# Estimating Steganographic Fisher Information in Real Images

Andrew D. Ker

Oxford University Computing Laboratory, Parks Road, Oxford OX1 3QD, England
adk@comlab.ox.ac.uk

**Abstract.** This paper is concerned with the estimation of steganographic capacity in digital images, using information theoretic bounds and very large-scale experiments to approximate the distributions of genuine covers. The complete distribution cannot be estimated, but with carefully-chosen algorithms and a large corpus we can make local approximations by considering groups of pixels. A simple estimator for the local quadratic term of Kullback-Leibler divergence (*Steganographic Fisher Information*) is presented, validated on some synthetic images, and computed for a corpus of covers. The results are interesting not so much for their concrete capacity estimates but for the comparisons they provide between different embedding operations, between the information found in differently-sized and -shaped pixel groups, and the results of DC normalization within pixel groups. This work suggests lessons for the future design of spatial-domain steganalysis, and also the optimization of embedding functions.

## 1 Introduction

Arguably the most vital question for steganographers is to estimate, given a particular cover and embedding method, the amount of information that can securely be embedded: this is the *capacity* of the cover, relative to a specific level of detection risk. Information theoretic quantities, particularly the *Kullback-Leibler (KL) divergence* between cover and stego distributions, can bound the secure embedding capacity [1, 2] but, naturally, they require knowledge of these distributions and this seems infeasible for digital media. Whilst information theory has produced interesting results about the rate of capacity growth [3, 4], these do not specify concrete payload sizes for particular, real-world, covers.

In this work, we aim to estimate enough of the distribution of digital images in order to draw conclusions about embedding capacity in real pictures (although it will turn out that the more interesting conclusions are not the concrete capacities derived, but the comparisons between embedding methods and the amounts of evidence found in pixel groups of different types). Our information theoretic starting point is the classic connection between hypothesis testing and KL divergence, promoted in steganography by Cachin [1]. Estimating the KL divergence empirically appears, at first sight, an impossible task: such estimators, e.g. [5], are notoriously unstable, subject to either bias or large dispersion, and typically require very large samples even for variables of low dimension. In the case of entire images the dimensionality is potentially millions, and even a huge corpus of cover images only samples the distribution with extreme sparsity. These problems do seem to prohibit direct estimation of KL divergence in images.

However, we can make some progress with two adjustments. First, rather than aiming directly for the KL divergence, we estimate its asymptotic behaviour as payload size tends to zero, which (under reasonable conditions) is determined by *Fisher's Information*; we propose an estimator which seems to exhibit more stability than those for KL divergence. Second, we consider small groups of pixels instead of complete images, modelling an image as an independent collection of pixel groups. This is certainly a significant move away from the original problem, because images are *not* constructed in this way, but it is a fact that most steganalysis methods treat images exactly as if they *were* made up of independent collections of small pixel groups (often groups as small as size 1 or 2), so the information theoretic bounds on capacity at least apply to such detectors.

This paper contains: (Sect. 2) definition of *Steganographic Fisher Information*, refined into two quantities, $I_c$ and $I_p$, which have different steganographic significance; (Sect. 3) a description of a simple estimator for $I_c$; (Sect. 4) a series of experiments estimating $I_c$ and $I_p$ for different types of pixel groups, and discussion of the significance of these results; (Sect. 5) a conclusion.

Some notation conventions are used throughout: random variables and distributions will be denoted by upper-case letters, and realizations of random variables the corresponding lower case. Vectors of either random variables or realizations will be set boldface $\boldsymbol{x} = (x_1, \ldots . x_n)$, with $n$ implicit. All logs will be to natural base.

## 2 KL Divergence and Steganographic Fisher Information

Steganalysis is an example of hypothesis testing, and capacity can be measured using the connection between KL divergence and accuracy of hypothesis tests. Let us model stego objects as random variables with distribution $P(\lambda)$, where $\lambda$ indicates the payload size. We assume that each stego object consists of a number of *locations*, whose interdependence may be arbitrary. Since detectors do not detect payload directly – they can only detect changes caused by the payload embedding – and since most embedding methods make embedding changes of approximately constant magnitude, we will measure payload *size* by the *rate* of embedding changes introduced (i.e. $\lambda$ indicates changes per cover location).

A detector must decide whether an object or sequence of objects is a realisation from $P(0)$ or $P(\lambda)$ for $\lambda > 0$. By the data processing theorem [1], any detector must have false positive and negative probabilities $(\alpha, \beta)$ satisfying

$$\alpha \log \tfrac{\alpha}{1-\beta} + (1-\alpha) \log \tfrac{1-\alpha}{\beta} \leq D_{\mathrm{KL}}(P(0) \,\|\, P(\lambda)), \tag{1}$$

where $D_{\mathrm{KL}}$ represents the Kullback-Leibler (KL) divergence

$$D_{\mathrm{KL}}(P \,\|\, Q) = \int \log\left(\tfrac{\mathrm{d}P}{\mathrm{d}Q}\right) \mathrm{d}P.$$

If the steganographer sets a maximum *risk* – a minimum on $\alpha$ and $\beta$, see [3] – the normal effect of (1) is to set a maximum on $\lambda$, the secure capacity relative to the maximum risk and chosen embedding method.

In general, (1) cannot be inverted to find the bound on $\lambda$. However, in [6] it is argued that the most important feature of embedding is its *asymptotic* capacity, as the relative payload tends to zero. This is because repeated communication must reduce the embedding rate, or face eventual certain detection. In which case, it is sufficient to consider the asymptotic behaviour of $D_{\mathrm{KL}}(P(0) \,\|\, P(\lambda))$ as $\lambda \to 0$, and given regularity conditions (see [7]), we have a simple expansion

$$D_{\mathrm{KL}}(P(0) \,\|\, P(\lambda)) \sim \tfrac{1}{2} I \lambda^2 + O(\lambda^3)$$

where $I$ is *Fisher's Information* for the distribution $P(\lambda)$ at zero, i.e. KL divergence is locally quadratic. In [6] the quadratic coefficient $\frac{1}{2}I$ was called the *Q-factor*, but we revert to the standard terminology and measure $I$.

