International Journal of Innovative
Computing, Information and Control
Volume 8, Number 10(A), October 2012

ICIC International ©2012 ISSN 1349-4198
pp. 6849–6868

A ROBUST INFORMED EMBEDDING WITH LOW COMPLEXITY FOR DIGITAL WATERMARKING

Jyun-Jie Wang[1], Houshou Chen[1] and Chi-Yuan Lin[2,*]

[1]Department of Electrical Engineering
National Chung Hsing University
No. 250, Guo Kuang Road, Taichung 402, Taiwan
d9464108@mail.nchu.edu.tw

[2]Department of Computer Science and Information Engineering
National Chin-Yi University of Technology
No.57, Sec. 2, Zhongshan Rd., Taiping Dist., Taichung 41170, Taiwan
*Corresponding author: chiyuan@ncut.edu.tw

Received July 2011; revised December 2011

Abstract. *This paper presents three novel low-complexity informed embedding algorithms based on a modified trellis structure for a digital watermarking system. These algorithms can embed adaptive robust watermarked bits according to various lengths of linear block codes in a host image of* 512×512 *pixels in size. Instead of using randomly generated reference vectors as arc labels, this algorithm uses the codewords of a linear block code to label the arcs in the trellis structure. The use of linear block code as arc offers two advantages as follows: first, to provide a satisfactory space partition for each trellis section; second, to perform embedding algorithm featuring linear block codes for each trellis section. Moreover, the algorithm proposed algorithm can perform iteration to find a tradeoff between robustness and fidelity by using a number of controllable parameters. Finally, the experimental results demonstrate three objectives as follows: first, to simulate the effect of controllable parameter on the proposed watermarking scheme; second, to report the robustness and fidelity performance of this algorithm in various attack channels, such as Gaussian noise, scaling, low pass filter, and JPEG compression; and third, to simulate computation complexity and the proposed trellis-based informed embedding, which requires less operation complexity compared with Miller's informed embedding method.*

Keywords: Data hiding, Informed embedding, Digital watermarking, Robustness

1. **Introduction.** Because of the large number of applications of the Internet and other public communication networks, information hiding has received rising interest, and has played an important role in multimedia technology. The encoding process of data hiding codes, also known as watermarking codes, is to hide or embed a watermark into another host signal, such as a photograph, music, video, or text. The two main requirements of information hiding are fidelity and robustness [1,2], that is, the watermark message must not cause severe degradation on the host signal and must suffer from some common signal processing and channel attacks. Other design criteria for digital watermarking are payload, security, and detectability. Fundamental tradeoffs occur among payload, robustness, and complexity. This study developed practical algorithms by analyzing these tradeoffs, robustness, fidelity and complexity. The applications of digital watermarking include copyright protection [3], fingerprinting tracing [4] media forensics [5, 6], and content authentication [7] and signature verification [8, 9]. Referring to [10, 11] for a comprehensive survey of data hiding codes, the considered watermarking system had no knowledge of the host signal in the receiver, that is, a watermarking system with a blind detector. To

embed a watermark in such a system, a host signal can be viewed purely as noise, called blind watermarking, or exploited as side information, called informed watermarking. The corresponding system with blind detector and informed watermarking can be modelled as communication with side information at the transmitter [12], and allows more effective watermark embedding and detection methods. In general, the encoding process of informed watermarking is divided into informed coding and informed embedding. The purpose of informed coding is to choose a message codeword from a collection of possible candidates to represent this watermark. This message codeword must have minimal perceptual distortion to the host signal compared with other candidates. The informed coding is also known as dirty paper codes [13, 14] or channel coding with side information [15-17], in which the binning scheme is used to achieve the information-theoretic capacity [18, 19]. In informed embedding, the message codeword from informed coding is subsequently modified according to the host signal, attempting to attain an optimal tradeoff between fidelity and robustness in the watermarked image [20-23]. This study focused on the informed embedding method, in which the watermarked image is a function of the watermarked message and the host signal to achieve near optimal robustness and maintain constant fidelity, or vice versa. Miller et al. [22] proposed a suboptimal trellis-based embedding algorithm that starts with the host signal and iteratively constructs an updated watermarked signal toward to the interior of the Voronoi region of the message codeword.

In [22], an informed embedding algorithm used randomly generated reference vectors as arc labels. A disadvantage is that the generated reference vectors can be selected randomly; therefore, the trellis code is not an optimally structured code. In addition, modification of trellis structure modifies such generated reference vectors. Thus, it is impractical to use generated reference vectors as arc labels. Although this [22] trellis-based algorithm can achieve an excellent tradeoff between the fidelity and robustness in watermarked images, this method is computationally intensive and difficult to implement. Instead of using randomly generated reference vectors as arc labels in [22], this trellis structure was modified by using the codewords of a linear block code to label the arcs in the trellis. The advantage of using such linear codewords is that the codewords as arc labels can be more easily produced by use of a generate matrix than randomly generated reference vectors as arc labels. The characteristic of the block codes is subsequently applied to the trellis partition.

This paper proposes a modified trellis structure based on a convolutional code, in which each arc uses a block codeword as arc labels for each trellis section. By using the input number and the memory state of a convolutional code, the embedded structure can modify the capacity and robustness of embedded message. By featuring block codes, both the fidelity and robustness can be changed by a number of parameters. Three embedded algorithms are presented in this paper. The first algorithm is to embed a message based on a memoryless trellis structure, that is, type-1 informed embedding algorithm. The type-1 algorithm is a section-based algorithm, rather than using entire trellis in one iteration. Because of section-based method for type-1 algorithm, the algorithm is performed in each section with iterative operation. The experiment indicates that the algorithm achieves a lower degree of complexity and excellent results under scaling attack, at the cost of robustness. The second algorithm, that is, type-2 informed embedding algorithm, used to improve the embedded distortion and the cost of operation by use of a trellis with memory, demonstrated superior robustness to the first algorithm at the same distortion, which is caused by four types of attacks such as Gaussian noise, low pass filter, and JPEG compression. The third algorithm, that is, type-3 informed embedding algorithm, using the Viterbi decoding for a Hamming metric version, can further reduce the degree

of complexity and leads to a satisfactory error rate performance. These algorithms can be easily implemented with less complexity, compared with other informed embedding methods. The experiment with the proposed algorithm was compared with that in [22]. First, considering embedded distortion, the parameters were simulated as a function of watermarked image quality. Second, we reported the robustness performance of this algorithm in Gaussian noise, low pass filter, scaling, and JPEG compression. Finally, the comparison of complexity was briefly tabulated.

The rest of the paper is organized as follows. Section 2 presents a brief review of trellis-based informed embedding in [22] and introduces the proposed informed embedding method with controllable parameters. Section 3 provides a description of our major work on encoding and decoding of the informed embedding. Section 4 provides experimental results and constructive discussions. Finally, Section 5 offers conclusions.

2. **Trellis-Based Informed Embedding.** The main goal of informed embedding is to find a good watermarked image, which is inside the decoding region of the message codeword, and has minimal perceptual distortion from the host signal. In general, it is difficult to find this optimal watermarked image. However, several approaches are used to find other suboptimal watermarked images, such as trellis-based informed embedding by Miller et al. [22]. Assuming that each path in the trellis corresponds to a message codeword of a watermark, the trellis-based informed embedding in [22] uses the Viterbi decoder to find a good watermarked image. The geometric interpretation of suboptimal embedding algorithm, as illustrated in Figure 1, requires iterative updating of the watermarked signal by running the Viterbi decoder to identify a vector c^1 in the first iteration that has the highest correlation with the current watermarked signal, $x^0 = v$.

