On the Statistical Properties of Syndrome Trellis Coding

Olaf Markus Köhler[1], Cecilia Pasquini[2] and Rainer Böhme[1,2]

[1] Department of Computer Science, Universität Innsbruck, Austria
[2] Department of Information Systems, Universität Münster, Germany

Abstract. Steganographic systems use Syndrome Trellis Coding (STC) to control the selection of embedding positions in a cover, subject to a set of stochastic constraints. This paper reports observations from a series of experiments on the ability of Syndrome Trellis Coding to approximate independent Bernoulli random variables. We find that approximation errors are generally small except for some outliers at boundary positions. Bivariate dependencies between embedding changes do reveal the use of the code and its parameters. While risky outliers can be hidden by permuting the cover before coding, or avoided by using the proposed "outlier corrected" variant OC-STC, the aggregate bivariate statistics are invariant to permutations and therefore constitute a potential security risk in the presence of powerful attackers.

1 Introduction

Syndrome coding is a key element of modern steganography. It allows the transmission of steganographic messages without the need to share with the recipient the location of the embedding changes in a cover. The most popular form of syndrome coding is known as Syndrome Trellis Coding (STC) [7]. STC is specifically suited for separating the concerns of *where* to embed and *how* to embed, combined with unparalleled computational efficiency and marginal coding loss. Indeed, since the introduction of STC, the research community has adopted the convention to test new embedding functions with simulated embedding rather than meaningful payloads, thereby relying on STC's ability to substitute random changes with a close to optimal encoding of payload bits [1, 7, 15]. In particular, the wide acceptance of and reliance on STC calls for a closer inspection of the statistical properties of the code, specifically its reference implementation [3]. That is where this paper seeks to contribute to the state of knowledge.

In a nutshell, STC takes as inputs the cover, a vector of change probabilities per cover element, and a message. It produces a vector of positions where the cover must be changed in order to embed the message. A common abstraction is that (binary) STC outputs a realization of a vector of independent Bernoulli random variables. However, the structure of the code and constraints to the solver clearly invalidate this assumption. Our objective in this research is to characterize this discrepancy with statistical means. In other words, our guiding questions are: how close does STC come to realize the prescribed change probabilities? If

there is measurable discrepancy, does it follow systematic patterns? And how do the code parameters influence the magnitude and pattern of the discrepancy?

We take an experimental approach, drawing on 150 million encodings under controlled conditions. We report observations made on the level of aggregate moments, univariate statistics, and indicators of bivariate dependency. A main finding is that the standard way STCs are presented in academic publications and reference implementations produces violates the embedding constraints at the beginning of the trellis. We propose a modified construction, called OC-STC, which avoids these outliers.

To be clear, theory predicts that some discrepancy is unavoidable. Even if characteristic patterns are statistically identifiable, we are not aware of an immediate path to mount steganalytic attacks even in the presence of outliers. This is because real-world attackers do not enjoy the same amount of control over related encodings as in our simulations. Moreover, standard constructions use a key-dependent pseudo-random permutation to shield the coding layer from the scrutiny of computationally bounded steganalysts. Nevertheless, we deem it worthwhile to explore this relevant building block of steganographic systems, with an eye on potential weaknesses in more exotic constructions, such as public key steganography [2], or as a second line of defense against side channels which reveal the stego key to the attacker [11, 13].

This paper is organized as follows. The next Section 2 recalls known theory. Section 3 describes the analytical approach and justifies parameter choices. The results are reported in Section 4, further structured in sub-sections per level of analysis, each of which includes a brief discussion. Section 5 proposes and briefly evaluates the construction of OC-STC. General observations and limitations are summarized in the concluding Section 6.

2 Background

Without loss of generality, we consider the spatial domain representation of natural gray scale images as communication channel. In this domain, positions are referred to as pixels. Even though images are two-dimensional, we index pixels column-wise by a single integer i.

Steganography by cover modification takes a cover image of length n, denoted as $\boldsymbol{x} = (x_i)_{i \in 1 \dots n}$, and modifies it to obtain a stego image $\boldsymbol{y} = (y_i)_{i \in 1 \dots n}$. The stego image contains the desired message $\boldsymbol{m} = (m_j)_{j \in 1 \dots \alpha n}$, where the embedding rate α is the ratio between message and cover length. For the sake of simplicity, we assume α to be chosen such that α^{-1} is an integer. Further, let cover $\boldsymbol{x}$ be arbitrary but fixed. Slightly overloading notation, we interpret $\boldsymbol{x}$ and $\boldsymbol{y}$ as *integer* vectors when they refer to the cover and stego image, and as *binary* vectors in relation to coding. This implicitly assumes a mapping between images and their (binary) steganographic semantic. Using LSBs to carry steganographic semantic is one popular approach, but more sophisticated (and more secure) embedding operations are possible.

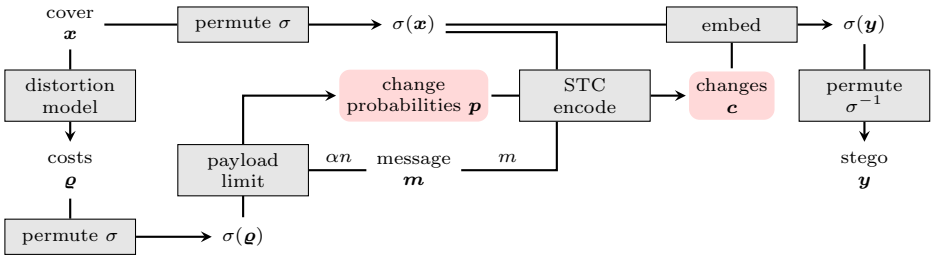

Fig. 1. System model of the embedding process.

