Steganography using Gibbs random fields

Tomáš Filler
SUNY Binghamton
Department of ECE
Binghamton, NY 13902-6000
tomas.filler@binghamton.edu

Jessica Fridrich
SUNY Binghamton
Department of ECE
Binghamton, NY 13902-6000
fridrich@binghamton.edu

ABSTRACT

Many steganographic algorithms for empirical covers are designed to minimize embedding distortion. In this work, we provide a general framework and practical methods for embedding with an arbitrary distortion function that does not have to be additive over pixels and thus can consider interactions among embedding changes. The framework evolves naturally from a parallel made between steganography and statistical physics. The Gibbs sampler is the key tool for simulating the impact of optimal embedding and for constructing practical embedding algorithms. The proposed framework reduces the design of secure steganography in empirical covers to the problem of finding suitable local potentials for the distortion function that correlate with statistical detectability in practice. We work out the proposed methodology in detail for a specific choice of the distortion function and validate the approach through experiments.

Categories and Subject Descriptors

I.4.9 [**Computing Methodologies**]: Image Processing and Computer Vision—*Applications*

General Terms

Security, Algorithms, Theory

Keywords

Steganography, embedding impact, Markov random field, Gibbs sampling

1. INTRODUCTION

Currently, the most successful principle for designing practical steganographic systems that embed in empirical covers, such as digital images, is based on minimizing a suitably defined distortion measure [10, 16, 18, 24]. 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 individual cover elements. With the development of near-optimal low-complexity coding schemes, such as the syndrome-trellis codes [7], this direction has essentially reached its limits. It is our firm belief that further substantial increase in secure payload is possible only when the sender leverages 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 a step in this direction.

The need for proper models of interactions among embedding changes and their incorporation in steganography is already apparent in prior art. Aided with an overcomplete decomposition of images, the creators of MPSteg [2] embed messages by replacing small blocks with other blocks to better preserve dependencies among neighboring pixels. The YASS algorithm [26] testifies to the fact that a high embedding distortion may not necessarily result in high statistical detectability, an unusual property that can most likely be attributed to the fact that the embedding modifications are content driven and mutually correlated. Both MPSteg and YASS are heuristic in nature and leave many important issues unanswered, including contrasting the methods' performance with theoretical bounds and creating a methodology for achieving near-optimal performance.

We offer the steganographer a complete framework for embedding while minimizing an arbitrarily defined distortion measure D. This includes algorithms for computing the rate–distortion bound and simulating the impact of optimal schemes. 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. The sender can, for example, let the cover content drive the embedding (adaptive steganography) while appropriately modeling dependencies among embedding changes. An appealing possibility discussed in this paper is to define D as a weighted norm of the difference between cover and stego feature vectors, such as those used in modern blind steganalysis. Because the feature space can be viewed as a cover model, this immediately connects minimum-distortion steganography with the model-preserving approach [12, 25, 27, 29]. In this case the distortion is far from being an additive function over the pixels because the features may contain higher-order statistics, such as sample transition probability matrices of pixels or DCT coefficients modeled as Markov chains [3, 20, 22, 28]. Since no practical embedding schemes exist that minimize non-additive distortion, in the past authors worked with its additive approximation and applied a model-correction step [17, 21]. The framework proposed here allows us to evaluate the loss introduced by such approximations and it offers other more effective and theoretically well-founded options to the steganographer.

In Section 2, we show that a steganographic method that minimizes embedding distortion must make embedding changes that follow a particular form of Gibbs distribution. Here, we also establish terminology and make connections between steganography and statistical physics. In Section 3, we introduce the so-called separation principle, which includes several distinct tasks that must be addressed when developing a practical steganographic method designed to minimize distortion. In the special case when the embedding distortion can be expressed as a sum of distortions at individual pixels, the design of near-optimal embedding algorithms has been successfully resolved in the past and is summarized in Section 3.1. Continuing with the case of a general distortion function, in Section 4 we describe a useful tool for steganographers – the Gibbs sampler, which can be used to simulate the impact of optimal embedding, compute the rate–distortion bound, and construct practical steganographic schemes (in Sections 5 and 6). Construction of practical embedding algorithms begins in Section 5, where we study distortion functions that can be written as a sum of local potentials defined on cliques. At the same time, we make a connection between the potentials and image models used in blind steganalysis. In Section 6, we discuss various options the new framework offers to the steganographer, describe a specific embedding algorithm, and compare its performance with selected prior art on two image databases. Finally, the paper is concluded in Section 8.

This paper is a workshop version of a journal submission [5]. The main difference between these two versions is a more extensive experimental validation of the approach by including results from different image databases, comparison of simulated embedding with a larger amplitude, and an experiment with a distortion-limited sender. Since this version is more oriented towards practical embedding schemes, some results, such as the computing the rate–distortion bounds, were omitted.

2. OPTIMALITY OF GIBBS FIELD

In this section, we recall a connection between steganography and statistical physics by showing that for a given expected embedding distortion, the maximal payload is embedded when the embedding changes follow a particular form of Gibbs distribution.

We start by introducing basic concepts, notation, and terminology. The calligraphic font will be used primarily for sets, random variables will be typeset in capital letters, while their corresponding realizations will be in lower-case. Vectors or matrices will be always typeset in boldface lower and upper case, respectively. Although the idea presented in this paper is certainly applicable to steganography in other objects than digital images, the entire approach is described using the terms “image” and “pixel” for concreteness to simplify the language and to allow a smooth transition from theory to experiments on digital images.

An image $\mathbf{x} = (x_1, \ldots, x_n) \in \mathcal{X} \triangleq \mathcal{I}^n$ is a regular lattice of elements (pixels) $x_i \in \mathcal{I}$, $i \in \mathcal{S}$, $\mathcal{S} = \{1, \ldots, n\}$. The dynamic range, $\mathcal{I}$, depends on the character of the image data. For example, for an 8-bit grayscale image, $\mathcal{I} = \{0, 1, \ldots, 255\}$. 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.

Given $\mathcal{J} \subset \mathcal{S}$, we define $\mathbf{x}_{\mathcal{J}} \triangleq \{x_i | i \in \mathcal{J}\}$ and $\mathbf{x}_{\sim\mathcal{J}} \triangleq \{x_i | i \in \mathcal{S} - \mathcal{J}\}$. The image $(x_1, \ldots, x_{i-1}, y_i, x_{i+1}, \ldots, x_n)$ will be abbreviated as $y_i \mathbf{x}_{\sim i}$. The Iverson bracket, $[P]$, is defined as $[P] = 1$ when the statement P is true and zero otherwise. The symbol $\log x$ stands for the logarithm at the base of 2, while $\ln x$ is the natural logarithm.

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 $P(\mathbf{Y} = \mathbf{y}) = \pi(\mathbf{y})$. 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 a given $\mathbf{x}$, $\mathcal{Y}$, and payload (distortion). Finally, we note that the maximal expected payload that the sender can communicate to the receiver in this manner is the entropy

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

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.

