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

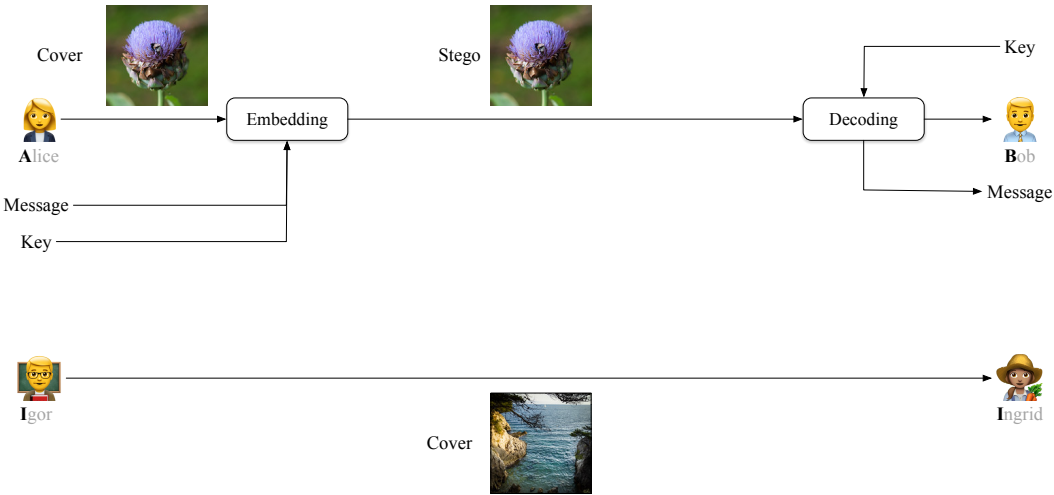
Part 1: Motivations / Inspirations / Problems

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

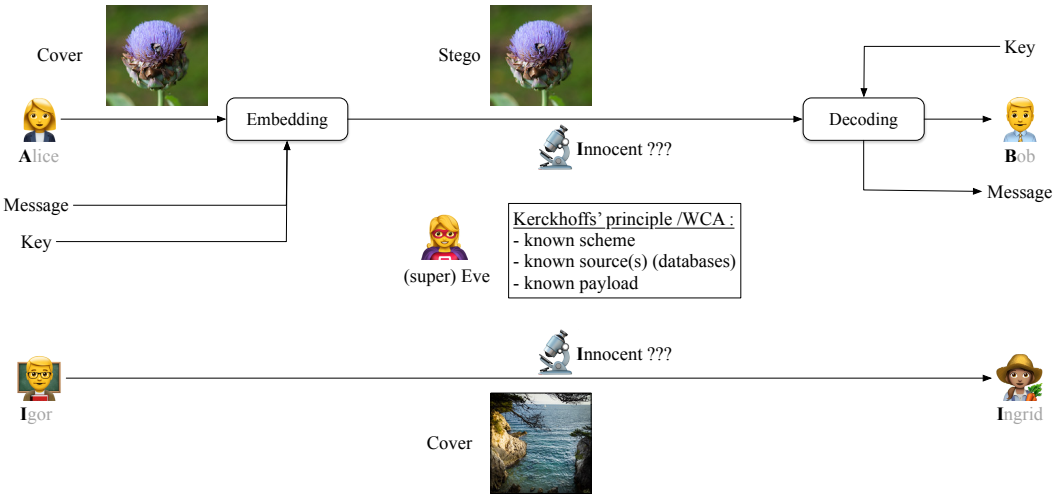
Part 3: Two Embedding strategies and associated results

Part 4: Conclusions

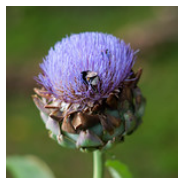
Alice's steganographic game



Alice's steganographic game



Alice's steganographic game



Alice's goals :
- maximize embedding capacity
- maximize Eve's error rate

Inspiration #1: Adaptive/Additive embedding - Sampling

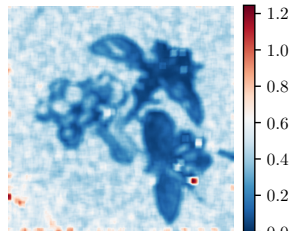
Fridrich's group [Filler2010]

- **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:

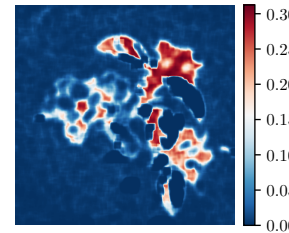
$$\pi_{i,k} = \frac{\exp\left(-\lambda\rho_{i,k}\right)}{\sum_{k}^{Q/2}\exp\left(-\lambda\rho_{i,k}\right)}$$

