Fast Estimation of Optimal Marked-Signal Distribution for Reversible Data Hiding

Xiaocheng Hu, Weiming Zhang, Xuexian Hu, Nenghai Yu, Xianfeng Zhao, and Fenghua Li

***Abstract*—Recently, code construction approaching the rate-distortion bound of reversible data hiding has been proposed by Lin *et al.*, in which the coding/decoding process needs the optimal probability distribution of marked-signals as parameters. Therefore, the efficiency and accuracy of estimating the optimal marked-signal distribution will greatly influence the speeds of encoding and decoding. In this paper, we propose a fast algorithm to solve the optimal marked-signal distribution. Furthermore, we modify the method to achieve the optimal distribution directly according to a given distortion constraint or an expected embedding rate, which makes it more practical for applications.**

***Index Terms*—Convex optimization, distortion, embedding rate, Lagrange duality, reversible data hiding.**

I. Introduction

As a technique that embeds messages into cover signals, information hiding has been widely applied in areas such as covert communication, copyright protection and media annotation. Reversible data hiding (RDH) is one kind of information hiding technique with the characteristic that not only the message needs to be precisely extracted, but also the cover itself should be restored losslessly. This property is important in some special scenarios such as medical imagery [1], military imagery and law forensics. In these applications, the cover is too precious or too important to be damaged [2]. Moreover, it has been found recently that reversible data hiding can be quite helpful in video error-concealment coding [3].

A plenty of reversible data hiding algorithms have been proposed in the past decade. Classical RDH algorithms roughly fall into three categories. The first class of algorithms follows the idea of compression-embedding framework, which was first introduced by Fridrich [4]. In these algorithms, a two-value feature is calculated for each pixel group, the sequence is compressible and messages can be embedded in the extra space left by lossless compression. The second class of methods is based on difference expansion (DE) [5]–[7], in which the differences of each pixel groups are expanded, e.g., multiplied by 2, and thus the least significant bits (LSBs) of the differences are all-zeros and can be used for embedding message. The last RDH algorithms are based on histogram shift (HS) [8]. The histogram of one special feature (for example, gray-scale value) for nature image is quite uneven, which implies that the histogram can be modified for embedding data. For instance, some space can be saved for watermarks by shifting the bins of histogram. In fact, better performance can be achieved by applying DE or HS to the residual part of images, e.g., the predicted errors (PE) [9]–[12].

Almost all recent RDH methods include two steps. The first step generates a host sequence with small entropy, i.e., the host has a steeper histogram, which usually can be realized by predicted errors (PE). The second step reversibly embeds messages into the host sequence by modifying its histogram with DE or HS. These are for the sake of maximizing the payload for a given distortion constraint or minimizing the overall distortion for a given payload.

One natural problem is what is the upper bound of the payload for a given host sequence and a distortion constraint. For independent and identically distributed (i.i.d.) host sequence, this problem has been solved by Kalker and Willems [13], who formulated the RDH as a special rate-distortion problem, and obtained the rate-distortion function, i.e., the upper bound of the embedding rate under a given distortion constraint Δ, as follows:

$$\rho_{rev}(\Delta) = maximize\{H(Y)\} - H(X) \tag{1}$$

where X and Y denote the random variables of host signal and marked-signal respectively. The maximum entropy is over all transition probability matrices $P_{Y|X}(y|x)$ satisfying the distortion constraint $\sum_{x,y} P_X(x)P_{Y|X}(y|x)D(x,y) \leq \Delta$. The distortion metric $D(x,y)$ is usually defined as the square error distortion, $D(x,y) = (x-y)^2$.

In fact, the optimal solution $P_{Y|X}(y|x)$ for (1) implies the optimal modification manner on the histogram of the host signal X. However, how to efficiently realize the optimal modification is still a problem. For binary host sequence, i.e., $x \in \{0,1\}$, Zhang *et al.* [14], [15], proposed a code construction approaching the rate-distortion bound (1). Recently, Lin *et al.* [16] proposed a code construction that can approach the rate-distortion bound for gray-scale host, i.e., $x \in \{0,1,\ldots,B-1\}$. The coding and decoding processes of the methods in [15],[16] need the optimal transition probability $P_{Y|X}(y|x)$ as parameters. In other words, one should first solve the optimization problem (1) before coding and decoding.

Manuscript received September 12, 2012; revised January 06, 2013 and March 14, 2013; accepted March 22, 2013. Date of publication April 03, 2013; date of current version April 15, 2013. This work was supported in part by the Natural Science Foundation of China under Grant 61170234 and Grant 60803155, and in part by the Strategic Priority Research Program of the Chinese Academy of Sciences under Grant XDA06030601. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Chiou-Ting Hsu. *(Corresponding author: W. Zhang.)*

X. Hu, W. Zhang, and N. Yu are with the School of Information Science and Technology, University of Science and Technology of China, Hefei, 230026, China (e-mail: hxc@mail.ustc.edu.cn; weimingzhang@yahoo.cn; ynh@ustc.edu.cn).

X. Hu is with the Department of Information Research, Zhengzhou Information Science and Technology Institute, Zhengzhou 450002, China, and also is with the Institute of Software, Chinese Academy of Sciences, Beijing, 100196, China (e-mail: xuexian_hu@yahoo.com.cn).

X. Zhao and F. Li are with the State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing, 100093, China (e-mail: zhaoxianfeng@iie.ac.cn; lifenghua@iie.ac.cn).

Digital Object Identifier 10.1109/TIFS.2013.2256131

Therefore, the efficiency and accuracy of solving problem (1) directly influence the speeds of coding and decoding. Although many optimization algorithms can be used to solve problem (1), it will cost long time for large B, because the matrix $P_{Y|X}(y|x)$ to optimize is of size $B \times B$.

Fortunately, in [16], Lin *et al.* found that the joint distribution $P_{X,Y}(x,y)$ can be expressed by the marginal distributions $P_X(x)$ and $P_Y(y)$. This means the coder and decoder only need to calculate the optimal marginal distribution $P_Y(y)$ instead of the conditional distribution $P_{Y|X}(y|x)$. Lin *et al.* [16] proposed a backward and forward iterative (BFI) algorithm to estimate $P_Y(y)$, in which the rate-distortion pair is controlled by a parameter $\alpha \in [1.0, +\infty)$. In practice, the coder needs the optimal distribution for a given distortion constraint or embedding rate. However, in BFI algorithm, no apparent translation between the parameter α and the distortion constraint Δ can be revealed. So for a given distortion constraint, we have to use binary search to repeat the BFI algorithm to estimate $P_Y(y)$, which is much less efficient.

In this paper, we propose a novel algorithm to estimate the optimal marginal distribution $P_Y(y)$ that is significantly faster than the BFI algorithm [16]. Furthermore, our algorithm can be easily modified to compute the optimal distribution $P_Y(y)$ directly according to a given distortion constraint or an expected embedding rate, which makes it more practical for applications. Our algorithm exhibits fast convergence rate based on Lagrange duality and logarithmic barrier method, and it scales well for large cardinality B and variant cover signal sources.

In the rest of the paper, we will first sketch the previous work in Section II. Then we reveal the Lagrange dual problems of the original problem (1) and its inverse problem in Section III. Next in Section IV, we propose a path-following algorithm to estimate the optimal marked-signal distribution and compare it with the BFI method. Modified algorithms in Section V demonstrate our algorithm's efficiency and scalability for practical scenarios. In Section VII, we combine our algorithm to achieve the rate-distortion bound for reversible data hiding applications. We leave some discussions and implementation details in Section VI and brief conclusions are drawn in Section VIII.

