https://doi.org/10.2352/ISSN.2470-1173.2019.5.MWSF-540

Reducing coding loss with irregular syndrome trellis codes

Christy Kin-Cleaves, Andrew D. Ker,
Department of Computer Science, Oxford University, Oxford, United Kingdom.

Abstract

We propose an extension to the Syndrome Trellis Code (STC) algorithm, aiming to reduce distortion by realizing embedding change probabilities closer to the optimal than the existing framework. A proxy for detectability, the minimization of distortion plays a critical role in producing good stego objects. STCs have become the tool of choice for many steganographers, because they approach the theoretical bound for embedding performance in quasi-linear time, for arbitrary length covers and payloads. However, until recently little attention has been paid to how closely STCs realize optimal change probabilities, particularly near the start and end of the embedding path. Recent work by Köhler et al, aimed to modify the parity-check matrix used by STCs to produce change vectors with change probabilities closer to the optimal probabilities. However, there is a cost of reduced capacity, or increased distortion. This paper demonstrates a modification to the block-structured STC parity-check matrix that both achieves changes closer to the optimal probabilities, and can be used for arbitrary length covers, and payloads.

Introduction

Adaptive steganography has widely adopted additive distortion minimisation as the state of the art. In order to generate as secure as possible stego objects, the steganographer has two aims. The first is to preserve the higher-order statistical properties of the cover through a distortion function which accurately models the cover source, the second is to maximise the embedding rate of the embedding function. By minimising the sum of additive distortion (as a proxy for detectability), we hope to maximise the security of our stego objects.

Improving the embedding efficiency using codes predates the idea of additive distortion functions, one of the earliest examples was Crandal using matrix embedding to improve the embedding efficiency of F5 [1]. Since then there have been several attempts to further improve the embedding efficiency, such as Golay Codes [2], and BCH codes [3]. However, the introduction of the STC algorithm brought a framework allowing steganographers to use any single letter distortion function, and achieve close to optimal embedding efficiency. As such, it is not surprising there has been a shift towards developing rich distortion functions as opposed to improving the embedding efficiency further.

We consider that not all changes are equal [4], meaning some changes may involve a higher "cost", as it is assumed making these changes will introduce a greater amount of distortion to the resulting stego object. We can split the embedding problem into two parts; finding the probabilities of change for each element of our cover and then finding an algorithm which makes changes with the probabilities previously found. Filler defines two forms of deriving said probabilities whilst minimising distortion; Payload-limited sender (PLS), and Distortion-limited sender (DLS) [5]. In this paper, we only consider the PLS case.

PLS embeds a fixed message $\boldsymbol{m}$ of length m bits into an arbitrary cover $\boldsymbol{x}$ of length n bits, whilst aiming to minimise the distortion $D(\boldsymbol{x},\boldsymbol{y})$. Essentially, Alice wishes to communicate her existing message, by embedding it into a cover, with the minimum distortion:

$$\text{Minimise} \sum_{i=0}^{n} p(x_i)c_i, \quad m \leq \sum_i H_2\Big(p(x_i)\Big), \tag{1}$$

where c_i is the cost of changing the cover value x_i. We can solve (1) by finding suitable values of $p(x_i)$, the probability that the i^{th} coefficient of the input cover is modified, defined as:

$$p(x_i) = \frac{e^{-\lambda c_i}}{1+e^{-\lambda c_i}}, \tag{2}$$

where λ is a non-negative scalar, and $H(x)$ is the binary entropy function:

$$H_2(x) = -x\log_2 x - (1-x)\log_2(1-x). \tag{3}$$

λ can be found by conducting a binary search. As such, $\boldsymbol{p}$ is the vector of optimal change probabilities for a given embedding. We can calculate the optimal distortion as:

$$D'(\boldsymbol{x},\boldsymbol{y}) = \sum_{i=0}^{n} p_i c_i. \tag{4}$$

We quantify the gap between achieved distortion and the optimal distortion as coding loss [6]. The aim of any steganographic coding algorithm, therefore, is to minimise the coding loss. Further, if an algorithm makes a change with higher probability than those dictated by the derived optimal change probabilities, it must make some other changes with a lower probability than the optimum, in order to generate enough entropy to convey the message. We call these changes with significant divergence from the optimal change probabilities outliers.

In this paper, we investigate modifications to the STC algorithm to reduce the number of outliers and thus achieve a lower coding loss. We propose a shift from using block sub-matrices for generating the parity check matrix used during STC encoding, instead we promote using segment-vectors which are tiled to produce the parity check matrix. The tiling of these segments affects the number of outliers and hence the distortion. We show a practical method to generate these matrices and highlight the optimal parameters for embedding. We benchmark our method against the plain STC algorithm, and the *Outlier-Corrected STC* as proposed by Köhler et al. [7].

Notation

We shall denote vectors as lowercase boldface $\boldsymbol{x}$, with the i^{th} element of $\boldsymbol{x}$ denoted x_i. Matrices are written in uppercase double struck: $\mathbb{H}$, indexed as $\mathbb{H}_{ij}$. We will only consider binary payloads and codes, and all vector arithmetic will be mod 2.

Prior art

The STC algorithm as proposed by Filler [8, 5] allows the embedder to minimize an additive distortion heuristic while solving $\boldsymbol{m} = \mathbb{H}\boldsymbol{y}$, where $\boldsymbol{y}$ represents the binary or ternary remainder of the pixels or DCT coefficients in the stego object. Since their introduction, they have become heavily adopted in the adaptive-steganography community, because of their minimal coding loss, linear time complexity, and because their generalised framework allows any additive distortion function to be used. STCs are elegant in that the embedding performance and time/space complexity of the algorithm is parameterized by the constraint height h of the matrix $\hat{\mathbb{H}}$. Whilst becoming widely popular in the research community, there has been a smaller focus on the algorithm itself, and how it could be improved. Most notably, the impact the construction of the parity check matrix $\mathbb{H}$ has on embedding performance.