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

$$D(\mathbf{y}) : \mathcal{Y} \to \mathbb{R}, \tag{2}$$

that is bounded, i.e., $|D(\mathbf{y})| < K$, for all $\mathbf{y} \in \mathcal{Y}$ for some sufficiently large K. 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 5.1.

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 [10, 18], 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.

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

Distortion-limited sender. Thus, 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:

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

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

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, $H(\pi_\lambda) - H(\mu) = D_{\mathrm{KL}}(\mu||\pi_\lambda) \geq 0$ [34]. 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

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

by direct evaluation. 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 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.*

It will be useful to think of the difference $\mathbf{s} = \mathbf{y} - \mathbf{x}$ as an embedding (flipping) pattern with a distortion (energy) $D(\mathbf{y})$ and of π_λ as a probability distribution on embedding patterns. Keep in mind, though, that the energy of an embedding pattern $\mathbf{s}$ in general needs the side-information in the form of the cover image $\mathbf{x}$ and is not just a function of $\mathbf{s}$. Indeed, when embedding in a single image, the cover $\mathbf{x}$ plays the role of a constant parameter that enters the definition of D and defines π_λ. Therefore, the optimal embedding rule will necessarily depend on the cover image and the rate–distortion bound will only be valid for a specific cover image $\mathbf{x}$.

To provide some examples, suppose the embedding algorithm flips the Least Significant Bits (LSBs) of x_i. Then, $\mathcal{Y} = \mathcal{I}_1 \times \cdots \times \mathcal{I}_n$ with $\mathcal{I}_i = \{x_i, \overline{x}_i\}$, where the bar denotes the operation of flipping the LSB. When using ±1 embedding (also called LSB matching [13]) in 8-bit grayscale images, $\mathcal{I}_i = \{x_i - 1, x_i, x_i + 1\}$ whenever $x_i \notin \{0, 255\}$ and $\mathcal{I}_i$ is appropriately modified for the boundary cases. For LSB embedding, $\mathbf{s}$ is a binary flipping pattern, while for ±1 embedding $s_i \in \{-1, 0, 1\}^n$. In general, when $|\mathcal{I}_i| = 2$ or 3 for all i, we will speak of binary and ternary embedding, respectively. In principle, the range of the embedding changes could be different at every pixel even though this case has been rarely considered in steganography so far. The wet paper scenario [9] is an example of this case. Here, wet pixels are required to attain only one value – the cover value, $\mathcal{I}_i = \{x_i\}$, while all other pixels can be modified.

3. THE SEPARATION PRINCIPLE

When designing practical steganographic methods that minimize distortion, one should compare their performance with the rate–distortion bound. This is a meaningful comparison for the distortion-limited sender who can assess the performance of a practical algorithm 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. This rather important practical issue could be resolved by simulating the impact of a scheme that operates *on the bound*. 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*.

The bound is obtained by solving the optimization problem (4) or (6). Depending on the form of the distortion function D, this task is usually rather difficult and one may have to resort to numerical methods.

Often, we may be able to *simulate* the impact of an optimal method (that embeds on the bound) even when we do not have the bound and do not know how to construct a practical embedding algorithm (see Section 4). The simulator can be tested with blind steganalyzers, giving the developer the ability to “prune” the design process and focus on implementing only the most promising candidates. Additionally, the simulator will inform the payload-limited sender about the potential improvement in statistical undetectability should the theoretical performance gap be closed.

We stress at this point that even though the optimal distribution of embedding modifications has a known analytic expression (8), it is in general 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 embedding patterns, 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 complicates the computation of the expected distortion (3) and entropy (1). Fortunately, to simulate optimal embedding and construct practical embedding algorithms, one needs to be able to merely *sample from* π_λ.

In some special cases, however, such as when the distortion D is additive, all three tasks of the separation principle can be realized. As this special case will be used later in Section 6 to design steganography with more general distortion functions D, we review it briefly below.

3.1 Additive distortion

We say that D is additive over the pixels (the embedding changes do not interact) when

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

with bounded $\rho_i : \mathcal{X} \times \mathcal{I}_i \to \mathbb{R}$ (that may depend on $\mathbf{x}$ in an arbitrary manner). In this case, the probability of an embedding pattern 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:

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

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

The impact of optimal embedding can be simulated by independently changing x_i to y_i with probabilities $\pi_\lambda(y_i)$. 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 payload ($H(\pi_\lambda) = m$) constraint amounts to solving an algebraic equation for λ. Practical near-optimal embedding algorithms exist that are based on syndrome-trellis codes [6, 7].

4. SIMULATING OPTIMAL EMBEDDING

As explained in Section 2, minimal-embedding-distortion steganography will introduce the embedding change $\mathbf{s} = \mathbf{y} - \mathbf{x}$ 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. This sampling algorithm will also allow us to construct practical embedding schemes in Sections 5 and 6. A useful resource containing the Gibbs sampler is [34].

4.1 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:

$$\pi_\lambda(Y_i = y_i' | \mathbf{Y}_{\sim i} = \mathbf{y}_{\sim i}) = \frac{\pi_\lambda(y_i'\mathbf{y}_{\sim i})}{\sum_{t_i\in\mathcal{I}_i} \pi_\lambda(t_i\mathbf{y}_{\sim i})}. \tag{16}$$

For all possible stego images $\mathbf{y}, \mathbf{y}' \in \mathcal{Y}$, the local characteristics (16) define the following matrices $\mathbf{\Pi}_i$, $i \in \{1, \dots, n\}$:

$$\mathbf{\Pi}_i(\mathbf{y}, \mathbf{y}') = \begin{cases} \pi_\lambda(Y_i = y_i' | \mathbf{Y}_{\sim i} = \mathbf{y}_{\sim i}) & \text{when } \mathbf{y}'_{\sim i} = \mathbf{y}_{\sim i} \\ 0 & \text{otherwise.} \end{cases} \tag{17}$$

Every matrix $\mathbf{\Pi}_i$ 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 $\mathbf{\Pi}_i$ is stochastic (the sum of its rows is one),

$$\sum_{\mathbf{y}'\in\mathcal{Y}} \mathbf{\Pi}_i(\mathbf{y}, \mathbf{y}') = 1, \text{ for all rows } \mathbf{y}, \tag{18}$$

$\mathbf{\Pi}_i$ is a transition probability matrix of some Markov chain on $\mathcal{Y}$. Every matrix $\mathbf{\Pi}_i$ satisfies the so-called detailed balance equation

$$\pi_\lambda(\mathbf{y})\mathbf{\Pi}_i(\mathbf{y}, \mathbf{y}') = \pi_\lambda(\mathbf{y}')\mathbf{\Pi}_i(\mathbf{y}', \mathbf{y}), \quad \text{for all } \mathbf{y}, \mathbf{y}' \in \mathcal{Y},\ i. \tag{19}$$

