Minimizing Additive Distortion in Steganography Using Syndrome-Trellis Codes

Tomáš Filler, *Member, IEEE*, Jan Judas, *Member, IEEE*, and Jessica Fridrich, *Member, IEEE*

***Abstract*—This paper proposes a complete practical methodology for minimizing additive distortion in steganography with general (nonbinary) embedding operation. Let every possible value of every stego element be assigned a scalar expressing the distortion of an embedding change done by replacing the cover element by this value. The total distortion is assumed to be a sum of per-element distortions. Both the payload-limited sender (minimizing the total distortion while embedding a fixed payload) and the distortion-limited sender (maximizing the payload while introducing a fixed total distortion) are considered. Without any loss of performance, the nonbinary case is decomposed into several binary cases by replacing individual bits in cover elements. The binary case is approached using a novel syndrome-coding scheme based on dual convolutional codes equipped with the Viterbi algorithm. This fast and very versatile solution achieves state-of-the-art results in steganographic applications while having linear time and space complexity w.r.t. the number of cover elements. We report extensive experimental results for a large set of relative payloads and for different distortion profiles, including the wet paper channel. Practical merit of this approach is validated by constructing and testing adaptive embedding schemes for digital images in raster and transform domains. Most current coding schemes used in steganography (matrix embedding, wet paper codes, etc.) and many new ones can be implemented using this framework.**

***Index Terms*—Coding loss, convolutional codes, embedding impact, matrix embedding, steganography, trellis-coded quantization, wet paper codes.**

I. Introduction

THERE exist two mainstream approaches to steganography in empirical covers, such as digital media objects: steganography designed to preserve a chosen cover model and steganography minimizing a heuristically-defined embedding distortion. The strong argument for the former strategy is that provable undetectability can be achieved w.r.t. a specific model. The disadvantage is that an adversary can usually rather easily identify statistical quantities that go beyond the chosen model that allow reliable detection of embedding changes. The latter strategy is more pragmatic—it abandons modeling the cover source and instead tells the steganographer to embed payload while minimizing a distortion function. In doing so, it gives up any ambitions for perfect security. Although this may seem as a costly sacrifice, it is not, as empirical covers have been argued to be incognizable [1], which prevents model-preserving approaches from being perfectly secure as well.

While we admit that the relationship between distortion and steganographic security is far from clear, embedding while minimizing a distortion function is an easier problem than embedding with a steganographic constraint (preserving the distribution of covers). It is also more flexible, allowing the results obtained from experiments with blind steganalyzers to drive the design of the distortion function. In fact, today's least detectable steganographic schemes for digital images [2]–[5] were designed using this principle. Moreover, when the distortion is defined as a norm between feature vectors extracted from cover and stego objects, minimizing distortion becomes tightly connected with model preservation insofar the features can be considered as a low-dimensional model of covers. This line of reasoning already appeared in [5] and [6] and was further developed in [7].

With the exception of [7], steganographers work with additive distortion functions obtained as a sum of single-letter distortions. A well-known example is matrix embedding where the sender minimizes the total number of embedding changes. Near-optimal coding schemes for this problem appeared in [8] and [9], together with other clever constructions and extensions [10]–[15]. When the single-letter distortions vary across the cover elements, reflecting thus different costs of individual embedding changes, current coding methods are highly suboptimal [2], [4].

This paper provides a general methodology for embedding while minimizing an arbitrary additive distortion function with a performance near the theoretical bound. We present a complete methodology for solving both the payload-limited and the distortion-limited sender. The implementation described in this paper uses standard signal processing tools—convolutional codes with a trellis quantizer—and adapts them to our problem by working with their dual representation. These codes, which we call the syndrome-trellis codes (STCs), can directly improve the security of many existing steganographic schemes, allowing them to communicate larger payloads at the same embedding distortion or to decrease the distortion for a given payload. In addition, this work allows an iterative design of

Manuscript received October 07, 2010; revised March 10, 2011; accepted March 14, 2011. Date of publication April 05, 2011; date of current version August 17, 2011. This work was done while J. Judas was visiting Binghamton University. The work on this paper was supported by the 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 associate editor coordinating the review of this manuscript and approving it for publication was Dr. Mauro Barni.

The authors are with the Department of Electrical and Computer Engineering, Binghamton University, NY, 13902 USA (e-mail: tomas.filler@gmail.com; snugar.i@gmail.com; fridrich@binghamton.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIFS.2011.2134094

new embedding algorithms by making successive adjustments to the distortion function to minimize detectability measured using blind steganalyzers on real cover sources [4], [5], [16].

This paper is organized as follows. In the next section, we introduce the central notion of a distortion function. The problem of embedding while minimizing distortion is formulated in Section III, where we introduce theoretical performance bounds as well as quantities for evaluating the performance of practical algorithms with respect to each other and the bounds. The syndrome coding method for steganographic communication is reviewed in Section IV. By pointing out the limitations of previous approaches, we motivate our contribution, which starts in Section V, where we introduce a class of syndrome-trellis codes for binary embedding operations. We describe the construction and optimization of the codes and provide extensive experimental results on different distortion profiles including the wet paper channel. In Section VI, we show how to decompose the problem of embedding using nonbinary embedding operations to a series of binary problems using a multilayered approach so that practical algorithms can be realized using binary STCs. The application and merit of the proposed coding construction is demonstrated experimentally in Section VII on covers formed by digital images in raster and transform (JPEG) domains. Both the binary and nonbinary versions of payload- and distortion-limited senders are tested by blind steganalysis. Finally, the paper is concluded in Section VIII.

This paper is a journal version of [17] and [18], where the STCs and the multilayered construction were introduced. This paper unifies these methods into a complete and self-contained framework. Novel performance results and comparisons are included.

All logarithms in this paper are at the base of 2. We use the Iverson bracket $[I]$ defined to be 1 if the logical expression I is true and zero otherwise. The binary entropy function $h(x) = -x\log x - (1-x)\log(1-x)$ is expressed in bits. The calligraphic font will be used solely for sets, random variables will be typeset in capital letters, while their corresponding realizations will be in lower-case. 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 $\mathbb{A}$).

II. Distortion Function

For concreteness, and without loss of generality, we will call $\mathbf{x}$ image and x_i its ith pixel, even though other interpretations are certainly possible. For example, x_i may represent an RGB triple in a color image, a quantized DCT coefficient in a JPEG file, etc. Let $\mathbf{x} = (x_1, \ldots, x_n) \in \mathcal{X} = \{\mathcal{I}\}^n$ be an n-pixel cover image with the pixel dynamic range $\mathcal{I}$. For example, $\mathcal{I} = \{0, \ldots, 255\}$ for 8-bit grayscale images.

The sender communicates a message to the receiver by introducing modifications to the cover image and sending a stego image $\mathbf{y} = (y_1, \ldots, y_n) \in \mathcal{Y} = \mathcal{I}_1 \times \mathcal{I}_2 \times \cdots \times \mathcal{I}_n$, where $\mathcal{I}_i \subset \mathcal{I}$ are such that $x_i \in \mathcal{I}_i$. We call the embedding operation *binary* if $|\mathcal{I}_i| = 2$, or *ternary* if $|\mathcal{I}_i| = 3$ for every pixel i. For example, ± 1 embedding (sometimes called LSB matching) can be represented by $\mathcal{I}_i = \{x_i - 1, x_i, x_i + 1\}$ with appropriate modifications at the boundary of the dynamic range.

The impact of embedding modifications will be measured using a distortion function D. The sender will strive to embed payload while minimizing D. In this paper, we limit ourselves to an additive D in the form[1]

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

where $\rho_i : \mathcal{X} \times \mathcal{I}_i \to [-K, K], 0 < K < \infty$, are bounded functions expressing the cost of replacing the cover pixel x_i with y_i. Note that ρ_i may arbitrarily depend on the entire cover image $\mathbf{x}$, allowing thus the sender to incorporate inter-pixel dependencies [5]. The fact that the value of $\rho_i(\mathbf{x}, y_i)$ is independent of changes made at other pixels implies that the embedding changes do not interact.

The boundedness of $D(\mathbf{x}, \mathbf{y})$ is not limiting the sender in practice since the case when a particular value y_i is forbidden (a requirement often found in practical steganographic schemes [16]) can be resolved by excluding y_i from $\mathcal{I}_i$. In practice, the sets $\mathcal{I}_i$, $i \in \{1, \ldots, n\}$, may depend on cover pixels and thus may not be available to the receiver. To handle this case, we expand the domain of ρ_i to $\mathcal{X} \times \mathcal{I}$ and define $\rho_i(\mathbf{x}, y_i) = \infty$ whenever $y_i \notin \mathcal{I}_i$.

We intentionally keep the definition of the distortion function rather general. In particular, we do *not* require $\rho_i(\mathbf{x}, x_i) \leq \rho_i(\mathbf{x}, y_i)$ for all $y_i \in \mathcal{I}_i$ to allow for the case when it is actually beneficial to make an embedding change instead of leaving the pixel unchanged. An example of this situation appears in [7].

III. Problem Formulation

This section contains a formal definition of the problem of embedding while minimizing a distortion function. We state the performance bounds and define some numerical quantities that will be used to compare coding methods w.r.t. each other and to the bounds.

We assume the sender obtains her payload in the form of a pseudo-random bit stream, such as by compressing or encrypting the original message. We further assume that the embedding algorithm associates every cover image $\mathbf{x}$ with a pair $\{\mathcal{Y}, \pi\}$, where $\mathcal{Y}$ is the set of all stego images into which $\mathbf{x}$ can be modified and π is their probability distribution characterizing the sender's actions, $\pi(\mathbf{y}) \stackrel{\Delta}{=} P(\mathbf{y} = \mathbf{y}|\mathbf{x})$. Since the choice of $\{\mathcal{Y}, \pi\}$ depends on the cover image, all concepts derived from these quantities necessarily depend on $\mathbf{x}$ as well. We think of $\mathbf{x}$ as a constant parameter that is *fixed in the very beginning* and thus we do not further denote the dependency on it explicitly. For this reason, we simply write $D(\mathbf{y}) \stackrel{\Delta}{=} D(\mathbf{x}, \mathbf{y})$.

If the receiver knew $\mathbf{x}$, the sender could send up to

$$H(\pi) = -\sum_{\mathbf{y} \in \mathcal{Y}} \pi(\mathbf{y}) \log \pi(\mathbf{y}) \tag{2}$$

bits on average while introducing the average distortion

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

by choosing the stego image according to π. By the Gel'fand–Pinsker theorem [19], the knowledge of $\mathbf{x}$ does not give any fundamental advantage to the receiver and the

[1]The case of embedding with nonadditive distortion functions is addressed in [7] by converting it to a sequence of embeddings with an additive distortion.

same performance can be achieved as long as $\mathbf{x}$ is known to the sender. Indeed, none of the practical embedding algorithms introduced in this paper requires the knowledge of $\mathbf{x}$ or D for reading the message.

The task of embedding while minimizing distortion can assume two forms:

- **Payload-limited sender (PLS)**: embed a *fixed average payload* of m bits while minimizing the average distortion,

$$\underset{\pi}{\text{minimize}}\ E_\pi[D] \qquad \text{subject to } H(\pi) = m. \tag{4}$$

- **Distortion-limited sender (DLS)**: maximize the average payload while introducing a *fixed average distortion* D_ϵ,

$$\underset{\pi}{\text{maximize}}\ H(\pi) \qquad \text{subject to } E_\pi[D] = D_\epsilon. \tag{5}$$

The problem of embedding a fixed-size message while minimizing the total distortion D (the PLS) is more commonly used in steganography when compared to the DLS. When the distortion function is content-driven, the sender may choose to maximize the payload with a constraint on the overall distortion. This DLS corresponds to a more intuitive use of steganography since images with different level of noise and texture can carry different amount of hidden payload and thus the distortion should be fixed instead of the payload (as long as the distortion corresponds to statistical detectability). The fact that the payload is driven by the image content is essentially a case of the batch-steganography paradigm [20].