By using vectors c^1 and x^0, we subsequently obtained a new watermarked signal x^1 closer to the decoding region of the message codeword w. The embedding process does not terminate until the final watermarked image falls inside the interior of the Voronoi region of w. The final watermarked image of this algorithm, x^2 in Figure 1, may not be the same as the optimal image, x in Figure 1.

This embedding process is time consuming because Viterbi decoding is usually repeated several times before a final watermarked image is obtained. This paper proposes a trellis-based informed embedding with controllable parameters by modifying the arc labels of

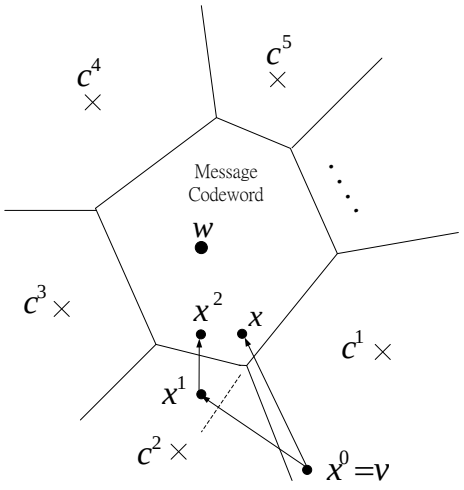

FIGURE 1. A trellis-based informed embedding [22]

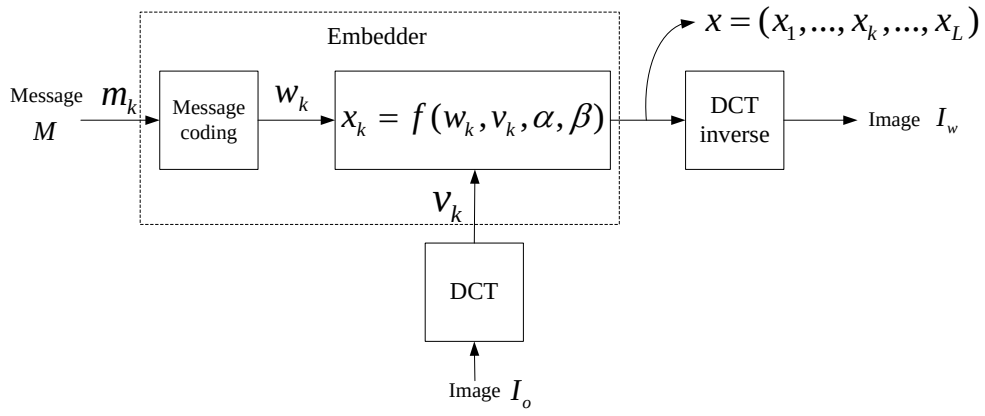

FIGURE 2. Block diagram of informed embedding based on controllable parameters

the trellis structure in [22]. The basic block diagram of proposed embedding method is shown in Figure 2.

As done in [22], the watermark was embedded in the frequency domain of the host signal, rather than on the host image. First, a host signal I_o with dimensions $N = 512 \times 512$ was divided into 4096 blocks of size 8×8; subsequently each block was converted into the frequency domain with the DCT transform. The first 12 low-frequency AC coefficients in each block, shown in Figure 2 of [22], where extracted and concatenated to form the extracted vector v. Every n coefficients of v was subsequently used to embed each bit of an L bits watermark, where $L = 4096 \cdot 12/n$, and forms the watermarked image x. Finally, we replaced the elements of x into their respective DCT coefficients, and converted all DCT blocks back to the spatial domain, called I_w in Figure 2. Because the extracted vector v was available at the transmitter, the output of the informed embedding was denoted by $x = f(w, v, \alpha, \beta)$, where robust factor α and step factor β are controllable parameters for message codeword w and extracted vector v, respectively. The embedding goal aims to satisfy two conflicting criteria, that is, x must be perceptually indistinguishable to v, and x must also be sufficiently close to w to enhance robustness.

3. **The Proposed Informed Embedding Algorithm.** For the proposed informed embedding scheme, this study used section-based embedding algorithm instead of the informed embedding algorithm of [22]. The four inputs to the embedder were the extracted vectors from the host sequence $v = \{v_1, v_2, \cdots, v_L\}$, the message codeword $w = \{w_1, w_2, \cdots, w_L\}$, and the controllable factors α and β, where the v_k and w_k are vectors of length n with $1 \leq k \leq L$. The parameters α and β control the quality of the watermarked image regarding fidelity and robustness.

The output of the embedder, watermarked sequence x, was subsequently passed through the attack channels, such as the Gaussian noise, and JPEG compression, as illustrated in Figure 2. The decoder produced the watermark estimate $\widehat{m} = g(y)$, where y is the extracted vector of the received signal after the channel distortion, as illustrated in Figure 2. The proposed informed embedding algorithm was based on trellis partition. In time k, the extracted vector v_k of n components is one of the a real space of dimension n. The real space of dimension n was partitioned into 2^m regions by a (n, m) linear block codes Γ in each trellis section. We used a simplex as linear block code. The purpose of using the simplex code is to obtain excellent robustness and space partition. Each trellis section is a mapping from the real space to the code space, which is represented by a codeword

index set. It is mapped B is mapped as

$$B : R^n \longrightarrow \{c_1, c_2, \cdots, c_{2^m}\}$$

where $\Gamma = \{c_1, c_2, \cdots, c_{2^m}\}$ denotes the set of 2^m disjoint regions. Each region in the partition is associated with a represented codeword. The set of represented codewords is referred to as the object w_k of an extracted vector v_k. In this study, the measure of distortion mean-squared error (MSE) distortion was as follows:

$$d(x_k, w_k) = E\left[|x_k - w_k|^2\right]$$

where x_k is an arbitrary vector over R^n, and w_k is a message codeword in the kth trellis section. In general, the $d(x_k, w_k)$ common choice is the Euclidean distance or hamming distance.

In the proposed informed embedding, we used a convolutional code to construct a trellis; and subsequently, the codewords of a linear block code as the arc labels in the trellis. First, the trellis structure of a binary (n_{out}, k_{out}, ν) convolutional code with 2^ν states in every depth was constructed, where ν is the memory of the convolutional code. Each trellis section contained $2^{\nu+k_{out}}$ arcs, and these arcs were subsequently labeled by all codewords of a linear block code $\Gamma(n, \nu + k_{out}, d)$, where d is the minimal distance of the code. In this structure, the number of bits embedded in each section is represented as k_{out}. With v as the number of memory and a controllable parameter to tune robustness, the number of trellis state equals 2^v. Parameter v can be adjusted to enhance the robustness, which increases the complexity. In contrast, a convolutional code with a larger value of k_{out} was chosen to increase the embedding capacity. In the proposed trellis structure, a real space of n dimensions was divided into $k_{out} + v$ blocks in each section corresponding to a simplex codeword. With the outer codes as the convolutional codes, the change of k_{out} leads to a simplex code of longer length. Maintaining $k_{out} + v$ constant, that is, n constant, k_{out} was tuned to increase the capacity, and v was tuned to enhance the robustness. Figure 3 shows the arc labels of the kth trellis section, in which the arc label m_k/w_k denotes the watermark message m_k and the message codeword w_k. This code trellis was obtained from a $(2, 1, 2)$ convolutional code, and the labels of the trellis arcs are the codewords of a $(7, 3, 4)$ simplex code. Using the trellis structure in Figure 3 can obtain an adequate space partition for n dimension real space. We considered the following three informed embedding algorithms.

