5

Sophisticated Spatial Domain Watermarking by Bit Inverting Transformation

Tadahiko Kimoto
Toyo University
Japan

1. Introduction

Digital watermarking is a technique for embedding additional signals, watermarks, into digital signals such as images and afterward extracting them (Macq (1999)). From the viewpoint of domains to embed watermark signals into, the watermarking techniques are mainly divided into two categories: spatial-domain-based techniques and transform-domain-based ones.

In the watermarking of digital images, the transform-domain-based techniques can usually provide not only good visual quality in the resulting images but also stronger robustness against image modification than the spatial-domain-based ones. However, it is hard to embed watermarks exactly into transform domains. In the embedding procedure, the pixel values of a source image, which are usually quantized levels or integers, are first transformed into frequencies. The frequency coefficients are then modified so as to represent watermarks. The values inversely transformed from such modified transform domain are usually real numbers with fractions. Consequently, the quantization errors occur inevitably when the integral spatial domain is reconstructed. These errors are likely to disturb the watermark that has been embedded in the transform domain.

In contrast, exact watermark signals can be embedded into the spatial domain though they are fragile under signal modification. The traditional method for spatial-domain-based image watermarking is first to select pixels in a source image and then, to modify the levels of the selected pixels so that the watermark can be expressed there (Wang et al. (2009)). The most primitive method for modifying a pixel level is to select a bit in the binary expression of the level and then, to invert the selected bit (Oka & Matsui (1997)). In this method, the bit position selected in the binary expression as well as the pixel location selected in a source image must be kept secret so that the watermark can be protected from unauthorized access.

The embedded watermark distorts the source signal to some extent. The transformation in the image spatial domain has especially direct effects on visual quality. In the bit inverting method, inverting the kth bit of a signal, denoted by the bit k, where the 0th bit is the least significant bit, changes the signal level by 2^k. Hence, the visual distortion caused by the level change increases with an increase in k. On the other hand, when a bit k represents a watermark bit, an attacker searches some range of k's for the correct value to get the bit k. With increasing k, the range to be searched becomes wider, and accordingly, the tolerance to unauthorized access can increase. Thus, determining appropriate values of k is very important

to the watermarks involving the bit inverting. Also, the extending of ranges of k and the preserving of visual quality are contradictory subjects.

It is desirable to determine the values of k automatically for given source images to carry out the watermarking efficiently. To choose an appropriate bit k to be inverted in terms of visual quality, a human perceptual model is necessary. The perceptual model means a relationship between quantities representing objective qualities and human subjective qualities of the images being viewed, based on the human visual system (Awrangjeb & Kankanhalli (2004)). In other words, such a perceptual model is a function of objective quality measures that produces a subjective quality measurement as an estimation of human subjective quality. Various kinds of quantities have been proposed for the measurement of objective quality (Wang et al. (2004)). By adaptively determining the embedding parameters such as the values of k by means of the human perceptual model, the watermarking scheme can perform as a *perceptually adaptive system* (Cox et al. (2002)).

In the next section, the function of inverting signal bits is discussed. There a method for inverting a signal bit with making the resultant level change minimum is presented (Kimoto (2005); Kimoto (2007)). In this method, the inversion of the kth bit ($k \geq 1$), where a 0th bit is a least significant bit, results in the change in the signal level by 2^{k-1} or under for any input level. Also, randomly varying signals are added to the transformation outputs so as to improve signal quality (Kimoto (2006)). Both the function of inverting bits and that of randomizing levels are given as a single transformation. The performance of the transformation is analyzed in detail and also demonstrated by some experiments.

The next section treats some subjects regarding the implementation of the bit inverting transformation. First, the transformation domains are considered; a principle for defining domains of the transformation in the input dynamic range under limitations on level changes is formulated. Also, a method for dividing the input dynamic range into a union of transformation domains so that the blind watermarking can be achieved is described. Next, a scheme for embedding watermark bits in every image block using the transformation is presented with some experimental results.

In the next section, the subject of determining a kth bit to be inverted, or the value of k in the same sense, is discussed; a scheme for implementing the bit inverting transformation to embed watermark bits so that a required subjective visual quality can be achieved on the resulting image is developed (Kimoto & Kosaka (2010)). To derive an appropriate perceptual model for such images that are watermarked by the bit inverting transformation, first, objective quality measures are defined based on the properties of the transformation. Then, a subjective visual quality measure based on the objective quality measures is established by subjective evaluations of human observers. A perceptually adaptive image watermarking scheme using the perceptual model is presented. This scheme is aimed at embedding watermark bits in every pixel all over the source images in contrast with the block watermarking scheme described in the preceding section. The performance of the scheme is demonstrated by some experiments.

2. Bit inverting transformation

2.1 Inverting signal bits

Inverting a single bit of a signal can be expressed by a level transformation. Signal levels are supposed to be uniformly quantized to M bits and expressed from the M-bit sequence of

$0 \cdots 0$ to that of $1 \cdots 1$ in natural binary. Each bit in the binary expression is denoted by bit k for $k = 0, 1, \ldots, M-1$, and bit 0 means the least significant bit. For a given level $v \in [0, 2^M - 1]$ and a specified value of $k \in [0, M-1]$, inverting the bit k of v transforms v to another level v'; this transformation is described as a function t_k of v in the relation

$$v' = t_k(v) = \begin{cases} v + 2^k, & \text{if } n = 2m \\ v - 2^k, & \text{if } n = 2m + 1 \end{cases} \tag{1}$$

where, letting $\lfloor x \rfloor$ denote the integral part of the real number x, $n = \left\lfloor v/2^k \right\rfloor$, and m has integers in the interval $[0, 2^{M-k-1} - 1]$ as a result. Each input level thus either increases or decreases just by 2^k.

2.2 Minimizing level changes

The amount of level change caused by an inverted bit can be minimized by altering the other bits of the signal appropriately. The principle for transforming the entire bit-pattern of a signal so as to achieve the smallest level change is illustrated in Fig. 1. Let $b(v, k)$ denote a binary value of the bit k of a level v, and $\overline{b}$ denotes the 1's complement of a bit value b. Suppose that the kth bit of a level v is to be inverted. Then, the inverting of the bit is performed by changing the level, that is, the entire bit-pattern to one of those patterns of level v'' which satisfy $b(v'', k) = \overline{b(v, k)}$. Furthermore, the one of the v''s that is closest to v can be chosen as v' to achieve the smallest level change. For the most of v, such v''s exist in both sides of v in the dynamic range as depicted in Fig. 1.

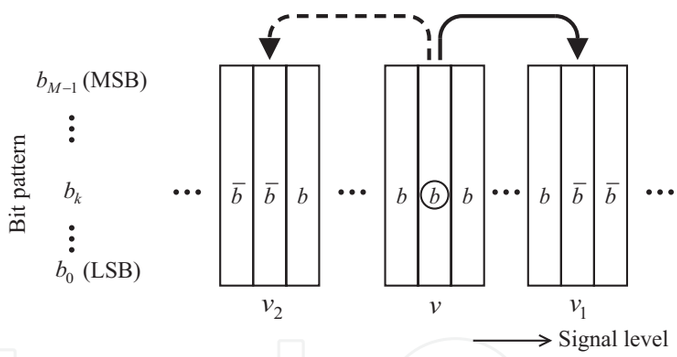


Fig. 1. Changing signal levels for inverting a signal bit: To invert b_k of v, v is changed to either v_1 or v_2.

Both inverting a kth bit and minimizing the resultant level change are performed by a single level transformation. Let $\Delta_k = 2^{k-1}$ where k now assumes values in the range $1, 2, \ldots, M-1$. For a given k, the transformation of a signal level v can be expressed as a function f_k of v, defined by

$$f_k(v) = \begin{cases} 2\Delta_k, & \text{if } n = 0 \\ 2m\Delta_k, & \text{if } n = 2m - 1 \\ 2m\Delta_k - 1, & \text{if } n = 2m \\ 2^M - 2\Delta_k - 1, & \text{if } n = 2^{M-k+1} - 1 \end{cases} \tag{2}$$

where $n = \lfloor v/\Delta_k \rfloor$ and m has integers in the interval $[1, 2^{M-k} - 1]$. The function f_k satisfies the relation

$$b(f_k(v), k) = \overline{b(v, k)}. \tag{3}$$

For $k = 0$, the transformation $f_0(v)$ has the special relation

$$f_0(v) = \begin{cases} v+1, & \text{if } v \text{ is even} \\ v-1, & \text{if } v \text{ is odd.} \end{cases} \tag{4}$$

Figure 2 illustrates $f_k(v)$ over the M-bit dynamic range $v \in [\,0,\ 2^M - 1\,]$, comparing with $t_k(v)$ of Eq. (1) depicted by the thick dashed lines, where $k \geq 1$ (the box of thin dashed lines will be explained later). As shown in this figure, $f_k(v)$ possesses the almost staircase relations between input and output levels.

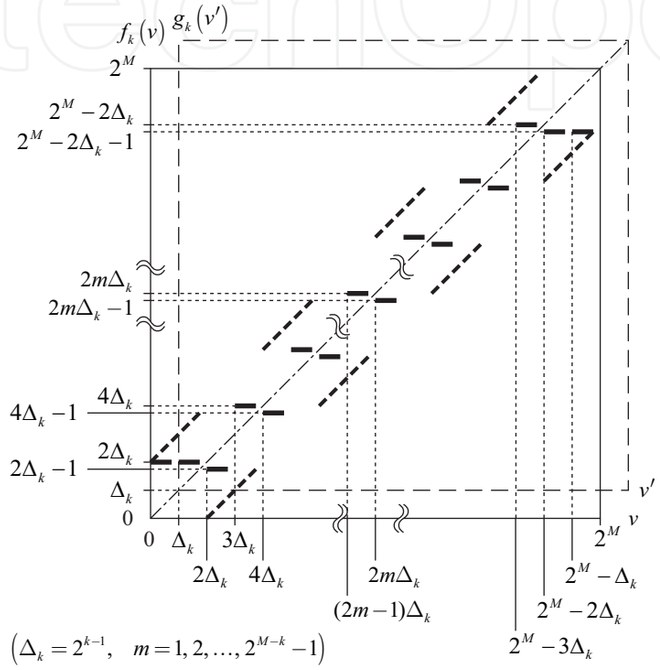


Fig. 2. Transformations for inverting kth bits of M-bit levels ($1 \leq k \leq M-1$): The bold line shows the transformation $f_k(v)$, and the thick dashed line, $t_k(v)$, where $\Delta_k = 2^{k-1}$; the dashed-and-dotted line shows the identity transformation for reference.