Liu [9] highlight that since the original proposal of the STC algorithm, there has been less work focused on improving the embedding efficiency of the algorithm, and the relationship between STCs and convolutional codes is a poorly known area. However, Liu's work focuses on reducing the space and time complexity of the STC algorithm by using a minimum span generator, whilst achieving the same embedding efficiency. To achieve this a minimum trellis is constructed, by converting a regular parity check matrix to 'minimal-span' form using Gaussian elimination. The resulting minimal span generator matrix (MSGM) has an LR property, in that no two rows in the matrix can have the same minimum and maximum span. Because of the structure of the MSGM a fast solution to $\mathbb{H}\boldsymbol{y} = \boldsymbol{m}$ can be found using a minimal trellis, reducing the time and space complexity to find optimal solutions.

Köhler [7] investigates how STCs perform with respect to optimal embedding probabilities. Köhler proposes a variation of the STC algorithm: outlier corrected Syndrome Trellis Codes (OC-STC). By cropping the first $h-1$ rows of $\mathbb{H}$, the matrix $\mathbb{H}$ becomes symmetrical, as the first and last $h-1$ rows contain subsets of the matrix $\hat{\mathbb{H}}$. This reduces the payload by $h-1$, as this is necessary to achieve the cropping. Experimentally, it was found that OC-STCs mitigated the positive outliers (values changed more frequently than optimal). Given negative outliers do not pose an intermediate security risk, the reduced efficiency of values (due to negative outliers still existing) is a sensible trade-off.

There have been several steganographic implementations utilising random-linear codes. Fridrich has proposed several schemes including Simplex Codes [10], and Wet Paper Codes [11, 12]. Both methods have since been rendered obsolete by the more powerful STC algorithm, yet both are based on random-linear codes under the premise that random-linear codes can approach the theoretic bound for embedding efficiency [13, p325, Theorem 12.3.5].

Methods of analysis

In this section, we introduce the reader to several methods of analysis used to quantify the performance attribution of our work.

Divergence from optimal change probabilities

To analyse the single-cover divergence from optimal change probabilities as described in (2), we can use the Hellinger distance. Let $\boldsymbol{p}$, and $\hat{\boldsymbol{p}}$ be the vectors of optimal and observed change probabilities respectively. Then p_i, and $\hat{p}_i$ are the i^{th} value of the optimal, and observed change probabilities from some stego object $\boldsymbol{y}$ of length n. For each value of the stego object, we calculate the Hellinger distance as

$$H(p_i,\hat{p}_i) = \sqrt{\sqrt{1-p_i}\sqrt{1-\hat{p}_i} + \sqrt{p_i}\sqrt{\hat{p}_i}}, \tag{5}$$

the per-element Hellinger distance is the divergence from the optimal change probability. If an embedding implementation generates positive outliers ($\hat{p}_i > p_i$), then there is both a chance of the outliers being used to identify stego objects, and the embedding algorithm cannot be achieving the optimal distortion.

To analyse outliers across the entire cover, we use the Kullback–Leibler divergence. Since $\boldsymbol{p}$, and $\hat{\boldsymbol{p}}$ are Bernoulli distributions, the KL-divergence is:

$$KL(\boldsymbol{p},\hat{\boldsymbol{p}}) = \sum_{i=0}^{n} p_i \log\left(\frac{p_i}{\hat{p}_i}\right) + (1-p_i)\log\left(\frac{1-p_i}{1-\hat{p}_i}\right). \tag{6}$$

Therefore, the embedding algorithm with the lowest KL-divergence is one which achieves less divergence from the optimal change probabilities, and a distortion closer to the optimal distortion.

Binary entropy of changes

Using (3) we can calculate the binary entropy of the observed embedding changes as:

$$\hat{H}_2(\hat{\boldsymbol{p}}) = \sum_{i=0}^{n} H_2(\hat{p}_i). \tag{7}$$

Since the maximum payload that any embedding algorithm is the entropy of the changes, using the entropy as a theoretical bound allows us to quantify the coding loss of the embedding operation [4, 14]. Thus, the embedding algorithm which generates the lowest binary entropy per the same message embedded has a lower coding loss.

Average distortion, variance, and skew

As defined earlier, single letter distortion has been widely adopted as the state of the art for adaptive steganography. It is clearly understood that there are limitations in adaptive steganography, mainly that interacting changes cannot be accounted for with single-letter distortion values [15]. Yet distortion minimisation is well considered a proxy for detectability [8]. We define the distortion of a stego object $\boldsymbol{y}$ as:

$$D(\boldsymbol{x},\boldsymbol{y}) = \sum_{i=0}^{n} c_i \quad [x_i \neq y_i] \quad | \quad c_i \in \mathbb{R}. \tag{8}$$

Given its additive nature, we can compare embedding algorithms by the average distortion for the same set of covers and messages when using the same distortion value.

For a cover set $\boldsymbol{X}$ of size k, $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_k\}$, we encode the messages $\{\boldsymbol{m}_1, \boldsymbol{m}_2, \ldots, \boldsymbol{m}_l\}$ of length m bits, resulting in the distortion result set $\boldsymbol{Z}$ of length $v = kl$:

$$\boldsymbol{Z} = \{D(\boldsymbol{x}_i, Emb(\boldsymbol{x}_i, \boldsymbol{m}_j)) \mid i = 1\ldots k, j = 1\ldots l\} \tag{9}$$

Over a large set of embedding, we can consider the variance of the distortion to give further insight into the distribution of distortion values per embedding algorithm. For this we use sample variance:

$$\sigma^2 = \sum_{i=0}^{v} \frac{1}{n-1}(z_i - \bar{z})^2, \tag{10}$$

where z_i is the individual distortion value, and $\bar{z}$ is the average distortion across $\boldsymbol{Z}$ for a particular embedding algorithm. We can then calculate the standard error of the mean, *SE*, as:

$$SE = \sqrt{\frac{\sigma^2}{v}}. \tag{11}$$

Z-test

Since we generate each individual distortion valuer per-embedding, we can use a Z-test to test the statistical difference between two embedding algorithms. The Z-test determines if two distributions are significantly different (the null hypothesis), given their mean and variance. The Z-test is constructed as follows:

$$z = \frac{\overline{\boldsymbol{z}_a} - \overline{\boldsymbol{z}_b}}{\sqrt{\frac{\sigma_a^2}{v} + \frac{\sigma_b^2}{v}}}, \tag{12}$$

for two distortion distributions $\boldsymbol{z}_a$, and $\boldsymbol{z}_b$, with the respective means and variances $\overline{\boldsymbol{z}_a}$, $\overline{\boldsymbol{z}_b}$, and σ_a^2, σ_b^2. With a resulting value greater than 1.96, or lower than -1.96, the null hypothesis is rejected.