II. Previous Work

A. Problem Formulation

In RDH, L bits of message $\mathbf{m} = (m_1, \ldots, m_L)$ are embedded by the sender into the cover sequence $\mathbf{x} = (x_1, \ldots, x_n) \in \mathcal{X} \times \mathcal{X} \times \cdots \times \mathcal{X} = \mathcal{X}^n$, through slightly modifying its elements to produce the marked-sequence $\mathbf{y} = (y_1, \ldots, y_n) \in \mathcal{Y}^n$. The two value sets are $\mathcal{X} = \{0, 1, \ldots, M-1\}$ and $\mathcal{Y} = \{0, 1, \ldots, N-1\}$. We denote the embedding rate by $R = L/n$. Schemes are usually constructed to minimize some distortion measure $D(x,y)$ between $\mathbf{x}$ and $\mathbf{y}$ for a given embedding rate R. The distortion metric $D(x,y)$ in this paper is by default defined as the square error distortion, $D(x,y) = (x-y)^2$. Note that it's not limited to the square error distortion metric. Lin *et al.* [16] also discussed other distortion metrics, such as L_1-Norm $D_1(x,y) = |x-y|$. We assume the cover $\mathbf{x} = (x_1, \ldots, x_n)$ is an n-tuple composed of n i.i.d. samples drawn from the probability distribution $P_X = \{P_X(x), x \in \mathcal{X}\}$. For example, $x = (1,2,1,0,2,1,0)$ is a 7-tuple, where $\mathcal{X} = \{0,1,2\}$.

Next, we discuss two application scenarios according to the limited conditions on the sender.

- **Distortion-limited sender (DLS)** maximize the embedding rate subject to an average distortion constraint D_{av}. According to (1), $H(X)$ is fixed, and thus the problem can be formulated as follows[1]:

$$\begin{aligned} \text{maximize} \quad & -\sum_{y=0}^{N-1} P_Y(y)\ln(P_Y(y)) \\ \text{subject to} \quad & \sum_{y=0}^{N-1}\sum_{x=0}^{M-1} P_X(x)P_{Y|X}(y|x)D(x,y) \le D_{av} \\ & \sum_{x=0}^{M-1} P_X(x)P_{Y|X}(y|x) = P_Y(y), \forall y \\ & \sum_{y=0}^{N-1} P_{Y|X}(y|x) = 1, \forall x \\ & P_{Y|X}(y|x) \ge 0, \forall x, y \end{aligned} \tag{2}$$

  where the variables are the transition probability matrix $P_{Y|X}(y|x)$, the constant parameters are the source distribution $P_X(x)$, the distortion measure matrix $D(x,y)$, and the distortion constraint D_{av}.
- **Payload-limited sender (PLS)** minimize the average distortion for a given embedding rate $H_Y - H(X)$, which is formulated as:

$$\begin{aligned} \text{minimize} \quad & \sum_{y=0}^{N-1}\sum_{x=0}^{M-1} P_X(x)P_{Y|X}(y|x)D(x,y) \\ \text{subject to} \quad & -\sum_{y=0}^{N-1} P_Y(y)\ln(P_Y(y)) \ge H_Y \\ & \sum_{x=0}^{M-1} P_X(x)P_{Y|X}(y|x) = P_Y(y), \forall y \\ & \sum_{y=0}^{N-1} P_{Y|X}(y|x) = 1, \forall x \\ & P_{Y|X}(y|x) \ge 0, \forall x, y \end{aligned} \tag{3}$$

  where the variables are the transition probability matrix $P_{Y|X}(y|x)$, the constant parameters are the source distribution $P_X(x)$, the distortion measures $D(x,y)$, and the entropy constraint H_Y.

Actually the two problems above are dual to each other, meaning that the optimal distribution for the first problem is, for some value of H_Y, also optimal for the second one. Due to their convexity and smoothness, many convex optimization algorithms can be used to solve them and are guaranteed to find the global optimal solution, like gradient projection[2] or interior-point methods [18],[21]. However, gradient based method exhibits slow convergence, and although interior-point method takes very few iterations to converge, they have difficulty in handling problems with large scale. The complexity

[1]Throughout this paper, we use natural logarithms for defining entropy.

[2]Available: http://en.wikipedia.org/wiki/Gradient descent.

of computing the step direction each iteration for interior-point method is $O((M \times N)^3)$, where $M \times N$ is the size of the transition probability matrix $P_{Y|X}(y|x)$. Therefore, solving the two primal problems above directly seems too limited for real applications.

It has been proved both in [16] and [17] that the optimal transition probability matrix of these two problems has a Noncrossing-edges Property. To be specific, given an optimal $P^*_{Y|X}$, for any two distinct possible transition events $P^*_{Y|X}(y_1|x_1) > 0$ and $P^*_{Y|X}(y_2|x_2) > 0$, if $x_1 < x_2$, then $y_1 \leq y_2$ holds. Lin *et al.* [16] pointed out that, by using this Noncrossing-edges property, the joint distribution of X and Y can be expressed as (4). $P_{CX}(x)$ and $P_{CY}(y)$ are cumulative probability distributions of X and Y defined by $P_{CX}(x) = \sum_{i=0}^{x} P_X(i), x = 0, \ldots, M-1$, and $P_{CY}(y) = \sum_{i=0}^{y} P_Y(i), y = 0, \ldots, N-1$. It is noted that $P_{CX}(-1) = P_{CY}(-1) = 0$ and $P_{CX}(M-1) = P_{CY}(N-1) = 1$.

$$P_{X,Y}(x,y) = \max\{0, \min\{P_{CX}(x), P_{CY}(y)\} - \max\{P_{CX}(x-1), P_{CY}(y-1)\}\}. \quad (4)$$

So the problems (2) and (3) can both be simplified to find the optimal marginal distribution of the marked-signal $P_Y(y)$ instead of the probability transition matrix $P_{Y|X}(y|x)$.

B. The Iterative Algorithm to Calculate $p_y(y)$

In [16], Lin *et al.* proposed a backward and forward iterative (BFI) algorithm to estimate the optimal distribution of the marked-signal $P_Y(y)$. The BFI algorithm assumes that the cover and marked-signal value sets, $\mathcal{X}$ and $\mathcal{Y}$, are equal, i.e., $M = N = B$.

Algorithm 1 The iterative algorithm to estimate the optimal marked-signal distribution $P_Y(y)$ (**BFI Algorithm [16]**.

Input: the cumulative probability distribution P_{CX}, a real number $\alpha \geq 1$, and the tolerance ϵ.
Output: the cumulative probability distribution $P_{CY}(y)$.

1. Given initial set $x = \{x_y = (y+1)/B,\ y = -1\ to\ B-1\}$. Or the user can design an arbitrary initialization as long as $0 = x_{-1} \leq x_0 \leq \cdots \leq x_{B-1} = 1$. Declare the variable $var = 0$.
2. For y form 0 to $B-2$, update each x_y through

$$x_y^{new} = \begin{cases} \frac{x_{y+1} - x_{y-1}}{1+\alpha^{D(s,y)-D(s,y+1)}} + x_{y-1}, & \text{if there exists } x_y \in (P_{CX}(s-1), P_{CX}(s)); \\ P_{CX}(s), & \text{if there exists } \alpha^{D(s,y)-D(s,y+1)} \leq \frac{x_{y+1}-P_{CX}(s)}{P_{CX}(s)-x_{y-1}} \leq \alpha^{D(s+1,y)-D(s+1,y+1)}. \end{cases}$$

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

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

3. For y from $B-2$ to 0, update each x_y and the maximal offset var through the same criterion in step 2.
4. If $var \geq \epsilon$, set $var = 0$ and then go to step 2; otherwise, output $P_{CY} = \{P_{CY}(y) = x_y\}$.

The BFI method controls an input parameter α for various rate-distortion pairs. The case $\alpha = 1$ admits the uniform distribution $P^*_Y = \{P^*_Y(y) = 1/B\}$ performing the maximal embedding rate, and another case $\alpha \to +\infty$ admits $P^*_Y = \{P^*_Y(y) = P_X(y)\}$ generating an unchanged marked-sequence with zero embedding rate. Details can be found in [16].

In most situations while the host sequence possesses a quite general probability distribution, no apparent translation between the parameter α and the average distortion D_{av} can be revealed. So in order to compute the optimal marked-signal distribution for a given D_{av}, we need to repeat the BFI algorithm by binary search to find the optimal α in return, which makes it less practical.

