Image Forensics from Chroma Subsampling in High-Quality JPEG Images

Benedikt Lorch and Christian Riess
IT Security Infrastructures Lab
Friedrich-Alexander University Erlangen-Nürnberg
July 4, 2019
Forensic cues from JPEG implementations

Camera manufacturers and software engineers fine-tune their JPEG implementation to suit their tastes and needs. [Farid 2016]

JPEG implementation characteristics provide cues for forensic tasks:

- estimate compression history [Bianchi 2012, Pasquini 2019]
- detect manipulations [Luo 2007]
- extract library fingerprint [Agarwal 2017, Bonettini 2018]

Our work: Fingerprint JPEG implementation from chroma subsampling

- Novel artifact arising from chroma subsampling
- Integer rounding lets every second column appear brighter than neighboring columns (chroma wrinkles)
**Chroma subsampling in JPEG**

1. **RGB Image**
2. **Color conversion**
3. **Split into blocks**
4. **Chroma subsampling**
5. **DCT**
6. **Quantization**

**Chroma subsampling variants**

- **Simple subsampling**
  - 16x16 chroma pixel block
  - Spatial subsampling to 8x8 block
  - 8x8 DCT
  - 8x8 DCT coefficients

- **DCT subsampling**
  - 16x16 DCT
  - Retain 8x8 lowest frequencies
  - 8x8 DCT coefficients
Simple vs. DCT subsampling

Simple subsampling

DCT subsampling

DCT difference in Y

DCT difference in Cb

DCT difference in Cr
Artifact origin

*Simple subsampling* as implemented in `jcsample.c`

\[
S_{x,y} = \left\lfloor \frac{I_{2x,2y} + I_{2x+1,2y} + I_{2x,2y+1} + I_{2x+1,2y+1} + \text{bias}}{4} \right\rfloor
\]

\[
\text{bias} = \begin{cases} 
1 & \text{for } x \mod 2 = 0 \\
2 & \text{for } x \mod 2 = 1 
\end{cases}
\]

Input channel \( I \)          Subsampled channel \( S \)
Artifact origin

**Simple subsampling** as implemented in `jcsample.c`

\[
S_{x,y} = \left\lfloor \frac{I_{2x,2y} + I_{2x+1,2y} + I_{2x,2y+1} + I_{2x+1,2y+1} + \text{bias}}{4} \right\rfloor
\]

\[
\text{bias} = \begin{cases} 
1 & \text{for } x \mod 2 = 0 \\
2 & \text{for } x \mod 2 = 1 
\end{cases}
\]

Input channel \( I \)  

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>3</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>3</td>
<td>2</td>
</tr>
</tbody>
</table>

Subsampled channel \( S \)

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Artifact origin

**Simple subsampling** as implemented in `jcsample.c`

\[
S_{x,y} = \left[ \frac{I_{2x,2y} + I_{2x+1,2y} + I_{2x,2y+1} + I_{2x+1,2y+1} + \text{bias}}{4} \right]
\]

\[
\text{bias} = \begin{cases} 
1 & \text{for } x \mod 2 = 0 \\
2 & \text{for } x \mod 2 = 1 
\end{cases}
\]

Input channel \( I \)

Subsampled channel \( S \)

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>3</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>3</td>
<td>2</td>
</tr>
</tbody>
</table>
Artifact origin

*Simple subsampling* as implemented in *jcsample.c*

\[
S_{x,y} = \left\lfloor \frac{l_{2x,2y} + l_{2x+1,2y} + l_{2x,2y+1} + l_{2x+1,2y+1} + \text{bias}}{4} \right\rfloor
\]

\[
\text{bias} = \begin{cases} 
1 & \text{for } x \mod 2 = 0 \\
2 & \text{for } x \mod 2 = 1 
\end{cases}
\]

Input channel \( I \)  
Subsampled channel \( S \)

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

Bias introduces periodic artifact in horizontal direction (chroma wrinkle)
Artifact origin