The role of coding is to determine the position of embedding changes between the stego and cover image. It is convenient to represented the set of changes by a binary vector $\boldsymbol{c} = (c_i)_{i=1\ldots n} \in \{0,1\}^n$, where $c_i = 1 \Leftrightarrow x_i \neq y_i$. An objective of the coding process is to minimize the statistical distinguishability due to the embedding changes between cover and stego images. This is connected to the protection goal of "undetectability" of stego images among cover images. Quantifying this distinguishability would require full knowledge of the distribution of cover objects, which is infeasible [9, Ch. 7], so that heuristic distortion measures are used as an approximation. As a result, coding techniques generally aim at embedding the desired message and, at the same time, minimizing some kind of analytically tractable distortion measure. In our work, we study the case of Syndrome Trellis Coding (STC), a state-of-the-art technique solving the coding task while minimizing a distortion measure that is assumed to be additive over all pixels in an image.

In the rest of this section we formalize the additive distortion measure considered in the context of our system model (Section 2.1), present the basic concepts of STC (Section 2.2), and recall the calculation of the optimal change probabilities induced by the distortion model and the payload size (Section 2.3).

2.1 System Model

The process of embedding message $\boldsymbol{m}$ into cover $\boldsymbol{x}$ is presented in system model Fig. 1. First, the cost map $\boldsymbol{\varrho} = (\varrho_i)_{i=1\ldots n}$ is derived from the cover over the additive distortion model. The additive distortion model assigns each pixel with a positive scalar ϱ_i, representing the cost of changing the pixel at position i. This can be done by means of different heuristics, such as WOW [10] and HILL [12]. In the case of basic (single layer) STC [6], the following simplifications are assumed in the computation of distortion:

- Pixel i's contribution to the global distortion is given by $c_i \varrho_i$.
- The global distortion d is the sum of individual distortions: $d = \sum_{i=1}^{n} c_i \varrho_i$.

Both cover and cost map are permuted under the same permutation σ. The permutation can be thought of as an interleaving method which distributes messages bits over cover positions approximately equally. This increases the chance

of successful embedding in the presence of dense local constraints. The convention to use a key-dependent pseudo-random permutation also improves security. It obscures the relationship between cover position and message position for steganalysts who do not know the key and cannot recover the embedding path.

We assume a payload limited sender. The optimal change probabilities $\boldsymbol{p}$ are calculated from the cost map $\sigma(\boldsymbol{\varrho})$ and payload size αn as detailed in Sect. 2.3.

The encoding operation takes the message $\boldsymbol{m}$, permuted cover $\sigma(\boldsymbol{x})$ and permuted cost map $\sigma(\boldsymbol{\varrho})$ as inputs and returns a vector of changes $\boldsymbol{c}$, representing the modifications that have to be applied to the permuted cover. Via the embedding operation the permuted cover $\sigma(\boldsymbol{x})$ is modified according to change vector $\boldsymbol{c}$, returning the permuted stego $\sigma(\boldsymbol{y})$. Before communicating the stego, it is permuted back to the order of the cover object. To extract the message $\boldsymbol{m}$, the recipient would permute the stego and multiply it with the parity-check matrix.

As our main concern is to examine the encoding operation, namely STC, we observe the change vector $\boldsymbol{c}$, which implicitly assumes knowledge of the embedding path given by permutation σ.

2.2 Syndrome Trellis Coding

We use syndrome coding for encoding. In syndrome coding, message $\boldsymbol{m}$ is derived as a syndrome via the stego $\boldsymbol{y}$ and parity-check matrix $\mathbb{H}$, $\mathbb{H}\boldsymbol{y} = \boldsymbol{m}$. STC [6] is a special case of syndrome coding where the matrix $\mathbb{H}$ is constructed by concatenating and shifting a submatrix $\hat{\mathbb{H}} \in \{0,1\}^{h \times \alpha^{-1}}$ multiple times, to create a sparse and systematically constructed parity-check matrix $\mathbb{H}$,

$$\mathbb{H} = \begin{pmatrix} \hat{\mathbb{H}} & 0\ldots0 & 0\ldots0 & & \\ 0\ldots0 & \hat{\mathbb{H}} & 0\ldots0 & & \\ 0\ldots0 & 0\ldots0 & \hat{\mathbb{H}} & & 0 \\ & & & \ddots & \\ & & & & \hat{\mathbb{H}} & 0\ldots0 & 0\ldots0 \\ 0 & & & & 0\ldots0 & \hat{\mathbb{H}} & \vdots \\ & & & & & 0\ldots0 & \ddots \end{pmatrix}, \tag{1}$$

where constant h is called constraint length. This construction allows the use of the Viterbi algorithm [14] to efficiently find a change vector $\boldsymbol{c}$ which minimizes distortion d and implies $\boldsymbol{m}$ as syndrome, i.e., $\mathbb{H}(\boldsymbol{x} \oplus \boldsymbol{c}) = \boldsymbol{m}$ over GF(2).

The algorithm traverses a trellis diagram of size $2^h \times n(\alpha + 1)$. Accordingly, the computational complexity is in $O(n)$ for a constant submatrix $\hat{\mathbb{H}}$ of height h. Regarding the constraint length h, the computational complexity is in $O(e^h)$. Thus, STC is only feasible for small constants h, usually $7 \leq h \leq 13$.

For a fixed cover $\boldsymbol{x}$ of length n and message $\boldsymbol{m}$ of length αn, the relevant parameter influencing the execution of the STC is the code parameter h, and more specifically, the submatrix $\hat{\mathbb{H}}$. Due to its impact on the computational complexity and the probability of finding valid stego objects, $\hat{\mathbb{H}}$ has to be chosen carefully, as described in [6, Sect. 5.2].

The strict minimization of global distortion d leads to different behavior of STC for different cost maps $\boldsymbol{\varrho}$. For instance, embedding in a cover with only

few low-distortion pixels produces embedding changes at these few pixels with a high probability, whereas for a cover with similar distortions for all pixels, such high probabilities are less likely. As we aim to include behavior under different circumstances in our experiments, variance in distortion is of interest.