Irregular STCs

From Köhler's work [7], it became clear that the improvement over the regular STC, was due to each row in $\mathbb{H}$ containing exactly the exact same values. To tile the $\mathbb{H}$ matrix in such a way, required $h-1$ rows to be truncated, reducing the payload by $h-1$ bits. However, the requirement to truncate $h-1$ rows only comes from using a $\hat{\mathbb{H}}$ matrix to tile the $\mathbb{H}$ matrix, as proposed in [8, 5]. If we look at building $\mathbb{H}$ matrices with the sole requirement of containing the same row, or at least the same number of ones, then we relax the constraints on how we can generate said matrices. This allows us to generate $n \times m$ matrices in a much simpler manner.

In Köhler's proposed Outlier-Corrected STC, each row contains exactly the same sequence, of length $h\frac{1}{\alpha}$, for a constraint height h. However, when tiling the matrix using $\hat{\mathbb{H}}$ matrices, we have a symmetrical tiling, each sub-matrix is of width $\frac{1}{\alpha}$, which requires $h-1$ rows to be truncated. To generate a $n \times m$ matrix with exactly the same segment (sequence of non-zero values) we first fix the length of the segment as $s = h\frac{1}{\alpha}$. Since we have m rows, if we tile each segment with a gap of $\frac{1}{\alpha}$, the length of the matrix would be $\alpha(m-1)+s$, which is larger than n. However, as we no longer have a constraint of how we tile the matrix, we can reduce the gaps between segments in any manner, as long as the tiling produces a matrix $\mathbb{H}$ of dimensions $n \times m$. To describe this, let $\boldsymbol{g} = \{\boldsymbol{g}_1, \boldsymbol{g}_2, \ldots, \boldsymbol{g}_{m-1}\}$ be the set of gaps between each row as shown in Fig 1, the *gap profile*. We now have the simple constraint of:

$$n = s + \sum_{i=0}^{m-1} g_i, \tag{13}$$

which allows any $n \times m$ sized $\mathbb{H}$ matrix to be constructed, with the same segment, or at least, the same number of ones.

We can modify the STC algorithm to handle the irregular $\mathbb{H}$ matrix in two steps, first, we decompose the matrix $\mathbb{H}$ into irregular blocks at each message bit(s) pruning. Then for each irregular block in the matrix, we generate the sub-trellis of possible syndromes, note how each block may be of different dimensions as shown in Fig 2.

There now exists more questions, about how exactly we should generate these matrices. In this section, we aim to explore the possible questions around the matrix generation, and how they impact performance. For all experiments in this paper, we will use costs generated with the WOW distortion function [16], following the experimental setup of [7]. Our experiments to find optimal parameters will use 1,000 covers from the BOSSbase set [17], cropped at random to $8 \times 8, n = 64$ cover crops. Using a message set of 1,000 messages of length $m = 32$.

Segment based matrices

With the move to segment-based matrices, our first experiment is to determine the best method to construct the rows of the matrix. From earlier we discussed how random-linear codes can approach the theoretic bound for embedding efficiency, however, finding the lowest distortion syndrome for a randomly generated matrix would be an NP-hard problem. We can, however, use randomly generated segments to construct the matrix. To test random segments versus repeated segments, we generated 100 $\mathbb{H}$ matrices for randomly generated segments and repeated segments. All segments had the constraint of the first and last $\frac{1}{\alpha}$ values being set to one, to ensure a large branching factor during trellis construction.

From Fig 3, we see that repeated segments slightly outperform randomly generated segments. To quantify if the difference is significant, we use the Z-test as in (12):

$$\frac{72,042.95 - 71,755.34}{\sqrt{\frac{1.225\times10^{12}}{2\times10^{8}} + \frac{1.197\times10^{12}}{2\times10^{8}}}} \approx 2.614, \tag{14}$$

since the result of the z-statistic lies outside of the 95% confidence interval (± 1.96), it is clear that repeated segments significantly outperform random segments.

Tiling the matrix

As each row in the matrix $\mathbb{H}$ contains exactly one segment of length s, there exists a tiling problem of how best to construct the matrix. Since we have m segments of length s, and we wish to construct a banded matrix to allow partial syndromes to be found in linear time, we wish to find some *gap profile* $\boldsymbol{g}$ (the placement of gaps between segments) to produce a perfect tiling which also performs best in terms of distortion minimisation.

If we consider the problem as highlighted in 13, we can come up with four basic cases:

(a) **Smaller gaps at the beginning.** Create the gap profile $\boldsymbol{g}$, with smaller values at the beginning, creating a denser matrix at the start of the trellis.

(b) **Smaller gaps at the beginning and end.** Create the gap profile $\boldsymbol{g}$, with smaller values at the start and end, creating a denser matrix at the start and end of the trellis.

