Optimizing Pixel Predictors for Steganalysis

Vojtěch Holub and Jessica Fridrich

Dept. of Electrical and Computer Engineering
SUNY Binghamton, New York

IS&T / SPIE 2012, San Francisco, CA
Steganography

- **The art of secret communication**

  $$E_{\text{mb}}(X, m, k)$$

  cover $X$ → key $k$

  channel with passive warden

  $$E_{\text{xting}}(Y, k)$$

  stego $Y$ → key $k$

  message $m$

  - **Steganography by cover modification**
    - $X$ is slightly modified to $Y$ to convey a secret message (by flipping LSBs, changing DCT coefficients, ...). Goal: make the embedding changes statistically undetectable.

  - **Steganalysis**
    - Warden’s job: tell whether a cover or stego object is sent.
Pixel Predictor

Warden represents images by features computed from noise residuals and builds the detector as a classifier in the feature space.

Noise residual

- Narrower dynamic range than $x_{ij}$
- Increased SNR

Predictor

- Estimates the value of pixel $x_{ij}$ from its neighborhood
- E.g., by fitting linear or quadratic polynomials, etc.

$$r_{ij} = x_{ij} - \text{Pred}(x_{ij})$$
Detection Framework

1. **Computing residual:** \( r_{ij} = x_{ij} - \text{Pred}(x_{ij}) \)

2. **Quantization and truncation:** \( r_{ij} \leftarrow \text{round} \left( \text{trunc}_T \left( \frac{r_{ij}}{q} \right) \right) \), \( q \in \mathbb{R}, T = 2 \). Thus, \( r_{ij} \in \{-2, 1, 0, 1, 2\} \)

3. **Forming 4D co-occurrence matrix:** \( C = C^{(h)} + C^{(v)} \)

\[
C^{(h)}_{d_1d_2d_3d_4} = \{ \#(i,j) | r_{ij} = d_1, r_{ij+1} = d_2, r_{ij+2} = d_3, r_{ij+3} = d_4 \}
\]

\[
\text{dim}(C) = 5^4 = 625
\]

4. **Symmetrization of C – Dim. reduction 625 \(\rightarrow\) 169**
   - **Sign-symmetry:** \( C_{d_1d_2d_3d_4} \leftarrow C_{d_1d_2d_3d_4} + C_{-d_1-d_2-d_3-d_4} \)
   - **Directional symmetry:** \( C_{d_1d_2d_3d_4} \leftarrow C_{d_1d_2d_3d_4} + C_{d_4d_3d_2d_1} \)

5. **Ensemble classifier [Kodovský-2011]**
Each predictor will be parametrized, for instance

\[ x_{ij} \]

\[ \Rightarrow K = \begin{pmatrix}
0 & d & c & d & 0 \\
0 & d & b & a & b & d \\
0 & c & a & 0 & a & c \\
0 & d & b & a & b & d \\
0 & d & c & d & 0 
\end{pmatrix} \]

- **Parameters** \(a, b, c, d\)
- **Sum over all elements must equal to** \(1\)
- **Free parameters** \(b, c, d\) since \(a\) can be computed from the rest
Optimization Methodology

**Optimized parameters**
- Free parameters of the predictor structure
- Quantization step \( q \)

**Objective function**
- \( L2R\_L2LOSS \) (margin width of linear SVM) proposed by [Filler-2011] – Problematic
- \( P_E = \min_{P_{FA}} \frac{1}{2} (P_{FA} + P_{MD}(P_{FA})) \) calculated using ensemble classifier on a subset of 2000 images.