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

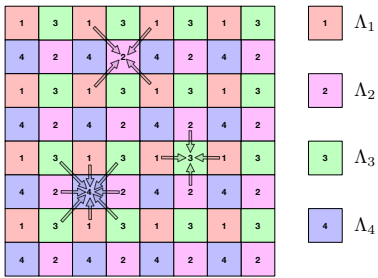
Principles:

1. Decompose the image samples (pixel/DCT) into disjoint lattices/groups $\{\Lambda_1, \ldots, \Lambda_n\}$
2. compute costs in Λ_1 (additive)
3. embed in Λ_1 (additive)
4. compute costs in Λ_2 **given the modifications** in Λ_1 (**non-additive**)
5. embed in Λ_2 (additive)
6. iterate until Λ_n

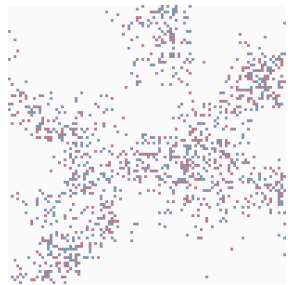
Inspiration #2: Correlations (synchronizations) in steganography

Updating "rules" (CMD):

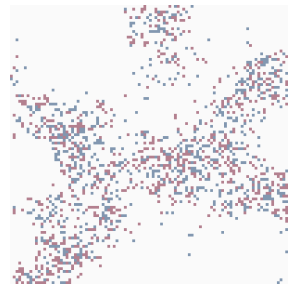
$$\rho_i'(+1) = \frac{1}{9}\rho_i(+1), \text{ if } \mu_i > 0,\ \rho_i'(+1) = \rho_i(+1), \text{ else}$$



Emb. on 4 lattices

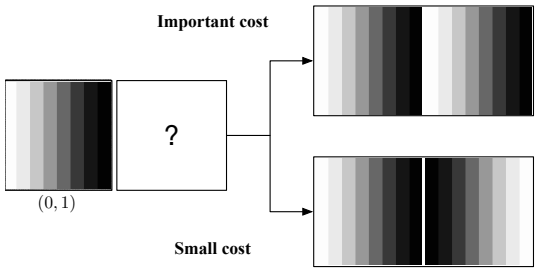


HILL, 0.3bpp



CMD, 0.3bpp

Inspiration #2: Correlations (synchronizations) in steganography



BBM and BBC

Problem:

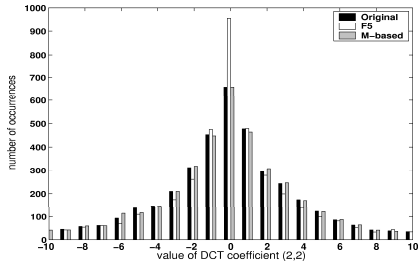
- Theoretical justifications behind theses synchronization heuristics?

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]

$\Rightarrow$ High capacity: source S, $\mathrm{capacity} = H(S)$

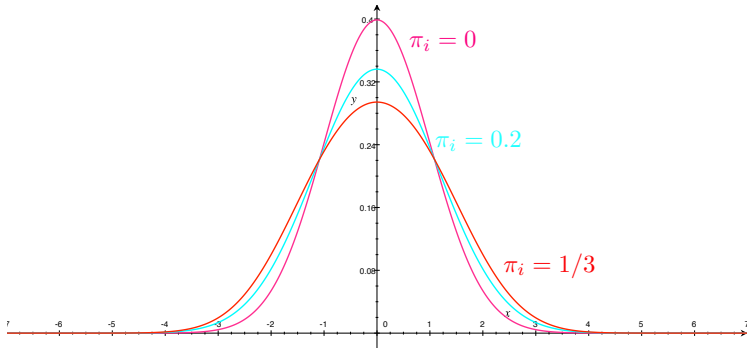


From [Sallee2003]

Inspiration #4: Steganography distorting a statistical model

MiPod [Sedighi2016]

- Image noise estimated using Wiener prediction
- Noise Variance σ_i^2 (Gaussian model) $\Rightarrow$ GLRT $\Rightarrow$ deflection coefficient $\delta_i^2 = \pi_i^2/\sigma_i^4$ $\Rightarrow$ Cost $\rho_i = \pi_i/\sigma_i^4$



Examples of Stego distributions ($\sigma_i^2 = 1$)

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, $P_E = 37\%$
- $\mathbf{\Sigma} - \mathit{JMiPod}$: 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