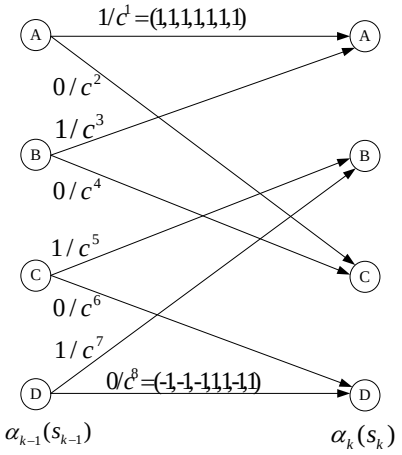

FIGURE 3. Trellis with eight arcs labelled by a $(7, 3, 4)$ simplex code

3.1. **Informed embedding without memory, type-1.** Let $w = (w_1, \cdots, w_L)$ be a valid path of the trellis, encoded from the watermark $m = (m_1, \cdots, m_L)$, and $v = (v_1, \cdots, v_L)$ the extracted vector from the host signal. Each vector w_k is a selected codeword of length n in Γ. The embedder produces a watermarked sequence $x = \{x_1, x_2, \cdots, x_L\}$ by a section-by-section trellis-based function $x_i = f(w_i, v_i, \alpha, \beta)$, $1 \leq i \leq L$, where step factor $\beta \in [0, 1]$ and robust factor $\alpha \geq 1$. The geometrical interpretation of the proposed embedding algorithm in the kth section is shown in Figure 4, in which the kth component of watermarked image was iteratively updated toward to the decoding region of w_k.

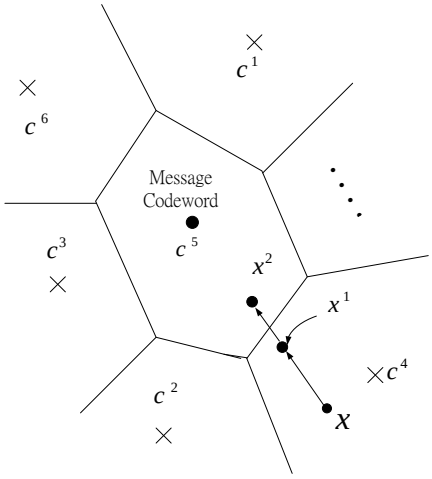

FIGURE 4. A geometrical interpretation of proposed informed embedding in kth trellis section

In the kth section of the trellis, we modified the kth component of the extract vector v_k to form the kth component of the watermarked image x_k iteratively. The proposed informed embedding attempts to find x_k to minimize the degradation of x_k from v_k, and simultaneously be closer to αw_k, compared with other candidates αc, $c \in \Gamma$, i.e.,

$$d(\alpha w_k, x_k) \leq d(\alpha c, x_k), \quad c \in \Gamma \text{ and } c \neq w_k, \tag{1}$$

where $d(a, b)$ is the Euclidean distance between a and b. The detailed procedure of finding such x_k is illustrated as follows.

Let h_k be the sign vector between v_k and w_k. That is for each component of v_k and w_k, we define

$$h_{k,i} = \text{sgn}(v_{k,i} \cdot w_{k,i}), \quad 1 \leq i \leq n, \tag{2}$$

where $\text{sgn}(a) = 1$ if $a \geq 0$ and $\text{sgn}(a) = -1$ if $a < 0$. Subsequently, we construct the ith component of x_k as follows: if $h_{k,i} = 1$, then $x_{k,i} = v_{k,i}$, and if $h_{k,i} = -1$, then

$$x_{k,i} = \begin{cases} v_{k,i} - \beta \cdot d(\alpha w_k, v_k), & \text{if } v_{k,i} \geq 0 \\ v_{k,i} + \beta \cdot d(\alpha w_k, v_k), & \text{if } v_{k,i} < 0. \end{cases} \tag{3}$$

In other words, we move v_k toward to w_k by a distance $\beta d(\alpha w_k, v_k)$ for those positions in which v_k and w_k are of opposite signs. If current x_k satisfies (1), we then move on to the $(k+1)$-section, otherwise we substitute v_k by current x_k and repeat the procedures in (2) and (3). The proposed informed embedding causes perceptual degradation of the host signal for distinct α and β, and we can thus adjust the value of α and β to achieve excellent tradeoff between the fidelity and robustness in watermarked images.

The proposed informed embedding algorithm is summarized as follows.

1. Let $k = 1$ and initialize $x_k = v_k$ with a choice of a robust parameter $\alpha \geq 1$ and step parameter $\beta \in [0, 1]$.
2. If the current x_k satisfies the criterion (1), move to Step 4, otherwise substitute v_k by x_k.
3. Update the kth watermarked image x_k by (2) and (3), and move to Step 2.
4. If $k = L$ then terminate, otherwise let $k = k + 1$ and $x_k = v_k$, and move to Step 2.

For example, an type-1 informed embedding is illustrated with a (7,3,4) simple code with $\alpha = 10$ and $\beta = 0.1$ in Table 1 and Table 2.

TABLE 1. The modified value of type-1, $\alpha = 10$, $\beta = 0.1$ and $d(\alpha w_k, v_k) = 31.733$

i	1	2	3	4	5	6	7
$v_{k,i}$	–8	2	2	–1	–3	1	–1
$\alpha w_{k,i}$	10	–10	–10	10	–10	10	10
$x_{k,i}$	–4.827	–1.173	–1.173	1.173	–3	1	2.1733

TABLE 2. The iterative procedures of type-1

x_k	initial	first iteration	second iteration
c^1	24.673	25.012	25.658
c^2	25.536	25.799	26.426
c^3	26.210	26.563	27.172
c^4	23.811	24.199	24.866
c^5	31.733	25.393	**23.999**
c^6	32.357	30.637	29.492
c^7	31.733	29.978	28.806
c^8	26.962	29.539	28.349

$M = \{1, 2, 3, 4, 7\}$ positions is the complement sign and thus, host v_k where $k \in M$ must be modified. First, the decoding region of the initial $x_k = v_k$ falls into c_4 and the type-1 algorithm achieves the object. In the first iterative procedure, the modified value x_k remains in region c_4. Finally, the decoding region falls into the object region c_5, when we maintain the work of type-1 algorithm until the second iterative procedure. The modified signal x_k is the closest to the region c_5 than others codewords, and the robustness measure is the distance of 23.9 between modified signal x_k, which should be run continuously to further close the object codeword c_5.

Because the embedding algorithm is performed by independent section, the type-1 algorithm can generate high robustness. Although type-1 algorithm is of superior robustness, the number of iterative operations increases. To decrease the complexity of iterative operation for type-1 algorithm, we modified the type-1 algorithm as memory version.

3.2. **Informed embedding with memory, type-2.** The algorithm in 3.1 has less complexity because the modification of x_k is executed section-by-section in the trellis without the accumulation of the distortion in the first $(k - 1)$ components between the message codeword and the current watermarked image. We propose another informed embedding scheme by use of Viterbi algorithm that accumulates the total perceptual distortion to improve the fidelity between the message codeword and the current watermarked image. First, we started with $x = v$, and initialize the state metric $\alpha_0(s_0 = 0) = 0$ and $\alpha_0(s_0 \neq 0) = \infty$. At the kth time unit of the trellis, we defined the accumulation metric

attributed to the label $c(s_{k-1}, s_k) \in \Gamma$ as the addition of the arc metric in $c(s_{k-1}, s_k)$ to the previously stored state metric $\alpha_{k-1}(s_{k-1})$, i.e.,

$$J_c(s_{k-1}, s_k) = \alpha_{k-1}(s_{k-1}) + d(\alpha c(s_{k-1}, s_k), x_k). \tag{4}$$

Instead of using (1) as the criterion for the kth component of the current watermarked sequence in the previous algorithm, we used the accumulation metric (4) as the criterion. That is, in the kth section, we continued to update x_k until the accumulation metric with respect to the message codeword w_k was smaller than the accumulation metric with respect to any other codeword in Γ, i.e.,

$$J_{w_k}(s_{k-1}, s_k) < J_c(s_{k-1}, s_k),\ c \in \Gamma \text{ and } c \neq w_k. \tag{5}$$