Fisher Information can be scaled in at least two different ways, so for avoidance of doubt we will write it $I_c$, which we call *Steganographic Fisher Information (SFI) with respect to change rate*, when $\lambda$, as above, measures the embedding change rate. If the logs are to natural base, KL divergence is measured in "nats", and $I_c$ is measured in *nats per change rate squared.*

However the quantity $I_c$ is not appropriate to compare the security of embedding methods which store different payloads per embedding change, nor is it appropriate to compare across different cover sizes. We therefore introduce another type of SFI, which we call $I_p$, taking these factors into account. Well-established terminology (see e.g. [8]) is to describe the average number of payload bits conveyed per embedding change as the *embedding efficiency* $e$: for simple LSB replacement $e = 2$, but for alternative embedding methods $e$ can be significantly higher. Then a measure of risk per information transmitted is $I_c/e^2$ (nats per payload rate squared). To compare differently-sized covers, we swap KL divergence for KL divergence *rate*: the limit as $n \to \infty$ of $1/n$ times the KL divergence of cover/stego objects of size $n$. KL divergence rate is well-defined for well-behaved sources, so it makes sense to divide SFI by the cover size $n$ to obtain a size-independent measure of evidence. Therefore we define *Steganographic Fisher Information with respect to payload rate*

$$I_p = \frac{I_c}{ne^2}. \tag{2}$$

$I_p$ is measured in *symbol nats per bit squared* and it has the following interpretation: if one embeds a (small) payload of $p$ bits in a cover with $n$ locations, using an embedding method with $I_p$ symbol nats per bit squared, one expects to produce a KL divergence of approximately $I_p(p^2/n)$ nats. This reflects the square root law of steganographic capacity [9]. $I_p$ can be used to compare the security of different embedding methods in covers of arbitrary size.

The constant $r$ in the asymptotic capacity $r\sqrt{n}$ of covers of size $n$ is called the *root rate*, and it can now be determined (it is inversely proportional to $\sqrt{I_p}$) but for our purposes this is not necessary. We merely need to know that higher SFI corresponds to less secure embedding and consequently lower capacity.

### 2.1 Embedding Domain and Embedding Operations

Although the theory in this paper is applicable to any finite-size, finite-valued cover, we will consider only single-channel, byte-valued, digital images. The embedding methods we investigate will include the classic least significant bit (LSB) replacement (abbreviated LSB-R), replacement of 2 LSBs of each pixel with payload (2LSB-R), and the modified LSB method known as *LSB matching* (LSB-M) where the decision to increment or decrement a byte with non-matching LSB is taken at random unless forced at the extreme of the range. LSM matching is also known as $\pm 1$ *embedding* but we eschew this terminology: although "$\pm 1$" accurately describes the effect of embedding in LSB matching, it also describes the embedding operation of *ternary embedding* (Ter-M), where each cover pixel conveys $\log_2 3$ bits of information in its remainder (mod 3). Note that these embedding methods have the following embedding efficiencies: for LSB-R and LSB-M $e = 2$, for 2LSB-R $e = 8/3$, and for Ter-M $e = (3/2)\log_2 3$.

### 2.2 On Groups of Pixels

It is not tractable to estimate the SFI for entire images. Instead, we will imagine that they are made up of many independent pixel *groups*, where the groups are of fixed size such as $1 \times 2$ pixels, $2 \times 2$, $3 \times 3$, etc. In effect, we reduce each image to its histogram of groups: in the case of $1 \times 1$ groups this is the standard histogram, in the case of $1 \times 2$ groups it is the *adjacency histogram* or *co-occurrence matrix*.

In reducing an image to its higher-order histogram, we are destroying information. Therefore the information theoretic bound (1) does not apply universally: there might be detectors which beat this bound, if they do not make the same reduction of information. However, it is a fact that most leading steganalysis methods use only a histogram of pixel groups of fixed size. It is obvious that histogram-based detectors such as the venerable Chi-Square detector [10] use only the histogram, but also consider that LSB replacement detectors such as SPA [11] and Couples/ML [12] use only the adjacency histogram, the Triples method [13] uses only the frequencies of triplets of pixels, and even the WS detector [14, 15] uses a local filtering operation to determine prediction residuals and sums weighted residuals, so it can be expressed as a function of the histogram of groups of $3 \times 3$, or $5 \times 5$ pixels. The same is true for many detectors of LSB matching: [16] is based on the histogram, and the calibrated version in [17] can be computed from the histogram of $2 \times 2$ groups in the original. Many detectors for steganography in JPEG images consider the $8 \times 8$ DCT blocks separately, and total histograms based on those blocks (the same is not necessarily true for so-called calibrated methods, but most of their features are expressible in terms of $16 \times 16$ blocks). None of this is very surprising when one considers that image models, and therefore steganalysis methods, tend to work locally.

So there is much useful information to be found in bounds on detection accuracy, even when only small groups of pixels are considered. Indeed, comparing the amount of information found in pixel groups of different size may give interesting pointers to groups from which better steganalysis could be developed.

Furthermore, most of the LSB replacement detectors do not even consider the full adjacency histogram, preserving only the pixel difference and parity: later, we will be able to measure the amount of information thus destroyed.