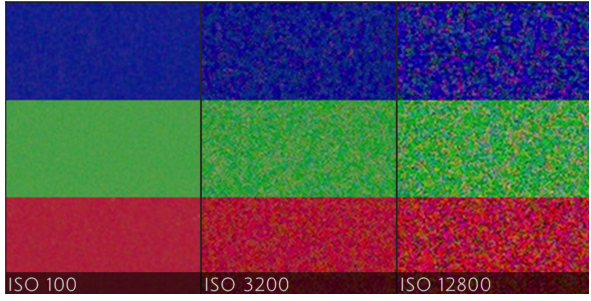
Part 3: Two Embedding strategies and associated results

Part 4: Conclusions

What's the sensor noise?

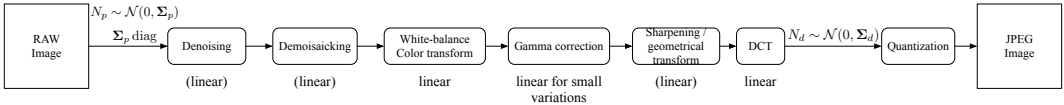
Poisson-Gaussian sensor noise

- additive **independent** noise $N_i \sim \mathcal{N}(0, a\mu_i + b)$
- μ_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



www.zavosphotography.com

Noise processing: from RAW to JPEG



Generic development pipeline

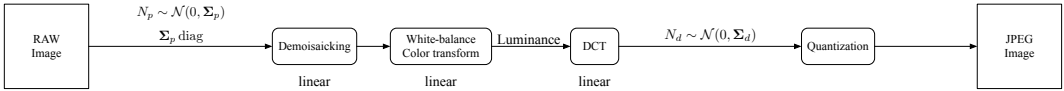
Sensor noise in the DCT domain:

- Multivariate Gaussian (linear approximation)
- Covariance matrix $\mathbf{\Sigma}_d$

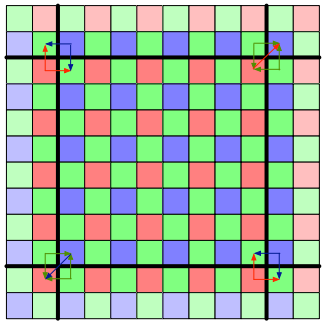
Q1: What's the good model of the noise in the JPEG domain ? and how to compute it?

- This model!
- See the 4 next slides

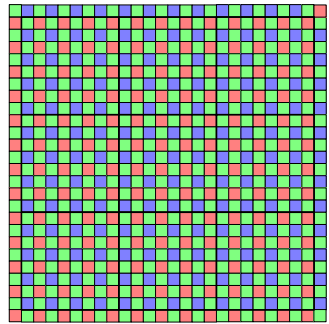
How to compute it? The algebraic way [Taburet2020]



Linear development pipeline

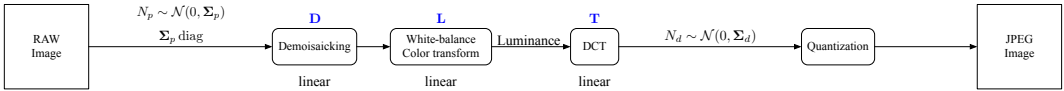


Considered DCT blocks



Correlation range (26x26)

How to compute it? The algebraic way [Taburet2020]

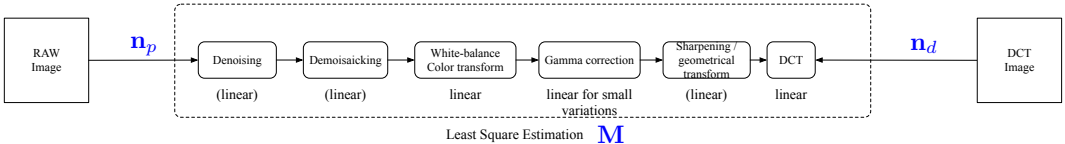


Linear development pipeline

Algebra:

- $\mathbf{n}_d = \mathbf{M}\mathbf{n}_p = \underbrace{\mathbf{T}\,\mathbf{L}\,\mathbf{D}}_{\mathbf{M}}\,\mathbf{s}_p$
- $\mathbf{\Sigma}_d = \mathbf{M}\mathbf{\Sigma}_p\mathbf{M}^t$
- size of $\mathbf{n}_p$ and $\mathbf{n}_d$: 26^2 and 24^2
- size of $\mathbf{\Sigma}_d$: $(24^2 \times 24^2)$

How to compute it? The estimation way [Giboulot2020]



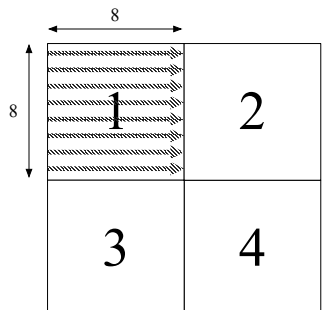
Development pipeline

Estimation setup:

- pairs $(\mathbf{n}_p, \mathbf{n}_d)$
- (a, b) known
- M can be estimated by Least Square Estimation
- More versatile!

