		Parrots	
ρ	PSNR	R_C	PSNR	R_C	PSNR	R_C	PSNR	R_C
1	27.2	0.88	25.1	0.76	28.2	0.89	27.5	0.93
1/2	30.3	0.92	28.1	0.84	31.2	0.93	30.4	0.96
1/4	33.3	0.96	31.1	0.90	34.2	0.96	33.3	0.97

(c)

δ	Lena	Aerial	Peppers	Parrots
3	30.4	27.4	30.5	30.3
5	30.0	27.2	30.0	29.8

large values of δ. It is also revealed from **Table 3(b)** that for any of the test images, although the PSNR value increases linearly with decreasing ρ as expected, it is still small even for $\rho = 1/4$: The PS method with $\delta = 3$ and $\rho = 1/4$ is comparable to ECC_{G1} in terms of PSNR. Also, from a comparison between **Tables 3(a)** and **(c)**, we find that each watermarked image has a WSNR value, on the average, 2.7 dB larger than the PSNR value.

To evaluate image appearance, **Figure 5** shows examples of the watermarked images. The ECC_{G0} hardly yields perceptible distortions on the resulting images as shown in **Figure 5(b)**. As seen in **Figure 5(c)**, on the contrary, the ECC_{G1} tends to generate visible artifacts similar to "false contours" [7] particularly in areas of smooth gray levels. As for the PS method, the image degradation is mainly due to the averaging operation, which is a kind of low-pass filtering: the watermarked images of small δs as shown in **Figure 5(d)** look considerably blurred; the images of large δs look grainy as well as blurred as seen in **Figure 5(e)**.

2) Error-correcting capability

The error-correcting capabilities of the ECC methods for signal changes were evaluated from the experimental

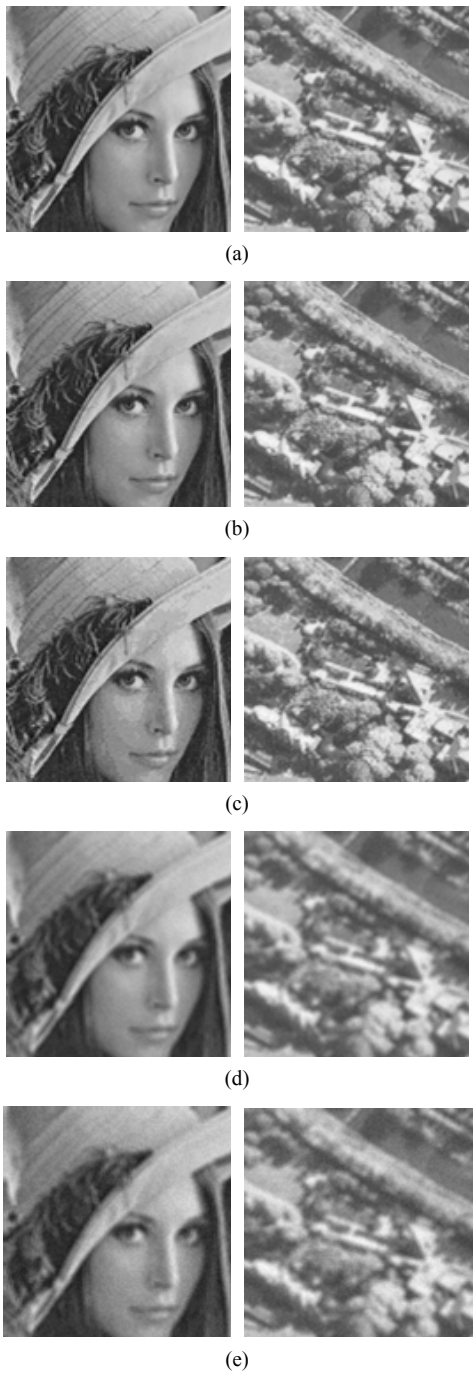

Figure 5. Examples of images "Lena" (left) and "Aerial" (right): (a) source images; results of watermarking (b) by ECC_{G0}, (c) by ECC_{G1}, (d) by PS method with $\rho = 1$ and $\delta = 3$, and (e) by PS method with $\rho = 1$ and $\delta = 5$. All the images show the 128 by 128-pixel section centered in their respective images.

