Using the Sensor Noise Model to Design Better Steganographic Schemes
IWDW 2020

Patrick Bas (Lille, France, 4°C),
joint work with Théo Taburet, Wadih Sawaya, Jessica Fridrich, Quentin Giboulot and Rémi Cogranne, Solène Bernard, Tomas Pevny
discussions with Andrew Ker

November 25, 2020, IWDW Keynote (Melbourne, Australia, 31°C)
Part 1: Motivations / Inspirations / Problems

Part 2: Covariance matrix of the sensor noise in the DCT domain

Part 3: Two Embedding strategies and associated results

Part 4: Conclusions
Part 1: Motivations / Inspirations / Problems

Part 2: Covariance matrix of the sensor noise in the DCT domain

Part 3: Two Embedding strategies and associated results

Part 4: Conclusions
Alice’s steganographic game
Alice’s steganographic game

Kerckhoffs’ principle / WCA:
- known scheme
- known source(s) (databases)
- known payload

Ingrid
Igor

Embedding
Stego
Decoding

Cover
Message
Key

(super) Eve

Innocent ???
Alice’s steganographic game

Alice’s goals:
- maximize embedding capacity
- maximize Eve’s error rate
Inspiration #1: Adaptive/Additive embedding - Sampling

Fridrich’s group [Filler2010]

- **Adaptive:**
  - Associate to each sample \(i\) a cost \(\rho_{i,k}\) for modification \(k\)
  - Associate to each sample \(i\) a modification probability \(\pi_{i,k}\) for modification \(k\)

- **Additive:**
  - Embed and minimize \(\sum_{i,k} \pi_{i,k} \rho_{i,k}\)
  - Practical solution: STC [Filler2011]

- Equivalent to **sampling** (nearly, see [Kin-Cleaves2020]):
  - Sample \(k\) from:
    \[
    \pi_{i,k} = \frac{\exp (-\lambda \rho_{i,k})}{\sum_k^{Q/2} \exp (-\lambda \rho_{i,k})}
    \]
Inspiration #1: Adaptive/Additive embedding - Sampling

Cover (HILL, 0.3bpp)  \( \rho_{i,\pm 1} \)  \( \pi_{i,\pm 1} \)

Problem:
- How to **sample** from multivariate probabilities?
- How to **embed** from multivariate probabilities?
Inspiration #2: Correlations (synchronizations) in steganography

Coding view:

▶ **Non-additive** costs

Statistical view:

▶ **Non-independent** modifications

Signal processing view:

▶ Induce **correlations** between embedding changes (ex: $\Pr(+1; +1) > \Pr(+1; -1)$)
▶ a.k.a. synchronizations
Inspiration #2: Correlations (synchronizations) in steganography

Related schemes:
- Synch [Filler2010], spatial, Gibbs
- CMD [Li2015], spatial
- DeJoin [Zhang2017], spatial, joint costs, conditional probabilities
- GINA [Wang2019], spatial extension of CMD, Color
- BBC, BBM [Wang2020], DCT

Principles:

1. Decompose the image samples (pixel/DCT) into disjoint lattices/groups \{\Lambda_1, \ldots, \Lambda_n\}
2. compute costs in \Lambda_1 (additive)
3. embed in \Lambda_1 (additive)
4. compute costs in \Lambda_2 \textbf{given the modifications} in \Lambda_1 (non-additive)
5. embed in \Lambda_2 (additive)
6. iterate until \Lambda_n
Inspiration #2: Correlations (synchronizations) in steganography

Updating "rules" (CMD):

\[ \rho_i' (+1) = \frac{1}{9}\rho_i (+1), \text{ if } \mu_i > 0, \rho_i' (+1) = \rho_i (+1), \text{ else} \]

Emb. on 4 lattices

HILL, 0.3bpp

CMD, 0.3bpp
Inspiration #2: Correlations (synchronizations) in steganography

Problem:

▶ Theoretical justifications behind theses synchronization heuristics?
Inspiration #3: Steganography mimicking a statistical model

Model based steganography [Sallee2003]

- \(D_{KL}(C, S) = 0\) (Stego-security in watermarking [Cachin1998][Cayre2008])
- Steganography mimicking the sensor noise of a scanner [Franz2002]