2.3 Optimal Change Probabilities

As presented in [4, Sect. II], choosing embedding changes by additive distortion minimization must follow a particular form of Gibbs distribution. This especially assumes independence between embedding changes at different pixels. For our set of assumptions presented in Sect. 2.1, the general results of [4] can be simplified. In accordance with [5, 7], we calculate the independent optimal change probabilities p_i for each pixel i with cost ϱ_i as

$$p_i = \frac{e^{-\lambda \varrho_i}}{1 + e^{-\lambda \varrho_i}} \ , \tag{2}$$

where λ is a scaling variable that needs to be chosen such that the overall entropy fits the length of the message, $\sum_{i=1\dots n} H(p_i) = n\alpha$, where (binary) entropy is defined as $H(p_i) = -p_i \log_2(p_i) - (1-p_i)\log_2(1-p_i)$. The scaling is based on the assumption that the message has full entropy.

By characterizing optimal embedding in terms of probability theory, let $\boldsymbol{C} = (C_i)_{i=1\dots n}$ be the random vector of embedding changes, whose sample space is $\{0,1\}^n$. Thus, $\boldsymbol{C}$ follows a multivariate Bernoulli distribution, which reduces to a product of univariate Bernoulli distributions due to the assumption of mutually independence of the embedding changes p_i. The pmf of $\boldsymbol{C}$ is then given by

$$P_{\boldsymbol{C}}(\boldsymbol{c}) = \prod_{i=1,\dots,n} (p_i)^{c_i} \, (1-p_i)^{1-c_i} \ . \tag{3}$$

Optimal coding would produce change vectors indistinguishable from this distribution model. A common conjecture is that STC can approximate optimal coding the better the more computational complexity is spent on coding via the choice of the constraint length h [15].

3 Experimental Approach

The goal is to observe STC's behavior in determining the change vector with respect to the optimal embedding change distribution $P_{\boldsymbol{C}}$. Supposing to embed N different messages for cover $\boldsymbol{x}$ and coding parameter h, this results in N change vector samples $(\boldsymbol{c}^{(1)}, \dots, \boldsymbol{c}^{(N)})$, i.e., N realizations of the random vector $\boldsymbol{C}$. As shown in Section 2.3, each of these vectors is supposed to follow the distribution specified in (3). We then define the relative frequency distribution $\hat{P}^h_{\boldsymbol{C}} : \{0,1\}^n \to [0,1]$, where $\hat{P}^h_{\boldsymbol{C}}(\boldsymbol{c})$ is given by the relative frequencies of occurrence of $\boldsymbol{c}$ in the observed N change vectors.

3.1 Levels of Analysis

In principle, $\hat{P}^h_{\boldsymbol{C}}(\boldsymbol{c})$ should be compared with $P_{\boldsymbol{C}}(\boldsymbol{c})$. However, there are 2^n different possible realizations $\boldsymbol{c}$ of the random vector $\boldsymbol{C}$, so that statistical observation of the whole vector would require an infeasible large sample size $N \gg 2^n$. Instead, we study different projections of the sample space with reduced dimensionality, of which we can derive theoretical distributions. In particular, we consider the following levels of analysis:

Count of Embedding Changes. First, we observe the scalar random variable A based on the count of embedding changes $A = \sum_{i=1}^{n} C_i$. Realizations of A are denoted by a. Under the assumption of optimal coding, A is the sum of independent Bernoulli variables with different parameters p_i, thus it should follow a Poisson binomial distribution. Deviations from this distribution lead to the conclusion that at least one of the underlying assumptions (independence among pixels or observed relative frequency equal to p_i) is violated.

We expect to observe deviations, as embedding changes are not independent due to the structured dependencies implied by parity-check matrix $\mathbb{H}$. Still, it is interesting to look at the influence of coding parameter h on this statistic. We expect results closer to the optimum for larger coding parameters h.

As a follow-up, we look into statistics that allow us to differentiate between the assumptions of empirical results fulfilling the correct individual probabilities and them being independent.

Single-Pixel Embedding Changes. Secondly, we observe the univariate random variables C_i given by the i-th components of $\boldsymbol{C}$. Realizations of C_i are denoted by c_i. Under optimal coding, they should follow a univariate Bernoulli distribution with probability p_i.

We expect to observe slight deviations and look into the structure and distribution of deviations, and the influence of coding parameter h.

Pair of Pixels Embedding Changes. Thirdly, we observe the bivariate random variables $\boldsymbol{C}_{i,j} = (C_i, C_j)$ given by pairs of components of $\boldsymbol{C}$. Realizations of $\boldsymbol{C}_{i,j}$ are denoted by $\boldsymbol{c}_{i,j} = (c_i, c_j)$. Under optimal coding, their components should be independent and we will evaluate such assumption via χ^2-tests for independence.

Computational Performance. Finally, since the statistical properties of STC depend on the choice of h, it is instructive to also evaluate the computational cost associated with this code parameter.

3.2 Selection of Distortion Profiles from Real Cover Images

In principle, STC can take arbitrary change probabilities $\boldsymbol{p}$ as input. For better validity in the application domain steganography, we use the probabilities produced by a typical distortion model applied to real cover images. However, to limit the computational effort, we use small patches sized 64×64 pixels. To reflect

the heterogeneity of real covers, we select covers with diverse distortion maps from a standard benchmark database used in steganography research. Therefore, we compare covers by their distortion profile, which is given by a sorted n-tuple of costs $\operatorname{sort}(\boldsymbol{\varrho})$ [8].

We use WOW [10] as distortion model. Since we are not interested in the security against signal-based steganalysis, this choice is not crucial for our results. The same holds for potential singularities at the patch boundaries.

We systematically select a set X of 1000 different 64×64 covers from the 10 000 512×512 images in BOSSBase v1.01 [1] as follows. The 10 000 images are cropped at random offsets to form 10 000 64×64 patches. The distortion profile for each of the 10 000 64×64 patches is calculated and scaled linearly to have coinciding maxima. Scaling allows easy comparability and does not harm the profiles' information on STC's behavior, as STC's behavior is invariant to a linear scaling of costs. The first cover is a randomly selected patch. The set of covers is incrementally expanded by the 64×64 patch with the largest product of distances to each element of the set of covers. We define the distance of two distortion profiles as the integral of absolute difference. Fig. 2 shows the first and last four 64×64 patches selected as covers by this process.