A. Performance Bounds and Comparison Metrics

Both embedding problems described above bear relationship to the problem of source coding with a fidelity criterion as described by Shannon [21] and the problem of source coding with side information available at the transmitter, the so-called Gel'fand–Pinsker problem [19]. Problems (4) and (5) are dual to each other, meaning that the optimal distribution for the first problem is, for some value of D_ϵ, also optimal for the second one. Following the maximum entropy principle [22, Th. 12.1.1], the optimal solution has the form of a Gibbs distribution (see [8, App. A] for derivation):

$$\pi(\mathbf{y}) = \frac{\exp\left(-\lambda D(\mathbf{y})\right)}{Z(\lambda)} \overset{(a)}{=} \prod_{i=1}^{n} \frac{\exp\left(-\lambda \rho_i(y_i)\right)}{Z_i(\lambda)} \triangleq \prod_{i=1}^{n} \pi_i(y_i) \tag{6}$$

where the parameter $\lambda \in [0, \infty)$ is obtained from the corresponding constraints (4) or (5) by solving an algebraic equation[2]; $Z(\lambda) = \sum_{\mathbf{y}\in\mathcal{Y}} \exp(-\lambda D(\mathbf{y}))$, $Z_i(\lambda) = \sum_{y_i \in \mathcal{I}_i} \exp(-\lambda \rho_i(y_i))$ are the corresponding partition functions. Step (a) follows from the additivity of D, which also leads to mutual independence of individual stego pixels y_i given $\mathbf{x}$.

By changing each pixel i with probability π_i (6) one can *simulate* embedding with optimal π. This is important for steganography developers who can test the security of a scheme that uses the pair $\{\mathcal{Y}, \pi\}$ using blind steganalysis without having to implement a practical embedding algorithm. The simulator of optimal embedding can also be used to assess the increase in statistical detectability of a practical (suboptimal) algorithm w.r.t. to the optimal one. This separation principle [7] simplifies the search for better distortion measures since only the most promising approaches can be implemented. In Section VII, we use the simulators to benchmark different coding algorithms we develop in this paper by comparing the security of practical schemes using blind steganalysis.

An established way of evaluating coding algorithms in steganography is to compare the *embedding efficiency* $e(\alpha) = \alpha n / E_\pi[D]$ (in bits per unit distortion) for a fixed expected relative payload $\alpha = m/n$ with the upper bound derived from (6). When the number of changes is minimized, e is the average number of bits hidden per embedding change. For general functions ρ_i, the interpretation of this metric becomes less clear. A different and more easily interpretable metric is to compare the payload, m, of an embedding algorithm w.r.t. the payload, m_{MAX}, of the optimal DLS for a fixed D_ϵ

$$l(D_\epsilon) = \frac{m_{\text{MAX}} - m}{m_{\text{MAX}}} \tag{7}$$

which we call the *coding loss*.

B. Binary Embedding Operation

In this section, we show that for binary embedding operations, it is enough to consider a slightly narrower class of distortion functions without experiencing any loss of generality. The binary case is very important as the embedding method introduced in this paper is first developed for this special case and then extended to nonbinary operations.

For binary embedding with $\mathcal{I}_i = \{x_i, \overline{x}_i\}$, $x_i \neq \overline{x}_i$, we define $\rho_i^{\min} = \min\{\rho_i(\mathbf{x}, x_i), \rho_i(\mathbf{x}, \overline{x}_i)\}$, $\varrho_i = |\rho_i(\mathbf{x}, x_i) - \rho_i(\mathbf{x}, \overline{x}_i)| \geq 0$, and rewrite (1) as

$$D(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{n} \rho_i^{\min} + \sum_{i=1}^{n} \varrho_i \cdot \left[\rho_i^{\min} < \rho_i(\mathbf{x}, y_i)\right]. \tag{8}$$

Because the first sum does not depend on $\mathbf{y}$, when minimizing D over $\mathbf{y}$ it is enough to consider only the second term. It now becomes clear that embedding in cover $\mathbf{x}$ while minimizing (8) is equivalent to embedding in cover $\mathbf{z}$

$$z_i = \begin{cases} x_i & \text{when } \rho_i^{\min} = \rho_i(\mathbf{x}, x_i) \\ \overline{x}_i & \text{when } \rho_i^{\min} = \rho_i(\mathbf{x}, \overline{x}_i) \end{cases} \tag{9}$$

while minimizing

$$\tilde{D}(\mathbf{z}, \mathbf{y}) = \sum_{i=1}^{n} \tilde{\rho}_i(\mathbf{z}, y_i) \triangleq \sum_{i=1}^{n} \varrho_i \cdot [y_i \neq z_i], \tag{10}$$

with nonnegative costs $\tilde{\rho}_i(\mathbf{z}, z_i) = 0 \leq \tilde{\rho}_i(\mathbf{z}, \overline{z}_i) = \varrho_i$ for all i (when the cover pixel z_i is changed to $\bar{z}_i$, the distortion $\tilde{D}$ always increases). Thus, from now on for binary embedding operations, we will always consider distortion functions of the form:

$$D(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{n} \varrho_i \cdot [y_i \neq x_i], \tag{11}$$

with $\varrho_i \geq 0$.

For example, F5 [23] uses the distortion function (11) with $\varrho_i = 1$ (the number of embedding changes), while nsF5 [16] employs wet paper codes, where $\varrho_i \in \{1, \infty\}$. In some embedding algorithms [2], [4], [24], where the cover is preprocessed

[2]A simple binary search will do the job because both $H(\pi)$ and $E_\pi[D]$ are monotone w.r.t. λ.

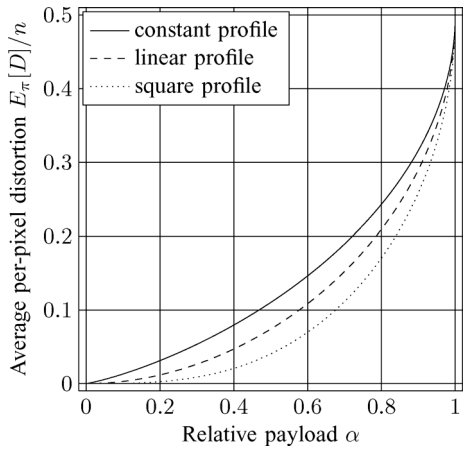


Fig. 1. Lower bound on the average per-pixel distortion, $E_\pi[D]/n$, as a function of relative payload α for different distortion profiles.

and quantized before embedding, ϱ_i is proportional to the quantization error at pixel x_i.

Additionally, for binary embedding operations we speak of a *distortion profile* ϱ if $\varrho_i = \varrho(i/n)$ for all i, where ϱ is a nondecreasing[3] function $\varrho : [0,1] \to [0,K]$. The following distortion profiles are of interest in steganography (this is not an exhaustive list): the *constant profile*, $\varrho(x) = 1$, when all pixels have the same impact on detectability when changed; the *linear profile*, $\varrho(x) = 2x$, when the distortion is related to a quantization error uniformly distributed on $[-Q/2, Q/2]$ for some quantization step $Q > 0$; and the *square profile*, $\varrho(x) = 3x^2$, which can be encountered when the distortion is related to a quantization error that is not uniformly distributed.

In this paper, we normalize the profile ϱ so that $E_\pi[D]/n = \sum_{i=1}^{n} \pi_i \varrho_i / n = 0.5$ when embedding a full payload $m = n$. With this convention, Fig. 1 displays the lower bounds on the average per-pixel distortion for three distortion profiles.

In practice, some cover pixels may require $\mathcal{I}_i = \{x_i\}$ and thus $\varrho_i = \infty$ (the so-called *wet pixels* [16], [24], [25]) to prevent the embedding algorithm from modifying them. Since such pixels are essentially constant, in this case we measure the relative payload α with respect to the set of *dry pixels* $\{x_i | \varrho_i < \infty\}$, i.e., $\alpha = m/|\{x_i|\varrho_i < \infty\}|$. The overall channel is called the wet paper channel and it is characterized by the profile ϱ of dry pixels and *relative wetness* $\tau = |\{x_i|\varrho_i = \infty\}|/n$. The wet paper channel is often required when working with images in the JPEG domain [16].

IV. Syndrome Coding

The PLS and the DLS can be realized in practice using a general methodology called *syndrome coding*. In this section, we briefly review this approach and its history paving our way to Section V and VI, where we explain the main contribution of this paper—the syndrome-trellis codes.

Let us first assume a binary version of both embedding problems. Let $\mathcal{P} : \mathcal{I}_i \to \{0,1\}$ be a parity function shared between the sender and the receiver satisfying $\mathcal{P}(x_i) \neq \mathcal{P}(y_i)$ such as $\mathcal{P}(x) = x \bmod 2$. The sender and the receiver need to implement the embedding and extraction mappings defined as $\mathrm{Emb} : \mathcal{X} \times \{0,1\}^m \to \mathcal{Y}$ and $\mathrm{Ext} : \mathcal{Y} \to \{0,1\}^m$ satisfying

$$\mathrm{Ext}\left(\mathrm{Emb}(\mathbf{x}, \mathbf{m})\right) = \mathbf{m} \quad \forall \mathbf{x} \in \mathcal{X}, \forall \mathbf{m} \in \{0,1\}^m,$$

respectively. In particular, we do not assume the knowledge of the distortion function D at the receiver and thus the embedding scheme can be seen as being universal in this sense. A common information-theoretic strategy for solving the PLS problem is known as binning [26], which we implement using cosets of a linear code. Such a construction, better known as syndrome coding, is capacity achieving for the PLS problem if random linear codes are used.

In syndrome coding, the embedding and extraction mappings are realized using a binary linear code $\mathcal{C}$ of length n and dimension $n - m$:

$$\mathrm{Emb}(\mathbf{x}, \mathbf{m}) = \arg \min_{\mathcal{P}(\mathbf{y}) \in \mathcal{C}(\mathbf{m})} D(\mathbf{x}, \mathbf{y}) \tag{12}$$

$$\mathrm{Ext}(\mathbf{y}) = \mathbb{H}\mathcal{P}(\mathbf{y}) \tag{13}$$

where $\mathcal{P}(\mathbf{y}) = (\mathcal{P}(y_1), \ldots, \mathcal{P}(y_n)), \mathbb{H} \in \{0,1\}^{m \times n}$ is a parity-check matrix of the code $\mathcal{C}$, $\mathcal{C}(\mathbf{m}) = \{\mathbf{z} \in \{0,1\}^n | \mathbb{H}\mathbf{z} = \mathbf{m}\}$ is the coset corresponding to syndrome $\mathbf{m}$, and all operations are in binary arithmetic.

Unfortunately, random linear codes are not practical due to the exponential complexity of the optimal binary coset quantizer (12), which is the most challenging part of the problem. In this work, we describe a rich class of codes for which the quantizer can be solved optimally with linear time and space complexity w.r.t. n.

Since the DLS is a dual problem to the PLS, it can be solved by (12) and (13) once an appropriate message size m is known. This can be obtained in practice by $m = m_{\mathrm{MAX}}(1 - l')$, where $m_{\mathrm{MAX}} = H(\pi_\lambda)$ is the maximal average payload obtained from the optimal distribution (6) achieving average distortion D_ϵ and l' is an experimentally obtained coding loss we expect the algorithm will achieve.

One possible approach for solving a nonbinary version of both embedding problems is to increase the size of the alphabet and use (12) and (13) with a nonbinary code $\mathcal{C}$, such as the ternary Hamming code. A more practical alternative with lower complexity is the multilayered construction proposed in Section VI, which decomposes (12) and (13) into a series of binary embedding subproblems. Such decomposition leads to the optimal solution of PLS and DLS as long as each binary subproblem is solved optimally. For this reason, in Section V we focus on the binary PLS problem for a large variety of relative payloads and different distortion profiles including the wet paper channel.

A. Prior Art

The problem of minimizing the embedding impact in steganography, introduced above as the PLS problem, has been already conceptually described by Crandall [27] in his essay posted on the steganography mailing list in 1998. He suggested that whenever the encoder embeds at most one bit per pixel, it should make use of the embedding impact defined for every pixel and minimize its total sum:

[3]By reindexing the pixels, we can indeed assume that $\varrho_1 \leq \varrho_2 \leq \cdots \leq \varrho_n \leq K$.

> "Conceptually, the encoder examines an area of the image and weights each of the options that allow it to embed the desired bits in that area. It scores each option for how conspicuous it is and chooses the option with the best score."

Later, Bierbrauer [28], [29] studied a special case of this problem and described a connection between codes (not necessarily linear) and the problem of minimizing the number of changed pixels (the constant profile). This connection, which has become known as matrix embedding (encoding), was made famous among steganographers by Westfeld [23] who incorporated it in his F5 algorithm. A binary Hamming code was used to implement the syndrome-coding scheme for the constant profile. Later on, different authors suggested other linear codes, such as Golay [30], BCH [31], random codes of small dimension [32], and nonlinear codes based on the idea of a blockwise direct sum [29]. Current state-of-the-art methods use codes based on low density generator matrices (LDGMs) [8] in combination with the ZZW construction [15]. The embedding efficiency of these codes stays rather close to the bound for arbitrarily small relative payloads [33].

The versatile syndrome-coding approach can also be used to communicate via the wet paper channel using the so-called wet paper codes [24]. Wet paper codes minimizing the number of changed dry pixels were described in [13], [14], [31], and [34].