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

Algorithm 1 One sweep of a Gibbs sampler.

```
Set pixel counter $i = 1$
while $i \leq n$ do
    Compute the local characteristics:
        $\boldsymbol{\Pi}_{\sigma(i)}(y'_{\sigma(i)}\mathbf{y}_{\sim\sigma(i)}, \mathbf{y}),\ y'_{\sigma(i)} \in \mathcal{I}_{\sigma(i)}$    (26)
    Select one $y'_{\sigma(i)} \in \mathcal{I}_{\sigma(i)}$ pseudorandomly according to
    the probabilities (26) and change $y_{\sigma(i)} \leftarrow y'_{\sigma(i)}$
    $i \leftarrow i + 1$
end while
return $\mathbf{y}$
```

following chain of equalities:

$$\pi_\lambda(\mathbf{y})\boldsymbol{\Pi}_i(\mathbf{y},\mathbf{y}') \overset{(a)}{=} \pi_\lambda(\mathbf{y})\frac{\pi_\lambda(y'_i\mathbf{y}_{\sim i})}{\sum_{t_i\in\mathcal{I}_i}\pi_\lambda(t_i\mathbf{y}_{\sim i})} \tag{20}$$

$$\overset{(b)}{=} \frac{\pi_\lambda(\mathbf{y})\pi_\lambda(\mathbf{y}')}{\sum_{t_i\in\mathcal{I}_i}\pi_\lambda(t_i\mathbf{y}_{\sim i})} \tag{21}$$

$$= \pi_\lambda(\mathbf{y}')\frac{\pi_\lambda(\mathbf{y})}{\sum_{t_i\in\mathcal{I}_i}\pi_\lambda(t_i\mathbf{y}'_{\sim i})} \tag{22}$$

$$\overset{(c)}{=} \pi_\lambda(\mathbf{y}')\boldsymbol{\Pi}_i(\mathbf{y}',\mathbf{y}). \tag{23}$$

Equality (a) follows from the definition of $\boldsymbol{\Pi}_i$ (17), (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 (17).

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 (19) and then (18), we can now easily show that the vector $\boldsymbol{\pi}_\lambda$ is the left eigenvector of $\boldsymbol{\Pi}_i$ corresponding to the unit eigenvalue:

$$(\boldsymbol{\pi}_\lambda\boldsymbol{\Pi}_i)(\mathbf{y}') = \sum_{\mathbf{y}\in\mathcal{Y}}\pi_\lambda(\mathbf{y})\boldsymbol{\Pi}_i(\mathbf{y},\mathbf{y}') \tag{24}$$

$$= \sum_{\mathbf{y}\in\mathcal{Y}}\pi_\lambda(\mathbf{y}')\boldsymbol{\Pi}_i(\mathbf{y}',\mathbf{y}) = \pi_\lambda(\mathbf{y}'). \tag{25}$$

In (24), $(\boldsymbol{\pi}_\lambda\boldsymbol{\Pi}_i)(\mathbf{y}')$ is the $\mathbf{y}'$th element of the product of the vector $\boldsymbol{\pi}_\lambda$ and the matrix $\boldsymbol{\Pi}_i$.

We are now ready to describe the Gibbs sampler [11], which is a key element in our framework. Let σ be a permutation of the index set $\mathcal{S}$ called the visiting schedule ($\sigma(i)$, $i = 1,\ldots,n$ 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 (16) computed from the current values of the stego image $\mathbf{y}$. The entire sweep can be described by a transition probability matrix $\boldsymbol{\Pi}_\sigma$ obtained by matrix-multiplications of the individual transition probability matrices $\boldsymbol{\Pi}_{\sigma(i)}$:

$$\boldsymbol{\Pi}_\sigma(\mathbf{y},\mathbf{y}') \triangleq \boldsymbol{\Pi}_{\sigma(1)}\boldsymbol{\Pi}_{\sigma(2)}\cdots\boldsymbol{\Pi}_{\sigma(n)}(\mathbf{y},\mathbf{y}'). \tag{27}$$

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 $\boldsymbol{\Pi}_\sigma$ 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 $\boldsymbol{\Pi}_i$, it is also a left eigenvector for $\boldsymbol{\Pi}_\sigma$ and thus its stationary distribution due to its uniqueness. A standard result from the theory of Markov chains (see, e.g. Chapter 4 in [34]) 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 $\boldsymbol{\pi}_\lambda$:

$$||\boldsymbol{\nu}\boldsymbol{\Pi}_\sigma^k - \boldsymbol{\pi}_\lambda|| \to 0 \text{ with } k \to \infty \tag{28}$$

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.[2] 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.

4.2 Simulator of optimal embedding

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 without any assumptions on the distortion function D and without knowing the rate–distortion bound. This is because the local characteristics (16)

$$\pi_\lambda(Y_i = y'_i|\mathbf{Y}_{\sim i} = \mathbf{y}_{\sim i}) = \frac{\exp(-\lambda D(y'_i\mathbf{y}_{\sim i}))}{\sum_{t_i\in\mathcal{I}_i}\exp(-\lambda D(t_i\mathbf{y}_{\sim i}))}, \tag{29}$$

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 [34]) that

$$\frac{1}{k}\sum_{j=1}^{k} D\big(\mathbf{y}^{(j)}\big) \to E_{\pi_\lambda}[D] \text{ as } k \to \infty \tag{30}$$

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 [19].

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

From (10) and (13), we obtain

$$\frac{\partial}{\partial\lambda}H(\pi_\lambda) = \frac{\lambda}{\ln 2}\frac{\partial}{\partial\lambda}E_{\pi_\lambda}[D]. \tag{31}$$

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

$$H(\pi_\lambda) = H(\pi_{\lambda_0}) + \left[\frac{\lambda'}{\ln 2}E_{\pi_{\lambda'}}[D]\right]_{\lambda_0}^{\lambda} - \frac{1}{\ln 2}\int\limits_{\lambda_0}^{\lambda} E_{\pi_{\lambda'}}[D]\mathrm{d}\lambda'. \tag{32}$$

The value of λ that satisfies the entropy (payload) constraint can be again obtained using a binary search. Having obtained the expected distortion and 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 (30), since no practical guidelines exist for determining a sufficient number of sweeps and heuristic criteria are often used [4, 34]. Although the convergence to π_λ is exponential in the number of sweeps, the large number of stego images $\mathbf{y}$ may require a very large number of sweeps to converge close enough. Generally speaking, the stronger the dependencies between embedding changes the more sweeps are needed by the Gibbs sampler. 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 this case, the steganographer should consider other methods to estimate the expected distortion and entropy, such as the Wang–Landau algorithm [33]. The authors note that in general it is not possible to determine ahead of time which method will provide satisfactory performance. In our experience, the thermodynamic integration worked very well.

Finally, note that computing the rate–distortion bound is not necessary for practical embedding. In Section 5, 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 5.1 and 5.2.

5. LOCAL DISTORTION FUNCTION

Thanks to the Gibbs sampler, we can simulate the impact of optimal embedding 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. The design of near-optimal schemes for a general D is, however, quite difficult. In this section, we give D a specific local form that will allow us to construct practical embedding algorithms; it will be a sum of local potentials defined on small groups of pixels called cliques. This local form is general enough to capture dependencies among pixels as well as embedding changes while allowing construction of practical embedding schemes (Section 6).

First, we define a neighborhood system as a collection of subsets of the index set $\{\eta(i) \subset \mathcal{S} | i = 1, \ldots, n\}$ 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 $\mathcal{C}$.

