Computing Dependencies between DCT Coefficients for Natural Steganography in JPEG Domain

Théo Taburet
Univ. Lille, CNRS, Centrale Lille, UMR 9189, CRIStAL
Lille, France
theo.taburet@centralelille.fr

Patrick Bas
Univ. Lille, CNRS, Centrale Lille, UMR 9189, CRIStAL
Lille, France
patrick.bas@centralelille.fr

Jessica Fridrich
Department of ECE
Binghamton
fridrich@binghamton.edu

Wadih Sawaya
IMT Lille-Douais, Univ. Lille, CNRS, CRIStAL
Lille, France
wadih.sawaya@imt-lille-douai.fr

ABSTRACT

This short paper is an extension of a family of embedding schemes called Natural Steganography, which embeds a message by mimicking heteroscedastic sensor noise in the JPEG domain. Under the assumption that the development from RAW uses linear demosaicking, we derive a closed-form for the covariance matrix of DCT coefficients from 3×3 JPEG blocks. This computation relies on a matrix formulation of all steps involved in the development pipeline, which includes demosaicking, conversion to luminance, DCT transform, and reordering. This matrix is then used for pseudo-embedding in the JPEG domain on four lattices of 8×8 DCT blocks. The results obtained with the computed covariance matrix are contrasted with the results previously obtained with the covariance matrix estimated using Monte Carlo sampling and scaling. The empirical security using DCTR features at JPEG quality 100 increased from $P_E = 14\%$ using covariance estimation and scaling to $P_E = 43\%$ using the newly derived analytic form.

CCS CONCEPTS

• **Security and privacy** → *Domain-specific security and privacy architectures*; **Intrusion/anomaly detection and malware mitigation**; **Malware and its mitigation**;

KEYWORDS

Digital image steganography, JPEG domain, sensor noise, image processing pipeline, covariance

1 INTRODUCTION

High capacity steganography, i. e., steganographic schemes with high empirical security at high embedding rates, can be designed using various strategies, such as model-based embedding, generative steganography, and side-informed embedding.

Model-based embedding first estimates a model of the cover and then embeds the message by either preserving the model [14] or by minimizing the statistical impact of embedding [15]. To some extent, the class of generative steganography [8] relies on the same rationale by sampling the content directly to carry a specific message. In [19], stego text is generated using LSTM and in [18] stego images are generated using auto-regressive models. One advantage of generative steganography is the fact that the steganographic capacity can be close to the entropy $H(X)$ of the source X by employing codes, such as the Syndrome-Trellis Codes (STCs) [7], which operate close to the corresponding rate-distortion bound.

Another possibility to increase the empirical security of an embedding scheme is to use side-information in the form of a pre-cover, a high-resolution version of the cover. Such methods use the quantization error computed during the conversion from the pre-cover to cover, large quantization errors inducing high embedding change probabilities. Notable examples of side-informed schemes include Perturbed Quantization [9], SI-UNIWARD [11], and other adaptions of cost-based embedding schemes [5]. Note that if the pre-cover is not available, one can try estimating it [17] or use multiple shots of the same image [6] to estimate "fluctuating" coefficients, i. e., coefficients associated with large quantization errors or with sensor noise of large variance.

In this paper, we propose another high capacity steganographic scheme for the JPEG domain that relies on the principles of model-based and side-informed steganography. It also belongs to the class of Natural Steganography (NS) [1, 2, 4, 16] that generates the stego signal by mimicking the sensor noise associated with a larger ISO sensitivity. Section 2 reviews the principle of NS and its recent implementation in the JPEG domain. Section 3 details the main contribution of this paper, the analytic computation of the covariance matrix among neighboring DCT coefficients after a linear development pipeline from RAW to JPEG. Section 4 compares the performances of the proposed scheme to previous implementations.

2 NATURAL STEGANOGRAPHY

Throughout this paper, we use capital letters for random variables and the corresponding lower-case symbols for their realizations. Matrices are typed in upper-case and vectors in lower-case boldface font. Matrix transpose will be denoted with a superscript "t".

2.1 The principle of NS

The main idea behind NS relies on the principle of cover-source switching [1], which consists of generating stego content that is statistically similar to the cover source acquired at a different camera setting. Here, the source is defined by the shot noise at the photo-site level due to the photon counting process occurring on CCD or CMOS sensors. This noise is independent across photosites and only depends on the sensor model and the *ISO* setting.