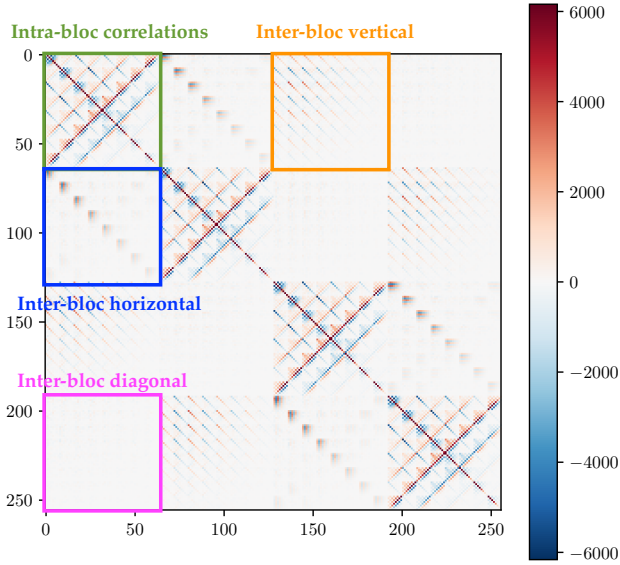
Analysis of the covariance matrix

- $\mu = \text{cst} \Leftrightarrow \mathbf{\Sigma}_p \propto \mathbf{I}$
- Linear pipeline, luminance
- Only 4 blocks, row scan:

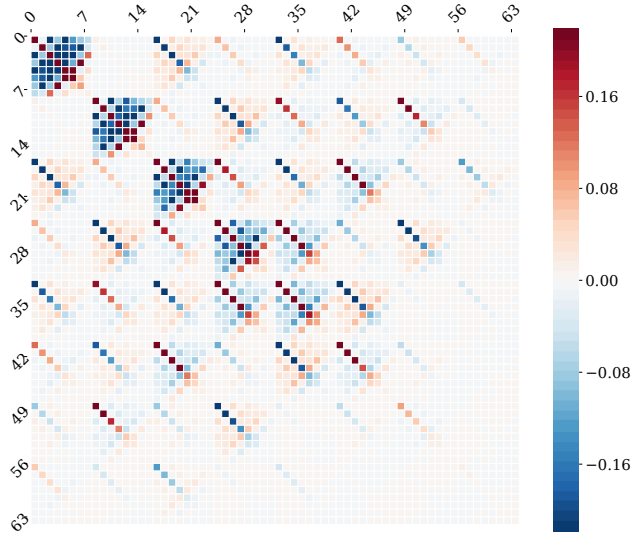


- Size of $\mathbf{\Sigma}'_d = 256 \times 256$

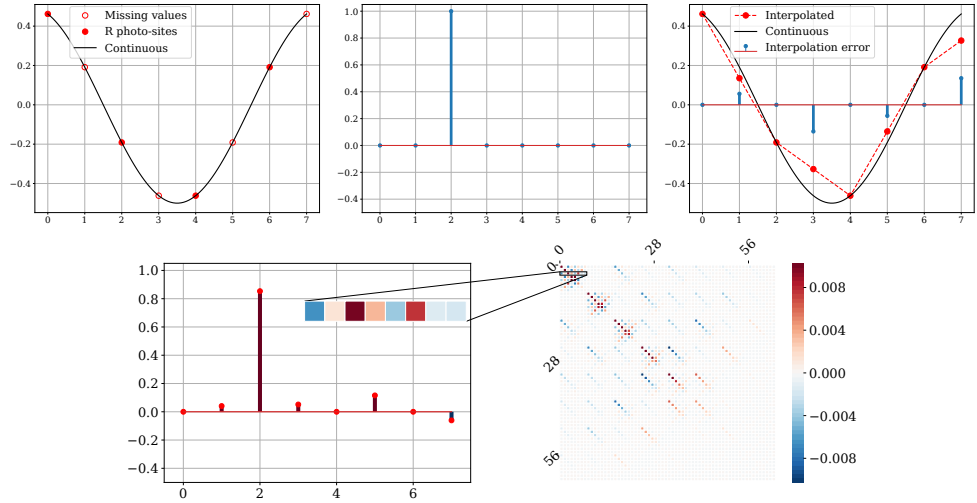
Analysis: whole covariance



Analysis: intra-block correlations

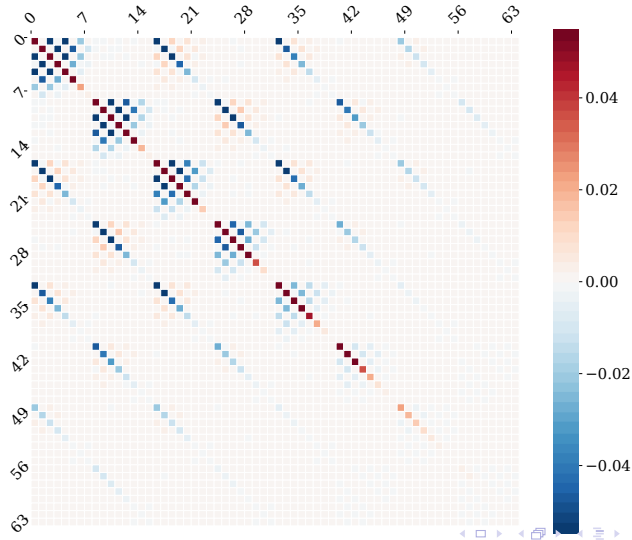


Analysis (intra-block): correlations due to demosaicking artifacts



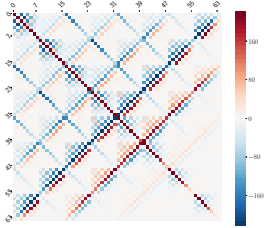
Analysis (intra-block): correlations due to low pass filtering

Effect of low-pass filtering of heteroscedastic noise:

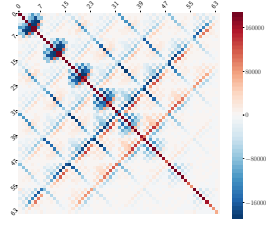


Intra-block Covariance matrix after different developments

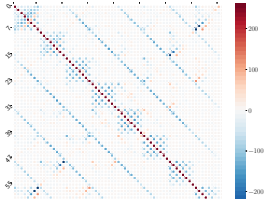
Bi-Linear (rawpy)



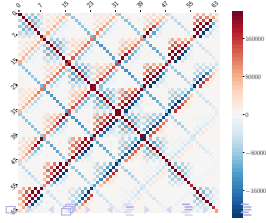
Preview (Mac OSX)



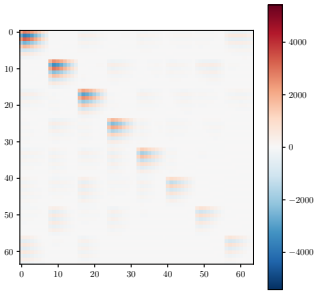
AAHD (rawpy)



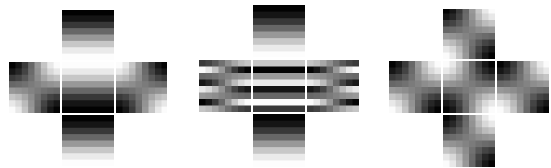
DXO Lab



Analysis (inter-block)



Covariance sub matrix



mode $(1, 0)$ mode $(5, 0)$ mode $(1, 1)$

Most correlated modes

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

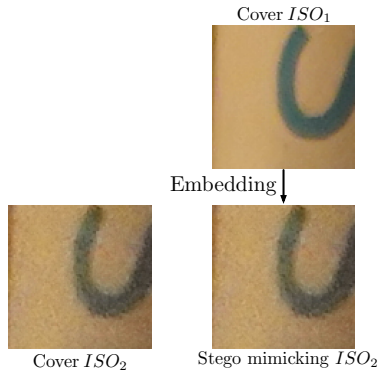
Part 3: Two Embedding strategies and associated results

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:
$ISO_2 > ISO_1$

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 $\mu_i \approx x_i \Rightarrow D_{KL}(X_i^{(1)} + S_i, X_i^{(1)}) \approx 0$

Requirement in the DCT domain (before quantization):

$$\mathbf{S} \sim \mathcal{N}(0, \mathbf{\Sigma}_d)$$

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:
   - $\{\mathbf{B}_1, \ldots, \mathbf{B}_n\}$ **independent** blocks