The level difference resulting from the level transformation is also a function of the source levels. Figure 3 shows the difference between the transformed level $v_{\text{out}} = f_k(v_{\text{in}})$ and the source level v_{in} in the entire source range, comparing with the difference caused by the transformation t_k. The absolute magnitude of the difference, $|f_k(v_{\text{in}}) - v_{\text{in}}|$, varies in the range from 1 to Δ_k, which is less than or equal to the half of the difference caused by t_k, for the source levels v_{in} over the interval $[\,\Delta_k,\ 2^M - \Delta_k - 1\,]$. On the contrary, for a level v_{in} in the interval $[\,0,\ \Delta_k - 1\,]$, those levels available for bit inversion exist only on the upper side of v_{in} because of the end of M-bit levels. Accordingly, all the source levels in the interval are to be transformed to the smallest one among the available levels, $2\Delta_k$. The resulting level differences consequently get over Δ_k in the interval. A similar end effect occurs for the input levels in the interval $[\,2^M - \Delta_k,\ 2^M - 1\,]$. In this paper we refer to these two intervals as the *end-effect intervals*.

2.3 Modifying level transformation

The M-bit end effects can be removed by translating the coordinate system. Both of the axes of the coordinate system are translated by Δ_k to the positive direction so as to avoid the lower

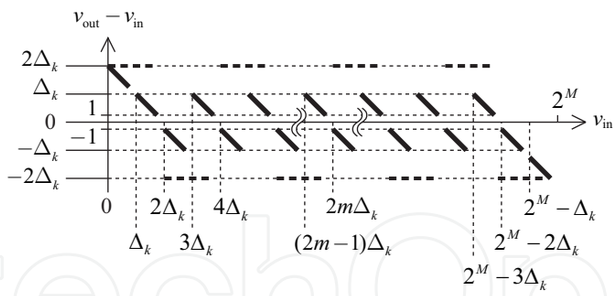


Fig. 3. Characteristics of level change caused by the transformations: The solid lines show $f_k(v_{\text{in}}) - v_{\text{in}}$; the dashed lines show $t_k(v_{\text{in}}) - v_{\text{in}}$.

end-effect interval $[\,0,\ \Delta_k - 1\,]$, that is, letting g_k denote the translated function,

$$\begin{cases} v' = v - \Delta_k \\ g_k(v') = f_k(v) - \Delta_k \end{cases} \tag{5}$$

The translated coordinate system is depicted in Fig. 2 by the thin dashed lines. Accompanied with the coordinate translation, both of the upper bounds of the dynamic range with regard to $f_k(v)$ are removed. Eq. (5) directly leads to the relation

$$f_k(v' + \Delta_k) = g(v') + \Delta_k. \tag{6}$$

Using Eqs. (5) and (6) in Eq. (3), we obtain the relation

$$b(g_k(v') + \Delta_k,\ k) = \overline{b(v' + \Delta_k,\ k)}. \tag{7}$$

This equation indicates that the inversion of bit-k holds between $v' + \Delta_k$ and $g_k(v') + \Delta_k$. By carrying out the above addition of Δ_k in modulus of 2^M, we can correspond the interval $[\,2^M - \Delta_k,\ 2^M - 1\,]$ of v' to the interval $[\,0,\ \Delta_k - 1\,]$ of v, and thus, the upper end-effect interval no longer exists in the translated coordinate system.

In the new coordinate system, the function g_k is defined by a level transformation from an input level v_{in}, that is,

$$g_k(v_{\text{in}}) = \begin{cases} (2m+1)\Delta_k, & \text{if } n = 2m \\ (2m+1)\Delta_k - 1, & \text{if } n = 2m+1 \end{cases}, \tag{8}$$

where $n = \lfloor v_{\text{in}}/\Delta_k \rfloor$ taking values in $[\,0,\ 2^{M-k+1} - 1\,]$ and, as described later, $k \geq 1$. Thus, this transformation maps 2^M consecutive source levels into 2^{M-k+1} discrete output levels.

Figure 4 shows the input-output relationship of the transformation g_k. The transformation is no longer affected by the M-bit end effects. Hence, the level change has the absolute magnitude in the range $[\,1,\ \Delta_k\,]$ throughout the entire source range, as shown in Fig. 5. Thus, the function g_k yields the smallest difference for each input level while achieving the bit k inversion in terms of Eq. (7). Note that it is the bit $(k-1)$ of $g_k(v_{\text{in}})$ that is actually inverted compared with v_{in}, and the level change is not the smallest for each input level in the situation of bit $(k-1)$ inversion. This performance holds for $1 \leq k \leq M-1$. Hence, g_k has been defined in this range of k.

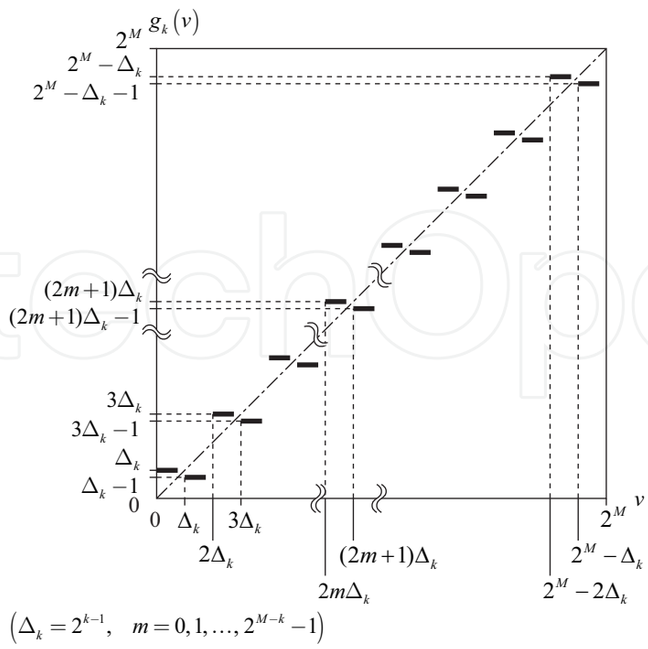


Fig. 4. Modified transformation function g_k.

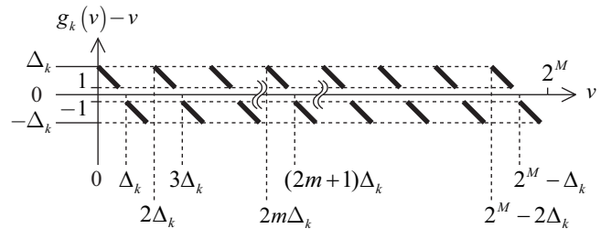


Fig. 5. Characteristics of level change caused by the transformation function g_k.

2.4 Varying transformed levels

The transformation g_k maps 2^M consecutive source levels into 2^{M-k+1} sparse output levels. Furthermore, the output levels consist of pairs of consecutive levels $(2m+1)\Delta_k - 1$ and $(2m+1)\Delta_k$ for $m = 0, 1, \ldots, 2^{M-k} - 1$, as shown in Eq. (8). Hence, the resulting images look like coarsely quantized at $M-k$ bits per pixel.

We extend the range of the transformation outputs within the M-bit dynamic range to improve the resulting image quality. In the pulse code modulation (PCM), dithering signals has effects on improving subjective image quality. Then, we consider distributing the transformed levels in the dynamic range by a stochastic process. A transformation from the outputs $v_{\text{out}} = g_k(v_{\text{in}})$ further to v'_{out} is developed so that all the following three conditions can be satisfied:

(i) The inversion of bit-k holds in terms of

$$b(v'_{\text{out}} + \Delta_k,\ k) = \overline{b(v_{\text{in}} + \Delta_k,\ k)}. \tag{9}$$

(ii) The range of v'_{out} extends to the whole of the M-bit dynamic range.

(iii) The amount of level change is limited so as not to increase excessively. Taking into account that the largest change caused by g_k is Δ_k levels, we determine this limitation as

$$|v'_{\text{out}} - v_{\text{in}}| \leq \Delta_k. \tag{10}$$

The above v'_{out} can be obtained by adding an appropriate random level to v_{out}. Generating such v'_{out} from input level v_{in} is performed using a single level transformation expressed in the relation $v'_{\text{out}} = h_k(v_{\text{in}})$: The function h_k is defined as

$$h_k(v_{\text{in}}) = \begin{cases} g_k(v_{\text{in}}) + r, & \text{if } n = 2m \\ g_k(v_{\text{in}}) - r, & \text{if } n = 2m+1 \end{cases}, \tag{11}$$

where $n = \lfloor v_{\text{in}}/\Delta_k \rfloor$ and r has arbitrary levels in the range $[\,0,\ R_k(v_{\text{in}})\,]$ and $R_k(v_{\text{in}})$ is a function of v_{in}, defined by

$$R_k(v_{\text{in}}) = \begin{cases} v_{\text{in}} - 2m\Delta_k, & \text{if } n = 2m \\ 2(m+1)\Delta_k - v_{\text{in}} - 1, & \text{if } n = 2m+1 \end{cases}. \tag{12}$$

Thus, $R_k(v_{\text{in}})$ varies in the range $[\,0,\ \Delta_k - 1\,]$. The value of r is supposed to be determined within the range in a stochastic manner. Thereby, $h_k(v_{\text{in}})$ is a stochastic function whose output range depends on v_{in} as follows:

$$\begin{cases} h_k(v_{\text{in}}) \in [\,g_k(v_{\text{in}}),\ v_{\text{in}} + \Delta_k\,], & \text{if } n \text{ is even} \\ h_k(v_{\text{in}}) \in [\,v_{\text{in}} - \Delta_k,\ g_k(v_{\text{in}})\,], & \text{if } n \text{ is odd} \end{cases}. \tag{13}$$

The number of levels composing the output range varies in $[\,1,\ \Delta_k\,]$. Hence, the level randomization is effective for $k \geq 2$.

Figure 6 illustrates the input-output relationship of $v'_{\text{out}} = h_k(v_{\text{in}})$ by showing the ranges where the output levels vary by the shaded areas. As seen in the figure, the range of v'_{out} spreads all over the M-bit dynamic range, compared with that of $v_{\text{out}} = g_k(v_{\text{in}})$. The difference $|v'_{\text{out}} - v_{\text{in}}|$, however, tends to get larger than its smallest value for each v_{in}, that is, $|v_{\text{out}} - v_{\text{in}}|$, according to the stochastic manner being used.

2.5 Properties of level transformation

2.5.1 Amount of level change

According to the definition of h_k, the output level $v'_{\text{out}} = h_k(v_{\text{in}})$ varies in a stochastic manner, and so do the level differences between v'_{out} and v_{in}. Figure 7 illustrates the level differences for input levels v_{in}.