After embedding, the stego image generated from a cover acquired at sensitivity ISO_1 should have the same statistical properties as a cover image acquired at sensitivity ISO_2 in order to guarantee high empirical security. The shot noise $N^{(i)}$ at ISO_i, $i \in \{1, 2\}$, follows a zero-mean Gaussian distribution with variance determined by the luminance of the noiseless photo-site value μ:

$$N^{(i)} \sim \mathcal{N}(0, a_i\mu + b_i), \tag{1}$$

where the pair of parameters (a_i, b_i) depends only on the sensor and ISO_i. Consequently, one can show [1] that, at the photo-site level, the stego image mimicking an image acquired at sensitivity ISO_2 is distributed as:

$$S \sim \mathcal{N}(0, (a_2 - a_1)\mu + b_2 - b_1). \tag{2}$$

This idea was the cornerstone of NS embedding schemes proposed for monochrome sensors in the spatial domain assuming a simplified development pipeline that includes quantization, gamma transform, and downscaling [1, 2]. The method was shown to provide high steganographic capacity with high empirical security.

2.2 Embedding algorithm in the JPEG domain

In the current paper, we study NS when the pre-cover is acquired with a color sensor, developed with linear demosaicking, converted to grayscale, and JPEG compressed. The embedding is executed directly in the JPEG domain by manipulating the quantized DCT coefficients.

The difference between monochrome and color sensors was studied in [4] with the conclusion that independent embedding on each DCT coefficient offers high empirical security for monochrome sensors, but not for color sensors. This is due to the fact that demosaicking introduces dependencies among neighboring DCT coefficients. When these dependencies are not taken into account, the embedding scheme becomes highly detectable at high JPEG Quality Factors (QF). To overcome this problem, the authors of paper [16] modeled these dependencies using the multi-variate Gaussian model with the covariance matrix of the stego signal in the DCT domain $\hat{\Sigma}$ estimated from a constant-luminosity RAW image altered by shot-noise. The embedding was then designed to respect the required covariance among stego DCT coefficients. The empirical security of a simulated-embedding scheme[1] was indeed larger than when the embedding was assuming independent DCT coefficients [4].

The embedding scheme is summarized in Figure 1 and can be decomposed into different steps described below:
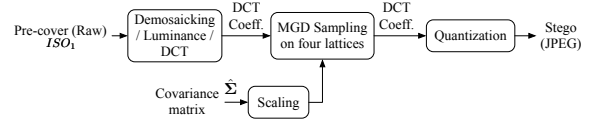
---

[1]As commonly done for cost-based embedding schemes, here the message is not embedded but the embedding changes are simulated by computing the embedding change probabilities and sampling according to them.


Figure 1: Overview of the embedding scheme presented in [16].

(1) **Estimation of the covariance matrix $\hat{\Sigma}$ between DCT coefficients of 3×3 neighboring 8×8 blocks.** Since it is computationally infeasible to estimate the covariance matrix for each block of the cover image, the estimation is performed on a constant-luminosity RAW image with photo-site values $\mu = 2^{12}$ coded on 14 bits and corrupted with the stego signal S given by (2). This estimation uses $N_o = 6 \times 10^4$ observations of 24×24 DCT coefficients obtained from a developed (3480×4640) RAW image.

(2) **Beginning of the development (demosaicking, luminance transform, and DCT).** The cover RAW image follows a classical development pipeline to generate grayscale JPEG images. After demosaicking, the standard RGB to luminance transform[2] given by: $\mathbf{y}_l = 0.299\,\mathbf{y}_r + 0.587\,\mathbf{y}_g + 0.114\,\mathbf{y}_b$, is applied, followed by a 2D-DCT transform on 8×8 blocks.

(3) **Scaling of $\hat{\Sigma}$.** Since we assume that the development is linear, and in order to take into account the conversion from RGB to luminance, the covariance matrix associated with each block is scaled as $\hat{\Sigma}' = \gamma\hat{\Sigma}$, where γ represents the scaling factor given by:

$$\gamma = \frac{0.299^2(a\bar{x}_r + b) + 0.587^2(a\bar{x}_g + b) + 0.114^2(a\bar{x}_b + b)}{(0.299^2 + 0.587^2 + 0.114^2)(a2^{12} + b)}, \tag{3}$$

where $\bar{x}_r$, $\bar{x}_g$, and $\bar{x}_b$ represent, respectively, the average photo-site value of the red, green, and blue component of the block that is sampled.