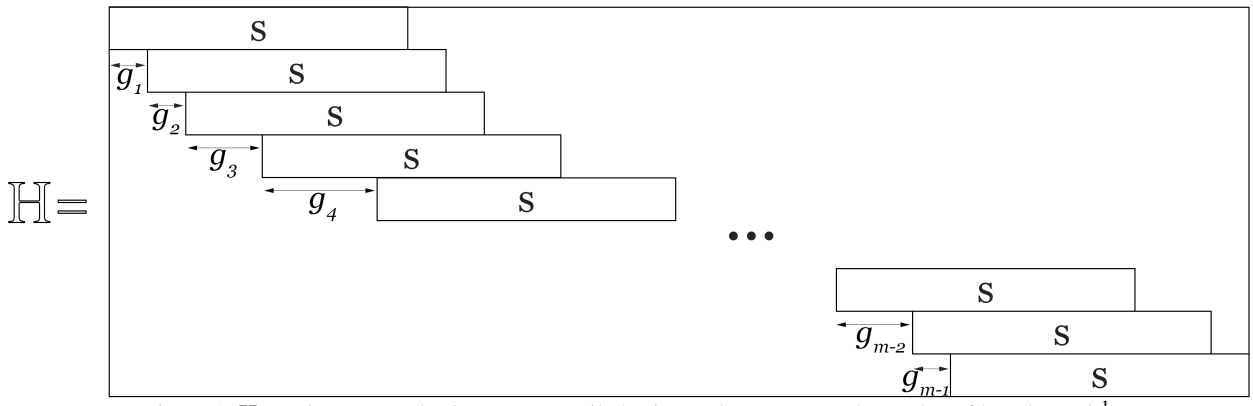


Figure 1: $\mathbb{H}$ matrix generated using segments tiled using $\boldsymbol{g}$. Segments are denoted $\boldsymbol{s}$, of length $s = h\frac{1}{\alpha}$.

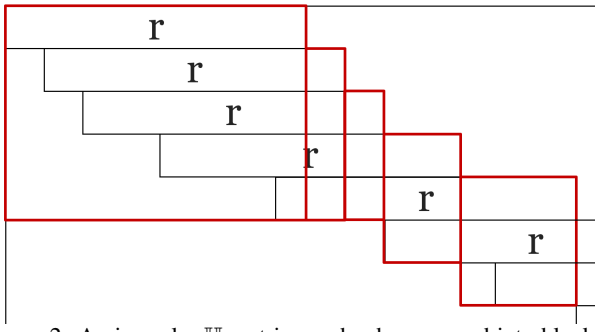


Figure 2: An irregular $\mathbb{H}$ matrix can be decomposed into blocks (highlighted in red). Note how blocks can be different widths and heights. Empty spaces are filled with zeros.

Method	$D(\boldsymbol{x},\boldsymbol{y})$	$\sigma^2\big(D(\boldsymbol{x},\boldsymbol{y})\big)$	SE
Random segments	72,043	1.225×10^{12}	109.40
Repeated segments	71,755	1.197×10^{12}	109.40

Figure 3: Average distortion, distortion variance, and SE for random and repeated segments in $\mathbb{H}$.

(c) **Smaller gaps in the middle** Create the gap profile $\boldsymbol{g}$, with smaller values in the middle, creating a denser matrix at the middle of the trellis.

(d) **Smaller gaps at the end.** Create the gap profile $\boldsymbol{g}$, with smaller values at the end, creating a denser matrix at the end of the trellis.

The four cases can be seen in 4. To test the four cases, we use the same experimental setup as in the previous experiment, however, this time we generate five new $\mathbb{H}$ matrices for each of the four tiling options, using repeated segments given the findings of the last experiment.

From fig 5 we can see that compression at the start and end of the matrix gives the best performance by quite a margin. The z-statistic between the two best methods (smaller gaps at the beginning and end, and smaller gaps at the end) is 26.609, indicating a significant performance gain for the former method. We attribute this to the increased density at the start and end of the trellis. With the additional states, the STC algorithm is able to find more solutions, avoiding positive outlier change probabilities. Reducing these outliers allows a lower distortion since values are not changed with higher observed probability.

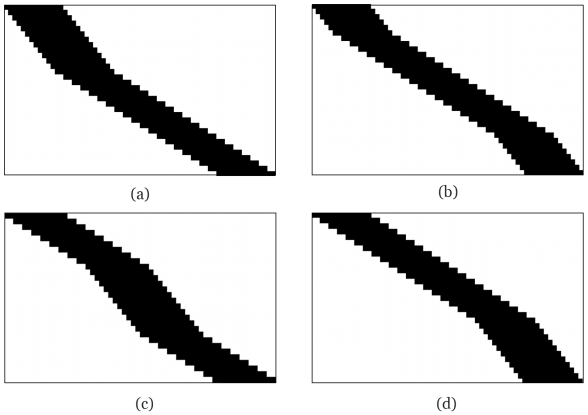


Figure 4: The effects of smaller gaps at different parts of the $\mathbb{H}$ matrix; (a) at the start, (b) at the start and end, (c) in the middle, (d) at the end. Dark regions are the non-zero sections of the $\mathbb{H}$ matrix.

Method	$D(\boldsymbol{x},\boldsymbol{y})$	$\sigma^2\big(D(\boldsymbol{x},\boldsymbol{y})\big)$	SE
Smaller gaps at the beginning	74,446	6.932×10^{10}	110.86
...at the beginning and end	72,182	6.145×10^{10}	117.74
...in the middle	76,661	7.943×10^{10}	116.73
...at the end	74,324	6.813×10^{10}	126.04

Figure 5: Average distortion, distortion variance, and SE for methods of tiling $\mathbb{H}$.

η - *density of the segment*

The density of ones in the segments $\boldsymbol{s}$ will greatly affect the performance. When constructing the trellis, a new state address is given by the XOR product of the old state label, and the column in $\mathbb{H}$. If the column in $\mathbb{H}$ is entirely zeros or ones, the likelihood that similar or identical columns exist. In the scenario of two columns being identical, a cycle between states occurs, which is wasteful. If x_i and x_j are two cover values for the i^{th}, and j^{th} (identical) columns of $\mathbb{H}$ respectively, then the assignment $x_i = 0, x_j = 0$ and $x_i = 1, x_j = 1$ will create two paths from each state at x_i to exactly one state in x_j. As one path will have lower distortion, we have halved the number of useful paths through the trellis between x_i, and x_j.