We evaluate the amount of level change caused by h_k by stochastic analysis. In the analysis, we assume that M-bit levels in the dynamic range $[\,0,\ 2^M - 1\,]$ are uniformly distributed over the input signals. Also, the values of r in Eq. (11) are assumed to be equally likely in the range assigned for each input level.

For a given k, an input level v_{in} is rewritten as $v_{\text{in}} = n\,\Delta_k + w$ using w in the case that $n = \lfloor v_{\text{in}}/\Delta_k \rfloor$ is even (See Fig. 7). Note that w is identical with $R_k(v_{\text{in}})$ in Eq. (12). By using Eq. (8) in Eq. (11), the output level $v'_{\text{out}} = h_k(v_{\text{in}})$ is given by $v'_{\text{out}} = (n+1)\,\Delta_k + r$, where r is randomly chosen in $[\,0,\ w\,]$, and hence, it is a stochastic variable of a uniform occurrence probability of $1/(w+1)$. The difference $d' = v'_{\text{out}} - v_{\text{in}}$ is then written as $d' = \Delta_k - w + r$ for

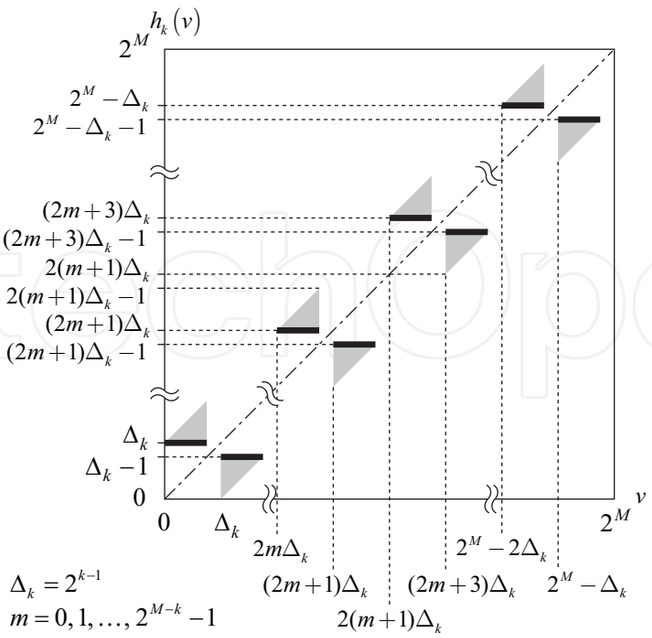


Fig. 6. Transformation function with level randomization, h_k: The thick solid lines show the input-output relationship of g_k for reference.

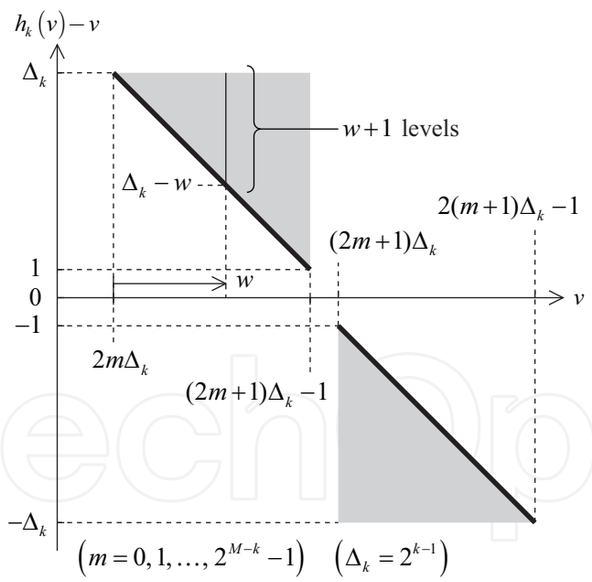


Fig. 7. Characteristics of level change caused by the transformation function h_k: The thick solid lines show the level change caused by g_k for reference.

a certain r. Thus, d' is also a stochastic variable ranging in $[\,\Delta_k - w,\ \Delta_k\,]$ for each w. Then, the expected value of the squared d' are summed over the range of w from 0 to $\Delta_k - 1$ in the form

$$\sum_{w=0}^{\Delta_k-1}\left(\sum_{d'=\Delta_k-w}^{\Delta_k} d'^2\cdot\frac{1}{w+1}\right). \tag{14}$$

The same value of sum is obtained in the case that n is odd. Consequently, averaging the above sum over Δ_k levels yields the mean of the squared level differences over the entire dynamic range, denoted by $E_{\mathrm{H}}(k)$, on the above assumptions. As a result, we have

$$E_{\mathrm{H}}(k) = \frac{1}{36}\left(22\,\Delta_k^2 + 15\,\Delta_k - 1\right). \tag{15}$$

2.5.2 Change of level occurrences

The difference in width among the ranges where the values $v'_{\mathrm{out}} = h_k(v_{\mathrm{in}})$ vary for a given v_{in} makes the occurrence probabilities of output levels unequal. We have analyzed the occurrence frequencies of v'_{out} on both the assumption that M-bit input levels are uniformly distributed in the dynamic range and the assumption that output levels for a given input level v_{in} are equally probable in the range of $h_k(v_{\mathrm{in}})$. Suppose that any input level has a frequency of 1. Then, for an output level $v'_{\mathrm{out}} \in [\,(2m+1)\Delta_k,\ 2(m+1)\Delta_k\,)$, by expressing v'_{out} by $v'_{\mathrm{out}} = (2m+1)\Delta_k + w$ $(0 \le w < \Delta_k)$, the frequency of v'_{out} is given as $\sum_{p=w+1}^{\Delta_k} 1/p$. A similar analysis is obtained for the interval $[\,(2m\Delta_k,\ (2m+1)\Delta_k\,)$ of v'_{out}. Hence, for input levels of the uniform frequency distribution, the output levels of h_k has the frequency distribution as illustrated in Fig. 8.

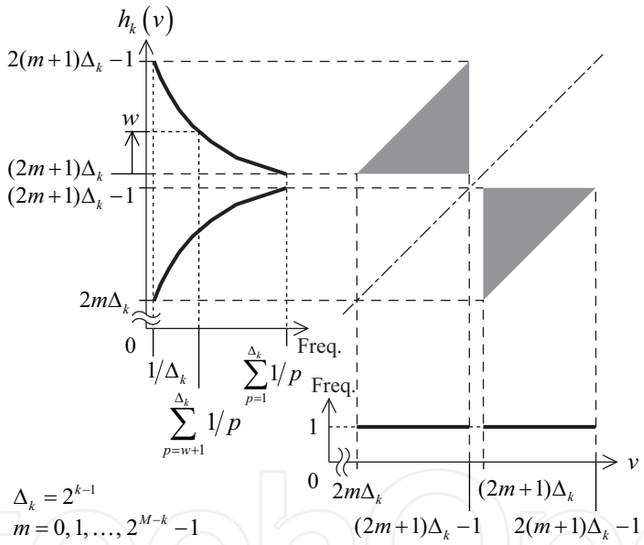


Fig. 8. Occurrence distribution of output levels by the transformation function h_k.

Although the output range of h_k spreads over the entire M-bit range, in every two ranges of Δ_k-level width the order of two ranges are reversed through the mapping, as shown in Fig. 9. These reversed ranges result in distortions on the picture surface.

2.6 Evaluation of level transformation

2.6.1 Amount of level change

We evaluate the amount of level change caused by h_k in terms of the mean squared level difference (MSD). For comparison with the MSD values of h_k in Eq. (15), the expected MSD values of other transformations are derived on the assumption that 2^M levels are uniformly distributed over the M-bit input signals as below. The level change caused by the function

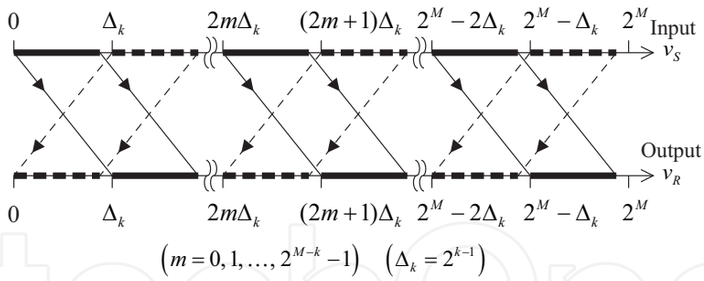


Fig. 9. Correspondence of h_k between input and output levels.

g_k can be evaluated similarly to that caused by h_k. For a given k, the mean squared level difference for the signals transformed by g_k, denoted by $E_{\mathrm{G}}(k)$, is given by

$$E_{\mathrm{G}}(k) = \frac{1}{6}(2\,\Delta_k + 1)(\Delta_k + 1). \tag{16}$$

With regard to the function t_k, for a given k, the magnitude of level change caused by the transformation is $2\Delta_k$ for any input level. Then, the mean squared level difference, denoted by $E_T(k)$, is obviously $4\Delta_k^2$.

For comparison, we consider another method for extending the output levels of g_k within the dynamic range in a stochastic manner; that is, for a given k, all the lowest $(k-1)$ bits of $g_k(v_{\mathrm{in}})$ are replaced with random ones for any v_{in}. We express $g_k(v_{\mathrm{in}})$ and the following bit replacing operation together as a single function of v_{in}, $h'_k(v_{\mathrm{in}})$. The value of $h'_k(v_{\mathrm{in}})$ is a random variable in either range $[\,g_k(v_{\mathrm{in}}),\ g_k(v_{\mathrm{in}}) + \Delta_k - 1\,]$ or $[\,g_k(v_{\mathrm{in}}) - \Delta_k + 1,\ g_k(v_{\mathrm{in}})\,]$ that is determined from the input interval including v_{in}. The whole range of h'_k coincides with the M-bit dynamic range. Also, on the assumption that input levels have a uniform frequency distribution in the dynamic range, the frequency distribution of the output levels gets uniform. As a disadvantage, h'_k tends to increase the resulting level change; the upper bound of the level difference $|h'_k(v_{\mathrm{in}}) - v_{\mathrm{in}}|$ depends on v_{in} and varies in the range $[\,\Delta_k,\ 2\Delta_k\,)$. The expected value of MSD is given on the assumption that output levels for each input level are equally probable in Δ_k levels by

$$E_{\mathrm{L}}(k) = \frac{1}{6}\left(7\,\Delta_k^2 - 1\right). \tag{17}$$

Table 1 lists the MSD values of each transformation for $k = 2,\ 3,\ 4$ and 5. In comparison for a given k, g_k reduces substantially the MSD from that of t_k; the ratio $E_{\mathrm{G}}/E_{\mathrm{T}}$ is, for example, 0.12 for $k = 3$, and approaches approximately to $1/12$ for large k's. The transformation h_k increases the MSD from that of g_k due to the level randomizing; the ratio $E_{\mathrm{H}}/E_{\mathrm{G}}$ is 1.52 for $k = 3$, and about $11/6$ for large k's. On the other hand, the MSD of h_k is smaller than that of h'_k; we find the ratio $E_{\mathrm{H}}/E_{\mathrm{L}}$ to be 0.62 for $k = 3$ and about $11/21$ for large k's.