(4) **Sampling on four lattices.** In [16], the authors have shown that for this development, the stego signal generated on two non 8-connected blocks is independent and that the dependencies between 8-connected blocks are solely due to demosaicking. Consequently, we can use four lattices $\{\Lambda_1, \Lambda_2, \Lambda_3, \Lambda_4\}$ depicted in Figure 2 to sample the stego signal in the DCT domain. DCT blocks belonging to lattice Λ_1 are sampled independently, blocks belonging to lattice Λ_2 are sampled conditionally on the four diagonal blocks, blocks belonging to lattice Λ_3 are sampled conditionally on the four vertical and horizontal blocks, and blocks of Λ_4 are sampled conditionally on the 8 surrounding blocks. We can show that for a linear development, the distribution of the stego signal for each lattice Λ_j ($j \in \{1, .., 4\}$) and block i follows a Multivariate Gaussian Distribution (MGD):

$$\mathcal{N}(\mathbf{m}_{i,j}, \Sigma_{i,j}), \tag{4}$$

where for $j \in \{2, 3, 4\}$, the expectation vector $\mathbf{m}_{i,j}$ and covariance matrix $\Sigma_{i,j}$ of the conditional distribution are computed

---

[2]Without loss of generality we assume that no other color transform is applied, however if one is applied, it only changes the different weights.

using the Schur complement of the estimated covariance matrix $\hat{\Sigma}$ (see [13]).

(5) **JPEG quantization.** The simulated stego signal is quantized using the JPEG quantization matrix for a given Quality Factor (QF). One can also compute the pmf of the stego signal on JPEG coefficients at the expense of increased complexity. Note that in order to perform practical embedding, one must use 64 sub-lattices in each block (one for each DCT coefficient), this can be done by computing the conditional pmf and the associated costs (see [16]). The pmf $\pi_{q,i}$ for each coefficient i considering a Q-ary alphabet and symbol q is also used to estimate the average payload embedded in each coefficient by computing the entropy

$$H(\pi_{q,i}) = -\sum_{q=1}^{Q} \pi_{q,i} \log_2 \pi_{q,i}. \tag{5}$$

As detailed in [16], the $\pi_{q,i}$ are computed by dividing the normal marginal distribution for coefficient i into Q bins and then computing the pmf.
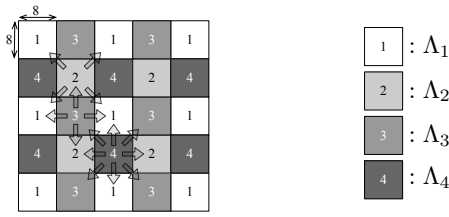
Figure 2: The four lattices Λ_1, Λ_2, Λ_3 and Λ_4, used for embedding.

The proposed simulated embedding scheme has pros and cons. On the one hand, it offers good empirical security for medium JPEG QFs (85 and 75) and for linear or close to linear demosaicking algorithms (bilinear or VNG, see Table 2 in [16]). On the other hand, the estimation of the covariance matrix combined with its scaling (3) are only approximations that decrease the empirical security of the whole scheme, especially for high QFs.

We consequently derive in this paper a closed-form of the covariance matrix Σ for bilinear demosaicking. This matrix is directly computed from the photo-site values of the RAW cover image and does not need to be scaled. This approach is validated in section 4 by showing the results are equivalent to the ones obtained by simulated embedding at the photo-site level.

3 ANALYTIC DERIVATION OF THE COVARIANCE MATRIX

A RAW image captured using a color sensor can be developed using a variety of processes, which include the bilinear interpolation or non-linear gradient based interpolations, such as DCB or AAHD. Below, we consider the simplest case of bilinear demosaicking.

In order to compute the covariance matrix Σ of the stego signal in the DCT domain, we need to decompose the development pipeline into a sequence of linear steps illustrated in Figure 3 and summarized below:

(1) Demosaicking, here we consider the Bayer CFA (Color Filter Array) pattern shown in Figure 4.
(2) Luminance averaging, we consider this process because the embedding is performed in the luminance domain.
(3) Crops to $8k \times 8k$ blocks, with $k \in \{1, 3\}$, this pre-processing is needed before the DCT transform. The parameter k is used to compute the covariance matrices used either for lattice Λ_1 ($8^2 \times 8^2$ covariance matrix), lattices Λ_2 and Λ_3 ($5 \cdot (8)^2 \times 5 \cdot (8)^2$ covariance matrices), and Λ_4 ($(3 \times 8)^2 \times (3 \times 8)^2$ covariance matrix).
(4) Conversion from row scan to block scan, this permutation is also needed before the DCT transform.
(5) DCT transform.

