Gibbs Construction in Steganography

Tomáš Filler, *Student Member, IEEE* and Jessica Fridrich, *Member, IEEE*

***Abstract*—We make a connection between steganography design by minimizing embedding distortion and statistical physics. The unique aspect of this work and one that distinguishes it from prior art is that we allow the distortion function to be arbitrary, which permits us to consider spatially-dependent embedding changes. We provide a complete theoretical framework and describe practical tools, such as the thermodynamic integration for computing the rate–distortion bound and the Gibbs sampler for simulating the impact of optimal embedding schemes and constructing practical algorithms. The proposed framework reduces the design of secure steganography in empirical covers to the problem of finding local potentials for the distortion function that correlate with statistical detectability in practice. By working out the proposed methodology in detail for a specific choice of the distortion function, we experimentally validate the approach and discuss various options available to the steganographer in practice.**

***Index Terms*—Steganography, embedding impact, Markov random field, Gibbs sampling**

I. Introduction

THERE exist two general and widely used principles for designing steganographic methods for empirical cover objects, such as digital images. The first one is model-preserving steganography in which the designer adopts a model of the cover source and then designs the embedding to either completely or approximately preserve the model [15], [25], [28], [30], [33]. This way, one can provide mathematical guarantee that the embedding is perfectly secure (or ϵ-secure) within the chosen model. A problem is that empirical cover objects are notoriously difficult to model accurately, and, as history teaches us, the model mismatch can be exploited by an attacker to construct a sensitive detection scheme. Even worse, preserving an oversimplified model could introduce a security weakness [2], [19], [37]. An obvious remedy is to use more complicated models that would better approximate the cover source. The major obstacle here is that most current model-preserving steganographic constructions are specific to a certain model and do not adapt easily to more complex models.

The second, quite pragmatic, approach avoids modeling the cover source altogether and, instead, minimizes a heuristically-defined embedding distortion (impact). Matrix embedding [6], wet paper codes [12], and minimal embedding distortion steganography [8], [10], [11], [18], [27] are examples of this philosophy. Despite its heuristic nature, the principle of minimum embedding distortion has produced the most secure steganographic methods for digital media known today, at least in terms of low statistical detectability as measured using blind steganalyzers [13], [18], [20], [27]. Most of these schemes, however, use a distortion function that is additive – the total distortion is a sum of individual pixel distortions *computed from the cover image*. Fundamentally, such a distortion function cannot capture interactions among embedding changes, which leads to suboptimality in practice. This deficiency affects especially adaptive schemes for which the embedding changes have a tendency to form clusters because the pixel distortion is derived from local content or some content-dependent side-information. For example, the embedding changes might follow edges or be concentrated in textured regions.

One discovers a relationship between both embedding principles when the distortion function is defined as a weighted norm of the difference between feature vectors of cover and stego objects in some properly chosen feature space [19], [23], an example of which are spaces utilized by blind steganalyzers. The projection onto the feature space is essentially equivalent to modeling the objects in a lower-dimensional Euclidean space. Consequently, minimizing the distortion between cover and stego objects in the feature space now becomes closely tied to model preservation. Yet again, in this case the distortion cannot be written as a sum of individual pixel distortions also because the features contain higher-order statistics, such as sample transition probability matrices of pixels or DCT coefficients modeled as Markov chains [4], [22], [24], [31].

The importance of modeling interactions among embedding changes in steganography has been indirectly recognized by the designers of MPSteg [3] (Matching Pursuit Steganography) and YASS [29], [32]. In MPSteg, the authors use an overcomplete basis and embed messages by replacing small blocks with other blocks with the hope of preserving dependencies among neighboring pixels. The YASS algorithm taught us that a high embedding distortion may not directly manifest as a high statistical detectability, a curious property that can most likely be attributed to the fact that the embedding modifications are content driven and mutually correlated. Both approaches are heuristic in nature and leave many important issues unanswered, including establishing performance bounds, evaluating the methods' performance w.r.t. to these bounds, and creating a methodology for achieving near-optimal performance.

The above discussion underlines the need for a more systematic approach to steganography that can consider mutual interaction of embedding modifications, which is the topic of this paper. The main contribution is a general framework for

The authors were supported by Air Force Office of Scientific Research under the research grants FA9550-08-1-0084 and FA9550-09-1-0147. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation there on. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied of AFOSR or the U.S. Government.

The authors would like to thank Avinash Varna and Tomáš Pevný for useful discussions.

Both authors are with the Department of Electrical and Computer Engineering, Binghamton University, NY 13902 USA (e-mail: tomas.filler@gmail.com, fridrich@binghamton.edu, Ph: 607-777-6177; fax: 607-777-4464)

embedding using arbitrary distortion functions and a complete practical methodology for minimizing embedding distortion in steganography. The approach is flexible as well as modular and allows the steganographer to work with non-additive distortion functions. We provide algorithms for computing the proper theoretical bounds expressing the maximal payload embeddable with a bounded distortion, for simulating the impact of a stegosystem operating on the bound, and for designing practical steganographic algorithms that operate near the bound. The algorithms leverage standard tools used in statistical physics, such as Markov chain Monte Carlo samplers or the thermodynamic integration.

The technical part of this paper starts in the next section, where we recall the basic result that embedding changes made by a steganographic method that minimizes embedding distortion must follow a particular form of Gibbs distribution. The main purpose of this section is to establish terminology and make connections between the concepts used in steganography and those in statistical physics. In Section III, we introduce the so-called separation principle, which includes several distinct tasks that must be addressed when developing a practical steganographic method. In particular, to design and evaluate practical schemes one needs to establish the relationship between the maximal payload embeddable using bounded distortion (the rate–distortion bound) and be able to simulate the impact of a scheme operating on the bound. In the special case when the embedding distortion can be expressed as a sum of distortions at individual pixels computed from the cover image (the so-called non-interacting embedding changes), the design of near-optimal embedding algorithms has been successfully resolved in the past. For completeness, and because the proposed framework builds upon these results, we briefly summarize such known achievements in Section IV. Continuing with the case of a general distortion function, in Section V we describe two useful tools for steganographers – the Gibbs sampler and the thermodynamic integration. The Gibbs sampler can be used to simulate the impact of optimal embedding and to construct practical steganographic schemes (in Sections VI and VII). The thermodynamic integration is a method for estimating the entropy and partition function in statistical physics and we use it for computing the rate–distortion bound in steganography. The design of practical embedding schemes begins in Section VI, where we study distortion functions that can be written as a sum of local potentials defined on cliques. In Section VII, we first discuss various options the new framework offers to the steganography designer and then make a connection between local potentials and image models used in blind steganalysis. The proposed framework is experimentally validated in Section VIII, where we also discuss various implementation issues. Finally, the paper is concluded in Section IX.

As this paper is directed towards researchers working in the field of information security and forensics, the authors decided to include in this paper some standard concepts and algorithms commonly used in statistical physics and explain their role and proposed usage in steganography. Even though this inclusion may seem redundant to some, we believe that this decision makes this paper self-contained as well as readable to a much wider spectrum of researchers who are anticipated to incorporate the proposed methods in their corresponding fields.

II. Gibbs distribution minimizes embedding distortion

We first recall a well-known and quite general fact that, for a given expected embedding distortion, the maximal payload is embedded when the embedding changes follow a Gibbs distribution. This establishes a connection between steganography and statistical physics, which, later in this paper, will enable us to compute rate–distortion bounds, simulate the impact of optimal embedding, and construct practical embedding algorithms.

First, we introduce basic concepts, notation, and terminology used throughout this paper. The calligraphic font will be used solely for sets, random variables will be typeset in capital letters, while their corresponding realizations will be in lowercase. Vectors will be always typeset in boldface lower case, while we reserve the blackboard style for matrices (e.g., $A_{i,j}$ is the ijth element of matrix $\mathbb{A}$). The symbol $\mathbb{R}$ denotes the set of real numbers.

Although the idea presented in this paper is certainly applicable to steganography in other objects than digital images, we describe the entire approach using the terms "image" and "pixel" for concreteness to simplify the language and to allow a smooth transition from theory to experimental validation, which is carried out for digital images.

An image $\mathbf{x} = (x_1, \dots, x_n) \in \mathcal{X} \triangleq \mathcal{I}^n$ is a regular lattice of elements (pixels) $x_i \in \mathcal{I}$, $i \in \mathcal{S}$, $\mathcal{S} = \{1, \dots, n\}$. The dynamic range, $\mathcal{I}$, depends on the character of the image data. For example, for an 8-bit grayscale image, $\mathcal{I} = \{0, 1, \dots, 255\}$. In general, x_i can stand not only for light intensity values in a raster image but also for transform coefficients, palette indices, audio samples, etc. The proposed framework remains valid even when x_i is organized into an arbitrary graph structure.

For notational simplicity and convenience, we adopt additional conventions. Given $\mathcal{J} \subset \mathcal{S}$, $\mathbf{x}_{\mathcal{J}} \triangleq \{x_i | i \in \mathcal{J}\}$ and $\mathbf{x}_{\sim\mathcal{J}} \triangleq \{x_i | i \in \mathcal{S} - \mathcal{J}\}$. The image $(x_1, \dots, x_{i-1}, y_i, x_{i+1}, \dots, x_n)$ will be abbreviated as $y_i\mathbf{x}_{\sim i}$. We will also use the Iverson bracket, $[P]$, defined as $[P] = 1$ when the statement P is true and zero otherwise. Finally, we reserve $\log x$ for the logarithm at the base of 2 and use $\ln x$ for the natural base, $h(x) = -x \log x - (1-x)\log(1-x)$ is the binary entropy function.

Every steganographic embedding scheme considered in this paper will be associated with a mapping that assigns to each cover $\mathbf{x} \in \mathcal{X}$ the pair $\{\mathcal{Y}, \pi\}$. Here, $\mathcal{Y} \subset \mathcal{X}$ is the set of all stego images $\mathbf{y}$ into which $\mathbf{x}$ is allowed to be modified by embedding and π is a probability mass function on $\mathcal{Y}$ that characterizes the actions of the sender. The embedding algorithm is such that, for a given cover $\mathbf{x}$, the stego image $\mathbf{y} \in \mathcal{Y}$ is sent with probability $\pi(\mathbf{y})$. The stego image is thus a random variable $\mathbf{Y}$ over $\mathcal{Y}$ with the distribution $P(\mathbf{Y} = \mathbf{y}) = \pi(\mathbf{y})$. Technically, the set $\mathcal{Y}$ and all concepts derived from it in this paper depend on $\mathbf{x}$. However, because $\mathbf{x}$ is simply a parameter that we *fix in the very beginning*, we simplify the notation and do not make the dependence on $\mathbf{x}$

explicit. Finally, we note that the maximal expected payload that the sender can communicate to the receiver in this manner is the entropy

$$H(\pi) \triangleq H(\mathbf{Y}) = -\sum_{\mathbf{y}\in\mathcal{Y}} \pi(\mathbf{y}) \log \pi(\mathbf{y}). \tag{1}$$

To put it another way, we define a steganographic method from the point of view of how it modifies the cover and only then we deal with the issues of how to use it for communication and how to optimize its performance. The optimization will involve finding the distribution π for given $\mathbf{x}$, $\mathcal{Y}$, and payload (distortion).

We will consider the following special form of the set $\mathcal{Y}$: $\mathcal{Y} = \mathcal{I}_1 \times \mathcal{I}_2 \times \cdots \times \mathcal{I}_n$, where $\mathcal{I}_i \subset \mathcal{I}$. For example, in Least Significant Bit (LSB) embedding, $\mathcal{I}_i = \{x_i, \overline{x}_i\}$, where the bar denotes the operation of flipping the LSB. In LSB matching [16] (also called ± 1 embedding) in an 8-bit grayscale image $\mathbf{x}$, $\mathcal{I}_i = \{x_i - 1, x_i, x_i + 1\}$ whenever $x_i \notin \{0, 255\}$ and $\mathcal{I}_i$ is appropriately modified for the boundary cases. When $|\mathcal{I}_i| = 2$ or 3 for all i, we will speak of binary and ternary embedding, respectively. In general, however, we allow the size of every set $\mathcal{I}_i$ to be different. For example, pixels not allowed to be modified during embedding (the so-called wet pixels [12]) have $\mathcal{I}_i = \{x_i\}$.

By sending a slightly modified version $\mathbf{y}$ of the cover $\mathbf{x}$, the sender introduces a distortion, which will be measured using a distortion function

$$D : \mathcal{Y} \to \mathbb{R}, \tag{2}$$

that is bounded, i.e., $|D(\mathbf{y})| < K$, for all $\mathbf{y} \in \mathcal{Y}$ for some sufficiently large K. Note that D also depends on $\mathbf{x}$. Allowing the distortion to be negative does not cause any problems because an embedding algorithm minimizes D if and only if it minimizes the non-negative distortion $D + K$. The need for negative distortion will become apparent later in Section VI-A.