⇒ High capacity: source \(S\), capacity = \(H(S)\)

From [Sallee2003]
Inspiration #4: Steganography distorting a statistical model

MiPod [Sedighi2016]

- Image noise estimated using Wiener prediction
- Noise Variance $\sigma^2_i$ (Gaussian model) $\Rightarrow$ GLRT $\Rightarrow$ deflection coefficient $\delta^2_i = \frac{\pi^2_i}{\sigma^4_i}$
  $\Rightarrow$ Cost $\rho_i = \frac{\pi_i}{\sigma^4_i}$

![Diagram showing examples of Stego distributions ($\sigma^2_i = 1$) with $\pi_i = 0$, $\pi_i = 0.2$, and $\pi_i = 1/3$.](image-url)
Inspiration #3/#4: Steganography mimicking/distorting a statistical model

Problems:

- What’s the "good" model of the noise in the JPEG domain?
- How to compute it?
- How to use the noise model in steganography?
Raised questions

Q1: What's the good model of the noise in the JPEG domain? and how to compute it?

Q2: Is there a theoretical justification behind the synchronization heuristics?
    ▶ See part 2

Q3: How to use the noise model in steganography?

Q4: How to sample from multivariate probabilities? How to embed from multivariate probabilities?
    ▶ See part 3
Why do you bother us with problems?

Teaser:

- Natural Steganography: QF100, 2 bits per non-zero-AC coef, SRNet, $P_E = 37\%$

- $\Sigma - JMiPod$: QF100, $P_E = +15\%$ w.r.t. SI-Uniward at 0.4 bit per coefficient
Part 1: Motivations / Inspirations / Problems

Part 2: Covariance matrix of the sensor noise in the DCT domain

Part 3: Two Embedding strategies and associated results

Part 4: Conclusions
What’s the sensor noise?

Poisson-Gaussian sensor noise

- additive **independent** noise $N_i \sim \mathcal{N}(0, a\mu_i + b)$
- $\mu_i$ “clean” photo-site value at location $i$
- parameters $a$ and $b$ constant for a given camera and a given **sensitivity** (ISO parameter), can be easily estimated on RAW images
Noise processing: from RAW to JPEG

Generic development pipeline

Sensor noise in the DCT domain:

- Multivariate Gaussian (linear approximation)
- Covariance matrix $\Sigma_d$

Q1: What's the good model of the noise in the JPEG domain? and how to compute it?

- This model!
- See the 4 next slides
How to compute it? The algebraic way [Taburet2020]

Linear development pipeline

Considered DCT blocks

Correlation range (26x26)
How to compute it? The algebraic way [Taburet2020]

Algebra:

\[ n_d = Mn_p = TLDsp \]

\[ \Sigma_d = M\Sigma_p M^t \]

- size of \( n_p \) and \( n_d \): 26\(^2\) and 24\(^2\)
- size of \( \Sigma_d \): \((24^2 \times 24^2)\)
How to compute it? The estimation way [Giboulot2020]

Development pipeline

**Estimation setup:**

- pairs \((n_p, n_d)\)
- \((a, b)\) known
- \(M\) can be estimated by Least Square Estimation
- More versatile!
Analysis of the covariance matrix

- \( \mu = \text{cst} \iff \Sigma_p \propto I \)
- Linear pipeline, luminance
- Only 4 blocks, row scan:

Size of \( \Sigma'_d = 256 \times 256 \)
Analysis: whole covariance
Analysis: intra-block correlations
Analysis (intra-block): correlations due to demosaicking artifacts
Analysis (intra-block): correlations due to low pass filtering

Effect of low-pass filtering of heteroscedastic noise:
Intra-block Covariance matrix after different developments

Bi-Linear (rawpy)  Preview (Mac OSX)

AAHD (rawpy)  DXO Lab
Analysis (inter-block)

Covariance sub matrix

Inter-block correlations encode block continuities!
  ▶ Rational for BBC [Wang2020]

Q2: Is there a theoretical justification behind the synchronization heuristics?
  ▶ Yes, preserve the natural correlations of the development pipeline
Part 1: Motivations / Inspirations / Problems

Part 2: Covariance matrix of the sensor noise in the DCT domain

Part 3: Two Embedding strategies and associated results

Part 4: Conclusions
Strategy 1: Steganography mimicking a statistical model

Q3: How to use the noise model in steganography?