These linear operations can be written as a sequence of matrix multiplications. From a flattened vector of observations obtained from $k \times k$ blocks of photo-sites, we can write this operation pipeline as a single matrix $\mathbf{M}$, such as $\mathbf{t_I} = \mathbf{M} \cdot \mathbf{y}$, where $\mathbf{t_I}$ is the flattened DCT block corresponding to the observation vector $\mathbf{y} \in \mathbb{N}^{(8k+2)^2}$. This vector $\mathbf{y}$ is an unfolded $(8k + 2) \times (8k + 2)$ matrix whose elements are photo-sites observations.
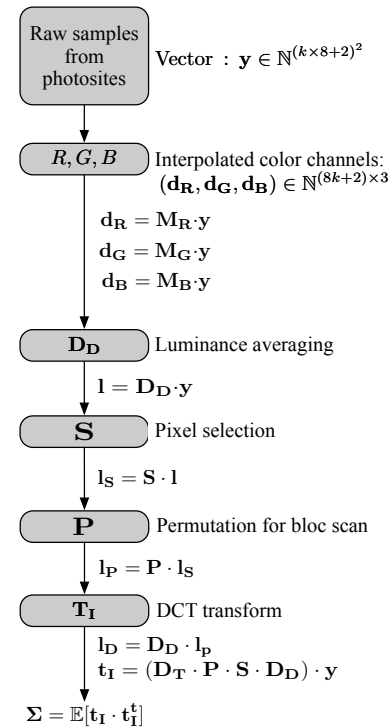
Figure 3: Different steps involved in the development pipeline.

3.1 Photo-site selection

The selection is composed of k^2 neighboring blocks, each composed of 8×8 photo-sites with the surrounding border included since interpolated values inside a 8×8 block need photo-site values surrounding this block (see Figure 4). The result of flattening this matrix is the vector of the photo-sites samples denoted $\mathbf{y} \in \mathbb{N}^{(8k+2)^2}$.

3.2 Demosaicking

The RAW cover image from the sensor first undergoes the demosaicking process. To this end, we retrieve the interpolation kernels used by the linear demosaicking process to generate the full color image (i. e., a full set of color triples) from the CFA (see Figure 4).

It is possible to write the demosaicking operations as matrix multiplications. We define matrices $\mathbf{M_R}$, $\mathbf{M_G}$, $\mathbf{M_B}$ of size $(8k + 2)^2 \times (8k + 2)^2$, such that the result of the matrix multiplication of the vector $\mathbf{y}$ with one of these matrices is the unfolded version of the corresponding color channel after demosaicking:

$$\mathbf{d_R} = \mathbf{M_R} \cdot \mathbf{y}, \ \mathbf{d_G} = \mathbf{M_G} \cdot \mathbf{y}, \ \mathbf{d_B} = \mathbf{M_B} \cdot \mathbf{y}. \tag{6}$$

For example, for the green channel, assuming that $i \in \mathbb{N}^{(8k+2)^2}$ represents the photo-site index in $\mathbf{y}$, there are two cases to compute the matrix $\mathbf{M_G}$.

(1) If index i corresponds to a Green photo-site on the Bayer CFA, this photo-site does not need color interpolation:

$$\text{row}_i(\mathbf{M_G}) = \text{unfold} \begin{pmatrix} \ddots & \vdots & \vdots & \vdots & \ddots \\ \cdots & 0 & 0 & 0 & \cdots \\ \cdots & 0 & 1 & 0 & \cdots \\ \cdots & 0 & 0 & 0 & \cdots \\ \ddots & \vdots & \vdots & \vdots & \ddots \end{pmatrix} i \tag{7}$$

(2) If index i corresponds to a pixel which needs to be interpolated, then:

$$\text{row}_i(\mathbf{M_G}) = \text{unfold} \begin{pmatrix} \ddots & \vdots & \vdots & \vdots & \ddots \\ \cdots & 0 & 0.25 & 0 & \cdots \\ \cdots & 0.25 & 0 & 0.25 & \cdots \\ \cdots & 0 & 0.25 & 0 & \cdots \\ \ddots & \vdots & \vdots & \vdots & \ddots \end{pmatrix} i \tag{8}$$