2.6.2 Experiments of transformation

To visualize the input-output relationship, we have conducted each transformation of t_k, g_k, h_k and h'_k on an 8-bit grayscale ramp image where the pixel level increases at one per pixel from 0 (black) to 255 (white) along the gradation. Figure 10 shows the transformed results with $k = 5$. Also, Fig. 11 illustrates a one-dimensional pixel sequence along the ramp of Fig. 10(d),

	k			
MSD	2	3	4	5
$E_{\mathrm{T}}(k)$	16	64	256	1024
$E_{\mathrm{G}}(k)$	2.5	7.5	25.5	93.5
$E_{\mathrm{H}}(k)$	3.3	11.4	42.4	163.1
$E_{\mathrm{L}}(k)$	4.5	18.5	74.5	298.5

Table 1. Expected values of mean squared difference.

showing an example of the stochastic output levels of h_k. In accordance with Fig. 2, although the results of t_5 has all levels in the dynamic range, there are level gaps every 32 ($= 2^5$) levels as observed in Fig. 10(b). In the gradation transformed by g_5 of Fig. 10(c), we may perceive eight *steps* while the result actually has 16 different levels.

Fig. 10(d) shows the result of adding random signals to Fig. 10(c) by h_5 and has all levels in the dynamic range. For a $v_{\mathrm{in}} = (2m+1)\Delta_k - 1$, the absolute difference between $g_k(v_{\mathrm{in}})$ and $g_k(v_{\mathrm{in}}+1)$ is one level, but the absolute difference between $h_k(v_{\mathrm{in}})$ and $h_k(v_{\mathrm{in}}+1)$, in contrast, can reach the utmost $2\Delta_k - 1$ as shown in Fig. 11. Hence, the visible steps in Fig. 10(c) are divided into halves in Fig. 10(d). The result of h'_k in Fig. 10(e) looks similar to that of h_k (Fig. 10(d)) though the level frequency distributions in the dynamic range are different. Note that an extremely large value of k has been used in Fig. 10 so that the input-output relationships get evident.

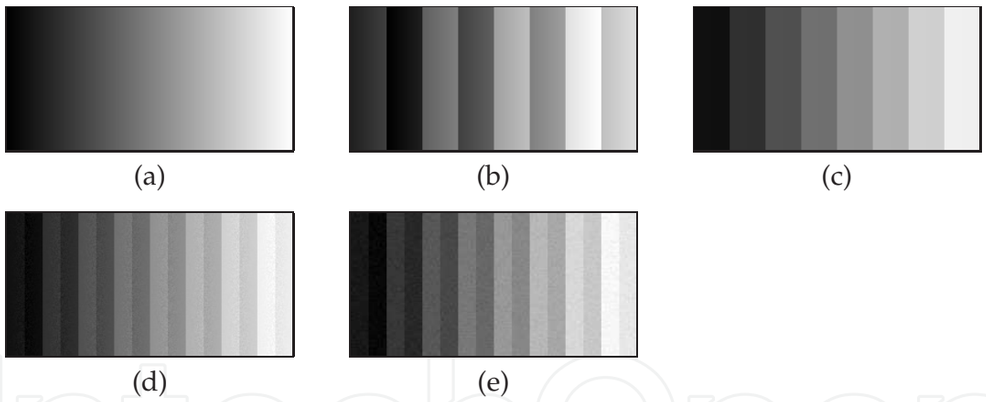


Fig. 10. Experimental results ($k = 5$): (a) Source (8-bit grayscale ramp); (b), (c), (d) and (e) transformed images by t_k, g_k, h_k and h'_k, respectively. All images are printed at 200 pels/inch.

We have also conducted experiments of each transformation on 8-bit monochrome test images. All the pixels in an input image were transformed to evaluate the level changes in terms of visual quality. Figure 12 shows results of the transformations for one of test images, *Lena*. We can observe a distortion pattern similar to that in Fig. 10, in particular, in smooth image areas such as the *shoulder* in each image of Fig. 12. Such distortions can be referred to as *false contours* after those caused by coarsely quantizing pixel levels.

The amount of level change from the source image has been estimated an MSD value for each transformed images. The MSD between a source image f and the transformed image f' is

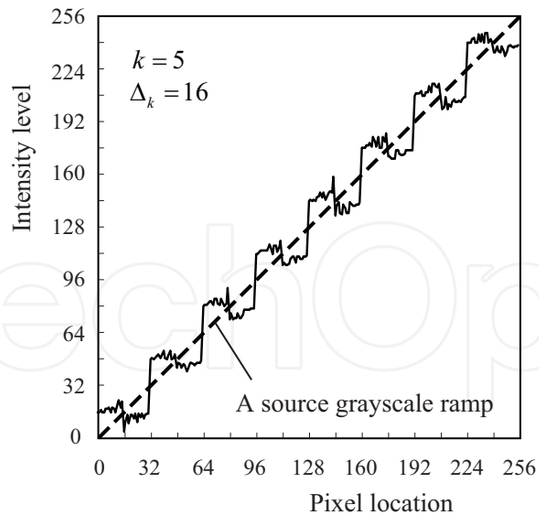


Fig. 11. An example of the transformed 256-grayscale ramp by h_k: levels varying along a horizontal line in Fig. 10(d).

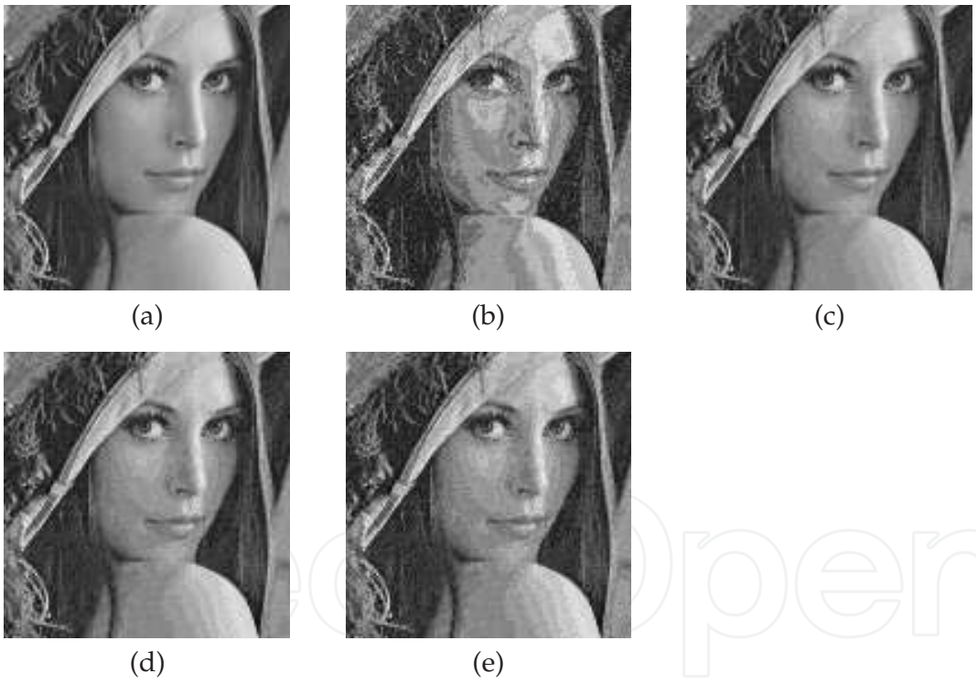


Fig. 12. Experimental result ($k = 4$): (a) zoomed source (128 × 128-pixel part of *Lena*); (b), (c), (d) and (e) transformed images by t_k, g_k, h_k and h'_k, respectively.

generally defined by

$$\mathrm{MSD} = \frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} \{f'(i,\ j) - f(i,\ j)\}^2 \tag{18}$$

where $f(i,\ j)$ and $f'(i,\ j)$ are the respective pixel levels in f and f' at the coordinates of $(i,\ j)$, for $i = 1,\ 2,\ \ldots,\ M$ and $j = 1,\ 2,\ \ldots,\ N$ supposing that both f and f' are $M \times N$ images.

Table 2 lists the measured MSD values for two test images. Although grayscale levels have nonuniform frequency distribution in each input image, the MSD values measured for each transformed image almost agree with the expected values listed in Table 1. Accordingly, the expected values of MSD can be generally used to estimate the actual amount of level change.

(a) Test image *Lena*

Transformation	k			
	2	3	4	5
g_k	2.5	7.5	26.1	91.4
h_k	3.3	11.4	42.8	161.5
h'_k	4.5	18.5	75.7	294.7

(b) Test image *Peppers*

Transformation	k			
	2	3	4	5
g_k	2.5	7.5	25.9	92.5
h_k	3.2	11.4	42.7	162.3
h'_k	4.5	18.5	75.2	296.9

Table 2. Measured values of mean squared difference.

3. Implementation of level transformation

3.1 Limitations on level changes

It is necessary to make level changes caused by a transformation imperceptible both for keeping the embedded watermarks secret and for preserving the image quality. Here, we assume that a range where a pixel level can be changed without being perceived depends only on the own pixel. Then, let us express the upper limit of the range for a source level v as a function only of v, $A(v)$, supposing that $A(v) \geq 0$; that is, a source level v is allowed to change within the range $[\, v - A(v),\ v + A(v)\,]$.

On the above assumption, for a pixel of level v, if the amount of level change caused by a transformation is to be under $A(v)$, the transformation can be actually applied to the pixel. We use the function h_k as the level transformation for watermarking in the rest of this chapter, and let $d_k(v) = h_k(v) - v$. As described in Sec. 2.5.1, for a given v, $d_k(v)$ varies in the range that depends on v, and the upper bound of $|d_k(v)|$ is fixed to Δ_k for any v. Accordingly, we regard $|d_k(v)|$ as the constant Δ_k to determine if the transformation h_k can be applied to a pixel or not. That is, given the function $A(v)$, for a pixel of level v, if $\Delta_k \leq A(v)$ with a certain k, then the level can be changed by $h_k(v)$. Otherwise, the level is left unchanged. The function that keeps v unchanged is expressed by the identity transformation, denoted by $f_I(v)$ such that $f_I(v) = v$.

3.2 Transformation domains

Given the bounding function A of an input level and a value of k, according to the scheme described in the preceding section, each input level v_{in} of the dynamic range is classified into two categories: one that h_k can be applied to, and the other that f_I is to be applied to.