## 3 Estimating Steganographic Fisher Information

We begin with a calculation of $I_c$, and then give a simple estimator for it. We can convert to $I_p$ later. Let us suppose that the cover is made up of a fixed-length sequence of symbols $(X_1, \ldots, X_n)$ draw from finite alphabet $\mathcal{X}$: these could represent the pixels of a cover image, quantized DCT coefficients, or some more complicated 1-1 transformation of the cover. The corresponding stego object is $(Y_1, \ldots, Y_n)$. For simplicity of calculations, in this work we will restrict our attention to certain classes of embedding:

1. We suppose that the embedding operation affects each cover symbol independently and equiprobably; when a cover symbol is altered by embedding, $Y_i \neq X_i$, we say that location $i$ has undergone an *embedding change*. The probability that each location receives an embedding change, the *change rate*, will be denoted $\alpha$. The embedding operation is therefore completely described by $\alpha$ and the embedding transition probabilities $P(Y = y \,|\, X = x \wedge Y \neq X)$.
2. We suppose that if a cover symbol changes, it changes to one of a fixed number of alternatives equiprobably. The number of alternatives for each cover symbol is known as the embedding *valency* $v$. That is, for each $x \in \mathcal{X}$ there exists a set $A(x)$ with cardinality $v$ such that
$$P(Y = y \,|\, X = x \wedge Y \neq X) = \begin{cases} \frac{1}{v}, & \text{for } y \in A(x) \\ 0, & \text{otherwise.} \end{cases}$$
3. We assume that the distribution of cover sequences $P(\boldsymbol{X} = \boldsymbol{x})$ is such that $P(\boldsymbol{X} = \boldsymbol{x}) = 0 \iff P(\boldsymbol{Y} = \boldsymbol{x}) = 0$; a simple sufficient condition is $P(\boldsymbol{X} = \boldsymbol{x}) > 0$ for all $\boldsymbol{x} \in \mathcal{X}^n$.

The first seems necessary to form a tractable probabilistic model of embedding. The second condition is included only because it simplifies the algebra; the last is necessary for SFI to be well-defined.

Examples of such embedding operations include simple LSB replacement: $v = 1$, and for each $x$, $A(x) = \{\overline{x}\}$ where $\overline{x}$ indicates the integer $x$ with the LSB flipped. In 2LSB replacement, $v = 3$ and $A(x) = \{\overline{x}, \hat{x}, \hat{\overline{x}}\}$, where $\hat{x}$ indicates $x$ with the 2nd LSB flipped. LSB matching does not, strictly speaking, fit these conditions because the valency is not constant (0 and 255 can only change to one alternative, all others to two alternatives). However, if the cover does not contain any extreme-valued symbols then this issue never occurs, and if the embedding were modified to allow 0 and 255 to interchange under embedding then LSB matching has $v = 2$ and $A(x) = \{x - 1, x + 1\}$ (addition modulo 256). Since changing 0 to 255, or vice versa, would be a rare occurrence, we may loosely model LSB matching under this framework and postpone to future work the extension to more general embedding operations [18].

Now we may compute $I_p$ for finite (fixed-length) sequences of cover symbols, by computing the KL divergence and extracting the leading term in $\alpha$, as follows. Fix $n$, then $P((Y_1,\ldots,Y_n)=(x_1,\ldots,x_n))$ can be expressed as

$$\begin{aligned}&P(\boldsymbol{X}=\boldsymbol{x})+\alpha\Bigg[-nP(\boldsymbol{X}=\boldsymbol{x})+\tfrac{1}{v}\sum_{|\boldsymbol{x}-\boldsymbol{y}|=1}P(\boldsymbol{X}=\boldsymbol{y})\Bigg]\\&+\alpha^2\Bigg[\tfrac{n(n-1)}{2}P(\boldsymbol{X}=\boldsymbol{x})-\tfrac{n-1}{v}\sum_{|\boldsymbol{x}-\boldsymbol{y}|=1}P(\boldsymbol{X}=\boldsymbol{y})+\tfrac{1}{v^2}\sum_{|\boldsymbol{x}-\boldsymbol{y}|=2}P(\boldsymbol{X}=\boldsymbol{y})\Bigg]+\ O(\alpha^3)\end{aligned}$$

where $|\boldsymbol{x}-\boldsymbol{y}|=1$ is shorthand to indicate that all but 1 of $x_i$ and $y_i$ are equal, and for the remaining index $y_i\in A(x_i)$, $|\boldsymbol{x}-\boldsymbol{y}|=2$ analogously. Then, using $\log(1+z)\sim z-\frac{z^2}{2}+O(z^3)$,

$$\begin{aligned}&-P(\boldsymbol{X}=\boldsymbol{x})\log\left(\frac{P(\boldsymbol{Y}=\boldsymbol{x})}{P(\boldsymbol{X}=\boldsymbol{x})}\right)\sim\alpha\Bigg[nP(\boldsymbol{X}=\boldsymbol{x})-\tfrac{1}{v}\sum_{|\boldsymbol{x}-\boldsymbol{y}|=1}P(\boldsymbol{X}=\boldsymbol{y})\Bigg]\\&+\frac{\alpha^2}{2}\Bigg[nP(\boldsymbol{X}=\boldsymbol{x})-\tfrac{2}{v}\sum_{|\boldsymbol{x}-\boldsymbol{y}|=1}P(\boldsymbol{X}=\boldsymbol{y})-\tfrac{2}{v^2}\sum_{|\boldsymbol{x}-\boldsymbol{y}|=2}P(\boldsymbol{X}=\boldsymbol{y})+\frac{\Big(\sum_{|\boldsymbol{x}-\boldsymbol{y}|=1}P(\boldsymbol{X}=\boldsymbol{y})\Big)^2}{v^2P(\boldsymbol{X}=\boldsymbol{x})}\Bigg]\\&+O(\alpha^3).\end{aligned}$$