Even though other distortion profiles, such as the linear profile, are of great interest to steganography, no general solution with performance close to the bound is currently known. The authors of [2] approached the PLS problem by minimizing the distortion on a block-by-block basis utilizing a Hamming code and a suboptimal quantizer implemented using a brute-force search that allows up to three embedding changes. Such an approach, however, provides highly suboptimal performance far from the theoretical bound (see Fig. 8). A similar approach based on BCH codes and a brute-force quantizer was described in [4] achieving a slightly better performance than Hamming codes. Neither Hamming or BCH codes can be used to deal with the wet paper channel without significant performance loss. To the best of our knowledge, no solution is known that could be used to solve the PLS problem with arbitrary distortion profile containing wet pixels.

One promising direction towards replacing the random linear codes while keeping the optimality of the construction has recently been proposed by Arikan [35], who introduced the so-called polar codes for the channel coding problem. One advantage is that the complexity of encoding and decoding algorithms for polar codes is $n \log n$. Moreover, most of the capacity-achieving properties of random linear codes are retained even for other information-theoretic problems and thus polar codes are known to be optimal for the PLS problem [36] (at least for the uniform profile). Unfortunately, to apply such codes, the number of pixels, n, must be very high, which may not be always satisfied in practice. We believe that the proposed syndrome-trellis codes offer better tradeoffs when used in practical embedding schemes.

V. Syndrome-Trellis Codes

In this section, we focus on solving the binary PLS problem with distortion function (10) and modify a standard trellis-coding strategy for steganography. The resulting codes are called the syndrome-trellis codes. These codes will serve as a building block for nonbinary PLS and DLS problems in Section VI.

The construction behind STCs is not new from an information-theoretic perspective, since the STCs are convolutional codes represented in a dual domain. However, STCs are very interesting for practical steganography since they allow solving both embedding problems with a very small coding loss over a wide range of distortion profiles even with wet pixels. The same code can be used with all profiles making the embedding algorithm practically universal. STCs offer general and state-of-the-art solution for both embedding problems in steganography. Here, we give the description of the codes along with their graphical representation, the syndrome trellis. Such construction is prepared for the Viterbi algorithm, which is optimal for solving (12). Important practical guidelines for optimizing the codes and using them for the wet paper channel are also covered. Finally, we study the performance of these codes by extensive numerical simulations using different distortion profiles including the wet paper channel.

Syndrome-trellis codes targeted to applications in steganography were described in [17], which was written for practitioners. In this paper, we expect the reader to have a working knowledge of convolutional codes which are often used in data-hiding applications such as digital watermarking. Convolutional codes are otherwise described in [37, Ch. 25 and 48]. For a complete example of the Viterbi algorithm used in the context of STCs, we refer the reader to [17].

Our main goal is to develop efficient syndrome-coding schemes for an *arbitrary* relative payload α with the main focus on small relative payloads (think of $\alpha \leq 1/2$ for example). In steganography, the relative payload must decrease with increasing size of the cover object in order to maintain the same level of security, which is a consequence of the square root law [38]. Moreover, recent results from steganalysis in both spatial [39] and DCT domains [40] suggest that the secure payload for digital image steganography is always far below 1/2. Another reason for targeting smaller payloads is the fact that as $\alpha \to 1$, all binary embedding algorithms tend to introduce changes with probability 1/2, no matter how optimal they are. Denoting with $R = (n-m)/n$ the rate of the linear code $\mathcal{C}$, then $\alpha \to 0$ translates to $R = 1 - \alpha \to 1$, which is characteristic for applications of syndrome coding in steganography.

A. From Convolutional Codes to Syndrome-Trellis Codes

Since Shannon [21] introduced the problem of source coding with a fidelity criterion in 1959, convolutional codes were probably the first "practical" codes used for this problem [41]. This is because the gap between the bound on the expected per-pixel distortion and the distortion obtained using the optimal encoding algorithm (the Viterbi algorithm) decreases exponentially with the constraint length of the code [41], [42]. The complexity of the Viterbi algorithm is linear in the block length of the code, but exponential in its constraint length (the number of trellis states grows exponentially in the constraint length).

When adapted to the PLS problem, convolutional codes can be used for syndrome coding since the best stego image in (12) can be found using the Viterbi algorithm. This makes convolu-

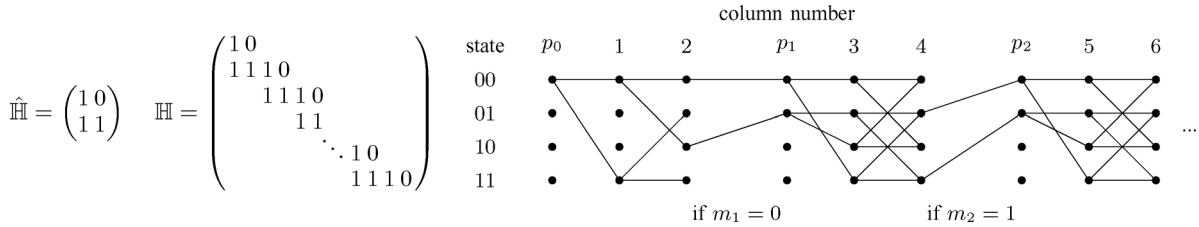


Fig. 2. Example of a parity-check matrix $\mathbb{H}$ formed from the submatrix $\hat{\mathbb{H}}$ ($h = 2, w = 2$) and its corresponding syndrome trellis. The last $h - 1$ submatrices in $\mathbb{H}$ are cropped to achieve the desired relative payload α. The syndrome trellis consists of repeating blocks of $w + 1$ columns, where "p_0" and "p_i", $i > 0$, denote the starting and pruning columns, respectively. The column labeled $l \in \{1, 2, \ldots\}$ corresponds to the lth column in the parity-check matrix $\mathbb{H}$.

tional codes (of small constraint length) suitable for our application because the entire cover object can be used and the speed can be traded for performance by adjusting the constraint length. Note that the receiver does not need to know D since only the Viterbi algorithm requires this knowledge. By increasing the constraint length, we can achieve the average per-pixel distortion that is arbitrarily close to the bounds and thus make the coding loss (7) approach zero. Convolutional codes are often represented with shift-registers (see [37, Ch. 48]) that generate the codeword from a set of information bits. In channel coding, codes of rates $R = 1/k$ for $k = 2, 3, \ldots$ are usually considered for their simple implementation.

Convolutional codes in standard trellis representation are commonly used in problems that are dual to the PLS problem, such as the distributed source coding [43]. The main drawback of convolutional codes, when implemented using shift-registers, comes from our requirement of small relative payloads (code rates close to one) which is specific to steganography. A convolutional code of rate $R = (k - 1)/k$ requires $k - 1$ shift registers in order to implement a scheme for $\alpha = 1/k$. Here, unfortunately, the complexity of the Viterbi algorithm in this construction grows exponentially with k. Instead of using puncturing (see [37, Ch. 48]), which is often used to construct high-rate convolutional codes, we prefer to represent the convolutional code in the dual domain using its parity-check matrix. In fact, Sidorenko and Zyablov [44] showed that optimal decoding of convolutional codes (our binary quantizer) with rates $R = (k - 1)/k$ can be carried out in the dual domain on the syndrome trellis with a much lower complexity and without any loss of performance. This approach is more efficient as $\alpha \to 0$ and thus we choose it for the construction of the codes presented in this paper.

In the dual domain, a code of length n is represented by a parity-check matrix instead of a generator matrix as is more common for convolutional codes. Working directly in the dual domain allows the Viterbi algorithm to exactly implement the coset quantizer required for the embedding function (12). The message can be extracted in a straightforward manner by the recipient using the shared parity-check matrix.

B. Description of Syndrome-Trellis Codes

Although syndrome-trellis codes form a class of convolutional codes and thus can be described using a classical approach with shift-registers, it is advantageous to stay in the dual domain and describe the code directly by its parity-check matrix. The parity-check matrix $\mathbb{H} \in \{0, 1\}^{m \times n}$ of a binary syndrome-trellis code of length n and codimension m is obtained by placing a small submatrix $\hat{\mathbb{H}}$ of size $h \times w$ along the main diagonal as in Fig. 2. The submatrices $\hat{\mathbb{H}}$ are placed next to each other and shifted down by one row leading to a sparse and banded $\mathbb{H}$. The height h of the submatrix (called the *constraint height*) is a design parameter that affects the algorithm speed and efficiency (typically, $6 \leq h \leq 15$). The width of $\hat{\mathbb{H}}$ is dictated by the desired ratio of m/n, which coincides with the relative payload $\alpha = m/n$ when no wet pixels are present. If m/n equals to $1/k$ for some $k \in \mathbb{N}$, select $w = k$. For general ratios, find k such that $1/(k + 1) < m/n < 1/k$. The matrix $\mathbb{H}$ will contain a mix of submatrices of width k and $k + 1$ so that the final matrix $\mathbb{H}$ is of size $m \times n$. In this way, we can create a parity-check matrix for an arbitrary message and code size. The submatrix $\hat{\mathbb{H}}$ acts as an input parameter shared between the sender and the receiver and its choice is discussed in more detail in Section V-D. For the sake of simplicity, in the following description we assume $m/n = 1/w$ and thus the matrix $\mathbb{H}$ is of the size $b \times (b \cdot w)$, where b is the number of copies of $\hat{\mathbb{H}}$ in $\mathbb{H}$.

Similar to convolutional codes and their trellis representation, every codeword of an STC $\mathcal{C} = \{\mathbf{z} \in \{0, 1\}^n | \mathbb{H}\mathbf{z} = 0\}$ can be represented as a unique path through a graph called the *syndrome trellis*. Moreover, the syndrome trellis is parametrized by $\mathbf{m}$ and thus can represent members of arbitrary coset $\mathcal{C}(\mathbf{m}) = \{\mathbf{z} \in \{0, 1\}^n | \mathbb{H}\mathbf{z} = \mathbf{m}\}$. An example of the syndrome trellis is shown in Fig. 2. More formally, the syndrome trellis is a graph consisting of b blocks, each containing $2^h(w + 1)$ nodes organized in a grid of $w + 1$ columns and 2^h rows. The nodes between two adjacent columns form a bipartite graph, i.e., all edges only connect nodes from two adjacent columns. Each block of the trellis represents one submatrix $\hat{\mathbb{H}}$ used to obtain the parity-check matrix $\mathbb{H}$. The nodes in every column are called *states*.

Each $\mathbf{z} \in \{0, 1\}^n$ satisfying $\mathbb{H}\mathbf{z} = \mathbf{m}$ is represented as a path through the syndrome trellis which represents the process of calculating the syndrome as a linear combination of the columns of $\mathbb{H}$ with weights given by $\mathbf{z}$. Each path starts in the leftmost all-zero state in the trellis and extends to the right. The path shows the step-by-step calculation of the (partial) syndrome using more and more bits of $\mathbf{z}$. For example, the first two edges in Fig. 2, that connect the state 00 from column p_0 with states 11 and 00 in the next column, correspond to adding $(\mathcal{P}(y_1) = 1)$ or not adding $(\mathcal{P}(y_1) = 0)$ the first column of $\mathbb{H}$ to the syndrome, respectively.[4] At the end of the first block, we terminate all paths for which the first bit of the partial syndrome does not match m_1. This way, we obtain a new column of the trellis, which will serve as the starting column of the next block. This column merely il-

[4]The state corresponds to the partial syndrome.

Forward part of the Viterbi algorithm

```
wght[0] = 0
wght[1,...,2^h-1] = infinity
indx = indm = 1
for i = 1,...,num of blocks (submatrices in H) {
 for j = 1,...,w {                  // for each column
  for k = 0,...,2^h-1 {             // for each state
   w0 = wght[k] + x[indx]*rho[indx]
   w1 = wght[k XOR H_hat[j]] + (1-x[indx])*rho[indx]
   path[indx][k] = w1 < w0 ? 1 : 0      // C notation
   newwght[k] = min(w0, w1)
  }
  indx++
  wght = newwght
 }
  // prune states
 for j = 0,...,2^(h-1)-1
  wght[j] = wght[2*j + message[indm]]
 wght[2^(h-1),...,2^h-1] = infinity
 indm++
}
```

Backward part of the Viterbi alg.

```
embedding_cost = wght[0]
state = 0, indx--, indm--
for i = num of blocks,...,1 (step -1) {
  for j = w,...,1 (step -1) {
    y[indx] = path[indx][state]
    state = state XOR (y[indx]*H_hat[j])
    indx--
  }
  state = 2*state + message[indm]
  indm--
}
```

Legend

```
INPUT: x, message, H_hat
  x = (x[1],...,x[n]) cover object
  message = (message[1],...,message[m])
  H_hat[j] = j th column in int notation

OUTPUT: y, embedding_cost
  y = (y[1],...,y[n]) stego object
```

Fig. 3. Pseudocode of the Viterbi algorithm modified for the syndrome trellis.

lustrates the transition of the trellis from representing the partial syndrome $(s_1, \ldots, s_h)$ to $(s_2, \ldots, s_{h+1})$. This operation is repeated at each block transition in the matrix $\mathbb{H}$ and guarantees that 2^h states are sufficient to represent the calculation of the partial syndrome throughout the whole syndrome trellis.