The expected embedding distortion introduced by the sender is

$$E_\pi[D] = \sum_{\mathbf{y}\in\mathcal{Y}} \pi(\mathbf{y}) D(\mathbf{y}). \tag{3}$$

An important premise we now make is that the sender is able to define the distortion function so that it is related to statistical detectability.[1] This assumption is motivated by a rather large body of experimental evidence, such as [13], [20], that indicates that even simple distortion measures that merely count the number of embedding changes correlate well with statistical detectability in the form of decision error of steganalyzers trained on cover and stego images. In general, steganographic methods that introduce smaller distortion disturb the cover source less than methods that embed with larger distortion.

Distortion-limited sender. To maximize the security, the so-called distortion-limited sender attempts to find a distribution π on $\mathcal{Y}$ that has the highest entropy and whose expected embedding distortion does not exceed a given D_ϵ:

$$\underset{\pi}{\text{maximize}} \quad H(\pi) = -\sum_{\mathbf{y}\in\mathcal{Y}} \pi(\mathbf{y}) \log \pi(\mathbf{y}) \tag{4}$$

$$\text{subject to} \quad E_\pi[D] = \sum_{\mathbf{y}\in\mathcal{Y}} \pi(\mathbf{y}) D(\mathbf{y}) = D_\epsilon. \tag{5}$$

By fixing the distortion, the sender fixes the security and aims to communicate as large payload as possible at this level of security. The maximization in (4) is carried over all distributions π on $\mathcal{Y}$. We will comment on whether the distortion constraint should be in the form of equality or inequality shortly.

Payload-limited sender. Alternatively, in practice it may be more meaningful to consider the payload-limited sender who faces a complementary task of embedding a *given* payload of m bits with minimal possible distortion. The optimization problem is to determine a distribution π that communicates a required payload while minimizing the distortion:

$$\underset{\pi}{\text{minimize}} \quad E_\pi[D] = \sum_{\mathbf{y}\in\mathcal{Y}} \pi(\mathbf{y}) D(\mathbf{y}) \tag{6}$$

$$\text{subject to} \quad H(\pi) = m. \tag{7}$$

The optimal distribution π for both problems has the Gibbs form

$$\pi_\lambda(\mathbf{y}) = \frac{1}{Z(\lambda)} \exp(-\lambda D(\mathbf{y})), \tag{8}$$

where $Z(\lambda)$ is the normalizing factor

$$Z(\lambda) = \sum_{\mathbf{y}\in\mathcal{Y}} \exp(-\lambda D(\mathbf{y})). \tag{9}$$

The optimality of π_λ follows immediately from the fact that for any distribution μ with $E_\mu[D] = \sum_{\mathbf{y}\in\mathcal{Y}} \mu(\mathbf{y}) D(\mathbf{y}) = D_\epsilon$, the difference between their entropies, $H(\pi_\lambda) - H(\mu) = D_{\mathrm{KL}}(\mu||\pi_\lambda) \geq 0$ [38]. The scalar parameter $\lambda > 0$ needs to be determined from the distortion constraint (5) or from the payload constraint (7), depending on the type of the sender. Provided m or D_ϵ are in the feasibility region of their corresponding constraints, the value of λ is unique. This follows from the fact that both the expected distortion and the entropy are monotone decreasing in λ. To see this, realize that by direct evaluation

$$\frac{\partial}{\partial\lambda} E_{\pi_\lambda}[D] = -Var_{\pi_\lambda}[D] \leq 0, \tag{10}$$

where $Var_{\pi_\lambda}[D] = E_{\pi_\lambda}[D^2] - (E_{\pi_\lambda}[D])^2$. Substituting (8) into (1), the entropy of the Gibbs distribution can be written as

$$H(\pi_\lambda) = \log Z(\lambda) + \frac{1}{\ln 2} \lambda E_{\pi_\lambda}[D]. \tag{11}$$

Upon differentiating and using (10), we obtain

$$\frac{\partial}{\partial\lambda} H(\pi_\lambda) = \frac{1}{\ln 2} \left(\frac{Z'(\lambda)}{Z(\lambda)} + E_{\pi_\lambda}[D] - \lambda Var_{\pi_\lambda}[D] \right) \tag{12}$$

$$= -\frac{\lambda}{\ln 2} Var_{\pi_\lambda}[D] \leq 0. \tag{13}$$

The monotonicity also means that the equality distortion constraint in the optimization problem (5) can be replaced

[1] The ability of a warden to distinguish between cover and stego images using statistical hypothesis testing.

with inequality, which is perhaps more appropriate given the motivating discussion above.

By varying $\lambda \in [0, \infty)$, we obtain a relationship between the maximal expected payload (1) and the expected embedding distortion (3). For brevity, we will call this relationship the rate–distortion bound. What distinguishes this concept from a similar notion defined in information theory is that we consider the bound for a *given* cover $\mathbf{x}$ rather than for $\mathbf{X}$, which is a random variable. At this point, we feel that it is appropriate to note that while it is certainly possible to consider $\mathbf{x}$ to be generated by a cover source with a known distribution and approach the design of steganography from a different point of view, namely one in which π_λ is determined by minimizing the KL divergence between the distributions of cover and stego images while satisfying a payload constraint, we do not do so in this paper.

Finally, we note that the assumption $|D(\mathbf{y})| < K$ implies that all stego objects appear with nonzero probability, $\pi_\lambda(\mathbf{y}) \geq \frac{1}{Z(\lambda)} \exp(-\lambda K)$, a fact that is crucial for the theory developed in the rest of this paper.

Remark 1: In statistical physics, the term distortion is known as energy. The optimality of Gibbs distribution is formulated as the Gibbs variational principle: "Among all distributions with a given energy, the Gibbs distribution (8) has the highest entropy." The parameter λ is called the inverse temperature, $\lambda = 1/kT$, where T is the temperature and k the Boltzmann constant. The normalizing factor $Z(\lambda)$ is called the partition function.

III. The separation principle

The design of steganographic methods that attempt to minimize embedding distortion should be driven by their performance. The obvious choice here is to contrast the performance with the rate–distortion bound. This is a meaningful comparison for the distortion-limited sender who can assess the performance of a practical embedding scheme by its loss of payload w.r.t. the maximum payload embeddable using a fixed distortion. This so-called "coding loss" informs the sender of how much payload is lost for a fixed statistical detectability. On the other hand, it is much harder for the payload-limited sender to assess how the increased distortion of a suboptimal practical scheme impacts statistical detectability in practice. We could resolve this rather important practical issue if we were able to simulate the impact of a scheme that operates *on the bound*.[2] Because the problems of establishing the bounds, simulating optimal embedding, and creating a practical embedding algorithm are really three separate problems, we call this reasoning the *separation principle*. It involves addressing the following three tasks:

1) **Establishing the rate–distortion bounds.** This means solving the optimization problems (4) or (6) and expressing the largest payload embeddable using a bounded distortion (or minimal distortion needed to embed a given payload). These bounds inform the steganographer about the best performance that can be theoretically achieved. Depending on the form of the distortion function D, establishing the bounds is usually rather challenging and one may have to resort to numerical methods (Section V-B). For an additive distortion (to be precisely defined shortly), an analytic form of the bounds may be obtained (Section IV).
2) **Simulating an optimal embedding method.** Often, it is very hard to construct a practical embedding method that performs close to the bound. However, we may be able to simulate the impact of such an optimal method and thus subject it to tests using steganalyzers even when we do not know how to construct a practical embedding algorithm or even compute the bound (see Section V). This is important for developers as one can effectively "prune" the design process and focus on implementing the most promising candidates. The simulator will also inform the payload-limited sender about the potential improvement in statistical undetectability should the theoretical performance gap be closed. A simple example is provided by the case of the Hamming distortion function $D(\mathbf{y}) = \sum_i [y_i \neq x_i]$. Here, the maximal relative payload $\alpha = m/n$ (in bits per pixel or bpp) is bounded by $\alpha \leq h(\beta)$, where $\beta = \frac{1}{n} D_\epsilon$ is the relative embedding distortion known as the change rate. In this case, one can simulate the embedding impact of the optimal scheme by independently changing each pixel with probability $h^{-1}(\alpha)$.
3) **Constructing a practical near-optimal embedding method.** This point is of most interest to practitioners. The bounds and the simulator are necessary to evaluate the performance of any practical scheme. The designer tries to maximize the embedding throughput (the number of bits embedded per unit time) while embedding as close to the distortion bound as possible.

It should be stressed at this point that even though the optimal distribution of embedding modifications has a known analytic expression (8), it may be infeasible to compute the individual probabilities $\pi_\lambda(\mathbf{y})$ due to the complexity of evaluating the partition function $Z(\lambda)$, which is a sum over all $\mathbf{y}$, whose count can be a very large number even for small images. (For example, there are 2^n binary flipping patterns in LSB embedding.) This also implies that at present we do not know how to compute the expected distortion (3) or the entropy (1) (these tasks are postponed to Section V). Fortunately, in many cases of practical interest we do not need to evaluate $\pi_\lambda(\mathbf{y})$ and will do just fine with being able to merely *sample from* π_λ. The ability to sample from π_λ is sufficient to simulate optimal embedding and realize practical embedding algorithms, and, in our case, even compute the rate–distortion bound.

In some special cases, however, such as when the embedding changes do not interact, the distortion D is additive and one can easily compute λ and the probabilities, evaluate the expected distortion and payload, and even construct near-optimal embedding schemes. As this special case will be used later in Section VII to design steganography with more general distortion functions D, we review it briefly in the next section.

[2] A scheme whose embedding distortion and payload lay on the rate–distortion bound derived for a given cover.

IV. Non-interacting embedding changes

When the distortion function D is additive over the pixels,

$$D(\mathbf{y}) = \sum_{i=1}^{n} \rho_i(y_i), \tag{14}$$

with bounded $\rho_i : \mathcal{I}_i \to \mathbb{R}$, we say that the embedding changes do not interact. In this case, the probability $\pi_\lambda(\mathbf{y})$ can be factorized into a product of marginal probabilities of changing the individual pixels (this follows directly from (8)):

$$\pi_\lambda(\mathbf{y}) = \prod_{i=1}^{n} \pi_\lambda(y_i) = \prod_{i=1}^{n} \frac{\exp(-\lambda\rho_i(y_i))}{\sum_{t_i\in\mathcal{I}_i}\exp(-\lambda\rho_i(t_i))}. \tag{15}$$

The expected distortion and the maximal payload are:

$$E_{\pi_\lambda}[D] = \sum_{i=1}^{n}\sum_{t_i\in\mathcal{I}_i} \pi_\lambda(t_i)\rho_i(t_i), \tag{16}$$

$$H(\pi_\lambda) = -\sum_{i=1}^{n}\sum_{t_i\in\mathcal{I}_i} \pi_\lambda(t_i)\log\pi_\lambda(t_i). \tag{17}$$

The impact of optimal embedding can be simulated by changing x_i to y_i with probabilities $\pi_\lambda(y_i)$ independently of the changes at other pixels. Since these probabilities can now be easily evaluated for a fixed λ, finding λ that satisfies the distortion ($E_{\pi_\lambda}[D] = D_\epsilon$) or the payload ($H(\pi_\lambda) = m$) constraint amounts to solving an algebraic equation for λ (see [10] or [9]). Because both the expected distortion and the entropy are monotone w.r.t. λ, the solution is unique. The only practical near-optimal embedding algorithm for this case known to the authors is based on syndrome-trellis codes [7].