The watermarked image based on the criterion (5) may lie in the boundary of the decoding region of the message codeword. We can enhance the robustness of the watermarked image by

$$R_k = \min_{c \in \Gamma, c \neq w_k} \{J_c(s_{k-1}, s_k)\} - J_{w_k}(s_{k-1}, s_k). \tag{6}$$

Several methods are used to choose R_k. We first choose a constant threshold R, and subsequently let $R_k = k\frac{R}{L}$. We used the same procedure as in 3.1; in particular, Equations (2) and (3) were used to update the current x_k until x_k satisfied the criteria (5) or (6). Moreover, after the final x_k for the kth section was found, we then updated each kth state metric in the k-section by

$$\alpha_k(s_k) = \min_{s_{k-1}} \{\alpha_{k-1}(s_{k-1}) + d(\alpha c(s_{k-1}, s_k), x_k)\}, \tag{7}$$

where the minimum is taken over those s_{k-1} connected to s_k. This informed embedding algorithm with distortion accumulation is summarized as follows:

1. Let $k = 1$ and initialize $x_k = v_k$ with a choice of a robust parameter $\alpha \geq 1$, and step parameter $\beta \in [0, 1]$.
2. If the current x_k satisfies the criterion (5), (6) respectively, move to Step 4, otherwise substitute v_k by x_k.
3. Update the kth watermarked image x_k by (2) and (3), and move to Step 2.
4. If $k = L$, terminate; otherwise let $k = k + 1$ and $x_k = v_k$, update the state metric $\alpha_k(s_k)$ by (7), and move to Step 2.

Figure 9 shows the robustness results of informed embedding for various algorithms over addition of Gaussian noise. For the variance greater than 100, the BER of type-2 algorithm is close to that of Miller for robustness measure. However, for variance lower than 100, the BER of type-2 informed embedding algorithm is inferior to that of Miller, and the complexity of type-2 algorithm is lower than that of others.

3.3. **Informed embedding without memory using hamming metric, type-3.** In each trellis section, $v_{k,i}$ is modified into $x_{k,i}$ when $v_{k,i} \neq w_{k,i}$; the system exhibits a superior robustness as $v_{k,i}$ approaches $w_{k,i}$. The minimal hamming distance with the linear block code was employed as fidelity metric. The hamming distance was defined in a different manner from the Euclidean distance. With the metric distance measured as the Euclidean distance between both $v_{k,i}$ and $w_{k,i}$, the hamming distance was just measured as the number of distinct bits. In the event that $v_{k,i}$ is far away from $w_{k,i}$, the distortion in a Euclidean distance measurement is greater than that in a hamming distance measurement. Hamming distance provided by

$$d_H(x_i, w_i) = |\{k : \text{sgn}(x_{k,i}) \neq w_{k,i}\}|. \tag{8}$$

Let h_k be the sign vector between v_k and w_k; that is, for each component of v_k and w_k, we define

$$h_{k,i} = \operatorname{sgn}(v_{k,i} \cdot w_{k,i}), \quad 1 \le i \le n, \tag{9}$$

where $\operatorname{sgn}(a) = 1$ if $a \ge 0$ and $\operatorname{sgn}(a) = -1$ if $a < 0$. The ith component of x_k is constructed as follows: if $h_{k,i} = 1$ then $x_{k,i} = v_{k,i}$, and if $h_{k,i} = -1$, all of the $|v_{k,i}|$ can be arranged to form a set Λ as follows:

$$\Lambda = \operatorname{sort}\{|v_{k,i}|\} = \{\widetilde{v}_{k,1}, \widetilde{v}_{k,2}, \cdots, \widetilde{v}_{k,l}\}, \tag{10}$$

where l is a constant, $t \le l$ and $|\widetilde{v}_{k,1}| \le |\widetilde{v}_{k,2}| \le \cdots \le |\widetilde{v}_{k,l}|$. With the codeword of the linear block code in each trellis section as the label, and t as the error correcting capability, t represents the number of the unchanged components and the lower $l - t$ values of $\widetilde{v}_{k,i}$ in Λ are modified. If it is required to modify such $l - t$ to remain as $\widetilde{v}_{k,i}$, a simple "out of phase" rule can be used, illustrated as follows:

$$x_{k,i} = \begin{cases} -R, & \text{if } \widetilde{v}_{k,i} \ge 0 \\ R, & \text{if } \widetilde{v}_{k,i} < 0, \end{cases} \tag{11}$$

where R is a constant. The type-3 informed embedding algorithm is summarized as follows:

1. Let $k = 1$ and initialize $x_k = v_k$ with a choice of a robust parameter R and error correcting capability t.
2. Generate Λ by use of (9) and (10).
3. If $|\Lambda| > t$, then let $k = k + 1$ and move to Step 2. Otherwise, move to the next step.
4. Reserve t components, and use (11) to modify the remaining $l - t$ components in Λ. Let $k = k + 1$ and move to Step 2.

According to this preceding algorithm, a $d(v_{k,i}, x_{k,i})$ degree of distortion occurs, which is independent of the Euclidean distance, but related to the hamming distance between $v_{k,i}$ and $x_{k,i}$. A path w closest to y can be found using the Viterbi algorithm to decode at the receiver. The received vector y is used for hard decision, followed by Viterbi decoding. Although the decoding performance with a hard Viterbi algorithm cannot parallel that by a soft type, the embedding coding with decoding performance can be improved at the same level of distortion based on the structure referred to in Section 3.1.

The idea is illustrated as follows: assuming the seven extracted vectors

$$v_k = (7.9, -5.0, 2.2, 7.7, -0.7, -0.2, 1.1),$$

the codeword of the (7,3,4) simplex code is chosen as the label of the trellis structure. With the minimal distance of simplex code as 4, the number of correctable bits is $t = \lfloor (4-1)/2 \rfloor = 1$. Setting R to 0.1 and assuming the message codeword $w_k = (1, -1, -1, 1, -1, 1, 1)$ in the section k, the modification of $v_{k,i}$ into $x_{k,i}$ is tabulated as follows: when $h_{k,i} = 1$ and $x_{k,i} = v_{k,i}$, that is, $v_{k,i}$ and $w_{k,i}$ are of the same sign, $v_{k,i}$ remains unchanged. The rest of $v_{k,i}$ of opposite signs are arranged in order as

$$\Lambda = \{\widetilde{v}_{k,6}, \widetilde{v}_{k,3}\} = \{-0.2, 2.2\}$$

$\widetilde{v}_{k,3}$ remains unchanged for the maximal $t = 1$ components, that is, $x_{k,3} = \widetilde{v}_{k,3}$. The remainder of $\widetilde{v}_{k,6}$ are defined as

$$x_{k,6} = \begin{cases} -0.1, & \text{if } \widetilde{v}_{k,6} > 0 \\ 0.1, & \text{if } \widetilde{v}_{k,6} \le 0 \end{cases}$$

From this equation, $x_{k,6} = 0.1$. Finally, x_k is presented in Table 3.

The algorithm proposed in this section uses preceding procedure to embed information. A low level of distortion is acquired by use of the hard decoding at the receiver. The

TABLE 3. The modified value of type-3

i	1	2	3	4	5	6	7
$v_{k,i}$	7.9	–5.0	2.2	7.7	–0.7	–0.2	1.1
$w_{k,i}$	1	–1	–1	1	–1	1	1
$h_{k,i}$	1	1	–1	1	1	–1	1
$x_{k,i}$	7.9	–5.0	2.2	4.7	–0.7	0.1	1.1

reasons that such algorithm requires less embedding complexity than those referred to in the previous section are stated as follows:

1. It is not essential to accumulate metric as compared with Viterbi algorithm when performing trellis operation in each section.
2. Iteration is not required when performing trellis operation in each section. It requires only one time of sign reverse.
3. Viterbi decoding with hamming metric is used at the decoder.