Note that for channels Red and Blue, we have to use four different convolution kernels to obtain the equations similar to equations (7) and (8):

$$[1], \begin{bmatrix} 0.25 & 0 & 0.25 \\ 0 & 0 & 0 \\ 0.25 & 0 & 0.25 \end{bmatrix}, \begin{bmatrix} 0.5 \\ 0 \\ 0.5 \end{bmatrix} \text{ and } \begin{bmatrix} 0.5 & 0 & 0.5 \end{bmatrix}.$$

3.3 Luminance averaging:

According to the standard ITU-R BT 601 for RGB color spaces, the relative luminance can be calculated as a linear combination of RGB components:

$$\mathbf{Y} = 0.299 \cdot \mathbf{d_R} + 0.587 \cdot \mathbf{d_G} + 0.114 \cdot \mathbf{d_B}. \tag{9}$$

In our case, this means that to perform color reconstruction and luminance averaging, we can define the demosaicking and averaging matrix $\mathbf{D_D}$ as:

$$\mathbf{l} = \mathbf{D_D} \cdot \mathbf{y} = (0.299 \cdot \mathbf{M_R} + 0.587 \cdot \mathbf{M_G} + 0.114 \cdot \mathbf{M_B}) \cdot \mathbf{y}. \tag{10}$$
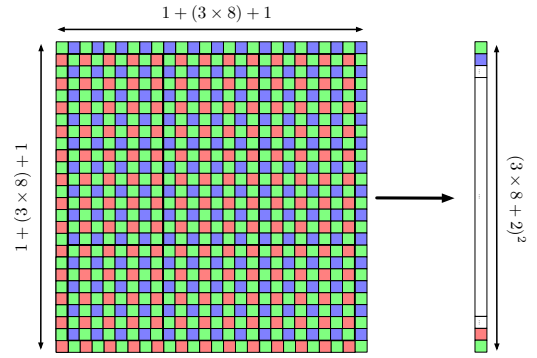

Figure 4: Unfolding of the photo-sites array for $k = 3$. Bayer patterns for the DCT blocks are surrounded by bold lines.
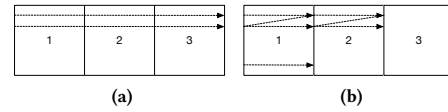

Figure 5: For $k = 3$, scan order of the unfolded version of (a) $\mathbf{l_S}$, (b) $\mathbf{l_P}$

3.4 Selection of pixels of interest

As stated above, the surrounding edges of $k \times k$ blocks of samples have been included in order to take into account the convolution window during demosaicking. Once the demosaicking operations have been carried out, they can now be discarded. We consequently define a selection matrix $\mathbf{S}$ to perform this operation on the flattened version of the output $\mathbf{l}$ of the luminance averaging step such that :

$$\mathbf{l_S} = \mathbf{S} \cdot \mathbf{l}, \tag{11}$$

where $\mathbf{S}$ is composed of an arrangement of $k \cdot 64 \times k \cdot 64$ identity matrices allowing us to select horizontal segments of $8k$ pixels. The resulting vector after the selection step is a $(8k)^2$ dimensional vector that we denote $\mathbf{l_S} \in \mathbb{R}^{(8k)^2}$ with $\mathbf{S} \in \mathbb{R}^{(8k)^2 \times (8k+2)^2}$.

3.5 Permutation

As explained above, we inherently work with an arrangement of $k \times k$ blocks of photo-site values. However, the flattening operation induces by default a row scan and not a block-by-block scan. To this end, we define a permutation matrix $\mathbf{P}$:

$$\mathbf{l_P} = \mathbf{P} \cdot \mathbf{l_S}, \tag{12}$$

where $\mathbf{P}$ is an arrangement of 8×8 photo-sites used to select horizontal segments of 8 pixels. This allows us to rearrange the vector $\mathbf{l_S}$ into $\mathbf{l_P}$ as illustrated in Figure 5.

3.6 Vectorized DCT Transform

Given an 8×8 block in the spatial domain, $\mathbf{X_s}$, its block 8×8 DCT transform denoted here as $\mathbf{X_{DCT}}$ can be written as the following matrix multiplication :

$$\mathbf{X_{DCT}} = \text{DCT}(\mathbf{X_S}) = \mathbf{A} \cdot \mathbf{X_S} \cdot \mathbf{A}^t = \mathbf{A} \cdot (\mathbf{A} \cdot \mathbf{X_S}^t)^t$$