III. The Lagrange Dual Problems

In constrained optimization, the Lagrange duality [19] is largely used to convert the primal problem to its dual one. It is showed that if the primal problem is convex, the strong duality property holds, meaning that the optimal objective function values and corresponding optimal solutions for both problems are equivalent.

Proposition 1: The Lagrange dual of the DLS problem (2) for an average distortion constraint D_{av} is of the following form:

$$\begin{aligned} \text{minimize} \quad & \sum_{j=0}^{N-1} e^{v_j - 1} + \sum_{i=0}^{M-1} u_i + \gamma D_{av} \\ \text{subject to} \quad & \gamma P_X(i) D(i,j) + u_i + P_X(i) v_j \geq 0, \forall i, j \\ & \gamma \geq 0 \end{aligned} \quad (5)$$

where the variables are $\mathbf{v} = (v_0, \ldots, v_{N-1}) \in \mathbb{R}^N$, $\mathbf{u} = (u_0, \ldots, u_{M-1}) \in \mathbb{R}^M$, and $\gamma \in \mathbb{R}$. And the constant parameters are $P_X(i)$, $D(i,j)$, and D_{av}. The optimal solution of this dual problem v_j yields the optimal marked-signal distribution of problem (2) with the following form:

$$P_Y(j) = e^{v_j - 1}, j = 0, \ldots, N-1\ . \quad (6)$$

Proof: The basic idea of Lagrangian duality is to take the constraints in (2) into account by augmenting the objective function with a weighted sum of the constraint functions. The Lagrangian function associated with problem(2) is:

$$\begin{aligned} &L(P_{Y|X}, P_Y, \gamma, \mathbf{v}, \mathbf{u}, \mathbf{\Lambda}) \\ &\quad = \sum_j P_Y(j) \ln(P_Y(j)) - \sum_{i,j} \lambda_{i,j} P_{Y|X}(j|i) \\ &\quad + \gamma \sum_{i,j} P_X(i) P_{Y|X}(j|i) D(i,j) - \gamma D_{av} \\ &\quad + \sum_j v_j \Big(\sum_i P_X(i) P_{Y|X}(j|i) - P_Y(j)\Big) \\ &\quad + \sum_i u_i \Big(\sum_j P_{Y|X}(j|i)\Big) - \sum_i u_i \end{aligned}$$

with Lagrange multipliers $\gamma \in \mathbb{R}$, $\mathbf{v} = (v_0, \ldots, v_{N-1}) \in \mathbb{R}^N$, $\mathbf{u} = (u_0, \ldots, u_{M-1}) \in \mathbb{R}^M$, and $\Lambda = \{\lambda_{i,j} | \forall i, j\} \in \mathbb{R}^{M \times N}$. Since γ and $\lambda_{i,j}$ correspond to inequality constraints, we have $\gamma \geq 0$ and $\lambda_{i,j} \geq 0$, $\forall i, j$. We get the Lagrange dual function $g(\gamma, \mathbf{v}, \mathbf{u}, \mathbf{\Lambda}) = \inf_{P_{Y|X}, P_Y} L(P_{Y|X}, P_Y, \gamma, \mathbf{v}, \mathbf{u}, \mathbf{\Lambda})$ by finding the $P_{Y|X}$ and P_Y that minimize L, which is a

convex function of $P_{Y|X}$ and P_Y. First we note that L is a linear function of $P_{Y|X}$, thus it's unbounded below unless $\gamma P_X(i)D(i,j)+u_i+P_X(i)v_j-\lambda_{i,j}=0, \forall i,j$. This is equivalent to $\gamma P_X(i)D(i,j)+u_i+P_X(i)v_j \geq 0$ since $\lambda_{i,j} \geq 0$. When the condition $\gamma P_X(i)D(i,j)+u_i+P_X(i)v_j \geq 0, \forall i,j$ holds, the Lagrangian becomes $\sum_j P_Y(j)\ln(P_Y(j))-\sum_j v_j P_Y(j)-\sum_i u_i - \gamma D_{av}$, which means now we should minimize over P_Y. To find the minimum of $P_Y(j)\ln(P_Y(j))-v_jP_Y(j)$ over $P_Y(j)$, we let the derivative with respect to $P_Y(j)$ equal zero, i.e., $\ln(P_Y(j))+1-v_j=0$. Thus we get $P_Y(j)=e^{v_j-1}$ with the associated minimum value $-e^{v_j-1}$. So the Lagrange dual function is given by (7) at the bottom of the page.

Strictly speaking, the Lagrange dual of the primal problem (2) is to maximize the dual function $g(\gamma,\mathbf{v},\mathbf{u})$ subject to $\gamma \geq 0$. By making the inequality constraints explicit, we get:

$$\begin{aligned} \text{maximize} \quad & -\sum_{j=0}^{N-1} e^{v_j-1} - \sum_{i=0}^{M-1} u_i - \gamma D_{av} \\ \text{subject to} \quad & \gamma P_X(i)D(i,j)+u_i+P_X(i)v_j \geq 0, \forall i,j \\ & \gamma \geq 0 \end{aligned}$$

which is equivalent to (5). ∎

Corollary 1: The Lagrange dual problem of the inverse PLS problem (3) for an expected embedding rate H_Y is of the following form:

$$\begin{aligned} \text{minimize} \quad & \gamma \sum_{j=0}^{N-1} e^{v_j/\gamma-1} + \sum_{i=0}^{M-1} u_i - \gamma H_Y \\ \text{subject to} \quad & P_X(i)D(i,j)+u_i+P_X(i)v_j \geq 0, \forall i,j \\ & \gamma \geq 0 \end{aligned} \tag{8}$$

where the variables are $\mathbf{v} \in \mathbb{R}^N$, $\mathbf{u} \in \mathbb{R}^M$, and $\gamma \in \mathbb{R}$. And the constant parameters are $P_X(i)$, $D(i,j)$, and H_Y. The optimal solution of this dual problem v_j and γ yields the optimal marked-signal distribution of problem (3) with the following form:

$$P_Y(j) = e^{v_j/\gamma-1}, j=0,\ldots,N-1. \tag{9}$$

Proof: The derivation is nearly the same as Proposition 1. ∎

IV. Path Following Algorithm to Calculate $p_y(y)$

A. Path-Following Algorithm

In the derived dual problem (5), the parameter $\gamma \in [0,+\infty]$ is dual associated with the distortion constraint D_{av}. The case $\gamma = 0$ admits the uniform distribution $P_Y^* = \{P_Y^*(y) = 1/N\}$ corresponding to the maximal embedding rate, and another case $\gamma \to +\infty$ admits $P_Y^* = \{P_Y^*(y) = P_X(y)\}$ corresponding to the zero embedding rate. So for a specific input parameter γ, the dual problem (5) can be rewritten as:

$$\begin{aligned} \text{minimize} \quad & \textstyle\sum_{j=0}^{N-1} e^{v_j-1} + \sum_{i=0}^{M-1} u_i \\ \text{subject to} \quad & \gamma P_X(i)D(i,j)+u_i+P_X(i)v_j \geq 0, \forall i,j \end{aligned}. \tag{10}$$

The corresponding optimal marked-signal distribution $P_Y(y)$ is given by (6).

By observing (10), the objective function is smooth and convex, but it contains too many constraints. From the basic idea of the barrier method [21], we can eliminate the inequality constraints by adding logarithmic penalty on them, and thus problem (10) is approximated as an unconstrained problem:

$$\begin{aligned} \text{minimize} \quad & f_t(\mathbf{w}) = f_t(\mathbf{v},\mathbf{u}) \\ & = t\Big(\sum_{j=0}^{N-1} e^{v_j-1} + \sum_{i=0}^{M-1} u_i\Big) \\ & -\sum_{i,j} \ln(\gamma P_X(i)D(i,j)+u_i+P_X(i)v_j) \end{aligned} \tag{11}$$

where t is a penalty parameter, and when $t \to \infty$, the approximating error tends to zeros. The variables v_j and u_i are concatenated into a row vector $\mathbf{w} = (\mathbf{v},\mathbf{u}) = (v_0,\ldots,v_{N-1},u_0,\ldots,u_{M-1})$ conveniently.