To find the closest stego object, we assign weights to all trellis edges. The weights of the edges entering the column with label l, $l \in \{1, \ldots, n\}$, in the syndrome trellis depend on the lth bit representation of the original cover object $\mathbf{x}$, $\mathcal{P}(x_l)$. If $\mathcal{P}(x_l) = 0$, then the horizontal edges (corresponding to not adding the lth column of $\mathbb{H}$) have a weight of 0 and the edges corresponding to adding the lth column of $\mathbb{H}$ have a weight of ϱ_l. If $\mathcal{P}(x_l) = 1$, the roles of the edges are reversed. Finally, all edges connecting the individual blocks of the trellis have zero weight.

The embedding problem (12) for binary embedding can now be optimally solved by the *Viterbi algorithm* with time and space complexity $\mathcal{O}(2^h n)$. This algorithm consists of two parts, the *forward* and the *backward* part. The forward part of the algorithm consists of $n + b$ steps. Upon finishing the ith step, we know the shortest path between the leftmost all-zero state and every state in the ith column of the trellis. Thus, in the final, $n + b$th step, we discover the shortest path through the entire trellis. During the backward part, the shortest path is traced back and the parities of the closest stego object $\mathcal{P}(\mathbf{y})$ are recovered from the edge labels. The Viterbi algorithm modified for the syndrome trellis is described in Fig. 3 using a pseudocode.

C. Implementation Details

The construction of STCs is not constrained to having to repeat the same submatrix $\hat{\mathbb{H}}$ along the diagonal. Any parity-check matrix $\mathbb{H}$ containing at most h nonzero entries along the main diagonal will have an efficient representation by its syndrome trellis and the Viterbi algorithm will have the same complexity $\mathcal{O}(2^h n)$. In practice, the trellis is built on the fly because only the structure of the submatrix $\hat{\mathbb{H}}$ is needed (see the pseudocode in Fig. 3). As can be seen from the last two columns of the trellis in Fig. 2, the connectivity between trellis columns is highly regular which can be used to speed up the implementation by "vectorizing" the calculations.

In the forward part of the algorithm, we need to store one bit (the label of the incoming edge) to be able to reconstruct the path in the backward run. This space complexity is linear and should not cause any difficulty, since for $h = 10$, $n = 10^6$, the total of $2^{10} \cdot 10^6/8$ bytes ($\approx$ 122 MB) of space is required. If less space is available, we can always run the algorithm on smaller blocks, say $n = 10^4$, without any noticeable performance drop. If we are only interested in the total distortion $D(\mathbf{y})$ and not the stego object itself, this information does not need to be stored at all and only the forward run of the Viterbi algorithm is required.

D. Design of Good Syndrome-Trellis Codes

A natural question regarding practical applications of syndrome-trellis codes is how to optimize the structure of $\hat{\mathbb{H}}$ for fixed parameters h and w and a given profile. If $\hat{\mathbb{H}}$ depended on the distortion profile, the profile would have to be somehow communicated to the receiver. Fortunately, this is not the case and a submatrix $\hat{\mathbb{H}}$ optimized for one profile seems to be good for other profiles as well. In this section, we study these issues experimentally and describe a practical algorithm for obtaining good submatrices.

Let us suppose that we wish to design a submatrix $\hat{\mathbb{H}}$ of size $h \times w$ for a given constraint height h and relative payload $\alpha = 1/w$. In [45], authors describe several methods for calculating the expected distortion of a given convolutional code when used in the source-coding problem with Hamming measure (uniform distortion profile). Unfortunately, the computational complexity of these algorithms do not permit us to use them for the code design. Instead, we rely on estimates obtained from embedding a pseudo-random message into a random cover object. The authors were unable to find a better algorithm than an exhaustive search guided by some simple design rules.

First, $\hat{\mathbb{H}}$ should not have identical columns because the syndrome trellis would contain two or more different paths with exactly the same weight, which would lead to an overall decrease

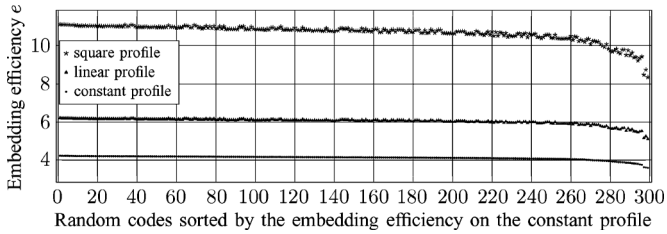


Fig. 4. Embedding efficiency of 300 random syndrome-trellis codes satisfying the design rules for relative payload $\alpha = 1/2$ and constraint height $h = 10$. All codes were evaluated by the Viterbi algorithm with a random cover object of $n = 10^6$ pixels and a random message on the constant, linear, and square profiles. Codes are shown in the order determined by their embedding efficiency evaluated on the constant profile. This experiment suggests that codes good for the constant profile are good for other profiles. Codes designed for different relative payloads have a similar behavior.

in performance. By running an exhaustive search over small matrices, we have observed that the best submatrices $\hat{\mathbb{H}}$ had ones in the first and last rows. For example, when $h = 7$ and $w = 4$, more than 97% of the best 1000 codes obtained from the exhaustive search satisfied this rule. Thus, we searched for good matrices among those that did not contain identical columns and with all bits in the first and last rows set to 1 (the remaining bits were assigned at random). In practice, we randomly generated 10–1000 submatrices satisfying these rules and estimated their performance (embedding efficiency) experimentally by running the Viterbi algorithm with random covers and messages. For a reliable estimate, cover objects of size at least $n = 10^6$ are required.

To investigate the stability of the design w.r.t. to the profile, the following experiment was conducted. We fixed $h = 10$ and $w = 2$, which correspond to a code with $\alpha = 1/2$. The code design procedure was simulated by randomly generating 300 submatrices $\hat{\mathbb{H}}_1, \ldots, \hat{\mathbb{H}}_{300}$ satisfying the above design rules. The goodness of the code was evaluated using the embedding efficiency ($e = m/D(\mathbf{x}, \mathbf{y})$) by running the Viterbi algorithm on a random cover object (of size $n = 10^6$) and with a random message. This was repeated independently for all three profiles from Section III-B. Fig. 4 shows the embedding efficiency after ordering all 300 codes by their performance on the constant profile. Because the codes with a high embedding efficiency on the constant profile exhibit high efficiency for the other profiles, we consider the code design to be stable w.r.t. the profile and use these matrices with other profiles in practice. All further results are generated by using these matrices.

E. Wet Paper Channel

In this section, we investigate how STCs can be used for the wet paper channel described by relative wetness $\tau = |\{i|\varrho_i = \infty\}|/n$ with a given distortion profile of dry pixels. Although the STCs can be directly applied to this problem, the probability of not being able to embed a message without changing any wet pixel may be positive and depends on the number of wet pixels, the payload, and the code. The goal is to make this probability very small or to make sure that the number of wet pixels that must be changed is small (e.g., one or two). We now describe two different approaches to address this problem.

Let us assume that the wet channel is i.i.d. with probability of a pixel being wet $0 \leq \tau < 1$. This assumption is plausible

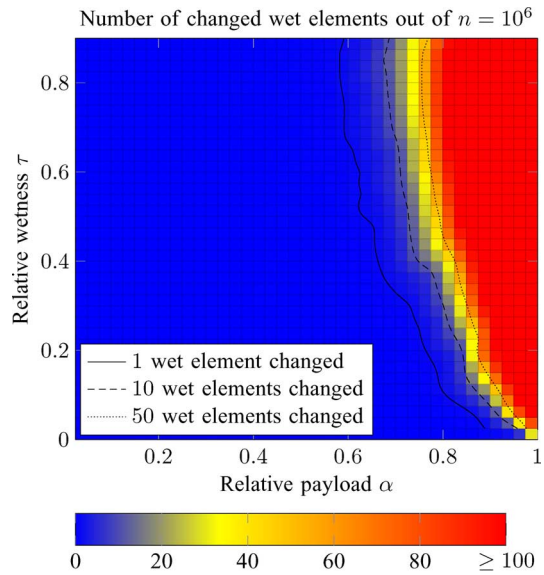


Fig. 5. Average number of wet pixels out of $n = 10^6$ that need to be changed to find a solution to (12) using STCs with $h = 11$.

because the cover pixels can be permuted using a stego key before embedding. For the wet paper channel, the relative payload is defined w.r.t. the dry pixels as $\alpha = m/|\{i|\rho_i < \infty\}|$. When designing the code for the wet paper channel with n-pixel covers, relative wetness τ, and desired relative payload α, the parity-check matrix $\mathbb{H}$ has to be of the size $[(1-\tau)\alpha n] \times n$.

The random permutation makes the Viterbi algorithm less likely to fail to embed a message without having to change some wet pixels. The probability of failure, p_{w}, decreases with decreasing α and τ and it also depends on the constraint height h. From practical experiments with $n = 10^6$ cover pixels, $\tau = 0.8$, and $h = 10$, we estimated from 1000 independent runs $p_{\mathrm{w}} \doteq 0.24$ for $\alpha = 1/2$, $p_{\mathrm{w}} \doteq 0.009$ for $\alpha = 1/4$, and $p_{\mathrm{w}} \doteq 0$ for $\alpha = 1/10$. In practice, the message size m can be used as a seed for the pseudo-random number generator. If the embedding process fails, embedding $m - 1$ bits leads to a different permutation while embedding roughly the same amount of message. In k trials, the probability of having to modify a wet pixel is at most p_{w}^k, which can be made arbitrarily small.

Alternatively, the sender may allow a small number of wet pixels to be modified, say one or two, without affecting the statistical detectability in any significant manner. Making use of this fact, one can set the distortion of all wet cover pixels to $\hat{\varrho}_i = C$, $C > \sum_{\varrho_i < \infty} \varrho_i$ and $\hat{\varrho}_i = \varrho_i$ for i dry. The weight c of the best path through the syndrome trellis obtained by the Viterbi algorithm with distortion $\hat{\varrho}_i$ can be written in the form $c = n_{\mathrm{c}} C + c'$, where n_{c} is the smallest number of wet cover pixels that had to be changed and c' is the smallest weight of the path over the pixels that are allowed to be changed.

Fig. 5 shows the average number of wet pixels out of $n = 10^6$ required to be changed in order to solve (12) for STCs with $h = 11$. The exact value of ϱ_i is irrelevant in this experiment as long as it is finite. This experiment suggests that STCs can be used with arbitrary τ as long as $\alpha \leq 0.7$. As can be seen from Fig. 6, increasing the amount of wet pixels does not lead to any noticeable difference in embedding efficiency for constant

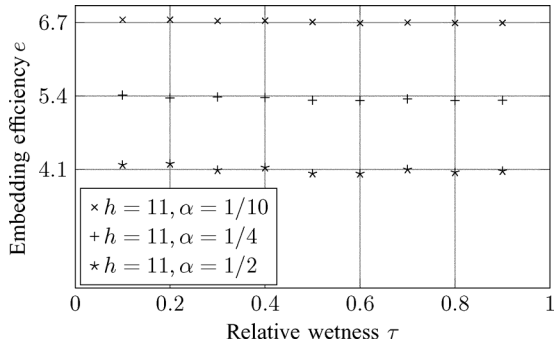


Fig. 6. Effect of relative wetness τ of the wet paper channel with a constant profile on the embedding efficiency of STCs. The distortion was calculated w.r.t. the changed dry pixels only and $\alpha = m/(n - \tau n)$. Each point was obtained by quantizing a random vector of $n = 10^6$ pixels.

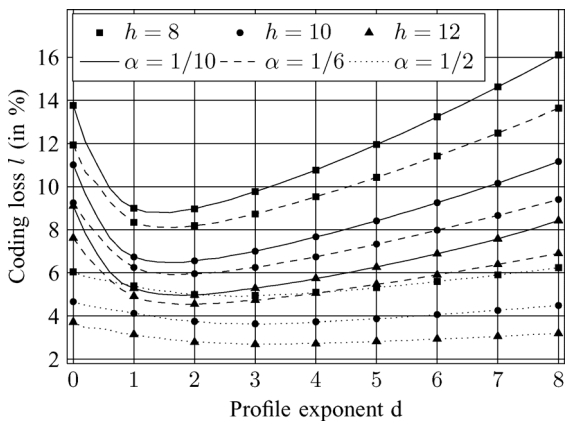


Fig. 7. Comparison of the coding loss of STCs as a function of the profile exponent d for different payloads and constraint heights of STCs. Each point was obtained by quantizing a random vector of $n = 10^6$ pixels.

profile. Similar behavior has been observed for other profiles and holds as long as the number of changed wet pixels is small.