**Optimization method**
Nelder-Mead – Derivative-free simplex-reflection algorithm

\[
K = \begin{pmatrix}
  b & a & b \\
  a & 0 & a \\
  b & a & b
\end{pmatrix}
\]
Cover Sources

Three image databases

- **BOSSbase ver. 0.92 [BOSS-2010]** – 9074 images, grayscale, 7 cameras, resized to $512 \times 512$
- **NRCS512** – 6644 images, grayscale, NRCS scans, two $512 \times 512$ cropped from the center of every image
- **LEICA512** – 8626 images, grayscale, Leica M9, 18 M pixels, two $512 \times 512$ cropped from the center of every image
Three stego algorithms
- HUGO (Highly Undetectable steGO) [Pevný et al.-2010]
- EA (Edge-Adaptive) [Luo et al.-2010]
- \( \pm 1 \) embedding with optimal ternary coder

Two payloads
- 0.1 bits per pixel (bpp)
- 0.4 bits per pixel (bpp)
Optimizing the $3 \times 3$ Predictor

We optimized symmetric $3 \times 3$ predictors with structure

$$
\begin{pmatrix}
  b & a & b \\
  a & 0 & a \\
  b & a & b \\
\end{pmatrix}
$$

Predictor parameters: $(a, b), q$

($b = \text{free parameter, } q = \text{quantization step}$)

Initial predictor parameters for optimization:

- Optimal cover predictor in the LSE sense
- $q = 1.5$
Reference predictors

- Predictor derived by [Böhme&Ker-2008]:

\[
KB = \begin{pmatrix}
-0.25 & 0.5 & -0.25 \\
0.5 & 0 & 0.5 \\
-0.25 & 0.5 & -0.25
\end{pmatrix}
\]

- Optimal $3 \times 3$ cover predictor in the LSE sense (LSE)
- Quantization $q$ selected as best $q \in \{1, 1.25, 1.5, 1.75, 2\}$
## Optimization Results – RAW

