The Scalar Scheme for Reversible Information-Embedding in Gray-Scale Signals: Capacity Evaluation and Code Constructions

Sian-Jheng Lin and Wei-Ho Chung, *Member, IEEE*

***Abstract*—Reversible information-embedding (RIE) is a technique transforming host signals and the message into the stego-signals, and the stego-signals can be losslessly reversed to the host signals and the message. We consider the conditions: 1) the host signals are composed of gray-scale independent and identically distributed (i.i.d.) samples; 2) the mean squared error is adopted as the measure of distortion; and 3) the procedure is a scalar approach, i.e., the encoder only reads a host signal and then outputs the corresponding stego-signal in each iteration. In this paper, we propose an iterative algorithm to calculate the signal transition probabilities approximating the optimal rate-distortion bound. Then we propose an explicit implementation to embed a message in an i.i.d. host sequence. The experiments show that the proposed method closely approaches the expected rate-distortions in i.i.d. gray-scale signals. By the image prediction model, the proposed method can be applied to gray-scale images.**

***Index Terms*—Arithmetic coding, gray-scale signals, rate-distortion function, reversible information embedding, steganography.**

Nomenclature

For the host signal s, stego-signal y and embedded message w, the lowercase letters denote the realizations, and capital letters denote random variables. The superscript S^N, Y^N denotes the sequence with length N, and the subscript of S_i, Y_i describes the ith signal in the sequence. The head notations $\hat{\bullet}$ denote the decoded versions of the signals.

N	Length of host sequence.
B	Size of possible gray-scale signals.
M	Size of possible messages.
$\mathbb{Z}_B$	Set of integer $\mathbb{Z}_B = \{0, 1, \ldots, B-1\}$.
$H(X)$	Entropy $H(X) = -\sum_{i=1}^{n} x_i p_X(x_i)$ of a discrete random variable $X \in \{x_1, \ldots, x_n\}$.
$Y^N = f_N(S^N, W)$	Encoding function.
$(\hat{S}^N, \hat{W}) = \phi_N(Y^N)$	Decoding function.
D_{av}	Distortion.
R	Embedding rate.
$p_S, p_Y, p_{S,Y}$	Probability mass functions (pmf) of host signal, stego-signal and transaction events.
$P_S, P_Y, P_{S,Y}$	Cumulative pmf of host signal, stego-signal and transaction events.
$p_Y^*, p_{S,Y}^*$	Optimal rate-distortion configurations for a given p_S.

Manuscript received October 24, 2011; revised February 20, 2012; accepted April 10, 2012. Date of publication May 03, 2012; date of current version July 09, 2012. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Adnan M. Alattar.

The authors are with the Research Center for Information Technology Innovation, Academia Sinica, Taipei 115 Taiwan (e-mail: sjlin@citi.sinica.edu.tw, whc@citi.sinica.edu.tw).

Digital Object Identifier 10.1109/TIFS.2012.2197614

I. Introduction

INFORMATION embedding is the art of writing secret messages in digital host files, such as images, videos and audios, subject to a distortion constraint. The major utilities of information embedding include the digital watermarking (copyright protection) [1], authentication [2], and steganography [3]. The fundamental theorems and theoretical bounds of information embedding, in terms of the distortion level, embedding rate, robustness and detectability, have been formulated in recent literature [4]–[7]. Reversible information embedding (RIE) has the property where the receiver can completely reconstruct the host file from the received stego-signals [8], [9]. This property protects the sensitive host data, such as medical photographs or military maps, and avoids the distortion derived from the embedding process. For independent and identically distributed (i.i.d.) host signals, Kalker and Willems [10] proved the fundamental property of the embedding rate subject to an admissible distortion. Certain generalized issues, such as partially RIE scheme [11] and robust embedding [12], [13], have also been investigated. Zhang *et al.* [22] provide a capacity-approaching code for binary i.i.d. signals. Haroutunian *et al.* [14] analyzed the error-exponent of the RIE system. Voloshynovskiy *et al.* [15] provides the theoretical results of the partial reversibility under the Gelfand-Pinsker formulation and Gaussian Costa setup. The scenario of decoding with the partial information of host signals is studied by Steinberg [16]. The models of multiple access and broadcast channels are considered by Kotagiri *et al.* [17].

In many practical RIE methods for gray-scale images [18]–[21], the curve of embedding rates versus PSNR distortion measures is frequently adopted as the metric of merits. Kalker and Willems [10] investigate the criteria of realizable rate-distortion pairs for i.i.d. signals, and a coding scheme for binary i.i.d. case is proposed. An improved version for the binary i.i.d. case is proposed by Zhang *et al.* [22] to approach

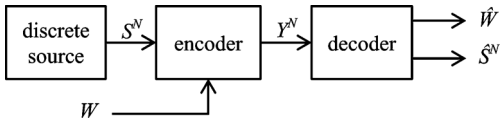


Fig. 1. Reversible noise-free information-embedding system.

the theoretical rate-distortion bound. However, the explicit capacity-approaching codes for gray-scale i.i.d. case are not being further investigated. The observations described here motivate the problem setup of this paper. We consider the host signals composed of i.i.d. gray-scale samples, and the metric of distortion is the mean squared error (MSE). Based on the results of [10], the considered embedding method is a scalar scheme which sequentially processes the host signals.

The rest of this paper is organized as follows. Section II gives the basic notations and definitions of the system. Section III defines the scalar approach of the embedding scheme. In Sections IV and V, we derive an equation for the optimal rate-distortion bound and then propose an iterative algorithm to evaluate the rate-distortion curve. In Section VI, we propose the coding method to achieve the rate-distortion bound. Section VII shows the experiments of the proposed algorithms. Section VIII discusses several topics w.r.t. the proposed scheme. Section IX concludes this work.

II. Noise-Free Embedding System

This section introduces the reversible noise-free embedding by Kalker and Willems [10], [13] shown in Fig. 1. The host sequence $S^N = (S_1, S_2, \ldots, S_N)$ is composed of samples i.i.d. drawn from the probability mass function (pmf) $p_S = \{p_S(s), s \in \mathbb{Z}_B\}$, where the set $\mathbb{Z}_B = \{0, 1, \ldots, B-1\}$ is a finite set of integers modulo the integer B. The random variable W denotes the message uniformly distributed in $\mathbb{Z}_M = \{0, 1, \ldots, M-1\}$. The encoder produces a stego-sequence $Y^N = (Y_1, Y_2, \ldots, Y_N)$ through an injective function $Y^N = f_N(S^N, W)$ to write the message W in the host sequence S^N, so that the decoder can losslessly reconstruct the message $\hat{W}$ and the host sequence $\hat{S}^N$ through the inverse function $(\hat{S}^N, \hat{W}) = \phi_N(Y^N)$, where $\hat{S}^N = S^N$ and $\hat{W} = W$. The stego-sequence is required to be as close as possible to the host sequence according to a distortion metric expressed as

$$D_{av} = M^{-1} \sum_{s^N, W} \Pr\{S^N = s^N\} d(s^N, f_N(s^N, W)) \quad (1)$$

where the distortion measure is defined as

$$d(s^N, y^N) = N^{-1} \sum_{i=1}^{N} D(s_i, y_i). \quad (2)$$

The condition of lossless reversible $H(W, S^N|Y^N) = 0 \Rightarrow H(Y^N) \geq H(S^N) + H(W)$ provides the upper bound of embedding rate R, in bits per signal, expressed as

$$R = N^{-1} \log_2 M \leq N^{-1}(H(Y^N) - H(S^N)). \quad (3)$$

Let (ρ, Δ) denote an realizable rate-distortion pair for the reversible noise-free embedding system. We define the random variables S and Y with $p_{S,Y}(s, y) = N^{-1} \sum_{i=1}^{N} P_{S_i, Y_i}(s, y)$. For $N \to \infty$, [10, Th. 1] proves a useful property that

$$\rho \leq \max H(Y) - H(S), \text{ and } \Delta \geq \sum_{s,y} p_{S,Y}(s, y) D(s, y) \quad (4)$$

where the maximum is over all possible configurations of the joint pmf $p_{S,Y}$.

III. Scalar Embedding Scheme With Square Error Distortion

In the proposed scheme, the distortion metric in (2) is defined as the square error distortion

$$D(s, y) = (s - y)^2. \quad (5)$$

It is noted that we discuss other distortion metrics in Section VII-B. We consider the scheme which decomposes the embedding function $f_N(S^N, W)$ into N sub-functions $(Y_i, W_{i+1}) = f(S_i, W_i)$, for $i = 1, 2, \ldots, N$, where the random variable W_i represents the information required to be embedded in the unprocessed host signals $S_i, S_{i+1}, \ldots, S_N$. The initialization is given by $W_1 = W$, and the termination condition $H(W_{N+1}) = 0$ occurs when the information of the message is completely embedded in the stego-sequence. The condition of lossless reversibility gives the property $H(Y_i, W_{i+1}|S_i, W_i) = H(S_i, W_i|Y_i, W_{i+1}) = 0$. In other words, the encoder embeds a portion of the message into each host signal to produce the corresponding stego-signal at each step. The decoder transforms the Y_i into S_i and W_i in reverse order through the decoding function $(S_i, W_i) = \phi(Y_i, W_{i+1})$, for $i = N$ to 1.

The previous conditions provide the property that the stego-sequence Y^N achieving the optimal rate-distortion performance are composed of gray-scale i.i.d. samples. Each pair (Y_i, W_{i+1}) is required to be mutually independent $H(Y_i|W_{i+1}) = H(Y_i)$ to prevent the redundant information in W_{i+1}. For any two stego-signals Y_i and Y_j, $i < j$, we have $H(Y_i|Y_j) \leq H(Y_i)$, and

$$\begin{aligned} H(Y_i|Y_j) &\geq H(Y_i|W_{j+1}, Y_j, Y_{j-1}, \ldots, Y_{i+1}) \\ &\geq H(Y_i|S_j, S_{j-1}, \ldots, S_{i+1}, W_{i+1}) \\ &\quad (\because \forall\, k = i+1, i+2, \ldots, j \;\; H(Y_k, W_{k+1}|S_k, W_k) = 0) \\ &= H(Y_i|W_{i+1}) \;\; (\because \forall\, j > i H(Y_i|S_j) = H(Y_i)) \\ &= H(Y_i) \end{aligned}$$

where Y_i and Y_j are shown to be independent and identically distributed. In the following, for simplicity of presentation, we omit the subscripts of the stego-signal and host signal.

The pmf of the stego-signal Y is defined as $p_Y = \{p_Y(y), y \in \mathbb{Z}_B\}$, and the probabilities of transition events $Y = y \to S = s$ is defined as the joint pmf $p_{S,Y} = \{p_{S,Y}(s, y), s \in \mathbb{Z}_B, y \in \mathbb{Z}_B\}$. The embedding

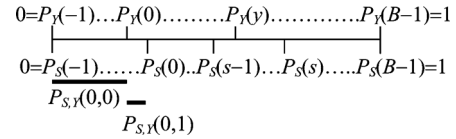


Fig. 2. Diagram showing relations of p_S, p_Y and $p_{S,Y}$ with intervals.

rate and the average distortion are formulated as

$$\begin{aligned} R &\le N^{-1}(H(Y^N) - H(S^N)) \\ &= -\sum_{y=0}^{B-1} p_Y(y)\log_2 p_Y(y) + \sum_{s=0}^{B-1} p_S(s)\log_2 p_S(s) \quad (6) \end{aligned}$$

$$D_{av} = \sum_{s=0}^{B-1}\sum_{y=0}^{B-1} p_{S,Y}(s,y) \times D(s,y). \quad (7)$$

It is noted that p_S is given by the host sequence, and the p_Y and $p_{S,Y}$ are determined by the applied encoder. For a given pmf p_S, the rate-distortion function gives the bound of embedding rate R upon a specific average distortion D_{av}.