F. Experimental Results

We have implemented the Viterbi algorithm in C++ and optimized its performance by using Streaming SIMD Extensions instructions. Based on the distortion profile, the algorithm chooses between the float and 1 byte unsigned integer data type to represent the weight of the paths in the trellis. The following results were obtained using an Intel Core2 X6800 2.93 GHz CPU machine utilizing a single CPU core.

Using the search described in Section V-D, we found good syndrome-trellis codes of constraint height $h \in \{6, \dots, 12\}$ for relative payloads $\alpha = 1/w$, $w \in \{1, \dots, 20\}$. Some of these codes can be found in [17, Table 1]. In practice, almost every code satisfying the design rules is equally good. This fact can also be seen from Fig. 4, where 300 random codes are evaluated over different profiles.

The effect of the profile shape on the coding loss for $\varrho(x) \approx x^d$ as a function of d is shown in Fig. 7. The coding loss increases with decreasing relative payload α. This effect can be compensated by using a larger constraint height h.

Fig. 8 shows the comparison of syndrome-trellis codes for three profiles with other codes which are known for a given profile. The ZZW family [12] applies only to the constant profile. For a given relative payload α and constraint height h, the same submatrix $\hat{\mathbb{H}}$ was used for all profiles. This demonstrates the versatility of the proposed construction, since the information about the profile does not need to be shared, or, perhaps more importantly, the profile does not need to be known *a priori* for a good performance.

Fig. 9 shows the average throughput (the number of cover pixels n quantized per second) based on the used data type. In practice, 1–5 seconds were enough to process a cover object with $n = 10^6$ pixels. In the same figure, we show the embedding efficiency obtained from very short codes for the constant profile. This result shows that the average performance of syndrome-trellis codes quickly approaches its maximum w.r.t. n. This is again an advantage, since some applications may require short blocks.

G. STCs in Context of Other Works

The concept of dividing a set of samples into different bins (the so-called binning) is a common tool used for solving many information-theoretic and also data-hiding problems [26]. From this point of view, the steganographic embedding problem is a pure source-coding problem, i.e., given cover $\mathbf{x}$, what is the "closest" stego object $\mathbf{y}$ in the bin indexed by the message. In digital watermarking, the same problem is extended by an attack channel between the sender and the receiver, which calls for a combination of good source and channel codes. This combination can be implemented using nested convolutional (trellis) codes and is better known as Dirty-paper codes [46]. Different practical application of the binning concept is in the distributed source coding problem [43]. Convolutional codes are attractive for solving these problems mainly because of the existence of the optimal quantizer—the Viterbi algorithm.

VI. Multilayered Construction

Although it is straightforward to extend STCs to nonbinary alphabets and thus apply them to q-ary embedding operations, their complexity rapidly increases (the number of states in the trellis increases from 2^h to q^h for constraint height h), limiting thus their performance in practice. In this section, we introduce a simple layered construction which has been largely motivated by [10] and can be considered as a generalization of this work. The main idea is to decompose the problems (4) and (5) with a nonbinary embedding operation into a sequence of similar problems with a binary embedding operation. Any solution to the binary PLS embedding problem, such as STCs, can then be used. This decomposition turns out to be optimal if each binary embedding problem is solved optimally. The multilayered construction was described in [18].

According to (11), the binary coding algorithm for (4) or (5) is optimal if and only if it modifies each cover pixel with probability

$$\pi_i = \frac{\exp(-\lambda \varrho_i)}{1 + \exp(-\lambda \varrho_i)}. \tag{14}$$

For a fixed value of λ, the values ϱ_i, $i = 1, \dots, n$, form sufficient statistic for π.

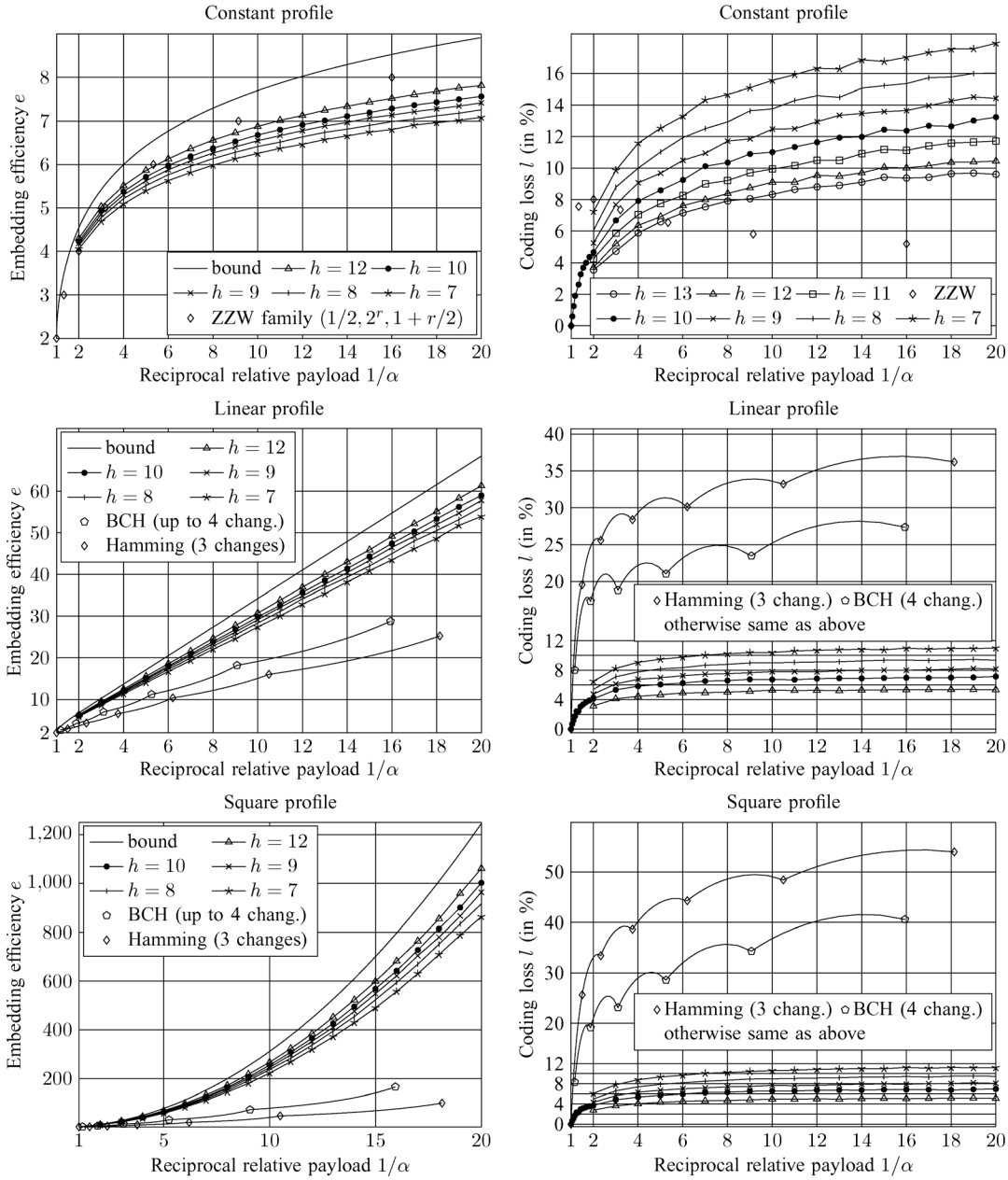


Fig. 8. Embedding efficiency and coding loss of syndrome-trellis codes for three distortion profiles. Each point was obtained by running the Viterbi algorithm with $n = 10^6$ cover pixels. Hamming [2] and BCH [3] codes were applied on a block-by-block basis on cover objects with $n = 10^5$ pixels with a brute-force search making up to three and four changes, respectively. The line connecting a pair of Hamming or BCH codes represents the codes obtained by their block direct sum. For clarity, we present the coding loss results in range $\alpha \in [0.5, 1]$ only for constraint height $h = 10$ of the syndrome-trellis codes.

A solution to the PLS with a binary embedding operation can be used to derive the following "Flipping lemma" that we will heavily use later in this section.

Lemma 1 (Flipping Lemma): Given a set of probabilities $\{p_i\}_{i=1}^n$, the sender wants to communicate $m = \sum_{i=1}^n h(p_i)$ bits by sending bit strings $\mathbf{y} = \{y_i\}_{i=1}^n$ such that $P(y_i = 0) = p_i$. This can be achieved by a PLS with a binary embedding operation on $\mathcal{I} = \mathcal{I}_i = \{0, 1\}$ for all i by embedding the payload in cover $x_i = [p_i < 1/2]$ with nonnegative per-pixel costs $\varrho_i = \ln(\tilde{p}_i/(1-\tilde{p}_i))$, $\tilde{p}_i = \max\{p_i, 1-p_i\}$.

Proof: Without loss of generality, let $\lambda = 1$. Since the inverse of $f(z) = \ln(z/(1-z))$ on [0,1] is $f^{-1}(z) = \exp(z)/(1+\exp(z))$, by (14) the cost ϱ_i causes x_i to change to $y_i = 1 - x_i$ with probability $P(y_i \neq x_i | x_i) = f^{-1}(-\varrho_i) = 1 - \tilde{p}_i$. Thus, $P(y_i = 0 | x_i = 1) = f^{-1}(-\varrho_i) = p_i$ and $P(y_i = 0 | x_i = 0) = 1 - f^{-1}(-\varrho_i) = p_i$ as required. ■

Now, let $|\mathcal{I}_i| = 2^L$ for some integer $L \geq 0$ and let $\mathcal{P}_1, \ldots, \mathcal{P}_L$ be parity functions uniquely describing all 2^L elements in $\mathcal{I}_i$, i.e., $(x_i \neq y_i) \Rightarrow \exists j, \mathcal{P}_j(x_i) \neq \mathcal{P}_j(y_i)$ for all $x_i, y_i \in \mathcal{I}_i$ and all $i \in \{1, \ldots, n\}$. For example, $\mathcal{P}_j(x)$ can be defined as the jth LSB of x. The individual sets $\mathcal{I}_i$ can be enlarged to satisfy the size constraint by setting the costs of added elements to ∞.

The optimal algorithm for (4) and (5) sends the stego symbols by sampling from the optimal distribution (6) with some λ. Let $\mathbf{Y}_i$ be the random variable defined over $\mathcal{I}_i$ representing the ith stego symbol. Due to the assigned parities, $\mathbf{Y}_i$ can be represented as $\mathbf{Y}_i = (Y_i^1, \ldots, Y_i^L)$ with Y_i^j corresponding to the jth parity function. We construct the embedding algorithm by

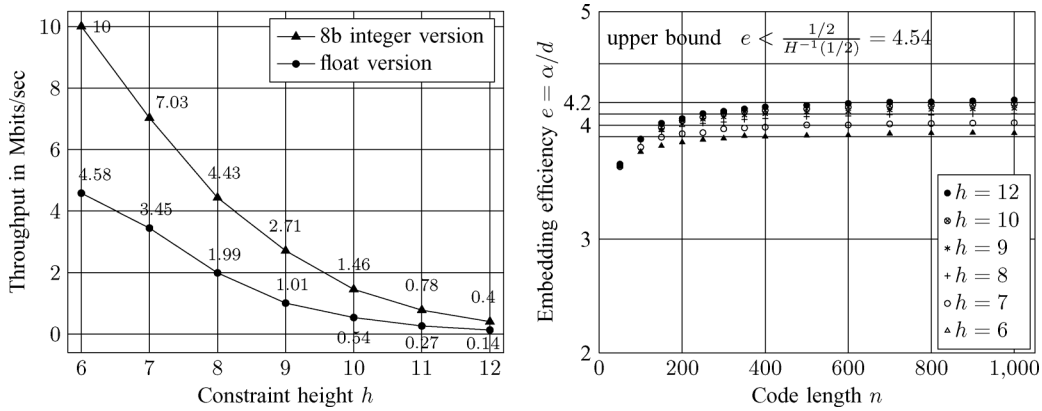


Fig. 9. Results for the syndrome-trellis codes designed for relative payload $\alpha = 1/2$. Left: Average number of cover pixels ($\times 10^6$) quantized per second (throughput) shown for different constraint heights and two different implementations. Right: Average embedding efficiency for different code lengths n (the number of cover pixels), constraint heights h, and a constant distortion profile. Codes of length $n > 1000$ have similar performance as for $n = 1000$. Each point was obtained as an average over 1000 samples.

induction over L, the number of layers. By the chain rule, for each i the entropy $H(\mathbf{Y}_i)$ can be decomposed into

$$H(\mathbf{Y}_i) = H\left(Y_i^1\right) + H\left(Y_i^2, \ldots, Y_i^L | Y_i^1\right). \qquad (15)$$

This tells us that $H(Y_i^1)$ bits should be embedded by changing the first parity of the ith pixel. In fact, the parities should be distributed according to the marginal distribution $P(Y_i^1)$. Using the Flipping lemma, this task is equivalent to a PLS, which can be realized in practice using STCs as reviewed in Section V. To summarize, in the first step we embed $m_1 = \sum_{i=1}^n H(Y_i^1)$ bits on average.