Consecutive input levels of the same category, then, are collected to compose domains of each transformation. The entire dynamic range of input levels, denoted by $U_M = [\,0,\ 2^M - 1\,]$, is consequently expressed as a disjoint union of one or more domains. Let $U_1^{(k)},\ U_2^{(k)},\ \ldots,\ U_{\phi(k)}^{(k)}$ be a sequence of these domains in order, where $\phi(k)$ is the number of domains for the k. Note that the domains of h_k and those of f_I alternate in the sequence.

Next, we modify the domains in U_M so that a blind watermark can be achieve, that is, so that a watermark can be recovered from the transformed image without referring to its source image. Our approach is to make the transformation output ranges disjoint to each other. Let $V_i^{(k)}$ be the range onto which a domain $U_i^{(k)}$ is mapped, for $i = 1,\ 2,\ \ldots,\ \phi(k)$. If these ranges are disjoint mutually, for any output level v_{out}, the range containing it, say, $V_j^{(k)}$, is determined uniquely. This range directly indicates not only the corresponding domain, $U_j^{(k)}$, but also the kind of transformation that is to be applied to the domain. Consequently, it is found whether a pixel is one of those pixels that h_k can be applied to or not. Furthermore, with regard to domains of h_k and the corresponding regions, say, $U_j^{(k)}$ and $V_j^{(k)}$, if that region to which $U_j^{(k)}$ can be mapped by f_I is included in $V_j^{(k)}$, that is,

$$\{f_I(v) \,|\, v \in U_j^{(k)}\} \subseteq V_j^{(k)}, \tag{19}$$

then, for pixels of source level $v_{\text{in}} \in U_j^{(k)}$, we can use either h_k to have the bit-k inverted or f_I to keep the bit-k unchanged.

Based on the input-output relationship of h_k, the domains that have the disjoint relationship among the corresponding regions are defined every $2\Delta_k$ levels in the dynamic range, that is, the boundaries of each domain must be located at the levels of $2m\Delta_k$ $(0 \le m \le 2^{M-k})$. Such domains also satisfy Eq. (19).

3.3 Bit-block watermarking

3.3.1 Data structure for watermarks

As a data structure for representing watermarks, we use bit-planes of a monochrome image or a color component image composed of M-bit pixels, which was also used by Oka & Matsui (1997). As depicted in Fig. 13, first, a source image is divided into pixel-blocks each consisting of N_B pixels. More generally, these blocks can have arbitrary shapes rather than rectangles, and besides, even N_B can be varied in the same image. Then, each pixel-block is regarded as a hierarchy of M bit-blocks. Suppose that a pixel is composed of M bits, a bit $(M-1)$ to a bit 0, that represent the signal level in the natural binary expression. A bit-block k is a set of all the bit k of each pixel in the pixel-block, for $k = 0,\ 1,\ \ldots,\ M-1$.

A watermark bit is represented by a parity value in one of the bit-blocks. Here, the parity value of a set of bits is defined by the sum of the bits in modulo 2. Suppose that to achieve the watermarking, one bit in each bit-block is to be inverted, if necessary, so that the resultant parity value can agree with the watermark bit. The details of the watermarking procedures will be described in the next section.

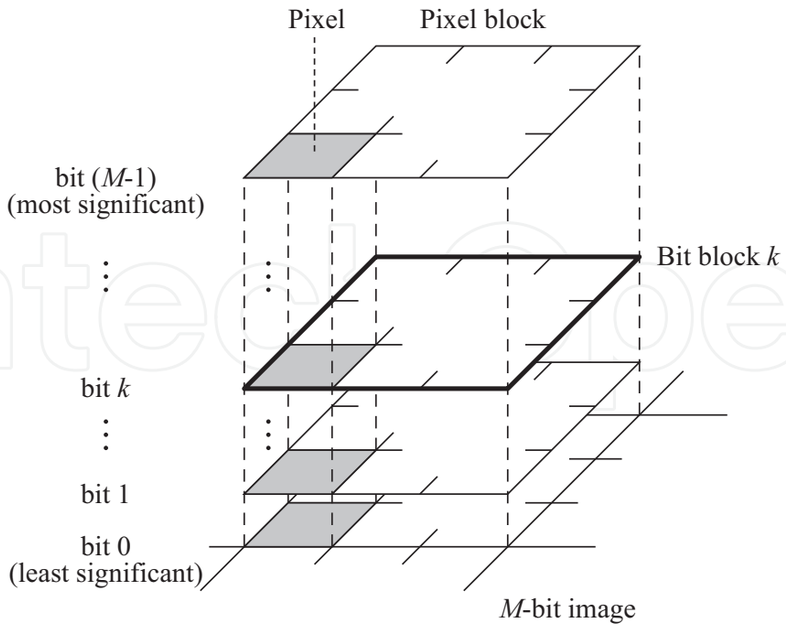


Fig. 13. Data structure for bit-block watermarking.

3.3.2 Watermarking procedures

The procedures for watermarking in the data structure of bit-blocks by using the function h_k are described below. Here, suppose that the domains of h_k in the M-bit dynamic range are given for each value of k. Suppose also that a value of k is assigned to each pixel-block of a source image F. A set of these values of k is associated with the watermarked image and it must be kept secret.

(1) Embedding procedure

Each pixel-block is processed in order. Let $P^{(i)}$ be the ith pixel-block being processed, which is expressed as a sequence of N_B pixels, that is, $P^{(i)} = \{p_1^{(i)}, p_2^{(i)}, \ldots, p_{N_B}^{(i)}\}$, for $i = 1, 2$ and so on. For the value k_i assigned to $P^{(i)}$, the block is processed by the following procedure:

Step 1: Extract from $P^{(i)}$ those pixels whose levels belong to the domains of h_{k_i}. Let $P_e^{(i)}$ be a set of these pixels and n_e be the number of the pixels.

Step 2: Compare n_e with a specified threshold N_E ($1 \leq N_E \leq N_B$). The result determines how to treat the block with a watermark bit as follows:

(a) If $n_e \geq N_E$, we use this block to represent a new watermark bit. Proceed to *Step* 3.

(b) If $n_e < N_E$, we skip this block without using it to represent any watermark bit. Proceed to the next pixel-block.

Step 3: Collect the bits $b(v_j^{(i)} + \Delta_{k_i}, k_i)$ of the pixel $p_j^{(i)} \in P_e^{(i)}$, where $v_j^{(i)}$ is a level of $p_j^{(i)}$, for $j = 1, 2, \ldots, n_e$. Let the resulting bits compose a bit-block, $Q_{k_i}^{(i)}$. Then, calculate the parity value, y, of $Q_{k_i}^{(i)}$.

Step 4: Fetch a new watermark bit, ω, and compare it with y.

(a) If $\omega \neq y$, choose one pixel from $P_e^{(i)}$, and apply h_{k_i} to it.

(b) Otherwise, no operation is done to $P_e^{(i)}$.

Proceed to the next pixel-block.

After all the pixel-blocks of F are processed, the watermark image F' is obtained.

(2) Extracting procedure

Let $P^{(i)\prime}$ denote the ith pixel-block of F' in the form $P^{(i)\prime} = \{p_1^{(i)\prime},\ p_2^{(i)\prime},\ \ldots,\ p_{N_B}^{(i)\prime}\}$. $P^{(i)\prime}$ is processed using the same k_i and N_E as those used for $P^{(i)}$ by the following procedure:

Step 1: Extract from $P^{(i)\prime}$ those pixels whose levels belong to the ranges of h_{k_i}. Let $P_e^{(i)\prime}$ be a set of these pixels and n_e' be the number of the pixels. The disjoint union of the transformation ranges ensures that n_e' is equal to n_e of $P^{(i)}$.

Step 2: Compare n_e' with a specified threshold N_E.

(a) If $n_e' \geq N_E$, proceed to *Step* 3 to find out the watermark bit.

(b) If $n_e' < N_E$, it is found that this block contains no watermark. Proceed to the next pixel-block.

Step 3: Collect the bits $b(v_j^{(i)\prime} + \Delta_{k_i},\ k_i)$ of the pixel $p_j^{(i)\prime} \in P_e^{(i)\prime}$, where $v_j^{(i)\prime}$ is a level of $p_j^{(i)\prime}$, for $j = 1,\ 2,\ \ldots,\ n_e'$. Let the resulting bits compose a bit-block, $Q_{k_i}^{(i)\prime}$. Then, calculate the parity value, y', of $Q_{k_i}^{(i)\prime}$. As a result, y' gives the watermark bit directly. Proceed to the next pixel-block.

The threshold N_E can be altered every pixel-block. Furthermore, being kept secret, N_E is expected to improve the concealment of watermarks.

3.4 Experiments

3.4.1 Domains defined by Weber's law

We demonstrate the defining of transformation domains by using a bounding function for level change, $A(v)$, as described in Sec. 3.1. To specify this function, as an example, we here use Weber's law, which is known as a description of human visual properties for luminance contrast (Jain (1989)). This law states the experimental result that $\Delta L/L$ is constant where ΔL is the magnitude just noticeably different from the surround luminance L. Assuming that this law holds true in the entire M-bit dynamic range, we can express it for a luminance level v in the fractional form:

$$\frac{\Delta v}{v + v_0} = \alpha \tag{20}$$

where v_0 is a fixed level equivalent to a light intensity on human eyes at $v = 0$, and α is a constant known as Weber's ratio. Letting $M = 8$, we now consider 8-bit levels in the range $[\,0,\ 255\,]$. From a result of our preliminary experiment for measuring contrast sensitivity of

human eyes on a liquid crystal display (LCD), we have observed two pairs of approximate values, $\Delta v = 8$ at $v = 192$ and $\Delta v = 4$ at $v = 64$. By using these values in Eq. (20), then, we obtained $v_0 = 64$ and $\alpha = 0.03$, which is comparable to a typical α of 0.02.

Suppose that we can replace Δv with $A(v)$; that is, Eq. (20) is rewritten as

$$A(v) = \alpha(v + v_0). \tag{21}$$

For simplicity, approximating $A(v)$ by $2^{D(v)}$ where $D(v)$ has integers, we have determined $D(v)$ as

$$D(v) = \begin{cases} 3, & 128 \leq v \leq 255 \\ 2, & 0 \leq v \leq 127. \end{cases} \tag{22}$$

Using $A(v)$ with Eq. (22) in the source range of $[\,0,\ 255\,]$, we have defined the transformation domains according to the manner described in Sec. 3.2. Table 3 lists the respective domains of the level transformation h_k and the identity transformation f_I for each k ($k \geq 1$). Under the limitation given by Eq. (22), the range of k such that one or more domains are available for h_k is found out to be $[\,1,\ 4\,]$. Hence, vales of k are to be chosen from this range for each pixel-block in the bit-block watermarking scheme.