It will be instructional to work out as an example the details of the special case of binary embedding for which $\mathcal{I}_i = \{x_i^{(0)}, x_i^{(1)}\}$ with $x_i^{(0)} = x_i$. Thus, ρ_i attains only two values, $\rho_i^{(t)} = \rho_i(x_i^{(t)})$, $t = 0, 1$. We stress at this point that we do *not* assume that $\rho_i^{(0)} = 0$ or even that $\rho_i^{(1)} \geq \rho_i^{(0)}$. This fact will be important when implementing practical embedding schemes in Section VI-A. The above expressions simplify to

$$\pi_\lambda\big(x_i^{(1)}\big) = \frac{\exp\big(-\lambda\rho_i^{(1)}\big)}{\exp\big(-\lambda\rho_i^{(1)}\big) + \exp\big(-\lambda\rho_i^{(0)}\big)} \tag{18}$$

$$= \frac{1}{1+\exp\big(-\lambda\big(\rho_i^{(0)} - \rho_i^{(1)}\big)\big)} \triangleq p_i(\lambda), \tag{19}$$

$$E_{\pi_\lambda}[D] = \sum_{i=1}^{n} \rho_i^{(0)}(1-p_i(\lambda)) + \rho_i^{(1)}p_i(\lambda), \tag{20}$$

$$H(\pi_\lambda) = \sum_{i=1}^{n} h(p_i(\lambda)). \tag{21}$$

The smallest distortion any binary embedding algorithm can impose is $D_{\min} = \sum_{i=1}^{n}\min\{\rho_i^{(0)}, \rho_i^{(1)}\}$, which would be incurred when selecting $y_i = x_i^{(t_i)}$, where $t_i = \arg\min_t\{\rho_i^{(t)}\}$. Thus,

$$D(\mathbf{y}) = \sum_{i=1}^{n} \rho_i^{(0)}\big[y_i = x_i^{(0)}\big] + \rho_i^{(1)}\big[y_i = x_i^{(1)}\big] \tag{22}$$

$$= D_{\min} + \sum_{i=1}^{n} \varrho_i\big[y_i \neq x_i^{(t_i)}\big], \tag{23}$$

where $\varrho_i = |\rho_i^{(1)} - \rho_i^{(0)}|$ is now a vector of non-negative distortions, which allows us to apply the practical embedding algorithm described in [8]. It accepts on its input a bit stream $\mathbf{c} = (c_1(\mathbf{x}), \ldots, c_n(\mathbf{x}))$ (representing the cover $\mathbf{x}$), the vector of non-negative distortions $(\varrho_1, \ldots, \varrho_n)$, and a binary message. It outputs a modified (stego) bit stream $\mathbf{y} \in \{0,1\}^n$ that conveys the message as a syndrome of a suitably chosen syndrome-trellis code so that the total embedding distortion $\sum_{i=1}^{n}\varrho_i[y_i \neq c_i]$ is near minimal. It follows from (23) that binary embedding as defined in this section can be implemented in practice by applying this algorithm to the bit stream $c_i(\tilde{\mathbf{x}})$, $\tilde{\mathbf{x}} = (x_1^{(t_1)}, \ldots, x_n^{(t_n)})$.

Finally, we note that the complete derivation of the rate-distortion bound for binary embedding appears, e.g., in Chapter 7 of [9].

V. Simulated embedding and rate–distortion bound

In Section II, we showed that minimal-embedding-distortion steganography should select the stego image $\mathbf{y}$ with probability $\pi_\lambda(\mathbf{y}) \propto \exp(-\lambda D(\mathbf{y}))$ expressed in the form of a Gibbs distribution. We now explain a general iterative procedure using which one can sample from any Gibbs distribution and thus simulate optimal embedding. The method is recognized as one of the Markov Chain Monte Carlo (MCMC) algorithms known as the Gibbs sampler.[3] This sampling algorithm will allow us to construct practical embedding schemes in Sections VI and VII. We also explain how to compute the rate–distortion bound for a fixed image using the thermodynamic integration. The Gibbs sampler and the thermodynamic integration appear, for example, in [38] and [21], respectively.

A. The Gibbs sampler

We start by defining the local characteristics of a Gibbs field as the conditional probabilities of the ith pixel attaining the value y_i' conditioned on the rest of the image:

$$\pi_\lambda(Y_i = y_i'|\mathbf{Y}_{\sim i} = \mathbf{y}_{\sim i}) = \frac{\pi_\lambda(y_i'\mathbf{y}_{\sim i})}{\sum_{t_i\in\mathcal{I}_i}\pi_\lambda(t_i\mathbf{y}_{\sim i})}. \tag{24}$$

For all possible stego images $\mathbf{y}, \mathbf{y}' \in \mathcal{Y}$, the local characteristics (24) define the following matrices $\mathbb{P}(i)$, for each pixel $i \in \{1, \ldots, n\}$:

$$P_{\mathbf{y},\mathbf{y}'}(i) = \begin{cases} \pi_\lambda(Y_i = y_i'|\mathbf{Y}_{\sim i} = \mathbf{y}_{\sim i}) & \text{when } \mathbf{y}'_{\sim i} = \mathbf{y}_{\sim i} \\ 0 & \text{otherwise.} \end{cases} \tag{25}$$

Every matrix $\mathbb{P}(i)$ has $|\mathcal{Y}|$ rows and the same number of columns (which means it is very large) and its elements are mostly zero except when $\mathbf{y}'$ was obtained from $\mathbf{y}$ by modifying y_i to y_i' and all other pixels stayed the same. Because $\mathbb{P}(i)$ is stochastic (the sum of its rows is one),

$$\sum_{\mathbf{y}'\in\mathcal{Y}} P_{\mathbf{y},\mathbf{y}'}(i) = 1, \text{ for all rows } \mathbf{y}, \tag{26}$$

[3] More detailed discussion regarding our choice of the MCMC sampler appear later in this section.

Algorithm 1 One sweep of a Gibbs sampler.

1: Set pixel counter $i = 1$
2: **while** $i \leq n$ **do**
3: Compute the local characteristics:

$$P_{y'_{\sigma(i)}\mathbf{y}_{\sim\sigma(i)},\mathbf{y}}(\sigma(i)),\ y'_{\sigma(i)} \in \mathcal{I}_{\sigma(i)} \tag{34}$$

4: Select one $y'_{\sigma(i)} \in \mathcal{I}_{\sigma(i)}$ pseudorandomly according to the probabilities (34) and change $y_{\sigma(i)} \leftarrow y'_{\sigma(i)}$
5: $i \leftarrow i + 1$
6: **end while**
7: **return** $\mathbf{y}$

$\mathbb{P}(i)$ is a transition probability matrix of some Markov chain on $\mathcal{Y}$. All such matrices satisfy the so-called detailed balance equation

$$\pi_\lambda(\mathbf{y})P_{\mathbf{y},\mathbf{y}'}(i) = \pi_\lambda(\mathbf{y}')P_{\mathbf{y}',\mathbf{y}}(i), \quad \text{for all } \mathbf{y},\mathbf{y}' \in \mathcal{Y}, i. \tag{27}$$

To see this, realize that unless $\mathbf{y}_{\sim i} = \mathbf{y}'_{\sim i}$, we are looking at the trivial equality $0 = 0$. For $\mathbf{y}_{\sim i} = \mathbf{y}'_{\sim i}$, we have the following chain of equalities:

$$\pi_\lambda(\mathbf{y})P_{\mathbf{y},\mathbf{y}'}(i) \overset{(a)}{=} \pi_\lambda(\mathbf{y})\frac{\pi_\lambda(y'_i\mathbf{y}_{\sim i})}{\sum_{t_i\in\mathcal{I}_i}\pi_\lambda(t_i\mathbf{y}_{\sim i})} \tag{28}$$

$$\overset{(b)}{=} \frac{\pi_\lambda(\mathbf{y})\pi_\lambda(\mathbf{y}')}{\sum_{t_i\in\mathcal{I}_i}\pi_\lambda(t_i\mathbf{y}_{\sim i})} \tag{29}$$

$$= \pi_\lambda(\mathbf{y}')\frac{\pi_\lambda(\mathbf{y})}{\sum_{t_i\in\mathcal{I}_i}\pi_\lambda(t_i\mathbf{y}'_{\sim i})} \tag{30}$$

$$\overset{(c)}{=} \pi_\lambda(\mathbf{y}')P_{\mathbf{y}',\mathbf{y}}(i). \tag{31}$$

Equality (a) follows from the definition of $\mathbb{P}(i)$ (25), (b) from the fact that $\mathbf{y}_{\sim i} = \mathbf{y}'_{\sim i}$, and (c) from $\pi_\lambda(\mathbf{y}) = \pi_\lambda(y_i\mathbf{y}'_{\sim i})$ and again (25).

Next, we define the boldface symbol $\boldsymbol{\pi}_\lambda \in [0,\infty)^{|\mathcal{Y}|}$ as the vector of $|\mathcal{Y}|$ non-negative elements $\boldsymbol{\pi}_\lambda = \pi_\lambda(\mathbf{y})$, $\mathbf{y} \in \mathcal{Y}$. Using (27) and then (26), we can now easily show that the vector $\boldsymbol{\pi}_\lambda$ is the left eigenvector of $\mathbb{P}(i)$ corresponding to the unit eigenvalue:

$$(\boldsymbol{\pi}_\lambda\mathbb{P}(i))_{\mathbf{y}'} = \sum_{\mathbf{y}\in\mathcal{Y}}\pi_\lambda(\mathbf{y})P_{\mathbf{y},\mathbf{y}'}(i) \tag{32}$$

$$= \sum_{\mathbf{y}\in\mathcal{Y}}\pi_\lambda(\mathbf{y}')P_{\mathbf{y}',\mathbf{y}}(i) = \pi_\lambda(\mathbf{y}'). \tag{33}$$

In (32), $(\boldsymbol{\pi}_\lambda\mathbb{P}(i))_{\mathbf{y}'}$ is the $\mathbf{y}'$th element of the product of the vector $\boldsymbol{\pi}_\lambda$ and the matrix $\mathbb{P}(i)$.

We are now ready to describe the Gibbs sampler [14], which is a key element in our framework. Let σ be a permutation of the index set $\mathcal{S}$ called the visiting schedule ($\sigma(i)$, $i = 1,\dots,n$ is the ith element of the permutation σ). One sample from π_λ is then obtained by repeating a series of "sweeps" defined below. As we explain the sweeps and the Gibbs sampler, the reader is advised to inspect Algorithm 1 to better understand the process.

The sampler is initialized by setting $\mathbf{y}$ to some initial value. For faster convergence, a good choice is to select y_i from $\mathcal{I}_i$ according to the local characteristics $\pi_\lambda(y_i\mathbf{x}_{\sim i})$. A sweep is a procedure applied to an image during which all pixels are updated sequentially in the order defined by the visiting schedule σ. The pixels are updated based on their local characteristics (24) computed from the current values of the stego image $\mathbf{y}$. The entire sweep can be described by a transition probability matrix $\mathbb{P}(\sigma)$ obtained by matrix-multiplications of the individual transition probability matrices $\mathbb{P}(\sigma(i))$:

$$P_{\mathbf{y},\mathbf{y}'}(\sigma) \triangleq \left(\mathbb{P}(\sigma(1))\cdot\mathbb{P}(\sigma(2))\cdots\mathbb{P}(\sigma(n))\right)_{\mathbf{y},\mathbf{y}'}. \tag{35}$$

After each sweep, the next sweep continues with the current image $\mathbf{y}$ as its starting position. It should be clear from the algorithm that at the end of each sweep each pixel i has a non-zero probability to get into any of its states from $\mathcal{I}_i$ defined by the embedding operation (because D is bounded). This means that all elements of $\mathcal{Y}$ will be visited with positive probability and thus the transition probability matrix $\mathbb{P}(\sigma)$ corresponds to a homogeneous irreducible Markov process with a *unique* left eigenvector corresponding to a unit eigenvalue (unique stationary distribution). Because $\boldsymbol{\pi}_\lambda$ is a left eigenvector corresponding to a unit eigenvalue for each matrix $\mathbb{P}(i)$, it is also a left eigenvector for $\mathbb{P}(\sigma)$ and thus its stationary distribution due to its uniqueness. A standard result from the theory of Markov chains (see, e.g. Chapter 4 in [38]) states that, for an irreducible Markov chain, no matter what distribution of embedding changes $\boldsymbol{\nu} \in [0,\infty)^{|\mathcal{Y}|}$ we start with, and independently of the visiting schedule σ, with increased number of sweeps, k, the distribution of Gibbs samples converges in norm to the stationary distribution $\boldsymbol{\pi}_\lambda$:

$$||\boldsymbol{\nu}\left(\mathbb{P}(\sigma)\right)^k - \boldsymbol{\pi}_\lambda|| \to 0 \text{ with } k \to \infty \tag{36}$$

exponentially fast. This means that in practice we can obtain a sample from π_λ after running the Gibbs sampler for a sufficiently long time.[4] The visiting schedule can be randomized in each sweep as long as each pixel has a non-zero probability of being visited, which is a necessary condition for convergence.

B. Simulating optimal embedding

When applied to steganography, the Gibbs sampler allows the sender to simulate the effect of embedding using a scheme that operates on the bound. It is interesting that this can be done for any distortion function D and without knowing the rate–distortion bound. This is because the local characteristics (24)

$$\pi_\lambda(Y_i = y'_i|\mathbf{Y}_{\sim i} = \mathbf{y}_{\sim i}) = \frac{\exp(-\lambda D(y'_i\mathbf{y}_{\sim i}))}{\sum_{t_i\in\mathcal{I}_i}\exp(-\lambda D(t_i\mathbf{y}_{\sim i}))}, \tag{37}$$

do not require computing the partition function $Z(\lambda)$. We do need to know the parameter λ, though.