After the first layer is embedded, we obtain the parities $\mathcal{P}_1(y_i)$ for all stego pixels. This allows us to calculate the conditional probability $P(Y_i^2, \ldots, Y_i^L | Y_i^1 = \mathcal{P}_1(y_i))$ and use the chain rule again, for example w.r.t. Y_i^2. In the second layer, we embed $m_2 = \sum_{i=1}^n H(Y_i^2 | Y_i^1 = \mathcal{P}_1(y_i))$ bits on average. In total, we have L such steps fixing one parity value at a time knowing the result of the previous parities. Finally, we send the values y_i corresponding to the obtained parities.

If all individual layers are implemented optimally, we send $m = m_1 + \cdots + m_L$ bits on average. By the chain rule, this is exactly $H(\mathbf{Y}_i)$ in every pixel, which proves the optimality of this construction. In theory, the order in which the parities are being fixed can be arbitrary. As is shown in the following example, the order is important for practical realizations when STCs are used. In all our experiments, we start with the *most* significant bits ending with the LSBs. Algorithm 1 describes the necessary steps required to implement ± 1 embedding with arbitrary costs using two layers of STCs.

Algorithm 1: ± 1 Embedding Implemented With Two Layers of STCs and Embedding the Payload of m bits

Require: $\mathbf{x} \in \mathcal{X} = \{\mathcal{I}\}^n \triangleq \{0, \ldots, 255\}^n$
$\rho_i(\mathbf{x}, z) \in [-K, +K], \quad z \in \mathcal{I}_i \triangleq \{x_i - 1, x_i, x_i + 1\}$

1: define $\mathcal{P}_1(z) = z \bmod 2$, $\mathcal{P}_2(z) = [(z \bmod 4) > 1]$

2: forbid other colors by $\rho_i(\mathbf{x}, z) = C \gg K$, $z \notin \mathcal{I}_i \cap \mathcal{I}$

3: find $\lambda \geq 0$ such that distr. π over $\mathcal{X}$ satisfies $H(\pi) = m$

4:

5: define $p_i'' = Pr_\pi(\mathcal{P}_2(Y_i) = 0)$, set $m_2 = \sum_i h(p_i'')$, $\mathbf{x}'' \in \{0,1\}^n$ with $x_i'' = [p_i'' < 1/2]$, and $\varrho_i'' = |\ln(p_i''/(1 - p_i''))|$

6: **embed** m_2 bits with binary STC into $\mathbf{x}''$ with costs ϱ_i'' and produce new vector $\mathbf{y}'' = (y_1'', \ldots, y_n'') \in \{0,1\}^n$

7:

8: define $p_i' = Pr_\pi(\mathcal{P}_1(Y_i) = 0 | \mathcal{P}_2(Y_i) = y_i'')$, $\mathbf{x}' \in \{0,1\}^n$ with $x_i' = [p_i' < 1/2]$, and $\varrho_i' = |\ln(p_i'/(1 - p_i'))|$

9: **embed** $m - m_2$ bits with binary STC into $\mathbf{x}'$ with costs ϱ_i' and produce a new vector $\mathbf{y}' = (y_1', \ldots, y_n') \in \{0,1\}^n$

10:

11: set $y_i \in \mathcal{I}_i$ such that $\mathcal{P}_2(y_i) = y_i''$ and $\mathcal{P}_1(y_i) = y_i'$

12: **return** stego image $\mathbf{y} = (y_1, \ldots, y_n)$

13: message can be extracted using STCs from $(\mathcal{P}_2(y_1), \ldots, \mathcal{P}_2(y_n))$ and $(\mathcal{P}_1(y_1), \ldots, \mathcal{P}_1(y_n))$

In practice, the number of bits hidden in every layer, m_j, needs to be communicated to the receiver. The number m_j is used as a seed for a pseudo-random permutation used to shuffle all bits in the jth layer. If, due to large payload and wetness, STCs cannot embed a given message, we try a different permutation by embedding a slightly different number of bits.

Example 2 (± 1 Embedding): For simplicity, let $x_i = 2$, $\mathcal{I}_i = \{1, 2, 3\}$, $\rho_i(1) = \rho_i(3) = 1$, and $\rho_i(2) = 0$ for $i \in \{1, \ldots, n\}$ and large n. For such ternary embedding, we use two LSBs as their parities. Suppose we want to solve the problem (4) with $\alpha = 0.9217$, which leads to $\lambda = 2.08$, $P(Y_i = 1) = P(Y_i = 3) = 0.1$, and $P(Y_i = 2) = 0.8$. To make $|\mathcal{I}_i|$ a power of two, we also include the symbol 0 and define $\rho_i(0) = \infty$ which implies $P(Y_i = 0) = 0$. Let $y_i = (y_i^2, y_i^1)$ be a binary representation of $y_i \in \{0, \ldots, 3\}$, where y_i^1 is the LSB of y_i.

Starting from the LSBs as in [10], we obtain $P(Y_i^1 = 0) = 0.8$. If the LSB needs to be changed, then $P(Y_i^2 = 0 | Y_i^1 = 1) = 0.5$ whereas $P(Y_i^2 = 0 | Y_i^1 = 0) = 0$. In practice, the first layer can be realized by any syndrome-coding scheme minimizing the number of changes and embedding $m_1 = n \cdot$

$h(0.2)$ bits. The second layer must be implemented with wet paper codes [25], since we need to embed either one bit or leave the pixel unchanged (the relative payload is 1).

If the weights of symbols 1 and 3 were slightly changed, however, we would have to use STCs in the second layer, which causes a problem due to the large relative payload ($\alpha = 1$) combined with large wetness ($\tau = 0.8$) (see Fig. 5). The opposite decomposition starting with the MSB y_i^2 will reveal that $P(Y_i^2 = 0) = 0.1$, $P(Y_i^1 = 0|Y_i^2 = 0) = 0$, and $P(Y_i^1 = 0|Y_i^2 = 1) = 0.8/0.9$. Both layers can now be easily implemented by STCs since here the wetness is not as severe ($\tau = 0.1$).

VII. Practical Embedding Constructions

In this section, we show some applications of the proposed methodology for spatial and transform domain (JPEG) steganography. In the past, most embedding schemes were constrained by practical ways of how to encode the message so that the receiver can read it. Problems such as "shrinkage" in F5 [16], [23] or in MMx [2] arose from this practical constraint. By being able to solve the PLS and DLS problems close to the bound for an arbitrary additive distortion function,[5] steganographers now have much more freedom in designing new embedding algorithms. They only need to select the distortion function and then apply the proposed framework. The only task left to the steganographer is the choice of the distortion function D. It should be selected so that it correlates with statistical detectability. Instead of delving into the difficult problem of how to select the best D, we provide a few examples of additive distortion measures motivated by recent developments in steganography and show their performance when blind steganalysis is used.

In the examples below, we tested the embedding schemes using blind feature-based steganalysis on a large database of images. 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 $(C, \gamma) \in \{(10^k, 2^{j-d})|k \in \{-3, \dots, 4\}, j \in \{-3, \dots, 3\}\}$, where 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

$$P_{\mathrm{E}} = \min_{P_{\mathrm{FA}}} \left(P_{\mathrm{FA}} + P_{\mathrm{MD}}(P_{\mathrm{FA}})\right)/2 \tag{16}$$

where P_{FA} and P_{MD} are the false-alarm and missed-detection probabilities.

A. DCT Domain Steganography

To apply the proposed framework, we first need to design an additive distortion function which can be tested by simulating the embedding as if the best codes are available. Finally, the most promising approach is implemented using STCs. We assume the cover to be a grayscale bitmap image which we JPEG compress to obtain the cover image. Let $\mathcal{A}$ be a set of indexes corresponding to AC DCT coefficients after the block-DCT transform and let c_i be the ith AC coefficient before it is quantized with the quantization step q_i for $i \in \mathcal{A}$. We let $\mathcal{X}$ represent the set of all vectors containing quantized AC DCT coefficients divided by their corresponding quantization step. In ordinary JPEG compression, the values c_i are quantized to $x_i \triangleq [c_i/q_i]$.

1) Proposed Distortion Functions: We define binary embedding operation $\mathcal{I}_i \triangleq \{x_i, \overline{x}_i\}$ by $\overline{x}_i = x_i + \mathrm{sign}(c_i/q_i - x_i)$, where $\mathrm{sign}(x)$ is 1 if $x > 0$, -1 if $x < 0$ and $\mathrm{sign}(0) \in \{-1, 1\}$ uniformly at random. In simple words, x_i is a quantized AC DCT coefficient and $\overline{x}_i$ is the same coefficient when quantized in the opposite direction. Let $e_i = |c_i/q_i - x_i|$ be the quantization error introduced by JPEG compression. By replacing x_i with $\overline{x}_i$ the error becomes $|c_i/q_i - \overline{x}_i| = 1 - e_i$. If $e_i = 0.5$, then the direction where c_i/q_i is rounded depends on the implementation of the JPEG compressor and only small perturbation of the original image may lead to different results. Let $\mathcal{P}(x) = x \bmod 2$. By construction, $\mathcal{P}$ satisfies the property of a parity function, $\mathcal{P}(x_i) \neq \mathcal{P}(\overline{x}_i)$. The distortion function is assumed to be in the form $D(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{n} \varrho_i \cdot [x_i \neq y_i]$, where $n = |\mathcal{A}|$.

The following four approaches utilizing values of e_i and q_i were considered. All methods assign $\varrho_i = \infty$ when $c_i/q_i \in (-0.5, 0.5)$ and differ in the definition of the remaining values ϱ_i as follows:

- **S1)** $\varrho_i = 1 - 2e_i$ if $c_i/q_i \notin (-0.5, 0.5)$ (as in perturbed quantization [24]),
- **S2)** $\varrho_i = q_i(1 - 2e_i)$ if $c_i/q_i \notin (-0.5, 0.5)$ (the same as S1 but ϱ_i is weighted by the quantization step),
- **S3)** $\varrho_i = 1$ if $c_i/q_i \in (-1, -0.5] \cup [0.5, 1)$ and $\varrho_i = 1 - 2e_i$ otherwise, and
- **S4)** $\varrho_i = q_i$ if $c_i/q_i \in (-1, -0.5] \cup [0.5, 1)$ and $\varrho_i = q_i(1 - 2e_i)$ otherwise which is similar weight assignment as proposed in [4].

To see the importance of the side-information in the form of the uncompressed cover image, we also include in our tests the nsF5 [16] algorithm, which can be represented in our formalism as $x_i = [c_i/q_i]$, $\overline{x}_i = x_i - \mathrm{sign}(x_i)$, and $\varrho_i = \infty$ if $x_i = 0$ and $\varrho_i = 1$ otherwise. This way, we always have $|\overline{x}_i| < |x_i|$. The nsF5 embedding minimizes the number of changes to nonzero AC DCT coefficients.

2) Steganalysis Setup and Experimental Results: The proposed strategies were tested on a database of 6, 500 digital camera images prepared as described in [47, Sec. 4.1] so that their smaller size was 512 pixels. The JPEG quality factor 75 was used for compression. The steganalyzer employed the 548-dimensional CC-PEV feature set [40]. Fig. 10 shows the minimum average classification error P_{E} achieved by simulating each strategy on the bound using the PLS formulation. The strategies S1 and S2, which assign zero cost to coefficients $c_i/q_i = 0.5$, were worse than the nsF5 algorithm that does not use any side-information. On the other hand, strategy S4, which also utilizes the knowledge about the quantization step, was the best. By implementing this strategy, we have to deal with a wet paper channel which can be well modeled by a linear profile with relative wetness $\tau \approx 0.6$ depending on the image content. We have implemented strategy S4 using STCs, where wet pixels were handled by setting $\varrho_i = C$ for a sufficiently large C. As seen from the results using STCs, payloads below

[5]The additivity constraint can be relaxed and more general distortion measures can be used with the PLS and DLS problems in practice [7].