The optimality of the scalar scheme is investigated as follows. The theoretical conditions for the reversible noise-free embedding system are given in (4). We can observe that (4) is identical to the scalar scheme (6)–(7). Thus, for each optimal rate-distortion pair (ρ^*, Δ^*) of the reversible noise-free embedding system, there exists a configuration of the scalar scheme to achieve the rate-distortion performance.

IV. Rate-Distortion Optimization

The embedding rate versus the average distortion can be formulated as an optimization problem:

$$\begin{aligned} &\text{Maximize}: R(p_{S,Y}) \\ &= -\sum_{y=0}^{B-1}\left(\sum_{s=0}^{B-1} p_{S,Y}(s,y)\right)\log_2\left(\sum_{s=0}^{B-1} p_{S,Y}(s,y)\right) - H(S) \end{aligned} \quad (8)$$

$$\text{s.t.}: D(p_{S,Y}) = \sum_{s=0}^{B-1}\sum_{y=0}^{B-1} p_{S,Y}(s,y) \times D(s,y) = D_{av} \quad (9)$$

$$h_s(p_{S,Y}) = \sum_{y=0}^{B-1} p_{S,Y}(s,y) = p_S(s), \quad \forall\, s \quad (10)$$

$$-p_{S,Y}(s,y) \le 0, \quad \forall\, s,y \quad (11)$$

where $H(S) = -\sum_{s=0}^{B-1} p_S(s)\log_2 p_S(s)$ is a constant. To simplify the notations in the following, we define

$$g(s,y) = -\log_2 p_Y^*(y) - \lambda D(s,y). \quad (12)$$

Lemma 1: Assume $p_{S,Y}^*$ maximizes (8) subject to the constraints (9)–(11). For any $p_{S,Y}^*(s_1,y_1) > 0$ and $p_{S,Y}^*(s_2,y_2) > 0$ where $s_1 \le s_2$, we have

$$g(s_1,y_1) \ge g(s_1,y_2) \text{ and } g(s_2,y_2) \ge g(s_2,y_1). \quad (13)$$

Proof: We attempt to solve the $p_{S,Y}^*$ with the KKT conditions. Since $p_{S,Y}^*$ is optimal, there exists the values λ, $\{\lambda_s | \forall\ s\}$ and $\{\mu_{s,y} | \forall\ \ s,y\}$ such that

$$\begin{aligned} &\nabla R(p_{S,Y}^*) - \lambda \nabla D(p_{S,Y}^*) - \sum_{s=0}^{B-1}\lambda_s \nabla h_s(p_{S,Y}^*) \\ &\quad - \sum_{s=0}^{B-1}\sum_{y=0}^{B-1}\mu_{s,y}\nabla p_{S,Y}^*(s,y) = 0 \\ &\Rightarrow -\log_2\left(\sum_{s=0}^{B-1} p_{S,Y}^*(s,y)\right) - \log_2 e - \lambda D(s,y) \\ &\quad - \lambda_s + \mu_{s,y} = 0 \\ &\Rightarrow g(s,y) = \log_2 e + \lambda_s - \mu_{s,y}, \quad \forall\, s,y; \end{aligned} \quad (14)$$

$$\mu_{s,y} p_{S,Y}^*(s,y) = 0 \quad \forall\, s,y \quad (15)$$

$$\mu_{s,y} \ge 0 \quad \forall\, s,y. \quad (16)$$

Because of $p_{S,Y}^*(s_1,y_1) > 0$ and $p_{S,Y}^*(s_2,y_2) > 0$, we have $\mu_{s_1,y_1} = \mu_{s_2,y_2} = 0$ by (15), and

$$g(s_j,y_j) = \lambda_{s_j} + \log_2 e, \quad \forall\, j = 1,2. \quad (17)$$

Equation (16) gives $\mu_{s_1,y_2} \ge 0$ and $\mu_{s_2,y_1} \ge 0$. By (14) and (17), we prove the (13)

$$\begin{aligned} &g(s_1,y_2) \le \log_2 e + \lambda_{s_1} \Rightarrow g(s_1,y_2) \le g(s_1,y_1). \\ &g(s_2,y_1) \le \log_2 e + \lambda_{s_2} \Rightarrow g(s_2,y_1) \le g(s_2,y_2). \end{aligned}$$

■

Lemma 1 supports the crossing-edges property proven by Willems *et al.* [26]. The proof of crossing-edges property is in [26, Cor. 1]. We give another proof as follows.

Corollary 1 (Crossing-Edges Property): Given an optimal $p_{S,Y}^*$, for any two distinct possible transaction events $p_{S,Y}^*(s_1,y_1) > 0$ and $p_{S,Y}^*(s_2,y_2) > 0$ with $s_1 \le s_2$, then the $y_1 \le y_2$ holds.

Proof: By Lemma 1, we have

$$\begin{aligned} &g(s_1,y_1) + g(s_2,y_2) \ge g(s_1,y_2) + g(s_2,y_1) \\ &\quad \Rightarrow -\lambda D(s_1,y_1) - \lambda D(s_2,y_2) \ge -\lambda D(s_1,y_2) \\ &\qquad - \lambda D(s_2,y_1) \\ &\quad \Rightarrow (s_1-y_2)^2 + (s_2-y_1)^2 - (s_1-y_1)^2 \\ &\qquad - (s_2-y_2)^2 \ge 0 \\ &\quad \Rightarrow 2(s_1-s_2)(y_1-y_2) \ge 0. \end{aligned} \quad (18)$$

Thus, the inequality holds. ■

Corollary 1 provides the graphical representation of the pmfs p_S, p_Y, and $p_{S,Y}$ with intervals shown in Fig. 2. The interval $[0,1)$ is divided into several subintervals according to the elements in cumulative pmfs P_S and P_Y, where $P_S = \{P_S(s) = \sum_{i=0}^{s} p_S(i), s \in \mathbb{Z}_B\}$ and $P_Y = \{P_Y(y) = \sum_{i=0}^{y} p_Y(i), s \in \mathbb{Z}_B\}$. It is noted that $P_S(-1) = P_Y(-1) = 0$, and $P_S(B-1) = P_Y(B-1) = 1$. The widths of intervals $[P_S(s-1), P_S(s))$ and $[P_Y(y-1), P_Y(y))$, respectively, represent the probabilities of events $S = s$ and $Y = y$. For each interval for P_S overlapping the interval for P_Y, the width of the overlapped range determines the probability of the transition event $S = s \to Y = y$,

expressed as $p_{S,Y}(s,y)=\min\{P_S(s),P_Y(y)\}-\max\{P_S(s-1),P_Y(y-1)\}$. Fig. 2 visually depicts the values of pmf $p_{S,Y}$ under the given p_S and p_Y, so we only need to compute the values of p_Y in the proposed algorithm.

Lemma 2: Assume (P_Y^*,α) achieves the optimal rate-distortion for a given P_S, where the P_Y^* is the cumulative pmf of stego-signal and $\alpha\in[1,\infty)$ is a constant. For $y=1$ to $B-1$, each $P_Y^*(y)$ satisfies one of the two conditions:

$$P_Y^*(y)=\frac{P_Y^*(y+1)-P_Y^*(y-1)}{1+\alpha^{D(s,y)-D(s,y+1)}}+P_Y^*(y-1),\quad \forall\, P_Y^*(y)\in(P_S(s-1),P_S(s)) \tag{19}$$

$$\alpha^{D(s,y)-D(s,y+1)}\le\frac{P_Y^*(y+1)-P_S(s)}{P_S(s)-P_Y^*(y-1)}\le\alpha^{D(s+1,y)-D(s+1,y+1)},\quad \forall\, P_Y^*(y)=P_S(s). \tag{20}$$

Proof: For the case $P_Y^*(y)\in(P_S(s-1),P_S(s))$, as shown in Fig. 2, the two transactions $p_{S,Y}^*(s,y)>0$ and $p_{S,Y}^*(s,y+1)>0$ are possible. By Lemma 1

$$\begin{aligned}
&g(s,y)\ge g(s,y+1)\text{ and }g(s,y)\le g(s,y+1)\\
&\quad\Rightarrow g(s,y)=g(s,y+1)\\
&\quad\Rightarrow -\log_2 p_Y^*(y)-\lambda D(s,y)\\
&\quad\quad=-\log_2 p_Y^*(y+1)-\lambda D(s,y+1)\\
&\quad\Rightarrow -\log_2(P_Y^*(y)-P_Y^*(y-1))-\lambda D(s,y)\\
&\quad\quad=-\log_2(P_Y^*(y+1)-P_Y^*(y))-\lambda D(s,y+1)\\
&\quad\Rightarrow P_Y^*(y)\\
&\quad\quad=\frac{P_Y^*(y+1)-P_Y^*(y-1)}{1+2^{\lambda(D(s,y)-D(s,y+1))}}+P_Y^*(y-1).
\end{aligned}\tag{21}$$

The coefficient $\alpha=2^\lambda$ is substituted in (21) to obtain (19). By crossing-edges property [26], for each $s_1<s_2$, we have

$$\begin{aligned}
D(s_1,y)+D(s_2,y+1)&\le D(s_1,y+1)+D(s_2,y)\\
&\Rightarrow D(s_1,y)-D(s_1,y+1)\\
&\le D(s_2,y)-D(s_2,y+1).
\end{aligned}\tag{22}$$

Therefore, supposing $P_Y^*(y)\in(P_S(\tilde{s}-1),P_S(\tilde{s}))$, we can iteratively substitute (21) with sequentially increasing the value $\tilde{s}$ to find the $P_Y^*(y)$ which is between $P_S(\tilde{s}-1)$ and $P_S(\tilde{s})$.

The case $P_Y^*(y)=P_S(s)$ is discussed as follows. Under the condition $P_Y^*(y)\in(P_S(\tilde{s}-1),P_S(\tilde{s}))$, (21) returns the $P_Y^*(y)$ larger than $P_S(\tilde{s})$. Under the condition $P_Y^*(y)\in(P_S(\tilde{s}),P_S(\tilde{s}+1))$, (21) returns the $P_Y^*(y)$ smaller than $P_S(\tilde{s})$. Summarizing, the $P_Y^*(y)$ is "squeezed" at $P_Y^*(y)=P_S(s)$. Based on the circumstances described previously, (20) can be calculated by

$$\begin{aligned}
\frac{P_Y^*(y+1)-P_Y^*(y-1)}{1+2^{\lambda(D(s-1,y)-D(s-1,y+1))}}+P_Y^*(y-1)&\ge P_S(s)\\
&\ge\frac{P_Y^*(y+1)-P_Y^*(y-1)}{1+2^{\lambda(D(s,y)-D(s,y+1))}}+P_Y^*(y-1).
\end{aligned}\tag{23}$$

Then lemma (19)–(20) is therefore proven. ∎

User can control the parameter α in (19)–(20) for various rate-distortion pairs. The case $\alpha=1$ admits the uniform distribution $p_Y^*=\{p_Y^*(y)=y/B,y\in\mathbb{Z}_B\}$ performing the maximal capacity, and another case $\alpha\to\infty$ admits $p_Y^*=\{p_Y^*(y)=p_S^*(y),y\in\mathbb{Z}_B\}$ generating a unvaried stego-sequence without embedding information. When α decreases from infinity to 1, the embedding rate increases from zero to maximum, but the stego-sequence changes from an unaltered version to a uniform random sequence.