The path-following method (barrier method) is based on solving a sequence of unconstrained minimization problems (11) for increasing values of t, using the last point found as the starting point for the next unconstrained minimization problem. The path-following algorithm to solve (10) is outlined as Algorithm 2.

Algorithm 2 The path-following algorithm to estimate the optimal marked-signal distribution $P_Y(y)$**(PF Algorithm)**

Input: The cover signal distribution $P_X(i)$, a real number $\gamma \geq 0$, and the tolerance value $\epsilon > 0$.
Output: The estimated marked-signal distribution $P_Y(j)$.

1. Initialize with any feasible points $\mathbf{v}^0 = \{v_0,\ldots,v_{N-1}\}$ and $\mathbf{u}^0 = \{u_0,\ldots,u_{M-1}\}$ which satisfy $\gamma P_X(i)D(i,j)+u_i+P_X(i)v_j > 0$, and $t^0 \geq 1$, $\mu > 1$.
2. **While** ($(NM)/t \geq \epsilon$)
   1) *Centering step:* Minimizing **(11)** with initial point $(\mathbf{v},\mathbf{u})$, and denote the computed solution by $(\mathbf{v}^t,\mathbf{u}^t)$.
   2) *Update:* $\mathbf{v} = \mathbf{v}^t$, $\mathbf{u} = \mathbf{u}^t$.
   3) *Increase t:* $t = \mu t$.

   end While
3. Output: $P_Y(j) = e^{v_j-1}, j=0,\ldots,N-1$.

$$g(\gamma,\mathbf{v},\mathbf{u}) = \begin{cases} -\sum_{j=0}^{N-1} e^{v_j-1} - \sum_{i=0}^{M-1} u_i - \gamma D_{av}, & \gamma P_X(i)D(i,j)+u_i+P_X(i)v_j \geq 0 \\ -\infty, & \text{otherwise.} \end{cases} \tag{7}$$

TABLE I
COMPUTATIONAL TIME COMPARISON BETWEEN BFI AND PF OVER DIFFERENT HOST SIGNAL DISTRIBUTIONS AND PROBLEM SCALES

B	Cover pmf	Algorithm	#Iteration	Time(s)
16	Laplace	BFI	83	0.0101
	$\mu = 7.5, b = 1$	PF	57	0.0103
	Laplace	BFI	80	0.0097
	$\mu = 7.5, b = 4$	PF	53	0.0099
32	Laplace	BFI	203	0.0510
	$\mu = 15.5, b = 2$	PF	65	0.0182
	Laplace	BFI	193	0.0509
	$\mu = 15.5, b = 4$	PF	63	0.0171
64	Laplace	BFI	802	0.4051
	$\mu = 31.5, b = 2$	PF	78	0.0472
	Laplace	BFI	748	0.3961
	$\mu = 31.5, b = 8$	PF	66	0.0407
128	Laplace	BFI	2315	2.2854
	$\mu = 63.5, b = 2$	PF	84	0.1531
	Laplace	BFI	2542	2.6774
	$\mu = 63.5, b = 16$	PF	68	0.1225
256	Laplace	BFI	6893	13.9276
	$\mu = 127.5, b = 2$	PF	90	0.8055
	Laplace	BFI	7166	15.3584
	$\mu = 127.5, b = 16$	PF	82	0.7280

Algorithm 3 Newton's method to solve **(11)**

Input: Initial point $\mathbf{w} = (\mathbf{v}, \mathbf{u})$.
Output: Computed point $\mathbf{w}^t = (\mathbf{v}^t, \mathbf{u}^t)$ minimizing $f_t(\mathbf{w})$.

1. given Newton's method tolerance $\epsilon_2 > 0$.
2. **While not converged**
   1) Form gradient vector $\mathbf{g} = \nabla f_t(\mathbf{w}) \in \mathbb{R}^{N+M}$ and Hessian matrix $\mathbf{H} = \nabla^2 f_t(\mathbf{w}) \in \mathbb{R}^{(N+M)\times(N+M)}$.
   2) Compute step direction $\Delta\mathbf{w} \in \mathbb{R}^{N+M}$ and decrement value $\delta \in \mathbb{R}$ by: Note here the superscript T stands for vector or matrix transpose, g^T turns a row vector into a column vector.

$$\Delta\mathbf{w} = (-\mathbf{H}^{-1}\mathbf{g}^T)^T; \quad \delta = -\mathbf{g}(\Delta\mathbf{w})^T.$$

   3) Stop criterion: **quit** if $\delta < \epsilon_2$.
   4) Choose step size s by backtracking line search.
   5) Update: $\mathbf{w} = \mathbf{w} + s\Delta\mathbf{w}$.
   End While
3. Output: $\mathbf{w}^t = (\mathbf{v}^t, \mathbf{u}^t)$.

In PF Algorithm[3], at each *Centering step* , we compute the central point $(\mathbf{v}^t, \mathbf{u}^t)$ starting from the previously computed central point, and then increase t by a factor $u > 1$. The initial point $\mathbf{v}^0$ and $\mathbf{u}^0$ is arbitrarily chosen as long as the condition $\gamma P_X(i)D(i,j) + u_i^0 + P_X(i)v_j^0 > 0$ holds. From (6) we initialize with $v_j^0 = \ln(1/N) + 1$ and $u_i^0 = |\ln(1/N) + 1| + 0.01$ to let $P_Y(j) = 1/N$, corresponding to the uniform distribution. For the penalty term, the general case $t^0 = 1$ is good. Faster convergence can be gained by choosing t^0 to minimize the norm of gradient $\|\nabla f_t(\mathbf{w}^0)\|_2^2$, as explained in Section VI. Parameter μ in the range $[10, 100]$ is a good choice and our experiment shows that $\mu = 20$ performs well. Because the approaching accuracy increases slightly when t gets larger, we terminate the outer iteration with $\epsilon = 10^{-6}$.

[3]Note here the superscript T stands for vector or matrix transpose, $\mathbf{g}^T$ turns a row vector into a column vector.

The Newton's method [20] solving the *Centering step* is described by Algorithm 3. Newton's method is a standard optimization framework for unconstrained minimization problems which is famous for its fast convergence rate. In each iteration it first forms the gradient vector $\mathbf{g}$ and the Hessian matrix $\mathbf{H}$, and according to these two terms, the step direction $\Delta\mathbf{w}$ is computed. Next along this step direction $\Delta\mathbf{w}$, a step size s is chosen to finally update the variable vector $\mathbf{w}$ by $\mathbf{w} = \mathbf{w} + s\Delta\mathbf{w}$. The stop criterion for Newton's method is set with tolerance $\epsilon_2 = 10^{-6}$.

Algorithm 4 Backtracking line search method to calculate the step size s

Input: a step direction $\Delta\mathbf{w}$ for $f_t(\mathbf{w})$ at $\mathbf{w} \in \mathbf{dom} f_t(\mathbf{w})$, $\alpha \in (0, 0.5), \beta \in (0, 1), s = 1$.
Output: the step size s.

1. Ensure the domain of the function $f_t(\mathbf{w})$:

$$\mathbf{while}\ (\mathbf{w} + s\Delta\mathbf{w}) \notin \mathbf{dom} f_t(\mathbf{w}),\ s = \beta s.$$

2. Decrease the function value sufficiently:

$$\mathbf{while}\ f_t(\mathbf{w}+s\Delta\mathbf{w}) > f_t(\mathbf{w}) + \alpha s(\nabla f_t(\mathbf{w}))(\Delta\mathbf{w})^T,\ s = \beta s.$$

3. Output: s.

To decide how far we update the variable vector $\mathbf{w}$ along the step direction $\Delta\mathbf{w}$, we use the Backtracking line search method [20] to estimate the step size s, which is described by Algorithm 4. The parameter β is a reduce factor to update s, and the parameter α measures the decrease degree for the function $f_t(\mathbf{w})$ we accept. In our experiments, $\alpha = 0.01$, $\beta = 0.6$ are used. We first multiply s by β until $\gamma P_X(i)D(i,j) + u_i + P_X(i)v_j > 0, \forall i, j$, which ensures the domain of the $\ln$ function. And then we start to check whether the inequality $f_t(\mathbf{w}+s\Delta\mathbf{w}) \le f_t(\mathbf{w}) + \alpha s(\nabla f_t(\mathbf{w}))(\Delta\mathbf{w})^T$ holds. Details of the Newton's method and backtracking line search can be found in [20].