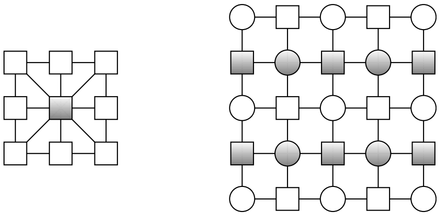

Figure 1: The 3×3 neighborhood and the tessellation of the index set $\mathcal{S}$ into four disjoint sublattices marked with four different symbols.

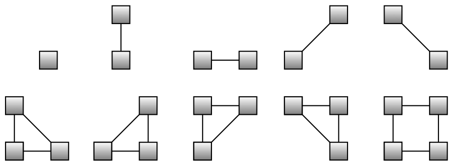

Figure 2: All possible cliques for the 3×3 neighborhood.

In this section and in Section 6, we will need to address pixels by their two-dimensional coordinates. We will thus be switching between using the index set $\mathcal{S} = \{1, \ldots, n\}$ 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 eight-element 3×3 neighborhood forms a neighborhood system (Figure 1). The cliques are formed by pairs of horizontally, vertically, and diagonally neighboring pixels, by three-pixel groups forming a right-angle triangle, and four-pixel cliques forming a 2×2 square (follow Figure 2). 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 3×3 neighborhood, there are four sublattices, $\mathcal{S} = \bigcup_{ab} \mathcal{S}_{ab}$, $1 \le a, b \le 2$,

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

For a clique c, we denote by $V_c(\mathbf{y})$ any bounded function that depends only on the values of $\mathbf{y}$ in the clique c, $V_c(\mathbf{y}) = V_c(\mathbf{y}_c)$ (the dependence on $\mathbf{x}$ may be arbitrary). We are now ready to introduce a local form of the distortion function as

$$D(\mathbf{y}) = \sum_{c \in \mathcal{C}} V_c(\mathbf{y}). \tag{33}$$

The important fact is that D is a sum of functions with a small support. Let us express the local characteristics (16) in terms of the newly-defined form (33):

$$\pi_\lambda(Y_i = y'_i | \mathbf{y}_{\sim i}) = \frac{\exp(-\lambda \sum_{c \in \mathcal{C}} V_c(y'_i \mathbf{y}_{\sim i}))}{\sum_{t_i \in \mathcal{I}_i} \exp(-\lambda \sum_{c \in \mathcal{C}} V_c(t_i \mathbf{y}_{\sim i}))} \tag{34}$$

$$\overset{(a)}{=} \frac{\exp(-\lambda \sum_{c \in \mathcal{C}(i)} V_c(y'_i \mathbf{y}_{\sim i}))}{\sum_{t_i \in \mathcal{I}_i} \exp(-\lambda \sum_{c \in \mathcal{C}(i)} V_c(t_i \mathbf{y}_{\sim i}))}, \tag{35}$$

Equality (a) holds because $\mathbf{y} = y'_i \mathbf{y}_{\sim i}$ on cliques c that do not contain the ith element and thus the terms V_c for such

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}
for $k = 1$ to $s$ do
    for every $i \in \mathcal{S}_k$ do
        Use (36) to calculate cost of changing $y_i \to y_i' \in \mathcal{I}_i$
    end for
    Embed $m/s$ bits while minimizing $\sum_{i \in \mathcal{S}_k} \rho_i(y_i' \mathbf{y}_{\sim i})$.
    Update $\mathbf{y}_{\mathcal{S}_k}$ with new values and keep $\mathbf{y}_{\sim \mathcal{S}_k}$ unchanged.
end for
return $\mathbf{y}$
```

cliques cancel from (35). 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 3×3 neighborhood system, changes made to pixels belonging, e.g., to the sublattice $\mathcal{S}_{11}$ 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 $\mathcal{S}_{12}$, etc.[3]

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).

5.1 Payload-limited sender

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

$$\rho_i(y_i' \mathbf{y}_{\sim i}) = \sum_{c \in \mathcal{C}, i \in c} V_c(y_i' \mathbf{y}_{\sim i}) \tag{36}$$

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, e.g., using the syndrome-trellis codes as described in Section 3.1. By repeating these embedding sweeps,[4] the introduced embedding pattern 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 $\{\rho_i(y_i' \mathbf{y}_{\sim i}) | i \in \mathcal{S}_k\}$ (36). 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. 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.

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}
for $k = 1$ to $s$ do
    for every $i \in \mathcal{S}_k$ do
        Use (36) to calculate cost of changing $y_i \to y_i' \in \mathcal{I}_i$
    end for
    Embed $m_k$ bits while $\sum_i \rho_i(y_i' \mathbf{y}_{\sim i}) = D_\epsilon \times |\{c \in \mathcal{C} | c \cap \mathcal{S}_k \neq \emptyset\}| / |\mathcal{C}|$.
    Update $\mathbf{y}_{\mathcal{S}_k}$ with new values and keep $\mathbf{y}_{\sim \mathcal{S}_k}$ unchanged.
end for
return $\mathbf{y}$ and $\sum_k m_k$ {stego image and number of bits}
```

5.2 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 4.2) and a 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,

$$E_{\pi_\lambda(\mathbf{Y}_{\mathcal{S}_k} | \mathbf{Y}_{\sim \mathcal{S}_k})}[D] = D_\epsilon \frac{|\{c \in \mathcal{C} | c \cap \mathcal{S}_k \neq \emptyset\}|}{|\mathcal{C}|}. \tag{37}$$

As described in Algorithm 3, the sender can realize this by embedding as many bits to every sublattice as possible while achieving the distortion (37). The embedding can be again implemented in practice using syndrome-trellis codes. Note that we do not need to compute the partition function for every image in order to realize the embedding. Moreover, when the distortion properties of every sublattice are the same, the search for correct parameter λ, as described in Section 4.2, is not needed either. This is because the syndrome-trellis codes [7] need the distortion at each lattice pixel (36) and not the embedding probabilities. (This eliminates the need for λ.) The effect of the number of sweeps during embedding needs to be studied specifically for each distortion measure.

At this point, we make a comment concerning Algorithms 2 and 3. By replacing the syndrome-trellis code 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 4.2. We still need to compute λ_k for each sublattice to compute the probabilities of modifying each pixel (15), but this can be done as described in Section 3.1 without having to run the Gibbs sampler or the expensive Wang–Landau algorithm.

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 $\mathbf{y}$), wet pixels are handled by forcing $\mathcal{I}_i = \{x_i\}$. Because this knowledge may not be available to the decoder in practice, the syndrome-trellis embedding algorithm should treat them either by setting $\rho_i(y_i \mathbf{y}_{\sim i}) = \infty$ or to some large constant for $y_i \neq x_i$ (for details, see [7]). Fortunately, the syndrome-trellis codes can generously accept various portions of wet pixels without any performance penalty [7].

[3] The Gibbs random field described by the joint distribution $\pi_x(y)$ with distortion (33) becomes a Markov random field on the same neighborhood system. This follows from the Hammersley-Clifford theorem [34].