1) Example 1: For a binary i.i.d. sequence, the optimal p_Y^* satisfies

$$p_Y^*(1)=\begin{cases}(\alpha^{-1}+1)^{-1}, & \text{if } p_S(1)>0.5 \text{ and } \alpha<(p_S(1)^{-1}-1)^{-1};\\ (\alpha+1)^{-1}, & \text{if } p_S(1)<0.5 \text{ and } \alpha<p_S(1)^{-1}-1;\\ p_S(1), & \text{otherwise;}\end{cases}$$

where the constant $\alpha\in[1,\infty)$ corresponds to various embedding rates and distortions. The rate-distortion pair is therefore $R=H(p_Y^*)-H(p_S)$, $D_{av}=|p_Y^*(1)-p_S(1)|$. ∎

2) Example 2: For a constant sequence $p_S(j)=1$, the optimal p_Y^* is $p_Y^*(y)=\alpha^{-(j-y)^2}\left(\sum_{y=0}^{B-1}\alpha^{-(j-y)^2}\right)^{-1}$, $\forall\ y\in\mathbb{Z}_B$. For example, if $j=1$ and $B=3$, we have $(p_Y(0),p_Y(1),p_Y(2))=((\alpha+2)^{-1},\alpha(\alpha+2)^{-1},(\alpha+2)^{-1})$. The embedding rate and the average distortion are $R=H(p_Y^*)$, and $D_{av}=2(\alpha+2)^{-1}$. ∎

V. Iterative Method to Calculate PMF of Stego-Signals

Except certain simple examples shown in Examples 1 and 2, it is difficult to obtain the closed form of the P_Y^* through Lemma 2. Thus, an iterative algorithm is designed to approach the unique solution of P_Y through (19)–(20). In Algorithm 1, each temporal value of $P_Y(y)$ is stored in the variable x_y, and the value x_y is iteratively updated to approach the desired solution.

Algorithm 1: The iterative algorithm of calculating the pmf of stego-signals.

Input: The cumulative pmf P_S, a real number $\alpha\ge1$, and the tolerance value ε.

Output: The cumulative pmf P_Y.

1) Given an initial set $x=\{x_y|x_y=(y+1)/B,y=-1\text{ to }B-1\}$. It is noted that the user can design an arbitrary initialization as long as $0=x_{-1}\le x_0\le\cdots\le x_{B-1}=1$. Declare the variable $var=0$.

2) For y from 0 to $B-2$, update each x_y through

$$x_y^{\text{new}}=\begin{cases}\frac{x_{y+1}-x_{y-1}}{1+\alpha^{D(s,y)-D(s,y+1)}}+x_{y-1}, & \text{if there exists } x_y\in(P_S(s),P_S(s+1));\\ P_S(s), & \text{if there exists } \alpha^{D(s,y)-D(s,y+1)}\le\frac{x_{y+1}-P_S(s)}{P_S(s)-x_{y-1}}\le\alpha^{D(s+1,y)-D(s+1,y+1)}\end{cases}\tag{24}$$

After finding the new value x_y^{new}, we record the maximal offset by

$$var=\max\{var,|x_y^{\text{new}}-x_y^{\text{old}}\}.\tag{25}$$

3) For y from $B-2$ to 0, update each x_y through **(24)** and record the maximal offset **(25)**.

4) If $var > \varepsilon$, set $var = 0$ and then go to step 2); otherwise, output $P_Y = \{P_Y(y) = x_y\}$.

By observing (24), it can be shown that for any values of x_{y-1} and x_{y+1}, there always exists a unique x_y^{new} in $[x_{y-1}, x_{y+1}]$. Algorithm 1 updates each x_y^{new} in alternating forward and backward manner, which achieves faster convergence than updating in one-way manner. We briefly explain the potential problem of calculating each x_y only in the forward direction. Suppose $x^* = \{x_y^*, y = -1, 0, \ldots, B-1\}$ being the fixed point to be achieved, and for a bad initialization $0 = x_{-1} = x_{-1}^* \le x_0 \le \cdots \le x_{B-2} \le x_0^* \le \cdots \le x_{B-1} = x_{B-1}^* = 1$, since each x_y is bounded in the two nearby elements $[x_{y-1}, x_{y+1})$, the values $x_0, x_1, \ldots, x_{B-2}$ are still smaller than X_0^* after an iteration. This phenomenon slows down the speed of convergence, but Algorithm 1 can properly deal with the case by iteratively reversing the order of updating each x_y. Lemma 4 proves the uniqueness of the iterative algorithm.

Lemma 4: After each iteration in Algorithm 1, the set $x = \{x_y | y = -1, 0, \ldots, B-1\}$ converges to an unique optimal solution.

Proof: We only prove the convergence of the forward manner (Step 2) by the Banach fixed point theorem [23], since the proof of the backward manner is similar by reversing the calculation order. Let a $B+1$-dimensional space U^{B+1} denote the domain of x, and for each $x \in U^{B+1}$, $0 = x_{-1} \le x_0 \le \cdots \le x_{B-1} = 1$. Thus, each feasible x can be assigned to an element of U^{B+1}. We define the distance function as the infinity norm

$$d(x, y) = \|x - y\|_\infty = \max_i \{|x_i - y_i|\}, \forall x, y \in U^{B+1}. \quad (26)$$

Thus, the (U^{B+1}, d) constructs a complete metric space. For $i = 1$ and 2, given any two elements $x_i^{\text{old}} \in U^{B+1}$, the updated elements $x_i^{\text{new}} = (x_{i,-1}^{\text{new}}, x_{i,0}^{\text{new}}, \ldots, x_{i,B-1}^{\text{new}})$ through updating (24) reformulated as

$$x_{i,y}^{\text{new}} = \begin{cases} \frac{x_{i,y+1}^{\text{old}} - x_{i,y-1}^{\text{new}}}{1+\alpha^{2(s-y)+1}} + x_{i,y-1}^{\text{new}}, & \text{if there exists } x_{i,y}^{\text{new}} \in (P_S(s), P_S(s+1)); \\ P_S(s), & \text{if there exists } \alpha^{2(s-y)-1} \le \log_\alpha \frac{x_{i,y+1}^{\text{old}} - P_S(s)}{P_S(s) - x_{i,y-1}^{\text{new}}} \le \alpha^{2(s-y)+1}. \end{cases} \quad (27)$$

The Banach fixed point theorem claims that if

$$\|x_1^{\text{new}} - x_2^{\text{new}}\|_\infty \le q \|x_1^{\text{old}} - x_2^{\text{old}}\|_\infty \quad (28)$$

for a constant $q < 1$ and $x_i^{\text{new}} \in U^{B+1}$, there exists a unique fixed point of the iteration algorithm. Before proving (28), we first prove the following inequality by mathematical induction:

$$|x_{1,y}^{\text{new}} - x_{2,y}^{\text{new}}| \le [1 - (1+\alpha^{2y-1})^{-y}] ||x_1^{\text{old}} - x_2^{\text{old}}||_\infty, \quad \text{for } y = -1 \text{ to } B-1. \quad (29)$$

For the base case $y = -1$, the statement (29) holds $|x_{1,-1}^{\text{new}} - x_{2,-1}^{\text{new}}| = |0 - 0| = 0$. Assume (29) holds for $y = k-1$, i.e.,

$$|x_{1,k-1}^{\text{new}} - x_{2,k-1}^{\text{new}}| \le [1 - (1+\alpha^{2k-3})^{-k+1}] ||x_1^{\text{old}} - x_2^{\text{old}}||_\infty \quad (30)$$

and then for $y = k$.

Case 1: For the case $x_{1,k}^{\text{new}} = x_{2,k}^{\text{new}}$, we have $|x_{1,k}^{\text{new}} - x_{2,k}^{\text{new}}| = 0$ satisfying (29).

Case 2: For the case $x_{1,k}^{\text{new}} > x_{2,k}^{\text{new}}$, there exist two unique constants y_1 and y_2 such that $x_{1,k}^{\text{new}} \in (P_S(y_1), P_S(y_1+1)]$, $x_{2,k}^{\text{new}} \in [P_S(y_2), P_S(y_2+1))$ and $y_1 \le y_2$. By (27)

$$\begin{aligned} & x_{1,k}^{\text{new}} - x_{2,k}^{\text{new}} \\ & \le \left(\frac{x_{1,k+1}^{\text{old}} - x_{1,k-1}^{\text{new}}}{1+\alpha^{2(y_1-k)+1}} + x_{1,k-1}^{\text{new}} \right) - \left(\frac{x_{2,k+1}^{\text{old}} - x_{2,k-1}^{\text{new}}}{1+\alpha^{2(y_2-k)+1}} + x_{2,k-1}^{\text{new}} \right) \\ & \le \left(\frac{x_{1,k+1}^{\text{old}} - x_{1,k-1}^{\text{new}}}{1+\alpha^{2(y_1-k)+1}} + x_{1,k-1}^{\text{new}} \right) - \left(\frac{x_{2,k+1}^{\text{old}} - x_{2,k-1}^{\text{new}}}{1+\alpha^{2(y_1-k)+1}} + x_{2,k-1}^{\text{new}} \right) \\ & = \frac{(x_{1,k+1}^{\text{old}} - x_{2,k+1}^{\text{old}}) + \alpha^{2(y_1-k)+1}(x_{1,k-1}^{\text{new}} - x_{2,k-1}^{\text{new}})}{\alpha^{2(y_1-k)+1} + 1}. \end{aligned} \quad (31)$$

Case 3: For the case $x_{1,k}^{\text{new}} < x_{2,k}^{\text{new}}$, there exist two unique constants y_1 and y_2 such that $x_{1,k}^{\text{new}} \in [P_S(y_1), P_S(y_1+1))$, $x_{2,k}^{\text{new}} \in (P_S(y_2), P_S(y_2+1)]$ and $y_1 \le y_2$. By (27)

$$\begin{aligned} & x_{2,k}^{\text{new}} - x_{1,k}^{\text{new}} \\ & \le \left(\frac{x_{2,k+1}^{\text{old}} - x_{2,k-1}^{\text{new}}}{1+\alpha^{2(y_2-k)+1}} + x_{2,k-1}^{\text{new}} \right) - \left(\frac{x_{1,k+1}^{\text{old}} - x_{1,k-1}^{\text{new}}}{1+\alpha^{2(y_1-k)+1}} + x_{1,k-1}^{\text{new}} \right) \\ & \le \left(\frac{x_{2,k+1}^{\text{old}} - x_{2,k-1}^{\text{new}}}{1+\alpha^{2(y_1-k)+1}} + x_{2,k-1}^{\text{new}} \right) - \left(\frac{x_{1,k+1}^{\text{old}} - x_{1,k-1}^{\text{new}}}{1+\alpha^{2(y_1-k)+1}} + x_{1,k-1}^{\text{new}} \right) \\ & = \frac{(x_{2,k+1}^{\text{old}} - x_{1,k+1}^{\text{old}}) + \alpha^{2(y_1-k)+1}(x_{2,k-1}^{\text{new}} - x_{1,k-1}^{\text{new}})}{\alpha^{2(y_1-k)+1} + 1}. \end{aligned} \quad (32)$$