For the distortion-limited sender (5), the Gibbs sampler could be used directly to determine the proper value of λ in the following manner. For a given λ, it is known (Theorem 5.1.4 in [38]) that

$$\frac{1}{k}\sum_{j=1}^{k} D\big(\mathbf{y}^{(j)}\big) \to E_{\pi_\lambda}[D] \text{ as } k \to \infty \tag{38}$$

[4]The convergence time may vary significantly depending on the Gibbs field at hand.

in L_2 and in probability, where $\mathbf{y}^{(j)}$ is the image obtained after the jth sweep of the Gibbs sampler. This requires running the Gibbs sampler and averaging the individual distortions for a sufficiently long time. When only a finite number of sweeps is allowed, the first few images $\mathbf{y}$ should be discarded to allow the Gibbs sampler to converge close enough to π_λ. The value of λ that satisfies $E_{\pi_\lambda}[D] = D_\epsilon$ can be determined, for example, using a binary search over λ.

To find λ for the payload-limited sender (4), we need to evaluate the entropy $H(\pi_\lambda)$, which can be obtained from $E_{\pi_\lambda}[D]$ using the method of thermodynamic integration [21]. From (10) and (13), we obtain

$$\frac{\partial}{\partial\lambda}H(\pi_\lambda) = \frac{\lambda}{\ln 2}\frac{\partial}{\partial\lambda}E_{\pi_\lambda}[D]. \tag{39}$$

Therefore, the entropy can be estimated from $E_{\pi_\lambda}[D]$ by integrating by parts:

$$H(\pi_\lambda) = H(\pi_{\lambda_0}) + \left[\frac{\lambda'}{\ln 2}E_{\pi_{\lambda'}}[D]\right]_{\lambda_0}^{\lambda} - \frac{1}{\ln 2}\int_{\lambda_0}^{\lambda} E_{\pi_{\lambda'}}[D]\mathrm{d}\lambda'. \tag{40}$$

The value of λ that satisfies the entropy (payload) constraint can be again obtained using a binary search. Having obtained the expected distortion and the entropy using the Gibbs sampler and the thermodynamic integration, the rate–distortion bound $[H(\pi_\lambda), E_{\pi_\lambda}[D]]$ can be plotted as a curve parametrized by λ.

In practice, one has to be careful when using (38), since no practical guidelines exist for determining a sufficient number of sweeps and heuristic criteria are often used [5], [38]. Although the convergence to π_λ is exponential in the number of sweeps, in general a large number of sweeps may be needed to converge close enough. Generally speaking, the stronger the dependencies between embedding changes the more sweeps are needed by the Gibbs sampler. In theory, the convergence of MCMC methods, such as the Gibbs sampler, may also slow down in the vicinity of "phase transitions," which we loosely define here as sudden changes in the spatial distribution of embedding changes when only slightly changing the payload (or distortion bound).

In our experiments reported later in this paper, the Gibbs sampler always behaved well and converged fast. We attribute this to the fact that the dependencies among embedding modifications as measured using our distortion functions are rather weak and limited to short distances. The convergence, however, could become an issue for other types of cover sources with different distortion functions. While it is possible to compute the rate–distortion bounds and simulate optimal embedding using other MCMC algorithms, such as the Metropolis-Hastings sampler [38], that may converge faster than the Gibbs sampler and can exhibit a more robust behavior in practice, it is not clear how to adopt these algorithms for practical embedding. This is because all known coding methods in steganography essentially sample from a distribution of independent symbols. Thus, the Gibbs sampler comes out as a natural choice (Section VI) because it works by updating individual pixels, which is exactly the effect of embedding using syndrome-trellis codes [7], [8].

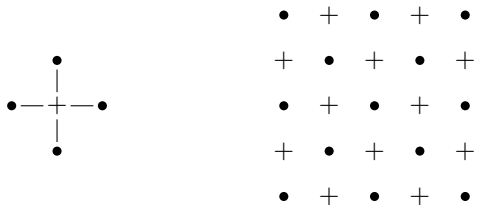

Figure 1. The four-element cross-neighborhood and the tessellation of the index set $\mathcal{S}$ into two disjoint sublattices $\mathcal{S}_e$ and $\mathcal{S}_o$.

Figure 2. All three possible cliques for the cross-neighborhood.

A notable alternative to the Gibbs sampler and the thermodynamic integration for computing the rate–distortion bound is the Wang–Landau algorithm [36] that estimates the so-called density of stego images (density of states in statistical physics), $g(D)$, defined as the number of stego images $\mathbf{y}$ with distortion (energy) D. The partition function (and thus, via (11), the entropy) and the expected distortion can be computed from $g(D)$ by numerical integration:

$$Z(\lambda) \doteq \sum_{D\in\mathcal{D}} g(D)\exp(-\lambda D)\Delta, \tag{41}$$

$$E_{\pi_\lambda}[D] \doteq \frac{1}{Z(\lambda)}\sum_{D\in\mathcal{D}} Dg(D)\exp(-\lambda D)\Delta, \tag{42}$$

where $\mathcal{D} = \{d_1, \ldots, d_{n_D}\}$, $d_1 = -K$, $d_{n_D} = K$, $d_i - d_{i-1} = \Delta$ is a set of discrete values into which the dynamic range of D, $[-K, K]$ is quantized.

The authors note that in general it is not possible to determine ahead of time which method will provide satisfactory performance. In our experiments described in Section VIII, the thermodynamic integration worked very well and provided results identical to the much more complex Wang–Landau algorithm.

Note that computing the rate–distortion bound is not necessary for practical embedding. In Section VI, we introduce a special form of the distortion in terms of a sum over local potentials. In this case, both types of optimal senders can be simulated using algorithms that do not need to compute λ in the fashion described above. This is explained in Sections VI-A and VI-B.

VI. Local distortion function

Thanks to the Gibbs sampler, we can simulate the impact of embedding that is optimal in the sense of (4) and (6) without having to construct a specific steganographic scheme. This is important for steganography design as we can test the effect of various design choices and parameters and then implement only the most promising constructs. However, it is rather difficult to design near-optimal schemes for a general $D(\mathbf{y})$. Fortunately, it is possible to give the distortion function a specific form that will allow us to construct practical embedding algorithms. We will assume that D is a sum of local potentials defined on small groups of pixels called cliques.

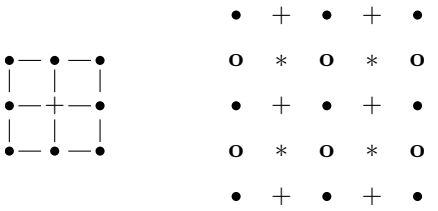

Figure 3. The eight-element neighborhood and the tessellation of the index set $\mathcal{S}$ into four disjoint sublattices marked with four different symbols.

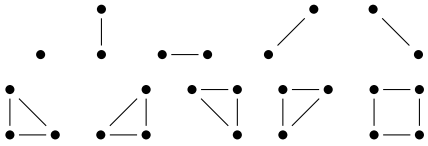

Figure 4. All possible cliques for the eight-element neighborhood.

This local form of the distortion will be still quite general to capture dependencies among embedding changes and it allows us to construct a large spectrum of diverse embedding schemes – a topic left for Section VII.

First, we define a neighborhood system as a collection of subsets of the index set $\{\eta(i) \subset \mathcal{S} | i = 1, \dots, n\}$ satisfying $i \notin \eta(i), \forall i$ and $i \in \eta(j)$ if and only if $j \in \eta(i)$. The elements of $\eta(i)$ are called neighbors of pixel i. A subset $c \subset \mathcal{S}$ is a clique if each pair of different elements from c are neighbors. The set of all cliques will be denoted $\mathcal{C}$. We do not use the calligraphic font for a clique even though it is a set (and thus deviate here from our convention) to comply with a well established notation used in previous art.

In this section and in Section VII, we will need to address pixels by their two-dimensional coordinates. We will thus be switching between using the index set $\mathcal{S} = \{1, \dots, n\}$ and its two-dimensional equivalent $\mathcal{S} = \{(i,j) | 1 \le i \le n_1, 1 \le j \le n_2\}$ hoping that it will cause no confusion for the reader.

Example 1: The four-element cross neighborhood of pixel $x_{i,j}$ consisting of $\{x_{i-1,j}, x_{i+1,j}, x_{i,j-1}, x_{i,j+1}\}$ with a proper treatment at the boundary forms a neighborhood system (see Figure 1). The cliques contain either a single pixel (one-element) cliques $\{x_{i,j}\}$ or two horizontally or vertically neighboring pixels, $\{x_{i,j}, x_{i,j+1}\}$, $\{x_{i,j}, x_{i+1,j}\}$ (Figure 2). No other cliques exist.

Example 2: The eight-element 3×3 neighborhood also forms a neighborhood system (Figure 3). The cliques are as in Example 1 as well as all cliques containing pairs of diagonally neighboring pixels, $\{x_{i,j}, x_{i+1,j+1}\}$, $\{x_{i,j}, x_{i-1,j+1}\}$, three-pixel cliques forming a right-angle triangle (e.g., $\{x_{i,j}, x_{i,j+1}, x_{i+1,j}\}$), and four-pixel cliques forming a 2×2 square ($\{x_{i,j}, x_{i,j+1}, x_{i+1,j}, x_{i+1,j+1}\}$) (follow Figure 4). No other cliques exist for this neighborhood system.

Each neighborhood system allows tessellation of the index set $\mathcal{S}$ into disjoint subsets (sublattices) whose union is the entire set $\mathcal{S}$, so that any two pixels in each lattice are not neighbors. For example, for the cross-neighborhood $\mathcal{S} = \mathcal{S}_e \cup \mathcal{S}_o$, where

$$\mathcal{S}_e = \{(i,j) | i+j \text{ is even}\}, \quad \mathcal{S}_o = \{(i,j) | i+j \text{ is odd}\}. \tag{43}$$

For the eight-element 3×3 neighborhood, there are four sublattices, $\mathcal{S} = \bigcup_{ab} \mathcal{S}_{ab}$, $1 \le a, b \le 2$, whose structure resembles the Bayer color filter array commonly used in digital cameras [9],

$$\mathcal{S}_{ab} = \{(a+2k, b+2l) | 1 \le a+2k \le n_1, 1 \le b+2l \le n_2\}. \tag{44}$$

For a clique $c \in \mathcal{C}$, we denote by $V_c(\mathbf{y})$ the local potential, which is an arbitrary bounded function that depends only on the values of $\mathbf{y}$ in the clique c, $V_c(\mathbf{y}) = V_c(\mathbf{y}_c)$. We remind that V_c may also depend on $\mathbf{x}$ in an arbitrary fashion. We are now ready to introduce a local form of the distortion function as

$$D(\mathbf{y}) = \sum_{c \in \mathcal{C}} V_c(\mathbf{y}_c). \tag{45}$$

The important fact is that D is a sum of functions with a small support. Let us express the local characteristics (24) in terms of this newly-defined form (45):

$$\pi_\lambda(Y_i = y_i' | \mathbf{y}_{\sim i}) = \frac{\exp(-\lambda \sum_{c \in \mathcal{C}} V_c(y_i' \mathbf{y}_{\sim i}))}{\sum_{t_i \in \mathcal{I}_i} \exp(-\lambda \sum_{c \in \mathcal{C}} V_c(t_i \mathbf{y}_{\sim i}))} \tag{46}$$

$$\stackrel{(a)}{=} \frac{\exp(-\lambda \sum_{c \in \mathcal{C}(i)} V_c(y_i' \mathbf{y}_{\sim i}))}{\sum_{t_i \in \mathcal{I}_i} \exp(-\lambda \sum_{c \in \mathcal{C}(i)} V_c(t_i \mathbf{y}_{\sim i}))}, \tag{47}$$

where $\mathcal{C}(i) = \{c \in \mathcal{C} | i \in c\}$, $i = 1, \dots, n$. Equality (a) holds because $V_c(t_i \mathbf{y}_{\sim i})$ does not depend on t_i for cliques $c \notin \mathcal{C}(i)$ as they do not contain the ith element. Thus, the terms V_c for such cliques cancel from (47). This has a profound impact on the local characteristics, making the realization of Y_i *independent* of changes made outside of the union of cliques containing pixel i and thus outside of the neighborhood $\eta(i)$. For the cross-neighborhood system from Example 1, changes made to pixels belonging to the sublattice $\mathcal{S}_e$ do not interact and thus the Gibbs sampler can be parallelized by first updating *all* pixels from this sublattice in parallel and then updating in parallel *all* pixels from $\mathcal{S}_o$.[5]

The possibility to update all pixels in each sublattice all at once provides a recipe for constructing practical embedding schemes. Assume $\mathcal{S} = \mathcal{S}_1 \cup \ldots \cup \mathcal{S}_s$ with mutually disjoint sublattices. We first describe the actions of a payload-limited sender (follow the pseudo-code in Algorithm 2).