Now observe that $\sum_{\boldsymbol{x}\in\mathcal{X}^n}P(\boldsymbol{X}=\boldsymbol{x})=1$, $\sum_{\boldsymbol{x}}\sum_{|\boldsymbol{x}-\boldsymbol{y}|=1}P(\boldsymbol{X}=\boldsymbol{y})=nv$, and $\sum_{\boldsymbol{x}}\sum_{|\boldsymbol{x}-\boldsymbol{y}|=2}P(\boldsymbol{X}=\boldsymbol{y})=\frac{n(n-1)v^2}{2}$. Thus, discarding terms $O(\alpha^3)$ and above,

$$D_{\mathrm{KL}}(\boldsymbol{X}\,\|\,\boldsymbol{Y})=\sum_{\boldsymbol{x}\in\mathcal{X}^n}-P(\boldsymbol{X}=\boldsymbol{x})\log\left(\frac{P(\boldsymbol{Y}=\boldsymbol{x})}{P(\boldsymbol{X}=\boldsymbol{x})}\right)\sim\frac{\alpha^2}{2}\left[\frac{\Big(\sum_{|\boldsymbol{x}-\boldsymbol{y}|=1}P(\boldsymbol{X}=\boldsymbol{y})\Big)^2}{v^2P(\boldsymbol{X}=\boldsymbol{x})}-n^2\right]$$

and we have computed the SFI $I_c$ (nats per change rate squared) as a relatively simple function of the distribution of cover sequences.

But how to estimate the SFI empirically? We propose a simple and naive solution. Suppose a large corpus of cover sequences (of fixed size) from which one extracts the empirical distribution, which we will write

$$p(\boldsymbol{x})=\#\text{occurrences of }\boldsymbol{x}/\text{size of corpus}.$$

For a particular embedding method, we can write $q(\boldsymbol{x})=\sum_{|\boldsymbol{x}-\boldsymbol{y}|=1}p(\boldsymbol{y})$ and then estimate the SFI by

$$\widehat{I_c}=\frac{1}{v^2}\sum_{\boldsymbol{x}}q(\boldsymbol{x})^2/p(\boldsymbol{x})-n^2. \qquad (3)$$

The sum must be taken over nonzero $p(\boldsymbol{x})$.

**Theorem 1.** *Under our assumptions, $\widehat{I_c}$ is a consistent estimator: for all $\epsilon>0$, $P(|\widehat{I_c}-I_c|>\epsilon)\to 0$ as the corpus size tends to $\infty$.*

The proof is omitted for lack of space, as was much of the preceding detail. Neither will we attempt to develop confidence intervals for the estimate (perhaps bootstraps can be applied). One could hope for much more accurate estimators inspired by those for KL divergence, but methods such as nearest-neighbours [5] seem primarily for continuous random variables. Our focus is on applying this simple estimator to get some indicative results about steganographic security.

### 3.1 Implementation Challenges

We do need a sufficiently large corpus: too few samples will lead not only to inaccuracies in $p(\boldsymbol{x})$, but also many cases of $p(\boldsymbol{x}) = 0$ and hence terms missing from the sum in (3)[1]. And, despite the simple form of the estimator, very large data sets present computational challenges.

When estimating the SFI for pixel groups of size $n$, there are potentially $256^n$ different values: while it is easy to store the histogram of such values for $n = 1, 2, 3$, current computer memories are unlikely to be large enough for $n = 4$, and certainly not for $n \geq 5$ (our experiments will involve $n$ as large as 9). The histogram may be somewhat sparse (with adjacent pixels more likely to take similar values) but even an efficient data structure cannot contain the entire histogram in memory at once. For example, in experiments on genuine images with $n = 8$ the histogram requires 47GB to store, and the data structures which make the counting process efficient more than double the memory requirement.

We must trade space for time, and therefore adopt the following procedure:

1. Each image in the corpus is considered in turn, divided into pixel groups (including, if required, DC normalization – see Subsect. 4.3), and a running total of the frequency of each group is kept. For this purpose a red-black tree is used, with the pixel value sequences as keys (sorted lexicographically) and frequency of occurrence as values. This allows logarithmic-time update and insertion, and also rapid access to the data, sorted by key, via a tree traversal. When all memory is exhausted, the (key, value) pairs are written out to disk in key order (the tree structure can be discarded) and a new histogram begins. At the end of this process, we have a number of histograms with overlapping keys, each sorted by key.
2. The histograms are merged by shuffle-sorting. The result is written to disk in chunks no larger than half the available memory; these chunks concatenate to the complete histogram of pixel groups $(\boldsymbol{x}, p(\boldsymbol{x}))$, sorted by $\boldsymbol{x}$.
3. We adjoin the value of $q(\boldsymbol{x})$ to each entry for $p(\boldsymbol{x})$:
   ```
   For each $i = 1, \ldots, n$,
       for each $y \in A(x_i)$,
           $q(\boldsymbol{x}) \mathrel{+}= p(x_1, \ldots, x_{i-1}, y, x_{i+1}, \ldots, x_n)$.
   ```
   Note that the value of $p(x_1, \ldots, x_{i-1}, y, x_{i+1}, \ldots, x_n)$ may be stored in a different histogram chunk than $p(\boldsymbol{x})$, so we must be prepared to load two chunks

[1] One warning is found in the observed value of $\sum_{p(\boldsymbol{x})>0} q(\boldsymbol{x})$: if the SFI is finite, this should total $nv$ and lower values indicate missing terms due to insufficiently-sampled data.

at a time. Because the chunks are sorted, $p(x_1, \ldots, x_{i-1}, y, x_{i+1}, \ldots, x_n)$ can be located using binary search and, in practice, is always found in one of the three or five nearest chunks to $p(\boldsymbol{x})$.

4. Finally, the entries $(\boldsymbol{x}, p(\boldsymbol{x}), q(\boldsymbol{x}))$ are scanned through, and $\widehat{I_c}$ computed via (3). At the same time, $\sum_{p(\boldsymbol{x})>0} q(\boldsymbol{x})$ can be computed for an indication as to whether the histograms are under-sampled.