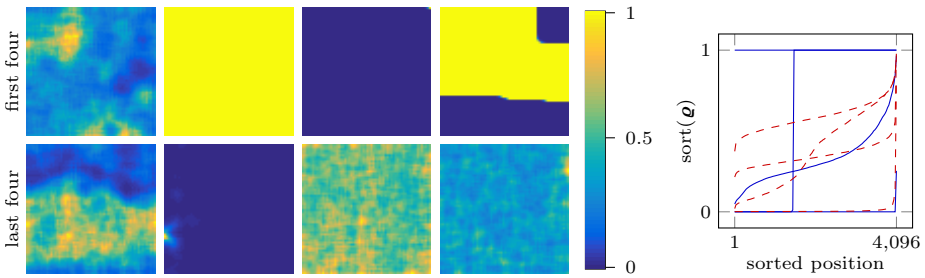

Fig. 2. WOW distortion maps (left) and profiles (right) of the first four (blue lines) and last four (red dashed lines) systematically selected covers.

3.3 Experimental Setup

To gather empirical data, we use our set of covers X and fix an embedding rate of $\alpha = 0.5$. As the reference implementation of STC [3] supports submatrices for constraint lengths $h \in \{7, \dots, 13\}$, we use the following submatrices,

$$\hat{\mathbb{H}}_7 = \begin{pmatrix} 1\,0\,1\,1\,0\,1\,1 \\ 1\,1\,1\,0\,0\,0\,1 \end{pmatrix}^T, \quad \hat{\mathbb{H}}_{10} = \begin{pmatrix} 1\,0\,1\,0\,0\,1\,1\,1\,1\,1 \\ 1\,1\,0\,0\,1\,1\,1\,0\,0\,1 \end{pmatrix}^T, \quad \hat{\mathbb{H}}_{13} = \begin{pmatrix} 1\,0\,1\,1\,0\,0\,1\,0\,0\,0\,1\,0\,1 \\ 1\,1\,1\,1\,1\,0\,1\,0\,0\,1\,0\,1\,1 \end{pmatrix}^T, \quad (4)$$

which correspond to the submatrices defined in [3] for $h \in \{7, 10, 13\}$.

For each cover $\boldsymbol{x} \in X$ and $h \in \{7, 10, 13\}$, $N = 50\,000$ random messages are embedded using a fixed permutation σ. This results in a set of change vectors

$Z_h = \{\boldsymbol{c}^{(j)}\}_{j=1\ldots N}$ per cover. Recall that each $\boldsymbol{c}$ is ordered along the embedding path and thus assumes knowledge of permutation σ. We use a C implementation of the Viterbi algorithm that has been tested to be functionally equivalent to the MATLAB reference implementation in [3]. Our implementation and analysis does not support multi-layered constructions of STC [5, 7].

4 Results and Discussion

We report and discuss results by level of analysis (cf. Sect. 3.1).

4.1 Count of Embedding Changes

As explained in Section 3.1, the count of embedding changes A under the assumption of optimal coding follows a Poisson binomial distribution [17]. Its probability mass function (pmf) is given by

$$P_A(a) = \sum_{c\in\{0,1\}^n:\ a=\sum_\iota c_\iota} \prod_{i:c_i=1} p_i \prod_{j:c_j=0} (1-p_j)\ . \tag{5}$$

For a fixed cover $\boldsymbol{x}$, we choose an asymmetric 95% confidence interval $[a_{\min}, a_{\max}]$ around the expected value $E_P[A] = \sum_{i=1}^n p_i$ such that $\sum_{a<a_{\min}} P_A(a) \approx 0.025$ and $\sum_{a<a_{\max}} P_A(a) \approx 0.975$. Then, we observe the ratio of change vectors with embedding change counts within the confidence interval. The histogram over these ratios is presented in Fig. 3.

Discussion. We see that for larger constraint lengths h, a higher percentage of cases falls into the confidence interval. Specifically, the ratio for $h = 13$ is 92%. This comes sufficiently close to 95%, which is the expected value if the true data generating process was Poisson binomial distributed.

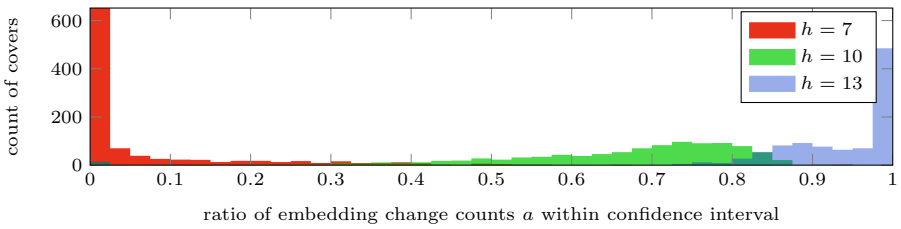

Fig. 3. Histogram of per-cover ratio of embedding change counts within the 95% confidence interval.

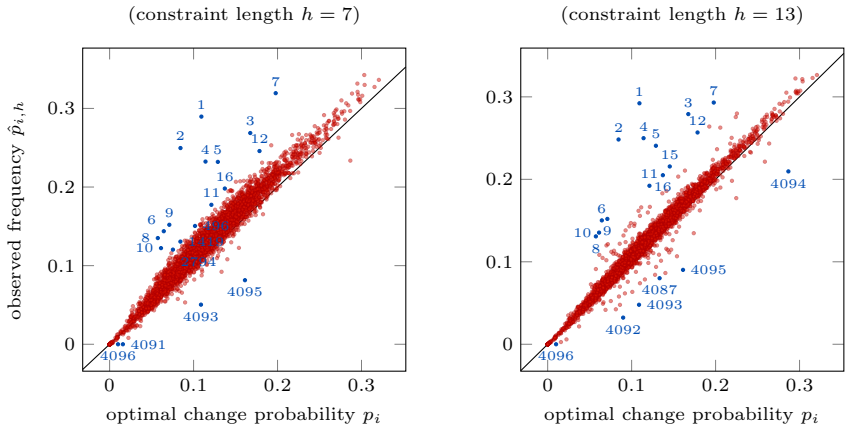

Fig. 4. Exemplary comparison of optimal change probabilities p_i and observed frequencies $\hat{p}_{i,h}$ at $h = 7$ (left) and $h = 13$ (right) for all pixels of a cover, annotated with positions during encoding. Points on the main diagonal refer to pixels with observed frequencies $\hat{p}_{i,h}$ that exactly meet the optimal change probabilities p_i. This example refers to image 5729 of the BOSSBase dataset, cropped with offset $(258, 53)$.

4.2 Single-Pixel Embedding Changes

According to (3), individual pixel changes C_i should follow a Bernoulli distribution with the pmf

$$P_{C_i}(c_i) = (p_i)^{c_i}(1-p_i)^{1-c_i} \quad . \tag{6}$$

The optimal change probability at any pixel i is given by $P_{C_i}(1) = p_i$. For a fixed cover $\boldsymbol{x}$, let the observed frequency $\hat{p}_{i,h}$ at any pixel i with constraint length h be defined by $\hat{p}_{i,h} = \frac{1}{N}\,|\{c \in Z_h : c_i = 1\}|$.

The univariate Bernoulli distributions $P_{C_i}(c_i)$ and $\hat{P}^h_{C_i}(c_i)$ are compared based on their success ratios p_i and $\hat{p}_{i,h}$, as they fully define the distributions. For a fixed cover $\boldsymbol{x}$ and constraint length h and all pixels i, an exemplary visual comparison of p_i and $\hat{p}_{i,h}$ is given in Fig. 4. To quantify the divergence between p_i and $\hat{p}_{i,h}$ we calculate the Hellinger distance, $D_{\text{Hellinger}}(p_i,\ \hat{p}_{i,h}) = \sqrt{\sqrt{1-p_i}\,\sqrt{1-\hat{p}_{i,h}} + \sqrt{p_i}\,\sqrt{\hat{p}_{i,h}} + 1}$.

In Fig. 4 the pixels with the 20 highest Hellinger distances are annotated with their position during encoding (i.e., in the permuted order). We observe that most of the marked pixels are located at the beginning and the end of the change vector.

To generalize from this cover-specific observation, we look at the mean Hellinger distance per pixel over all covers. As visible in Fig. 5, the concentration at beginning and end can be observed across covers $\boldsymbol{x}$ and constraint lengths h.

Another way of looking at this result is by aggregating the difference $\hat{p}_{i,h} - p_i$ over all pixels of all covers and presenting it in a histogram, as shown in Fig. 6.

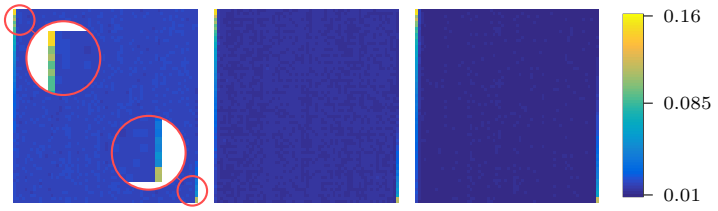

Fig. 5. Mean Hellinger distance per pixel, in order of coding, coding length $h = 7$ (left), 10 (middle), 13 (right).

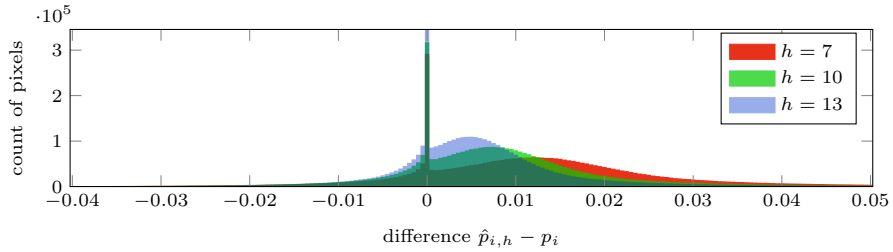

Fig. 6. Histogram of difference between observed frequency and optimal change probabilities $\hat{p}_{i,h} - p_i$ over all pixels of all covers.

Discussion. High Hellinger distances at the borders of the change vector (as shown in Fig. 5) can be explained by the construction of the parity-check matrix $\mathbb{H}$ in (1). Consider the case of $\alpha = 0.5$. The first message bit depends only on the first 2 bits of the permuted stego object $\sigma(\boldsymbol{y})$. This means that only these two bits can be changed to embed the first message bit. Assuming that (the binary representation of) x_1, x_2 and m_1 are uniformly distributed, the probability of introducing an embedding change at one of these positions is 50 % even though the sum of the optimal change probabilities $p_1 + p_2$ can be much lower. A similar border effect happens, albeit to a lesser extent, at other positions $j < h$.

To recall, the calculation of optimal change probabilities p_i is based on the assumption of embedding at the entropy limit. However, STC is only able to embed as close to this limit as it is constrained by the parity-check matrix $\mathbb{H}$, specifically its constraint length h. Thus, a positive deviation of the mean of differences from zero is expected. Smaller h values impose greater restrictions on coding and thus imply higher observed frequencies, as a consequence. These expectations coincide with the findings presented in the example in Fig. 4, as well as the overall positive mean deviation (Fig. 6).

4.3 Pair of Pixels Embedding Changes

According to the distribution of optimal embedding changes P, pairwise pixel changes $\boldsymbol{C}_{i,j}$ follow a bivariate Bernoulli distribution with pmf

$$P_{\boldsymbol{C}_{i,j}}(\boldsymbol{c}_{i,j}) = \left(p_{i,j}^{(00)}\right)^{(1-c_i)(1-c_j)} \left(p_{i,j}^{(01)}\right)^{(1-c_i)c_j} \left(p_{i,j}^{(10)}\right)^{c_i(1-c_j)} \left(p_{i,j}^{(11)}\right)^{c_i c_j} \quad , \tag{7}$$

where the co-occurrence probabilities $p_{i,j}^{(c_i c_j)}$ describe the probability of the embedding changes at pixels i and j being equal to c_i and c_j, respectively. Due to their pixel changes' mutual independence, (7) can be rewritten as

$$P_{\boldsymbol{C}_{i,j}}(\boldsymbol{c}_{i,j}) = (p_i)^{c_i} (1-p_i)^{1-c_i} (p_j)^{c_j} (1-p_j)^{1-c_j} \quad , \tag{8}$$

consistently with (3). Similar to (7), from the observed pairwise pixel changes $\boldsymbol{c}_{i,j}$ we can compute the relative co-occurrence frequencies $\hat{p}_{i,j,h}^{(c_i c_j)}$, determined by the ratio of change vectors in which the embedding changes at pixels i and j are equal to c_i and c_j. The empirical distribution $\hat{P}^h_{\boldsymbol{C}_{i,j}}$ can then be expressed as a bivariate Bernoulli distribution with the pmf

$$\hat{P}^h_{\boldsymbol{C}_{i,j}}(\boldsymbol{c}_{i,j}) = \left(\hat{p}_{i,j,h}^{(00)}\right)^{(1-c_i)(1-c_j)} \left(\hat{p}_{i,j,h}^{(01)}\right)^{(1-c_i)c_j} \left(\hat{p}_{i,j,h}^{(10)}\right)^{c_i(1-c_j)} \left(\hat{p}_{i,j,h}^{(11)}\right)^{c_i c_j}. \tag{9}$$

We examine dependencies between pixels i and j under frequency distribution $\hat{P}^h_{\boldsymbol{C}_{i,j}}$ by computing p-values of the χ-squared independence test on C_i and C_j.

First, we observe dependencies in relation to the first pixel $i = 1$ by collecting the p-values for all $j \neq i$. This is repeated for all covers and depicted as a mean p-value map in Fig. 7 (first row).

Then, we observe the dependencies between all pixel pairs $i \neq j$ by collecting the mean p-value for each i against all $j \neq i$. This is repeated for all covers, and depicted as a mean p-value map in Fig. 7 (second row).

To get a better understanding of the distribution of p-values, we calculate the ratio of p-values $\leq 5\%$ per cover. A ratio histogram is presented in Fig. 8.

Discussion. The first row of Fig. 7 indicates low p-values of the independence test between the first pixel and its neighbors, while p-values increase for more distant pixels. This can be explained by the construction of parity-check matrix $\mathbb{H}$ in (1), as it induces linear dependencies. The first $h\alpha^{-1}$ pixels are involved in achieving parities according to the first h message bits. These dependencies are a cascading effect, as this relation is true for any pixel. This observation is done in context of a known permutation. In case of a fixed but unknown permutation, examining such dependencies potentially allows reconstruction of the embedding path, yielding the permutation σ.

An opposite border effect in the final part of the vector is observable in both rows of Fig. 7: last pixels tend to be independent from all the other ones. Again, this can be explained as result of the parity-check matrix $\mathbb{H}$ construction in (1). The last α^{-1} stego pixels only contribute to the last message bit. The only connection to other pixels is that the previous $h\alpha^{-1}$ pixels (together with the last

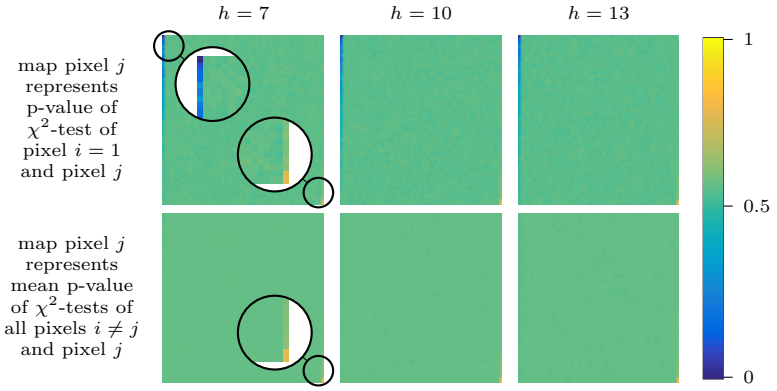

Fig. 7. Mean p-value map of chi-squared tests of first pixel against all other pixels j (first row), and of each pixel j against all other pixels (second row), j column-wise in order of coding, mean over all cover objects, $h = 7$ (left), 10 (middle), 13 (right).

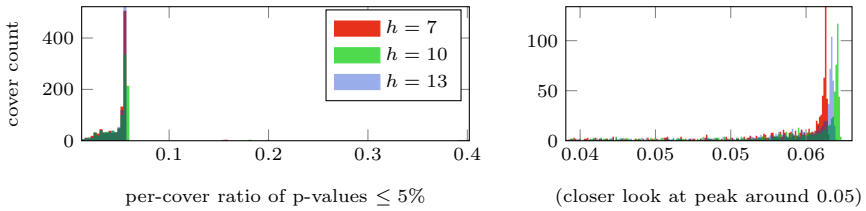

Fig. 8. Histograms of per-cover ratio of p-values $\leq 5\%$ of all ratios (left), and of ratios in the local neighborhood of the peak at around 0.05 (right).

message bit) determine whether one of the last α^{-1} pixels has to be changed to meet the correct parity. Thus, embedding changes at these pixels have only small cascading effects on the choices at other pixels. These dependency behaviors can only be observed, when knowledge about the permutation σ is given. The count of pixels with low dependencies equals the width of submatrix $\hat{\mathbb{H}}$ and thereby contains information about the ratio α.

What matters from a security point of view is that in case of an unknown but fixed permutation, an attacker can obtain and evaluate a histogram of p-values, such as the one in Fig. 8. The expected per-cover ratio of pixel pairs with p-values below 5%, would be 5% for independently chosen embedding changes. Instead we observe ratios around 5.6%. This not only adds information to the steganalysis decision, but may reveal information about the embedding path.

Interestingly, in our experiment, the mean ratio in case of $h = 10$ is higher than in case of $h = 13$. In other words, for $h = 10$ there are on average more pixel pairs fulfilling the independence test than for $h = 13$. This can be attributed to the values in submatrix $\hat{\mathbb{H}}$ in context of the set of distortion profiles, as the submatrix implies how dependencies are formed and propagated. Clearly, such analyses should guide the choice of secure submatrices in future work.

The previous discussion focused on the analysis of STC for different constraint lengths h. Besides its impact on the achievable security, the constraint length h has exponential impact on the computational complexity of STC.

4.4 Computational Performance

The empirical measurements in this section confirm the theoretical predictions (within the tested range) and inform the tradeoff between security and performance when choosing h in practice. To give some intuition regarding the computation time, Fig. 9 presents the time per pixel of an STC run given different scenarios. The computations are done on the LEO3E HPC [16], fitted with Intel Xeon E5-2650-v3 processors. The performance is evaluated on a single core at 2.3 GHz CPU clock speed and 8 GB RAM.

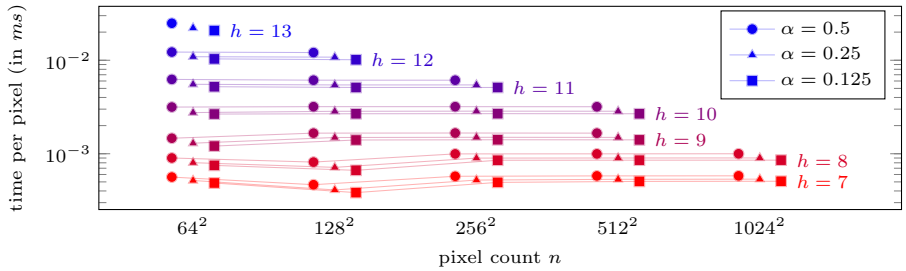

Fig. 9. STC computation time per pixel for different embedding ratios α, cover sizes n, and constraint lengths h.

Running STC, the time per pixel is mostly constant for cover sizes n and, expectedly, exponential in the constraint length h. Furthermore, the time per pixel increases slightly for increasing ratios α. These measurements fit the expected computational scaling due to the trellis size $2^h \times n(\alpha + 1)$.

5 Outlier Correction

Here we propose an improvement to STC which avoids security-critical outliers. We differentiate between outliers above and below the main diagonal in Fig. 4. Outliers above the main diagonal refer to pixels being changed with higher observed frequencies than optimal and are denoted as positive outliers. Outliers below the main diagonal are changed less than optimal and are denoted as negative outliers. Positive outliers are very risky as they might cause instances where steganalysis succeeds with certainty. Negative outliers do not impose an immediate security risk.

Positive outliers are an immediate result of the code construction, as previously discussed in Sect. 4.2. Mitigation attempts that modify the costs of the

leading pixels, e.g. by windowing, are futile as this cannot overcome the restrictions imposed by the parity-check matrix. A more viable approach would be to detect risky deviations post-embedding and repeat the embedding with another cover if necessary. However, this meddles with the separation of duties, makes embedding time less predictable, and comes close to the (insecure) practice of steganography by cover selection. We do not recommend this approach.

Alternatively, we suggest to modify the code construction and use a parity-check matrix $\mathbb{H}_{OC}$ by cropping the first $h-1$ rows of $\mathbb{H}$ (1) as follows:

$$\mathbb{H}_{\mathrm{OC}} = \begin{pmatrix} \ddots & \hat{\mathbb{H}} & \hat{\mathbb{H}} & 0\ldots0 & & \\ 0\ldots0 & \vdots & & \hat{\mathbb{H}} & 0 & \\ 0\ldots0 & 0\ldots0 & 0\ldots0 & & & \\ 0\ldots0 & 0\ldots0 & 0\ldots0 & & & \\ & & & \ddots & & \\ & & & & \hat{\mathbb{H}} & 0\ldots0 \quad 0\ldots0 \\ 0 & & & & \hat{\mathbb{H}} & \ddots \quad 0\ldots0 \end{pmatrix} . \tag{10}$$

We refer to this modification as *outlier corrected* Syndrome Trellis Coding (OC-STC), noting that it corrects risky (positive) outliers only. The resulting parity-check matrix differs in one important property: each row of $\mathbb{H}_{\mathrm{OC}}$ contains each element of $\hat{\mathbb{H}}$ exactly once. However, cropping the first $h-1$ rows shortens the payload by $h-1$ bits. For the sake of presenting the impact of OC-STC correctly, we recalculate the optimal change probabilities based on the lowered payload. Therefore, a new scaling λ' according to the reduced maximal message entropy is chosen. OC-STC can be solved in the same time with the same Viterbi algorithm as STC. Figure 10 demonstrated (by example) that using parity-check matrix $\mathbb{H}_{\mathrm{OC}}$ successfully mitigates positive outliers. A more thorough investigation is left for future research.

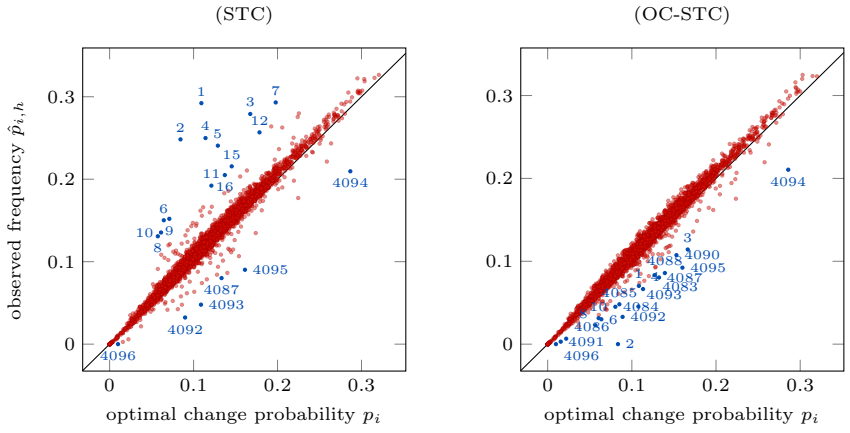

Fig. 10. Exemplary comparison of optimal change probabilities p_i and observed frequencies $\hat{p}_{i,h}$ as in Fig. 4, for default STC (left) and the proposed OC-STC (right).