A. Payload-limited sender

The sender divides the payload of m bits into s equal parts of m/s bits, computes the local distortions

$$\rho_i(y_i' \mathbf{y}_{\sim i}) = \sum_{c \in \mathcal{C}(i)} V_c(y_i' \mathbf{y}_{\sim i}) \tag{48}$$

for pixels $i \in \mathcal{S}_1$, and embeds the first message part in $\mathcal{S}_1$. Then, it updates the local distortions of all pixels from $\mathcal{S}_2$ and embeds the second part in $\mathcal{S}_2$, updates the local distortions again, embeds the next part in $\mathcal{S}_3$, etc. Because the embedding changes in each sublattice do not interact, the embedding can be realized as discussed in Section IV. After all sublattices are

[5]The Gibbs random field described by the joint distribution $\pi_\lambda(\mathbf{y})$ with distortion (45) becomes a Markov random field on the same neighborhood system. This follows from the Hammersley-Clifford theorem [38].

Algorithm 2 One sweep of a Gibbs sampler for embedding m-bit message (payload-limited sender).

Require: $\mathcal{S} = \mathcal{S}_1 \cup \ldots \cup \mathcal{S}_s$ {mutually disjoint sublattices}
1: **for** $k = 1$ to s **do**
2: **for** every $i \in \mathcal{S}_k$ **do**
3: Use (48) to calculate cost of changing $y_i \to y_i' \in \mathcal{I}_i$
4: **end for**
5: Embed m/s bits while minimizing $\sum_{i\in\mathcal{S}_k} \rho_i(y_i'\mathbf{y}_{\sim i})$.
6: Update $\mathbf{y}_{\mathcal{S}_k}$ with new values and keep $\mathbf{y}_{\sim\mathcal{S}_k}$ unchanged.
7: **end for**
8: **return** $\mathbf{y}$

Algorithm 3 One sweep of a Gibbs sampler for a distortion-limit sender, $E_{\pi_\lambda}[D] = D_\epsilon$.

Require: $\mathcal{S} = \mathcal{S}_1 \cup \ldots \cup \mathcal{S}_s$ {mutually disjoint sublattices}
1: **for** $k = 1$ to s **do**
2: **for** every $i \in \mathcal{S}_k$ **do**
3: Use (48) to calculate cost of changing $y_i \to y_i' \in \mathcal{I}_i$
4: **end for**
5: Embed m_k bits while $\sum_i \rho_i(y_i'\mathbf{y}_{\sim i}) = D_\epsilon \times |\{c \in \mathcal{C} | c \cap \mathcal{S}_k \neq \emptyset\}|/|\mathcal{C}|$.
6: Update $\mathbf{y}_{\mathcal{S}_k}$ with new values and keep $\mathbf{y}_{\sim\mathcal{S}_k}$ unchanged.
7: **end for**
8: **return** $\mathbf{y}$ and $\sum_k m_k$ {stego image and number of bits}

processed, we say that one embedding sweep was completed. By repeating these embedding sweeps,[6] the resulting modified images will converge to a sample from π_λ.

The embedding in sublattice $\mathcal{S}_k$ will introduce embedding changes with probabilities (15), where the value of λ_k is determined by the individual distortions $\{\rho_i(y_i'\mathbf{y}_{\sim i}) | i \in \mathcal{S}_k\}$ (48) to satisfy the payload constraint of embedding m/s bits in the kth sublattice (again, e.g., using a binary search for λ_k). Because each sublattice extends over a different portion of the cover image while we split the payload evenly across the sublattices, λ_k may slightly vary with k because of variations in the individual distortions. This represents a deviation from the Gibbs sampler. Fortunately, the sublattices can often be chosen so that the image does not differ too much on every sublattice, which will guarantee that the sets of individual distortions $\{\rho_i(y_i'\mathbf{y}_{\sim i}) | i \in \mathcal{S}_k\}$ are also similar across the sublattices. Thus, with an increased number of sweeps, λ_k will converge to an approximately common value and the whole process represents a correct version of the Gibbs sampler.

In binary embedding ($\mathcal{I}_i = \{x_i^{(0)}, x_i^{(1)}\}$), note that the two distortions $\rho_i^{(0)}(x_i^{(0)}\mathbf{y}_{\sim i}) = D(x_i^{(0)}\mathbf{y}_{\eta(i)})$, $\rho_i^{(1)}(x_i^{(1)}\mathbf{y}_{\sim i}) = D(x_i^{(1)}\mathbf{y}_{\eta(i)})$ at pixel i depend on the current pixel values in its neighborhood $\eta(i)$. Therefore, both $\rho_i^{(0)}$ and $\rho_i^{(1)}$ can be non-zero at the same time and we can even have $\rho_i^{(1)} < \rho_i^{(0)}$. It is the neighborhood of i that ultimately determines whether or not it is beneficial to preserve the value of the pixel!

B. Distortion-limited sender

A similar approach can be used to implement the distortion-limited sender with a distortion limit D_ϵ. Consider a simulation of such embedding by a Gibbs sampler with the correct λ (obtained from a binary search as described in Section V-B) on the sublattice $\mathcal{S}_k \subset \mathcal{S}$. Assuming again that all sublattices have the same distortion properties, the distortion obtained from cliques containing pixels from $\mathcal{S}_k$ should be proportional to the number of such cliques. Formally,

$$E_{\pi_\lambda(\mathbf{Y}_{\mathcal{S}_k}|\mathbf{Y}_{\sim\mathcal{S}_k})}[D] = D_\epsilon \frac{|\{c \in \mathcal{C} | c \cap \mathcal{S}_k \neq \emptyset\}|}{|\mathcal{C}|}. \quad (49)$$

As described in Algorithm 3, the sender can realize this by embedding as many bits to every sublattice as possible while achieving the distortion (49). Note that we do not need to compute the partition function for every image in order to realize the embedding. Moreover, in practice when the embedding is implemented using syndrome-trellis codes [8], the search for the correct parameter λ, as described in Section V-B, is not needed either as long as the distortion properties of every sublattice are the same. This is because the codes need the local *distortion* $\rho_i(y_i'\mathbf{y}_{\sim i})$ (48) at each lattice pixel i and not the embedding probabilities. (This eliminates the need for λ.)

The issue of the minimal sufficient number of embedding sweeps for both algorithms needs to be studied specifically for each distortion measure (see the discussion in the experimental Section VIII). By replacing a specific practical embedding method with a simulator of optimal embedding, we can simulate the impact of optimal algorithms (for both senders) without having to determine the value of the parameter λ as described in Section V-B. We still need to compute λ_k for each sublattice $\mathcal{S}_k$ to obtain the probabilities of modifying each pixel (15), but this can be done as described in Section IV without having to use the Gibbs sampler or the thermodynamic integration.

Finally, we comment on how to handle wet pixels within this framework. Since we assume that the distortion is bounded ($|D(\mathbf{y})| < K$ for all $\mathbf{y} \in \mathcal{Y}$), wet pixels are handled by forcing $\mathcal{I}_i = \{x_i\}$. Because this knowledge may not be available to the decoder in practice, practical coding schemes should treat them either by setting $\rho_i(y_i) = \infty$ or to some large constant for $y_i \neq x_i$ (for details, see [8]).

C. Practical limits of the Gibbs sampler

Thanks to the bounds established in Section II, we know that the maximal payload that can be embedded in this manner is the entropy of π_λ (11). Assuming the embedding proceeds on the bound for the individual sublattices, the question is how close the total payload embedded in the image is to $H(\pi_\lambda)$. Following the Gibbs sampler, the configuration of the stego image will converge to a sample $\mathbf{y}$ from π_λ. Let us now go through one more sweep. We denote by $\mathbf{y}^{[k]}$ the stego image before starting embedding in sublattice $\mathcal{S}_k$, $k = 1, \ldots, s$. In each sublattice, the following payload is embedded:

$$H\big(\mathbf{Y}_{\mathcal{S}_k} \big| \mathbf{Y}_{\sim\mathcal{S}_k} = \mathbf{y}^{[k]}_{\sim\mathcal{S}_k}\big). \quad (50)$$

[6] After each embedding sweep, at each pixel the previous change is *erased* and the pixel is reconsidered again, just like in the Gibbs sampler.

We now use the following result from information theory. For any random variables $X_1, \dots, X_s$,

$$\sum_{k=1}^{s} H(X_k|X_{\sim k}) \le H(X_1, \dots, X_s), \tag{51}$$

with equality only when all variables are independent.[7] Thus, we will have in general

$$H^-(\mathbf{Y}) \triangleq \sum_{k=1}^{s} H\big(\mathbf{Y}_{\mathcal{S}_k}\big|\mathbf{Y}_{\sim\mathcal{S}_k} = \mathbf{y}_{\sim\mathcal{S}_k}^{[k]}\big) < H(\mathbf{Y}) = H(\pi_\lambda). \tag{52}$$

The term $H^-(\mathbf{Y})$ is recognized as the erasure entropy [34], [35] and it is equal to the conditional entropy $H(\mathbf{Y}^{(l+1)}|\mathbf{Y}^{(l)})$ (entropy rate) of the Markov process defined by our Gibbs sampler (c.f., (35)), where $\mathbf{Y}^{(l)}$ is the random variable obtained after l sweeps of the Gibbs sampler.

The erasure-entropy inequality (52) means that the embedding scheme will be suboptimal, unable to embed the maximal payload $H(\pi_\lambda)$. The actual loss can be assessed by evaluating the entropy of $H(\pi_\lambda)$, e.g., using the algorithms described in Section V. An example of such comparison is presented in Section VIII-C.

The last remaining issue is the choice of the potentials V_c. In the next section, we show one example, where V_c are chosen to tie the principle of minimal embedding distortion to the preservation of the cover-source model. We also describe a specific embedding method and subject it to experiments using blind steganalyzers.

VII. Practical embedding constructions

We are now in the position to describe a practical embedding method that uses the theory developed so far. First and foremost, the potentials V_c should measure the detectability of embedding changes. We have substantial freedom in choosing them and the design may utilize reasoning based on theoretical cover source models as well as heuristics stemming from experiments using blind steganalyzers. The proper design of potentials is a complicated subject in itself and is beyond the scope of this paper, whose main purpose is introducing a general framework rather than optimizing the design. Here, we describe a specific example of a more general approach that builds upon the latest results in steganography and steganalysis and one that gave us an opportunity to validate the proposed framework by showing an improvement over the current state of the art in Section VIII.

A. Additive approximation

As argued in the introduction, the steganography design principles based on model preservation and on minimizing distortion coincide when the distortion is defined as a norm of the difference of feature vectors used to model cover images:

$$D(\mathbf{y}) = ||f(\mathbf{x}) - f(\mathbf{y})|| \triangleq \sum_{k=1}^{d} w_k |f_k(\mathbf{x}) - f_k(\mathbf{y})|. \tag{53}$$

[7]For $k = 2$, this result follows immediately from $H(X_1|X_2) + H(X_2|X_1) = H(X_1, X_2) - I(X_1; X_2)$. The result for $s > 2$ can be obtained by induction over s.

Here, $f(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_d(\mathbf{x})) \in \mathbb{R}^d$ is a d-dimensional feature vector of image $\mathbf{x}$ and $\mathbf{w} = (w_1, \dots, w_d)$ are weights. The properties of D defined in this manner depend on the properties of the functions f_k. In general, however, D is not additive. In the past, steganographers were forced to use some *additive approximation* of D to realize the embedding in practice. A general method for turning an arbitrary distortion measure into an additive proceeds is:

$$\hat{D}(\mathbf{y}) = \sum_{i=1}^{n} D(y_i \mathbf{x}_{\sim i}). \tag{54}$$

Embedding with the additive measure $\hat{D}$ can be simulated (and realized) as explained in Section IV. The approximation, of course, ensues a capacity loss due to a mismatch in the minimized distortion function. Thanks to the methods introduced in Section V-B, this loss can now be contrasted against the rate–distortion bound for the original measure D. However, we cannot build a practical scheme unless D can be written as a sum of *local* potentials. Next, we explain how to turn D into this form using the idea of a bounding distortion.