As explained in Section Section VI, the Hessian matrix $\nabla^2 f(\mathbf{w})$ for Newton's step exhibits special structures which enable fast solving the corresponding step direction $\Delta\mathbf{w}$. Due to the fast convergence rate of the Newton's method, our PF algorithm is quite efficient as experiments demonstrate.

B. Simulations

In this section, we compare our proposed PF algorithm with BFI algorithm [16] (offered by Lin *et al.*) using numerical simulations. The host sequences are drawn from discrete Laplace distributions with different scale parameters b. We take the case of $M = N = B$ and test with incremental values for B. We range the input parameters γ and α with 15 evenly spaced values so that the corresponding embedding rate varies from minimal to maximal, and the performances are averaged. All the simulations in this paper are conducted and timed on the same PC with an Intel Core i5 M 520 2.40 GHz CPU that has 4 GB memory, running Windows 7 (64-bit) and Matlab (version R2011b).

Table I shows that our PF algorithm performs much faster than BFI algorithm and almost 15 times faster on average for $B \ge 32$. Our method exhibits a consistent convergent rate as

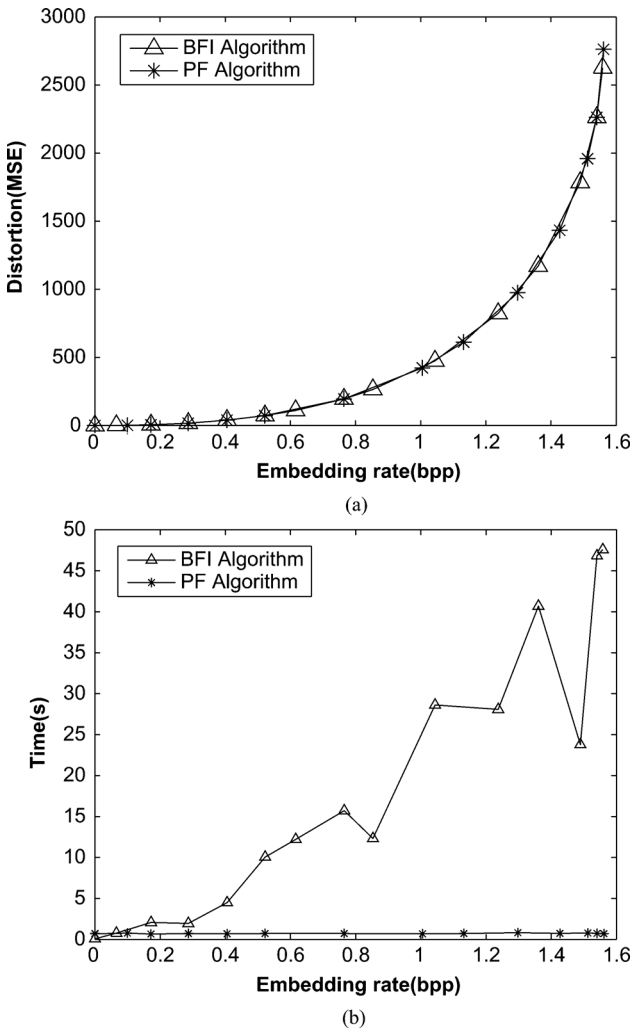


Fig. 1. Performance comparison of PF and BFI algorithms for a Laplace host sequence with $\mu = 127.5, b = 16$, and $B = 256$. (a) Rate-distortion curves. (b) Computational time comparison with respect to variant embedding rates.

the problem size scales, while the main efforts are involved in solving a set of linear equations per iteration as illustrated in Section VI. Relatively, the BFI algorithm enjoys cheap computations per iteration but its convergence is slow when problem scale gets large. For a specified problem scale, the BFI algorithm actually favors low embedding rate and exhibits performance reduction when the embedding rate increases as shown in Fig. 1(b).

Fig. 1(a) depicts the rate-distortion curves of both algorithms for a specific Laplace host signal : $\mu = 127.5, b = 16$ and $B = 256$. It shows that the two curves coincide with each other to the expected rate-distortion bound.

Algorithms For DLS and PLS Problems

C. PF Algorithm for DLS Problem

The PF algorithm stated in Section IV can be modified to solve the DLS problem(2) directly for a given distortion constraint D_{av}. By adding the penalty terms, the dual problem (5) can be rewritten as the following unconstrained optimization problem:

$$\begin{aligned} \text{minimize} \quad f_t(\mathbf{w}) &= f_t(\mathbf{v}, \mathbf{u}, \gamma) \\ &= t\Big(\sum_{j=0}^{N-1} e^{v_j - 1} + \sum_{i=0}^{M-1} u_i + \gamma D_{av}\Big) \\ &\quad - \sum_{i,j} \ln(\gamma P_X(i) D(i,j) + u_i + \\ &\quad P_X(i) v_j) - \ln(\gamma) \end{aligned} \tag{12}$$

where $\mathbf{w} = (\mathbf{v}, \mathbf{u}, \gamma) \in \mathbb{R}^{N+M+1}$ is the optimization variable. The corresponding optimal marked-signal distribution $P_Y(y)$ in this case is given by (6). The PF algorithm for DLS problem is described as Algorithm 5.

Algorithm 5 The path-following algorithm for DLS problem to estimate the optimal marked-signal distribution $P_Y(y)$ **(DLS PF Algorithm).**

Input: The cover signal distribution $P_X(i)$, a distortion constraint $D_{av} \geq 0$, and the tolerance value $\epsilon > 0$.

Output: The estimated marked-signal distribution $P_Y(j)$.

1. Initialize with any feasible points $\mathbf{v}^0 = \{v_0, \ldots, v_{N-1}\}$ and $\mathbf{u}^0 = \{u_0, \ldots, u_{M-1}\}$ which satisfy $\gamma P_X(i) D(i,j) + u_i + P_X(i) v_j > 0$, and $\gamma^0 > 0, t^0 \geq 1, \mu > 1$.
2. **While** $((NM)/t \geq \epsilon)$
   1) *Centering step:* Minimizing **(12)** with initial point $(\mathbf{v}, \mathbf{u}, \gamma)$, and denote the computed solution by $(\mathbf{v}^t, \mathbf{u}^t, \gamma^t)$.
   2) *Update:* $\mathbf{v} = \mathbf{v}^t$, $\mathbf{u} = \mathbf{u}^t$, $\gamma = \gamma^t$.
   3) *Increase t:* $t = \mu t$.
   end while
3. Output: $P_Y(j) = e^{v_j - 1}, j = 0, \ldots, N-1$.

Algorithm 6 The path-following algorithm for PLS problem to estimate the optimal marked-signal distribution $P_Y(y)$ **(PLS PF Algorithm).**

Input: The cover signal distribution $P_X(i)$, an expected embedding rate $H_Y \geq 0$, and the tolerance value $\epsilon > 0$.

Output: The estimated marked-signal distribution $P_Y(j)$.

1. Initialize with any feasible points $\mathbf{v}^0 = \{v_0, \ldots, v_{N-1}\}$ and $\mathbf{u}^0 = \{u_0, \ldots, u_{M-1}\}$ which satisfy $P_X(i) D(i,j) + u_i + P_X(i) v_j > 0$, and $\gamma^0 > 0, t^0 \geq 1, \mu > 1$.
2. **While** $((NM)/t \geq \epsilon)$
   1) *Centering step:* Minimizing **(13)** with initial point $(\mathbf{v}, \mathbf{u}, \gamma)$, and denote the computed solution by $(\mathbf{v}^t, \mathbf{u}^t, \gamma^t)$.
   2) *Update:* $\mathbf{v} = \mathbf{v}^t$, $\mathbf{u} = \mathbf{u}^t$ and $\gamma = \gamma^t$.
   3) *Increase t:* $t = \mu t$.
   end while
3. Output: $P_Y(j) = e^{v_j/\gamma - 1}, j = 0, \ldots, N-1$.