results. As a measure of the capability, we define a error-correction success ratio, R_S, of f_Z for f_Y as the ratio of those pixels from which correct watermark bits are able to be extracted among those watermarked pixels of f_Z whose signal levels differ from those of f_Y. To formulate R_S for calculation, let K_{err} be a set of locations of those watermarked pixels whose values differ between f_Y and f_Z; that is,

$$K_{\text{err}} \triangleq \left\{ \left(x_i, y_i\right) \,\middle|\, \left(x_i, y_i\right) \in K,\ f_Z\left(x_i, y_i\right) \neq f_Y\left(x_i, y_i\right) \right\}. \tag{37}$$

Then, using the original watermark W and the extracted W', R_S can be calculated by

$$R_S = 1 - \frac{\sum_{i=0}^{N_W - 1} \left| w_i' - w_i \right|}{\left\langle K_{\text{err}} \right\rangle} \tag{38}$$

where $\left\langle K_{\text{err}} \right\rangle$ represents the number of elements of K_{err}.

Before evaluating error-correcting performance, we examine the amount of signal distortion caused by JPEG compression. **Figure 6** shows examples of occurrence frequencies of difference levels between a JPEG-compressed image and its source image for various degrees of data compression. As this figure demonstrates, the difference levels tend to concentrate into small ones and also, the range is limited to a few levels particularly in the distorted images of PSNR larger than 40 dB.

Figure 7 shows the experimental results for two test images, depicting variations of R_S with PSNR values of JPEG-compressed images for each of the ECC methods. Both the images have almost the same result. These experimental results agree well with the stochastic analyses described in Section 3.3: As the PSNR value increases, the MSE value equivalently decreases, and accordingly, the difference levels concentrate in a narrow range as **Figure 6** has shown. Hence, the accumulative probability of correct decoding can explain the variation of R_S: R_S values of ECC_{G0}, ECC_{G1}, ECC_{N0} and ECC_{N1} get close to

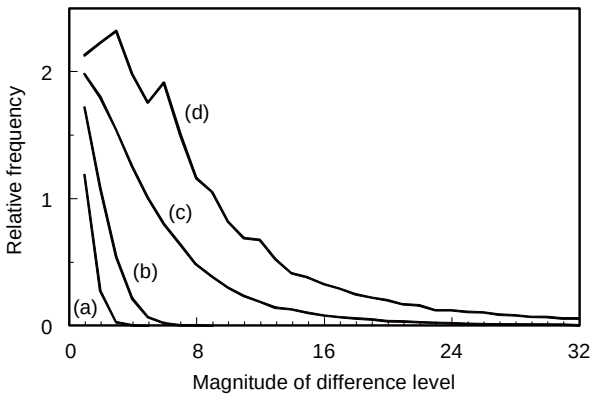

Figure 6. Examples of distribution of difference levels in JPEG compressed images: The source image to JPEG encoder is "Lena" watermarked by ECC_{G0}; JPEG compressed images have respective PSNRs of (a) 48.0 dB, (b) 42.5 dB, (c) 31.4 dB, and (d) 27.0 dB; The relative frequency of difference level *d* is defined as the ratio of the number of pixels of difference level *d* to the number of unchanged pixels in each image.