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

5.3 Practical limits of the Gibbs sampler

Thanks to the bounds established in Section 2, 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:

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

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

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

with equality only when all variables are independent.[5] Thus, we will have in general

$$H^-(\mathbf{Y}) \triangleq \sum_{k=1}^{s} H\big(\mathbf{Y}_{\mathcal{S}_k} \big| \mathbf{Y}_{\sim\mathcal{S}_k} = \mathbf{y}^{[k]}_{\sim\mathcal{S}_k}\big) < H(\mathbf{Y}) = H(\pi_\lambda). \tag{38}$$

The term $H^-(\mathbf{Y})$ is recognized as the erasure entropy [31, 32] and it is equal to the conditional entropy (entropy rate) $H(\mathbf{Y}^{(l+1)}|\mathbf{Y}^{(l)})$ of the Markov process defined by our Gibbs sampler (c.f., (27)), where $\mathbf{Y}^{(l)}$ is the random variable obtained after l sweeps of the Gibbs sampler.

The sender will, in general, be unable to embed the maximal payload $H(\pi_\lambda)$ due to the limited number of sweeps of the Gibbs sampler, slight variations of the parameter λ among sublattices, and the erasure entropy inequality (38). The actual loss of payload can be assessed by evaluating the entropy of $H(\pi_\lambda)$, e.g., using the thermodynamic integration as explained in Section 4. In the journal version of this paper, it is shown that for the distortion function from Section 6 this payload loss is negligible even when only two sweeps of the Gibbs sampler are used. In general, though, the loss depends on the strength of interactions among pixels and must be investigated for each D separately.

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.

6. PRACTICAL EMBEDDING

We are now ready to describe a practical embedding algorithm that uses the ideas and theory developed so far. Instead of describing the most general setting, we opted for a simple variant, hoping that generalization to more complex cases will appear transparent to the reader. In Section 7, we compare the performance of a specific embedding scheme with other practical embedding algorithms by simulating their optimal performance.

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. In this section, we describe an approach inspired by models used in blind steganalysis, where images are projected onto a lower-dimensional feature space carefully selected to model well the noise component of cover images and to be sensitive to embedding changes. Here, a good distortion measure could be some norm of the difference between the cover and stego features. This way, by minimizing the embedding distortion, the cover model is also approximately preserved.

Most steganalysis features f_k, $k = 1, \ldots, d$, can be written as a sum of locally-supported functions across the image

$$f_k(\mathbf{x}) = \sum_{c \in \mathcal{C}} f_c^{(k)}(\mathbf{x}). \tag{39}$$

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

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

while the klth element of a horizontal co-occurrence matrix

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

is a sum over horizontally adjacent pixels. This is good because (39) already looks like a sum of potentials. However, the difference between features expressed in the form of a weighted norm,

$$\begin{aligned} ||f(\mathbf{x}) - f(\mathbf{y})|| &= \sum_{k=1}^{d} w_k |f_k(\mathbf{x}) - f_k(\mathbf{y})| \\ &= \sum_{k=1}^{n} w_k \Big| \sum_{c \in \mathcal{C}} f_c^{(k)}(\mathbf{x}) - \sum_{c \in \mathcal{C}} f_c^{(k)}(\mathbf{y}) \Big|, \end{aligned}$$

is no longer a sum of *local* potentials. Fortunately, we can obtain an upper bound on the norm that has the required form:

$$\begin{aligned} ||f(\mathbf{x}) - f(\mathbf{y})|| &= \sum_{k=1}^{d} w_k \Big| \sum_{c \in \mathcal{C}} f_c^{(k)}(\mathbf{x}) - \sum_{c} f_c^{(k)}(\mathbf{y}) \Big| && (40) \\ &\le \sum_{k=1}^{d} w_k \sum_{c \in \mathcal{C}} |f_c^{(k)}(\mathbf{x}) - f_c^{(k)}(\mathbf{y})| && (41) \\ &= \sum_{c \in \mathcal{C}} \sum_{k=1}^{d} w_k |f_c^{(k)}(\mathbf{x}) - f_c^{(k)}(\mathbf{y})| && (42) \\ &= \sum_{c \in \mathcal{C}} V_c(\mathbf{y}), && (43) \end{aligned}$$

where

$$V_c(\mathbf{y}) = \sum_{k=1}^{d} w_k |f_c^{(k)}(\mathbf{x}) - f_c^{(k)}(\mathbf{y})|. \tag{44}$$

We will call the sum $\sum_{c \in \mathcal{C}} V_c(\mathbf{y})$ the bounding distortion. Following our convention explained in Section 2, we describe

[5] 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.

the methodology for a fixed cover image $\mathbf{x}$ and thus do not make the dependence of V_c on $\mathbf{x}$ explicit.

We now provide a specific example of this approach. Our choice is motivated by our desire to work with a modern, well-established feature set so that later, in Section 7, 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 [21]. It is a slight modification of the SPAM set [20], 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 (44).

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 $D_{i,j}^{\rightarrow}(\mathbf{x}) = x_{i,j+1} - x_{i,j}$, $i = 1, \ldots, n_1, j = 1, \ldots, n_2 - 1$:

$$A_{kl}^{\rightarrow}(\mathbf{x}) = \frac{1}{n_1(n_2-2)} \sum_{i=1}^{n_1} \sum_{j=1}^{n_2-2} [(D_{i,j}^{\rightarrow}, D_{i,j+1}^{\rightarrow})(\mathbf{x}) = (k,l)]. \tag{45}$$

For compactness, in (45) we abbreviated the argument of the Iverson bracket from $D_{i,j}^{\rightarrow}(\mathbf{x}) = k \;\&\; D_{i,j+1}^{\rightarrow}(\mathbf{x}) = l$ to $(D_{i,j}^{\rightarrow}, D_{i,j+1}^{\rightarrow})(\mathbf{x}) = (k,l)$. Clearly, $A_{ij}^{\rightarrow}(\mathbf{x})$ 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 "→" denotes the fact that the differences are computed by subtracting the left pixel from the right one. Similarly,

$$A_{kl}^{\leftarrow}(\mathbf{x}) = \frac{1}{n_1(n_2-2)} \sum_{i=1}^{n_1} \sum_{j=3}^{n_2} [(D_{i,j}^{\leftarrow}, D_{i,j-1}^{\leftarrow})(\mathbf{x}) = (k,l)] \tag{46}$$

with $D_{ij}^{\leftarrow}(x) = x_{i,j-1} - x_{ij}$. By analogy, we can define vertical, diagonal, and minor diagonal matrices $A_{kl}^{\downarrow}$, $A_{kl}^{\uparrow}$, $A_{kl}^{\nearrow}$, $A_{kl}^{\swarrow}$, $A_{kl}^{\searrow}$, $A_{kl}^{\nwarrow}$. All eight matrices are sample joint probabilities of observing the differences k and l between three consecutive pixels along a certain direction. Due to $D_{ij}^{\rightarrow}(\mathbf{x}) = -D_{i,j+1}^{\leftarrow}(\mathbf{x})$ only $A_{kl}^{\rightarrow}$, $A_{kl}^{\nearrow}$, $A_{kl}^{\uparrow}$, $A_{kl}^{\nwarrow}$ are needed since $A_{kl}^{\rightarrow} = A_{-l,-k}^{\leftarrow}$, 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 [20], 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.[6] 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 [17].