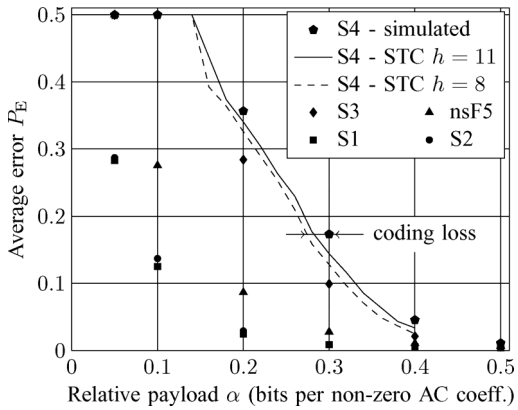


Fig. 10. Comparison of methods with four different weight-assignment strategies S1–S4 and nsF5 as described in Section VII-A when simulated as if the best coding scheme was available. The performance of strategy S4 when practically implemented using STCs with $h = 8$ and $h = 11$ is also shown.

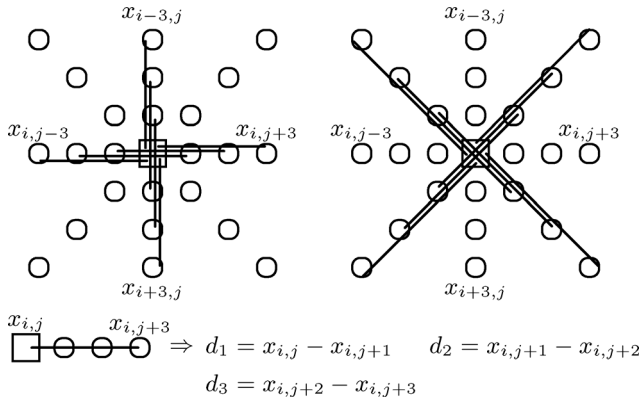


Fig. 11. Set of 4-pixel cliques used for calculating the distortion for digital images represented in the spatial-domain. The final distortion $\rho_{i,j}(y_{i,j})$ is obtained as a sum of terms penalizing the change in pixel $x_{i,j}$ measured w.r.t. each clique containing $x_{i,j}$.

0.15 bits per nonzero AC DCT coefficient were undetectable using our steganalyzer.

Note that our strategies utilized only the information obtainable from a single AC DCT coefficient. In reality, ϱ_i will likely depend on the local image content, quantization errors, and quantization steps. We leave the problem of optimizing D w.r.t. statistical detectability for our future research.

B. Spatial Domain Steganography

To demonstrate the merit of the STC-based multilayered construction, we present a practical embedding scheme that was largely motivated by [5] and [7]. Single per-pixel distortion function $\rho_{i,j}(y_{i,j})$ should assign the cost of changing i,jth pixel $x_{i,j}$, first, from its neighborhood and then also based on the new value $y_{i,j}$. Changes made in smooth regions often tend to be highly detectable by blind steganalysis which should lead to high distortion values. On the other hand, pixels which are in busy and hard-to-model regions can be changed more often.

1) Proposed Distortion Functions: We design our distortion function based on a model build from a set of all straight 4-pixel lines in four different orientations containing i,jth pixel which we call cliques (see Fig. 11). Based on the set of all such cliques, we decide on the value $\rho_{i,j}(y_{i,j})$. Due to strong inter-pixel dependencies, most cliques contain very similar values and thus differences between neighboring pixels tend to be very close to zero. It has been experimentaly observed [5], that number of cliques with differences falls of quickly as the differences gets larger. From this point of view, any clique with small differences should lead to larger distortion because there are more samples the warden can use for training her steganalyzer and the better she can detect the change.

More formally, let $\mathbf{x} \in \{0,\dots,255\}^{n_1 \times n_2}$ be an $n_1 \times n_2$ grayscale cover image, $n = n_1 n_2$, represented in the spatial domain. Define the co-occurrence matrix computed from horizontal pixel differences $D^{\rightarrow}_{i,j}(\mathbf{x}) = x_{i,j+1} - x_{i,j}$, $i = 1,\dots,n_1$, $j = 1,\dots,n_2 - 1$:

$$A^{\rightarrow}_{p,q,r}(\mathbf{x}) = \sum_{i=1}^{n_1} \sum_{j=1}^{n_2-3} \frac{\left[\left(D^{\rightarrow}_{i,j}, D^{\rightarrow}_{i,j+1}, D^{\rightarrow}_{i,j+2}\right)(\mathbf{x}) = (p,q,r)\right]}{n_1(n_2-3)}$$

where $[(D^{\rightarrow}_{i,j}, D^{\rightarrow}_{i,j+1}, D^{\rightarrow}_{i,j+2})(\mathbf{x}) = (p,q,r)] = [(D^{\rightarrow}_{i,j}(\mathbf{x}) = p) \& (D^{\rightarrow}_{i,j+1}(\mathbf{x}) = q) \& (D^{\rightarrow}_{i,j+2}(\mathbf{x}) = r)]$. Clearly, $A^{\rightarrow}_{p,q,r}(\mathbf{x}) \in [0,1]$ is the normalized count of neighboring quadruples of pixels $\{x_{i,j}, x_{i,j+1}, x_{i,j+2}, x_{i,j+3}\}$ with differences $x_{i,j+1} - x_{i,j} = p$, $x_{i,j+2} - x_{i,j+1} = q$, and $x_{i,j+3} - x_{i,j+2} = r$ 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, we define matrices $A^{\nearrow}_{p,q,r}(\mathbf{x})$, $A^{\uparrow}_{p,q,r}(\mathbf{x})$, and $A^{\searrow}_{p,q,r}(\mathbf{x})$. Let $y_{i,j}\mathbf{x}_{\sim i,j}$ be an image obtained from $\mathbf{x}$ by replacing the (i,j)th pixel with value $y_{i,j}$. Finally, we define the distortion measure $D(\mathbf{y}) = \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \rho_{i,j}(y_{i,j})$ by

$$\rho_{i,j}(y_{i,j}) = \sum_{\substack{p,q,r \in \{-255,\dots,255\} \\ s \in \{\rightarrow,\nearrow,\uparrow,\searrow\}}} w_{p,q,r} \left| A^s_{p,q,r}(\mathbf{x}) - A^s_{p,q,r}(y_{i,j}\mathbf{x}_{\sim i,j}) \right| \tag{17}$$

where $w_{p,q,r} = 1/(1+\sqrt{p^2+q^2+r^2})$ are heuristically chosen weights.

2) Steganalysis Setup and Experimental Results: All tests were carried out on the BOWS2 database [48] containing approximately 10,800 grayscale images with a fixed size of 512 $\times$ 512 pixels coming from rescaled and cropped natural images of various sizes. Steganalysis was implemented using the second-order SPAM feature set with $T = 3$ [39].

Fig. 12 contains the comparison of embedding algorithms implementing the PLS and DLS with the costs (17). All algorithms are contrasted with LSB matching simulated on the binary and ternary bounds. To compare the effect of practical codes, we first simulated the embedding algorithm as if the best codes were available and then compared these results with algorithms implemented using STCs with $h = 10$. Both types of senders are implemented with binary, ternary ($\mathcal{I}_i = \{x_i - 1, \dots, x_i + 1\}$), and pentary ($\mathcal{I}_i = \{x_i - 2, \dots, x_i + 2\}$) embedding operations. Before embedding, the binary embedding operation was initialized to $\mathcal{I}_i = \{x_i, y_i\}$ with y_i randomly chosen from $\{x_i - 1, x_i + 1\}$. The reported payload for the DLS with a fixed D_ϵ was calculated as an average over the whole database after embedding.

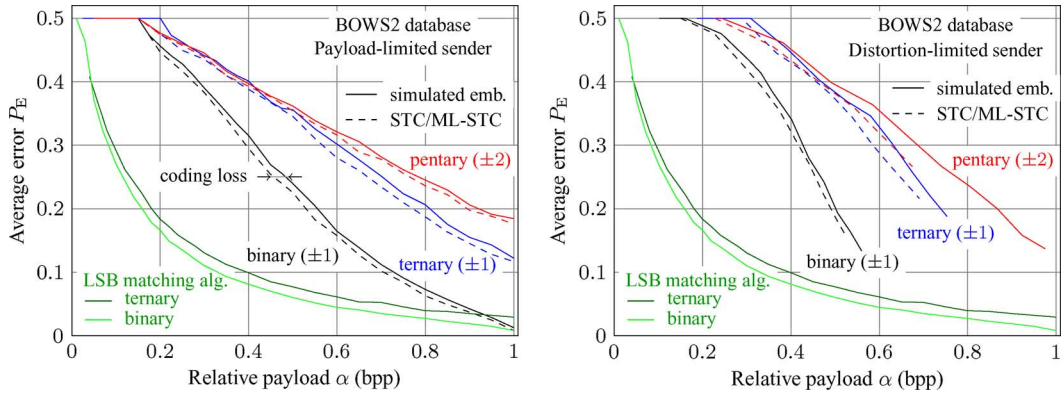


Fig. 12. Comparison of LSB matching with optimal binary and ternary coding with embedding algorithms based on the additive distortion measure (17) using embedding operations of three different cardinalities.

The relative horizontal distance between the corresponding dashed and solid lines in Fig. 12 is bounded by the coding loss. Most of the proposed algorithms are undetectable for relative payloads $\alpha \leq 0.2$ bits per pixel (bpp). For payloads $\alpha \leq 0.5$, the DLS is more secure. For larger payloads, the distortion measure seems to fail to capture the statistical detectability correctly and thus the algorithms are more detectable than when implemented in the payload-limited regime. Finally, the results suggest that larger embedding changes are useful for steganography when placed adaptively.

VIII. Conclusion

The concept of embedding in steganography that minimizes a distortion function is connected to many basic principles used for constructing embedding schemes for complex cover sources today, including the principle of minimal-embedding-impact [16], approximate model-preservation [5], or the Gibbs construction [7]. The current work describes a complete practical framework for constructing steganographic schemes that embed by minimizing an additive distortion function. Once the steganographer specifies the form of the distortion function, the proposed framework provides all essential tools for constructing practical embedding schemes working close to their theoretical bounds. The methods are not limited to binary embedding operations and allow the embedder to choose the amplitude of embedding changes dynamically based on the cover-image content. The distortion function or the embedding operation do not need to be shared with the recipient. In fact, they can even change from image to image. The framework can be thought of as an off-the-shelf method that allows practitioners to concentrate on the problem of designing the distortion measure instead of the problem of how to construct practical embedding schemes.

The merit of the proposed algorithms is demonstrated experimentally by implementing them for the JPEG and spatial domains and showing an improvement in statistical detectability as measured by state-of-the-art blind steganalyzers. We have demonstrated that larger embedding changes provide a significant gain in security when placed adaptively. Finally, the construction is not limited to embedding with larger amplitudes but can be used, e.g., for embedding in color images, where the LSBs of all three colors can be seen as 3-bit symbols on which the cost functions are defined. Applications outside the scope of digital images are possible as long as we know how to define the costs.

The implicit premise of this paper is the direct relationship between the distortion function D and statistical detectability. Designing (and possibly learning) the distortion measure for a given cover source is an interesting problem by itself and is left for our future research. We reiterate that our focus is on constructing practical coding schemes for a given D. Examples of distortion measures presented in this work are unlikely to be optimal and we include them here mainly to illustrate the concepts.

C++ implementation with Matlab wrappers of STCs and multilayered STCs are available at http://dde.binghamton.edu/download/syndrome/.

Acknowledgment

The authors would like to thank X. Zhang for useful discussions.

References

[1] R. Böhme, "Improved Statistical Steganalysis Using Models of Heterogeneous Cover Signals," Ph.D. dissertation, Faculty of Comput. Sci., Technische Universität, Dresden, Germany, 2008.

[2] Y. Kim, Z. Duric, and D. Richards, "Modified matrix encoding technique for minimal distortion steganography," in *Proc. 8th Int. Workshop Inf. Hiding*, J. L. Camenisch, C. S. Collberg, N. F. Johnson, and P. Sallee, Eds., Alexandria, VA, Jul. 10–12, 2006, vol. 4437, Lecture Notes in Computer Science, pp. 314–327.

[3] R. Zhang, V. Sachnev, and H. J. Kim, "Fast BCH syndrome coding for steganography," in *Proc. 11th Int. Workshop Inf. Hiding,*, S. Katzenbeisser and A.-R. Sadeghi, Eds., Darmstadt, Germany, Jun. 7–10, 2009, vol. 5806, Lecture Notes in Computer Science, pp. 31–47.