By (31) and (32) and the assumption (30), we proved that (29) holds for $i \ge 0$ in (33), as shown at the bottom of the next page. Thus, (29) holds. Then condition (28) is proven by (29) as

$$\begin{aligned} \|x_1^{\text{new}} - x_2^{\text{new}}\|_\infty &= \max_i \{|x_{1,i}^{\text{new}} - x_{2,i}^{\text{new}}|\} \\ &\le \max_i \{1 - (1+\alpha^{2i-1})^{-i}\} \|x_1^{\text{old}} - x_2^{\text{old}}\|_\infty \\ &= [1 - (1+\alpha^{2B-3})^{-B+1}] \|x_1^{\text{old}} - x_2^{\text{old}}\|_\infty \end{aligned} \quad (34)$$

and then we have $q = 1 - (1 + \alpha^{-2B-3})^{-B+1} < 1$. Therefore, the $x = \{x_y | y = -1, 0, \ldots, B-1\}$ in Algorithm 1 converges to the unique fixed point. Lemma 2 gives the condition that the optimal P_Y^* must satisfy (19)–(20), and we had proven that x in Algorithm 1 converges to the unique solution, so the unique solution is optimal. ■

VI. Implementation Method

In this section, we propose a coding method to embed the message W into the host sequence S^N by the given cumulative pmfs P_S and P_Y. The proposed coding method adopts four variables (a_1, a_2, a_3, a_4) and $a_1 \le a_2 < a_3 \le a_4$ to store the temporal information during the coding process. The space interval $[a_1, a_4)$ identifies the space of the possible code values, and the information can be interpreted as an arbitrary code value within the code interval $[a_2, a_3)$. The coding framework is similar to the arithmetic coding [24], [25], but we do not restrict the space interval $[a_1, a_4)$ to $[0, 1)$. We can linearly transform the code space (a_1, a_2, a_3, a_4) to another (b_1, b_2, b_3, b_4), where $b_2 = (b_4 - b_1)(a_2 - a_1)/(a_4 - a_1) + b_1$ and $b_3 = (b_4 - b_1)(a_3 - a_1)/(a_4 - a_1) + b_1$. In the following, the s_i and y_i, respectively, represent the values of host signal and the stego-signal at the position i in the sequences.

A. Model

The model follows the design shown in Fig. 2. The encoder maintains two values $(l^{(i)}, u^{(i)}) \subseteq [0, 1]$, $i = 1, 2, \ldots, N$ to interpret the joint information of the host signal S_i and the temporal information produced at the last step. At the initialization, the vector $(l^{(1)}, u^{(1)}) = (P_S(s_1 - 1), P_S(s_1))$ represents the information of the first host signal. At the ith step, the encoder transforms the vector $(l^{(i)}, u^{(i)})$ and the next host s_{i+1} to the stego-signal y_i and the next vector $(l^{(i+1)}, u^{(i+1)})$, expressed as $(y_i, l^{(i+1)}, u^{(i+1)}) = f'(s_{i+1}, l^{(i)}, u^{(i)})$. First, we need to find the possible y_i within the interval $(l^{(i)}, u^{(i)}]$, then $y_i \in [y_1^{(i)}, y_2^{(i)}]$ iff $[P_Y(y_1^{(i)}), P_Y(y_2^{(i)} - 1)) \subset [l^{(i)}, u^{(i)}) \subseteq [P_Y(y_1^{(i)} - 1), P_Y(y_2^{(i)}))$. Second, for the case $y_1^{(i)} = y_2^{(i)}$, the stego-signal $y_i = y_1^{(i)}$ is determined. The $y_1^{(i)}$ destines the interval $[P_Y(y_1^{(i)} - 1), P_Y(y_1^{(i)}))$, which is larger than $(l^{(i)}, u^{(i)}]$, so the decoder needs more information, expressed as the information vector $(P_Y(y_1^{(i)} - 1), l^{(i)}, u^{(i)}, P_Y(y_1^{(i)}))$, for lossless decoding. The next host signal s_{i+1} destines the $[P_S(s_{i+1} - 1), P_S(s_{i+1}))$, and we scale the information vector to $[P_S(s_{i+1} - 1), P_S(s_{i+1}))$ to obtain $(l^{(i+1)}, u^{(i+1)}) = (N(l^{(i)}, s_{i+1}, y_i), N(u^{(i)}, s_{i+1}, y_i))$, where

$$N(x, s, y) = \frac{P_S(s) - P_S(s-1)}{P_Y(y) - P_Y(y-1)}(x - P_Y(y-1)) + P_S(s-1). \tag{35}$$

Then we proceed to process the next host signal. The graphical diagram of this case is shown in Fig. 3(a).

For another case $y_1^{(i)} > y_2^{(i)}$, the y_i has several possible values corresponding to each interval of $(l^{(i)}, P_Y(y_1^{(i)}), \ldots, P_Y(y_2^{(i)} - 1), u^{(i)})$. The *adaptive arithmetic decoding* is applied on the binary representation of message W to determine the value y_i. To fit the usage requirement of the arithmetic coding, the two ends of the vector are scaled to 0 and 1, resulting in

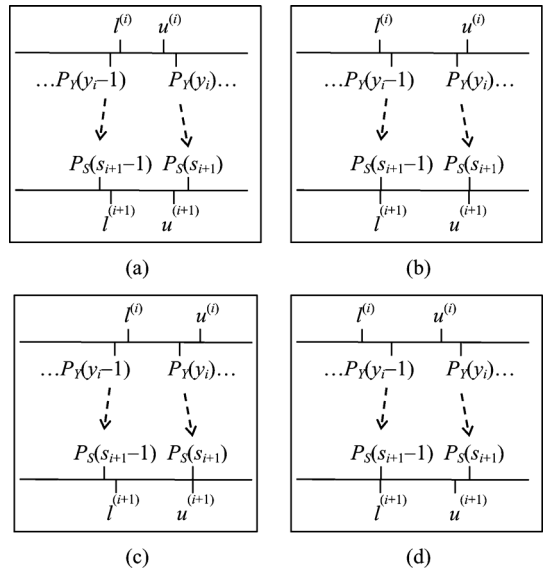


Fig. 3. Four cases of updating $l^{(i)}$ and $u^{(i)}$ in encoding. (a) Case $l^{(i)} \ge P_Y(y_i - 1)$ and $u^{(i)} \le P_Y(y_i)$. (b) Case $l^{(i)} < P_Y(y_i - 1)$ and $u^{(i)} > P_Y(y_i)$. (c) Case $l^{(i)} \ge P_Y(y_i - 1)$ and $u^{(i)} > P_Y(y_i)$. (d) Case $l^{(i)} < P_Y(y_i - 1)$ and $u^{(i)} \le P_Y(y_i)$.

$$\begin{aligned}
\left|x_{1,k}^{\text{new}} - x_{2,k}^{\text{new}}\right| &\le \frac{\left|x_{1,k+1}^{\text{old}} - x_{2,k+1}^{\text{old}}\right| + \alpha^{2(y_1-k)+1}\left|x_{1,k-1}^{\text{new}} - x_{2,k-1}^{\text{new}}\right|}{\alpha^{2(y_1-k)+1} + 1} \\
&\le \frac{\left\|x_1^{\text{old}} - x_2^{\text{old}}\right\|_\infty + \alpha^{2(y_1-k)+1}[1 - (1 + \alpha^{2k-3})^{-k+1}]\left\|x_1^{\text{old}} - x_2^{\text{old}}\right\|_\infty}{\alpha^{2(y_1-k)+1} + 1} \\
&= \left[1 - \frac{(1 + \alpha^{2k-3})^{-k+1}}{1 + \alpha^{-2(y_1-k)-1}}\right]\left\|x_1^{\text{old}} - x_2^{\text{old}}\right\|_\infty \\
&\le \left[1 - \frac{(1 + \alpha^{2k-3})^{-k+1}}{1 + \alpha^{2k-1}}\right]\left\|x_1^{\text{old}} - x_2^{\text{old}}\right\|_\infty \\
&< \left[1 - (1 + \alpha^{2k-1})^{-k}\right]\left\|x_1^{\text{old}} - x_2^{\text{old}}\right\|_\infty
\end{aligned} \tag{33}$$

$(0, R(P_Y(y_1^{(i)}), l^{(i)}, u^{(i)}), \ldots, R(P_Y(y_2^{(i)} - 1), l^i, u^{(i)}), 1)$ where

$$R(x, l, u) = \frac{(x - l)}{(u - 1)}. \tag{36}$$

The *arithmetic decoder* determines the value y_i, and the corresponding residual information depends on the y_i values discussed as follows. For the case $y_1^{(i)} + 1 \le y_i \le y_2^{(i)} - 1$, the $[P_Y(y_i - 1), P_Y(y_i))$ is in the $(l^{(i)}, u^{(i)}]$, so we do not have the residual information. As shown in Fig. 3(b), the updated vector is $(l^{(i+1)}, u^{(i+1)}) = (P_S(s_{i+1} - 1), P_S(s_{i+1}))$ representing the next host signal. For the case $y_i = y_1^{(i)}$, the residual information is interpreted as the vector $(P_Y(y_i - 1), l^{(i)}, P_Y(y_i), P_Y(y_i))$, which is scaled into the $[P_S(s_{i+1} - 1), P_S(s_{i+1}))$ to obtain the updated vector $(l^{(i+1)}, u^{(i+1)}) = (N(l^{(i)}, s_{i+1}, y_i), P_S(s_{i+1}))$ shown in Fig. 3(c). For the case $y_i = y_2^{(i)}$, the residual information is interpreted as the vector $(P_Y(y_i - 1), P_Y(y_i - 1), u^{(i)}, P_Y(y_i))$, which is scaled into the $[P_S(s_{i+1} - 1), P_S(s_{i+1}))$ to obtain the updated vector $(l^{(i+1)}, u^{(i+1)}) = (P_S(s_{i+1} - 1), N(u^{(i)}, s_{i+1}, y_i))$ shown in Fig. 3(d). By observing the four cases shown in Fig. 3, we summarize the updating step as

$$l^{(i+1)} = \max\{P_S(s_{i+1} - 1), N(l^{(i)}, s_{i+1}, y_i)\}$$

and

$$u^{(i+1)} = \min\{P_S(s_{i+1}), N(u^{(i)}, s_{i+1}, y_i)\}. \tag{37}$$

At the final step, there is no next host signal, and after designating the y_N the encoder sends out a prefix-free code value $v \in [\tilde{N}, (l^{(N)}, y_N), \tilde{N}(l^{(N)}, y_N))$ with minimal code length, where

$$\tilde{N}(x, y) = \frac{(x - P_Y(y - 1))}{(P_Y(y) - P_Y(y - 1))}. \tag{38}$$

The decoding process consists of two processes: The host-sequence decoder reads the received stego-sequence in a backward manner, to produce the host sequence and the message decoder extracts the message through simulating the encoding steps. The host-sequence decoder transforms the stego-signal y_i into the host signal $\hat{s}_i$ by maintaining a real number $v^{(i)} \in [l^{(i)}, u^{(i)})$ at the $(N - i + 1)$th step. The initialization is given as $v^{(N)} = \tilde{M}(v, y_N)$, where

$$\tilde{M}(x, y) = (P_Y(y) - P_Y(y - 1))x + P_Y(y - 1). \tag{39}$$

Equation (39) is the inverse function of (38) by fixing the variable y, so $v^{(N)} \in [l^{(N)}, u^{(N)})$. At the $(N - i + 1)$th step, the decoder determines the host value $\hat{s}_i$ under $P_S(\hat{s}_i - 1) \le v^{(i)} < P_S(\hat{s}_i)$ and then calculates $v^{(i-1)} = M(v^{(i)}, \hat{s}_i, y_{i-1})$ where

$$M(x, s, y) = \frac{P_Y(y) - P_Y(y - 1)}{P_S(s) - P_S(s - 1)}(x - P_S(s - 1)) + P_Y(y - 1). \tag{40}$$