With :

$$\mathbf{A} = \begin{bmatrix} a & a & a & a & a & a & a & a \\ b & d & e & g & -g & -e & -d & -b \\ c & f & -f & -c & -c & -f & f & c \\ d & -g & -b & -e & e & b & g & -d \\ a & -a & -a & a & a & -a & -a & a \\ e & -b & g & d & -d & -g & b & -e \\ f & -c & c & -f & -f & c & -c & f \\ g & -e & d & -b & b & -d & e & -g \end{bmatrix},$$

and :

$$[a, b, c, d, e, f, g]^t =$$
$$[\cos \tfrac{\pi}{4}, \cos \tfrac{\pi}{16}, \cos \tfrac{\pi}{8}, \cos \tfrac{3\pi}{16}, \cos \tfrac{5\pi}{16}, \cos \tfrac{3\pi}{8}, \cos \tfrac{7\pi}{16}]^t.$$

It should be observed that the multiplication by $\mathbf{A}$ and $\mathbf{A}^t$ is due to the fact that the DCT transform is separable and processes the columns and rows independently.

In order to compute the covariance matrix of the spatial signal $\mathbf{X_S}$, we use vector notation by transforming the matrix $\mathbf{X_S} \in \mathbb{R}^{8 \times 8}$ into a vector $\mathbf{x_S} \in \mathbb{R}^{64}$ by concatenating the columns.

As a result, the 8×8 matrix $\mathbf{A}$ is transformed into a 64×64 matrix $\mathbf{A_v}$ given by :

$$\mathbf{A_v} = \begin{bmatrix} \mathbf{A} & 0 & \dots & 0 \\ 0 & \mathbf{A} & 0 & \vdots \\ \vdots & 0 & \ddots & 0 \\ 0 & \cdots & 0 & \mathbf{A} \end{bmatrix}. \tag{13}$$

We also define a transpose operator as the matrix $\mathbf{T} \in \mathbb{R}^{64 \times 64}$ (see [4]) and the transpose operation $\mathbf{X_S}^t$ is then equivalent to the multiplication $\mathbf{T} \cdot \mathbf{x_S}$, and the vector form of the DCT 8×8 block $\mathbf{X_{DCT}}$ finally becomes :

$$\mathrm{DCT}_v(\mathbf{x_S}) = \mathbf{A_v} \cdot \mathbf{T} \cdot \mathbf{A_v} \cdot \mathbf{T} \cdot \mathbf{x_S} = \mathbf{D_T} \cdot \mathbf{x_S}, \tag{14}$$

where $\mathbf{D_T} = \mathbf{A_v} \cdot \mathbf{T} \cdot \mathbf{A_v} \cdot \mathbf{T}$.

3.7 Covariance matrix computation

The matrix multiplication sequence can thus be written as

$$\mathbf{t_I} = (\mathbf{D_T} \cdot \mathbf{P} \cdot \mathbf{S} \cdot \mathbf{D_D}) \cdot \mathbf{y}. \tag{15}$$

Each matrix is thus a linear operator allowing us to calculate the vectorized DCT transform of $k \times k$ blocks from a sample of photo-sites. Therefore, only the $\mathbf{y}$ vector depends on the content of the RAW image. The matrix product $\mathbf{D_T} \cdot \mathbf{P} \cdot \mathbf{S} \cdot \mathbf{D_D}$ is content independent and can therefore be written as a single matrix.

The covariance matrix can now be calculated as follows:

$$\Sigma = \mathbb{E}[\mathbf{t_I} \cdot \mathbf{t_I^t}] = \mathbf{M} \cdot \mathbb{E}[\mathbf{y} \cdot \mathbf{y^t}] \cdot \mathbf{M^t}, \tag{16}$$

with

$$\mathbf{M} = \mathbf{D_T} \cdot \mathbf{P} \cdot \mathbf{S} \cdot \mathbf{D_D}. \tag{17}$$

4 RESULTS

In this section, we evaluate the empirical security of NS in the JPEG domain for images acquired with a color sensor. A total of 4800 512×512 images were obtain by partitionning into non-overlapping patches 100 RAW images acquired at ISO_2 using the Z-CAM-E1 camera [4] to build the covers subset. In a likewise way the pseudo-stego subset of 4800 512×512 images is built from 100 RAW images

acquired at ISO_1 and undergoes a switch [1] from $ISO_1 = 100$ to $ISO_2 = 200$. The parameters to perform the switch from ISO_1 to ISO_2 were $(a, b) = (1.15, -1150)$. The empirical security is evaluated as the minimal total classification error probability under equal priors, $P_E = \min_{P_{FA}} \tfrac{1}{2}(P_{FA} + P_{MD})$, with P_{FA} and P_{MD} standing for the false-alarm and missed detection rates. The JPEG images are steganalyzed with the DCTR feature set [10] and the low-complexity linear classifier [3].