- **Natural steganography**: The stego signal mimics the sensor noise [Bas2016]

Note:

ISO₂ > ISO₁
Strategy 1: Steganography mimicking a statistical model

Principle of Model-Based steganography (RAW domain):

- **Cover at ISO\(_1\):**
  \[ X_{i}^{(1)} \sim \mathcal{N}(\mu_i, a_1 \mu_i + b_1) \]

- **Cover at ISO\(_2\):**
  \[ X_{i}^{(2)} \sim \mathcal{N}(\mu_i, a_2 \mu_i + b_2) \]

- **Steganographic signal** \( S_i \sim \mathcal{N}(0, (a_2 - a_1)x_i + b_2 - b_1) \)

- **If** \( \mu_i \approx x_i \)** ⇒** \( D_{KL}(X_{i}^{(1)} + S_i, X_{i}^{(1)}) \approx 0 \)

**Requirement in the DCT domain (before quantization):**

\[ S \sim \mathcal{N}(0, \Sigma_d) \]
How to sample from multivariate probabilities?

Q4: How to sample from multivariate probabilities? How to embed from multivariate probabilities?

▶ See the 8 next slides

Three probabilistic properties:

1. Independency rule:
   ▶ \(\{B_1, \ldots, B_n\}\) independent blocks
   \[
   \Pr(B_1, \ldots, B_n) = \Pr(B_1) \cdots \Pr(B_n)
   \]

2. Dependency and chain rule of conditional probabilities:
   ▶ \(\{B_1, \ldots, B_n\}\) dependent blocks
   \[
   \Pr(B_1, \ldots, B_n) = \Pr(B_1)\Pr(B_2|B_1) \cdots \Pr(B_n|B_{n-1}, \ldots, B_1)
   \]

3. Computing \(\Pr(B_n|B_{n-1}, \ldots, B_1)\) is easy on Gaussian distributions (Schur and Cholesky decompositions)
Independent blocks?

⇒ Two non-connected blocks are independent
Embedding at the block level

Decomposition into 4 macro-lattices \( \{ \Lambda_1, \Lambda_2, \Lambda_3, \Lambda_4 \} \)

\[
P(s_d^1) = P(s_{\Lambda_1}),
\]
Embedding at the block level

Decomposition into 4 macro-lattices \( \{\Lambda_1, \Lambda_2, \Lambda_3, \Lambda_4\} \)

\[
P(s^2_d) = P(s_{\Lambda_2} | s_{\Lambda_1}),
\]
Embedding at the block level

Decomposition into 4 macro-lattices \( \{ \Lambda_1, \Lambda_2, \Lambda_3, \Lambda_4 \} \)

\[
P (s^3_d) = P (s_{\Lambda_3} | s_{\Lambda_2}, s_{\Lambda_1}),
\]
Embedding at the block level

Decomposition into 4 macro-lattices \( \{\Lambda_1, \Lambda_2, \Lambda_3, \Lambda_4\} \)

<table>
<thead>
<tr>
<th>1</th>
<th>3</th>
<th>1</th>
<th>3</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>

Compliant with the chain rule:

\[
P(s_d) = P(s_{\Lambda_4} | s_{\Lambda_3}, s_{\Lambda_2}, s_{\Lambda_1})
\]

\[
P (s_d) = P (s_{\Lambda_1}, s_{\Lambda_2}, s_{\Lambda_3}, s_{\Lambda_4})
\]

\[
= P (s_{\Lambda_1}) P (s_{\Lambda_2} | s_{\Lambda_1}) P (s_{\Lambda_3} | s_{\Lambda_1}, s_{\Lambda_2}) P (s_{\Lambda_4} | s_{\Lambda_1}, s_{\Lambda_2}, s_{\Lambda_3})
\]
Embedding at the coefficient level