Overall, the algorithm has time complexity $O(N \log N)$, where $N$ is the number of pixel groups in the corpus (though this ignores the effect of the available memory size: more memory means that the histograms are split into fewer chunks, making everything more efficient). Similar performance could be achieved by storing more data on disk and using B-trees to cache as much in memory as possible, but the implementation would be more complicated.

In the interests of making the most of our corpus, we allow pixel groups to overlap at the counting stage. Additionally, we may expect that the statistical properties of natural images are invariant under rotation, so we re-use each image in four orientations (except for groups of size $1 \times 1$, for which nothing is gained). Although this might introduce minor discrepancies into the calculations, since the groups are not truly independent, we expect any such effect to be negligible, though we recognise that some image acquisition operations (e.g. colour filter arrays) are not necessarily rotationally symmetric.

## 4 Results

We now perform some experiments, first to validate the accuracy of the SFI estimator on synthetic images, then to examine properties of genuine images under various embedding methods.

One thousand synthetic images, sized $2000 \times 1500$, were generated by taking each pixel independently at random with distribution $P(X = x) = \frac{1}{3} 2^{-|128-x|}$ (we may ignore the negligible probability that this value falls outside the range 0–255). As well as providing simple calculations for exact KL divergence, this distribution produced joint histograms with a sparsity somewhat similar to the genuine images tested later (nonetheless, further validation experiments should be carried out). Depending on pixel group size, just under $3 \times 10^9$ pixel groups are available from which to construct the empirical histograms.

For real images, we used a library of 3200 never-compressed grayscale images taken from a digital camera in RAW format, approximately 4.5 Mpixels each. As part of the conversion process from RAW to grayscale, some denoising was applied. We initially performed experiments on all 3200 images, but those results were skewed by areas of saturation in the pictures. Although saturation is an arguably natural, and certainly common, artifact we preferred to avoid it and selected 2118 of the images under the crude criterion that no more than 5000 pixels per image fall outside the range $[5, 250]$. It may be valuable to return to the study of saturated images in future work. The genuine image corpus is therefore 10 Gpixels in size; including the extra data obtained by rotating the images, just over $4 \times 10^{10}$ pixel groups are available to construct empirical histograms.

**Table 1.** Synthetic images: comparison of estimates for $I_c$ with the true values; 4 sig. fig. displayed.

| Group shape | Number of bins | Embed function | $\widehat{I_c}$ | True $I_c$ |
|---|---|---|---|---|
| $1 \times 1$ | 61 | LSB-R | 0.5000 | 0.5000 |
| $1 \times 2$ | 1719 | LSB-R | 1.000 | 1.000 |
| $1 \times 3$ | 28385 | LSB-R | 1.500 | 1.500 |
| $1 \times 4$ | 315408 | LSB-R | 2.002 | 2.000 |
| $1 \times 5$ | 2480380 | LSB-R | 2.515 | 2.500 |
| $1 \times 6$ | 14307151 | LSB-R | 3.062 | 3.000 |
| $1 \times 7$ | 61866127 | LSB-R | 3.533 | 3.500 |
| $1 \times 8$ | 203899284 | LSB-R | 2.804 | 4.000 |

*Experimental base: approx.* $3 \times 10^9$ *pixel groups*

The SFI for pixel groups of two or three can be estimated in matter of minutes, but for groups as large as eight or nine it takes many days, and the work was distributed amongst a small cluster of 12 machines. The total CPU time required for our experiments was approximately 6 weeks and the joint histogram rows $(\boldsymbol{x}, p(\boldsymbol{x}), q(\boldsymbol{x}))$ required 630GB of storage (intermediate calculations required over 2TB of storage). We also studied three smaller cover image sets, and report their results very briefly in the Subsect. 4.5.

### 4.1 Series 1: Synthetic Images

If $P(X = x) = \frac{1}{3}2^{-|128-x|}$ then it is simply to verify that, after LSB flipping,

$$P(Y = y) = \begin{cases} \frac{1}{6}2^{-|128-y|}, & y \text{ even and } y \geq 128 \text{ or } y \text{ odd and } y < 128, \\ \frac{2}{3}2^{-|128-y|}, & y \text{ odd and } y > 128 \text{ or } y \text{ even and } y < 128. \end{cases}$$

For single pixels, therefore, the true value of $I_c$ for LSB replacement evaluates to $\frac{1}{2}$. Since pixels are independent in these synthetic images, the KL divergence of a group of pixels is additive, so for a group of $n$ pixels $I_c = \frac{n}{2}$.

We computed the empirical histograms for groups of $1, 2, \ldots, 8$ pixels, from the 1000 synthetic images, and hence estimated $I_c$ using (3). Tab. 1 demonstrates how we will display the results: the number of nonempty histogram bins can be used for an indication of how dispersed were the observed groups. The estimator shows close accordance with the true value up to $n = 7$, but the histograms are under-sampled when $n = 8$ and the estimate is far off. As an indication of under-sampling, the value of $\sum_{p(\boldsymbol{x})>0} q(\boldsymbol{x})$ was only 7.67 when it should have been 8: this indicates that approximately 4% of the true terms of (3) are missing from the estimate. We also tested estimators for $I_c$ under 2LSB-R and LSB-M embedding, with similar results. These experiments validate the accuracy of the estimator, but we must beware of under-sampled histograms.