The following embedding schemes are compared:

- *Pseudo-embedding*, simulated embedding is performed at the photo-site level with the stego signal generated using (2). These results can be considered as a baseline but do not correspond to any practical embedding scheme in the JPEG domain.

- *4 Lattices-scaling*: The embedding pipeline uses the estimated covariance matrix to perform the sampling. The covariance matrix used is $\hat{\Sigma}$ as explained in Section 2.2.

- *4 Lattices-analytic*: The sampling mechanism is the same but we use the closed-form expression for the covariance matrix Σ, detailed in Section 3, to simulate the stego signal in the DCT domain.

- *SI-UNIWARD*: For comparison with the current state of the art, all images have also been embedded using SI-UNIWARD [11] with the embedding rate set to 1 bit per nzAC DCT coefficient which corresponds to the maximal payload of this embedding scheme.

The capacity of both 4-Lattice implementations ranges from approximately 1 bpnzAC at QF 75 to 2 bpnzAC at QF 100 (see Figure 11 of [16]). The results of these experiments are shown in Table 1. First, observe that there is no difference between generating the stego signal at the photo-sites or in the JPEG domain, which validates the fact that our statistical model in the JPEG domain is equivalent to the one at the photo-site level. Second, the closed-form of the covariance matrix provides a security gain w.r.t. the scheme proposed in [16], especially for high JPEG QFs. This is due to the fact that the covariance estimation proposed in [16] deals with blocks of constant photo-site values. This approximation is detrimental whenever ones wants to generate a high resolution stego signal. Note, however, that the computation of the covariance matrix is associated at a high computational burden since the Shur complement matrix needed to compute the conditional probability distribution has to be evaluated for each block and not only once as in [16]. The comparison with SI-UNIWARD shows that cost-based SI-embedding is more detectable than NS embedding.[3]

In Table 2 we evaluate the sensitivity of our methods w.r.t. other popular demosaicking schemes and compare the security of NS implemented with the closed-form of the covariance matrix and with scaled estimated covariance as proposed in [16]. Cover images are developed using a speficic demosaicking method and stego images are generated either assuming bilinear demosaicking (the value on the left) or with the scaled estimated covariance matrix [16] as also explained in Section 2.2. While the closed-form of the covariance matrix offers the best performance for bilinear demosaicking, it cannot be used to model other demosaicking schemes, in which case it is better to estimate the covariance matrix and scale it, especially for high QFs.

---

[3]Note that SI-UNIWARD only needs an uncompressed image while NS needs substantially more information – the RAW image.

JPEG QF	Pseudo embedding	4 Lattices scaling [16]	4 Lattices analytical	SI-Uniward 1 bpnzac
100	40.2	13.9	42.9	0.0
95	40.9	30.3	41.2	0.4
85	41.9	39.8	41.2	12.3
75	41.3	40.4	41.6	24.8

Table 1: Empirical security (P_E in %) for different quality factors and embedding strategies on E1Base with bilinear demosaicking. DCTR features combined with regularized linear classifier are used for steganalysis.

JPEG QF	Bilinear	VNG	DCB	AAHD
100	42.9 / 13.9	0.0 / 0.0	0.0 / 0.1	0.0 / 0.0
95	41.2 / 30.3	0.1 / 22.7	0.1 / 4.5	0.1/ 3.4
85	41.2 / 39.8	0.4 / 36.9	0.6 / 32.6	1.1/ 25.4
75	41.6 / 40.4	1.5 / 40.9	3.9 / 39.8	1.1/ 35.7

Table 2: Empirical security (P_E (%) and sensitivity w.r.t the demosaicking algorithm used to develop cover images using analytic covariance matrix computed for bilinear demosaicking (left), or using scaled estimated covariance matrix (right) [16].

5 CONCLUSIONS AND FURTHER WORKS

This paper is an extension of the former works done on Natural Steganography in the JPEG domain [4, 16] and on color sensors. We have shown that for the bilinear development pipeline, it is possible to derive an explicit form of the covariance matrix of the stego signal in the DCT domain. Similar derivations should be possible for other linear demosaicking algorithms such as VNG. When used for embedding, the closed-form of the covariance matrix enables us to simulate embedding in the JPEG domain with the same empirical security as in the photo-site domain. However, when the demosaicking process is not linear, it is then better to estimate the covariance matrix using Monte Carlo sampling as in [16] and scale it according to (3) than to use the closed-form derived here.