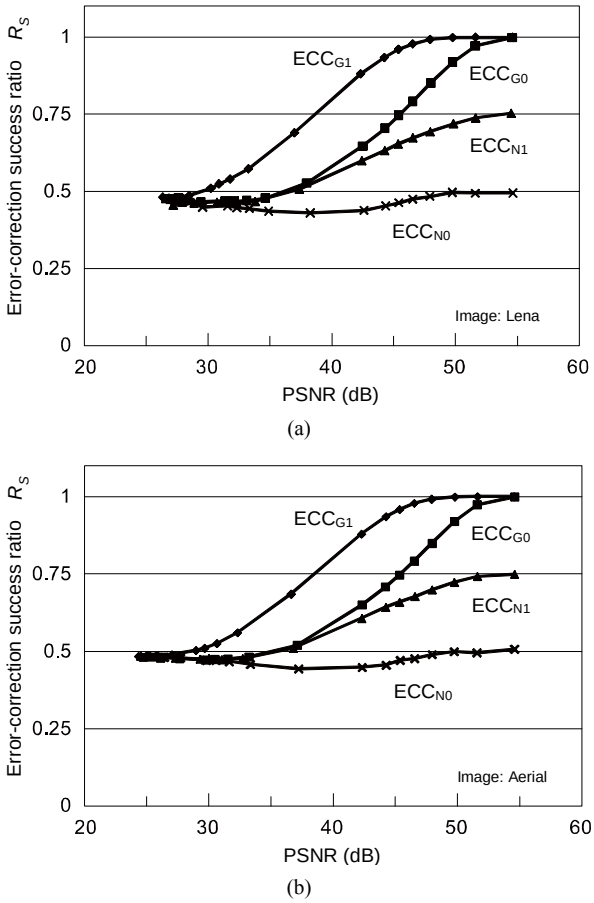

Figure 7. Variation of error-correction success ratio with amount of signal distortion caused by JPEG lossy compression for the test image (a) "Lena" and (b) "Aerial."

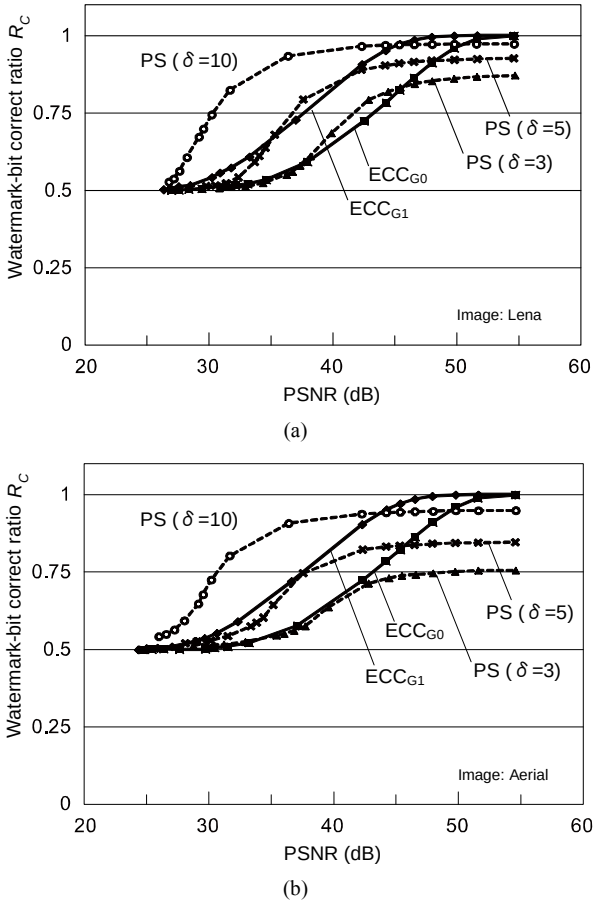

Figure 8. Variation of watermark-bit correct ratio with amount of signal distortion caused by JPEG lossy compression for the test image (a) "Lena" and (b) "Aerial."

the respective values of $\hat{P}_{G0}(0)$, $\hat{P}_{G1}(0)$, $\hat{P}_{N0}(0)$ and $\hat{P}_{N1}(0)$, which have been shown in **Figures 3(b)** and **4(b)**. In **Figure 7**, ECC$_{G0}$ and ECC$_{G1}$ have R_S over 0.5 beyond about 35 dB and 30 dB, respectively, and achieve R_S over 0.9 above about 50 dB and 45 dB, respectively. Thus, these two methods have the capability of error correction for JPEG distorted images. In contrast, ECC$_{N0}$ is unable to afford the error correction for any distribution of difference levels; ECC$_{N1}$ has R_S values greater than 0.5 and as much as 0.75 beyond about 37 dB.