**Table 2.** Real images: comparison of different embedding methods; 3 sig. fig. displayed.

| Group shape | Number of bins | Embed function | $\widehat{I_c}$ | $\widehat{I_p}$ |
|---|---|---|---|---|
| $1 \times 1$ | 256 | LSB-R | 0.000826 | 0.000207 |
| | | 2LSB-R | 0.00236 | 0.000332 |
| | | LSB-M | 0.000110 | 0.0000275 |
| | | Ter-M | 0.000110 | 0.0000195 |
| $1 \times 2$ | 56603 | LSB-R | 0.775 | 0.0968 |
| | | 2LSB-R | 2.26 | 0.159 |
| | | LSB-M | 0.247 | 0.0309 |
| | | Ter-M | 0.247 | 0.0219 |
| $1 \times 3$ | 4430576 | LSB-R | 3.96 | 0.330 |
| | | 2LSB-R | 26.4 | 1.24 |
| | | LSB-M | 1.86 | 0.155 |
| | | Ter-M | 1.86 | 0.110 |
| $1 \times 4$ | 116786674 | LSB-R | 15.6 | 0.973 |
| | | 2LSB-R | 355 | 12.5 |
| | | LSB-M | 9.00 | 0.563 |
| | | Ter-M | 9.00 | 0.398 |
| $1 \times 5$ | 897195813 | LSB-R | 40.5 | 2.02 |
| | | 2LSB-R | 5440 | 153 |
| | | LSB-M | 24.3 | 1.21 |
| | | Ter-M | 24.3 | 0.859 |
| $1 \times 6$ | 2822410982 | LSB-R | 75.2 | 3.13 |
| | | 2LSB-R | 8940 | 209 |
| | | LSB-M | 44.7 | 1.86 |
| | | Ter-M | 44.7 | 1.32 |

*Experimental base: approx.* $4 \times 10^{10}$ *pixel groups*

### 4.2 Series 2: Comparison of Spatial-Domain Embedding Functions

Next, we fix on groups of size 1-6 pixels, and turn to our library of genuine images. We will compare the SFI of the embedding methods LSB-R, 2LSB-R, LSB-M, and Ter-M. As previously mentioned, a fair comparison must take into account the greater payload carried by 2LSB-R and Ter-M, so we convert estimates of $I_c$ into $I_p$ via (2). The results are displayed in Tab. 2 with estimated SFI displayed to 3 sig. fig., but we stress that the estimator accuracy, for large pixel groups, is probably not as high as this.

The most obvious feature is that more evidence about the presence of steganography (higher SFI) is found in larger pixel groups: this will be examined separately in a later series of experiments. We expected to see that LSB matching is more secure than LSB replacement, and this is well-supported by the larger SFI of the latter. Rather surprising, to the author, is the observation that the difference between $I_p$ estimate for LSB replacement and LSB matching reduces for larger pixel groups, appearing to settle on a ratio of only roughly 1.7, which means that a payload approximately 1.3 times the size can be embedded by LSB-M at equivalent risk (but when restricted to pixel pairs, the ratio is higher). Thus LSB-M is "approximately 1.3 times more secure" than LSB-R in a fundamental sense. This ratio is smaller than one might expect from experimental performance of the current best detectors for LSB-R and LSB-M.

The relationship between ternary embedding and LSB matching is not very interesting, merely a result of the increased embedding efficiency of the former. The comparison between LSB replacement and 2LSB replacement was a surprise: in [19] it was conjectured that 2LSB replacement might be slightly more secure on a per-payload basis, but our results here are quite the opposite. Closer examination of the sum (3) showed that pixel groups such as $(4x, 4x, 4x+3, 4x)$ were dominant, occurring almost never in cover images but often in stego images (because of cover groups of flat pixels). This might be an artifact of the denoising process undergone by the covers and we stress that these results, comparing security of embedding methods, are only applicable to this set of covers.

### 4.3 Series 3: The Effect of DC Normalization

We might believe that the DC level of pixel groups is immaterial to their frequency, e.g. that for fixed $y$ the pairs $(x, x+y)$ should occur approximately equally often, regardless of the value of $x$. This amounts to normalizing the overall DC level of each group: we could exploit this to get a better estimate of the pooled histograms. And even if we do not believe that DC level is irrelevant, we know that many steganalysis methods – particularly those for LSB replacement – discard some of the DC information. For example the Triples [13] uses the frequencies of the *trace subsets* $(2x, 2x+y, 2x+z)$, $(2x+1, 2x+1+y, 2x+1+z)$, where $y$ and $z$ are fixed but $x$ may vary: effectively, this removes the DC information except for preserving the parity of the leading pixel. Intuitively, parity is important for exploiting the "pairs of values" effects inherent in LSB replacement, so one would expect that it should be preserved. As another example, the detectors for 2LSB replacement in [19] preserve the value of the leading pixel up to (mod 4), again reflecting the structure of the embedding process.

We implemented options to subtract a constant from each pixel group so that the value of a "key pixel" in the group (selected to be as close as possible to the centre) is reduced either to zero (effectively only pixel *differences* are preserved), or to its remainder (mod 2) or (mod 4). Table 3 displays the resulting estimates of $I_p$ for two different group sizes and with LSB-M, LSB-R, and 2LSB-R embedding. From left to right, more information is discarded: first all DC information is preserved, then only the value (mod 4) of the key pixel is kept, then the value (mod 2), and finally the key pixel is zeroed and only the differences are retained. As expected, the evidence (SFI) decreases as the information is removed. But the decrease is uneven: most information about LSB replacement is preserved under normalization (mod 2) or (mod 4), but not complete normalization; this is exactly what was expected. Most information about 2LSB replacement is

**Table 3.** Real images: the effect of pixel group DC normalization; 3 sig. fig. displayed.

| Group shape | Embed function | $\widehat{I_p}$, with DC-normalization | | | |
|---|---|---|---|---|---|
| | | none | (mod 4) | (mod 2) | complete |
| $1 \times 2$ | LSB-R | 0.0968 | 0.0831 | 0.0831 | 0.0233 |
| | 2LSB-R | 0.159 | 0.129 | 0.0689 | 0.0585 |
| | LSB-M | 0.0309 | 0.0233 | 0.0233 | 0.0233 |
| $1 \times 4$ | LSB-R | 0.973 | 0.813 | 0.813 | 0.460 |
| | 2LSB-R | 12.5 | 9.50 | 3.61 | 2.79 |
| | LSB-M | 0.563 | 0.460 | 0.460 | 0.460 |

*Experimental base: approx.* $4 \times 10^{10}$ *pixel groups*

preserved if the normalization preserves DC (mod 4), but not (mod 2). And the security of LSB matching and LSB replacement are almost exactly equal if no DC information is preserved. We can interpret this to mean that the *only* additional weakness of LSB replacement, over LSB matching, is the pairs of values effect.