	Domains	
k	h_k	f_I
≤ 3	$[\,0,\ 255\,]$	ϕ
4	$[\,0,\ 127\,]$	$[\,128,\ 255\,]$
≥ 5	ϕ	$[\,0,\ 255\,]$

Table 3. Example of transformation domains based on Weber's law.

3.4.2 Examples

We have carried out an experiment of the bit-block watermarking scheme using 8-bit grayscale test images to evaluate visual quality of the resulting images. In the experiment, the location of an isolated pixel to be transformed was fixed at the center of each block. Generally, a pixel to be transformed can be chosen at random among available pixels in each block. Besides, the transformation h_k was always performed for each block. Note that, assuming that a watermark bit is a random variable, a half of the pixels being processed in the experiment are expected to be actually transformed by h_k.

Figure 14 shows an example of the results transformed with pixel-blocks of 3×3 pixels and $k = 4$. Figure 14(b) includes those pixels which were transformed by t_4 and changed by 2^4 levels. These pixels are considerably perceptible. In contrast, those pixels which were transformed by h_4 in Fig. 14(c) have their levels changed by at most 2^3 levels. We can hardly observe any difference between the transformed and source images.

4. Perceptually adaptive watermarking

4.1 Perceptual modeling

4.1.1 Image distortions

Distortions caused by h_k have two phases. The first phase is caused by g_k; although the change in each signal level is made minimum, the number of levels appearing in the transformed

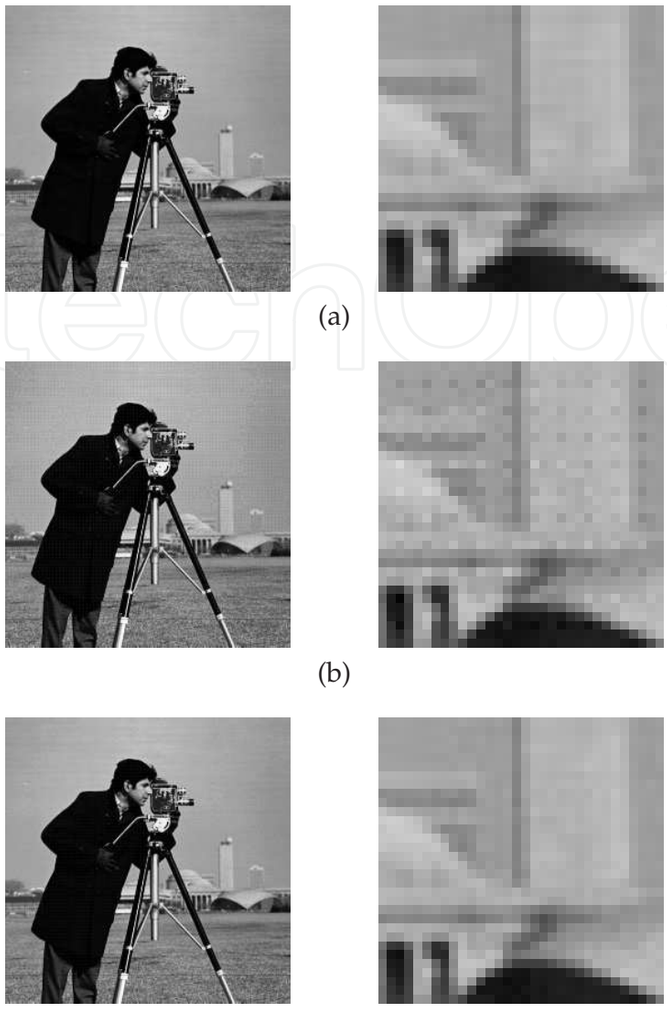


Fig. 14. Results of watermarking with 3×3 pixel-blocks and $k = 4$ on a 8-bit test image *Cameraman*: (a) original image; (b) and (c) watermarked image by t_k and h_k, respectively. The left images are of 256×256 pixels; the right images are enlarged portion of 32×32 pixels of the left images.

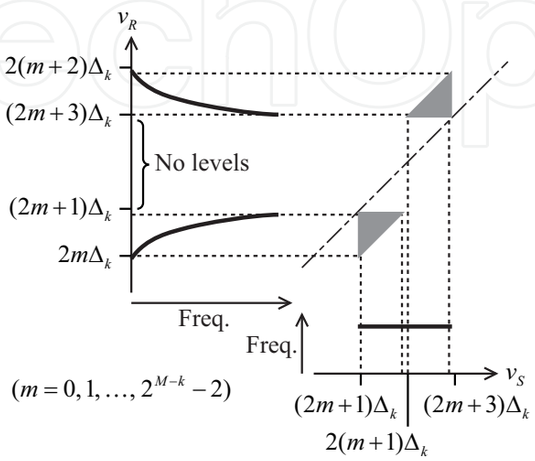
image is reduced. The second phase is caused by the level randomization. In this phase the number of levels appearing in the image increases while the change from the source image increases accordingly. Each phase of distortion affects the image quality in different manners.

(1) Distortion in low-detail image regions

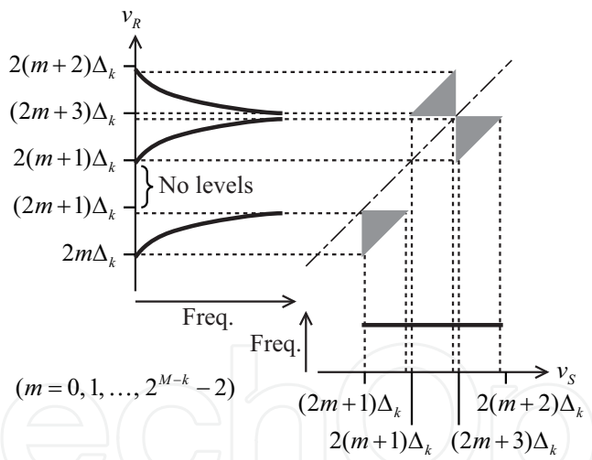
The first phase of the distortion affects particularly the quality of low-detail image areas. Visible false contours are likely to appear in such smooth areas due to the effect similar to coarse quantization. As described in Sec. 2.6.2, randomizing output levels can improve visual quality by making the steps of false contours narrow.

(2) Distortion in high-detail image regions

We consider a local image area where source levels are bounded in a small range. If the source level range is $[2m\Delta_k,\ 2(m+1)\Delta_k)$, the transformed levels lie in the same range, as Fig. 8 has shown. On the contrary, if the source range of $2\Delta_k$ levels is $[(2m+1)\Delta_k,\ (2m+3)\Delta_k)$, the transformed levels lie outside the range and no levels appear inside the original range as shown in Fig. 15(a). If the source range of $3\Delta_k$ levels is $[(2m+1)\Delta_k,\ 2(m+2)\Delta_k)$, there also exists an empty range where no transformed levels appear (Fig. 15(b)). Both the replacement of ranges and the missing of ranges cause distortions in the texture of the local areas.



(a)



(b)

Fig. 15. Transformation h_k from bounded source ranges: Source levels are assumed to occur uniformly in the range.

4.1.2 Objective quality measures

The level transformation h_k causes distortion in the source image, and thus, degrades the image quality. According to the performance analysis of h_k, we have defined two kinds of objective measures to evaluate the distortion.

(1) Change of signal levels

The first measure is the mean squared difference (MSD) between two images, defined by Eq. (18). The MSD value, D_{msd}, of a watermarked image (or an image region) evaluates the mean distortion over the entire area of measurement.

(2) Change of level occurrence distributions

By h_k in every interval of $2\Delta_k$ levels in the input dynamic range, the upper half and the lower half are mapped inversely into the output dynamic range, as already shown in Fig. 9. To evaluate the change in the level occurrence distribution, we define the square variation of level occurrence between a source image X and the transformed image X', denoted by D_{dst}, by

$$D_{\mathrm{dst}} \triangleq \sum_{i=0}^{2^M-1} (\lambda'_i - \lambda_i)^2 \Bigg/ \sum_{i=0}^{2^M-1} \lambda_i^2 \tag{23}$$

where λ_i and λ'_i are the relative occurrence frequencies of a signal level i in the picture X and X', respectively, for $i = 0,\ 1,\ \ldots,\ 2^M - 1$, satisfying $\sum_{i=0}^{2^M-1} \lambda_i = 1$ and $\sum_{i=0}^{2^M-1} \lambda'_i = 1$.

4.1.3 Subjective testing

To find a correlation between the objective qualities and the subjective quality of the transformed images, we have carried out the subjective evaluations by human observers (Kimoto (2008)). In the measurement of image quality, we used a *rating-scale* method (Netravali & Haskell (1988)) where the observers viewed the test images and assigned each image to one of the given ratings. The results were presented by computing a mean value from the numerical values corresponding to the ratings, which is referred to as a Mean Opinion Score (MOS).

The testing materials were prepared in the following way: Here, assuming that $M = 8$, that is, we have only considered 8-bit images.

- The function h_k was implemented with a pseudo-random number generator of a computer in the stochastic process.
- The test images were printed on photographic papers in 200 pels per inch with an image printer, which has a printing resolution of 400 dots per inch.

All the observers, who were all in their twenties, were unfamiliar with the performance of h_k. They were asked to look at the materials sitting at a desk under ceiling lights inside a room.

Two kinds of source images were used in the testing. The first one is a 256-grayscale ramp image where the pixel levels vary linearly from 0 to 255 extending from the top side to the bottom side. Thus, the ramp image represents a low-detail region. The transformation h_k makes the linear gradation of levels distorted in the output image. Accordingly, transforming the source ramp with varying both the value of k and the ratio of pixels chosen to transform in the image, which is referred to as the transformation ratio τ, yields the distorted images of various values of D_{msd}. In the measurement of subjective quality each test image was not compared with the source image, but evaluated from a viewpoint of the appearance of level gradation and assigned one of the five ratings of the absolute rating scale listed in Table 4(b).

The other kind of source image was a granular image representing a high-detail region. The source images used in the measurement were composed of pixels of (pseudo-)random

levels ranging uniformly in bounded intervals of a width of a multiple of Δ_k levels for a given k. According to the stochastic analysis of the mapping characteristics, the transformed images have the same D_{msd} (more strictly, almost same D_{msd} with a small difference due to the pseudo-random numbers) and the different D_{dst} depending on the source intervals. Each transformed image was compared with the source image just printed beside on the same paper and then, evaluated according to the degree of perceptible difference with the impairment rating scale listed in Table 4(a). Thus, eleven scores were collected for each test image, and the MOS was calculated from them.