It is noted that (40) is the inverse function of (35) by fixing the two variables s, y. Thus, the host-sequence decoder can losslessly reconstruct the host sequence in reverse order. The message decoder simulates the encoding steps upon the decoded version of host signals and the chosen indices of the *arithmetic decoder* are recognized by the stego-signals. The message is reconstructed by the *adaptive arithmetic encoding*. The decoder initializes the vector $(l^{(1)}, u^{(1)}) = (P_S(\hat{s}_1 - 1), P_S(\hat{s}_1))$. At the ith step, the decoder determines the $y_1^{(i)}, y_2^{(i)}$ satisfying $[P_Y(y_1^{(i)}), P_Y(y_2^{(i)} - 1)) \subset [l^{(i)}, u^{(i)}) \subseteq [P_Y(y_1^{(i)} - 1), P_Y(y_2^{(i)}))$; for the case $y_1^{(i)} = y_2^{(i)}$, we have $l^{(i+1)} = N(l^{(i)}, \hat{s}_{i+1}, y_i)$ and $u^{(i+1)} = N(u^{(i)}, \hat{s}_{i+1}, y_i)$, and then go to process the next signal. For another case $y_1^{(i)} < y_2^{(i)}$, the *adaptive arithmetic encoding* is applied; in the interval $[l^{(i)}, u^{(i)})$, the interval for y_i is the chosen index of the *arithmetic decoder* in the encoding process, so we input the scaled interval, expressed as $[\max\{R(P_Y(y_i - 1), l^{(i)}, u^{(i)}), 0\}, \min\{R(P_Y(y_i), l^{(i)}, u^{(i)}), 1\})$, to the *adaptive arithmetic encoder*. The updated $[l^{(i+1)}, u^{(i+1)})$ is calculated through (37), and then we process the next signal until finishing the decoding.

Within the context of binary i.i.d. sequence, it is noted that the proposed model is similar to the recursive reversible code [10]. In recursive reversible code, the host sequence s^N is segmented into disjoint sectors with length K. At the ith step, the message w_i is embedded in the ith segment s_i^K, resulting in the stego sector y_i^K and the information needed to reconstruct s_i^K. The recursive reversible code recursively embeds the reconstruction information in the next segment s_{i+1}^K. We observe that the theoretical rate-distortion bound of the proposed codes is identical to the recursive reversible code by comparing Example 1 and [10, Th. 2] for the binary signals. However, [10] does not clearly address the code constructions for ray-scale case, so the proposed codes can be viewed as a practical version of recursive reversible code [10] for gray-scale signals. Another difference is that the proposed codes can be viewed as another version of recursive construction by setting the segment length to be equal to 1.

B. Implementation With Integer Arithmetic

It is computationally unfeasible to maintain $l^{(i)}$ and $u^{(i)}$ with infinite precision real numbers, so the proposed feasible algorithm adopts the $b^{(i)}$-bit integer arithmetic to implement the $l^{(i)}$ and $u^{(i)}$. The $L^{(i)}$ and $U^{(i)}$ denote the fractional parts of $l^{(i)}$ and $u^{(i)}$ with $b(i)$-bit integers. Then (35) is reformulated as

$$N^{[b]}(x, s, y) = \left\lfloor \frac{P_S(s) - P_S(s - 1)}{P_Y(y) - P_Y(y - 1)}(x - P_Y(y - 1) \times 2^b) + P_S(s - 1) \times 2^b \right\rfloor. \tag{41}$$

After designating the stego-signal $Y_i = y_i$, the formulas of next boundaries are expressed as

$$\begin{aligned} L^{(i+1)} &= \left(\max\{P_S^{[b(i)]}(\hat{s}_{i+1} - 1), N^{[b(i)]}(L^{(i)}, \hat{s}_{i+1}, y_i)\} + 1\right) \\ &\quad \times 2^{b(i+1) - b(i)}; \\ U^{(i+1)} &= \min\{P_S^{[b(i)]}(\hat{s}_{i+1}), N^{[b(i)]}(U^{(i)}, \hat{s}_{i+1}, y_i)\} \\ &\quad \times \{2^{b(i+1) - b(i)}\}. \end{aligned} \tag{42}$$

At the final step, the encoder sends out a prefix-free code $V \in [\tilde{N}^{[b(N)]}(l^{(N)}, y_N) \times 2^{v_b - b(N)}, \tilde{N}^{[b(N)]}(u^{(N)}, y_N) \times 2^{v_b - b(N)})$ with $v_b = b(N) + 2 - \lfloor \log_2(U^{(N)} - L^{(N)}) \rfloor$ bits, where

$$\tilde{N}^{[b]}(x, y) = \left\lfloor \frac{x - P_Y(y-1) \times 2^b}{P_Y(y) - P_Y(y-1)} \right\rfloor \tag{43}$$

is reformulated from (38). The number of fractional bits $b(i)$ affects the performance and efficiency of the algorithm. If $b(i)$ is a constant during the encoding, the encoding may suffer the computational precision problem in certain special cases. For example, (41) gives the inequality $U^{(i+1)} - L^{(i+1)} \leq (U^{(i)} - L^{(i)}) \times (P_S(s) - P_S(s-1))/(P_Y(y) - P_Y(y-1))$, and if a long period of encoding steps has the property $P_S(s) - P_S(s-1) \leq P_Y(y) - P_Y(y-1)$, the $U^{(i)}$ gradually approaches the $L^{(i)}$ until the two values coincide, and the further encoding is impossible. Thus, the encoder dynamically adjusts the number of fractional bits $b(i)$ to fit the condition $2^m \leq U^{(i)} - L^{(i)} < 2^{m+1}$ during the encoding process, where m is a user-defined constant integer. Algorithm 2 shows the proposed encoder. In Algorithm 2, we use the notations $P_j^{[b]} = \lfloor P_j \times 2^b \rfloor$, and $Q_k^{[b]} = \lfloor Q_k \times 2^b \rfloor$; the instruction $ArithmeticDecoder(W)$ denotes applying the adaptive arithmetic coding decoder on the W; and the instruction *DecodeNextSymbol(symbols, intervals)* outputs a signal of the vector *symbols* with decoding the W, where each element in the vector *symbols* respectively corresponds to an interval of the vector $intervals = (0, \ldots, 1)$. Line 24 of Algorithm 2 records the maximal number of fractional bits c to facilitate decoding.

Algorithm 2: The encoder.

Input: The host sequence S^N, the binary sequence W, and the cumulative pmfs P_S and P_Y.

Output: The stego sequence Y^N, an integer c, and a binary float $v \in [0, 1)$.

1. $ArithmeticDecoder(W)$
2. $b = m - \lceil \log_2(P_S(s_1) - P_S(s_1 - 1)) \rceil$
3. $L = P_S^{[b]}(s_1 - 1) + 1$, and $U = P_S^{[b]}(s_1)$
4. $c = b$
5. **for** $i = 1$ to $N - 1$ **do**
6. Determine y_1 and y_2 under $P_Y^{[b]}(y_1 - 1) \leq L < P_Y^{[b]}(y_1)$ and $P_Y^{[b]}(y_2 - 1) < U \leq P_Y^{[b]}(y_2)$.
7. **if** $y_1 = y_2$ **then**
8. $y_i = y_1$
9. $L = N^{[b]}(L, s_{i+1}, y_i) + 1$, and $U = N^{[b]}(U, s_{i+1}, y_i)$
10. **else**
11. $intervals = (0, R(P_Y(y_1), L, U), \ldots, R(P_Y(y_2 - 1), L, U), 1)$
12. $symbols = (y_1, \ldots, y_2)$
13. $y_i = DecodeNextSymbol(symbols, intervals)$
14. **if** $y_1 + 1 \leq y_i \leq y_2 - 1$ **then**
15. $L = P_S^{[b]}(s_{i+1} - 1) + 1$, and $U = P_S^{[b]}(s_{i+1})$
16. **else if** $y_i = y_1$ **then**
17. $L = N^{[b]}(L, s_{i+1}, y_i) + 1$, and $U = P_S^{[b]}(s_{i+1})$
18. **else if** $y_i = y_2$ **then**
19. $L = P_S^{[b]}(s_{i+1} - 1) + 1$, and $U = N^{[b]}(U, s_{i+1}, y_i)$
20. **end if**
21. **end if**
22. $shift = m - \lfloor \log_2(U - L) \rfloor$
23. $L = L \times 2^{shift}$, $U = U \times 2^{shift}$, and $b = b + shift$
24. $c = \max\{c, b\}$
25. **end for**
26. Determine y_1 and y_2 under and $P_Y^{[b]}(y_1 - 1) \leq L < P_Y^{[b]}(y_1)$ and $P_Y^{[b]}(y_2 - 1) < U \leq P_Y^{[b]}(y_2)$.
27. **if** $y_1 = y_2$ **then**
28. $y_N = y_1$
29. $L = \tilde{N}^{[b]}(L, y_N) + 1$ and $U = \tilde{N}^{[b]}(U, y_N)$
30. **else**
31. $interval = (0, R(P_Y(y_1), L, U), \ldots, R(P_Y(y_2 - -1), L, U), 1)$
32. $symbols = (y_1, \ldots, y_2)$
33. $y_N = DecodeNextSymbol(symbols, interval)$
34. **if** $y_1 + 1 \leq y_N \leq y_2 - 1$ **then**
35. $L = 0$, and $U = 2^b - 1$
36. **else if** $y_N = y_1$ **then**
37. $L = \tilde{N}^{[b]}(L, y_N) + 1$, and $U = \tilde{N}^{[b]}(U, y_N)$
38. **else if** $y_N = y_2$ **then**
39. $L = 0$, and $U = \tilde{N}^{[b]}(U, y_N)$
40. **end if**
41. **end if**
42. Send out a prefix-free code word $v \in [L \times 2^{v_b - b}, U \times 2^{v_b - b}]$ of $v_b = b + 2 - \lfloor \log_2(U^{(N)} - L^{(N)}) \rfloor$ bits.