TABLE II
COMPUTATIONAL TIME COMPARISON BETWEEN DLS_PF AND PF WITH RESPECT TO DIFFERENT HOST SIGNAL DISTRIBUTIONS AND PROBLEM SCALES

B	Cover pmf	Algorithm	#Iteration	Time(s)
16	Laplace	PF	57	0.0109
	$\mu = 7.5, b = 1$	DLS_PF	65	0.0160
	Laplace	PF	54	0.0103
	$\mu = 7.5, b = 4$	DLS_PF	70	0.0161
32	Laplace	PF	50	0.0135
	$\mu = 15.5, b = 2$	DLS_PF	65	0.0203
	Laplace	PF	68	0.0278
	$\mu = 15.5, b = 4$	DLS_PF	70	0.0299
64	Laplace	PF	80	0.0526
	$\mu = 31.5, b = 2$	DLS_PF	85	0.0853
	Laplace	PF	67	0.0435
	$\mu = 31.5, b = 8$	DLS_PF	79	0.0770
128	Laplace	PF	86	0.1667
	$\mu = 63.5, b = 2$	DLS_PF	97	0.3026
	Laplace	PF	70	0.1357
	$\mu = 63.5, b = 16$	DLS_PF	98	0.3120
256	Laplace	PF	91	0.8437
	$\mu = 127.5, b = 2$	DLS_PF	114	1.6180
	Laplace	PF	83	0.7549
	$\mu = 127.5, b = 16$	DLS_PF	116	1.6616

D. PF Algorithm for PLS Problem

For their similar formulation, the PLS problem (3) for an expected embedding rate H_Y can also be solved by the path-following algorithm. The penalized unconstrained objective function for the dual problem (8) is:

$$\begin{aligned}\text{minimize} f_t(\mathbf{w}) &= f_t(\mathbf{v}, \mathbf{u}, \gamma) \\ &= t(\gamma \sum_{j=0}^{N-1} e^{v_j/\gamma - 1} + \sum_{i=0}^{M-1} u_i - \gamma H_Y) \\ &- \sum_{i,j} \ln(P_X(i)D(i,j) + u_i + \\ &\quad P_X(i)v_j) - \ln(\gamma) \end{aligned} \tag{13}$$

where $\mathbf{w} = (\mathbf{v}, \mathbf{u}, \gamma) \in \mathbb{R}^{N+M+1}$ is the optimization variable. The corresponding optimal marked-signal distribution $P_Y(y)$ in this case is given by (9). The PF algorithm for PLS problem is described by Algorithm 6.

E. Simulations

To evaluate the performance of our DLS_PF algorithm for the practical DLS problem (2) for a given distortion constraint D_{av}, we compare it with the original PF algorithm for an input parameter γ. Considering the case $M = N = B$, we first run the PF algorithm and then use its resulted D_{av} as the input for DLS_PF algorithm. We range the embedding rate from maximal to minimal with 15 trials and average the performance. Results in Table II demonstrate that the time cost for the DLS_PF algorithm nearly doubles that of PF algorithm on average.

Analogously, the performance of our PLS_PF algorithm for the PLS problem (3) is nearly the same as the DLS_PF algorithm, thus comparison details are omitted.

V. IMPLEMENTATION DETAILS AND DISCUSSIONS

A. Choice of t^0

An important issue for path-following algorithm is the choice of initial value of t. If t^0 is too large, the first *Centering step* will require too many iterations. If t^0 is too small, the algorithm will require extra outer iterations, and possibly too many inner iterations in the first *Centering step*. As suggested in [21], a good choice of t^0 associated with the initial value $\mathbf{w}^0$ is given by minimizing the following problem:

$$\text{minimize} \quad \|\nabla f_t(\mathbf{w}^0)\|_2^2$$

which is a simple least-square problem, and can be solved analytically.

B. Special Structure of Hessian Matrix

For the original PF algorithm, the most time-consuming part is the computation for the step direction $\Delta\mathbf{w}$ in the *Centering step* (Newton's method). It's equivalent to solve the set of linear equations:

$$H(\Delta\mathbf{w})^T = g^T.$$

According to (11), for the original PF Algorithm, the gradient vector g is given by (14), and the Hessian matrix H is given by (15) at the bottom of the next page. The Hessian matrix in this case is highly structured, to be specific:

$$H = \nabla^2 f(\mathbf{w}) = \begin{bmatrix} A_{11} & A_{12} \\ A_{12}^T & A_{22} \end{bmatrix}$$

where A_{11} and A_{22} are both diagonal. So it is desirable to use *Schur complement*[4] [22] to solve the linear equations, and the *Schur complement* matrix is formed by

$$S = A_{22} - A_{12}^T A_{11}^{-1} A_{12} \tag{16}$$

which is symmetric and positive definite, thus make it efficient to be inverted by using Cholesky factorization. The roughly complexity of solving the linear equations is then

$$O(NM + M^2N + \left(\frac{1}{3}\right)M^3) \approx O(\underbrace{M^2N}_{form\ S} + \underbrace{\left(\frac{1}{3}\right)M^3}_{invert\ S}).$$

In our paper we assume $M = N = B$, which means that the time cost is dominated by matrix multiplication to form the symmetric *Schur complement* matrix S.

For the DLS_PF algorithm and PLS_PF Algorithm, the only difference is that the matrix A_{22} is an addition of a diagonal matrix and a highly sparse matrix. By observing (16), this change is negligible to form S, so the complexity for each inner iteration in Newton's method is nearly the same.

[4]Available: http://en.wikipedia.org/wiki/Schur complement.

c1	*c3*	*c4*
c2	*hi*	

Fig. 2. Context of a pixel.

C. PLS_PF Algorithm

In the PLS_PF Algorithm, when the input embedding rate approaches the upper bound of the entropy of the marked-signal, i.e., $H(Y) \rightarrow \ln(N)$, the Hessian matrix H in Newton's step may become ill-conditioned (singular) while the solution vector $\mathbf{w}$ gets close to optimal. Therefore, we have to check this and force quit the inner loop to avoid getting trapped, or alternatively we can use modified Cholesky factorization to rectify the positive definiteness of the Hessian matrix.

VI. Application in Code Construction for Reversible Data Hiding

Lin *et al.* [16] proposed a scalar code construction which can approach the rate-distortion bound (1). This coding method modifies the cover signals according to the optimal distribution of marked-signals, $P_Y(y)$, so both the encoder and decoder must estimate the optimal distribution of marked-signals before executing the encoding (decoding) process. We refer readers to [16] for the details of the coding method.

To illustrate the power of the scalar code construction, Lin *et al.* [16] also proposed a RDH scheme for gray-scale images by applying their coding method to the prediction errors (PE) of pixels. The embedding process first computes the PEs from left to right, and from top to bottom. Fig. 2 depicts the four neighboring pixels used for predicting the pixel h_i. The predicted value is defined as

$$\hat{h}_i = \frac{3}{8} \times c2 + \frac{3}{8} \times c3 + \frac{1}{8} \times c1 + \frac{1}{8} \times c4. \tag{17}$$