Example \((\Lambda_2 | \Lambda_1)\):

\[ s_{\Lambda_2 | \Lambda_1} \sim \mathcal{N}(m_{\Lambda_2 | \Lambda_1}, \Sigma_{\Lambda_2 | \Lambda_1}) \]

1. Compute the conditional matrix

\[
\begin{bmatrix}
\Lambda_1 & \Lambda_1 \\
\Lambda_2 & \Lambda_1 \\
\Lambda_1 & \Lambda_1
\end{bmatrix}
\]

\[ \Sigma_{\Lambda_1, \Lambda_2} \]

Schur, Cholesky

\[
(m_{\Lambda_1 | \Lambda_2}, \Sigma_{\Lambda_1 | \Lambda_2})
\]

Sample/Embed within each block

\[
\begin{cases}
    s_0 = m_0 + L(0, 0) \cdot n_0 \\
    s_{1|0} = m_1 + L(1, 0) \cdot n_0 + L(1, 1) \cdot n_1 \\
    \vdots
\end{cases}
\]

with \(n_i \sim \mathcal{N}(0, 1)\)

Compute the PMFs \(\pi_i(k)\), the costs, the Capacity Q-arry embedding
Embedding scheme (J-Cov-NS)

Algorithm 1 J-Cov-NS

- **Inputs**: Cover RAW $X_p$, message, key, developed Cover $X_d$ (DCT domain) and JPEG $X_J$;
- **For** each sub-lattice $\Lambda_i \in \{\Lambda_1, \Lambda_2, \Lambda_3, \Lambda_4\}$ and **For** each DCT block **do**:
  - Compute the covariance matrix $\Sigma_d$
  - Compute $m|\Lambda_{i-1} \ldots \Lambda_1$ and $\Sigma|\Lambda_{i-1} \ldots \Lambda_1$
    - **For** each JPEG coefficient of $X_J$ **do**:
      - Compute the conditional (Gaussian) distribution
      - Compute the PMF $\pi(k)$;
      - Sample w.r.t. $\pi(k)$ or embed w.r.t the costs;
      - Sample in the continuous domain (needed for conditioning)
- **Output** JPEG stego $Y$.

Embedding on $4 \times 64$ lattices
Benchmark: steganalysis

Dedicated setup:

- Cover $ISO_1$
- $ISO_1 = 100$,
- $ISO_2 = 200$:
  - 10000 covers $ISO_2$
  - 10000 stegos $ISO_1 \rightarrow 2$

Setup (E1Base):

Classifier train on covers/stego 5000 pairs and tested on 5000 pairs

$$P_E = \min \left( \frac{P_{FA} + P_{MD}}{2} \right)$$
Empirical security for NS [Taburet2020]

<table>
<thead>
<tr>
<th>$P_E$ (%) / QF</th>
<th>SI-Uniward 1 bpnzAC</th>
<th>$H$ (bpnzAC)</th>
<th>J-Cov-NS</th>
<th>Intra-block only</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>0.0</td>
<td>2.0</td>
<td>42.9</td>
<td>0.0</td>
</tr>
<tr>
<td>95</td>
<td>0.4</td>
<td>2.2</td>
<td>41.2</td>
<td>0.2</td>
</tr>
<tr>
<td>85</td>
<td>12.3</td>
<td>2.4</td>
<td>41.2</td>
<td>15.8</td>
</tr>
<tr>
<td>75</td>
<td>24.8</td>
<td>7.0</td>
<td>41.6</td>
<td>25.2</td>
</tr>
</tbody>
</table>

Linear Classifier with DCTR

<table>
<thead>
<tr>
<th>$P_E$ (%) / QF</th>
<th>J-Cov-NS</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>37.4</td>
</tr>
<tr>
<td>95</td>
<td>31.2</td>
</tr>
<tr>
<td>85</td>
<td>39.8</td>
</tr>
<tr>
<td>75</td>
<td>35.0</td>
</tr>
</tbody>
</table>