B. Bounding distortion

Most features used in steganalysis can be written as a sum of locally-supported functions across the image

$$f_k(\mathbf{x}) = \sum_{c\in\mathcal{C}} f_c^{(k)}(\mathbf{x}), \quad k = 1, \dots, d. \tag{55}$$

For example, the kth histogram bin of image $\mathbf{x}$ can be written using the Iverson bracket as

$$h_k(\mathbf{x}) = \sum_{i\in\mathcal{S}} [x_i = k], \tag{56}$$

while the klth element of a horizontal co-occurrence matrix

$$C_{k,l}(\mathbf{x}) = \sum_{i=1}^{n_1}\sum_{j=1}^{n_2-1} [x_{i,j} = k][x_{i,j+1} = l] \tag{57}$$

is a sum over horizontally adjacent pixels (horizontal two-pixel cliques). For such locally-supported features, we can obtain an upper bound on $D(\mathbf{y}) = ||f(\mathbf{x}) - f(\mathbf{y})||$, $\mathbf{y} \in \mathcal{Y}$, that has the required form:

$$||f(\mathbf{x}) - f(\mathbf{y})|| = \sum_{k=1}^{d} w_k \left|\sum_{c\in\mathcal{C}} f_c^{(k)}(\mathbf{x}) - \sum_{c} f_c^{(k)}(\mathbf{y})\right| \tag{58}$$

$$\le \sum_{k=1}^{d} w_k \sum_{c\in\mathcal{C}} |f_c^{(k)}(\mathbf{x}) - f_c^{(k)}(\mathbf{y})| \tag{59}$$

$$= \sum_{c\in\mathcal{C}}\sum_{k=1}^{d} w_k |f_c^{(k)}(\mathbf{x}) - f_c^{(k)}(\mathbf{y})| \tag{60}$$

$$= \sum_{c\in\mathcal{C}} V_c(\mathbf{y}), \tag{61}$$

where

$$V_c(\mathbf{y}) = \sum_{k=1}^{d} w_k |f_c^{(k)}(\mathbf{x}) - f_c^{(k)}(\mathbf{y})|. \tag{62}$$

Following our convention explained in Section II, we describe the methodology for a fixed cover image $\mathbf{x}$ and thus do

not make the dependence of V_c on $\mathbf{x}$ explicit. The sum $\sum_{c\in\mathcal{C}} V_c(\mathbf{y})$ will be called the *bounding distortion*.

We now provide a specific example of this approach. The choice is motivated by our desire to work with a modern, well-established feature set so that later, in Section VIII, we can validate the usefulness of the proposed framework by constructing a high-capacity steganographic method undetectable using current state-of-the-art steganalyzer. The motivation and justification of the feature set appears in [23]. It is a slight modification of the SPAM set [22], which is the basis of the current most reliable blind steganalyzer in the spatial domain. The features are constructed by considering the differences between neighboring pixels (e.g., horizontally adjacent pixels) as a higher-order Markov chain and taking the sample joint probability matrix (co-occurrence matrix) as the feature. The advantage of using the joint matrix instead of the transition probability matrix is that the norm of the feature difference can be readily upper-bounded by the desired local form (62).

To formally define the feature for an $n_1 \times n_2$ image $\mathbf{x}$, let us consider the following co-occurrence matrix computed from horizontal pixel differences $D^{\rightarrow}_{i,j}(\mathbf{x}) = x_{i,j+1} - x_{i,j}$, $i = 1,\ldots,n_1, j = 1,\ldots,n_2-1$:

$$A^{\rightarrow}_{k,l}(\mathbf{x}) = \frac{1}{n_1(n_2-2)}\sum_{i=1}^{n_1}\sum_{j=1}^{n_2-2}[(D^{\rightarrow}_{i,j}, D^{\rightarrow}_{i,j+1})(\mathbf{x}) = (k,l)]. \tag{63}$$

For compactness, in (63) we abbreviated the argument of the Iverson bracket from $D^{\rightarrow}_{i,j}(\mathbf{x}) = k \,\&\, D^{\rightarrow}_{i,j+1}(\mathbf{x}) = l$ to $(D^{\rightarrow}_{i,j}, D^{\rightarrow}_{i,j+1})(\mathbf{x}) = (k,l)$. Clearly, $A^{\rightarrow}_{i,j}(\mathbf{x})$ is the normalized count of neighboring triples of pixels $\{x_{i,j}, x_{i,j+1}, x_{i,j+2}\}$ with differences $x_{i,j+1} - x_{i,j} = k$ and $x_{i,j+2} - x_{i,j+1} = l$ in the entire image. The superscript arrow "$\rightarrow$" denotes the fact that the differences are computed by subtracting the left pixel from the right one. Similarly,

$$A^{\leftarrow}_{k,l}(\mathbf{x}) = \frac{1}{n_1(n_2-2)}\sum_{i=1}^{n_1}\sum_{j=3}^{n_2}[(D^{\leftarrow}_{i,j}, D^{\leftarrow}_{i,j-1})(\mathbf{x}) = (k,l)] \tag{64}$$

with $D^{\leftarrow}_{i,j}(\mathbf{x}) = x_{i,j-1} - x_{i,j}$. By analogy, we can define vertical, diagonal, and minor diagonal matrices $A^{\downarrow}_{k,l}$, $A^{\uparrow}_{k,l}$, $A^{\nearrow}_{k,l}$, $A^{\swarrow}_{k,l}$, $A^{\searrow}_{k,l}$, $A^{\nwarrow}_{k,l}$. All eight matrices are sample joint probabilities of observing the differences k and l between three consecutive pixels along a certain direction. Due to the antisymmetry $D^{\rightarrow}_{i,j}(\mathbf{x}) = -D^{\leftarrow}_{i,j+1}(\mathbf{x})$ only $A^{\rightarrow}_{k,l}$, $A^{\nearrow}_{k,l}$, $A^{\uparrow}_{k,l}$, $A^{\nwarrow}_{k,l}$ are needed since $A^{\rightarrow}_{k,l} = A^{\leftarrow}_{-l,-k}$, and similarly for other matrices.

Because neighboring pixels in natural images are strongly dependent, each matrix exhibits a sharp peak around $(k,l) = (0,0)$ and then quickly falls off with increasing k and l. When such matrices are used for steganalysis [22], they are truncated to a small range, such as $-T \le k,l \le T$, $T = 4$, to prevent the onset of the "curse of dimensionality." On the other hand, in steganography we can use large-dimensional models ($T = 255$) because it is easier to preserve a model than to learn it.[8] Another reason for using a high-dimensional feature space is to avoid "overtraining" the embedding algorithm to a low-dimensional model as such algorithms may become detectable by a slightly modified feature set, an effect already reported in the DCT domain [19].

[8]Similar reasoning for constructing the distortion function was used in the HUGO algorithm [23].

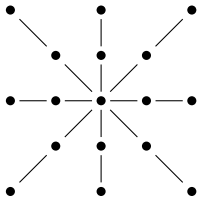

Figure 5. The union of all 12 cliques consisting of three pixels arranged in a straight line in the 5×5 square neighborhood.

By embedding a message, $A^{\rightarrow}_{k,l}(\mathbf{x})$ is modified to $A^{\rightarrow}_{k,l}(\mathbf{y})$. The differences between the features will thus serve as a measure of embedding impact closely tied to the model (the indices i and j run from 1 to n_1 and n_2-2, respectively):

$$\begin{aligned}
&|A^{\rightarrow}_{k,l}(\mathbf{y}) - A^{\rightarrow}_{k,l}(\mathbf{x})| = && (65)\\
&\quad = \frac{1}{n_1(n_2-2)}\Big|\sum_{i,j}[(D^{\rightarrow}_{i,j}, D^{\rightarrow}_{i,j+1})(\mathbf{y}) = (k,l)] && (66)\\
&\qquad - [(D^{\rightarrow}_{i,j}, D^{\rightarrow}_{i,j+1})(\mathbf{x}) = (k,l)]\Big| && (67)\\
&\quad \le \frac{1}{n_1(n_2-2)}\sum_{i,j}\big|[(D^{\rightarrow}_{i,j}, D^{\rightarrow}_{i,j+1})(\mathbf{y}) = (k,l)] && (68)\\
&\qquad - [(D^{\rightarrow}_{i,j}, D^{\rightarrow}_{i,j+1}(\mathbf{x}) = (k,l)]\big| && (69)\\
&\quad = \sum_{c\in\mathcal{C}^{\rightarrow}} H_c^{(k,l)\rightarrow}(\mathbf{y}), && (70)
\end{aligned}$$

where we defined the following locally-supported functions

$$\begin{aligned}
H_c^{(k,l)\rightarrow}(\mathbf{y}) &= \frac{1}{n_1(n_2-2)}\\
&\cdot\Big|[(D^{\rightarrow}_{i,j}, D^{\rightarrow}_{i,j+1})(\mathbf{y}) = (k,l)] - [(D^{\rightarrow}_{i,j}, D^{\rightarrow}_{i,j+1})(\mathbf{x}) = (k,l)]\Big|
\end{aligned} \tag{71}$$

on all horizontal cliques $\mathcal{C}^{\rightarrow} = \{c | c = \{(i,j),(i,j+1),(i,j+2)\}\}$. Notice that the absolute value had to be pulled into the sum to give the potentials a small support. Again, we drop the symbol for the cover image, $\mathbf{x}$, from the argument of $H_c^{(k,l)}$ for the same reason why we do not make the dependence on $\mathbf{x}$ explicit for all other variables, sets, and functions.

Since the other three matrices can be written in this manner as well, we can write the distortion function in the following final form

$$D(\mathbf{y}) = \sum_{c\in\mathcal{C}} V_c(\mathbf{y}), \tag{72}$$

now with $\mathcal{C} = \mathcal{C}^{\rightarrow} \cup \mathcal{C}^{\nearrow} \cup \mathcal{C}^{\uparrow} \cup \mathcal{C}^{\nwarrow}$, the set of three-pixel cliques along all four directions, and

$$V_c(\mathbf{y}) = \sum_{k,l} w_{k,l} H_c^{(k,l)\rightarrow}(\mathbf{y}), \text{ for each clique } c \in \mathcal{C}^{\rightarrow}, \tag{73}$$

and similarly for the other three clique types. Notice that we again introduced weights $w_{k,l} > 0$ into the definition of V_c so that we can adjust them according to how sensitive steganalysis is to the individual differences. For example, if we

observe that a certain difference pair (k,l) varies significantly over cover images, by assigning it a smaller weight we allow it to be modified more often, while those differences that are stable across covers but sensitive to embedding should be intuitively assigned a larger value so that the embedding does not modify them too much.

To complete the picture, the neighborhood system here is formed by 5×5 neighborhoods and thus the index set can be decomposed into nine disjoint sublattices $\mathcal{S} = \bigcup_{ab} \mathcal{S}_{ab}$, $1 \le a, b \le 3$,

$$\mathcal{S}_{ab} = \{(a+3k, b+3l) | 1 \le a+3k \le n_1, 1 \le b+3l \le n_2\}. \tag{74}$$

To better explain the effect of embedding changes on the distortion, realize that each pixel belongs to three horizontal, three vertical, three diagonal, and three minor-diagonal cliques. When a single pixel $x_{i,j}$ is changed, it affects only the 12 potentials whose clique contains $x_{i,j}$. Let us say that the original pixel values $c_0 = \{x_{i,j}, x_{i,j+1}, x_{i,j+2}\}$ had differences k, l, and the pixel value changed from $x_{i,j}$ to $y_{i,j} = x_{i,j} + 1$. Then, the pixel differences will be modified to $k-1, l$. Considering just the contribution from $H_{c_0}^{(k,l)\rightarrow}$ to the potential V_{c_0} (73), it will increase by the sum of $w_{k,l}$ (the pair k,l is leaving cover) and $w_{k-1,l}$ (a new pair appears in the stego image).

C. Other options

The framework presented in this paper allows the sender to formulate the local potentials directly instead of obtaining them as the bounding distortion. For example, the cliques and their potentials may be determined by the local image content or by learning the cover source using the method of fields of experts [26]. The merit of these possibilities can be evaluated by steganalyzers trained on a large set of images. The important question of optimizing the local potential functions w.r.t. statistical detectability is an important direction the authors intend to explore in the future.

VIII. Experiments

In this section, we validate the proposed framework experimentally and include a comparison between simple steganographic algorithms, such as binary and ternary ± 1 embedding and steganography implemented via the bounding distortion and the additive approximation (54). For the case of the bounding distortion, the capacity loss w.r.t. the optimal payload given by $H(\pi_\lambda)$ is evaluated by means of the thermodynamic integration algorithm from Section V-B.

A. Tested embedding methods

For the methods based on additive approximation and the bounding distortion, we used as a feature vector the joint probability matrix $A^{\rightarrow}_{k,l,m}(\mathbf{x})$ defined similarly as in (63) with the difference vector computed from *four* consecutive pixels $(D^{\rightarrow}_{i,j}, D^{\rightarrow}_{i,j+1}, D^{\rightarrow}_{i,j+2}) = (k,l,m)$. As above, four such matrices corresponding to four spatial directions were computed. The matrices were used at their full size $T = 255$ leading to model dimensionality $d = 4 \times 511^3 \approx 5 \cdot 10^8$.