In order to maximise the number of distinct paths through the trellis, we want to generate columns in $\mathbb{H}$ that are as different as possible. As mentioned earlier, as the rate of zeros or ones ap-

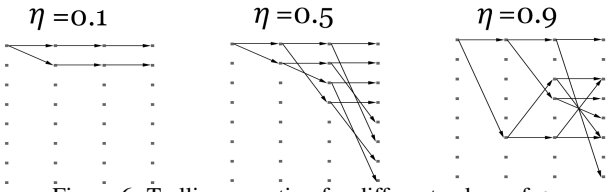


Figure 6: Trellis generation for different values of η.

η	$D(\boldsymbol{x},\boldsymbol{y})$	$\sigma^2\big(D(\boldsymbol{x},\boldsymbol{y})\big)$	SE
0.2	93,561	1.150×10^{11}	151.65
0.3	82,484	9.438×10^{10}	137.39
0.4	74,456	6.670×10^{10}	115.47
0.5	73,280	6.189×10^{10}	102.47
0.6	72,476	6.189×10^{10}	111.25
0.7	72,459	6.283×10^{10}	112.09
0.8	73,076	6.394×10^{10}	113.09

Figure 7: Average distortion, and distortion variance for methods of different values of η.

proaches 1.0, the probability of duplicate columns will increase. First, let η be the rate of ones in each segment $\boldsymbol{s}$. Specifically the number of ones in each segment $\boldsymbol{s}$ is $\lfloor \eta s \rceil$.

To test this, we again use the same experimental setup as before. We generate five $\mathbb{H}$ matrices for each density $\eta \in \{0.2, 0.3, \ldots, 0.8\}$.

In Fig. 7 we show the distortion values for the different values of η. As expected, the extreme values of η, when the segments approach entirely being constituted of zeros or ones, perform significantly worse. At first, we may expect the value $\eta = 0.5$ to perform best, given the probability of distinct segments is the greatest. However, the more zeros in each column reduce the number of paths through the trellis, and therefore the number of solutions. We attribute this explanation to why the slightly higher $\eta = 0.7$ value performs best. The z-statistic between eta values $\eta \in \{0.6, 0.7\}$ is 0.280 indicating there is not a large significance between the two values.

γ - *the skew of the matrix*

Up to now, we have shown that when we move from tiling $\hat{\mathbb{H}}$ matrices to generate $\mathbb{H}$ we remove certain constraints. and allow a more flexible way to generate the matrices. Specifically, we reduce the tiling constraint to as previously introduced in 13. In previous sections, we have shown that a gap profile which reduces the space at the start and end of the matrix performs best. We now introduce a method to produce the tiling and parameterize the skew of the matrix. By skew, we refer to how aggressively the density is increased at the start, and end of the matrix. For comparison, a regular STC $\mathbb{H}$ matrix would have no skew. Using $\gamma \in [0, 1]$ we can generate the tiling of the matrix using the algorithm 1.

To test values of gamma, we use the same experimental setup as before, generating four $\mathbb{H}$ matrices for the gamma values $\gamma \in \{0.1, 0.3, 0.5, 0.7\}$.

Clearly lower values of gamma are better for distortion minimisation. We attribute this to the lower effective constraint height in the middle of the trellis, at higher gamma values the contrast between the height of the trellis in the middle, versus the start and ends is greater. The z-statistic between the gamma values $\gamma \in \{0.1, 0.5\}$ is 22.07, indicating a significant performance gain

Algorithm 1 γ tiling

procedure TILING(n, m, γ, r)
  $m' \leftarrow m$
  $n' \leftarrow n - r$
  while $i < m$ **do**
    $\boldsymbol{g}_i = round\big((1-\gamma)\frac{m'-2}{m-3}\frac{n'}{m'-1}\big)$
    $n' = n' - \boldsymbol{g}_i$
    $m' = m' - 1$
  return $\boldsymbol{g}$

γ	$D(\boldsymbol{x},\boldsymbol{y})$	$\sigma^2\big(D(\boldsymbol{x},\boldsymbol{y})\big)$	SE
0.1	72,139	6.206×10^{10}	111.41
0.3	72,139	6.206×10^{10}	111.41
0.5	73,468	6.484×10^{10}	113.88
0.7	78,454	6.083×10^{10}	123.69

Figure 8: Average distortion, and distortion variance for methods of different values of η.

for lower values of gamma.

Because of the rounding of the gaps during the tiling, it is possible for two gamma values to produce the same tiling. This is evident as the values $\gamma \in \{0.1, 0.3\}$ produced the same tiling in our experiment.

Performance

To compare our method against the state of the art, we measured achieved distortion, and mitigation of outliers. We compared our proposed method to the STC and Outlier-Corrected STC algorithms using the best parameters found in previous sections. Using the $8 \times 8, n = 64$ cover crops from previous experiments, we first benchmarked our method minimisation against the others. Tabular results in show that our method does indeed have a lower distortion.

To better simulate practical applications we increased the cover sizes. We took a cover set of 1,000 BOSSBase images, cropped at random to 64×64, $n = 4096$ and embedded 1,000 randomly generated messages with $m = 2048, \alpha \in \{\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}\}$. For comparison, we used a STC, and OC-STC with constraint heights $h \in \{7, 8, 9\}$, which for $\alpha = \frac{1}{2}$ gives segments of length $s \in \{14, 16, 18\}$, and for other α gives longer segments. We see that our method does reduce distortion, which we attribute to the mitigation of positive outliers, which in turn, achievers change probabilities closer to the optimal change probabilities. In Fig. 10, we show how our method achieves a slightly lower distortion when compared to the other methods. Our method achieves a slightly lower distortion. Of course, the distortion cannot be reduced below the *optimal* distortion given by (4), so a metric for comparison is the *coding loss*, which we define as the ratio of the average optimal distortion divided by the average achieved distortion. The IR-STC method reduces coding loss by a tiny amount in the case of $\alpha = \frac{1}{2}$, but as much as 1.35% in the case of smaller payloads ($h = 7, \alpha = \frac{1}{8}$). Our method also has a lower distortion variance (itself an indicator that outliers have been reduced).