(a)

Value	Rating
5	Imperceptible
4	Perceptible but not annoying
3	Slightly annoying
2	Annoying
1	Very annoying

(b)

Value	Rating
5	Excellent
4	Good
3	Fair
2	Poor
1	Bad

Table 4. Ratings used in the subjective testing.

4.1.4 Subjective quality measure

In the evaluation of the ramp images, a subjective quality of each test image of a different D_{msd} value was measured in MOS. The result has indicated an approximately linear correlation between the subjective quality and the logarithms of D_{msd}. Consequently, we have observed that D_{msd} has the primary effect on the estimation of MOS for the images transformed by h_k.

In the evaluation of the granular images, those test images transformed with the same k have the same D_{msd} and the different D_{dst}. The result has demonstrated that, for a given k, the MOS values S_{mos} decrease as the D_{dst} increases. Accordingly, we suppose that the correlation between S_{mos} and D_{dst} can be considered approximately linear, while the gradient of linearity varies with the value k.

For a transformed image region, let us consider a subjective quality measure as a function of two parameters, D_{msd} and D_{dst}, that can estimate a subjective quality of the distortion, where D_{msd} and D_{dst} are measured by comparing the distorted image with the source image. According to the above analysis of the measurements, we suppose that a subjective quality measure S_{mos} can be expressed as a linear combination of the logarithm of D_{msd} and D_{dst}: That is,

$$S_{\mathrm{mos}} = \alpha \cdot \ln D_{\mathrm{msd}} + \beta \cdot D_{\mathrm{dst}} + \gamma \tag{24}$$

where α, β and γ are parameters to be estimated by multiple linear regression analysis.

To carry out the regression analysis, we used the triplets of $\{S_{\mathrm{mos}},\ D_{\mathrm{msd}},\ D_{\mathrm{dst}}\}$ measured from the test granular images described above. Thus, 54 measurements of the dependent variable S_{mos} at 12 different values of the independent variable vector $(D_{\mathrm{msd}},\ D_{\mathrm{dst}})$ were obtained and used in the multiple linear regression analysis. As a result, the values of α, β and γ are determined as

$$\hat{\alpha} = -0.46,\ \hat{\beta} = -0.70 \text{ and } \hat{\gamma} = 6.4, \tag{25}$$

respectively. Here, the resulting coefficient of determination, which is commonly denoted by R^2, is 0.86. Let S_e be a predicted value of S_{mos} by Eq. (24) with Eq. (25). Thus, S_e yields values comparable to the MOS values.

To evaluate the effect of D_{dst} on the modeling of S_{mos}, we have also used a simple linear regression model of one independent variable of D_{msd}. In this model, S_{mos} is expressed in the form

$$S_{\mathrm{mos}} = \alpha' \cdot \ln D_{\mathrm{msd}} + \gamma' \tag{26}$$

where α' and γ' are the model parameters. By using the same data as those used in the above multiple regression analysis, these two parameters were estimated from 54 measurements of S_{mos} at four different values of D_{msd} in simple linear regression analysis. As a result, we obtained the estimated values of α' and γ' as

$$\hat{\alpha'} = -0.55 \text{ and } \hat{\gamma'} = 5.7, \tag{27}$$

respectively. Here, the resulting value of R^2 was 0.55. The values of S_{mos} predicted by this model have been compared with the measured values. The relation between the measured and predicted values of S_{mos} of Eq. (24) has higher correlation than that of Eq. (26) as a result. Thus, D_{dst} improves the linear model of S_{mos} as one of independent variables.

4.2 Experiments

4.2.1 Block processing procedures

The subjective quality measure S_e gives an estimate of subjective quality for each image region on the assumption that the entire region is transformed with a given value of k. Let $S_e(k)$ denote the value of S_e resulting from the transformation with the parameter k.

By using the subjective quality measure S_e, we can examine whether the transformation of an image region with a value of k satisfies a given condition of subjective quality. Accordingly, the measure can determine the values of k to transform an image region with so that the desired subjective quality can be achieved. To enhance the difficulty of detecting the values of k in use, the largest one of the available k's is to be chosen. Furthermore, the value of k to use can be changed for each image region.

To implement the above adaptive watermarking, we determine both a threshold value S_T comparable to the subjective quality measure and an upper limit of available k, which is set equal to $M-1$ here for simplicity. The S_T can be determined from the ratings listed in Table 4. Here, let us assume the transformation ratio is 1. Then, the value of k for each image region, k_B, is determined as

$$k_B = \max_{1 \le k \le M-1} \{k \mid S_e(k) \ge S_T\}. \tag{28}$$

The region is actually transformed with k_B. Thereby, the subjective image quality of $S_e(k_B)$ is achieved in the transformed region.

The procedure for implementing Eq. (28) is carried out in each image region, starting from $k = 1$ as follows:

Step 1: Transform the region with the value of k.

Step 2: Calculate D_{msd} by Eq. (18) and D_{dst} by Eq. (23) within the region.

Step 3: Calculate $S_e(k)$ as S_e of Eq. (24) with the specified coefficients of Eq. (25).

Step 4: Compare $S_e(k)$ with the given S_T. If $S_e(k) < S_T$, then, $k_B = k - 1$. Otherwise, increase k by one, and repeat from *Step 1.*

4.2.2 Examples

Simulations of the adaptive watermarking were carried out. The test images of 256×256 8-bit pixels, which are well known as *Lena*, *Peppers*, *Cameraman* and so on, were used as the source images. The adaptively watermarking procedure was implemented in each block of 4×4 pixels in the simulation with various threshold value S_T.

Fig. 16 shows an example of the values of the parameter k_B that were determined for each block by the adaptive scheme. From this figure it is observed that the adaptive scheme performs such that in the low-detail regions such as the *sky*, where the human visual system is sensitive to distortion (Netravali & Haskell (1988)), k_B's of small values are assigned, and in the high-detail regions such as the lower half area of the image, k_B's of large values can be used.

(a) (b) k_B values

Fig. 16. The embedding parameters k_B that are determined for each block by the adaptive scheme: The block size is 4×4 pels and the threshold $S_T = 3.5$; (a) source image *Cameraman* of 256×256 8-bit pels; (b) k_B values, black means $k_B = 1$ and the brighter, the larger k_B.

Table 5 shows for the source image *Peppers* the distribution of the values of k_B that were determined by the adaptive scheme for various S_T. With increasing S_T the ratios of large k_B's decreases, and the average of k_B decreases accordingly.

Table 5 also shows the value of S_e averaged over all the blocks in the image. As this result indicates, the averaged value of S_e was achieved at about 0.5 greater than the given threshold for each of the images in the simulation. The reason why this difference is likely to occur is considered that only integers are available for k.

Examples of the images resulting from the same source image for $S_T = 3.0$, 4.0 and 5.0 are shown in Fig. 17. From these images it is observed that the better visual quality is certainly achieved as the threshold quality S_T is set larger.

4.2.3 Validity of subjective quality measure

We consider the validity of the subjective quality measure in this section. When we look at an image, we usually evaluate the quality of the whole image. Taking account of this fact, we compare the subjective quality measure to human evaluations in terms of the whole image quality.

Using various images produced in the simulation of the adaptive scheme, first, we have carried out the subjective evaluations of visual quality by use of the impairment rating scale listed in Table 4(a). The MOS value S_{mos} was then obtained from the scores of about forty

		Threshold S_T				
		3.0	3.5	4.0	4.5	5.0
Number of blocks	$k_B = 1$	0	0	0	2.3	44.6
(%)	$k_B = 2$	0	0.68	18.7	67.5	52.2
	$k_B = 3$	7.7	54.3	74.9	30.0	3.2
	$k_B = 4$	71.0	44.5	6.5	0.07	0
	$k_B = 5$	21.3	0.51	0	0	0
	$k_B = 6$	0.07	0	0	0	0
Averaged k_B		4.14	3.45	2.88	2.28	1.59
$\overline{S_e}$		3.60	4.10	4.54	4.99	5.46
PSNR (dB)		30.3	34.0	37.3	40.6	44.2

Table 5. Experimental result of perceptually adaptive watermarking for the source image *Peppers*: The block size is 4×4 pixels.

people for each image, who were given no information about the making of the images. Note that these S_{mos}'s are the human evaluation of the whole image quality. Next, to estimate subjective quality of the whole image from a collection of the calculated values of block quality, the block values S_e were averaged over all the blocks in each image, and the mean value $\overline{S_e}$ is obtained.

Fig. 18(a)–(c) shows the relationship between the mean value $\overline{S_e}$ and the measured S_{mos} in each of the three test images. A linear correlation between S_{mos} and $\overline{S_e}$ is clearly observed from either result in this figure. However, the slope of the linear regression line is 1.9, 1.8 and 2.2 in Fig. 18(a), (b) and (c), respectively.

This inclined linear correlation results in the incorrect prediction of subjective quality by the developed measure; for example, as Fig. 18 shows, viewers evaluated the quality of the image at the worst MOS of 1, while the mean value of block quality indicates that the image possesses the quality of MOS of 3.

The value $\overline{S_e}$ can be corrected using the corresponding value of S_{mos} by linear regression analysis. The linear regression model is expressed in the form

$$S_{\mathrm{mos}} = \mu \cdot \overline{S_e} + \nu \tag{29}$$

where μ and ν are to be estimated by simple linear regression analysis. To carry out the analysis, we collected the pairs of $\{\overline{S_e}, S_{\mathrm{mos}}\}$, where $1 \leq \overline{S_e} \leq 5$, from the results of three images shown in Fig. 18(a)–(c). As a result, the parameters μ and ν are estimated as

$$\hat{\mu} = 1.9, \text{ and } \hat{\nu} = -4.4, \tag{30}$$

respectively.

Using Eq. (30), the values of $\overline{S_e}$ in Fig. 18(a)–(c) were modified to $\overline{S_e}'$. The resulting relationships between S_{mos} and $\overline{S_e}'$ are shown in Fig. 18(a)'–(c)'. In this figure, each value of S_{mos} is shown with its 95% confidence interval. The slope of any linear regression line in the figure is about 1.1. Furthermore, from the viewpoint of the confidence intervals of MOS, the linear correlation looks almost valid. Consequently, $\overline{S_e}'$ can be used to predict the evaluation of subjective quality for at least these three images.

(a) Source image *Peppers*

(b) $S_T = 3.0$; $\overline{S_e} = 3.60$, PSNR=30.3 dB

(c) $S_T = 4.0$; $\overline{S_e} = 4.54$, PSNR=37.3 dB

(d) $S_T = 5.0$; $\overline{S_e} = 5.46$, PSNR=44.2 dB

Fig. 17. Results of the adaptive watermarking with the various threshold values S_T: Each right column image magnifies a 64 by 64 pel region of the left one, whose size is 256 by 256 pels.