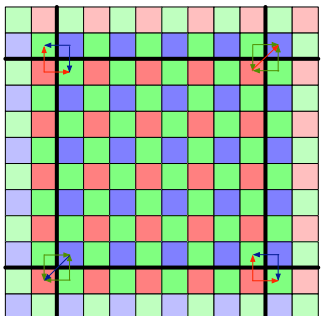
$$\Pr(\mathbf{B}_1, \ldots, \mathbf{B}_n) = \Pr(\mathbf{B}_1) \ldots \Pr(\mathbf{B}_n)$$

2. Dependency and chain rule of conditional probabilities:
   - $\{\mathbf{B}_1, \ldots, \mathbf{B}_n\}$ **dependent** blocks

$$\Pr(\mathbf{B}_1, \ldots, \mathbf{B}_n) = \Pr(\mathbf{B}_1)\Pr(\mathbf{B}_2|\mathbf{B}_1) \ldots \Pr(\mathbf{B}_n|\mathbf{B}_{n-1}, \ldots, \mathbf{B}_1)$$

3. Computing $\Pr(\mathbf{B}_n|\mathbf{B}_{n-1}, \ldots, \mathbf{B}_1)$ is easy on Gaussian distributions (Schur and Cholesky decompositions)

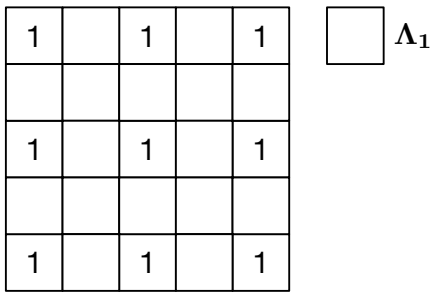
Independent blocks?



$\Rightarrow$ Two non-connected blocks are independent

Embedding at the block level

Decomposition into 4 macro-lattices $\{\mathbf{\Lambda}_1, \mathbf{\Lambda}_2, \mathbf{\Lambda}_3, \mathbf{\Lambda}_4\}$

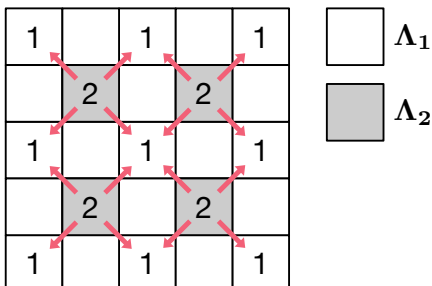


$\mathbf{\Lambda_1}$

$$P\left(\mathbf{s}_d^1\right) = P\left(\mathbf{s}_{\Lambda_1}\right),$$

Embedding at the block level

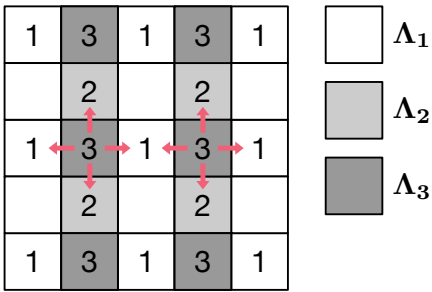
Decomposition into 4 macro-lattices $\{\mathbf{\Lambda}_1, \mathbf{\Lambda}_2, \mathbf{\Lambda}_3, \mathbf{\Lambda}_4\}$



$$P\left(\mathbf{s}_d^2\right) = P\left(\mathbf{s}_{\Lambda_2} | \mathbf{s}_{\Lambda_1}\right),$$

Embedding at the block level

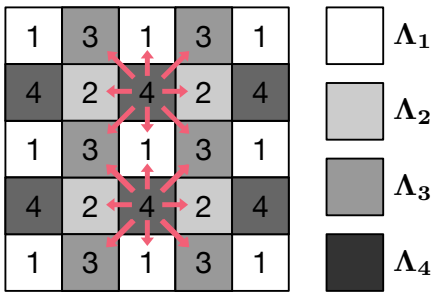
Decomposition into 4 macro-lattices $\{\mathbf{\Lambda}_1, \mathbf{\Lambda}_2, \mathbf{\Lambda}_3, \mathbf{\Lambda}_4\}$



$$P\left(\mathbf{s}_d^3\right) = P\left(\mathbf{s}_{\Lambda_3} | \mathbf{s}_{\Lambda_2}, \mathbf{s}_{\Lambda_1}\right),$$

Embedding at the block level

Decomposition into 4 macro-lattices $\{\mathbf{\Lambda}_1, \mathbf{\Lambda}_2, \mathbf{\Lambda}_3, \mathbf{\Lambda}_4\}$



$$P\left(\mathbf{s}_d^4\right) = P\left(\mathbf{s}_{\Lambda_4} | \mathbf{s}_{\Lambda_3}, \mathbf{s}_{\Lambda_2}, \mathbf{s}_{\Lambda_1}\right).$$