To better quantify outlier mitigation, we took a randomly chosen cover from the BOSSbase, and cropped it to 64×64, $n = 4096$, using all three methods, we encoded 10,000 randomly generated messages. We then plot observed change probabilities against optimal change probabilities, for constraint heights

Method	$D(\boldsymbol{x},\boldsymbol{y})$	$\sigma^2\big(D(\boldsymbol{x},\boldsymbol{y})\big)$	SE
IR-STC	67,187	4.791×10^9	69.21
OC-STC	69,756	5.123×10^9	71.58
STC	70,253	5.315×10^9	72.90

Figure 9: Average distortion, distortion variance, and SE for IR-STC, OC-STC, and STCs, with $8\times8, n=64$ cover crops.

$h\in\{7,9\}$, in Fig. 11. We also display the average KL divergence between observed and optimal probabilities, the achieved distortion, and the achieved change entropy. Note that IR-STC reduces the average KL divergence by around 10%, showing that outliers have been mitigated.

On the other hand, we have an increased number of states at the start and the end of our trellis, which is determines the algorithm's time complexity. This is caused by the smaller gaps between segments, effectively making the trellis denser at those points. Critically, this increases the constraint height, however, this is only for a very small section at the start and end of embedding, because we decomposed the matrix $\mathbb{H}$ into variable-sized blocks. In fact, over the entire cover, we see a very small increase in the number of states. The encoding process is still (pseudo-)linear time. The small increase in the number of states is seen in Fig. 10.

Conclusion

STCs have become incredibly popular over the last few years. We have explored a new way of constructing the matrix $\mathbb{H}$ in order to achieve closer-to-optimal change probabilities, whilst achieving a lower distortion, without changing the Viterbi algorithm that underlies STCs. The change is to allow the trellis to have variable height, with more height at the start and end. We also stress the value of constructing $\mathbb{H}$ from row segments, rather than blocks. There has been little research into what makes a good good parity matrix for STC performance. Whilst our improvement is small, the relative cost in terms of complexity is minimal.

We were only able to benchmark images up to size 64×64. Bear in mind the requirement to embed thousands of messages per cover, and thousands of covers. In future work, we hope to perform similar tests on full-size images. Similarly, we have not measured the *detectability* of our method. We have focussed only on distortion, since the aim of STCs is to minimize distortion, but the ultimate test of an embedding method is its ability to evade detection.

Other ways in which $\mathbb{H}$ is constructed may also influence the performance of STCs. Note that convolutional codes, when used in error correction, are highly sensitive to the parity-check matrix (and good matrices are often found by exhaustion). Perhaps a steganographic analogy exists.

Acknowledgements

This work was supported by the Engineering and Physical Sciences Research Council [1744549].

References

[1] R. Crandall. Some Notes on Steganography. Steganography Mailing List, available from http://os.inf.tu-dresden.de/ westfeld/crandall.pdf, 1998.

[2] M. Van Dijk and F. Willems. Embedding Information in Grayscale Images. In *Proceedings of the 22nd Symposium on Information and Communication Theory in the Benelux, Enschede, The Netherlands*, pages 147–154. Citeseer, 2001.

[3] D. Schönfeld and A. Winkler. Embedding with Syndrome Coding Based on BCH Codes. In *Proceedings of the 8th Workshop on Multimedia and Security*, MM&Sec'06, pages 214–223, New York, NY, USA, 2006. ACM.

[4] J. Fridrich. *Steganography in Digital Media.* Cambridge University Press, 2009.

[5] T. Filler, J. Judas, and J. Fridrich. Minimizing Additive Distortion in Steganography Using Syndrome-Trellis Codes. *IEEE Transactions on Information Forensics and Security*, 6(3):920–935, 2011.

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

[7] O. Köhler, C. Pasquini, and R. Böhme. On the Statistical Properties of Syndrome Trellis Coding. In *International Workshop on Digital Watermarking*, pages 331–346. Springer, 2017.

[8] T. Filler, J. Judas, and J. Fridrich. Minimizing Embedding Impact in Steganography Using Trellis-Coded Quantization. In *Media Forensics and Security*, page 754105, 2010.

[9] W. Liu, G. Liu, and Y. Dai. Syndrome Trellis Codes Based on Minimal Span Generator Matrix. *Annals of telecommunications-annales des télécommunications*, 69(7-8):403–416, 2014.

[10] J. Fridrich and D. Soukal. Matrix Embedding for Large Payloads. *IEEE Transactions on Information Forensics and Security*, 1(3):390–395, 2006.

[11] J. Fridrich, M. Goljan, and D. Soukal. Wet Paper Codes With Improved Embedding Efficiency. In *Security, Steganography, and Watermarking of Multimedia Contents VIII*, volume 6072, page 607215. International Society for Optics and Photonics, 2006.

[12] J. Fridrich, M. Goljan, and D. Soukal. Steganography via Codes for Memory with Defective Cells. In *43rd Conference on Coding, Communication, and Control*, pages 28–30, 2005.

[13] G. Cohen, I. Honkala, S. Litsyn, and A. Lobstein. *Covering Codes*. Elsevier, 1997.

[14] Joachim J. Eggers, R. Buml, and B. Girod. A Communications Approach to Image Steganography. In *Proceedings of SPIE: Electronic Imaging 2002, Security and Watermarking of Multimedia Contents IV*, pages 26–37, 2002.