Transmitting the side information P_S and P_Y is an extra overhead for the proposed algorithm. The two constants $P_S(0) = 0$ and $P_S(B) = 1$ can be truncated to reduce the transmission size, and other $P_S(s)$ are individually coded with w-bit fractional numbers, so storing P_S requires $w(B-1)$ bits, similarly for P_Y. Thus, the total $2w(B-1)$ bits are required. Another strategy is to only transfer P_S, and the decoder side generates the P_Y with Algorithm 1 by synchronizing the initialization and the stop criterion at encoder and decoder sides. This strategy consumes $w(B-1)$ bits to transmit P_S, but the decoder side requires the computational cost to run Algorithm 1.

Algorithm 3–1 shows the procedure of decoding the host sequence. We define the digit $V^{(i)}$ to express the fractional parts of $v^{(i)}$ with a c-bits integer. Equation (40) is reformulated as

$$M^{[c]}(x, s, y) = \left\lfloor \frac{P_Y(y) - P_Y(y-1)}{P_S(s) - P_S(s-1)} (x - P_S(s-1) \times 2^c) + P_Y(y-1) \times 2^c \right\rfloor. \tag{44}$$

Algorithm 3–1: The host-sequence decoder.

Input: The stego-sequence Y^N, an integer c, a binary code word v, and the cumulative pmfs P_S, P_Y.

Output: The host sequence $\hat{S}^N$.

1. $V = \tilde{M}^{[c]}(v/v^b, y_N)$

2. Determine the $\hat{s}_N$ under $P_S^{[c]}(\hat{s}_N - 1) \le V < P_s^{[c]}(\hat{s}_N)$.
3. **for** $i = N - 1$ **to** 1 **do**
4. $V = M^{[c]}(V, \hat{s}_{i+1}, y_i)$
5. Determine the $\hat{s}_i$ under $P_S^{[c]}(\hat{s}_i - 1) \le V < P_S^{[c]}(\hat{s}_i)$.
6. **end for**

The initialization is given as $V^{(1)} = \tilde{M}^{[c]}(v, y_N)$ where

$$\tilde{M}^{[c]}(x,y) = \Bigg\lfloor (P_Y(y) - P_Y(y-1))x \times 2^{c-v_b}) + P_Y(y-1) \times 2^c \Bigg\rfloor. \quad (45)$$

Lemma 5 proves that $V^{(i)}$ is always within $[L^{(i)} \times 2^{c-b(i)}, U^{(i)} \times 2^{c-b(i)}]$ at each decoding step, so the Algorithm 3–1 can correctly decode the host sequence.

Lemma 5: Each $V^{(i)}$ calculated by (44) must satisfy

$$L^{(i)} \times 2^{c-b(i)} \le V^{(i)} \le U^{(i)} \times 2^{c-b(i)}. \quad (46)$$

Proof: For $i = 1$, by Line 26–42 of Algorithm 2, we have

$$\begin{aligned} &\max\left\{0, \tilde{N}^{[b(N)]}(L^{(N)}, y_N) + 1\right\} \times 2^{v_b - b(N)} \\ &\quad \le v \le \min\left\{2^b - 1, \tilde{N}^{[b(N)]}(U^{(N)}, y_N)\right\} \times 2^{v_b - b(N)} \\ &\quad \Rightarrow (\tilde{N}^{[b(N)]}(L^{(N)}, y_N) + 1) \times 2^{v_b - b(N)} \\ &\quad \le v \le \tilde{N}^{[b(N)]}(U^{(N)}, y_N) \times 2^{v_b - b(N)}. \end{aligned} \quad (47)$$

Substituting (47) in (43), we have

$$\begin{aligned} &\left(\left\lfloor \frac{L^{(N)} - P_Y(y_N - 1) \times 2^{b(N)}}{P_Y(y_N) - P_Y(y_N - 1)} \right\rfloor + 1\right) \times 2^{v_b - b(N)} \\ &\quad \le v \le \left\lfloor \frac{U^{(N)} - P_Y(y_N - 1) \times 2^{b(N)}}{P_Y(y_N) - P_Y(y_N - 1)} \right\rfloor \times 2^{v_b - b(N)} \\ &\quad \Rightarrow \frac{L^{(N)} - P_Y(y_N - 1) \times 2^{v_b}}{P_Y(y_N) - P_Y(y_N - 1)} \\ &\quad \le v \le \frac{U^{(N)} - P_Y(y_N - 1) \times 2^{v_b}}{P_Y(y_N) - P_Y(y_N - 1)} \\ &\quad \Rightarrow L^{(N)} \le (P_Y(y_N) - P_Y(y_N - 1))v \\ &\qquad + P_Y(y_N - 1) \times 2^{v_b} \le U^{(N)} \\ &\quad \Rightarrow L^{(N)} \times 2^{B - b(N)} \le (P_Y^{[c]}(y_N) \\ &\qquad - P_Y^{[c]}(y_N - 1))v \times 2^{c - v_b} \\ &\qquad + P_Y^{[c]}(y_N - 1)2^c \le \{U^{(N)}\} \times 2^{c - b(N)} \\ &\quad \Rightarrow L^{(N)} \times 2^{c-b(N)} \le V^{(1)} \le U^{(N)} \times 2^{c-b(N)}. \end{aligned} \quad (48)$$

Thus, (46) holds for $i = 1$. Assume the inequality holds for $i = k - 1$, i.e., $L^{(k-1)} \times 2^{c-b(k-1)} \le V(k-1) \le U^{(k-1)} \times 2^{c-b(k-1)}$. For $i = k$, we first prove the left inequality of (46). By (44) and the assumption, we have

$$\begin{aligned} V^{(k)} &= M^{[c]}(V^{(k-1)}, \hat{s}_{k-1}, y_{k-1}) \\ &\ge M^{[c]}(L^{(k-1)} \times 2^{c-b(k-1)}, \hat{s}_{k-1}, y_{k-1}) \\ &= \left\lfloor \frac{(P_Y(y_k) - P_Y(y_k - 1))}{P_S(\hat{s}_{k-1}) - P_S(\hat{s}_{k-1} - 1)} (L^{(k-1)} \times 2^{c-b(k-1)} \right. \\ &\qquad - P_S(\hat{s}_{k-1} - 1) \times 2^c) \\ &\qquad \left. + P_Y(y_k - 1) \times 2^c \right\rfloor. \end{aligned} \quad (49)$$

By (42) and (41), the lower bound of $L^{(k-1)}$ is given by

$$\begin{aligned} L^{(k-1)} &= \Big(\max\{P_S^{[b(k)]}(\hat{s}_{k-1} - 1), N^{[b(k)]} \\ &\qquad \times (L^{(k)}, \hat{s}_{k-1}, y_k)\} + 1\Big) \\ &\qquad \times 2^{b(k-1) - b(k)} \\ &\ge \left(N^{[b(k)]}(L^{(k)}, \hat{s}_{k-1}, y_k) + 1\right) \times 2^{b(k-1) - b(k)} \\ &\ge \frac{P_S(\hat{s}_{k-1}) - P_S(\hat{s}_{k-1} - 1)}{P_Y(y_k) - P_Y(y_k - 1)} \\ &\qquad \times (L^{(k)} \times 2^{b(k-1) - b(k)} - P_Y(y_k - 1) \times 2^{b(k-1)}) \\ &\qquad + P_S(\hat{s}_{k-1} - 1) \times 2^{b(k-1)}. \end{aligned} \quad (50)$$

The left inequality of (46) is proved by substituting the term $L^{(k-1)}$ in (49) with (50). Then we have

$$V^{(k)} \ge \left\lfloor L^{(k)} \times 2^{c-b(k)} \right\rfloor = L^{(k)} \times 2^{c-b(k)}. \quad (51)$$

For the right inequality of (46), by (44), we have

$$\begin{aligned} V^{(k)} &= M^{[c]}(V^{(k-1)}, \hat{s}_{k-1}, y_k) \\ &\le M^{[c]}(U^{(k-1)} \times 2^{c-b(k-1)}, \hat{s}_{k-1}, y_k) \\ &= \left\lfloor \frac{P_Y(y_k) - P_Y(y_k - 1)}{P_S(\hat{s}_{k-1}) - P_S(\hat{s}_{k-1} - 1)} \right. \\ &\qquad \times (U^{(k-1)} \times 2^{c-b(k-1)} - P_S(\hat{s}_{k-1} - 1) \times 2^c) \\ &\qquad \left. + P_Y(y_k - 1) \times 2^c \right\rfloor. \end{aligned} \quad (52)$$

By (42), the upper bound of $U^{(k-1)}$ is

$$\begin{aligned} U^{(k-1)} &= \min\{P_S^{[b(k)]}(\hat{s}_{k-1}), N^{[b(k)]}(U^{(k)}, \hat{s}_{k-1}, y_k)\} \\ &\qquad \times 2^{b(k-1) - b(k)} \\ &\le N^{[b(k)]}(U^{(k)}, \hat{s}_{k-1}, y_k) \times 2^{b(k-1) - b(k)} \\ &= \frac{P_S(\hat{s}_{k-1}) - P_S(\hat{s}_{k-1} - 1)}{P_Y(y_k) - P_Y(y_k - 1)} \\ &\qquad \times (U^{(k)} \times 2^{b(k-1) - b(k)} - P_Y(y_k - 1) \times 2^{b(k-1)}) \\ &\qquad + P_S(\hat{s}_{k-1} - 1) \times 2^{b(k-1)}. \end{aligned} \quad (53)$$

The right inequality of (46) is proved by substituting the term $U^{(N-k+2)}$ in (52) with (53). Then we have

$$V^{(k)} \le \left\lfloor U^{(k)} \times 2^{c-b(k)} \right\rfloor = U^{(k)} \times 2^{c-b(k)}. \quad (54)$$

■

Algorithm 3–2 gives the message decoder. The encoding intervals are identified by simulating the encoding steps and the corresponding stego-signals. The instruction

$EncodeNextSymbol(l', u')$ denotes that the adaptive arithmetic coding encodes the given interval $[l', u') \subseteq [0, 1)$, and the $GetSequence()$ outputs the encoded sequence.

Algorithm 3–2: The message decoder.

Input: The decoded host sequence $\hat{S}^N$, the stego-sequence Y^N, and the cumulative pmfs P_S and P_Y.

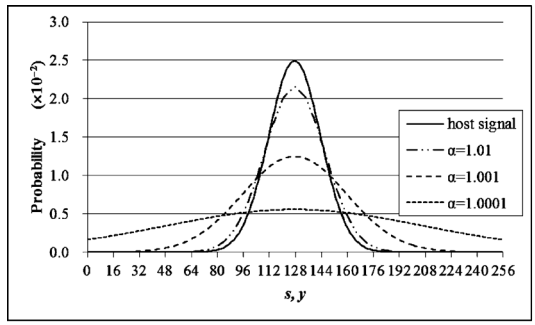
Output: The binary representation message W.