For the pixels at left-most column and top line, we pick the nearby pixel, i.e., $c3$ or $c2$, as the predicted value. For the right-most column we let $c4 = c3$ and we omit the top-left corner pixel. Then we have the PE defined as

$$\hat{d}_i = (h_i - \hat{h}_i + 128) \; mod \; 256. \tag{18}$$

Taking PEs as cover signals, the message is embedded in the sequence of PEs by executing the coding method in [16] to result a marked-sequence $\mathbf{y} = (y_1, \ldots, y_n)$. The corresponding pixel of the watermarked image is defined by

$$w_i = (y_i + \hat{h}_i - 128) \; mod \; 256. \tag{19}$$

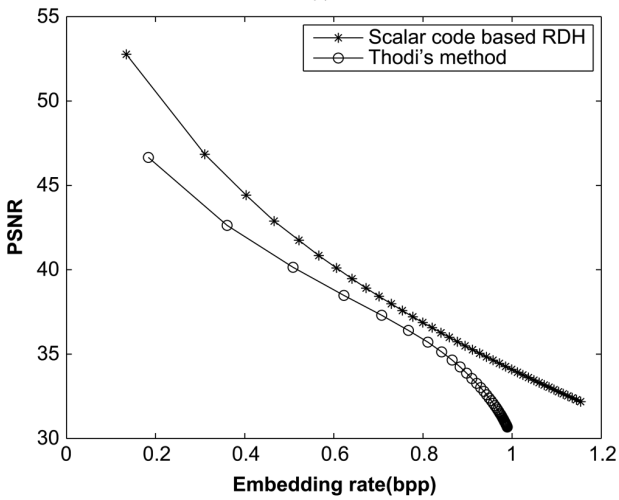
We call the above scheme as "Scalar code based RDH", in which we replace BFI algorithm with our DLS_PF algorithm to estimate the optimal distribution of marked-signals $P_Y(y)$. We compare this new RDH scheme with the method proposed by Thodi *et al.* [6]. Fig. 3(a) shows the test image Lena and Fig. 3(b) draws the rate-distortion curves of both methods, which shows that scalar code based RDH can increase the PSNR. Furthermore, the scalar code based method is capable of larger embedding rates as shown in Fig. 3(c). In fact, from Fig. 3(b), we can see that the scalar code based method significantly outperforms Thodi *et al.*'s method when the embedding rate is larger than 1.

Although both BFI algorithm [16] and the proposed DLS_PF (PLS_PF) algorithm can accurately estimate the optimal distribution of marked-signals, DLS_PF (PLS_PF) algorithm is more practical than BFI for applications. For instance, embedding message into the cover for a given distortion constraints D_{av} directly is generally appreciated. For the BFI algorithm, we need to use binary search to find the optimal α for the given D_{av} because no apparent translation between α and D_{av} is revealed. We implemented scalar code based RDH based on binary search

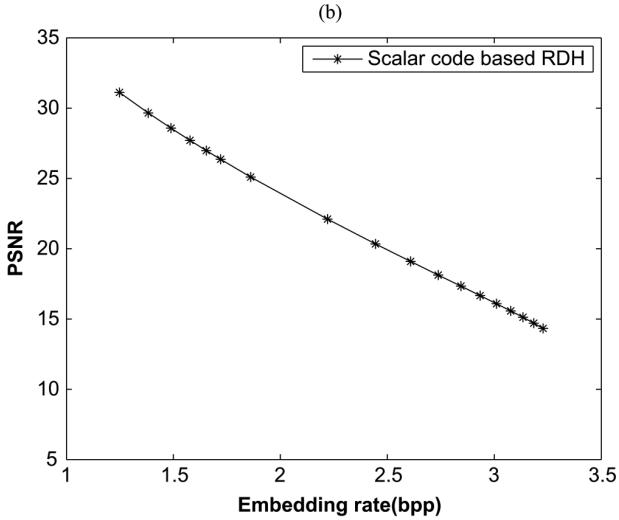
$$\begin{aligned} g = \nabla f_t(\mathbf{w}) &= \left(\frac{\partial f_t}{\partial v_0}, \ldots, \frac{\partial f_t}{\partial v_{N-1}}, \frac{\partial f_t}{\partial u_0}, \ldots, \frac{\partial f_t}{\partial u_{M-1}}\right) \\ &= \Bigg(t * e^{v_0 - 1} - \sum_{i=0}^{M-1} \frac{P_X(i)}{\gamma P_X(i) D(i,0) + u_i + P_X(i) v_0}, \ldots, t * e^{v_{N-1}-1} \\ &\quad - \sum_{i=0}^{M-1} \frac{P_X(i)}{\gamma P_X(i) D(i,N-1) + u_i + P_X(i) v_{N-1}}, \\ &\quad t - \sum_{j=0}^{N-1} \frac{1}{\gamma P_X(0) D(0,j) + u_0 + P_X(0) v_j}, \ldots, \\ &\quad t - \sum_{j=0}^{N-1} \frac{1}{\gamma P_X(M-1) D(M-1,j) + u_{M-1} + P_X(M-1) v_j}\Bigg) \end{aligned} \tag{14}$$

$$\begin{aligned} H = \nabla^2 f(\mathbf{w}) &= \begin{bmatrix} \frac{\partial^2 f_t}{\partial v_0 \partial v_0} & \cdots & \frac{\partial^2 f_t}{\partial v_0 \partial u_{M-1}} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f_t}{\partial u_{M-1} \partial v_0} & \cdots & \frac{\partial^2 f_t}{\partial u_{M-1} \partial u_{M-1}} \end{bmatrix} \\ &= \begin{bmatrix} t * e^{v_0 - 1} + \sum_{i=0}^{M-1} \left(\frac{P_X(i)}{\gamma P_X(i) D(i,0) + u_i + P_X(i) v_0}\right)^2 & \cdots & \frac{P_X(M-1)}{(\gamma P_X(M-1) D(M-1,0) + u_{M-1} + P_X(M-1) v_0)^2} \\ \vdots & \ddots & \vdots \\ \frac{P_X(M-1)}{(\gamma P_X(M-1) D(M-1,0) + u_{M-1} + P_X(M-1) v_0)^2} & \cdots & \sum_{j=0}^{N-1} \frac{1}{(\gamma P_X(M-1) D(M-1,j) + u_{M-1} + P_X(M-1) v_j)^2} \end{bmatrix}. \end{aligned} \tag{15}$$

(a)



(b)



(c)

Fig. 3. Scalar code construction applied to gray-scale images. (a) Host image Lena (512 × 512). (b) Rate-distortion comparisons with Thodi's method [6]. (c) Rate-distortion curve for high embedding rates.

TABLE III
TIME CONSUMED FOR THE EMBEDDING PROCESS OF SCALAR CODE BASED RDH

	Time (s)
Binary search BFI	79.6065 + 1.3188 = 80.9253
DLS_PF	2.4496 + 1.3654 = 3.8150

BFI method and DLS_PF algorithm respectively, and compared their embedding speeds over 29 increasing values for D_{av}. For the Lena image, the average time consumed for both methods are illustrated in Table III.

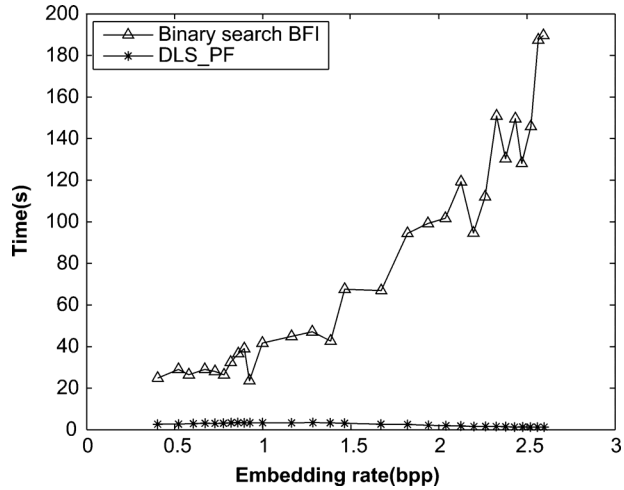


Fig. 4. Computational time comparison of scalar code based RDH with different distribution estimating algorithms for given distortion constraints.

As shown in Table III, the latter coding stage costs less time than the former optimizing stage in which the optimal marked-signal distribution $P_Y(y)$ is estimated. This means that the total time consumed by the embedding process is dominated by the former process to find the optimal $P_Y(y)$. From Fig. 4 we can see that the binary searching BFI method performs poorly, and our method is much more efficient in applications for reversible information hiding.