[15] J. Fridrich and T. Filler. Practical Methods for Minimizing Embedding Impact in Steganography. In *Proceedings of SPIE: Electronic Imaging 2007, Security and Watermarking of Multimedia Contents IX*, volume 6505, pages 6505 – 6505 – 15, 2007.

[16] V. Holub and J. Fridrich. Designing Steganographic Distortion using Directional Filters. In *2012 IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 234–239, Dec 2012.

[17] P. Bas, T. Filler, and T. Pevny. 'Break Our Steganographic System': The Ins and Outs of Organizing BOSS. *Information Hiding*, pages 59–70, 2011.

Author Biography

Christy Kin-Cleaves is a current DPhil (PhD) student at the University of Oxford, under the supervision of Dr Andrew D. Ker, with a focus on adaptive steganography. He received his BSc and MSc from Durham University in 2013, and 2015 respectively.

Andrew Ker is an Associate Professor at Oxford University Department of Computer Science, where he received a BA in 1997 and DPhil (PhD) in 2000. His work is on the theory and practice of steganography. He is a Senior Member of IEEE and a member of ACM.

Payload	Constraint height	Method	$D(\boldsymbol{x},\boldsymbol{y})$	SE	Coding loss	$\gamma_1(D(\boldsymbol{x},\boldsymbol{y}))$	States reached
$\alpha=\frac{1}{2}$	$h=7$	IR-STC	3,006,231	2,859.02	0.1252	8.174×10^{12}	532,476
		OC-STC	3,008,140	2,851.32	0.1259	8.183×10^{12}	522,362
		STC	3,008,176	2,860.42	0.1259	8.182×10^{12}	521,846
	$h=8$	IR-STC	2,954,325	2,812.47	0.1057	7.910×10^{12}	1,064,694
		OC-STC	2,956,160	2,813.89	0.1064	7.918×10^{12}	1,043,962
		STC	2,956,375	2,814.25	0.1065	7.920×10^{12}	1,042,678
	$h=9$	IR-STC	2,904,853	2,767.49	0.0872	7.659×10^{12}	2,189,664
		OC-STC	2,906,689	2,768.75	0.0879	7.666×10^{12}	2,093,566
		STC	2,906,893	2,769.12	0.0880	7.668×10^{12}	2,089,980
$\alpha=\frac{1}{4}$	$h=7$	IR-STC	1,084,931	1,101.82	0.1696	1.214×10^{12}	545,268
		OC-STC	1,085,784	1,102.72	0.1706	1.216×10^{12}	519,794
		STC	1,086,099	1,102.72	0.1709	1.216×10^{12}	520,726
	$h=8$	IR-STC	1,063,323	1,081.20	0.1463	1.169×10^{12}	1,114,100
		OC-STC	1,064,463	1,082.13	0.1476	1.171×10^{12}	1,037,810
		STC	1,064,639	1,082.13	0.1478	1.171×10^{12}	1,040,518
	$h=9$	IR-STC	1,044,247	1,061.04	0.1258	1.129×10^{12}	2,287,604
		OC-STC	1,045,362	1,063.48	0.1270	1.131×10^{12}	2,072,050
		STC	1,045,816	1,063.95	0.1275	1.132×10^{12}	2,078,086
$\alpha=\frac{1}{8}$	$h=7$	IR-STC	423,303	453.54	0.1893	2.057×10^{11}	570,815
		OC-STC	428,071	456.40	0.2027	2.083×10^{11}	519,794
		STC	428,121	456.29	0.2028	2.082×10^{11}	519,152
	$h=8$	IR-STC	417,767	445.98	0.1737	1.989×10^{11}	1,137,636
		OC-STC	418,618	446.77	0.1761	1.996×10^{11}	1,025,506
		STC	418,950	446.99	0.1770	1.998×10^{11}	1,032,846
	$h=9$	IR-STC	410,051	437.95	0.1520	1.918×10^{11}	2,353,124
		OC-STC	410,943	437.84	0.1545	1.917×10^{11}	2,043,362
		STC	411,229	438.75	0.1553	1.925×10^{11}	2,063,534
$\alpha=\frac{1}{16}$	$h=7$	IR-STC	176,477	186.71	0.1853	3.486×10^{10}	585,668
		OC-STC	176,584	187.67	0.1858	3.522×10^{10}	504,386
		STC	176,603	188.02	0.1859	3.535×10^{10}	512,542
	$h=8$	IR-STC	171,817	191.00	0.1632	3.648×10^{10}	1,217,476
		OC-STC	172,342	191.47	0.1658	3.666×10^{10}	1,000,898
		STC	172,497	191.55	0.1665	3.669×10^{10}	1,020,894
	$h=9$	IR-STC	168,047	196.21	0.1444	3.850×10^{10}	2,542,532
		OC-STC	168,931	196.01	0.1489	3.842×10^{10}	1,985,986
		STC	169,098	196.19	0.1498	3.849×10^{10}	2,033,886

Figure 10: Average distortion, coding loss, variance, and number of trellis states reached for IR-STC, OC-STC, and STC, with $64\times64, n=4096$ cover crops.