3) Robustness for JPEG lossy compression

Figure 8 shows the experimental results for two test images, depicting variations of R_C with PSNR values of JPEG-compressed images for ECC$_{G0}$ and ECC$_{G1}$. As regards these ECC methods, both the images have almost the same result. These results indicate that correct watermark bits can be recovered from JPEG-distorted images of PSNR of over 30 dB by either Gray-ECC method.

Figure 8 includes the results of the PS method with $\rho = 1$ and $\delta = 3$, 5, and 10 for comparison. In accordance with **Table 3**, R_C values for the image "Lena" are larger than those for the image "Aerial." **Figure 8(a)** implies that the performance of ECC$_{G0}$ is comparable to that of the PS method with $\delta = 3$ in the range of about 35 to 45 dB, and the performance of ECC$_{G1}$, that of the PS method with $\delta = 5$ in the range of about 35 to 40 dB. To obtain a better performance than that of ECC$_{G1}$, the PS method needs to be implemented with a large δ value such as 10 at the expense of enormous embedding distortions.

5. Conclusions

In this paper, the error-correction coding in Gray codes (ECC-in-Gray) has been considered to enhance the error-correcting capabilities, and its performance has been evaluated using the Gray-ECC method where ECC-in-Gray is applied to the protection of image signals watermarked in the spatial domain. The stochastic analyses and the experimental results have shown that although level changes caused by error-correcting codewords are somewhat enlarged by the transformation from Gray to natural

binary codes, ECC-in-Gray is suitable for distorted signals of such error-distribution as caused by lossy data compression.

The performance of the Gray-ECC method is summarized in comparison with that of the PS method as follows:

1) The Gray-ECC method has robustness against signal distortions caused by JPEG lossy compression comparable to that of the PS method.

2) The Gray-ECC method yields a considerably smaller amount of embedding distortion than the PS method in terms of PSNR.

3) The embedding distortion caused by the Gray-ECC method looks similar to the signal distortion by coarse quantization and tends to get noticeable particularly in smooth image areas. As for the PS method, the watermarked images primarily look blurred because of the use of neighborhood averaging, and increasing the capacity of robustness makes the image appearance grainier.

ECC_{G1} has stronger robustness than ECC_{G0}, but it possibly yields more perceptible distortions in smooth image areas. From this point of view, effectively robust watermarks are expected to be achieved by switching between ECC_{G0} and ECC_{G1} for each image area depending on, for example, the degree of smoothness of the area. Such an adaptive system will be a subject in the further study.

REFERENCES

[1] B. M. Macq, "Special Issue on Identification and Protection of Multimedia Information," *Proceeding of the IEEE*, Vol. 87, No. 7, 1999, pp. 1059-1276.

[2] I. J. Cox, M. L. Miller and J. A. Bloom, "Digital Watermarking," Morgan Kaufmann Publishers, San Francisco, 2002.

[3] F. H. Wang, J. S. Pan and L. C. Jain, "Innovations in Digital Watermarking Techniques," Chapter 4, Springer-Verlag, Berlin, 2009. doi:10.1007/978-3-642-03187-8

[4] G. Clark Jr., and J. Cain, "Error-Correction Coding for Digital Communications," Plenum Press, New York, 1981. doi:10.1007/978-1-4899-2174-1

[5] J. R. Hernández, F. P. González and J. M. Rodriguez, "The Impact of Channel Coding on the Performance of Spatial Watermarking for Copyright Protection," *Proceedings of the* 1998 *IEEE International Conference on Acoustics*, *Speech*, *and Signal Processing*, Seattle, May 1998, pp. 2973-2976.

[6] Y. Yi, M. H. Lee, J. H. Kim and G. Y. Hwang, "Robust Wavelet-Based Information Hiding through Low-Density Parity-Check (LDPC) Codes," In: T. Kalker, I. J. Cox and Y. M. Ro, Eds., *Digital Watermarking*, Springer-Verlag, Berlin, 2004, pp. 117-128. doi:10.1007/978-3-540-24624-4_9

[7] R. C. Gonzalez and R. E. Woods, "Digital Image Processing," Addison-Wesley, Boston, 1993.

[8] ISO CD 10918, "Digital Compression and Coding of Continuous-Tone Still Images," ISO/IEC JTC1/SC2/WG10, 1991.