[4] V. Sachnev, H. J. Kim, and R. Zhang, "Less detectable JPEG steganography method based on heuristic optimization and BCH syndrome coding," in *Proc. 11th ACM Multimedia Security Workshop*, J. Dittmann, S. Craver, and J. Fridrich, Eds., Princeton, NJ, Sep. 7–8, 2009, pp. 131–140.

[5] T. Pevný, T. Filler, and P. Bas, "Using high-dimensional image models to perform highly undetectable steganography," in *Proc. 12th Int. Workshop Inf. Hiding*, P. W. L. Fong, R. Böhme, and R. Safavi-Naini, Eds., Calgary, Canada, Jun. 28–30, 2010, vol. 6387, Lecture Notes in Computer Science, pp. 161–177.

[6] J. Kodovský and J. Fridrich, "On completeness of feature spaces in blind steganalysis," in *Proc. 10th ACM Multimedia Security Workshop*, A. D. Ker, J. Dittmann, and J. Fridrich, Eds., Oxford, U.K., Sep. 22–23, 2008, pp. 123–132.

[7] T. Filler and J. Fridrich, "Gibbs construction in steganography," *IEEE Trans. Inf. Forensics Security*, vol. 5, pp. 705–720, Sep. 2010.

[8] J. Fridrich and T. Filler, "Practical methods for minimizing embedding impact in steganography," in *Proc. SPIE, Electron. Imag., Security, Steganography, Watermark. Multimedia Contents IX*, E. J. Delp and P. W. Wong, Eds., San Jose, CA, Jan. 29–Feb. 1, 2007, vol. 6505, pp. 02–03.

[9] T. Filler and J. Fridrich, "Binary quantization using belief propagation over factor graphs of LDGM codes," presented at the 45th Annu. Allerton Conf. Commun., Control, Comput., Allerton, IL, Sep. 26–28, 2007.

[10] X. Zhang, W. Zhang, and S. Wang, "Efficient double-layered steganographic embedding," *Electron. Lett.*, vol. 43, pp. 482–483, Apr. 2007.

[11] W. Zhang, S. Wang, and X. Zhang, "Improving embedding efficiency of covering codes for applications in steganography," *IEEE Commun. Lett.*, vol. 11, pp. 680–682, Aug. 2007.

[12] W. Zhang, X. Zhang, and S. Wang, "Maximizing steganographic embedding efficiency by combining Hamming codes and wet paper codes," in *Proc. 10th Int. Workshop Inf. Hiding,* , K. Solanki, K. Sullivan, and U. Madhow, Eds., Santa Barbara, CA, Jun. 19–21, 2008, vol. 5284, Lecture Notes in Computer Science, pp. 60–71.

[13] T. Filler and J. Fridrich, "Wet ZZW construction for steganography," presented at the 1st IEEE Int. Workshop Inf. Forensics Security, London, U.K., Dec. 6–9, 2009.

[14] W. Zhang and X. Zhu, "Improving the embedding efficiency of wet paper codes by paper folding," *IEEE Signal Process. Lett.*, vol. 16, pp. 794–797, Sep. 2009.

[15] W. Zhang and X. Wang, "Generalization of the ZZW embedding construction for steganography," *IEEE Trans. Inf. Forensics Security*, vol. 4, pp. 564–569, Sep. 2009.

[16] J. Fridrich, T. Pevný, and J. Kodovský, "Statistically undetectable JPEG steganography: Dead ends, challenges, and opportunities," in *Proc. 9th ACM Multimedia Security Workshop*, J. Dittmann and J. Fridrich, Eds., Dallas, TX, Sep. 20–21, 2007, pp. 3–14.

[17] T. Filler, J. Judas, and J. Fridrich, "Minimizing embedding impact in steganography using trellis-coded quantization," in *Proc. SPIE, Electron. Imag., Security, Forensics Multimedia XII*, N. D. Memon, E. J. Delp, P. W. Wong, and J. Dittmann, Eds., San Jose, CA, Jan. 17–21, 2010, vol. 7541, pp. 05-01–05-14.

[18] T. Filler and J. Fridrich, "Using non-binary embedding operation to minimize additive distortion functions in steganography," presented at the 2nd IEEE Int. Workshop Inf. Forensics Security, Seattle, WA, Dec. 12–15, 2010.

[19] S. I. Gel'fand and M. S. Pinsker, "Coding for channel with random parameters," *Problems Control Inf. Theory*, vol. 9, no. 1, pp. 19–31, 1980.

[20] A. D. Ker, "Batch steganography and pooled steganalysis," in *Proc. 8th Int. Workshop Inf. Hiding*, J. L. Camenisch, C. S. Collberg, N. F. Johnson, and P. Sallee, Eds., Alexandria, VA, Jul. 10–12, 2006, vol. 4437, Lecture Notes in Computer Science, pp. 265–281.

[21] C. E. Shannon, "Coding theorems for a discrete source with a fidelity criterion," *IRE Nat. Conv. Rec.*, vol. 4, pp. 142–163, 1959.

[22] T. M. Cover and J. A. Thomas*, Elements of Information Theory*. New York: Wiley, 2006.

[23] A. Westfeld, "High capacity despite better steganalysis (F5—A steganographic algorithm)," in *Proc. 4th Int. Workshop Inf. Hiding*, I. S. Moskowitz, Ed., Pittsburgh, PA, Apr. 25–27, 2001, vol. 2137, Lecture Notes in Computer Science, pp. 289–302.

[24] J. Fridrich, M. Goljan, and D. Soukal, "Perturbed quantization steganography," *ACM Multimedia Syst. J.*, vol. 11, no. 2, pp. 98–107, 2005.

[25] J. Fridrich, M. Goljan, D. Soukal, and P. Lisoněk, "Writing on wet paper," *IEEE Trans. Signal Process., Special Issue on Media Security*, vol. 53, pp. 3923–3935, Oct. 2005.

[26] P. Moulin and R. Koetter, "Data-hiding codes," *Proc. IEEE*, vol. 93, no. 12, pp. 2083–2126, 2005.

[27] R. Crandall, "Some notes on steganography," in Steganography Mailing List [Online]. Available: http://os.inf.tu-dresden.de/ westfeld/crandall.pdf 1998

[28] J. Bierbrauer, On Crandall's Problem [Online]. Available: http:// www.ws.binghamton.edu/fridrich/covcodes.pdf 1998

[29] J. Bierbrauer and J. Fridrich, "Constructing good covering codes for applications in steganography," *LNCS Trans. Data Hiding Multimedia Security*, vol. 4920, pp. 1–22, 2008.

[30] M. van Dijk and F. Willems, "Embedding information in grayscale images," in *Proc. 22nd Symp. Inf. Commun. Theory*, Enschede, The Netherlands, May 15–16, 2001, pp. 147–154.

[31] D. Schönfeld and A. Winkler, "Embedding with syndrome coding based on BCH codes," in *Proc. 8th ACM Multimedia Security Workshop*, S. Voloshynovskiy, J. Dittmann, and J. Fridrich, Eds., Geneva, Switzerland, Sep. 26–27, 2006, pp. 214–223.

[32] J. Fridrich and D. Soukal, "Matrix embedding for large payloads," *IEEE Trans. Information Forensics and Security*, vol. 1, no. 3, pp. 390–394, 2006.

[33] J. Fridrich, "Asymptotic behavior of the ZZW embedding construction," *IEEE Trans. Inf. Forensics Security*, vol. 4, pp. 151–153, Mar. 2009.

[34] J. Fridrich, M. Goljan, and D. Soukal, "Wet paper codes with improved embedding efficiency," *IEEE Trans. Inf. Forensics Security*, vol. 1, no. 1, pp. 102–110, 2006.

[35] E. Arikan, "Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels," *IEEE Trans. Inf. Theory*, vol. 55, pp. 3051–3073, Jul. 2009.

[36] S. B. Korada and R. L. Urbanke, "Polar codes are optimal for lossy source coding," *IEEE Trans. Inf. Theory*, vol. 56, pp. 1751–1768, Apr. 2010.

[37] D. MacKay*, Information Theory, Inference, and Learning Algorithms*. Cambridge, U.K.: Cambridge Univ. Press, 2003 [Online]. Available: http://www.inference.phy.cam.ac.uk/mackay/itila/

[38] T. Filler, A. D. Ker, and J. Fridrich, "The Square Root Law of steganographic capacity for Markov covers," in *Proc. SPIE, Electron. Imag., Security, Forensics Multimedia XI*, N. D. Memon, E. J. Delp, P. W. Wong, and J. Dittmann, Eds., San Jose, CA, Jan. 18–21, 2009, vol. 7254, pp. 08 1–08 11.

[39] T. Pevný, P. Bas, and J. Fridrich, "Steganalysis by subtractive pixel adjacency matrix," in *Proc. 11th ACM Multimedia Security Workshop*, J. Dittmann, S. Craver, and J. Fridrich, Eds., Princeton, NJ, Sep. 7–8, 2009, pp. 75–84.

[40] J. Kodovský and J. Fridrich, "Calibration revisited," in *Proc. 11th ACM Multimedia Security Workshop*, J. Dittmann, S. Craver, and J. Fridrich, Eds., Princeton, NJ, Sep. 7–8, 2009, pp. 63–74.

[41] A. Viterbi and J. Omura, "Trellis encoding of memoryless discrete-time sources with a fidelity criterion," *IEEE Trans. Inf. Theory*, vol. 20, pp. 325–332, May 1974.

[42] I. Hen and N. Merhav, "On the error exponent of trellis source coding," *IEEE Trans. Inf. Theory*, vol. 51, no. 11, pp. 3734–3741, 2005.

[43] S. Pradhan and K. Ramchandran, "Distributed source coding using syndromes (DISCUS): Design and construction," *IEEE Trans. Inf. Theory*, vol. 49, no. 3, pp. 626–643, 2003.

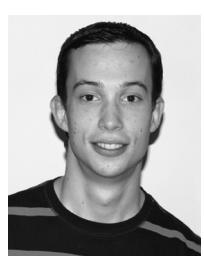
[44] V. Sidorenko and V. Zyablov, "Decoding of convolutional codes using a syndrome trellis," *IEEE Trans. Inf. Theory*, vol. 40, no. 5, pp. 1663–1666, 1994.

[45] A. Calderbank, P. Fishburn, and A. Rabinovich, "Covering properties of convolutional codes and associated lattices," *IEEE Trans. Inf. Theory*, vol. 41, no. 3, pp. 732–746, 1995.

[46] C. K. Wang, G. Doërr, and I. Cox, "Trellis coded modulation to improve dirty paper trellis watermarking," in *Proc. SPIE, EI, Security, Steganography, Watermarking Multimedia Contents IX*, E. J. Delp and P. W. Wong, Eds., San Jose, CA, Jan. 29–Feb. 1, 2007, p. 65050G.

[47] J. Kodovský, T. Pevný, and J. Fridrich, "Modern steganalysis can detect YASS," in *Proc. SPIE, Electron. Imag., Security, Forensics Multimedia XII*, N. D. Memon, E. J. Delp, P. W. Wong, and J. Dittmann, Eds., San Jose, CA, Jan. 17–21, 2010, vol. 7541, pp. 02-01–02-11.

[48] P. Bas and T. Furon, BOWS-2 [Online]. Available: http://bows2.gipsa-lab.inpg.fr Jul. 2007



Tomáš Filler (S'08–M'11) received the M.S. degree (*summa cum laude*) in computer science from the Czech Technical University, Prague, Czech Republic, in 2007. He is currently working towards the Ph.D. degree under the supervision of Prof. J. Fridrich.

He is currently a Research Assistant at the Department of Electrical and Computer, Binghamton University, State University of New York. His research interest are focused in the area of data hiding, information, and coding theory.

Mr. Filler received the Graduate Student Award for Excellence in Research from Binghamton University in 2010 and Best Paper Awards from Digital Watermarking Alliance in 2009 and 2010.

Jan Judas (S'09–M'10) received the M.S. degree (*summa cum laude*) in computer science from the Czech Technical University, Prague, Czech Republic, in 2010.

He worked on the paper while he was a visiting scholar at Binghamton University, State University of New York, in 2009 and 2010. He now works as a software developer in Prague.

Jessica Fridrich (M'05) received the Ph.D. degree in systems science from Binghamton University, State University of New York, in 1995 and the M.S. degree in applied mathematics from Czech Technical University, Prague, in 1987.

She is a Professor of Electrical and Computer Engineering at Binghamton University, State University of New York. Her main interests are in steganography, steganalysis, and digital image forensic. She has authored over 120 papers on data embedding and steganalysis and holds seven U.S. patents.

Dr. Fridrich received the IEEE Signal Processing Society Best Paper Award for her work on sensor fingerprints. She is a member of the ACM.