Compliant with the chain rule:

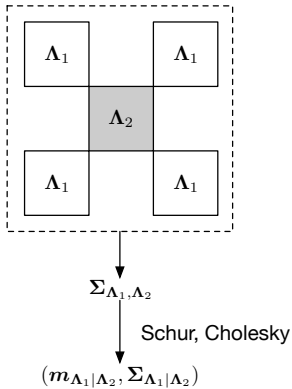
$$\begin{aligned} P\left(\mathbf{s}_d\right) &= P\left(\mathbf{s}_{\Lambda_1}, \mathbf{s}_{\Lambda_2}, \mathbf{s}_{\Lambda_3}, \mathbf{s}_{\Lambda_4}\right), \\ &= P\left(\mathbf{s}_{\Lambda_1}\right) P\left(\mathbf{s}_{\Lambda_2} | \mathbf{s}_{\Lambda_1}\right) P\left(\mathbf{s}_{\Lambda_3} | \mathbf{s}_{\Lambda_1}, \mathbf{s}_{\Lambda_2}\right) P\left(\mathbf{s}_{\Lambda_4} | \mathbf{s}_{\Lambda_1}, \mathbf{s}_{\Lambda_2}, \mathbf{s}_{\Lambda_3}\right). \end{aligned}$$

Embedding at the coefficient level

Example $(\Lambda_2 \mid \Lambda_1)$:

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

1. Compute the conditional matrix

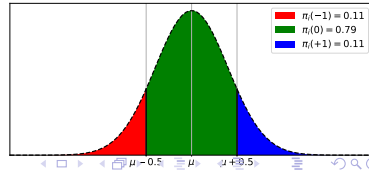


Sample/Embed within each block

$$\begin{cases} s_0 = m_0 + L(0,0) \cdot n_0 \\ s_{1|0} = \underbrace{m_1 + L(1,0) \cdot n_0}_{m_{1|0}} + \underbrace{L(1,1)}_{\sigma^2_{1|0}} \cdot n_1 \\ \vdots \end{cases}$$

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**)

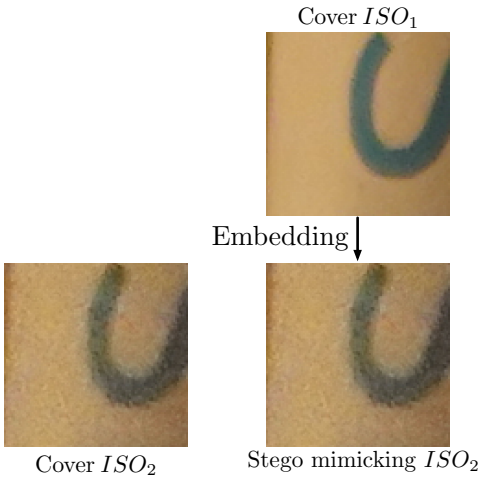
Algorithm 1 J-Cov-NS

- **Inputs**: Cover RAW $\mathbf{X}_p$, message, key, developed Cover $\mathbf{X}_d$ (DCT domain) and JPEG $\mathbf{X}_J$;
- **For** each sub-lattice $\mathbf{\Lambda}_i \in \{\mathbf{\Lambda}_1, \mathbf{\Lambda}_2, \mathbf{\Lambda}_3, \mathbf{\Lambda}_4\}$ and **For** each DCT block **do**:
  - Compute the covariance matrix $\mathbf{\Sigma}_d$
  - Compute $\mathbf{m}|\mathbf{\Lambda}_{i-1} \dots \mathbf{\Lambda}_1$ and $\mathbf{\Sigma}|\mathbf{\Lambda}_{i-1} \dots \mathbf{\Lambda}_1$
    - **For** each JPEG coefficient of $\mathbf{X}_J$ **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 $\mathbf{Y}$.