By embedding a message, $A_{kl}^{\rightarrow}(\mathbf{x})$ is modified to $A_{kl}^{\rightarrow}(\mathbf{y})$. 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):

$$\begin{aligned}
&|A_{kl}^{\rightarrow}(\mathbf{y}) - A_{kl}^{\rightarrow}(\mathbf{x})| = &(47)\\
&\quad = \frac{1}{n_1(n_2-2)} \Bigg| \sum_{i,j} [(D_{i,j}^{\rightarrow}, D_{i,j+1}^{\rightarrow})(\mathbf{y}) = (k,l)] &(48)\\
&\qquad - [(D_{i,j}^{\rightarrow}, D_{i,j+1}^{\rightarrow})(\mathbf{x}) = (k,l)] \Bigg| &(49)\\
&\quad \le \frac{1}{n_1(n_2-2)} \sum_{i,j} \Big| [(D_{i,j}^{\rightarrow}, D_{i,j+1}^{\rightarrow})(\mathbf{y}) = (k,l)] &(50)\\
&\qquad - [(D_{i,j}^{\rightarrow}, D_{i,j+1}^{\rightarrow}(\mathbf{x}) = (k,l)] \Big| &(51)\\
&\quad = \sum_{c \in \mathcal{C}^{\rightarrow}} H_c^{(k,l)\rightarrow}(\mathbf{y}), &(52)
\end{aligned}$$

where we defined the following locally-supported functions

$$H_c^{(k,l)\rightarrow}(\mathbf{y}) = \Big| [(D_{i,j}^{\rightarrow}, D_{i,j+1}^{\rightarrow})(\mathbf{y}) = (k,l)] - [(D_{i,j}^{\rightarrow}, D_{i,j+1}^{\rightarrow})(\mathbf{x}) = (k,l)] \Big| \tag{53}$$

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

$$D(\mathbf{y}) = \sum_{c \in \mathcal{C}} V_c(\mathbf{y}), \tag{54}$$

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

$$V_c(\mathbf{y}) = \sum_{k,l} w_{kl} H_c^{(k,l)\rightarrow}(\mathbf{y}), \text{ for each clique } c \in \mathcal{C}^{\rightarrow}, \tag{55}$$

and similarly for the other three clique types. Notice that we again introduced weights $w_{kl} > 0$ 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 (Figure 3) and thus the index set can be decomposed into nine disjoint sublattices $\mathcal{S} = \bigcup_{ab} \mathcal{S}_{ab}$, $1 \le a,b \le 3$,

$$\mathcal{S}_{ab} = \{(a+3k, b+3l) | 1 \le a+3k \le n_1, 1 \le b+3l \le n_2\}. \tag{56}$$

7. EXPERIMENTS

In this section, we discuss the options the new framework offers to the steganographer and then compare them

[6] Similar reasoning for constructing the distortion function was used in the HUGO algorithm [21].

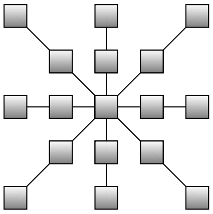

Figure 3: The union of all 12 cliques consisting of three pixels arranged in a straight line in the 5×5 square neighborhood.

with selected standard steganographic methods on two image databases. We investigate both the payload-limited sender and the distortion-limited sender.

When the distortion is defined as a norm of the difference of feature vectors used to model cover images, $D(\mathbf{y}) = \|f(\mathbf{x}) - f(\mathbf{y})\|$, the steganography design principles based on model preservation and on minimizing distortion coincide. Because such D is non-additive, up until now steganographers had to use an additive approximation of D, such as

$$\hat{D}(\mathbf{y}) = \sum_{i=1}^{n} D(y_i \mathbf{x}_{\sim i}). \tag{57}$$

Embedding with $\hat{D}$ can be simulated and realized as explained in Section 3.1. However, the mismatch in the minimized distortion function leads to a capacity loss. Moreover, the additive approximation can no longer capture interactions among embedding changes.

This paper allows the sender to work directly with D and *simulate* the impact of optimal embedding using methods of Section 4.2. However, the sender cannot embed in practice due to the non-local character of D. One possibility is to use the bounding distortion (44), which has a local character, and apply the embedding algorithms described in Section 5.1 and 5.2. Because we can compute the rate–distortion bound for D and realize the simulator of optimal embedding, we can now assess how much payload (or security) is lost when using both approximations above by evaluating the performance w.r.t. to the bounds and comparing the statistical detectability obtained using blind steganalyzers.

The question of optimizing the local potential functions w.r.t. statistical detectability is an important direction the authors intend to explore in the future. For example, the framework described in this paper allows the sender to formulate the local potentials directly instead of obtaining them as the bounding distortion. The cliques and their potentials may be determined by the local image content or by learning the cover source, for example, using the method of fields of experts [23].

In the rest of this section, we experimentally compare steganography implemented via the bounding distortion and the additive approximation (57) with other standard steganographic methods. We do so for the payload-limited sender in Section 7.1 as well as the distortion-limited sender (Section 7.2). Following the separation principle, we study the security of all embedding algorithms by comparing their performance when simulated at their corresponding rate–distortion bounds.

We start with $D(\mathbf{y}) = \|f(\mathbf{x}) - f(\mathbf{y})\|$ defined as the weighted norm in the feature space formed by joint probability matrices $A_{klm}^{\rightarrow}(\mathbf{x})$ computed in four spatial directions similarly as described in (45). The difference vector was computed from *four* consecutive pixels $(D_{ij}^{\rightarrow}, D_{ij+1}^{\rightarrow}, D_{ij+2}^{\rightarrow}) = (k, l, m)$ rather than three. All matrices were used at their full size ($T = 255$) leading to model dimensionality of $d = 4 \times 511^3 \approx 5 \cdot 10^8$. The weights w entering the norm, were

$$w_{klm} = \left(\sigma + ||(D_{ij}^{\rightarrow}, D_{ij+1}^{\rightarrow}, D_{ij+2}^{\rightarrow})||_2\right)^{-\theta}, \tag{58}$$

with $\sigma = 1$ and $\theta = 1$ ($||x||_2$ denotes the L_2 norm). The weights encourage the embedding algorithm to modify those parts of the cover that are difficult to model accurately, forcing thus the steganalyst to use a more accurate model. Here, the advantage goes to the steganographer because, as already mentioned above, preserving a high-dimensional feature vector is more feasible than accurately modeling it [21].

Because the neighborhood in this case contains 7×7 pixels, the image was divided into 16 square sublattices on which embedding was simulated independently as described in Section 3.1. The payload-limited sender was simulated using the Gibbs sampler (Algorithm 2) constrained to two sweeps.