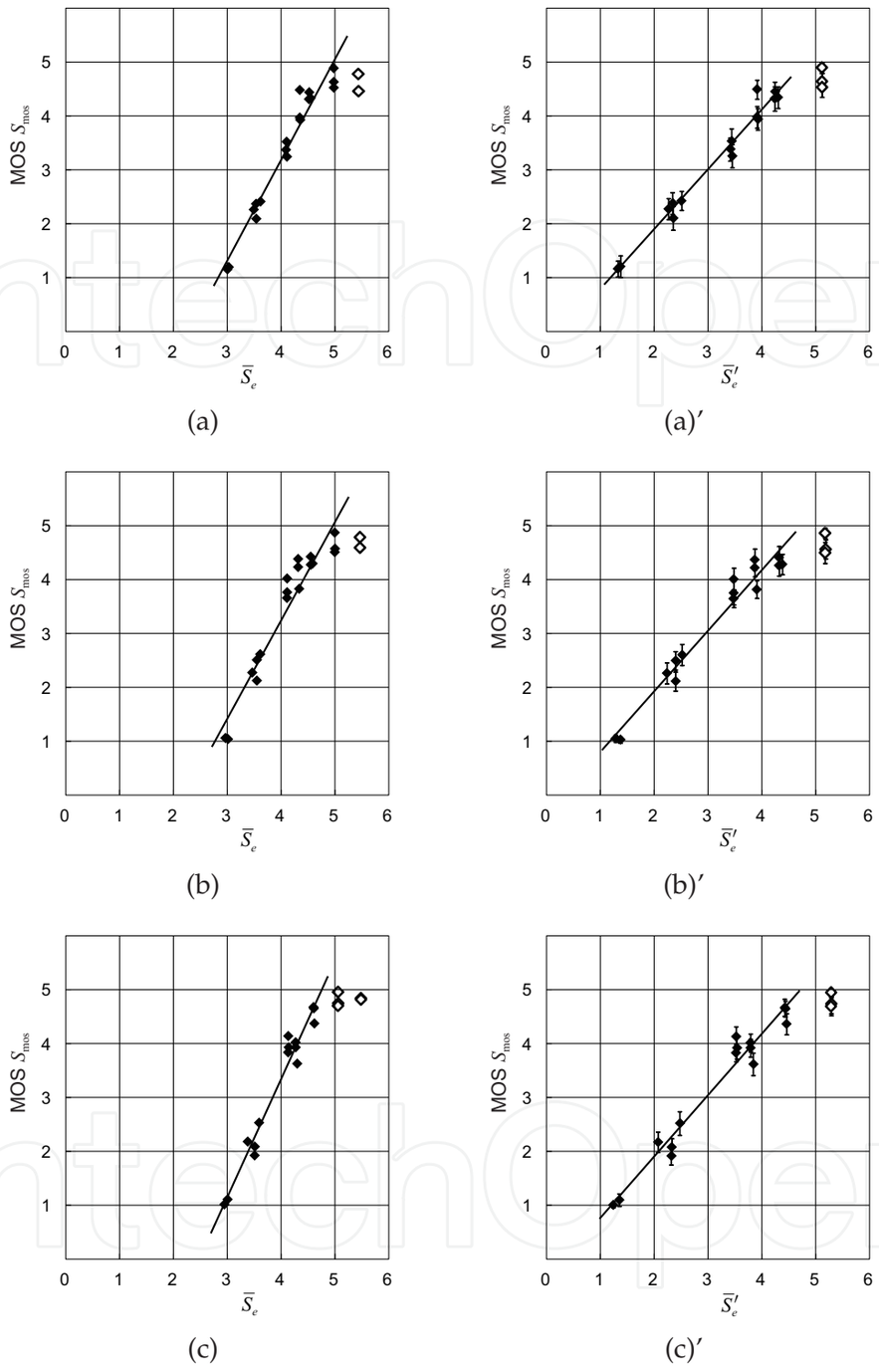


Fig. 18. MOS versus the subjective quality measure averaged over the blocks $\overline{S_e}$ ((a)–(c)) and MOS versus the modified values $\overline{S_e}'$ ((a)′–(c)′) for each source image: (a) and (a)′ source *Lena*; (b) and (b)′ *Peppers*; (c) and (c)′ *Cameraman*; the line in each figure shows the linear regression of the data points; the points painted solid in white are excluded due to out of range $[1, 5]$.

5. Conclusion

The first result of this chapter is the bit inverting transformation, h_k. This level transformation performs all the three functions simultaneously: (a) It represents the inversion of a specified bit; (b) it reduces the level change caused by the bit inversion to the minimum; and (c) it adds a random variation to the output levels under limitations on level changes. The transformed level that has both the specified, say, kth bit inverted and the level change minimized includes the lowest $(k-1)$ bits either of all 1's or all 0's. In contrast, for most of the input levels, some of these bits or all of them are replaced with random bits by randomly varying the transformed levels. Accordingly, the transformed pixels are hard to discriminate without any information regarding the locations in the watermarked images.

The properties of the subjective quality measure, which is the second result of this chapter, are summarized below:

- The subjective quality measure is the function of two objective quality measures, D_{msd} and D_{dst}, which are both obtained from each image region. Consequently, the subjective quality measure is essentially to be carried out in block processing.
- The subjective quality measure estimates a value of MOS in the impairment rating scale. Hence, a threshold value for the measurements can be given in terms of subjective evaluation.
- The values of the subjective quality measure are dependent both on the bit position to be inverted and on the image texture. This property enables the bit position to be altered according to the image texture so as to achieve the adaptive scheme.

The subjective quality measure was derived from the measurements of the computer synthesized test patterns. The validity of the measure for images of natural scenery was examined in terms of the whole image quality. We obtained the whole image quality value by averaging the block quality values over the image. In the experiment using three test images, a highly but inclined linear correlation was observed between the mean value and the actually measured MOS. Although the inclined gradient was successfully corrected by simple linear regression analysis for the images used in this experiment, the cause of the difference between the estimated values and measured values may be related to human eye's characteristics (Netravali & Haskell (1988)). Accordingly, we have to consider a method for estimating the whole image quality from block qualities in more detail.

Another remaining subject is related to the block size of the subjective quality measure. The block size of 4×4 has been used in the experiment. The block size is related to the resolution of quality measure. Besides, because a bit position to be inverted is decided at each block in the adaptive scheme, as many bit positions as the blocks must be stored in a secure manner. From these viewpoints, the appropriate block size should be considered.

6. References

Macq, B. M. ed. (1999). Special issue on identification and protection of multimedia information. *Proc. IEEE*, Vol. 87, No. 7, July 1999, pp. 1059–1276.

Wang, F.H., Pan, J.S. & Jain, L.C. (2009). *Innovations in Digital Watermarking Techniques*. Springer-Verlag, Berlin Heidelberg.

Oka, K., Matsui, K. (1997). Signature method into gray-scale images with embedding function. *IEICE Transactions on Information and Systems* , Vol. J80-D-II, No. 5, May 1997, pp. 1186–1191 (in Japanese).

Kimoto, T. (2005). Implementation of level transformations for hiding watermarks in image bit-planes under limited level changes. *Proc. of the IEEE International Conference on Image Processing (ICIP 2005)* , pp. 253–256. Genova, Italy, Sept. 2005.

Kimoto, T. (2007). Modified level transformation for bit inversion in watermarking. *Proc. of the IEEE International Conference on Image Processing (ICIP 2007)* . San Antonio, USA, Sept. 2007.

Kimoto, T. (2006). A sophisticated bit-conversion method for digital watermarking. *Proc. of the 8th IASTED International Conference on Signal and Image Processing* , pp. 139–144. Honolulu, USA, Aug. 2006.

Kimoto, T. (2009). An advanced method for watermarking digital signals in bit-plane structure. *Proc. of IEEE International Conference on Communications (ICC 2009)* , SPC-P2.8. Dresden, Germany, June 2009.

Awrangjeb, M., Kankanhalli, M.S (2004). Lossless watermarking considering the human visual system. In: *Digital watermarking*, pp. 581–592. Kalker, T., Cox, I.J., Ro, Y.M. eds (2004). Springer, 2004.

Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P (2004). Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Transactions on Image Processing* 13 (4) 600–612.

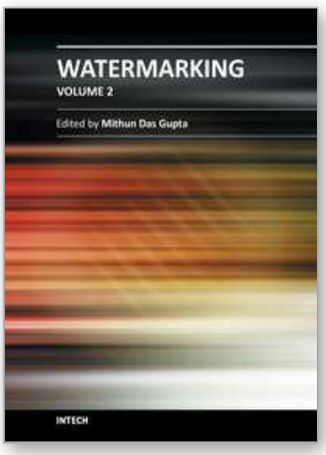
Cox, I.J., Miller, M.L., Bloom, J.A (2002). *Digital watermarking*. Morgan Kaufmann Publishers.

Mikami, D., Shimizu, M., Makabe, S., Kamiyoshihara, Y., Kimoto, T (2008). Measurement of subjective quality of watermarked images made by inverting bits. *Proc. of IEEE TENCON 2008* , O17-7. Hyderabad, India, Nov. 2008.

Netravali, A.N., Haskell, B.G (1988). *Digital Pictures*. Plenum Press, New York, USA.

Jain A.K. (1989). *Fundamentals of digital image processing* Englewood Cliffs NJ, Prentice-Hall.

Kimoto, T., Kosaka, F. (2010). A perceptually adaptive scheme for image bit-inversion-based watermarks. *Proc. of the 6th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS2010)* , pp. 114–122. Kuala Lumpur, Malaysia, Dec. 2010.



Watermarking - Volume 2

Edited by Dr. Mithun Das Gupta

ISBN 978-953-51-0619-7
Hard cover, 276 pages
Publisher InTech
Published online 16, May, 2012
Published in print edition May, 2012

This collection of books brings some of the latest developments in the field of watermarking. Researchers from varied background and expertise propose a remarkable collection of chapters to render this work an important piece of scientific research. The chapters deal with a gamut of fields where watermarking can be used to encode copyright information. The work also presents a wide array of algorithms ranging from intelligent bit replacement to more traditional methods like ICA. The current work is split into two books. Book one is more traditional in its approach dealing mostly with image watermarking applications. Book two deals with audio watermarking and describes an array of chapters on performance analysis of algorithms.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Tadahiko Kimoto (2012). Sophisticated Spatial Domain Watermarking by Bit Inverting Transformation, Watermarking - Volume 2, Dr. Mithun Das Gupta (Ed.), ISBN: 978-953-51-0619-7, InTech, Available from: http://www.intechopen.com/books/watermarking-volume-2/sophisticated-spatial-domain-watermarking-by-bit-inverting-transformation

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen