Exploring Non-Additive Distortion in Steganography

Tomáš Pevný
Dept. of Computer Science
CTU in Prague
pevnak@gmail.com

Andrew D. Ker
Dept. of Computer Science
University of Oxford
adk@cs.ox.ac.uk

ABSTRACT

Leading steganography systems make use of the Syndrome-Trellis Code (STC) algorithm to minimize a distortion function while encoding the desired payload, but this constrains the distortion function to be additive. The Gibbs Embedding algorithm works for a certain class of non-additive distortion functions, but has its own limitations and is highly complex.

In this short paper we show that it is possible to modify the STC algorithm in a simple way, to minimize a non-additive distortion function suboptimally. We use it for two examples. First, applying it to the S-UNIWARD distortion function, we show that it does indeed reduce distortion, compared with minimizing the additive approximation currently used in image steganography, but that it makes the payload more – not less – detectable. This parallels research attempting to use Gibbs Embedding for the same task. Second, we apply it to distortion defined by the output of a specific detector, as a counter-move in the steganography game. However, unless the Warden is forced to move first (by fixing the detector) this is highly detectable.

KEYWORDS

Steganography, Syndrome-Trellis Codes, Distortion Minimization

1 MOTIVATION

The Prisoners' Problem models steganography as follows: Alice (the sender and steganographer) wishes to send secret messages to Bob (the receiver) without raising the suspicion of the warden Eve (the steganalyst), who inspects all messages for illicit content. Alice and Bob use steganography to hide their secret messages inside innocuous-looking objects, as invisibly as possible, while Eve wants to detect hidden messages as accurately as possible. Thus Alice and Eve have antagonistic goals.

In steganography by cover modification the embedding function

$$f_{\text{emb}} : \mathcal{X} \times \mathcal{M} \times \mathcal{K} \mapsto \mathcal{X}$$

accepts the cover object $\mathbf{x} \in \mathcal{X}$, message $\mathbf{m} \in \mathcal{M}$, and secret key $k \in \mathcal{K}$, and produces a stego object $\mathbf{y} \in \mathcal{X}$ containing the hidden message. This can be extracted by a function

$$f_{\text{ext}} : \mathcal{X} \times \mathcal{K} \mapsto \mathcal{M}$$

provided with the correct key k. The sets $\mathcal{X}$, $\mathcal{M}$, and $\mathcal{K}$ correspond to the spaces of cover and stego objects (in this paper, digital images), messages, and keys respectively.

Current state of the art embedding functions (steganographic algorithms) are built on the principle of distortion minimization [6]: while embedding the message $\mathbf{m}$ into $\mathbf{x}$, Alice tries to minimize some *distortion function*

$$f_{\text{dis}} : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}_0^+.$$

The distortion function should be related to detectability, i.e. the smaller the distortion $f_{\text{dis}}(\mathbf{x}, \mathbf{y})$, the smaller the chance that Eve detects covert communication. During embedding the algorithm therefore tries to solve the following optimization problem

$$\arg\min_{\mathbf{y} \in \mathcal{X}} f_{\text{dis}}(\mathbf{x}, \mathbf{y}) \text{ subject to } f_{\text{ext}}(\mathbf{y}, k) = \mathbf{m}.$$

This is an NP-complete problem for general extraction and distortion functions, and remains NP-complete when – as is often the case – the extraction functions are restricted to linear maps $f_{\text{ext}}(\mathbf{y}, k) = \mathbf{H}\mathbf{P}_{\mathbf{k}}\mathbf{m}$, where $\mathbf{P}_{\mathbf{k}}$ is some key-dependent permutation matrix, and $\mathbf{H}$ a fixed parity-check matrix of some linear code (so-called *syndrome coding*). However, for *additive* distortion functions, given by the sum of local distortions caused by changing individual pixels independently, Syndrome-Trellis Coding (STC) [6] allows finding a solution, typically within 5–7% of the optimum [20]. The computational efficiency and near-optimality of STC have narrowed the design of new steganographic algorithms to the search for better additive distortion functions, as the coding seems to be solved.

There does exist a technique for optimizing non-additive distortion functions, the Gibbs Embedding method of [5], much more complex than STCs, requires multiple sweeps through the stego object (of an uncertain number), and is restricted to the class of distortion functions which are *sums of local potentials*. It was used to boost the performance of the additive distortion in HUGO in [9], and for HILL [11] and MVG [19] to promote changes of nearby pixels in the same direction in [4, 12]. In [8] it was applied to the non-additive S-UNIWARD distortion function [9], but this failed to improve on the additive version, due in part to a technical limitation of the Gibbs sampler.

This paper explores another method for minimizing non-additive distortion functions, a variation on the STC algorithm (Sect. 2). It does *not* find optimal solutions but it avoids some of the limitations of Gibbs Embedding: it is easy to understand, relatively cheap to implement, and does not suffer the same technical limitation.

We will test this so-called *variable-cost STC* against the non-additive S-UNIWARD distortion function, that is typically minimized only in an additive approximation via the standard STC algorithm (Sect. 3). Our method does not fail in the same way that Gibbs embedding appears to in [8], but we will discover parallel results: we will find lower distortion than the additive approximation, but the stego objects will be more detectable. This casts doubt on the link between UNIWARD distortion (do not confuse with its additive approximation) and detectability.

We also show how our method can be used if the embedder knows exactly the detector that Eve uses: a kind of adversarial embedding (Sect. 4), something not necessarily possible even with Gibbs Embedding. It allows the embedder to optimize against one specific detector, although this only makes them more detectable by others.

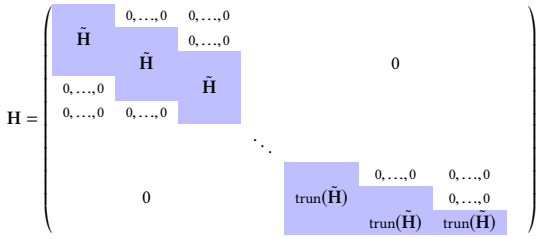

Figure 1: Block diagonal matrix used in Syndrome-Trellis Codes, as proposed in [20]. trun($\tilde{H}$) denotes $\tilde{H}$ with bottom rows appropriately removed.

This work is aimed at exploring some consequences of non-additive distortion. We do not claim that the variable-cost STC method is optimal, or even particularly good, and the results for S-UNIWARD distortion raise more questions than they answer. We simply hope that these questions stimulate research different from micro-optimization of additive distortion functions.

2 SYNDROME-TRELLIS CODES

Our codes will use alphabet $\Sigma = \{0, 1, 2, \ldots, q-1\}$ and all operations will be performed in mod q arithmetic unless otherwise specified. Let cover and stego objects be represented as vectors $\mathbf{x} \in \Sigma^n$, $\mathbf{y} \in \Sigma^n$ respectively, with n being the number of pixels. For simplicity of notation, we assume that the cover and stego object have already been subject to some key-dependent permutation. This is independent of the rest of the procedure and the key will be omitted from now on.

The communicated message is also a q-ary vector $\mathbf{m} \in \Sigma^m$ with $m < n$ being the length of the message. The cover object $\mathbf{x}$ is modified to $\mathbf{y}$ such that the message $\mathbf{m}$ is communicated as a syndrome of a parity check matrix $\mathbf{H} \in \Sigma^{m,n}$, i.e.

$$\mathbf{H}\mathbf{y} = \mathbf{m}.$$

Since $m < n$, the solution of $\mathbf{H}\mathbf{y} = \mathbf{m}$ is not unique. This freedom is used to select $\mathbf{y}$ which not only communicates the message but also minimizes distortion measured by the function $f_{\text{dis}}(\mathbf{y}, \mathbf{x})$.

Ref. [6] proposed the Viterbi algorithm to solve the above problem, provided that *(i)* the distortion function is additive, i.e.

$$f_{\text{dis}}(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{n} f_{\text{dis}}(\mathbf{x}, \mathbf{y}_i),$$

where $f_{\text{dis}}(\mathbf{x}, \mathbf{y}_i)$ is a distortion between cover image $\mathbf{x}$ and stego image $\mathbf{y}$ differing from $\mathbf{x}$ only at the i-th pixel, and *(ii)* $\mathbf{H}$ is a sparse block-diagonal matrix constructed by repeatedly placing sub-matrix $\tilde{\mathbf{H}} \in \Sigma^{h \times \frac{n}{m}}$ next to each other and shifted down by one (see Figure 1)[1]. The parameter h is known in convolutional codes as the *constraint height*, and its value affects the performance of the algorithm: larger h will find solutions with lower distortion, at higher computational cost.

[1]For clarity of explanation it is assumed that the payload length m is an integer divisor of the cover size n. Otherwise, the banded structure of the matrix $\mathbf{H}$ can be achieved by interleaving sub-matrices $\tilde{\mathbf{H}}$ of different widths, and some of the indexing in the algorithm must be adapted accordingly.

2.1 The Viterbi Algorithm

We now use the following notation. For $\mathbf{x}$ a vector of length n, and $0 < i \le j \le n$ integers, $\mathbf{x}_{i:j}$ is the sub-vector of $\mathbf{x}$ consisting of elements $(\mathbf{x}_i, \mathbf{x}_{i+1}, \ldots, \mathbf{x}_j)$. If this happens to access a vector beyond its length, which will happen as our embedding reaches the end of the cover object, we can pad the answer with zeros. $\mathbf{e}_i$ will denote the i-th basis vector (with n implicit). Let $\mathbf{H} \in \Sigma^{m,n}$ be a matrix and i, j as before, then $\mathbf{H}_{i:j}$ is a sub-matrix containing only columns $(i, i+1, \ldots, j)$ of $\mathbf{H}$. Finally, $\mathbf{H}_i$ represents the single column i of $\mathbf{H}$.

We will describe the Viterbi algorithm in a style different from [6], avoiding explicit construction of the trellis. Perverse as it may seem to remove the Trellis from Syndrome-Trellis Code, it will present a classical algorithm from an alternative view, and also allow us to show simply the modification to non-additive distortions, albeit not optimally minimized.

The algorithm maintains three variables:

$$\begin{aligned} i &\in \mathbb{N}, \\ j &\in \mathbb{N}, \\ \mathcal{S} &\subseteq \Sigma^n \times \mathbb{R} \times \Sigma^h, \end{aligned}$$

where i represents the number of cover pixels completed, j the number of payload symbols correctly and optimally encoded, and $(\mathbf{y}, d, \mathbf{s}) \in \mathcal{S}$ if stego object $\mathbf{y}$ encoded such payload, with distortion d, and $\mathbf{s} = (\mathbf{H}_{1:i}\mathbf{y}_{1:i})_{(j+1):(j+h)}$. This last quantity is called the *partial syndrome* in [6], where its evolution is traced by the trellis.

We can express these properties formally by invariants $0 \le i \le n$, $0 \le j \le m$, and $(\mathbf{y}, d, \mathbf{s}) \in \mathcal{S}$ implies $(\mathbf{H}\mathbf{y})_{1:j} = \mathbf{m}_{1:j}$, $d = f_{\text{dis}}(\mathbf{x}, \mathbf{y})$, $\mathbf{s} = (\mathbf{H}_{1:i}\mathbf{y}_{1:i})_{(j+1):(j+h)}$, and if $\mathbf{y}'$ satisfies the same properties then $f_{\text{dis}}(\mathbf{x}, \mathbf{y}) \le f_{\text{dis}}(\mathbf{x}, \mathbf{y}')$. The invariant is established with *initialization*

$$i := 0,\ j := 0,\ \mathcal{S} := \{(\mathbf{x}, 0, \mathbf{0})\}.$$

At each stage we explore the space of possible partial syndromes, in what we might call an *expansion step*: set $i := i + 1$, identify the column of $\tilde{\mathbf{H}}$ corresponding to i^{th} pixel[2] and call it $\mathbf{h}$, then set

$$\mathcal{S} := \left\{ \left(\mathbf{y} + k_i \mathbf{e}_i, f_{\text{dis}}(\mathbf{x}, \mathbf{y} + k_i \mathbf{e}_i), \mathbf{s} + k_i \mathbf{h} \right) \,\middle|\, (\mathbf{y}, d, \mathbf{s}) \in \mathcal{S},\ k_i \in \Sigma \right\}.$$

Then we perform a *greedy decimation step*, keeping only the optimal stego objects for each partial syndrome: for each pair $(\mathbf{y}_1, d_1, \mathbf{s}) \in \mathcal{S}$ and $(\mathbf{y}_2, d_2, \mathbf{s}) \in \mathcal{S}$, keep only the first if $d_1 < d_2$, otherwise keep the second (ties may be broken arbitrarily). This ensures that $|\mathcal{S}| \le q^h$, as there are only q^h distinct partial syndromes.

There is one further step in the case that we reached the final column of a block in $\mathbf{H}$.[3] In such cases we perform a *payload matching* step: set $j := j + 1$, then remove stego objects which do not match the first j payload symbols and restore the invariant by setting

$$\mathcal{S} := \left\{ (\mathbf{y}, d, \mathbf{s} \ll 1) \,\middle|\, (\mathbf{y}, d, \mathbf{s}) \in \mathcal{S} \wedge \mathbf{s}_1 = \mathbf{m}_j \right\},$$

where $\ll 1$ denotes a left-shift operation on a vector in Σ^h, padding with zero on the right. The condition ensures correctness of payload symbol j because of the invariant and the banded structure of $\mathbf{H}$. Note that $\mathcal{S}$ is nonempty as long as the first row of $\tilde{\mathbf{H}}$ is not all zeros.

[2]When m is an integer divisor of n, it is $\tilde{\mathbf{H}}_k$ with $k = (i-1) \bmod (n/m) + 1$.
[3]When m is in an integer divisor of n, if $i = kn/m$ for integer k.

The sequence *expansion*, *greedy decimation*, and sometimes *payload matching*, are iterated until $i = n$ and $j = m$. Then choose the l such that d_l is least in $(\mathbf{y}_l, d_l, \mathbf{s}_l) \in \mathcal{S}$; the corresponding $\mathbf{y}_l$ is the stego object that, by the invariant, correctly encodes payload $\mathbf{m}$ with lowest distortion. The optimality of the solution relies on the *Principle of Optimality* of dynamic programming: if $\mathbf{y}$ is the optimal solution to the entire problem then $\mathbf{y}_{1:n'}$ is necessarily the optimal solution to the subproblem consisting of the first $n' < n$ pixels (and the corresponding length of payload). Thus the greedy decimation step, which filters such solutions, cannot exclude an optimal solution to the entire problem.

In the case of additive distortion, a simplification may be made. We precalculate per-change costs

$$c_{i,k} = f_{\text{dis}}(\mathbf{x}, \mathbf{x} + k\mathbf{e}_i)$$

and at the expansion step we need not recalculate $f_{\text{dis}}(\mathbf{x}, \mathbf{y} + k\mathbf{e}_i)$ since, by additivity, it must equal $f_{\text{dis}}(\mathbf{x}, \mathbf{y}) + c_{i,k}$. In this case we could recover the trellis of the classical Viterbi algorithm by storing, instead of the entirety of $\mathbf{y}$ for each member of $\mathcal{S}$, the sequence of k_i: at the end of the algorithm then performing a *backward pass* through the trellis to recover $\mathbf{y}$.

2.2 Variable-Cost STCs

With our description the entire stego object is available at each i and for each partial syndrome $\mathbf{s}$. This allows recalculating the distortion function at each expansion step, applying the same procedure to the case of non-additive distortion functions. The algorithm becomes a heuristic, and optimality is not necessarily (or even probably) still true: in non-additive distortion, the Principle of Optimality does not hold. If the optimal solution to the entire problem is $\mathbf{y}$, it is not necessarily the case that $\mathbf{y}_{1:n'}$ is the optimal solution to the subproblem consisting of the first $n' < n$ pixels, since early sub-optimal decisions might unlock better solutions further along. But the method can still be used for steganography because the correctness of the payload remains true.

This version of the algorithm will be called the *variable-cost* STC, as it corresponds to the classical STC in which outgoing costs are updated at every state of the trellis. Note that the time complexity of the original STC is $O(nhq^{h+1})$ arithmetic operations to maintain $\mathcal{S}$ (n expansion steps, each with up to q^h partial syndromes, each with q outgoing edges; by storing only k_i these operations can all be on vectors of length at most h) and only $O((q-1)n)$ calls on the distortion function when the costs are precomputed for each location. For the variable-cost STC, there are the same number of vector operations, but because we maintain a complete stego object at each step they are vectors of length n: thus $O(n^2q^{h+1})$ arithmetic operations to maintain $\mathcal{S}$. More significantly, we now require $O(nq^{h+1})$ calls on the distortion function, one for each possible change for each partial syndrome. Since in the practice of digital image steganography, distortion functions involve multiple filters on the entire image, this is a significant overhead, as will be reported in Section 3.

This method can be contrasted with Gibbs Embedding [5]. It is conceptually much simpler, and can be accomplished with minor modifications to the standard STC algorithm. It does not require multiple sweeps; indeed, at the start of the cover image it functions rather like the first Gibbs sweep, and at the end rather like the second. There is no need to divide the payload amongst sublattices, nor does *erasure entropy* penalize us when the changes are clustered: we will explore this in the next section.

3 OPTIMIZATION OF S-UNIWARD

The variable-cost STC was tested in spatial domain steganography using S-UNIWARD [9] with ternary alphabet $q = 3$, implemented as ± 1 embedding changes. The layered construction [6] was *not* used, as it causes additional coding loss.

The reasons for choosing this distortion function are *(i)* it is still a leading steganographic scheme, *(ii)* it was derived from a non-additive cost function that depends on the entire cover and stego object, unlike other leading cost-assignment rules [11, 18] which are pixel-wise and additive by design, and *(iii)* a similar experiment was attempted with Gibbs Embedding in [8].

The UNIWARD distortion function originally proposed in [9] is

$$f_{\text{dis}}(\mathbf{x}, \mathbf{y}) = \sum_{k=1}^{3} \sum_{u=1}^{n_1} \sum_{v=1}^{n_2} \frac{|W_{uv}^{(k)}(\mathbf{x}) - W_{uv}^{(k)}(\mathbf{y})|}{|W_{uv}^{(k)}(\mathbf{x}) + 1|}, \tag{1}$$

where $W_{uv}^{(k)}(\mathbf{x})$ are u, v pixels of the image $\mathbf{x}$ convolved with a k^{th} filter; the exact filters are irrelevant to this paper so we direct the reader to [9] for details.

Because UNIWARD was designed for Syndrome-Trellis Codes, it must make an additive approximation to the non-additive distortion function (1). In the case of spatial-domain images and ± 1 embedding changes, the reference implementation of S-UNIWARD[4] uses $f_{\text{dis}}(\mathbf{x}, \mathbf{y}) = \sum_{u,v} \rho_{uv}(\mathbf{x}, \mathbf{y})$, where

$$\rho_{uv}(\mathbf{x}, \mathbf{y}) = \sum_{k=1}^{3} \frac{|x_{uv} - y_{uv}|}{|W_{uv}^{(k)}(\mathbf{x}) + 1|}, \tag{2}$$

is the distortion at pixel location (u, v).

In [8], the Gibbs Embedding algorithm was applied to the original, non-additive, UNIWARD distortion function, and the results compared against the additive approximation. Gibbs Embedding can be applied because the form of (1) is a sum of local potentials. To paraphrase the results obtained, more sweeps of the Gibbs algorithm caused *increased* distortion, and the cause was traced to a limitation of the Gibbs sampler: it only achieves the *erasure entropy* of the conditionally-independent subfields [5, VI.C], which is strictly lower than the entropy of the entire Gibbs field. In the case of this distortion function, the sampler attempts to cluster changes very tightly, which in turn reduces the erasure entropy, forcing higher numbers of embedding changes.

We can use variable-cost STCs to investigate the same question: with this heuristic, the erasure entropy problem is not present. We also need not perform multiple sweeps and try to guess when to stop. On the other hand, unlike Gibbs Embedding we do not converge to a true optimum. Compared with standard STCs optimizing the additive approximation we pay a price in complexity, because there are many more distortion computations: Table 1 shows that it is approximately 15 times slower for this size of image and distortion function, even after we optimized the calculation of distortion so

[4]Reference implementation is provided at http://dde.binghamton.edu/download/stego_algorithms/download/S-UNIWARD_matlab.zip.

	payload (bpp)		
method	0.1	0.2	0.3
embedding time (mm:ss)			
additive	1:07	1:06	1:07
non-additive	17:07	16:30	16:04
distortion achieved			
additive	18871	35973	57089
non-additive	11747	24184	40953
additive optimal	13196	27408	42231

Table 1: Average embedding time per image, and distortion as measured by Equation (1), for different payload sizes. The images were grayscale 512×512, and times measured on Amazon's AWS `m4.2xlarge` instance (Intel® Xeon® CPU E5-2686 with 32Gb of RAM).

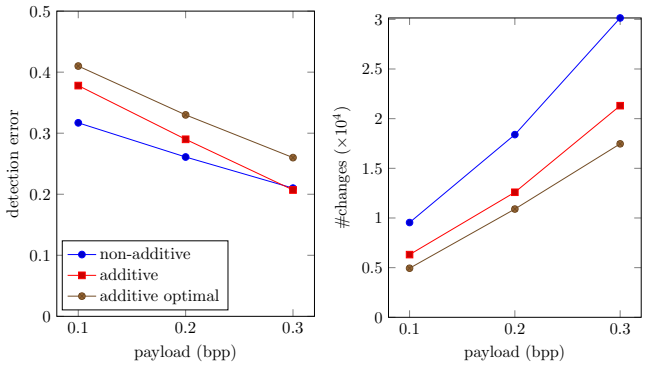

Figure 2: Error of linear detectors (left), and number of changes made during embedding (right), for additive cost via STC, non-additive costs via variable-cost STC, and simulated optimal additive costs.

that only the part of the image affected by each embedding change is updated, and we must admit that our algorithm will not find the global minimum distortion. Nonetheless, the results tell a story, and parallel observations in [8].

Using both the additive S-UNIWARD distortion function (2), and the non-additive original (1), we embedded payloads corresponding to 0.1, 0.2, and 0.3 bits per pixel into the 10000 never-compressed images of BOSSBase v1.01 [1]. To isolate unnecessary variation, for each image we generated a random payload, random sub-matrix $\tilde{\mathbf{H}}$, and random permutation of the pixels; then used the same choices for both STC and variable-cost STC embedding of that image.

The height of the sub-matrix $\tilde{\mathbf{H}}$ was set to three, which means that the algorithm explored $3^3 = 27$ partial syndromes. This low number was used to reduce the computational complexity, but we must admit that it may take both versions of the STC far from the optimum. In order to estimate this effect (called coding loss), the same relative payload was embedded using *simulated optimal coding* for the additive distortion function (we call this *additive optimal* simulation). According to [6], the optimal probability of making change $k\mathbf{e}_i$, i.e. adding $k \in \Sigma$ to location i, is

$$\pi_i(k) = \frac{e^{-\lambda f_{\text{dis}}(\mathbf{x}, \mathbf{x}+k\mathbf{e}_i)}}{\sum_{k' \in \Sigma} e^{-\lambda f_{\text{dis}}(\mathbf{x}, \mathbf{x}+k'\mathbf{e}_i)}} \tag{3}$$

where λ is determined by $\sum_{i=1}^{n} \sum_{k \in \Sigma} -\pi_i(k) \log_2 \pi_i(k) = m$.

The detectability of each steganographic scheme, at each payload, has been estimated by the probability of error of a Fisher Linear Classifier [2] that has used the full set of 34671 dimensional SRM steganalytic features [7]. These features are one of the standard benchmarks for contemporary steganalysis, amongst those that do not exploit knowledge of the selection channel. The classifier was trained for each embedding method separately following the standard setting, where half of the images were left for testing and the tolerance parameter was optimized by grid-search on values $\{10^{-5}, 10^{-4}, \ldots, 10^{15}\}$, with the probability of error estimated by five-fold cross-validation.

Table 1 shows the average distortion of stego-images measured by Equation (1) for the three compared methods on the three payloads. As expected, minimizing the additive approximation results in total distortion which is higher than minimizing the true non-additive distortion via variable-cost STC. The non-additive version optimizing the true distortion is on average better by 30%. We can be sure that this is not simply a poor performance by the standard STC due to the relatively small value of h, because the distortion the variable-cost STC is still lower than the optimal distortion calculated by (1). This demonstrates that the proposed variable-cost modification of STC is able to find better solutions when minimization of the original UNIWARD distortion is the aim.

However, Figure 2 (left) shows the error of steganalyzers for these steganographic schemes. Optimizing original UNIWARD, despite giving lower distortion, is more detectable. (The only exception is on the payload 0.3 bpp, where it is better by an insignificant 0.3%.) This parallels the results in [8], though in our case it is not due to the erasure entropy limitation of Gibbs samplers. Instead, it suggests that UNIWARD distortion (1) is simply not as related to steganographic security (as measured by this detector) as was originally thought. Its additive approximation is a better measure.

Similarly to [8], we explored the number and patterns of changes caused by minimizing the non-additive distortion. Figure 2 (right) shows the average number of embedding changes for all three schemes on all three payloads. The non-additive algorithm makes more embedding changes, apparently finding ways to reduce distortion by adding changes that, to some extent, cancel out the effects of others. An additive approximation, when the costs are all positive, never makes such a choice, implicitly regularizing the number of embedding changes. In this sense, the additive approximation corrects a flaw of the S-Uniward's distortion function.

We also explored whether the non-additive distortion function tends to cluster its changes or align nearby changes in the same direction: such synchronization has been recently proposed in [4, 12, 21], as an adjunct to additive distortion functions. Figure 3 shows changes of the image under a single message. A closer look shows that changes of the non-additive version are slightly more clustered and aligned. We measured these by, for each changed pixel, counting the number of aligned versus non-aligned changes in its 3×3 neighborhood. For the additive S-UNIWARD, the proportion is 0.760 : 0.758, whereas the same ratio for the non-additive method

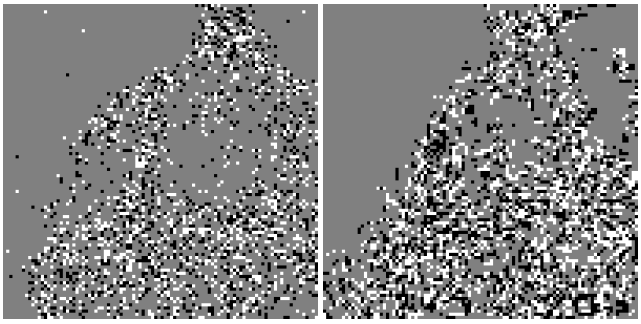

Figure 3: Embedding changes made during embedding of a message of 0.3 bpp, using additive (left) and non-additive (right) S-UNIWFeRD distortion function, in a crop from image number 1013 from BOSSBase v1.01. A white pixel indicates a positive change, grey indicates no change, and a black pixel indicates a negative change.

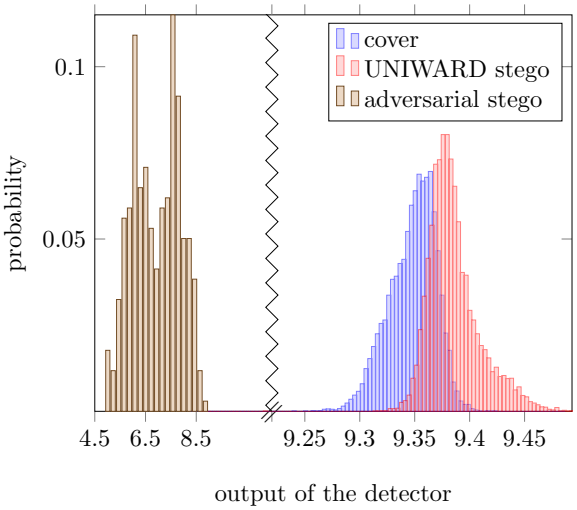

Figure 4: Histogram of outputs of the detector on cover images, images containing 0.3 bpp embedded by additive S-UNIWARD, and images containing the same message embedded by variable-cost STC minimizing the output of the detector ('adversarial stego'). The detector is targeted to the S-UNIWARD images, and uses first-order SRM features.

is 1.941 : 1.098. This is more evidence in favor of the principle of synchronizing nearby changes [4, 12, 21]. To some extent these parallel the results seen at the end of Chapter 6 of [8], but comparing with its Figure 6.4.2 we see that variable-cost STCs cluster and align changes to a much lesser degree than Gibbs Embedding. This provides probably because it does not find an optimal solution to the minimum distortion.

4 DISTORTION DEFINED BY A DETECTOR

Consider a scenario where Alice knows the steganalytic detector used by Eve. Strange as it may sound, it occurs in security domains and commonly-used antivirus software is a prime example: miscreants can (and do) use them to test detectability of their new malicious products, and there are even specialized services for this job. Another example are Oracle attacks [3] in watermarking where the attacker has unrestricted access to the detector (oracle).

Early steganographic algorithms implicitly considered this scenario, as they frequently targeted a particular type of steganalysis. For example, OutGuess's [15] statistical restoration aimed to preserve first-order statistics of DCT coefficients in JPEG images. Similarly, Model-Based Steganography [17] aimed to preserve first order statistics of individual DCT modes and statistics of pixels on the border of DCT blocks (avoiding 'blockiness' artifacts).

Subsequent work of a similar nature assumed Eve to base her detector on a particular steganalytic feature set, and attempted to stay undetectable with respect detectors utilizing them. HUGO [14] used a heuristically-defined distortion measure based on SPAM features [13]. This has been further improved by [10, 20], defining pixelwise costs from the error of L2-Support Vector Machines or Fisher Linear Discriminant classifiers. More generally, when particular detectors are used to assess the security of new algorithms that depend on parameters, such parameters are implicitly being optimized towards undetectability by those detectors.

A steganographic detector can be represented as a function $f : \mathcal{X} \mapsto \mathbb{R}$. If the output exceeds some threshold, then the object is classified as stego, otherwise as cover. Given a method for optimizing arbitrary functions while coding a message, such f can be used as a distortion function itself. But the distortion is no longer a proper name of f, because f measures *probability* or *likelihood* of image being classified as stego. By minimizing its output Alice tries to create objects that will not trip the detector, even if they are not close to covers. The same idea can be applied to any detector: those based on a combination of steganalytic features and machine-learning classifiers [7], or those based on convolutional neural networks (CNN) [16]. But STCs and Gibbs Embedding cannot be used, because the distortion is unlikely to be additive or a sum of local potentials. Variable-cost STCs could provide a method, albeit being suboptimal. It is not necessary to understand the inner workings of the detector, as it can be treated as an oracle; all that is needed is a optimization heuristic.

To demonstrate the application of variable-cost STCs to this problem, we set Eve's detector as a linear classifier utilizing the 3588 first-order part of SRM features [7] (choosing only the subset for reasons of computational complexity). Denoting by $\phi(x)$ the SRM feature extraction function, the detector is implemented as $f(x) = w^{\mathrm{T}}\phi(x)$, where w is the solution of L_2-regularized Fisher's Linear Discriminant [2] detecting stego images created by additive S-UNIWARD with payload 0.3 bpp. The regularization parameter was chosen such that it minimizes error on unseen images, estimated by five-fold cross-validation. This detector has error rate 0.28 on images embedded by S-UNIWARD with payload 0.3.

Figure 4 shows the distributions of outputs of the detector on cover images, stego images embedded by S-UNIWARD with payload 0.3, and stego images created by the proposed variable-cost STC hiding the same payload. For reasons of complexity, the last class only contained 500 images. Stego images created by S-UNIWARD have higher values of $w^{\mathrm{T}}\phi(x)$, which is consistent with the aims of the detector. Cover images are located closer to the center, and a

practical detector will have a threshold to the cover distribution. Finally, stego images created by Alice knowing Eve's detector are far off to the left, with very low values of $w^{\mathrm{T}}\phi(x)$. For this specific detector, they would be classified as covers.

Of course, they are detectable. As is evident from the histogram, even the exact same detector, but swapping the sign of its output, would classify them as stego objects with perfect accuracy. The embedding method in this form can only be recommended if Alice is completely certain that Eve has fixed her detector. How to use it in practice is a subject for further research.

5 CONCLUSION

By re-presenting Syndrome-Trellis Codes without a trellis, we have demonstrated that the same algorithm can be used as a heuristic for minimizing non-additive distortion functions. An application was to optimize S-UNIWARD, which is normally only be minimized in its additive approximation, and where Gibbs Embedding did not succeed. The variable-cost STC does indeed reduce distortion, but we observed the same result as in [8]: detectability does not reduce. Although we do not cluster changes too tightly, as Gibbs Embedding tries to, a similar feature to [8] is that the non-additive distortion makes additional changes, presumably in the belief that they partially cancel each other out.

Although sub-optimal, the variable-cost STC is relatively simple to run, and we hope that it will spur research into non-additive distortion functions that can take advantage of it. The paucity of such functions limits the applicability of our method.

The same method can be used for distortion functions that evade a specific detector, in the case that Alice knows Eve's exact behaviour, something not necessarily possible even with Gibbs Embedding. Such a scheme cannot properly be called undetectable because it is highly detectable by anyone else! Most literature is concerned with the reverse situation, where the detector knows everything about the embedder save whether they are active or not. Both scenarios, and the equilibrium case where each knows the other's behaviour, deserve scrutiny, and benchmarks for steganalysis should not focus purely on one case.

ACKNOWLEDGMENT

The authors thank Patrick Bas and Christy Kin-Cleaves for fruitful discussions.

The work of Tomás Pevný was supported by OP VVV Research Center for Informatics no. CZ.02.1.01/0.0/0.0/16_019/0000765.

REFERENCES

[1] Patrick Bas, Tomáš Filler, and Tomáš Pevný. 2011. "Break Our Steganographic System": The Ins and Outs of Organizing BOSS. In *International Workshop on Information Hiding*, Vol. 6958, LNCS. Springer Berlin Heidelberg, 59–70.

[2] Rémi Cogranne, Vahid Sedighi, Jessica Fridrich, and Tomáš Pevný. 2015. Is ensemble classifier needed for steganalysis in high-dimensional feature spaces?. In *IEEE International Workshop on Information Forensics and Security (WIFS)*. 1–6.

[3] Pedro Comesana, Luis Pérez-Freire, and Fernando Pérez-González. 2006. Blind newton sensitivity attack. *IEE Proceedings-Information Security* 153, 3 (2006), 115–125.

[4] Tomáš Denemark and Jessica Fridrich. 2015. Improving Steganographic Security by Synchronizing the Selection Channel. In *Proceedings of the 3rd ACM Workshop on Information Hiding and Multimedia Security (IH&MMSec '15)*. ACM, 5–14.

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

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

[7] Jessica Fridrich and Jan Kodovsky. 2012. Rich models for steganalysis of digital images. *IEEE Transactions on Information Forensics and Security* 7, 3 (2012), 868–882.

[8] Vojtěch Holub et al. 2014. *Content Adaptive Steganography: Design and Detection*. Ph.D. Dissertation.

[9] Vojtěch Holub, Jessica Fridrich, and Tomáš Denemark. 2014. Universal distortion function for steganography in an arbitrary domain. *EURASIP Journal on Information Security* 2014, 1 (2014), 1.

[10] Sarra Kouider, Marc Chaumont, and William Puech. 2013. Adaptive steganography by oracle (ASO). In *IEEE International Conference on Multimedia and Expo (ICME)*. 1–6.

[11] Bin Li, Ming Wang, Jiwu Huang, and Xiaolong Li. 2014. A new cost function for spatial image steganography. In *IEEE International Conference on Image Processing (ICIP)*. 4206–4210.

[12] B. Li, M. Wang, X. Li, S. Tan, and J. Huang. 2015. A Strategy of Clustering Modification Directions in Spatial Image Steganography. *IEEE Transactions on Information Forensics and Security* 10, 9 (Sept 2015), 1905–1917.

[13] Tomáš Pevný, Patrick Bas, and Jessica Fridrich. 2010. Steganalysis by subtractive pixel adjacency matrix. *IEEE Transactions on information Forensics and Security* 5, 2 (2010), 215–224.

[14] Tomáš Pevný, Tomáš Filler, and Patrick Bas. 2010. Using high-dimensional image models to perform highly undetectable steganography. In *International Workshop on Information Hiding*, Vol. 6387, LNCS. Springer Berlin Heidelberg, 161–177.

[15] Niels Provos. 2001. Defending Against Statistical Steganalysis.. In *Usenix security symposium*, Vol. 10. 323–336.

[16] Yinlong Qian, Jing Dong, Wei Wang, and Tieniu Tan. 2015. Deep learning for steganalysis via convolutional neural networks. In *Media Watermarking, Security, and Forensics 2015*, Vol. 9409. 94090J.

[17] Phil Sallee. 2005. Model-based methods for steganography and steganalysis. *International Journal of Image and graphics* 5, 01 (2005), 167–189.

[18] Vahid Sedighi, Rémi Cogranne, and Jessica Fridrich. 2016. Content-Adaptive Steganography by Minimizing Statistical Detectability. *IEEE Transactions on Information Forensics and Security* 11, 2 (2016), 221–234.

[19] Vahid Sedighi, Jessica Fridrich, and Rémi Cogranne. 2015. Content-adaptive pentary steganography using the multivariate generalized Gaussian cover model. In *Media Watermarking, Security, and Forensics 2015*, Vol. 9409. 94090H.

[20] Jessica Fridrich Tomáš Filler. 2011. Design of adaptive steganographic schemes for digital images. (2011), 7880 - 7880 - 14 pages.

[21] Wenbo Zhou, Weiming Zhang, and Nenghai Yu. 2017. A New Rule for Cost Reassignment in Adaptive Steganography. *IEEE Transactions on Information Forensics and Security* 12, 11 (2017), 2654–2667.