This suggests that LSB replacement detectors are right to consider pixel groups (usually pairs) only up to (mod 2) and pixel difference: little could be gained by retaining all DC information, and their cover models would be less widely-applicable if no normalization were performed.

### 4.4 Series 4: The Effect of Pixel Group Size and Shape

Our final series of SFI estimates is to compare the information found in pixel groups of different size and shape. We tested groups ranging from 1 to 9 pixels, including groups of different shape ($2 \times 2$ versus $1 \times 4$, $1 \times 5$ versus five pixels arranged in a "plus" shape, etc). The experiments are confined to LSB replacement, and were repeated with and without DC normalization up to (mod 2) of the key pixel. For large groups of pixels, only the normalized groups are reported, because the raw pixel groups are grossly under-sampled.

Estimates of $I_c$ and $I_p$ are displayed in Tab. 4. There are many comparisons to draw. As we saw before, DC normalization does not destroy very much information if parity is preserved, typically 10–20%. There is almost no information in the individual pixel histograms but, as one would expect, more and more information is found in groups of larger size. Comparing the values of $I_p$, we see that this is true even on a per-pixel basis. We had hoped to observe the per-pixel information levelling off as the group size was increased, and there is some suggestion that it might approach a limit on the order of $I_p = 3{-}4$, but experiments on even larger group sizes would be necessary to validate this.

An initially-surprising result was that pixel groups $2 \times n$ contained substantially less information than $1 \times 2n$ (and a cross of 5 pixels less than a group of $1 \times 5$). This is counterintuitive since one expects that pixels spatially-closer should be more tightly coupled, but recall that the question is whether embedding creates unusual groups: a row of four pixels $(x, x, x+1, x)$ is less common in

**Table 4.** Real images: the effect of pixel group size and shape; 3 sig. fig. displayed.

| Group shape | Raw groups | | | DC normalized (mod 2) | | |
|---|---|---|---|---|---|---|
| | No. bins | $\widehat{I_c}$ | $\widehat{I_p}$ | No. bins | $\widehat{I_c}$ | $\widehat{I_p}$ |
| $1 \times 1$ | 256 | 0.000826 | 0.000207 | | — | |
| $1 \times 2$ | 56603 | 0.775 | 0.0968 | 512 | 0.665 | 0.0831 |
| $1 \times 3$ | 4430576 | 3.96 | 0.330 | 108456 | 3.36 | 0.280 |
| $1 \times 4$ | 116786674 | 15.6 | 0.973 | 6025600 | 13.0 | 0.815 |
| $2 \times 2$ | 123249057 | 7.52 | 0.470 | 5628177 | 6.77 | 0.423 |
| $1 \times 5$ | 897195813 | 40.5 | 2.02 | 105345419 | 31.9 | 1.59 |
| "plus" | 1190184977 | 9.94 | 0.497 | 129473835 | 8.15 | 0.408 |
| $1 \times 6$ | 2822410982 | 75.2 | 3.13 | 662797209 | 57.5 | 2.40 |
| $2 \times 3$ | 2771668936 | 32.8 | 1.37 | 631647082 | 26.7 | 1.11 |
| $1 \times 8$ | | — | | 4107782343 | 111 | 3.48 |
| $2 \times 4$ | | — | | 3594071886 | 68.2 | 2.13 |
| $3 \times 3$ | | — | | 5624145091 | 71.1 | 1.98 |

*Experimental base: approx.* $4 \times 10^{10}$ *pixel groups*

covers, relative to its frequency in stego images, than the same pixels arranged in a square because of smooth gradients. Much more could be said on this issue, particularly in regard to image pre-processing, but lack of space precludes it.

### 4.5 A Brief Robustness Check

The experimental results above are for just one set of covers, and the images were particularly well-behaved: they had been denoised in the conversion to bitmap format, and saturated images were excluded. In order to test the robustness of our conclusions, we repeated the experiments (only for small pixel groups) in three other cover sets: one of never-compressed pictures from a mixture of digital cameras (1.5 Gpixels), one of JPEG compressed images (5 Gpixels), and one set of resampled JPEG images (4 Gpixels). That more evidence is found in $1 \times 4$ than $2 \times 2$ groups was confirmed in all sets, as was that LSB-M is more secure than LSB-R, though their $I_p$ ratio varied widely. In our main set, 2LSB-R appeared more detectable than LSB-R, and this also held for the alternative set of digital camera images, but not in the JPEG images: this probably reflects that quantization noise masks the larger stego-signal of 2LSB-R.

Finally, the preservation of almost all evidence of LSB-R under DC normalization (mod 2) was confirmed in two of the three case, but not the digital camera images. Further inspection showed that this was due to saturation in the covers: DC normalization deletes such evidence. The effect of saturation seems important: if not excluded from the corpus, it is the contribution of almost-saturated groups of pixels which dominate the sum in (3). This suggests that saturation might be exploited by steganalysis to substantial effect. Although it is easy to dismiss such detectors as trivial and dependent on flawed covers, they might be a valuable addition in practice, since saturation seems to be a common occurrence.

## 5 Conclusions

The beauty of empirical Steganographic Fisher Information estimation is that it enables us to quantify, in a properly information-theoretic way, how much evidence of payload exists in various types of pixel groups. Since almost all steganalysis methods can be described in terms of a high-order histogram, it also tells us about the fundamental security of different embedding functions[2].