Consequently, the execution complexity is reduce substantially.

3.4. **Detection of message.** The proposed embedding system was built by a trellis code, and each path through this trellis represents a specific watermark message. We subsequently executed the Viterbi decoder to find an optimal path of the highest correlation with the detected sequence, $\mathbf{y} = \{y_1, \cdots, y_L\}$. Finally, the watermark message can be identified from this optimal message codeword. The proposed system is to build a trellis code, as illustrated in Figure 3. Each path through this trellis represents a specific message. Because two arcs exit each node, 2^L possible paths occur, with L as the length of the message M. Thus, the system encodes L bits. Linear block codes and convolutional codes have a natural trellis structure, in which every path represents a codeword. Let $S = \{s_0, s_1, \cdots, s_L\}$ represent one state sequence in a L-section trellis T. In kth trellis section, the system is performed by convolutional codes at a rate of $1/2$, which presents the transmission scheme, and data is packetized as a codeword w_k of n bits. $w_k = (c_{k,1}, c_{k,2}, \cdots, c_{k,n})$ represents a n code bits sequence per block codeword at time k with each bit $c_{k,i} \in \{-1, 1\}$, and $\mathbf{y}$ is the received vector through attack channel. Viterbi algorithm [24] is the most common approach to decode trellis codes, and maintains the optimal algorithm in regard to the maximum-likelihood criterion of the sequence. If the S is equally distributed, it is known that the Viterbi algorithm is optimal with respect to the MAP sequence criterion. By generalizing the Viterbi algorithm, we obtain

$$P(S|\mathbf{y}) = \frac{P(\mathbf{y}|S)P(S)}{P(\mathbf{y})} \tag{12}$$

where $P(\mathbf{y})$ denotes a constant with respect to the MAP sequence of states such that

$$\begin{aligned}
\arg\max_{S\in T} p(S|\mathbf{y}) &= \arg\max_{S\in T} p(\mathbf{y}|S)p(S) \\
&= \arg\max_{S\in T} \left(\ln \prod_{k=1}^{L} p(y_k|s_{k-1}, s_k) \right. \\
&\quad \left. + \ln \prod_{k=1}^{L} p(s_{k-1}|s_k) \right) \\
&= \arg\max_{S\in T} \left(\ln \prod_{l=1}^{L} \prod_{i=1}^{2^m-1} p(y_{k,i}|s_{k-1}, s_k) \right.
\end{aligned}$$

$$+\ln\prod_{k=1}^{L}p(s_{k-1}|s_k)\Bigg)$$
$$=\arg\max_{S\in T}\sum_{k=1}^{L}\Bigg(\sum_{i=1}^{2^m-1}\ln(p(y_{k,i}|s_{k-1},s_k))$$
$$+\ln(p(s_{k-1}|s_k))\Bigg) \tag{13}$$

The Viterbi algorithm [24] is the most efficient method to find an optimal state sequence S^*, or an equivalent optimal path, with respect to maximum-likelihood criterion. In command, the prior probability is usually equally distributed or unknown; therefore, the equation is as follows:

$$\begin{aligned}S^* &= \arg\max_{S\in T}p(S|\mathbf{y})\\ &=\arg\max_{S\in T}\sum_{k=1}^{L}\left(\sum_{i=1}^{2^m-1}\ln(p(y_{k,i}|s_{k-1},s_k))\right)\\ &=\arg\max_{S\in T}\sum_{k=1}^{L}\ln p(y_k|s_{k-1},s_k)\\ &=\arg\min_{S\in T}\sum_{k=1}^{L}d(y_k,w_k)\end{aligned} \tag{14}$$

where $d(y_k,w_k)=|y_k-w_k|^2$ is the kth arc metric between states s_{k-1} and s_k in the trellis. $\sum_{i=1}^{2^m-1}\ln(p(y_{k,i}|s_{k-1},s_k))$ is ML algorithm metric to the BSC form as

$$\arg\max_{S\in T}\sum_{k=1}^{L}\sum_{i=1}^{2^m-1}\ln(p(y_{k,i}|s_{k-1},s_k))$$
$$=\arg\max_{S\in T}\sum_{k=1}^{L}d(y_k,w_k)\ln p_e+[2^m-1-d(y_k,w_k)]\ln(1-p_e)$$
$$=\arg\min_{S\in T}\sum_{k=1}^{L}d(y_k,w_k)\ln\frac{p_e}{1-p_e}+C_1 \tag{15}$$

where the metric $d(y_k,w_k)$ is hamming distance and the p_e is the bit error probability. The metric of the soft Viterbi decoding algorithm in Sections 3.1 and 3.2 is estimated as (14), whereas that in Section 3.3 is estimated by (15). Equation (15) requires a lower decoding complexity than a hardware decoding; however, the performance is not as good as that of a software Viterbi decoding algorithm. However, a low level of distortion was obtained by using the algorithm proposed in Section 3.3. Attacked in a channel, a remarkable bit error rate is attained for the same distortion. It provides a superior bit error rate and robustness to those of hardware decoding.

4. **Simulation Results.** As done in [22], a host signal with dimensions $N=512\times512$ was first divided into 4096 blocks of size 8×8; subsequently, each block was converted into the frequency domain using its DCT transform. The first 12 low-frequency AC coefficients in each block, shown in Figure 3 of [22], were extracted and concatenated, and every $n=31$ coefficient was subsequently used for embedding each bit of a watermark of $L=4096\cdot12/31=1585$-bits. The trellis was constructed by a $(2,1)$ convolutional

code, and the labels of the trellis arcs were a $(31, 5)$ simplex code. Subsequently, the watermarked image quality was defined as

$$\mathrm{PSNR} = 10 \log_{10} \frac{255^2}{\mathrm{MSE}},$$

where MSE represents the mean square error between the original image I_0 and watermarked image I_w as

$$\mathrm{MSE} = \frac{1}{\mathrm{N}} \sum_{i=1}^{512} \sum_{j=1}^{512} (I_o(i,j) - I_w(i,j))^2.$$

The robustness for Gaussian noise and JPEG compression was evaluated as follows:

1. Fidelity experiments
   (a) Parameters α and R
   Simulating the fidelity by aiming at α (the parameter of informed embedding) without an attack channel. The PSNR, as shown in Figure 5, was presented as a function of α and decreased with α; however, the proposed algorithm required a higher number of iterations. The image quality (PSNR) obtained by the algorithm of type-2 was 2-3 dB higher than that by the algorithm of type-1 when varying α.