We implemented this framework with three different ranges of stego pixels: binary flipping patterns, $\mathcal{I}_i = \{x_i, y_i\}$, where y_i was selected randomly and uniformly from $\{x_i - 1, x_i + 1\}$ and then fixed for all experiments with cover $\mathbf{x}$, ternary patterns, $\mathcal{I}_i = \{x_i - 1, x_i, x_i + 1\}$, and pentary patterns, $\mathcal{I}_i = \{x_i - 2, \ldots, x_i + 2\}$. For all three cases, we simulated the method based on the bounding distortion (44) and the additive approximation (57) on the $d = 4 \times 511^2$-dimensional feature space of joint probability matrices $A_{klm}^{\rightarrow}(\mathbf{x})$.

For comparison, we contrasted the performance against two standard embedding methods: binary ±1 embedding constrained to the same sets $\mathcal{I}_i$ as the Gibbs sampler and ternary ±1 embedding with $\mathcal{I}_i = \{x_i - 1, x_i, x_i + 1\}$. Both schemes are special cases of our framework with $D(\mathbf{y}) = \sum_i [x_i \neq y_i]$. We repeat that all schemes were simulated on their corresponding bounds.

All algorithms were tested on two image sources with different noise characteristics: 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, and the NRCS database[7] with 3322 color scans of analogue photographs mostly of size 2100×1500 pixels converted to grayscale. For algorithms based on the Gibbs construction, simulating the optimal noise in C++ took less than 5 seconds for BOWS2 images and 60 seconds for the larger images from the NRCS database (for both the payload and distortion-limited sender).

Steganalysis was carried out using the second-order SPAM feature set with $T = 3$ [20]. Each image database was evenly divided into a training and a testing set of cover and stego images, respectively. For each database, a separate 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 $(C, \gamma) \in \left\{(10^k, 2^{j-L}) | k \in \{-3, \ldots, 4\}, j \in \{-3, \ldots, +3\}\right\}$, where L is the binary logarithm of the number of features used for steganalysis.

The steganalysis results are reported using a measure frequently used in steganalysis – the minimum average classification error $P_{\mathrm{E}} = \min(P_{\mathrm{FA}} + P_{\mathrm{MD}})/2$, where P_{FA} and

[7] http://photogallery.nrcs.usda.gov/

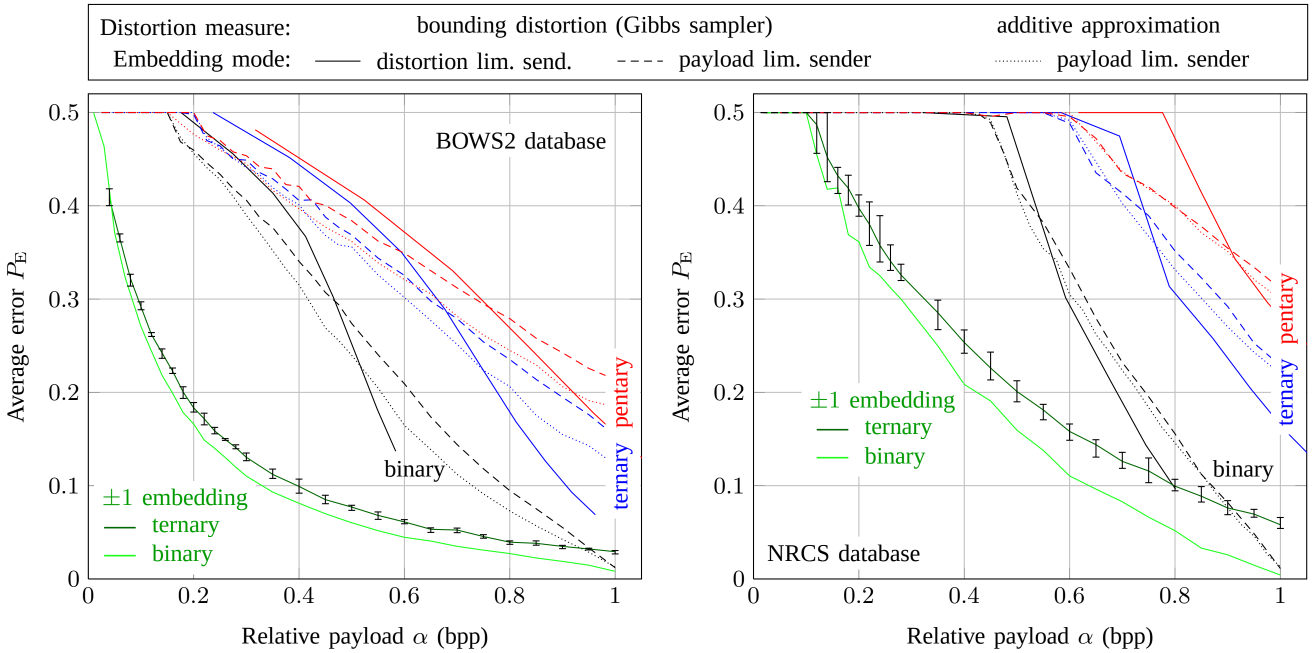

Figure 4: Comparison of ±1 embedding with optimal binary and ternary coding with embedding algorithms proposed in Section 7 for both payload-limited and distortion-limited sender. Error bars depict the minimum and maximum P_E over five runs (BOWS2) or ten runs (NRCS) of SVM classifiers with different division of images into training and testing set. Error bars for other experiments were similar and are not displayed.

P_{MD} are the false-alarm and missed-detection probabilities. A randomly guessing detector has $P_E = 0.5$.

7.1 Payload-limited sender

Figure 4 displays the comparison of all tested embedding methods. For the BOWS2 database, the methods based on the additive approximation and the bounding distortion are completely undetectable for payloads smaller than 0.15 bpp (bits per pixel), which suggests that the embedding changes are made in pixels not covered by the SPAM features. This number increases to at least 0.45 bpp for the NRCS database which is expected because its images are more noisy. For such payloads, the detector makes random guesses and, thus, due to the large number of testing samples, its error becomes exactly $P_E = 0.5$. With the relative payload α approaching 1, binary embedding schemes degenerate to binary ±1 embedding and thus become equally detectable. The same holds for ternary schemes. Both schemes allow communicating more than ten times larger payloads with $P_E = 40\%$, when compared to ternary ±1 embedding (on the BOWS2 database), and roughly four times larger payloads for the NRCS database. The results also suggest that secure payload can be further increased by allowing embedding changes of larger amplitude (up to ±2). Of course, this benefit is closely tied to the design of D because larger changes are easily detectable when not made adaptively [30].

The advantage of using the Gibbs sampler for embedding is more apparent for larger payloads, when the embedding changes start to interact (the BOWS2 database only). We believe this is due to strong inter-pixel dependencies caused by resizing the original image.

7.2 Distortion-limited sender

In this paper, we worked out the proposed methodology for both the 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.[8] 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 [14].

Since the payload is now determined by image content, it varies over the database. In this setup, we trained the steganalyzer on stego images embedded with a fixed distortion constraint D_ϵ. To be able to display the results in Figure 4, we reparametrized P_E to be a function of the relative payload α, which we obtain for each D_ϵ by averaging α over all images from the database. The solid lines represent the results obtained from the Gibbs sampler (Algorithm 3 with three sweeps) with $D(\mathbf{y})$ defined as the bounding distortion. As long as the distortion adequately measures statistical detectability, the distortion-limited sender should be more secure than the payload-limited sender. Figure 4 confirms this up to a certain payload where the performance is swapped. This means that either our distortion function is suboptimal or the steganalyzer does not properly measure statistical detectability.