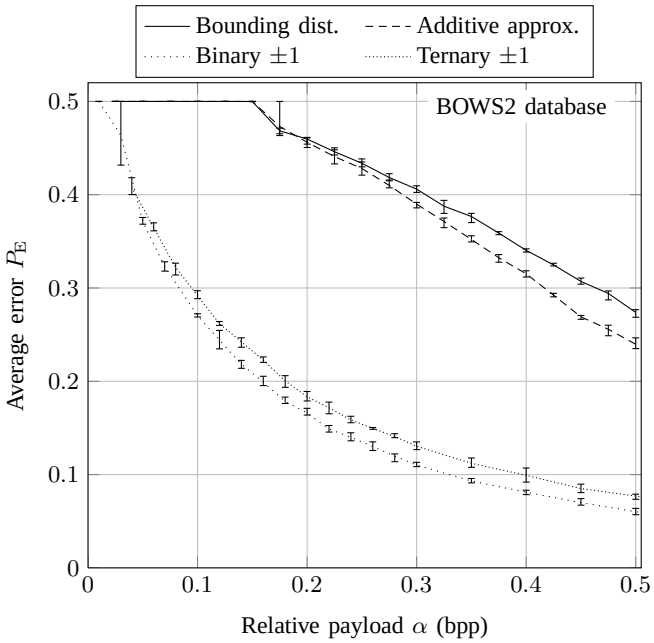

Figure 6. Comparison of ± 1 embedding with optimal binary and ternary coding with binary embedding algorithms based on the Gibbs construction with a bounding distortion and the additive approximation as described in Section VIII-A. The error bars depict the minimum and maximum steganalyzer error P_E (76) over five runs of SVM classifiers with different division of images into training and testing set.

The weights were chosen to be small for those triples $(D^{\rightarrow}_{i,j}, D^{\rightarrow}_{i,j+1}, D^{\rightarrow}_{i,j+2}) = (k,l,m)$ that occur infrequently in images and large for frequented triples. Following the recommendation described in [23], since the frequency of occurrence of the triples falls off quickly with their norm, we choose the weights as

$$w_{k,l,m} = \left(\sigma + \sqrt{k^2+l^2+m^2}\right)^{-\theta}, \tag{75}$$

with $\theta = 1$ and $\sigma = 1$. The purpose of the weights is to force the embedding algorithm to modify those parts of the model that are difficult to model accurately, forcing thus the steganalyst to use a more accurate model. Here, the advantage goes to the steganographer, because preserving a high-dimensional feature vector is more feasible than accurately modeling it.

Because the neighborhood $\eta(i)$ in this case contains 7×7 pixels, the image was divided into 16 square sublattices on which embedding was carried out independently. We tested binary embedding, $\mathcal{I}_i = \{x_i, x_i'\}$, where x_i' was selected randomly and uniformly from $\{x_i - 1, x_i + 1\}$ and then fixed for all experiments with cover $\mathbf{x}$. The payload-limited sender was simulated using the Gibbs sampler constrained to only two sweeps. Increasing the number of sweeps did not lead to further improvement. The curiously low number of sweeps sufficient to properly implement the Gibbs sampler is most likely due to the fact that the dependencies dictated by the bounding distortion are rather weak. The simulation of embedding for one image took less than 5 seconds when implemented in C++ on a single-processor PC.

To summarize, the following four steganographic methods were tested:

1) Binary embedding using the Gibbs construction with sets $\mathcal{I}_i = \{x_i, x_i'\}$ and bounding distortion (72) of (53) with weights (75) for the $d = 4 \times 511^3$-dimensional feature space given by matrices $A^{\rightarrow}_{k,l,m}$, $A^{\nearrow}_{k,l,m}$, $A^{\uparrow}_{k,l,m}$, $A^{\nwarrow}_{k,l,m}$.
2) Additive approximation (54) of (53) for the same sets $\mathcal{I}_i$, feature space, and norm as in 1).
3) Binary ± 1 embedding with the same sets $\mathcal{I}_i$ equipped with a matrix embedding scheme operating on the binary bound.
4) Ternary ± 1 embedding with $\mathcal{I}_i = \{x_i - 1, x_i, x_i + 1\}$ equipped with a ternary matrix embedding scheme operating on the ternary bound (the bounds appear, e.g., in [9]).

We note that practical near-optimal codes for the two ± 1 embedding methods can be found in [10] and [39].

B. Testing methodology and final results

Following the separation principle, we study the security of all schemes when operating on the rate–distortion bound. All tests were carried out on the BOWS2 database [1] containing approximately 10800 grayscale images with a fixed size of 512×512 pixels coming from rescaled and cropped natural images of various sizes. Steganalysis was implemented using the second-order SPAM feature set with $T = 3$ [22]. The image database was evenly divided into a training and a testing set of cover and stego images, respectively. A soft-margin support-vector machine was trained using the Gaussian kernel. The kernel width and the penalty parameter were determined using five-fold cross validation on the grid $(C, \gamma) \in \{(10^k, 2^j) | k \in \{-3, \dots, 4\}, j \in \{-L-3, \dots, -L+3\}\}$, where $L = \log_2 d$ is the binary logarithm of the number of features.

We report the results using a measure frequently used in steganalysis – the minimum average classification error

$$P_{\mathrm{E}} = (P_{\mathrm{FA}} + P_{\mathrm{MD}})/2, \tag{76}$$

where P_{FA} and P_{MD} are the false-alarm and missed-detection probabilities. Smaller values of P_{E} correspond to better steganalysis and thus larger statistical detectability (lower security).

Figure 6 displays the comparison of all four embedding methods listed above. The methods based on the the bounding distortion and the additive approximation (denoted as "Bounding dist." and "Additive approx.") are completely undetectable for payloads smaller than 0.15 bpp, which suggests that the embedding changes are made in pixels not covered by the SPAM features. Since both schemes are binary with $\mathcal{I}_i = \{x_i, x_i'\}$ with x_i' randomly chosen from $\{x_i - 1, x_i + 1\}$, they become equivalent to simple binary ± 1 embedding (Method 3) as $\alpha \to 1$ and thus become detectable. Comparing the capacity, both schemes allow communicating ten times larger payloads with $P_{\mathrm{E}} = 40\%$ as compared to ternary ± 1 embedding. The advantage of using the Gibbs sampler with the bounding distortion over the additive approximation becomes more evident for larger payloads, where the embedding changes start to interact. This confirms our expectation that in this range the additive approximation is unable to cope with the interactions among changes and thus its detectability increases. This result, however, may change for different distortion measures and cover sources. The fact that the Gibbs sampler with bounding distortion did not bring a substantial performance improvement over the additive approximation indicates that the interactions among embedding changes are in general quite weak (at least as far as they are captured by the bounding distortion). The low strength of interactions also explains why only two sweeps of the Gibbs sampler were sufficient in practice.

C. Analysis of upper bounds

As described in Section VI-C, Algorithm 2 for the payload-limited sender is unable to embed the optimal payload of $H(\pi_\lambda)$ for three reasons. The performance may be affected by the small number of sweeps of the Gibbs sampler, the parameter λ may vary slightly among the sublattices, and the algorithm embeds the erasure entropy $H^-(\pi_\lambda) \le H(\pi_\lambda)$. The combined effect of these factors is of great importance for practitioners and is evaluated below for two images using the Gibbs sampler and the thermodynamic integration as explained in Section V-B.

Since the Gibbs construction depends on the cover image $\mathbf{x}$, we present the results for two grayscale images of size 512×512 pixels coming from two different sources. The test image "0.png" is from the BOWS2 database and "Lenna" was obtained from http://en.wikipedia.org/wiki/File:Lenna.png and converted to grayscale using GNU Image Manipulation Program (GIMP). In both cases, we used the same sets $\mathcal{I}_i$ and the same feature set as in the previous section with the bounding distortion with weight parameters $\sigma = 1$ and $\theta = 1$.

The image "0.png" contains more areas with edges and textures than "Lenna" and thus for small distortions, it offers a larger capacity than "Lenna" because the weights (75) around edges and complex texture are small. This is apparent from the slopes of the rate–distortion bounds in Figure 7.

The same figure compares the rate–distortion performance of the payload-limited sender simulated by the Gibbs sampler with only two sweeps as described in Algorithm 2. For a given payload, the distortion was obtained as an average over 100 random messages. The comparison shows that the payload loss of Algorithm 2 to the optimal $H(\pi_\lambda)$ is quite small. Note that the erasure entropy, $H^-(\pi_\lambda)$, plotted in the figure has been computed over the sublattices after two sweeps and thus already contains the impact of all three factors discussed at the beginning of this section.

IX. Conclusion

Currently, the most successful principle for designing practical steganographic systems that embed in empirical covers is based on minimizing a suitably defined distortion measure. Implementation difficulties and a lack of practical embedding methods have so far limited the application of this principle to a rather special class of distortion measures that are additive over pixels. With the development of near-optimal low-complexity coding schemes, such as the syndrome-trellis codes [8], this direction has essentially reached its limits. It is

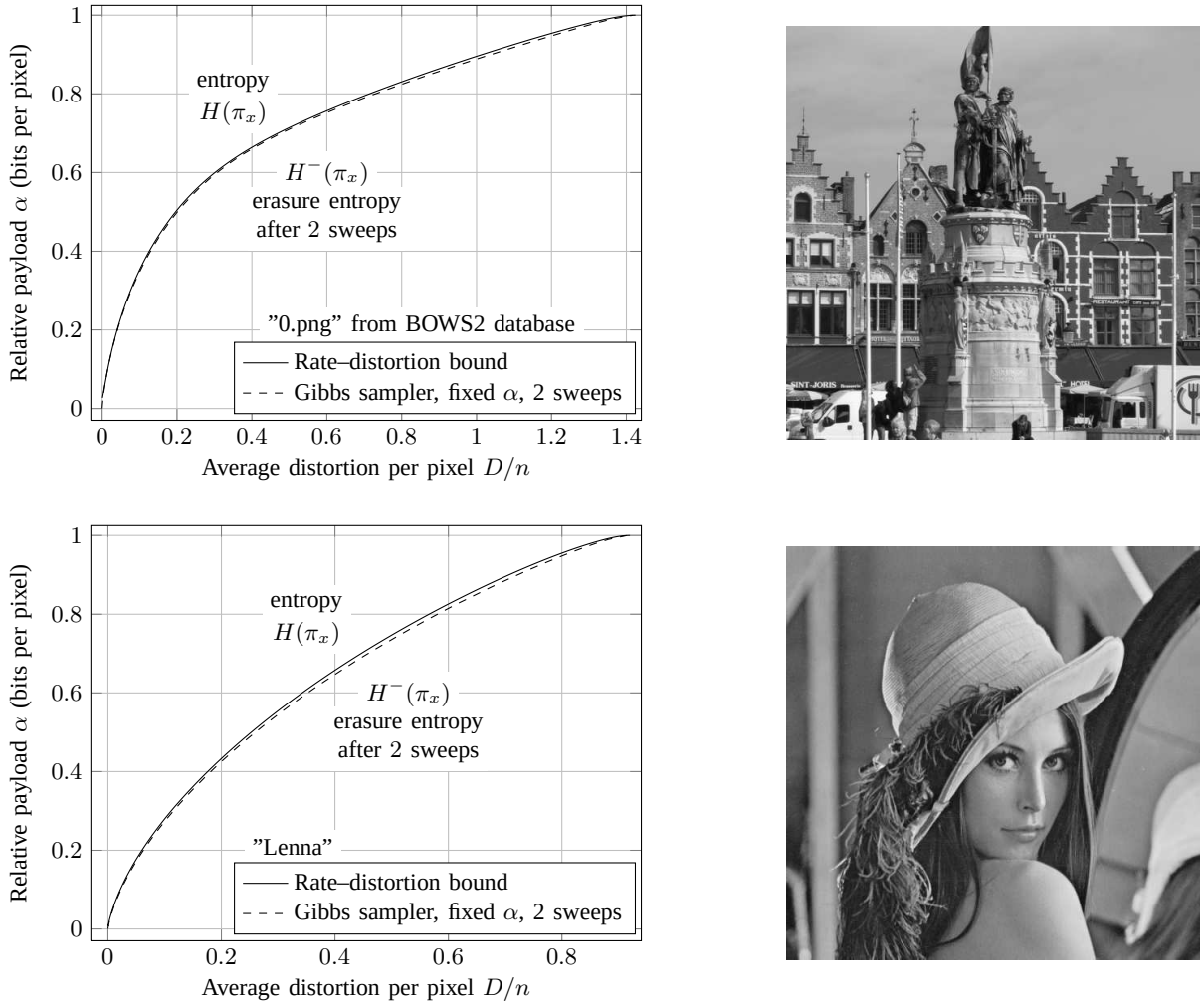

Figure 7. Comparison of the payload loss of Algorithm 2 for cover images "0.png" and "Lenna" shown on the right. The rate–distortion bounds were obtained using the Gibbs sampler (38) and the thermodynamic integration (40).

our firm belief that further substantial increase in secure payload is possible only when the sender uses adaptive schemes that place embedding changes based on the local content, that dare to modify pixels in some regions by more than 1, and that consider interactions among embedding changes while preserving higher-order statistics among pixels. This paper is an important step in this direction.