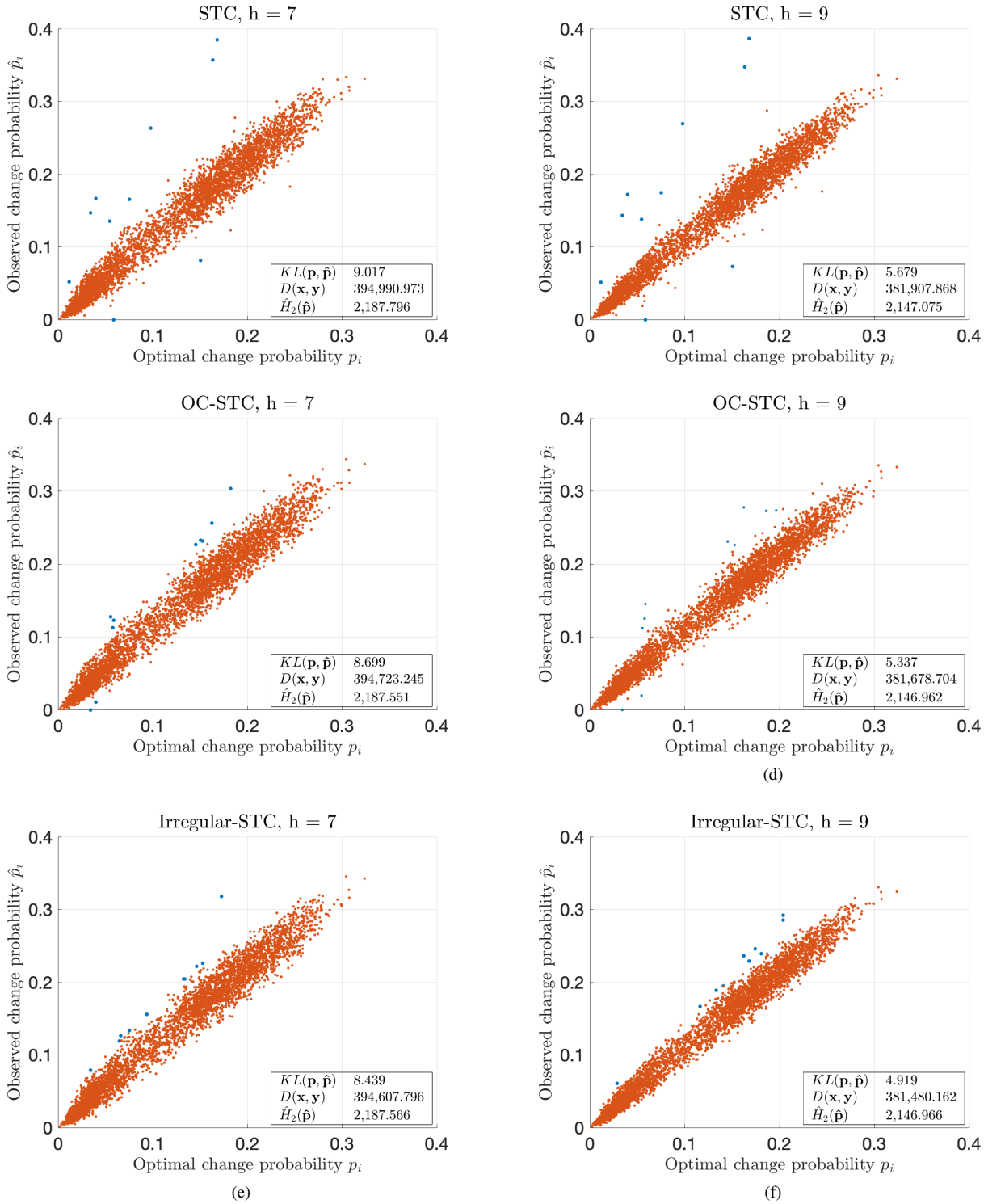


Figure 11: Optimal change probability versus observed change probability comparison between our proposed method, STC, and OC-STC. Blue markers the signify values with the largest Hellinger distance from optimal change probabilities.

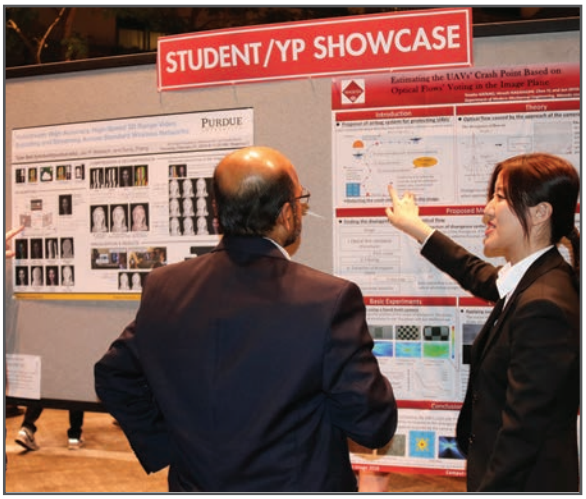
JOIN US AT THE NEXT EI!

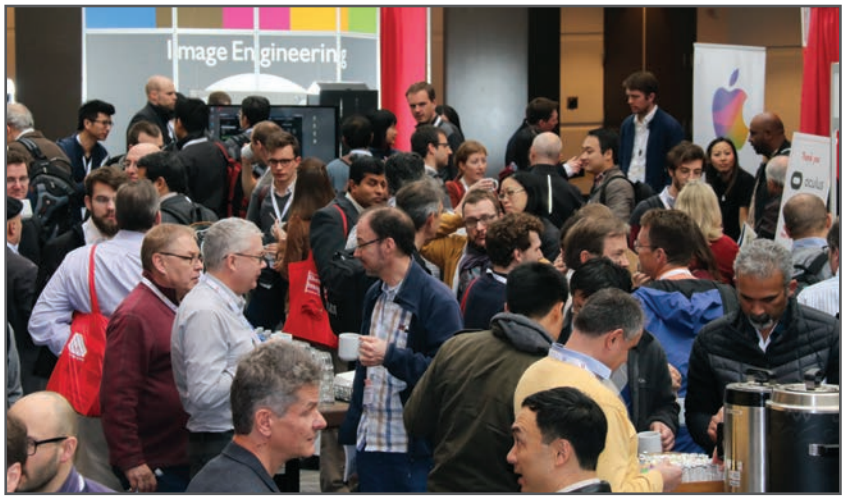
IS&T International Symposium on

Electronic Imaging

SCIENCE AND TECHNOLOGY

Imaging across applications . . . Where industry and academia meet!





• SHORT COURSES • EXHIBITS • DEMONSTRATION SESSION • PLENARY TALKS •
• INTERACTIVE PAPER SESSION • SPECIAL EVENTS • TECHNICAL SESSIONS •

www.electronicimaging.org

IS&T
imaging.org