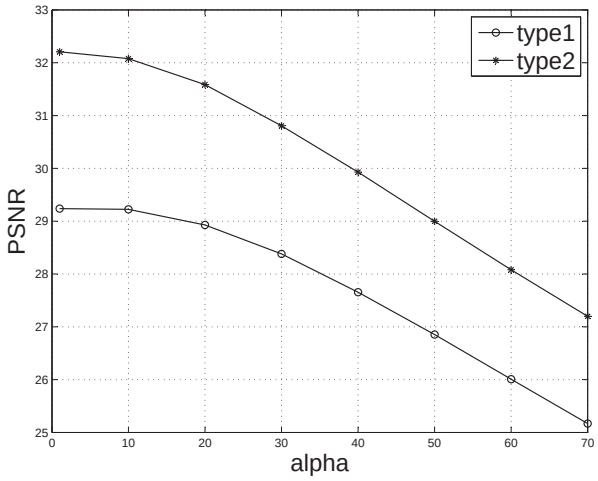

FIGURE 5. Fidelity experiments with variant α

   The robustness was improved at the expense of PSNR when increasing the value of α in the algorithm of type-1 and type-2. The embedded image quality, associated with a variety of robustness parameters R between type-1 and type-2, is compared in Figure 6. It is demonstrated that, in type-2, the image quality decreased with the value of R; however, a larger value of R results in a superior bit error rate. In contrast to type-1, the increase of R does not result in an inferior image.
   (b) Parameters β
   The dependence of image quality on the parameter β, the iteration step factor, is depicted in Figure 7. The higher the value of β, the lower the number of iteration times required to reach the expected robustness of objective codeword, with degrading of the image quality. Therefore, the value of β can be varied to change the operation complexity when the algorithms of type-1 and type-2 are performed.

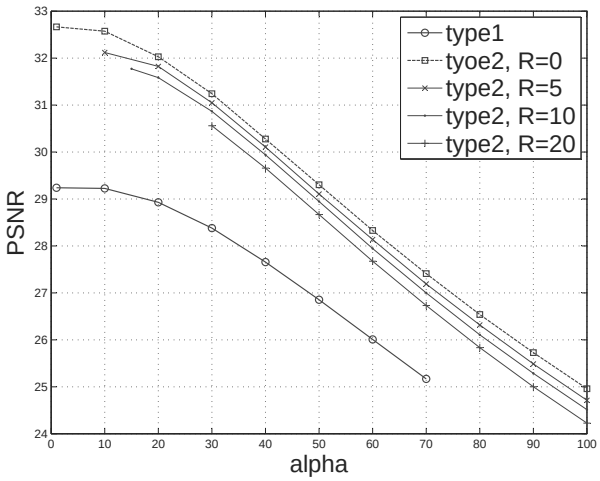

FIGURE 6. Fidelity experiments with variant parameters R

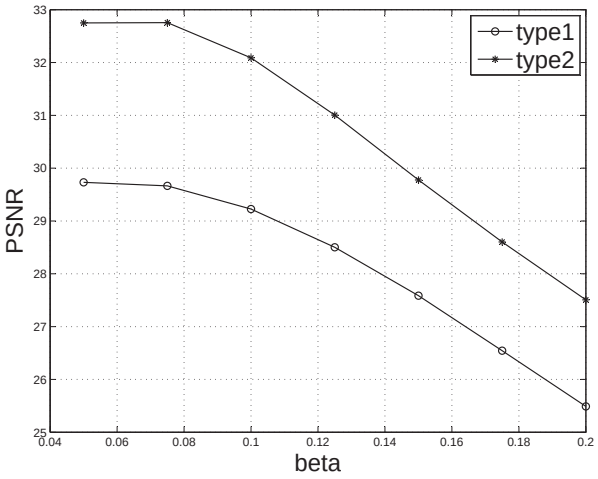

FIGURE 7. Fidelity experiments with variant β

(c) Parameters R of type-3
The simulated PSNR is depicted in Figure 8 as a function of R by use of the algorithm of type-3. That is, the same sign as α, an increase of R results in an enhanced robustness at the expense of image quality.

2. The experiment on an attack channel
(a) AWGN
Considering real-valued x and y, the receive pixel y over an attack channel is given by

$$y = x + w,$$

where w is additive white Gaussian noise (AWGN), distributed as $N(0, \sigma_w^2)$. Gaussian noise variance σ_w^2 was added to each pixel of the watermarked images. The experiment was repeated for variance σ_w^2, and the BER was computed. The result is shown in Figure 9.
Figure 9 shows the results of testing various levels of noise with variance ranging from 50 to 400 obtained from the proposed algorithm. With PSNR $\approx$ 30 dB in each case, the figure shows that the BER of the proposed algorithm with

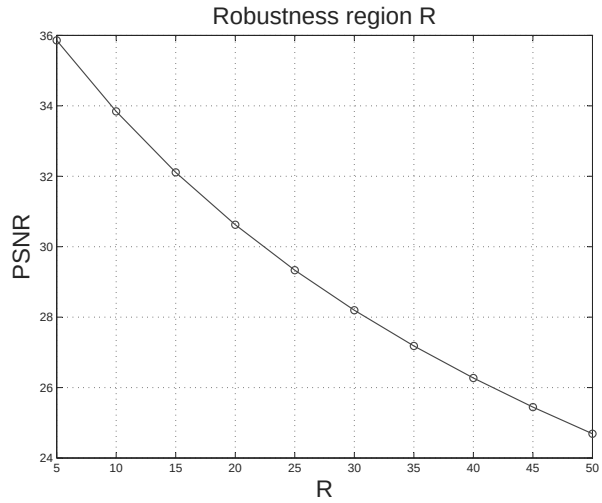

FIGURE 8. Robustness region R variation of the proposed type-3 informed embedding

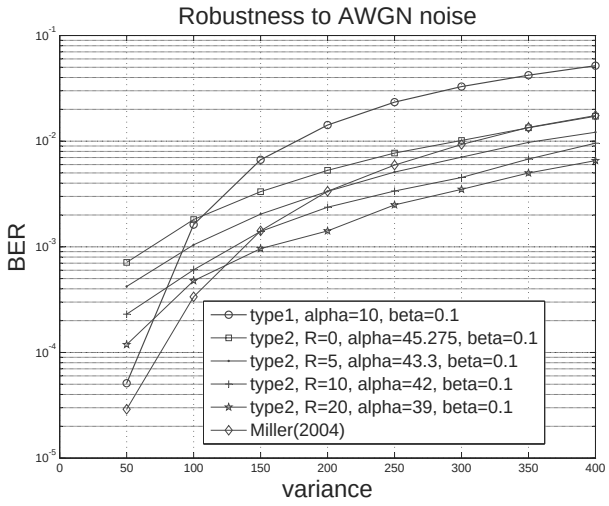

FIGURE 9. Watermark robustness to AWGN

FIGURE 10. Watermarked image to AWGN noise with variance = 300

parameter $R \geq 0$ is lower than that of Miller when $\sigma^2 > 200$ under AWGN noise. Figure 10 shows the watermarked image by AWGN attack.

In the experimental result, the proposed informed embedding with memory algorithm, type-2, exhibited superior BER performance when the controllable parameter R increased. The BER of informed embedding without memory algorithm was the highest of the variable environment; however, it consumed a lower time complexity than type-1 and [22] algorithms.

(b) Scaling

Image scaling is another common; distortion of significance is the scaling in amplitude. That is,

$$I'_w = s_f \times I_0$$

where I_0 is an original image and s_f is a scaling factor. This corresponds to brightness or contrast attack for images. Two tests were performed. The first test reduced the image intensities from 1 to 0.1. The second experiment increased the image intensities from 1 to 2.

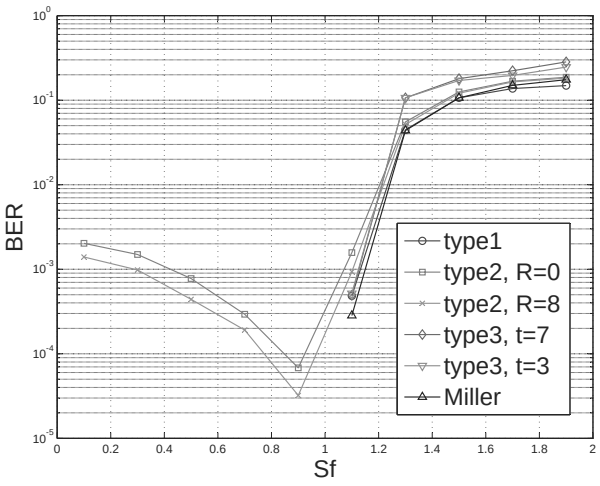

FIGURE 11. Robustness to scaling

Figure 11 shows the results for type-1, type-2, type-3 and [22], respectively. The figure demonstrates the trend in which BER variants as scaling factor s_f increase from 0 to 2. In our experiments, type-1 and [22] achieved appropriate performance when scaling factor from 1 to 2. Figure 12 shows the watermarked image by scaling.