Embedding on 4×64 lattices

Benchmark: steganalysis

Dedicated setup:



Setup (E1Base), $ISO_1 = 100$, $ISO_2 = 200$:

- 10000 covers ISO_2
- 10000 stegos $ISO_{1\to2}$

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

$$P_E = min(\frac{P_{FA} + P_{MD}}{2})$$

Empirical security for NS [Taburet2020]

P_E (%) / QF	SI-Uniward 1 bpnzAC	H (bpnzAC)	**J-Cov-NS**	Intra-block only
100	0.0	**2.0**	**42.9**	0.0
95	0.4	**2.2**	**41.2**	0.2
85	12.3	**2.4**	**41.2**	15.8
75	24.8	**7.0**	**41.6**	25.2

Linear Classifier with DCTR

P_E (%) / QF	**J-Cov-NS**
100	37.4
95	31.2
75	35.0

SRNet

QF / P_E in %	$K = 1$	$K = 2$	$K = 3$	$K = 5$
100	1.0	12.9	28.7	**40.4**
95	3.5	23.6	**39.3**	**40.9**
85	**39.8**	**39.8**	**39.8**	**41.8**
75	**40.4**	**40.4**	**40.4**	**41.2**

Q-arry embedding

Strategy 2: Steganography distorting a statistical model [Giboulot2020]

Minimize the **multivariate** deflexion

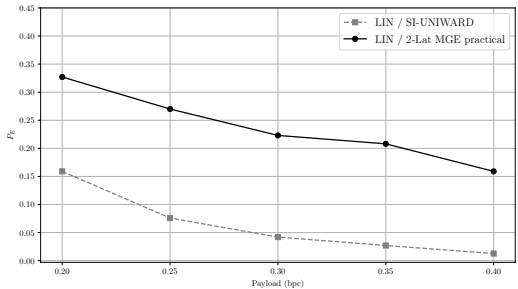
- Sensor noise distributed as $\mathcal{N}(0, \mathbf{\Sigma}_d)$
- Additive stego-signal minimizing the D_{KL} for a given payload size distributed as $\mathcal{N}(0, \alpha\mathbf{\Sigma}_d)$

Embedding mechanism:

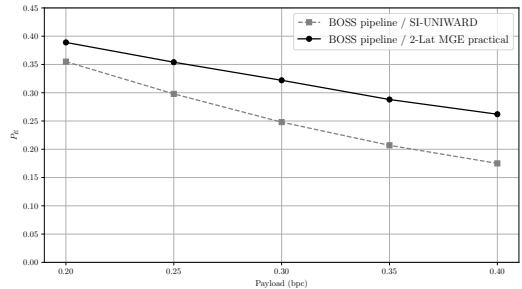
- Very similar to NS (but only 2×64 lattices, diagonal correlations are negligible)
- Can be applied on any (estimated) pipeline

Strategy 2: Steganography distorting a statistical model [Giboulot2020]

Results, Effcicient-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

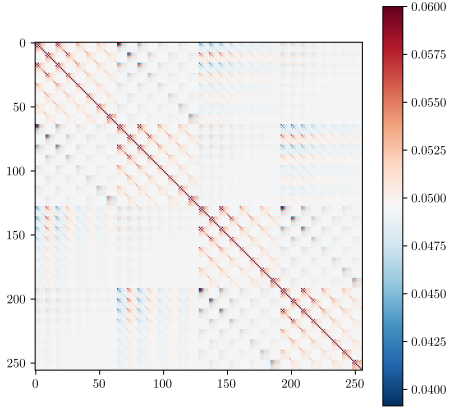
Part 4: Conclusions

Conclusions

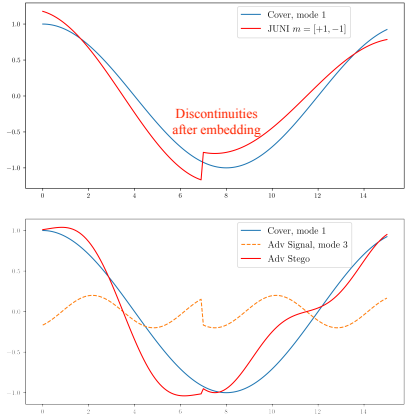
- Correlations matters (especially for high QF or in the spatial/color domain)
- Correlations are due to demosaicking, resizing, denoising, DCT transform, ...
- Joint probabilities matters (correlations within a block and between a block)
- The development pipeline matters
- Assumptions matters (doesn't 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

(Zhang2016) : W. Zhang, Z. Zhang, L. Zhang, H. Li, and N. Yu. Decomposing joint distortion for adaptive steganography, IEEE TCSVT

(Wang2019) : Y. Wang, W. Zhang, W. Li, X. Yu, and N. Yu. Non-additive cost functions for color image steganography based on inter-channel correlations and differences, 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

(Denemark2018) : T. Denemark, P. Bas, and J. Fridrich. Natural Steganography in JPEG Compressed Images,IE

(Taburet2019) : T. Taburet, P. Bas, J. Fridrich, and W. Sawaya. Computing Dependencies between DCT Coefficients for NS in JPEG Domain, IH-MMSec

(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 steganog- raphy. TIFS