SRNet

<table>
<thead>
<tr>
<th>QF / $P_E$ in %</th>
<th>$K = 1$</th>
<th>$K = 2$</th>
<th>$K = 3$</th>
<th>$K = 5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>1.0</td>
<td>12.9</td>
<td>28.7</td>
<td>40.4</td>
</tr>
<tr>
<td>95</td>
<td>3.5</td>
<td>23.6</td>
<td>39.3</td>
<td>40.9</td>
</tr>
<tr>
<td>85</td>
<td>39.8</td>
<td>39.8</td>
<td>39.8</td>
<td>41.8</td>
</tr>
<tr>
<td>75</td>
<td>40.4</td>
<td>40.4</td>
<td>40.4</td>
<td>41.2</td>
</tr>
</tbody>
</table>

Q-arry embedding
Strategy 2: Steganography distorting a statistical model [Giboulot2020]

Minimize the **multivariate** deflexion

- Sensor noise distributed as $\mathcal{N}(0, \Sigma_d)$

- Additive stego-signal minimizing the $D_{KL}$ for a given payload size distributed as $\mathcal{N}(0, \alpha \Sigma_d)$

**Embedding mechanism:**

- Very similar to NS (but only $2 \times 64$ lattices, diagonal correlations are negligible)

- Can be applied on any (estimated) pipeline
Strategy 2: Steganography distorting a statistical model [Giboulot2020]

Results, Efficient-net B3-stride 1, BOSSBase

Linear development pipeline

BOSS development pipeline
Part 1: Motivations / Inspirations / Problems

Part 2: Covariance matrix of the sensor noise in the DCT domain

Part 3: Two Embedding strategies and associated results

Part 4: Conclusions
Conclusions

- Correlations matters (especially for high QF or in the spatial/color domain)
- Correlations are due to demosaicking, resizing, denoising, DCT transform, ...
- Joint probabilities matters (correlations within a block and between a block)
- The development pipeline matters
- Assumptions matters (doesn’t work on Sigma sensors, on Leica/Kodak sensors which is not Gaussian)
One more correlation!

Correlations are everywhere [Bernard2020]

Covariance matrix of the stego signal after Adv-Emb [Tang2019]

Analysis (canceling discontinuities)
Inspirations / References

(Filler2010) : T. Filler and J. Fridrich. Gibbs construction in steganography, TIFS
(Filler2011) : T. Filler, J. Judas, and J. Fridrich. Minimizing additive distortion in steganography using syndrome-trellis codes, TIFS
(Kin-Clea2020) : C. Kin-Cleaves and A.D. Ker. Simulating Suboptimal Steganographic Embedding, IH-MMSec
(Li2015) : B. Li, M. Wang, X. Li, S. Tan, and J. Huang. A strategy of clustering modification directions in spatial image steganography, TIFS
(Wang2020) : Y. Wang, W. Zhang, W. Li, and N. Yu. Non-additive cost functions for jpeg steganography based on block boundary maintenance, TIFS
(Sallee2003) : P. Sallee. Model-based steganography, IWDW
(Cachin1998) : C. Cachin. An information-theoretic model for steganography, IH
(Cayre2008) : F. Cayre and P. Bas. Kerckhoffs-based embedding security classes for WOA data-hiding, TIFS
(Franz2002) : E. Franz. Steganography preserving statistical properties, IH
(Sedighi2016) : V. Sedighi, R. Cogranne, and J. Fridrich. Content-adaptive steganography by minimizing statistical detectability, TIFS
(Taburet2020) : T. Taburet, P. Bas, W. Sawaya, and J. Fridrich. Natural steganography in jpeg domain with a linear development pipeline, TIFS
(Giboulot2020) : Q. Giboulot, R. Cogranne, and P. Bas. Jpeg Steganography With Side Information From The Processing Pipeline, ICASSP
(Bas2016) : P. Bas. Steganography via Cover-Source Switching. WIFS
(Giboulot2020) : Q. Giboulot, P. Bas, R. Cogranne. Synchronization Minimizing Statistical Detectability for Side-Informed JPEG Steganography, WIFS
(Bernard2020) : S. Bernard, P. Bas, J. Klein, T. Pevny, Adversarial Embedding in the JPEG Domain Induces Correlations Between DCT Coefficients to Remove Blocking Artifacts Generated by Additive Embedding, Arxiv
(Tang2019) : W. Tang, B. Li, S. Tan, M. Barni, and J. Huang. CNN-based adversarial embedding for image steganography. TIFS