$b = m - \lceil \log_2(P_S(\hat{S}_1) - P_S(\hat{S}_1 - 1)) \rceil$
$L = P_S^{[b]}(\hat{s}_1 - 1) + 1$, and $U = P_S^{[b]}(\hat{S}_1)$
for $i = 1$ **to** $N - 1$ **do**
  Determine y_1 and y_2 under $P_Y^{[b]}(y_1 - 1) \le L < P_Y^{[b]}(y_1)$ and $P_Y^{[b]}(y_2 - 1) < U \le P_Y^{[b]}(y_2)$.
  if $y_1 = y_2$ **then**
    $L = N^{[b]}(L, \hat{S}_{i+1}, y_i) + 1$, and $U = N^{[b]}(U, \hat{S}_{i+1}, y_i)$
  else
    $l' = R(\max\{L, P_Y(y_i - 1)\}, L, U)$
    $u' = R(\min\{U, P_Y(y_i)\}, L, U)$
    $EncodeNextSymbol(l', u')$
    if $y_1 + 1 \le y_i \le y_2 - 1$ **then**
      $L = P_S^{[b]}(\hat{S}_{i+1} - 1) + 1$, and $U = P_S^{[b]}(\hat{S}_{i+1})$
    else if $y_i = y_1$ **then**
      $L = N^{[b]}(L, \hat{S}_{i+1}, y_i) + 1$, and $U = P_S^{[b]}(\hat{S}_{i+1})$
    else if $y_i = y_2$ **then**
      $L = P_S^{[b]}(\hat{S}_{i+1} - 1) + 1$, and $U = N^{[b]}(U, \hat{S}_{i+1}, y_i)$
    end if
  end if
  $shift = m - \lceil \log_2(U - L) \rceil$
  $L = l \times 2^{shift}$, $U = u \times 2^{shuft}$, and $b = b + shift$
end for
Determine y_1 and y_2 under $P_Y^{[b]}(y_1 - 1) \le L < P_Y^{[b]}(y_1)$ and $P_Y^{[b]}(y_2 - 1) < U \le P_Y^{[b]}(y_2)$.
if $y_1 \ne y_2$ **then**
  $l' = R(\max\{L, P_Y(y_i - 1)\}, L, U)$
  $u' = R(\min\{U, P_Y(y_i)\}, L, U)$
  $EncodeNextSymbol(l', u')$
end if
$W = Getsequence()$

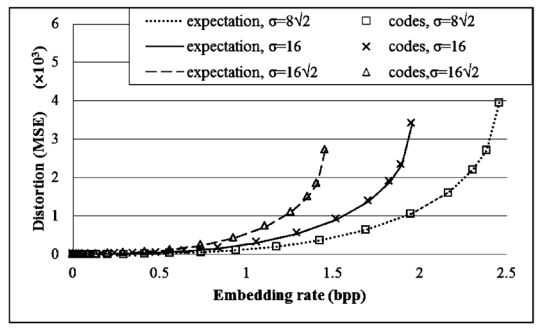
VII. Experiments

A. Experiments for i.i.d. Sequence

Fig. 4 shows the results of the purposed algorithms for the host sequences drawn from discrete normal distributions. The host signals are 8-bit gray-scale with $B = 256$, and the mean of the normal distribution is at 127.5. Fig. 4(a) illustrates the pmf of the host signal for the variance $\sigma = 16$, and the pmfs of the corresponding stego-signals by applying Algorithm 1, for $\alpha = 1.01, 1.001$ and 1.0001. In our implementation of Algorithm 1, the variable x_y is stored with a 52-bits fractional number, and the tolerance value ε is set to zero; i.e., the output pmf p_Y is a fixed point on the 52-bits fractional number space. For $\alpha = 1.01$, the embedding rate is 0.222 bits per pixel (bpp),



(a)



(b)

Fig. 4. Demonstrations for host sequences drawn from normal distributions. (a) The pmf of the host signal versus pmfs of the corresponding stego-signal for various α. (b) Rate-distortion curves of the given normal distributions for various variances σ, and rate-distortion values of implementation codes.

and the mean squared error (MSE) is 7.278; for $\alpha = 1.001$, the embedding rate is 0.992 bpp, and the MSE is 250.396; for $\alpha = 1.0001$, the embedding rate is 1.871 bpp, and the MSE is 2165.611. Fig. 4(a) shows that with the value of α approaching 1 from the right, the pmfs p_Y regress to uniform distribution, so that the embedding rate and the distortion are both increased. Fig. 4(b) shows the rate-distortion curves for the pmfs p_S with variances σ, where the $m = 22$ is used in Algorithm 2. The dotted line, solid line and dashed line, respectively, depict the expected rate-distortion curves for $\sigma = 8\sqrt{2}, 16$, and $16\sqrt{2}$. We implement Algorithm 2 with $m = 22$ to embed the message in 65 536 host signals for each corresponding pmf p_S and p_Y, and the rate-distortion values are respectively marked as squares, crosses and triangles in Fig. 4(b). It is shown that the performance of the proposed algorithm is very close to the expected rate-distortion bound.

Fig. 5 shows another result of the proposed algorithms for the host sequences drawn from discrete Laplace distributions. The host signals are also stored with 8-bit gray-scales, and the mean of the Laplace distribution μ is at 127.5. Fig. 5(a) illustrates the pmfs of the host signals for the scale parameter $b = 16$, and the pmfs of the corresponding stego-signals by using Algorithm 1. For $\alpha = 1.01, 1.001$ and 1.0001, the embedding rate and the MSE are respectively shown as (0.172, 5.528), (0.766, 196.323) and (1.489, 1784.877). Fig. 5(b) illustrates the rate-distortion curves for the pmfs p_S with various scale parameters b. For $b = 8\sqrt{2}, 16$, and $16\sqrt{2}$, the dotted line, solid line and dashed line respectively depict the expected rate-distortion curves; and the squares, crosses and triangles respectively mark the rate-distortion values of the codes with implementing Algorithm 2 on 65 536 host signals.

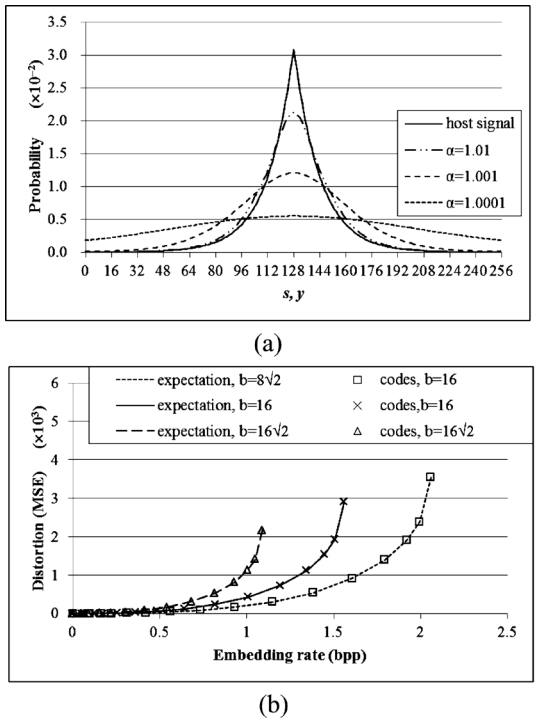


Fig. 5. Demonstrations for host sequences drawn from Laplace distributions. (a) The pmf of host sequence versus pmfs of stego-sequences for various α. (b) Rate-distortion curves of given Laplace distributions corresponding to various scalar parameter b, and rate-distortion values of implementation codes.

B. Reversible Information Embedding for Gray-Scale Images

We present a method to apply the proposed coding scheme to gray-scale images. To reduce the correlations of the neighboring image pixels, we preprocess the host image with predictive coding and then embed the message in the difference map. The predictive coding processes the host pixels from left to right, and from top to bottom. Fig. 6(a) depicts the four neighboring pixels used for predicting the pixel h_i. The predicted value is defined as

$$\tilde{h}_i = \frac{3}{8} \times a_0 + \frac{3}{8} \times a_2 + \frac{1}{8} \times a_1 + \frac{1}{8} \times a_3. \tag{55}$$

For the pixels at rightmost column and bottom line, we pick the nearby pixel, i.e., a_2 or a_0, as the predicted value. The predictive coding omits the bottom-right corner pixel. Then we have the differenced values defined as

$$\tilde{d}_i = (h_i - \tilde{h}_i + 128) \bmod 256. \tag{56}$$

The message is embedded in the differenced values by applying the embedding scheme (Algorithm 1 and Algorithm 2), resulting in a stego-sequence. For each stego-value $\tilde{y}_i$ in the stego-sequence, the corresponding pixel of the watermarked image is defined as

$$\tilde{w}_i = (\tilde{y}_i + \tilde{h}_i - 128) \bmod 256. \tag{57}$$

We observe that

$$|\tilde{w}_i - h_i| = \left|(\tilde{y}_i + \tilde{h}_i - 128) \bmod 256 - (\tilde{d}_i + \tilde{h}_i - 128) \bmod 256\right| \in \left\{\left|\tilde{y}_i - \tilde{d}_i\right|, 256 - \left|\tilde{y}_i - \tilde{d}_i\right|\right\}.$$

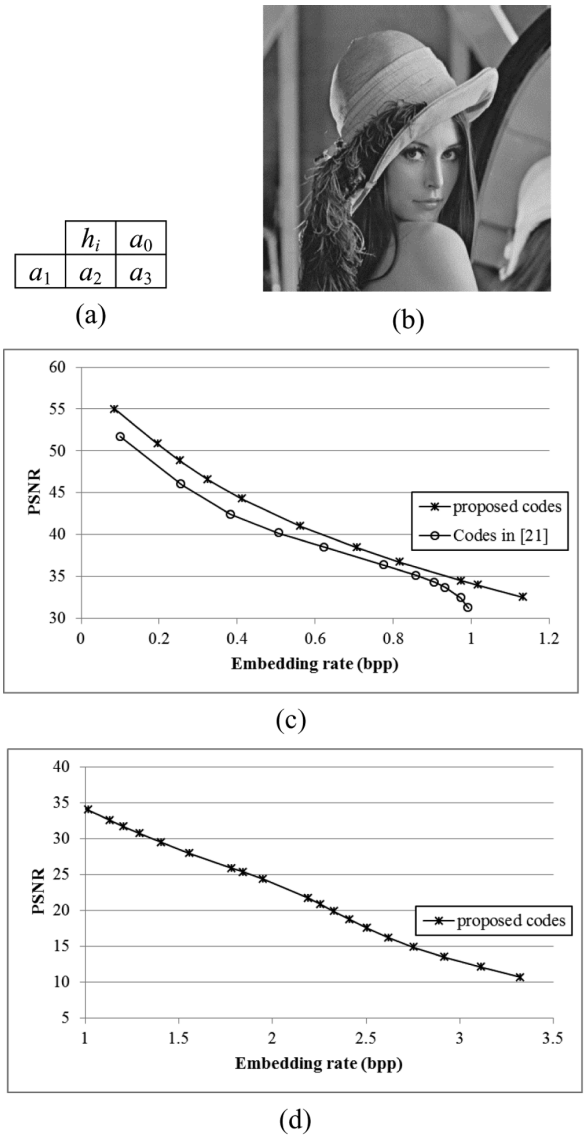


Fig. 6. Proposed scheme applied to gray-scale images. (a) Four neighboring pixels adopted in predictor. (b) Host image Lena. (c) Comparisons with Thodi and Rodriguez [21]. (d) Rate-distortion curve of the proposed scheme for larger embedding rate.

The event $|\tilde{w}_i - h_i| = \left|\tilde{y}_i - \tilde{d}_i\right|$ usually occurs while $\tilde{y}_i$ is quite close to $\tilde{d}_i$. Thus, the distortion is almost dominated by the D_{av} given in the embedding scheme.