Future work will focus first on finding non-Gaussian models that match the distributions of non-linear demosaicking algorithms, such as DCB or AAHD. This can be done for example by continuing to use a covariance matrix to model dependencies but also by modifying marginals to fit arbitrary distributions. A second line of research concerns applications of the derived closed-form covariance matrix Σ to steganography. It may allow considering proper synchronization of embedding changes in classical, cost-based steganography similar to [12].

ACKNOWLEDGMENTS

REFERENCES

[1] Patrick Bas. 2016. Steganography via Cover-Source Switching. IEEE Workshop on Information Forensics and Security (WIFS).
[2] Patrick Bas. 2017. An embedding mechanism for Natural Steganography after down-sampling. IEEE ICASSP.
[3] Rémi Cogranne, Vahid Sedighi, Jessica Fridrich, and Tomáš Pevný. 2015. Is ensemble classifier needed for steganalysis in high-dimensional feature spaces?. In Information Forensics and Security (WIFS), 2015 IEEE International Workshop on. IEEE, 1–6.
[4] Tomáš Denemark, Patrick Bas, and Jessica Fridrich. 2018. Natural Steganography in JPEG Compressed Images. In Electronic Imaging. San Francisco, United States.
[5] Tomas Denemark and Jessica Fridrich. 2015. Side-informed steganography with additive distortion. In Information Forensics and Security (WIFS), 2015 IEEE International Workshop on. IEEE, 1–6.
[6] T. Denemark and J. Fridrich. 2017. Steganography with Multiple JPEG Images of the Same Scene. 12, 10 (October 2017), 2308–2319.
[7] Tomas Filler, Jan Judas, and Jessica Fridrich. 2011. Minimizing additive distortion in steganography using syndrome-trellis codes. Information Forensics and Security, IEEE Transactions on 6, 3 (2011), 920–935.
[8] Jessica Fridrich. 2009. Steganography in digital media: principles, algorithms, and applications. Cambridge University Press.
[9] Jessica Fridrich, Miroslav Goljan, and David Soukal. 2004. Perturbed quantization steganography with wet paper codes. In Proceedings of the 2004 workshop on Multimedia and security. ACM, 4–15.
[10] Vojtěch Holub and Jessica Fridrich. 2015. Low-complexity features for JPEG steganalysis using undecimated DCT. IEEE Transactions on Information Forensics and Security 10, 2 (2015), 219–228.
[11] 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), 1–13.
[12] Weixiang Li, Weiming Zhang, Kejiang Chen, Wenbo Zhou, and Nenghai Yu. 2018. Defining Joint Distortion for JPEG Steganography. In Proceedings of the 6th ACM Workshop on Information Hiding and Multimedia Security. ACM, 5–16.
[13] A. Papoulis and U. Pillai. 2002. Probability, Random Variables and Stochastic Processes. Mac Graw Hill.
[14] P. Sallee. 2003. Model-Based Steganography. In International Workshop on Digital Watermarking (IWDW), LNCS, Vol. 2.
[15] Vahid Sedighi, Rémi Cogranne, and Jessica Fridrich. 2016. Content-Adaptive Steganography by Minimizing Statistical Detectability. Information Forensics and Security, IEEE Transactions on 11, 2 (2016), 221–234.
[16] Théo Taburet, Patrick Bas, Wadih Sawaya, and Jessica Fridrich. 2019. A Natural Steganography Embedding Scheme Dedicated to Color Sensors in the JPEG Domain. In Electronic Imaging. Burlingame, United States.
[17] Z. Wang, Z. Qian, X. Zhang, M. Yang, and D. Ye. 2018. On Improving Distortion Functions for JPEG Steganography. IEEE Access 6 (2018), 74917–74930. https://doi.org/10.1109/ACCESS.2018.2884198
[18] Kuan Yang, Kejiang Chen, Weiming Zhang, and Nenghai Yu. 2018. Provably Secure Generative Steganography Based on Autoregressive Model. In International Workshop on Digital Watermarking. Springer, 55–68.
[19] Zhongliang Yang, Xiaoqing Guo, Ziming Chen, Yongfeng Huang, and Yu-Jin Zhang. 2018. RNN-Stega: Linguistic Steganography Based on Recurrent Neural Networks. IEEE Transactions on Information Forensics and Security (2018).