6 Concluding Remarks

The results of the first (to the best of our knowledge) experimental analysis tailored to explore the statistical properties of Syndrome Trellis Coding (STC) confirms the trust it enjoys from the community: in general, STC does a good job. Even though we chose a difficult steganography setup with small covers and long messages, STC closely approximates the optimal change probabilities, as supported by the relatively small bias (coding loss) and its balanced distribution over the dimensions analyzed. Also the computational cost scales as expected.

However, it is worth noting that STC produces outliers at the boundaries of the cover vector, which seem risky in particular at the leading elements (shielded in many practical systems only by the key-dependent permutation). Using the default parity-check matrix construction, the first positions of the embedding path are prone to be changed significantly more often than optimal or intended. Our proposed modification OC-STC mitigates positive outliers by using a different construction for the parity-check matrix.

OC-STC, as STC, still produces negative outliers, in the sense of pixels being changed less frequently than prescribed by the optimal change probabilities. Negative outliers do not immediately induce a security concern, although it would be desirable to use all available pixels to their full capacity. This is an avenue for future work.

Another relevant insight gained from this work is the possibility to evaluate pairwise dependencies in the change vector. This analysis does not require knowledge of the permutation and is thus possible for attackers as soon as the permutation is fixed for sufficiently many objects. More research is needed to assess the practical security loss by using the dependency structure in the steganalysis decision directly; or by indirectly trying to recover the embedding path from the dependency structure, exploiting knowledge of the locality of pairwise dependencies as a result of message encoding with STC (and OC-STC).

Finally, the discovered non-monotonic relation between h and the ratio of independence-rejected pixels motivates to look deeper into the specifics of how the choice of submatrix $\hat{\mathbb{H}}$ influences the measurable formation of dependencies.

In conclusion, this research has highlighted that the effect of coding on steganographic security leaves relevant open questions. It is worth recalling that the results presented here (and suggested for follow-up work) do not immediately invalidate research on steganographic security that follows the common practice of simulating change vectors and thus assumes optimal encoding. Rather, these results should be seen as upper bounds for the security of steganographic systems that replace the simulation with STC for real messages.

Acknowledgements. Alexander Schlögl helped us with implementing STC on the HPC. Pascal Schöttle and the anonymous reviewers of IWDW provided us with very valuable comments. The computational results presented have been achieved using the HPC infrastructure "LEO" of the University of Innsbruck. This research was supported by Archimedes Privatstiftung, Innsbruck, and by Deutsche Forschungsgemeinschaft (DFG) under the grant "Informationstheoretische Schranken digitaler Bildforensik".

References

[1] Bas, P., Filler, T., Pevnỳ, T.: "Break our steganographic system": The ins and outs of organizing BOSS. In: Filler, T., Pevný, T., Craver, S., Ker, A. (eds.) Information Hiding (13th International Conference). Lecture Notes in Computer Science, vol. 6958, pp. 59–70. Springer, Berlin, Heidelberg (2011)

[2] Carnein, M., Schöttle, P., Böhme, R.: Predictable rain? Steganalysis of public-key steganography using wet paper codes. In: ACM Information Hiding and Multimedia Security Workshop. pp. 97–108. Salzburg, Austria (2014)

[3] Filler, T., Fridrich, J., Judas, J.: Syndrome Trellis Coding, Binghamton reference implementation. `http://dde.binghamton.edu/download/syndrome/` (last access: June 2017)

[4] Filler, T., Fridrich, J.: Gibbs construction in steganography. IEEE Transactions on Information Forensics and Security 5(4), 705–720 (2010)

[5] Filler, T., Fridrich, J.: Minimizing additive distortion functions with non-binary embedding operation in steganography. In: IEEE International Workshop on Information Forensics and Security (WIFS). pp. 1–6. Tenerife, Spain (2010)

[6] Filler, T., Judas, J., Fridrich, J.: Minimizing embedding impact in steganography using trellis-coded quantization. In: Proceedings of SPIE-IS&T Electronic Imaging: Security, Forensics, Steganography and Watermarking of Multimedia Contents X. pp. 754105–754105. San Jose, CA (2010)

[7] Filler, T., Judas, J., Fridrich, J.: Minimizing additive distortion in steganography using syndrome-trellis codes. IEEE Transactions on Information Forensics and Security 6(3), 920–935 (2011)

[8] Fridrich, J.: Minimizing the embedding impact in steganography. In: ACM Multimedia and Security Workshop. pp. 2–10. Geneva, Switzerland (2006)

[9] Fridrich, J.: Steganography in digital media: principles, algorithms, and applications. Cambridge University Press (2009)

[10] Holub, V., Fridrich, J.: Designing steganographic distortion using directional filters. In: IEEE International Workshop on Information Forensics and Security (WIFS). pp. 234–239. Tenerife, Spain (2012)

[11] Ker, A.D.: Locating steganographic payload via WS residuals. In: ACM Multimedia and Security Workshop. pp. 27–32. Oxford, UK (2008)

[12] Li, B., Wang, M., Huang, J., Li, X.: A new cost function for spatial image steganography. In: IEEE International Conference on Image Processing (ICIP). pp. 4206–4210. Paris, France (2014)

[13] Pevnỳ, T., Ker, A.D.: Steganographic key leakage through payload metadata. In: ACM Information Hiding and Multimedia Security Workshop. pp. 109–114. Salzburg, Austria (2014)

[14] Reed, I.S., Chen, X.: Error-control coding for data networks. Springer Science & Business Media, New York, US (2012)

[15] Sedighi, V., Cogranne, R., Fridrich, J.: Content-adaptive steganography by minimizing statistical detectability. IEEE Transactions on Information Forensics and Security 11(2), 221–234 (2016)

[16] University of Innsbruck: Supercomputer LEO3E. `https://www.uibk.ac.at/zid/systeme/hpc-systeme/leo3e/` (last access: June 2017)

[17] Wang, Y.H.: On the number of successes in independent trials. Statistica Sinica 3(2), 295–312 (1993)