(c) Low pass filter

Low pass filter, such as run average filters or Gaussian filters, are often applied in image processing. We chose the Gaussian filter for our simulation. As shown in Figure 13, the bit error rate performance of these four algorithms can be optimized under the low pass filter attack when setting $t = 3$ and 7 in type-3. The [22] algorithm provides superior bit error rate performance; however, it requires a higher complexity than type-1 and type-2. The algorithm of type-3 requires the lowest complexity and provides superior performance to [22]. Figure 14 shows the watermarked image through the Gaussian filter.

(d) JPEG

Lossy compression is common; therefore, watermark robustness is desirable. The effects of JPEG compression were tested. We observed BERs at various JPEG compression levels. This specified the quantization values of DCT coefficients by

FIGURE 12. Watermarked image with $s_f = 1.5$

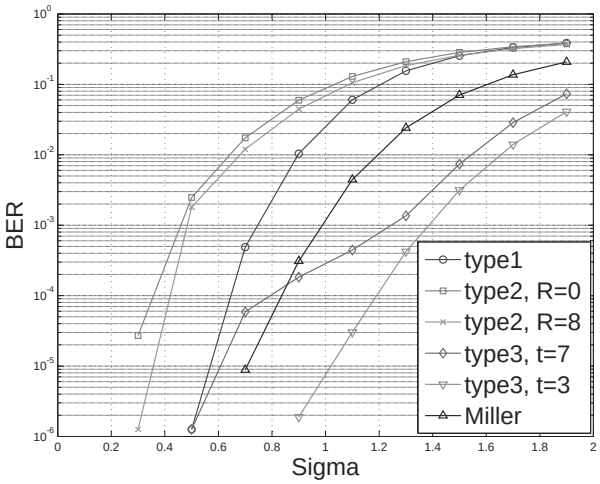

FIGURE 13. Robustness to low pass filtering

FIGURE 14. Watermarked image to low pass filtering with Gaussian filter $\sigma_f = 1.5$

multiplying a quantization matrix

$$\mu * \begin{pmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{pmatrix}$$

by a global quantization level μ. The quantization level is related to a user-specified quality factor, QF, in the range of 0-100, defined as

$$\mu = \begin{cases} \frac{50}{QF}, & \text{if } QF < 50 \\ 2 - 0.02 * QF, & \text{if } QF >= 50 \end{cases}$$

We compared BER at various JPEG compression levels, with quality factor QF from 20 to 80. Figure 15 shows the BER difference between the proposed algorithm and [22]. By using the informed embedding with memory algorithm, BER decreased rapidly as the QF increased, when $QF < 25$. The results demonstrated excellent watermark robustness of the informed embedding with memory in the case of low JPEG compression quality. Figure 16 shows the watermarked image by JPEG compression when $QF = 20$.

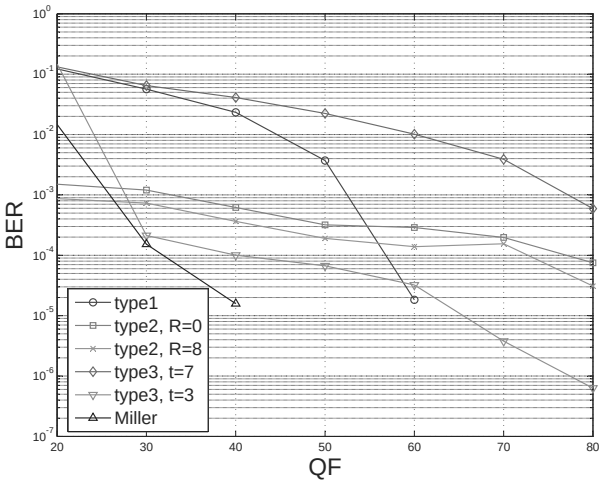

FIGURE 15. Robustness versus JPEG compression with variant QF

FIGURE 16. Watermarked image with $QF = 20$

3. Computational complexity

The algorithm complexity in [22] and that proposed in this paper is compared in this section. When the proposed algorithm and [22] were performed, each arc of each section of the trellis was executed as a unit when estimating the complexity. That is, the number of times that each arc is calculated after all the embedded information is completed. The Add-Compare-Select (ACS) operation in each section in the memory or accumulated Viterbi algorithm leads to a more complex embedding algorithm than that of a memoryless structure in regard to operation complexity. Although the ACS

operation in memory version increased, the average times of operation decreased in the trellis. In addition to the number of times that each arc of the trellis was used, the total number of times that each arc was used for various algorithms was also tabulated and compared. Thus, three complexity parameters in trellis structure are defined as follows.

The decoding process in Miller's algorithm and ours are both performed on the basis of a Viterbi algorithm. The operational complexities required in performing a Viterbi algorithm are detailed compared under the same conditions. An Euclidean distance is treated in experiments as a metric to evaluate a likelihood path in a Viterbi algorithm, according to which the Euclidean which must be evaluated between a modified watermarked image and an object message for each arc in a Viterbi trellis. Assuming that there are C_a number of arcs in each section in the trellis, then it is required to calculate the same number of Euclidean distances. With C_e symbolizing the complexity in evaluating each Euclidean distance, a total complexity of $C_a \times C_e$ is required for each section. In addition, the survive path is up to Add-Compare-Select (ACS) operations in each section, and it is for sure that different informed embedding algorithm brings about different number of ACS operations. Consequently, for the same length of watermarked image, the computational complexity is directly related to the average number of ACS operations in each section. In simple terms, a larger number of ACS operations gives rise to a higher complexity and a longer period of time to get the operations done. Taking a trellis with 32 states, each with 2 arcs, as an instance, it is known that the metric accumulated in the previous section pertains to the trellis states and the number of arcs. Since each current state is connected to two arcs, thus two adders are required to perform additions, that is, it necessitates C_a number of adders in each section. Inasmuch as there are two arcs connected to each next state, a comparer is thus required for comparison. In brief, there are 32 next states and 64 arcs in each section, that is, 32 comparers and 64 adders. Hence, the ACS complexity C_s for each section is given as

$$C_s = C_a \times C_e + C_a \times \texttt{adders} + (C_a/2) \times \texttt{comparers}$$

As presented in Section 3.1, the type-1 algorithm is a section-based informed embedding algorithm, i.e., a memoryless informed embedding approach performed independently in each section. In other words, it does not require any adder or comparer to perform any accumulation operations. Yet, there are C_a number of comparers required in search for an object message codeword out of all the arcs. The complexity required is expressed as

$$C_a \times C_e + C_a \times \texttt{comparers}$$

As for the type-2 algorithm, as presented in Section 3.2, there exists a trade off between the fidelity and the robustness specified a robust parameter R. Just as regular Viterbi algorithms, it requires an accumulated metric, except that it is performed in an iterative manner until the target value R_t of the robustness is reached with a specified complexity C_s. Due to iterations, the total operation load varies across sections. Accordingly, the average value of C_s is adopted as the complexity for each section. With C_{avg} representing the average number of operations in each section, the total load to perform a complete algorithm operation is given as $C_t = C_s \times C_{avg} \times L$. In contrast, the type-3 informed embedding algorithm in Section 3.3 is not performed iteratively, such that merely one time of operations is needed. Besides, the modification is made merely in the host image, i.e., an accumulated

metric is not required. As a consequence, there is no need to perform C_s number of operations, and the complexity is evaluated as $C_a \times C_e$ with $C_{avg} = 1$.