Because the images in both databases are all of the same size, a fixed value of D_ϵ was used for all images. When dealing with images of varying size, we should set $D_\epsilon =$

[8] For schemes with uniform embedding cost, these two cases coincide.

$d_\epsilon\sqrt{n}$, at least for stegosystems falling under the square root law [8, 15].

As a final remark, we would like to point out that even though the improvement brought by the Gibbs construction over the additive approximation is not very large (and negligible for the NRCS database) it will likely increase in the future as practical steganalysis manages to better exploit inter-pixel dependencies. This is because mutually independent embedding cannot properly preserve dependencies or model interactions among embedding changes. For example, steganography in digital-camera color images will likely benefit from the Gibbs construction due to strong dependencies among color planes caused by color interpolation and other in-camera processing.

8. CONCLUSION

Recent developments in steganography for real digital media suggest that substantial increase in secure payload can no longer be achieved by improving embedding efficiency of systems that minimize additive embedding distortion, such as the number of embedding changes. As this approach has essentially reached its limits, further increase in secure payload can only be achieved by adaptive embedding algorithms modifying the cover object by larger than minimal amplitudes while minimizing a suitably-defined non-additive distortion function capable of capturing the interaction among embedding changes and preserving inter-pixel dependencies. Non-additive distortion also arises when the sender embeds to approximately preserve the cover feature vector. The proposed work allows the steganographer to preserve a high-dimensional model, providing thus an important advantage over the steganalyst who is facing the much harder task of having to learn a high-dimensional cover source model using statistical learning tools.

In this paper, we have introduced a complete methodology for constructing steganographic methods that minimize an arbitrarily defined distortion measure D. In doing so, we gave the steganographer substantial freedom in defining D to properly capture statistical detectability. The proposed framework is called the Gibbs construction and it connects steganography with statistical physics, which contributed with many practical algorithms. These algorithms, mainly based on the Gibbs sampler, allowed us to address important problems, such as deriving the rate–distortion bounds, simulating the optimal stego noise, and realizing near-optimal embedding schemes. The losses obtained from individual design steps can be evaluated separately (the so-called "separation principle").

When D is defined as a sum of local potentials, practical near-optimal embedding methods can be implemented with syndrome-trellis codes [6, 7] by following the Gibbs sampler. When D is not in this form, 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 D (or its best local approximation) is left as part of our future effort due to its inherent complexity.

Finally, note that the distortion measure is only used 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.

The source code and the results from all experiments are available at `http://dde.binghamton.edu/download/gibbs`.

9. ACKNOWLEDGMENTS

The work on this paper was supported by Air Force Office of Scientific Research under the research grant number FA9550-08-1-0084. 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. Special thanks belong to Tomáš Pevný, Radford M. Neal, and Avinash Varna for useful discussions.

10. REFERENCES

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

[2] G. Cancelli and M. Barni. MPSteg-color: A new steganographic technique for color images. In *Information Hiding, 9th International Workshop*, volume 4567 of *Lect. Notes in Computer Sc.*, pages 1–15, Saint Malo, France, June 11–13, 2007.

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

[4] 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.

[5] T. Filler and J. Fridrich. Gibbs construction in steganography. *IEEE Trans. on Information Forensics and Security*, 2010. Submitted.

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

[7] T. Filler, J. Judas, and J. Fridrich. Minimizing embedding impact in steganography using trellis-coded quantization. In *Proceedings SPIE, Electronic Imaging, Security and Forensics of Multimedia XII*, volume 7541, pages 05–01–05–14, January 17–21, 2010.

[8] T. Filler, A. D. Ker, and J. Fridrich. The Square Root Law of steganographic capacity for Markov covers. In *Proceedings SPIE, Electronic Imaging, Security and Forensics of Multimedia XI*, volume 7254, pages 08 1–08 11, San Jose, CA, January 18–21, 2009.

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

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

[11] 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.

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

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

[14] A. D. Ker. Batch steganography and pooled steganalysis. In *Information Hiding, 8th Intern. Workshop*, volume 4437 of *Lect. Notes in Computer Sc.*, pages 265–281, Alexandria, VA, July 10–12, 2006.

[15] A. D. Ker, T. Pevný, J. Kodovský, and J. Fridrich. The Square Root Law of steganographic capacity. In *Proceedings of the 10th ACM Multimedia & Security Workshop*, pages 107–116, Oxford, UK, September 22–23, 2008.

[16] Y. Kim, Z. Duric, and D. Richards. Modified matrix encoding technique for minimal distortion steganography. In *Information Hiding, 8th Intern. Workshop*, volume 4437 of *Lect. Notes in Computer Sc.*, pages 314–327, Alexandria, VA, July 10–12, 2006.

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

[18] J. Kodovský, T. Pevný, and J. Fridrich. Modern steganalysis can detect YASS. In *Proceedings SPIE, Electronic Imaging, Security and Forensics of Multimedia XII*, volume 7541, pages 02–01–02–11, January 17–21, 2010.

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

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

[21] T. Pevný, T. Filler, and P. Bas. Using high-dimensional image models to perform highly undetectable steganography. Lect. Notes in Computer Sc. Springer-Verlag, New York, 2010.

[22] T. Pevný and J. Fridrich. Merging Markov and DCT features for multi-class JPEG steganalysis. In *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.

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

[24] V. Sachnev, H. J. Kim, and R. Zhang. Less detectable JPEG steganography method based on heuristic optimization and BCH syndrome coding. In *Proceedings of the 11th ACM Multimedia & Security Workshop*, pages 131–140, Princeton, NJ, Sept. 2009.

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

[26] 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 *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, pages 1292–1295, 2008.

[27] A. Sarkar, K. Solanki, U. Madhow, and B. S. Manjunath. Secure steganography: Statistical restoration of the second order dependencies for improved security. In *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE Intern. Conf. on*, volume 2, pages II–277–II–280, April 15–20, 2007.

[28] Y. Q. Shi, C. Chen, and W. Chen. A Markov process based approach to effective attacking JPEG steganography. In *Information Hiding, 8th Intern. Workshop*, volume 4437 of *Lect. Notes in Computer Sc.*, pages 249–264, Alexandria, VA, July 10–12, 2006.

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

[30] D. Soukal, J. Fridrich, and M. Goljan. Maximum likelihood estimation of secret message length embedded using $\pm k$ steganography in spatial domain. In E. J. Delp and P. W. Wong, editors, *Proceedings SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents VII*, volume 5681, pages 595–606, San Jose, CA, January 16–20, 2005.

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

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

[33] 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.

[34] G. Winkler. *Image Analysis, Random Fields and Markov Chain Monte Carlo Methods: A Mathematical Introduction.* Springer-Verlag, Berlin Heidelberg, 2nd edition, 2003.