<table>
<thead>
<tr>
<th>Alg.</th>
<th>Pld.</th>
<th>Ker</th>
<th>BOSSbase $(a, b), q$</th>
<th>$P_E$</th>
<th>NRCS512 $(a, b), q$</th>
<th>$P_E$</th>
<th>LEICA512 $(a, b), q$</th>
<th>$P_E$</th>
</tr>
</thead>
<tbody>
<tr>
<td>HUGO</td>
<td>0.1</td>
<td>KB</td>
<td>(0.50, -0.25), 1.00</td>
<td>43.90</td>
<td>(0.50, -0.25), 2.00</td>
<td>48.62</td>
<td>(0.50, -0.25), 1.75</td>
<td>38.13</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LSE</td>
<td>(0.45, -0.20), 2.00</td>
<td>44.31</td>
<td>(0.51, -0.26), 1.75</td>
<td>48.90</td>
<td>(0.48, -0.23), 1.50</td>
<td>38.43</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Opt</td>
<td>(0.49, -0.24), 2.00</td>
<td>43.78</td>
<td>(0.60, -0.35), 1.69</td>
<td>48.86</td>
<td>(0.57, -0.32), 1.52</td>
<td>36.54</td>
</tr>
<tr>
<td></td>
<td>0.4</td>
<td>KB</td>
<td>(0.50, -0.25), 1.00</td>
<td>26.37</td>
<td>(0.50, -0.25), 1.00</td>
<td>43.95</td>
<td>(0.50, -0.25), 1.75</td>
<td>13.58</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LSE</td>
<td>(0.45, -0.20), 1.50</td>
<td>27.65</td>
<td>(0.51, -0.26), 2.00</td>
<td>43.91</td>
<td>(0.48, -0.23), 1.50</td>
<td>13.35</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Opt</td>
<td>(0.51, -0.26), 1.58</td>
<td>26.49</td>
<td>(0.37, -0.12), 2.37</td>
<td>43.50</td>
<td>(0.38, -0.13), 1.98</td>
<td>12.07</td>
</tr>
<tr>
<td>EA</td>
<td>0.1</td>
<td>KB</td>
<td>(0.50, -0.25), 2.00</td>
<td>37.85</td>
<td>(0.50, -0.25), 2.00</td>
<td>47.66</td>
<td>(0.50, -0.25), 2.00</td>
<td>24.77</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LSE</td>
<td>(0.45, -0.20), 2.00</td>
<td>35.64</td>
<td>(0.51, -0.26), 1.75</td>
<td>47.66</td>
<td>(0.48, -0.23), 2.00</td>
<td>23.94</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Opt</td>
<td>(0.46, -0.21), 1.91</td>
<td>35.42</td>
<td>(0.67, -0.42), 1.84</td>
<td>47.36</td>
<td>(0.37, -0.12), 2.34</td>
<td>17.96</td>
</tr>
<tr>
<td></td>
<td>0.4</td>
<td>KB</td>
<td>(0.50, -0.25), 1.75</td>
<td>17.93</td>
<td>(0.50, -0.25), 1.00</td>
<td>39.56</td>
<td>(0.50, -0.25), 1.75</td>
<td>4.62</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LSE</td>
<td>(0.45, -0.20), 1.75</td>
<td>16.00</td>
<td>(0.51, -0.26), 1.50</td>
<td>39.48</td>
<td>(0.48, -0.23), 2.00</td>
<td>4.30</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Opt</td>
<td>(0.26, -0.01), 1.92</td>
<td>13.74</td>
<td>(0.39, -0.14), 1.58</td>
<td>37.06</td>
<td>(0.40, -0.15), 2.09</td>
<td>3.52</td>
</tr>
<tr>
<td>±1</td>
<td>0.1</td>
<td>KB</td>
<td>(0.50, -0.25), 1.00</td>
<td>31.05</td>
<td>(0.50, -0.25), 1.00</td>
<td>47.82</td>
<td>(0.50, -0.25), 1.00</td>
<td>36.89</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LSE</td>
<td>(0.45, -0.20), 1.00</td>
<td>32.56</td>
<td>(0.51, -0.26), 1.50</td>
<td>48.54</td>
<td>(0.48, -0.23), 1.50</td>
<td>38.19</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Opt</td>
<td>(0.55, -0.30), 0.58</td>
<td>31.42</td>
<td>(0.67, -0.42), 0.72</td>
<td>47.41</td>
<td>(0.56, -0.31), 0.93</td>
<td>37.11</td>
</tr>
<tr>
<td></td>
<td>0.4</td>
<td>KB</td>
<td>(0.50, -0.25), 1.00</td>
<td>12.50</td>
<td>(0.50, -0.25), 1.00</td>
<td>40.52</td>
<td>(0.50, -0.25), 1.00</td>
<td>10.49</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LSE</td>
<td>(0.45, -0.20), 1.00</td>
<td>13.66</td>
<td>(0.51, -0.26), 1.00</td>
<td>41.99</td>
<td>(0.48, -0.23), 1.50</td>
<td>11.09</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Opt</td>
<td>(0.52, -0.27), 1.03</td>
<td>12.48</td>
<td>(0.73, -0.48), 0.55</td>
<td>39.70</td>
<td>(0.32, -0.07), 1.27</td>
<td>8.28</td>
</tr>
</tbody>
</table>
Interpretation of EA Results (1/2)

**EA, BOSSbase, payload 0.4 bpp**

\[ \text{KB} = \begin{pmatrix} -0.25 & 0.5 & -0.25 \\ 0.5 & 0 & 0.5 \\ -0.25 & 0.5 & -0.25 \end{pmatrix} \rightarrow P_E = 17.93\% \]

\[ \text{Opt} = \begin{pmatrix} -0.01 & 0.26 & -0.01 \\ 0.26 & 0 & 0.26 \\ -0.01 & 0.26 & -0.01 \end{pmatrix} \rightarrow P_E = 13.74\% \]

**Why?**

Message is embedded only to horizontal/vertical pixel pairs depending only on their value difference.

\[ \implies \text{Adding diagonal neighbors does not improve steganalysis.} \]
Interpretation of EA Results (2/2)

**EA algorithm**

- Image is divided into square blocks of a randomly selected size $B \times B$, $B \in \{1, 4, 8, 12\}$
- Every block is randomly rotated by $d$ degrees, $d \in \{0, 90, 180, 270\}$
- Embedding into two horizontally neighboring pixels $(x_{i,j}, x_{i,j+1}), i$ odd, where $x_{i,j} - x_{i,j+1} > T$. At most one value from the pair is modified.
- Blocks are rotated back to their original direction.