In decoding, we first decode the host image in the reverse order from right-to-left, and bottom-to-top. Similar to the host-sequence decoder (Algorithm 3–1), the host image decoder maintains a variable V by transforming the stego-value $\tilde{y}_i$ to the differenced value $\tilde{d}_i$. Consider the step of decoding h_i shown in Fig. 6(a), the four neighboring host pixels a_0, a_1, a_2, a_3 had been decoded by the previous decoding steps. Therefore we have the predicted value $\tilde{h}_i$ through (55). By reversing (57), we have the stego-value

$$\tilde{y}_i = (\tilde{w}_i - \tilde{h}_i + 128) \bmod 256. \tag{58}$$

Then the decoder enters the host-sequence decoder mode. The value of V is updated by (44) $V = M^{[c]}(V, \tilde{d}_{i+1}, \tilde{y}_i)$, and then we determine the difference value $\tilde{d}_i$ under $P_S^{[c]}(\tilde{d}_i - 1) \le V <$

$P_S^{[c]}(\tilde{d}_i)$. Then by reversing (56), the host value is calculated through

$$h_i = (\tilde{d}_i + \tilde{h}_i - 128) \bmod 256. \tag{59}$$

After decoding the host image, the message can be extracted from the obtained difference values with Algorithm 3–2. We compare the above method with the histogram shifting method [21] proposed by Thodi and Rodriguez. Fig. 6(b) shows the test image Lena, and Fig. 6(c) draws the rate-distortion curves of [21] and the proposed RIE. Furthermore, the proposed RIE is capable of larger embedding rates shown in Fig. 6(d).

VIII. Discussion

A. Coding Scheme With Other Distortion Metrics

The square error distortion is the adopted distortion metric in the paper. For other distortion metrics, in order to efficiently obtain the optimal P_Y and $P_{S,Y}$ through iterative algorithm, the adopted distortion metric should follow the crossing-edges property shown in Corollary 1, where, for $s_1 \leq s_2$ and $y_1 \leq y_2$, the distortion metric must satisfy the inequality $D(s_1, y_1) + D(s_2, y_2) \leq D(s_1, y_2) + D(s_2, y_1)$. One example is to replace the $D(s, y)$ with L1-Norm $D_1(s, y) = |s - y|$.

B. Hiding Efficiency of the Proposed Coding Scheme

The proposed coding scheme consists of p_Y computation and the practical coding algorithm. There are two major factors decreasing the efficiency of the proposed scheme: 1) The accuracy of the computed p_Y and 2) the efficiency lost in the encoding process. For the first factor, the iterative method updates the $p_Y(y)$ by using the two nearby elements $p_Y(y-1)$ and $p_Y(y+1)$. If the nearby element is incorrect due to the machine precisions, the precision error will influence the updating. In our experiments, for a given P_S, if the x in Algorithm 1 is stored with finite precision fractional numbers, it is still feasible to obtain different outputs from differential initializations, even though the tolerance value ε is set to zero. For the second factor, the practical algorithm cannot store the real interval $[l^{(i)}, u^{(i)})$ due to the machine precision. Suppose that (l, u) is the precise value during encoding, Algorithm 2 maintains the variables $L = (\lfloor l \rfloor + 1) \times 2^b$, and $U = \lfloor u \rfloor \times 2^b$ with length b. Thus, the interval $[L, U)$ is slightly narrower than the original interval, which will slightly decrease the capacity of the coding algorithm. The two major factors come from the precisions of the computing machines, and therefore the strategy to improve the efficiency is to increase the calculating precisions. With the theoretical machine with near-infinite precisions, we conjecture that the proposed codes can almost approach the theoretical rate-distortion bound.

IX. Conclusion

In this paper, we proposed a near optimal coding method for the scalar approach of RIE on i.i.d. gray-scale signals. First, we show that the pmfs for host signal and stego-signal can be separately represented by the intervals within 0 and 1, and the overlapping intervals represent the joint pmfs. We formulate the pmf of stego-signal achieving the optimal rate-distortion performance. Second, we propose an iterative algorithm to calculate the pmf of stego-signal, and we also prove that the iterative algorithm approaches the unique optimal solution. Third, based on the result of the iterative algorithm, a coding scheme is proposed to embed a message in the i.i.d. host sequence. The experiments show that the proposed coding scheme closely achieves the expected rate-distortion performance, and the coding scheme can be applied to gray-scale images.

References

[1] 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. IEEE*, vol. 87, no. 7, pp. 1267–1276, Jul. 1999.

[2] F. Bartolini, A. Tefas, M. Barni, and I. Pitas, "Image authentication techniques for surveillance applications," *Proc. IEEE*, vol. 89, no. 10, pp. 1403–1418, Oct. 2001.

[3] F. A. P. Petitcolas, R. J. Anderson, and M. G. Kuhn, "Information hiding: A survey," *Proc. IEEE*, vol. 87, no. 7, pp. 1062–1078, Jul. 1999.

[4] P. Moulin and J. A. O'Sullivan, "Information-theoretic analysis of information hiding," *IEEE Trans. Inform. Theory*, vol. 49, no. 3, pp. 563–593, Mar. 2003.

[5] J. J. Harmsen and W. A. Pearlman, "Capacity of steganographic channels," *IEEE Trans. Inform. Theory*, vol. 55, no. 4, pp. 1775–1792, Apr. 2009.

[6] Y. Wang and P. Moulin, "Perfectly secure steganography: Capacity, error exponents, and code constructions," *IEEE Trans. Inform. Theory*, vol. 54, no. 6, pp. 2706–2722, Jun. 2008.

[7] F. M. J. Willems and M. V. Dijk, "Capacity and codes for embedding information in gray-scale signals," *IEEE Trans. Inform. Theory*, vol. 51, no. 3, pp. 1209–1214, Mar. 2005.

[8] C. W. Honsinger, P. Jones, M. Rabhani, and J. C. Stoffel, "Lossless recovery of an original image containing embedded data," U.S. Patent, US6278791, 1999.

[9] J. Fridrich, M. Goljan, and R. Du, "Lossless data embedding for all image formats," *Proc. SPIE, Security, Steganography and Watermarking of Multimedia Contents Conf.*, vol. 4675, pp. 572–583, 2002.

[10] T. Kalker and F. M. J. Willems, "Capacity bounds and constructions for reversible data-hiding," in *Proc. Int. Conf. Digital Signal Processing*, 2002, pp. 71–76.

[11] F. M. J. Willems and T. Kalker, "Reversible embedding methods," in *Proc. 40th Annu. Allerton Conf. Communication, Control, and Computing*, Oct. 2002, pp. 1463–1472.

[12] T. Kalker and F. M. J. Willems, "Capacity bounds and code constructions for reversible data-hiding," *Proc. Electronic Imaging. Proc. IS&T/SPIE Security and Watermarking of Multimedia Contents Conf.*, vol. 5020, pp. 604–611, 2003.

[13] F. M. J. Willems and T. Kalker, "Coding theorems for reversible embedding," *DIMACS Ser. Discrete Mathematics and Theoretical Computer Science*, 2004.

[14] M. Haroutunian, S. Tonoyan, O. Koval, and S. Voloshynovskiy, "On reversible information hiding system," in *Proc. IEEE Int. Symp. Information Theory*, Jul. 2008, pp. 940–944.

[15] S. Voloshynovskiy, O. Koval, E. Topak, J. Vila-Forcén, and P. Comesana-Alfaro, Partially reversible data hiding with pure message communications over state-dependent channels (2009) [Online]. Available: http://cvml.unige.ch/publications/postscript /2009/2009.reversibility_SP.pdf

[16] Y. Steinberg, "Coding for channels with rate-limited side information at the decoder, with applications," *IEEE Trans. Inform. Theory*, vol. 54, no. 9, pp. 4283–4295, Sep. 2008.

[17] S. P. Kotagiri and J. N. Laneman, "Variations on information embedding in multiple access and broadcast channels," *IEEE Trans. Inform. Theory*, vol. 56, no. 5, pp. 2225–2240, May 2010.

[18] Z. Ni, Y. Q. Shi, N. Ansari, and W. Su, "Reversible data hiding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 3, pp. 354–362, Mar. 2006.

[19] A. M. Alattar, "Reversible watermark using the difference expansion of a generalized integer transform," *IEEE Trans. Image Processing*, vol. 13, no. 8, pp. 1147–1156, Aug. 2004.

[20] M. U. Celik, G. Sharma, A. M. Tekalp, and E. Saber, "Lossless generalized-LSB data embedding," *IEEE Trans. Image Process.*, vol. 14, no. 2, pp. 253–266, Feb. 2005.
[21] D. M. Thodi and J. J. Rodriguez, "Expansion embedding techniques for reversible watermarking," *IEEE Trans. Image Process.*, vol. 16, no. 3, pp. 721–730, Mar. 2007.
[22] W. Zhang, B. Chen, and N. Yu, "Capacity-approaching codes for reversible data hiding," *Lecture Notes in Computer Science*, vol. 6958/2011, pp. 255–269, 2011.
[23] M. A. Khamsi and W. A. Kirk, *An Introduction to Metric Spaces and Fixed Point Theory*. New York: Wiley, 2001.
[24] K. Sayood, "Arithmetic coding," in *Introduction to Data Compression*, 2nd ed. San Francisco, CA: Morgan Kaufmann, 2000, ch. 4.
[25] J. J. Rissanen and G. G. Langdon Jr., "Arithmetic coding," *IBM J. Res. Development*, vol. 23, no. 2, pp. 149–162, Mar. 1979.
[26] F. Willems, D. Maas, and T. Kalker, "Semantic lossless source coding," in *Proc. 42nd Annu. Allerton Conf. Communication, Control and Computing*, 2004.

Sian-Jheng Lin was born in Taiwan. He received the B.S., M.S., and Ph.D. degrees in computer science from National Chiao Tung University, in 2004, 2006, and 2010, respectively.

He is currently a Postdoctoral Fellow with the Research Center for Information Technology Innovation, Academia Sinica, Taiwan. His recent research interests include data hiding, error control coding, and secret sharing.

Wei-Ho Chung (M'11) was born in Kaohsiung, Taiwan, in 1978. He received the B.Sc. and M.Sc. degrees in electrical engineering from National Taiwan University, Taipei City, Taiwan, in 2000 and 2002, respectively. From 2005 to 2009, he was with the Electrical Engineering Department, University of California, Los Angeles, where he obtained the Ph.D. degree.

From 2000 to 2002, he worked on routing protocols in the mobile ad hoc networks in the M.Sc. program in National Taiwan University. From 2002 to 2005, he was a System Engineer at ChungHwa Telecommunications Company, where he worked on data networks. In 2008, he was an Research Intern working on CDMA systems in Qualcomm, Inc. From 2007 to 2009, he was a Teaching Assistant at UCLA. From June to December 2009, he worked as a Research Associate in San Diego, CA, on wireless communications for multimedia communications and unequal error protection for video transmission. He has been an Assistant Research Fellow at the Research Center for Information Technology Innovation, Academia Sinica, Taiwan, since January 2010. His research interests include communications, signal processing, and networks.

Dr. Chung received the Taiwan Merit Scholarship from 2005 to 2009, and the Best Paper Award in IEEE WCNC 2012.