Simple subsampling as implemented in jcsample.c

\[ S_{x,y} = \left[ \frac{I_{2x,2y} + I_{2x+1,2y} + I_{2x,2y+1} + I_{2x+1,2y+1} + bias}{4} \right] \]

\[ bias = \begin{cases} 
1 & \text{for } x \mod 2 = 0 \\
2 & \text{for } x \mod 2 = 1 
\end{cases} \]

Input channel \( I \)

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>3</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>3</td>
<td>2</td>
</tr>
</tbody>
</table>

Subsampled channel \( S \)

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>

Bias introduces periodic artifact in horizontal direction (chroma wrinkle)
Correlation-based detector

- Zero mean normalized cross-correlation with template in DCT domain
- Average correlations over all blocks
- Linear SVM on average correlation

![Correlation-based detector diagram](image)

- Cb component score
- Cr component score
- Horizontal frequencies
- Vertical frequencies
Detection accuracy: Simple vs. DCT subsampling

Classification with linear SVM on block correlations

Correlation as simple yet effective detector for higher quality factors
Repeated up- and downsampling

Effect of recompression on chroma wrinkles with quality factor 100

<table>
<thead>
<tr>
<th>1\textsuperscript{st} compression</th>
<th>Decompression</th>
<th>2\textsuperscript{nd} compression</th>
<th>Cb correlation</th>
</tr>
</thead>
<tbody>
<tr>
<td>simple</td>
<td>-</td>
<td>-</td>
<td>0.081 ± 0.032</td>
</tr>
<tr>
<td>simple</td>
<td>simple</td>
<td>0.080 ± 0.031</td>
<td></td>
</tr>
<tr>
<td>simple</td>
<td>DCT</td>
<td>0.071 ± 0.028</td>
<td></td>
</tr>
<tr>
<td>DCT</td>
<td>simple</td>
<td>0.130 ± 0.038</td>
<td></td>
</tr>
<tr>
<td>DCT</td>
<td>DCT</td>
<td>0.080 ± 0.031</td>
<td></td>
</tr>
</tbody>
</table>

- Interplay of \textit{DCT up-} and \textit{simple subsampling} increases artifact strength
- Hint about previous compression
Resilience to global post-processing

Correlation scores in Cb channel after applying one of four common post-processing operations

![Graph showing correlation scores after different post-processing operations.](image-url)
Manipulation localization

Splicing
Host: No wrinkles
Donor: Wrinkles

Desynchronized splicing
Host: Wrinkles
Donor: Wrinkles

Content-aware fill
Original: Wrinkles
# Prevalence in software with default configuration

## libjpeg and forks

<table>
<thead>
<tr>
<th>Library</th>
<th>Chroma wrinkles?</th>
<th>Chroma wrinkles?</th>
</tr>
</thead>
<tbody>
<tr>
<td>libjpeg 6b</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>libjpeg 7</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>libjpeg 8 (and a, b, c, d)</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>libjpeg 9 (and a, b, c)</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>libjpeg-turbo 2.0.1</td>
<td>Yes</td>
<td>Yes, for quality factors &lt; 90</td>
</tr>
<tr>
<td>mozjpeg 3.3.1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

## Editing software

<table>
<thead>
<tr>
<th>Software</th>
<th>Chroma wrinkles?</th>
<th>Chroma wrinkles?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adobe Photoshop CC 2019</td>
<td>No for quality levels 7 through 12</td>
<td>No for quality levels 7 through 12</td>
</tr>
<tr>
<td>Gimp 2.8.22</td>
<td>Yes, depends on OS(^1)</td>
<td>Yes, depends on OS(^1)</td>
</tr>
</tbody>
</table>

\(^1\) *libjpeg-turbo* is used or provided by Android, Debian, Ubuntu, among others.
Summary on chroma wrinkles

- Periodic artifact arising from chroma subsampling in *libjpeg-turbo*
- Correlation in DCT domain as simple yet effective detector

Applications

- Fingerprint JPEG compression library
- Detect double compression
- Localize manipulations

Future work

- Surpress scene content to isolate artifact from scene
- More effective detector against higher compression rates
Thank you
References