We offer the steganographer a complete methodology for embedding while minimizing an arbitrarily defined distortion measure D. The absence of any restrictions on D means that the remaining task left to the sender is to find a distortion measure that correlates with statistical detectability. An appealing possibility is to define D as a weighted norm of the difference between cover and stego feature vectors used in steganalysis. This immediately connects the principle of minimum-distortion steganography with the concept of model preservation which has so far been limited to low-dimensional models. Being able to preserve a large-dimensional model gives the steganographer a great advantage over the steganalyst because of the difficulties associated with learning a high-dimensional cover source model using statistical learning tools.

The proposed framework is called the Gibbs construction and it connects steganography with statistical physics, which contributed with many practical algorithms. In particular, the Gibbs sampler combined with the thermodynamic integration can be used to derive the rate–distortion bound, simulate the impact of optimal embedding, and realize near-optimal embedding algorithms. These three tasks can be addressed separately (the so-called "separation principle") giving the sender a great amount of design flexibility as well as control over losses of practical schemes.

An important case elaborated in this paper corresponds to D defined as a sum of local potentials over small pixel neighborhoods. Here, the optimal distribution of embedding modifications reduces to a Markov random field and the Gibbs sampler can be turned into a practical embedding algorithm able to consider dependencies among embedding changes. When D cannot be written as a sum of local potentials, practical (suboptimal) methods can be realized by approximating D either with an additive distortion measure or with local potentials. The problem of finding the best approximation for a given non-local D is of its own interest. We did not cover the task of minimizing the statistical detectability with

respect to the distortion function completely due to its inherent complexity; it is left as part of our future effort.

We described the proposed methodology both for a payload-limited sender and the distortion-limited sender. The former embeds a fixed payload in every image with minimal distortion, while the latter embeds the maximal payload for a given distortion in every image. The distortion-limited sender better corresponds to our intuition that, for a fixed statistical detectability, more textured or noisy images can carry a larger secure payload than smoother or simpler images. The fact that the size of the hidden message is driven by the cover image essentially represents a more realistic case of the batch steganography paradigm [17]. We postpone the study of the distortion-limited sender to our future effort.

Note that the distortion measure is used only by the sender and thus does not need to be shared. The only information needed by the receiver to decode the message is its size which can be communicated separately in the same cover image. This opens up the intriguing possibility to develop embedding schemes able to learn the proper distortion function while observing the impact of embedding on the cover source.

Finally, the proposed methodology can be applied to other data hiding problems where the statistical detectability constraint could be replaced by a perceptual distortion constraint.

The source code used for all experiments in this paper can be found at http://dde.binghamton.edu/download/gibbs.

REFERENCES

[1] P. Bas and T. Furon. BOWS-2. http://bows2.gipsa-lab.inpg.fr, July 2007.

[2] R. Böhme and A. Westfeld. Breaking Cauchy model-based JPEG steganography with first order statistics. In P. Samarati, P. Y. A. Ryan, D. Gollmann, and R. Molva, editors, *Computer Security - ESORICS 2004. Proceedings 9th European Symposium on Research in Computer Security*, volume 3193 of Lecture Notes in Computer Science, pages 125–140, Sophia Antipolis, France, September 13–15, 2004. Springer, Berlin.

[3] G. Cancelli and M. Barni. MPSteg-color: A new steganographic technique for color images. In T. Furon, F. Cayre, G. Doërr, and P. Bas, editors, *Information Hiding, 9th International Workshop*, volume 4567 of Lecture Notes in Computer Science, pages 1–15, Saint Malo, France, June 11–13, 2007. Springer-Verlag, Berlin.

[4] C. Chen and Y. Q. Shi. JPEG image steganalysis utilizing both intrablock and interblock correlations. In *Circuits and Systems, 2008. ISCAS 2008. IEEE International Symposium on*, pages 3029–3032, May 2008.

[5] M. K. Cowles and B. P. Carlin. Markov chain Monte Carlo convergence diagnostics: A comparative review. *Journal of the American Statistical Association*, 91(434):883–904, June 1996.

[6] R. Crandall. Some notes on steganography. *Steganography Mailing List*, available from http://os.inf.tu-dresden.de/~westfeld/crandall.pdf, 1998.

[7] T. Filler, J. Judas, and J. Fridrich. Minimizing additive distortion in steganography using syndrome-trellis codes. *IEEE Transactions on Information Forensics and Security*, 2010. Under preparation.

[8] T. Filler, J. Judas, and J. Fridrich. Minimizing embedding impact in steganography using trellis-coded quantization. In N. D. Memon, E. J. Delp, P. W. Wong, and J. Dittmann, editors, *Proceedings SPIE, Electronic Imaging, Security and Forensics of Multimedia XII*, volume 7541, pages 05–01–05–14, San Jose, CA, January 17–21, 2010.

[9] J. Fridrich. *Steganography in Digital Media: Principles, Algorithms, and Applications*. Cambridge University Press, 2009.

[10] J. Fridrich and T. Filler. Practical methods for minimizing embedding impact in steganography. In E. J. Delp and P. W. Wong, editors, *Proceedings SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents IX*, volume 6505, pages 02–03, San Jose, CA, January 29–February 1, 2007.

[11] J. Fridrich, M. Goljan, and D. Soukal. Perturbed quantization steganography. *ACM Multimedia System Journal*, 11(2):98–107, 2005.

[12] J. Fridrich, M. Goljan, D. Soukal, and P. Lisoněk. Writing on wet paper. In E. J. Delp and P. W. Wong, editors, *Proceedings SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents VII*, volume 5681, pages 328–340, San Jose, CA, January 16–20, 2005.

[13] J. Fridrich, T. Pevný, and J. Kodovský. Statistically undetectable JPEG steganography: Dead ends, challenges, and opportunities. In J. Dittmann and J. Fridrich, editors, *Proceedings of the 9th ACM Multimedia & Security Workshop*, pages 3–14, Dallas, TX, September 20–21, 2007.

[14] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741, November 1984.

[15] S. Hetzl and P. Mutzel. A graph-theoretic approach to steganography. In J. Dittmann, S. Katzenbeisser, and A. Uhl, editors, *Communications and Multimedia Security, 9th IFIP TC-6 TC-11 International Conference, CMS 2005*, volume 3677 of Lecture Notes in Computer Science, pages 119–128, Salzburg, Austria, September 19–21, 2005.

[16] A. D. Ker. Steganalysis of LSB matching in grayscale images. *IEEE Signal Processing Letters*, 12(6):441–444, June 2005.

[17] A. D. Ker. Batch steganography and pooled steganalysis. In J. L. Camenisch, C. S. Collberg, N. F. Johnson, and P. Sallee, editors, *Information Hiding, 8th International Workshop*, volume 4437 of Lecture Notes in Computer Science, pages 265–281, Alexandria, VA, July 10–12, 2006. Springer-Verlag, New York.

[18] Y. Kim, Z. Duric, and D. Richards. Modified matrix encoding technique for minimal distortion steganography. In J. L. Camenisch, C. S. Collberg, N. F. Johnson, and P. Sallee, editors, *Information Hiding, 8th International Workshop*, volume 4437 of Lecture Notes in Computer Science, pages 314–327, Alexandria, VA, July 10–12, 2006. Springer-Verlag, New York.

[19] J. Kodovský and J. Fridrich. On completeness of feature spaces in blind steganalysis. In A. D. Ker, J. Dittmann, and J. Fridrich, editors, *Proceedings of the 10th ACM Multimedia & Security Workshop*, pages 123–132, Oxford, UK, September 22–23, 2008.

[20] J. Kodovský, T. Pevný, and J. Fridrich. Modern steganalysis can detect YASS. In N. D. Memon, E. J. Delp, P. W. Wong, and J. Dittmann, editors, *Proceedings SPIE, Electronic Imaging, Security and Forensics of Multimedia XII*, volume 7541, pages 02–01–02–11, San Jose, CA, January 17–21, 2010.

[21] R. M. Neal. Probabilistic inference using Markov chain Monte Carlo methods. Technical Report CRG-TR-93-1, University of Toronto, September 25 1993.

[22] T. Pevný, P. Bas, and J. Fridrich. Steganalysis by subtractive pixel adjacency matrix. In J. Dittmann, S. Craver, and J. Fridrich, editors, *Proceedings of the 11th ACM Multimedia & Security Workshop*, pages 75–84, Princeton, NJ, September 7–8, 2009.

[23] T. Pevný, T. Filler, and P. Bas. Using high-dimensional image models to perform highly undetectable steganography. In R. Böhme and R. Safavi-Naini, editors, *Information Hiding, 12th International Workshop*, Lecture Notes in Computer Science, Calgary, Canada, June 28–30, 2010. Springer-Verlag, New York.

[24] T. Pevný and J. Fridrich. Merging Markov and DCT features for multi-class JPEG steganalysis. In E. J. Delp and P. W. Wong, editors, *Proceedings SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents IX*, volume 6505, pages 3 1–3 14, San Jose, CA, January 29–February 1, 2007.

[25] N. Provos. Defending against statistical steganalysis. In *10th USENIX Security Symposium*, pages 323–335, Washington, DC, August 13–17, 2001.

[26] S. Roth and M. J. Black. Fields of experts. *International Journal of Computer Vision*, 82(2):205–229, January 2009.

[27] V. Sachnev, H. J. Kim, and R. Zhang. Less detectable JPEG steganography method based on heuristic optimization and BCH syndrome coding. In J. Dittmann, S. Craver, and J. Fridrich, editors, *Proceedings of the 11th ACM Multimedia & Security Workshop*, pages 131–140, Princeton, NJ, September 7–8, 2009.

[28] P. Sallee. Model-based methods for steganography and steganalysis. *International Journal of Image Graphics*, 5(1):167–190, 2005.

[29] A. Sarkar, L. Nataraj, B. S. Manjunath, and U. Madhow. Estimation of optimum coding redundancy and frequency domain analysis of attacks for YASS - a randomized block based hiding scheme. In *Proceedings IEEE, International Conference on Image Processing, ICIP 2008*, pages 1292–1295, San Diego, CA, October 12–15, 2008.

[30] A. Sarkar, K. Solanki, U. Madhow, and B. S. Manjunath. Secure steganography: Statistical restoration of the second order dependencies for improved security. In *Proceedings IEEE, International Conference*

on Acoustics, Speech, and Signal Processing, volume 2, pages II–277–II–280, April 15–20, 2007.

[31] Y. Q. Shi, C. Chen, and W. Chen. A Markov process based approach to effective attacking JPEG steganography. In J. L. Camenisch, C. S. Collberg, N. F. Johnson, and P. Sallee, editors, *Information Hiding, 8th International Workshop*, volume 4437 of Lecture Notes in Computer Science, pages 249–264, Alexandria, VA, July 10–12, 2006. Springer-Verlag, New York.

[32] K. Solanki, A. Sarkar, and B. S. Manjunath. YASS: Yet another steganographic scheme that resists blind steganalysis. In T. Furon, F. Cayre, G. Doërr, and P. Bas, editors, *Information Hiding, 9th International Workshop*, volume 4567 of Lecture Notes in Computer Science, pages 16–31, Saint Malo, France, June 11–13, 2007. Springer-Verlag, New York.

[33] K. Solanki, K. Sullivan, U. Madhow, B. S. Manjunath, and S. Chandrasekaran. Provably secure steganography: Achieving zero K–L divergence using statistical restoration. In *Proceedings IEEE, International Conference on Image Processing, ICIP 2006*, pages 125–128, Atlanta, GA, October 8–11, 2006.

[34] S. Verdú and T. Weissman. Erasure entropy. In *Proc. of ISIT*, Seattle, WA, July 9–14, 2006.

[35] S. Verdú and T. Weissman. The information lost in erasures. *IEEE Transactions on Information Theory*, 54(11):5030–5058, November 2008.

[36] F. Wang and D. P. Landau. Determining the density of states for classical statistical models: A random walk algorithm to produce a flat histogram. *Phys. Rev. E*, 64(5):056101, 2001. arXiv:cond-mat/0107006v1.

[37] A. Westfeld and R. Böhme. Exploiting preserved statistics for steganalysis. In J. Fridrich, editor, *Information Hiding, 6th International Workshop*, volume 3200 of Lecture Notes in Computer Science, pages 82–96, Toronto, Canada, May 23–25, 2004. Springer-Verlag, Berlin.

[38] G. Winkler. *Image Analysis, Random Fields and Markov Chain Monte Carlo Methods: A Mathematical Introduction (Stochastic Modelling and Applied Probability)*. Springer-Verlag, Berlin Heidelberg, 2nd edition, 2003.

[39] X. Zhang, W. Zhang, and S. Wang. Efficient double-layered steganographic embedding. *Electronics Letters*, 43:482–483, April 2007.