For operation complexity, the type-1 algorithm in 3.1 must find the minimal distance $d(\alpha c_k, x_k)$, regardless of whether the arc operation is closer to selected codeword αw_k in trellis section k. The computation required 32 comparer-operations. The type-2 algorithm was accumulated by an ACS operation. In a trellis section, the current state and next state is 16. The computation required 16 adder-operations and comparer-operations. The type-3 algorithm was not accumulated, and was executed section-by-section. Thus, we computed only by using "out of phase" on the arc. Finally, we compared with [22] and tabulated as Table 4.

TABLE 4. Numbers of operation for type-1, type-2, type-3 and [22] algorithms

Algorithm	C_s	C_t
type-1	$64 \times C_e$ + 32 comparers	$C_s \times 1.4872 \times 1585$
type-2 (R = 0)	$64 \times C_e$ + 64 adders + 32 comparers	$C_s \times 0.5915 \times 1585$
type-2 (R = 1)	$64 \times C_e$ + 64 adders + 32 comparers	$C_s \times 0.6697 \times 1585$
type-3	$64 \times C_e$	$C_s \times 1 \times 1585$
[22]	$64 \times C_e$ + 64 adders + 32 comparers	$C_s \times 69.841 \times 1585$

As shown in Table 4, the computational cost of four algorithms depends on performing each arc and ACS operation. For total operation complexity C_t, these algorithms were ordered as follows:

$$\text{Miller's} > \text{type-1} > \text{type-2(R} = 1) > \text{type-2(R} = 0) > \text{type-3}$$

Although the proposed algorithms have low complexity, they are robust. In the simulation results, type-3 exhibited high robustness in a number of attacks, such as low pass filter attack channel and compression attack channel, and allowed low operation complexity.

5. **Conclusions.** This paper proposes the modified informed embedding scheme for data hiding. We used this trellis code with the modified trellis structure and simplex code to embed 1585 bits in 512×512 pixels image. These three algorithms used the codewords of a linear block code to label the arcs in the trellis, and subsequently adjusted the fidelity and robustness of the watermarked images by a number of controllable parameters. The proposed algorithms can resist the channel attack and maintain the fixed amount fidelity efficiently. Moreover, the proposed algorithms require less operation, compared with that of Miller. The proposed methods reduced the operation complexity for a watermarked image, and provide superior BER performance, in comparison with Miller's informed embedding.

Acknowledgment. The research was supported by the National Science Council, Taipei, Taiwan under Contract NSC 99-2221-E-005-081-MY2, NSC 96-2221-E-167-021 and NSC-100-2221-E-167-024.

REFERENCES

[1] M.-F. Horng, B.-C. Huang, M.-H. Lee and Y.-H. Kuo, A DC-based approach to robust watermarking with Hamming-code, *International Journal of Innovative Computing, Information and Control*, vol.5, no.7, pp.1911-1917, 2009.

[2] L.-D. Li, B.-L. Guo and J.-S. Pan, Robust image watermarking using feature based local invariant regions, *International Journal of Innovative Computing, Information and Control*, vol.4, no.8, pp.1977-1986, 2008.

[3] J. A. Bloom, I. J. Cox, T. Kalker, J.-P. M. G. Linnartz, M. L. Miller and C. B. S. Traw, Copy protection for digital video, *Proc. of IEEE Special Issue on Identification and Protection of Multimedia Information*, vol.87, no.7, pp.1267-1276, 1999.
[4] D. Boneh and J. Shaw, Collusion-secure fingerprinting for digital data, *IEEE Trans. Inf. Theory*, vol.44, no.5, pp.1897-1905, 1998.
[5] D. Kundeur and D. Hatzinakos, Digital watermarking for telltale tamper proofing and authentication, *Proc. of IEEE*, vol.87, no.7, pp.1167-1180, 1999.
[6] A. lmaz and A. A. Alatan, Error detection and concealment for video transmission using information hiding, *Signal Processing: Image Communication*, vol.23, no.4, pp.298-312, 2008.
[7] S. D. Lin and Y.-H. Huang, An integrated watermarking technique with tamper detection and recovery, *International Journal of Innovative Computing, Information and Control*, vol.5, no.11(B), pp.4309-4316, 2009.
[8] F. Bartolini, A. Tefas, M. Barni and I. Pitas, Image authentication techniques for surveillance applications, *Proc. of IEEE*, vol.89, no.10, pp.1403-1418, 2001.
[9] E. Martinian and G. W. Wornell, Authentication with distortion constraints, *Proc. of IEEE Int. Conf. Image Processing,* pp.17-20, 2002.
[10] I. J. Cox, M. L. Miller and J. A. Bloom, *Digital Watermarking*, Morgan Kaufmann, New York, 2001.
[11] P. Moulin and R. Koetter, Data-hiding codes, *Proc. of IEEE*, vol.93, no.12, pp.2083-2126, 2005.
[12] I. J. Cox, M. L. Miller and A. McKellips, Watermarking as communications with side information, *Proc. of IEEE*, vol.87, pp.1127-1141, 1999.
[13] M. H. M. Costa, Writing on dirty paper, *IEEE Trans. Inf. Theory*, vol.29, pp.439-441, 1993.
[14] Y. Sun, Y. Yang, A. D. Liveris, V. Stankovic and Z. Xiong, Near-capacity dirty-paper code design: A source-channel coding approach, *IEEE Trans. Inf. Theory*, vol.55, no.7, pp.3013-3031, 2009.
[15] R. Barron, B. Chen and G. W. Wornell, The duality between information embedding and source coding with side information and some applications, *IEEE Trans. Inf. Theory*, vol.49, no.5, pp.1159-1180, 2003.
[16] S. S. Pradhan, J. Chou and K. Ramchandran, Duality between source coding and channel coding and its extension to the side information case, *IEEE Trans. Inf. Theory*, vol.49, no.5, pp.1181-1203, 2003.
[17] C.-S. Lu, Towards robust image watermarking: Combining content-dependent key, moment normalization, and side-informed embedding, *Signal Processing: Image Communication*, vol.20, no.2, pp.129-150, 2005.
[18] R. Zamir, S. Shamai and U. Erez, Nested linear/lattice codes for structured multierminal binning, *IEEE Trans. Inf. Theory*, vol.48, no.6, pp.1250-1276, 2002.
[19] M. J. Wainwright, Sparse graph codes for side information and binning, *IEEE Signal Processing Mag.*, vol.24, no.5, pp.47-57, 2007.
[20] C. Desset, B. Macq and L. Vandendorpe, Block error-correcting codes for systems with a very high BER: Theoretical analysis and application to the protection of watermarks, *Signal Processing: Image Communication*, vol.17, no.5, pp.409-421, 2002.
[21] M. L. Miller, I. J. Cox and J. A. Bloom, Informed embedding: Exploiting image and detector information during watermark insertion, *IEEE Int. Conf. on Image Processing*, 2000.
[22] M. L. Miller, G. J. Doerr and I. J. Cox, Applying informed coding and embedding to design a robust high-capacity watermark, *IEEE Trans. Image Process.*, vol.13, no.6, pp.792-807, 2004.
[23] L. Lin, G. Doerr, I. Cox and M. Miller, An efficient algorithm for informed embedding of dirty-paper trellis codes for watermarking, *Proc. of IEEE Int. Conf. Image Processing*, Italy, 2005.
[24] G. D. Forney, The Viterbi algorithm, *Proc. IEEE*, vol.61, pp.268-278, 1973.