VII. CONCLUSIONS AND FUTURE WORK

In this paper, we have proposed a path-following algorithm to estimate the optimal distribution of the marked-signal for reversible data hiding. In comparison with the previous BFI method, the proposed algorithm performs much better and is efficient and scalable for practical applications.

Besides its simplicity for implementation, the proposed path-following algorithm is of high accuracy to approach the optimal solution as the penalty parameter t increases, even though each *Centering step* is not solved extremely accurately. The reason is that the resulted minimizer of the Newton's method is used to good-initialize the next unconstrained optimization problem for a larger t, so an extremely accurate minimizer is not necessary. This means we can choose a looser value for the parameter ϵ_2 in Algorithm 3. It will be valuable to explore gradient based or *quasi-Newton* methods to solve the *Centering step* in our proposed path-following algorithm in later work, which may make the algorithm more suitable for very large cardinality B.

To help readers to compare and use the proposed algorithms, we have posted our Matlab implementation of the algorithms at the following website: http://home.ustc.edu.cn/∼ hxc.

VIII. ACKNOWLEDGEMENTS

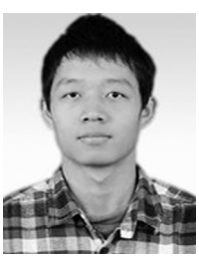
The authors thank S.-J. Lin *et al.* for offering the source codes of methods in their paper [16].

REFERENCES

[1] F. Bao *et al.*, "Tailored reversible watermarking schemes for authentication of electronic clinical atlas," *IEEE Trans. Inf. Technol. Biomed.*, vol. 9, no. 4, pp. 554–563, Dec. 2005.

[2] J. Feng *et al.*, "Reversible watermarking: Current status and key issues," *Int. J. Netw. Security*, vol. 2, no. 3, pp. 161–171, May 2006.

[3] K. Chung *et al.*, "Reversible data hiding-based approach for intra-frame error concealment in H.264/AVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 11, pp. 1643–1647, Nov. 2010.
[4] J. Fridrich and M. Goljan, "Lossless data embedding for all image formats," in *Proc. SPIE Photonics West, Electron. Imaging, Security and Watermarking of Multimedia Contents*, San Jose, Jan. 2002, vol. 4675, pp. 572–583.
[5] J. Tian, "Reversible data embedding using a difference expansion," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 8, pp. 890–896, Aug. 2003.
[6] D. Thodi and J. Rodriguez, "Expansion embedding techniques for reversible watermarking," *IEEE Trans. Image Process.*, vol. 16, no. 3, pp. 721–730, Mar. 2007.
[7] Y. Hu, H. Lee, and J. Li, "DE-based reversible data hiding with improved overflow location map," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 2, pp. 250–260, Feb. 2009.
[8] Z. Ni, Y. Shi, N. Ansari, and S. Wei, "Reversible data hiding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 3, pp. 354–362, Mar. 2006.
[9] P. Tsai, Y. C. Hu, and H. L. Yeh, "Reversible image hiding scheme using predictive coding and histogram shifting," *Signal Process.*, vol. 89, pp. 1129–1143, 2009.
[10] L. Luo *et al.*, "Reversible image watermarking using interpolation technique," *IEEE Trans. Inf. Forensics Security*, vol. 5, no. 1, pp. 187–193, Mar. 2010.
[11] F. Peng, X. Li, and B. Yang, "Adaptive reversible data hiding scheme based on integer transform," *Signal Process.*, vol. 92, no. 1, pp. 54–62, Jan. 2012.
[12] X. Li, B. Yang, and T. Zeng, "Efficient reversible watermarking based on adaptive prediction-error expansion and pixel selection," *IEEE Trans. Image Process.*, vol. 20, no. 12, pp. 3524–3533, Dec. 2011.
[13] T. Kalker and F. M. Willems, "Capacity bounds and code constructions for reversible data-hiding," in *Proc. 14th Int. Conf. Digital Signal Processing (DSP 2002)*, 2002, pp. 71–76.
[14] W. Zhang, B. Chen, and N. Yu, "Capacity-approaching codes for reversible data hiding," in *Proc. 13th Inform. Hiding Conf.*, Prague, May 2011.
[15] W. Zhang, B. Chen, and N. Yu, "Improving various reversible data hiding schemes via optimal codes for binary covers," *IEEE Trans. Image Process.*, vol. 21, no. 6, pp. 2991–3003, Jun. 2012.
[16] S.-J. Lin and W.-H. Chung, "The scalar scheme for reversible information-embedding in gray-scale signals: Capacity evaluation and code constructions," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 4, pp. 1155–1167, Aug. 2012.
[17] F. Willems, D. Maas, and T. Kalker, "Semantic lossless source coding," in *Proc. 42nd Ann. Allerton Conf. Commun., Control and Comput.*, Monticello, IL, USA, 2004 [Online]. Available: http://www.sps.ele.tue.nl/members/f.m.j.willems/research_files/SemanticCoding/allerton2004.pdf
[18] CVX: Matlab Software for Disciplined Convex Programming. ver. 2.0 beta, CVX Research, Inc., Sep. 2012 [Online]. Available: http://cvxr.com/cvx
[19] S. Boyd and L. Vandenberghe, "Duality," in *Convex Optimization*. New York, NY, USA: Cambridge Univ. Press, 2004, ch. 5.
[20] S. Boyd and L. Vandenberghe, "Unconstrained Minimization," in ,*Convex Optimization*. New York, NY, USA: Cambridge Univ. Press, 2004, ch. 9.
[21] S. Boyd and L. Vandenberghe, "Interior-Point Methods," in *Convex Optimization*. New York, NY, USA: Cambridge Univ. Press, 2004, ch. 11.
[22] S. Boyd and L. Vandenberghe, "Block elimination and Schur complements," in ,*Convex Optimization*. New York, NY, USA: Cambridge Univ. Press, 2004, Appendices C.4.



Xiaocheng Hu received the B.S. degree in 2010 from the University of Science and Technology of China, where he is now working toward the Ph.D. degree.

His research interests include multimedia security, image and video processing, video compression, and information hiding.

Weiming Zhang received the M.S. and Ph.D. degrees in 2002 and 2005, respectively, from the Zhengzhou Information Science and Technology Institute, China.

Currently, he is an associate professor with the School of Information Science and Technology, University of Science and Technology of China. His research interests include multimedia security, information hiding, and cryptography.

Xuexian Hu received the B.S. degree in 2004, M.E. degree in 2007, and Ph.D. degree in 2010, from Zhengzhou Information Science and Technology Institute of China, where he has been a lecturer since then.

Currently, he is a postdoctor at the Institute of Software, Chinese Academy of Sciences. His research interests include cryptology, network security, and information hiding.

Nenghai Yu received the B.S. degree in 1987 from Nanjing University of Posts and Telecommunications, the M.E. degree in 1992 from Tsinghua University, and the Ph.D. degree in 2004 from University of Science and Technology of China, where he is currently a professor.

His research interests include multimedia security, multimedia information retrieval, video processing, and information hiding.

Xianfeng Zhao received the Ph.D. degree in computer science from Shanghai Jiao Tong University, Shanghai, China, in 2003.

From 2003 to 2005, he was a postdoctoral fellow with the Data Assurance and Communication Security Center, Chinese Academy of Sciences (CAS), Beijing. From 2006 to 2011, he was an associate professor with the State Key Laboratory of Information Security (SKLOIS), Institute of Software, CAS, Beijing. Since 2012, he has been a professor with SKLOIS, which was moved to the Institute of Information Engineering, CAS, Beijing, in 2012. He is a member of IEEE, China Computer Federation, and Chinese Association for Cryptologic Research. His research interests include information hiding and multimedia security.

Fenghua Li received the B.S., M.S., and Ph.D. degrees in computer software and computer systems architecture from Xidian University, China, in 1987, 1990, and 2009 respectively.

He had been a lecturer with Xidian University from 1992 to 1994. He became an associate professor and professor with Beijing Electronic Science and Technology Institute in 1995 and 2001, respectively. Since 2011 he has been with State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences as a professor, and doctoral supervisor director. His research interests include network security, system security and evaluation, and trusted computation.