![Diagram showing rotation of blocks](image)

$B = 4$

**rot. 180°**  
**rot. 90°**
Interpretation of LEICA512 Results

±1, LEICA512, payload 0.4 bpp

\[ S = \begin{pmatrix} b & a & b \\ a & 0 & a \\ b & a & b \end{pmatrix} \]

\[ \text{KB} = \begin{pmatrix} -0.25 & 0.5 & -0.25 \\ 0.5 & 0 & 0.5 \\ -0.25 & 0.5 & -0.25 \end{pmatrix} \rightarrow P_E = 10.49\% \]

\[ \text{Opt} = \begin{pmatrix} -0.07 & 0.32 & -0.07 \\ 0.32 & 0 & 0.32 \\ -0.07 & 0.32 & -0.07 \end{pmatrix} \rightarrow P_E = 8.28\% \]

LEICA512 images are 512 × 512 crops of 18 Mpix originals

⇒ Strong dependencies among neighboring pixels

⇒⇒ [Böhme-2008] recommends optimal LSE predictors for steganalysis satisfying \( \left| \frac{a}{b} \right| = \frac{1}{2\rho} \), where \( \rho \) is the correlation among neighboring pixels.

⇒⇒ In contrast, our study suggests that \( \left| \frac{a}{b} \right| \) should increase
JPEG Results

- RAW images compressed to 80% quality JPEG, then decompressed.
- Predictor optimization did not improve performance, why?

KB – BOSSbase

<table>
<thead>
<tr>
<th>Payload (bpp)</th>
<th>RAW</th>
<th>JPEG</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>1.1</td>
<td>1.5</td>
</tr>
<tr>
<td>0.4</td>
<td>0.7</td>
<td>0.1</td>
</tr>
<tr>
<td>0.1</td>
<td>1.5</td>
<td>0.1</td>
</tr>
<tr>
<td>1.12</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6.25</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Change Rate (%)

- 0.1%
- 0.4%
- 0.1%
- 1.2%
- 0.4%
- 0.1%
JPEG Results

- RAW images compressed to 80% quality JPEG, then decompressed.
- Predictor optimization did not improve performance, why?

KB – NRCS512

<table>
<thead>
<tr>
<th>Payload (bpp)</th>
<th>Change Rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>1.6</td>
</tr>
<tr>
<td>0.4</td>
<td>0.6</td>
</tr>
<tr>
<td>0.1</td>
<td>0.8</td>
</tr>
<tr>
<td>0.4</td>
<td>0.6</td>
</tr>
</tbody>
</table>

$P_E$ (%)
JPEG Results

- RAW images compressed to 80% quality JPEG, then decompressed.
- Predictor optimization did not improve performance, why?

KB – LEICA512

<table>
<thead>
<tr>
<th>Payload (bpp)</th>
<th>Change Rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>0.8</td>
</tr>
<tr>
<td>0.4</td>
<td>0.03</td>
</tr>
<tr>
<td>0.1</td>
<td>0.2</td>
</tr>
<tr>
<td>0.4</td>
<td>0.05</td>
</tr>
<tr>
<td>0.1</td>
<td>0.2</td>
</tr>
<tr>
<td>1.12</td>
<td>0.01</td>
</tr>
<tr>
<td>0.4</td>
<td>0.01</td>
</tr>
</tbody>
</table>