Some results were as expected: larger groups contain more evidence, LSB-M is more secure than LSB-R, and pixel difference preserves most of the information about LSB-R if, and only if, the parity information is maintained. Some of the results were a surprise: more information exists in $1 \times 2n$ than $2 \times n$ groups, LSB-M is not orders of magnitude more secure than LSB-R, and 2LSB-R is particularly poor, though this last may not hold in noisier covers.

Natural directions for further research include the estimation of SFI in JPEG images, where perhaps the results can be used for feature selection as well as evaluation of embedding methods. Our experimental results required a lot of computation; they call for a better estimator for SFI than the simple plug-in histogram used here, otherwise larger pixel groups will require a heroic effort and a massive corpus. Some sort of confidence interval for the estimate can be found by bootstrapping, but this cannot take into account terms missing from (3). We believe that such effects, and more generally the problem that KL divergence is infinite if there is even the tiniest (and therefore insignificant) chance that a stego object takes a value never taken by a cover, are worthy of more study, and perhaps KL divergence can be replaced by something else. Also, we had to assume (implicitly) that the steganalyst has complete knowledge of the cover source. By Kerckhoffs' principle we should certainly be cautious about assuming less, but this is imposed anyway when we use KL divergence to measure security.

In principle, SFI estimation allows optimization of embedding: it is possible to extend this work to arbitrary independent embedding (beyond the constant-valency model in Sect. 3) and then to balance embedding efficiency against $I_c$ to derive the optimal embedding function [18]. It should be stressed that such optimality holds only for the cover sets on which the experiments are performed. Still, wide experimentation may help clarify the best shape for stego noise and the best embedding strategies amongst LSB, 2LSB, and (mod $k$)-matching.

It is instructive to compare the performance of contemporary detectors with the KL divergence predictions given, for small payloads, by $D_{\mathrm{KL}}(P(0) \parallel P(\lambda)) \sim \frac{1}{2} I_c \lambda^2$. Such results are postponed to future work, but initial experiments showed that structural detectors based on pairs of pixels DC normalized (mod 2), such as SPA [11], come very close to the bound. This may explain why, despite much literature using the same techniques, only small increments in performance have been achieved. On the other hand, the Triples detector [13], based on pixel

[2] Some steganalysis methods may, however, make use of heterogeneity of the groups of pixels within individual images, even if they are described solely in terms of a joint histogram. It is difficult to understand how this affects the information theoretic bounds provided by SFI.

triplets, is a long way from the bound. This suggests that the *symmetries* cover model is not using all the possible information in larger pixel groups, and perhaps structural steganalysis should look there for performance improvements.

## References

1. Cachin, C.: An information-theoretic model for steganography. Information and Computation **192**(1) (2004) 41–56
2. Wang, Y., Moulin, P.: Perfectly secure steganography: Capacity, error exponents, and code constructions. IEEE Transactions on Information Theory **54**(6) (2008) 2706–2722
3. Ker, A.: A capacity result for batch steganography. IEEE Signal Processing Letters **14**(8) (2007) 525–528
4. Filler, T., Ker, A., Fridrich, J.: The square root law of steganographic capacity for Markov covers. In: Media Forensics and Security XI. Volume 7254 of Proc. SPIE. (2009) 0801–0811
5. Pronzato, L., Leonenko, N., Savani, V.: A class of Renyi information estimators for multidimensional densities. Annals of Statistics **36**(5) (2008) 2153–2182
6. Ker, A.: The ultimate steganalysis benchmark? In: Proc. 9th ACM Workshop on Multimedia and Security. (2007) 141–148
7. Kullback, S.: Information Theory and Statistics. Dover, New York (1968)
8. Fridrich, J., Soukal, D.: Matrix embedding for large payloads. IEEE Transactions on Information Forensics and Security **1**(3) (2006) 390–394
9. Ker, A., Pevný, T., Kodovský, J., Fridrich, J.: The square root law of steganographic capacity. In: Proc. 10th ACM Workshop on Multimedia and Security. (2008) 107–116
10. Westfeld, A., Pfitzmann, A.: Attacks on steganographic systems. In: Proc. 3rd Information Hiding Workshop. Volume 1768 of Springer LNCS. (1999) 61–76
11. Dumitrescu, S., Wu, X., Wang, Z.: Detection of LSB steganography via sample pair analysis. IEEE Transactions on Signal Processing **51**(7) (2003) 1995–2007
12. Ker, A.: A fusion of maximum likelihood and structural steganalysis. In: Proc. 9th Information Hiding Workshop. Volume 4567 of Springer LNCS. (2007) 204–219
13. Ker, A.: A general framework for the structural steganalysis of LSB replacement. In: Proc. 7th Information Hiding Workshop. Volume 3727 of Springer LNCS. (2005) 296–311
14. Fridrich, J., Goljan, M.: On estimation of secret message length in LSB steganography in spatial domain. In: Security, Steganography, and Watermarking of Multimedia Contents VI. Volume 5306 of Proc. SPIE. (2004) 23–34
15. Ker, A., Böhme, R.: Revisiting WS steganalysis. In: Security, Forensics, Steganography and Watermarking of Multimedia Contents X. Volume 6819 of Proc. SPIE. (2008) 0501–0517
16. Harmsen, J., Pearlman, W.: Higher-order statistical steganalysis of palette images. In: Security and Watermarking of Multimedia Contents V. Volume 5020 of Proc. SPIE. (2003) 131–142
17. Ker, A.: Steganalysis of LSB matching in grayscale images. IEEE Signal Processing Letters **12**(6) (2005) 441–444
18. Ker, A.: Estimating the Information Theoretic Optimal Stego Noise. To appear in: Proc. 8th International Workshop on Digital Watermarking (2009)
19. Ker, A.: Steganalysis of embedding in two least significant bits. IEEE Transactions on Information Forensics and Security **2**(1) (2007) 46–54