Holub, Fridrich

Optimizing Pixel Predictors for Steganalysis
Interpretation of JPEG Results

1. JPEG compression nearly empties some co-occurrence bins.

2. Embedding repopulates them from neighboring bins.

Example: ±1 embedding, BOSSbase 80, payload 0.4 bpp

<table>
<thead>
<tr>
<th>k–best</th>
<th>bin</th>
<th>avg. RAW bin</th>
<th>avg. JPEG bin</th>
<th>JPEG $P_E$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Cover</td>
<td>Stego</td>
<td>Cover</td>
</tr>
<tr>
<td>1.</td>
<td>(1,−1,2,−1),...</td>
<td>5889</td>
<td>7100</td>
<td>1407</td>
</tr>
<tr>
<td>2.</td>
<td>(1,1,0,0),...</td>
<td>3492</td>
<td>3481</td>
<td>5774</td>
</tr>
<tr>
<td>3.</td>
<td>(2,0,0,0),...</td>
<td>2644</td>
<td>2786</td>
<td>5874</td>
</tr>
</tbody>
</table>

Detection exploits a cover-source singularity rather than effects of embedding.
## Conditional optimization

- Predictor optimization with respect to already existing predictors – cascading

**Example 1:** HUGO, BOSSbase, 0.4 bpp

<table>
<thead>
<tr>
<th>Structure</th>
<th>Optimized predictor, ( q )</th>
<th>( P_{\text{indiv}}^E )</th>
<th>( P_{\text{merged}}^E )</th>
<th>Dim</th>
</tr>
</thead>
<tbody>
<tr>
<td>((a \ 0 \ a))</td>
<td>((0.5 \ 0 \ 0.5)), 1.95</td>
<td>28.76</td>
<td>28.76</td>
<td>169</td>
</tr>
<tr>
<td>((a \ 0 \ b))</td>
<td>((0.048 \ 0 \ -0.952)), 0.93</td>
<td>30.04</td>
<td>25.09</td>
<td>338</td>
</tr>
</tbody>
</table>

The second-order difference is optimally supplemented by the first-order difference
Conditional optimization

- Predictor optimization with respect to already existing predictors – cascading

**Example 2:** HUGO, BOSSbase, 0.4 bpp

<table>
<thead>
<tr>
<th>Structure</th>
<th>Optimized predictor, $q$</th>
<th>$P_E^{\text{indiv}}$</th>
<th>$P_E^{\text{merged}}$</th>
<th>Dim</th>
</tr>
</thead>
<tbody>
<tr>
<td>(b a b a)</td>
<td>(-0.259 0.509 -0.259)</td>
<td>26.49</td>
<td>26.49</td>
<td>169</td>
</tr>
<tr>
<td>(a 0 a b)</td>
<td>(0.509 0.0 0.509)</td>
<td>(b b a)</td>
<td>(-0.259 0.509 -0.259)</td>
<td>1.58</td>
</tr>
<tr>
<td>(c 0 a c)</td>
<td>(-0.034 0.503 -0.034)</td>
<td>27.22</td>
<td>21.77</td>
<td>338</td>
</tr>
<tr>
<td>(b c b c)</td>
<td>(0.064 0.0 0.064)</td>
<td>32.21</td>
<td>20.25</td>
<td>507</td>
</tr>
</tbody>
</table>

**Result comparable with HUGO BOSS winners only with 507 features**

Holub, Fridrich  
Optimizing Pixel Predictors for Steganalysis
Summary

Predictor optimization for covers is a different problem than for a binary detection (cover/stego) within a framework.

Advantages

- Noticeable improvement for some cover sources (LEICA512) and steganographic algorithms (EA).
- Conditional optimization to improve the performance–dimensionality ratio or to build a rich model.

Limitations

- Optimization only over a small parameter vector (e.g., up to dimension of five) due to noisy objective function.

Other

- Astonishingly accurate detection in decompressed JPEGs (future direction).