Synthesis of Color Filter Array Pattern in Digital Images

Matthias Kirchner and Rainer Böhme
{matthias.kirchner,rainer.boehme}@inf.tu-dresden.de

Media Forensics and Security XI

San Jose, CA · 2009/01/20
Digital image forensics and tamper hiding

- variety of different forensic tools can be found in the literature
- existing schemes work well under laboratory conditions

How reliable are forensic results if the presumed counterfeiter is aware of the forensic tools?
Digital image forensics and tamper hiding

- variety of different forensic tools can be found in the literature
- existing schemes work well under laboratory conditions

How reliable are forensic results if the presumed counterfeiter is aware of the forensic tools?

Tamper hiding

- mislead forensic tools such that they produce false negatives
CFA Synthesis
Problem statement

- typical digital cameras use a color filter array (CFA) to capture full color images

- color filter interpolation introduces periodic correlation pattern between neighboring pixels
Problem statement

- Typical digital cameras use a **color filter array** (CFA) to capture full color images.
- CFA pattern has to be restored to conceal traces of manipulation.
- Color filter interpolation introduces a **periodic correlation pattern** between neighboring pixels.
Problem statement

- typical digital cameras use a color filter array (CFA) to capture full color images

- CFA pattern has to be restored to conceal traces of manipulation
  - straight-forward: re-interpolation

- color filter interpolation introduces periodic correlation pattern between neighboring pixels

- overwrites two thirds of all pixels with new (interpolated) values
A minimal distortion approach

Linear model

- CFA interpolation follows a linear equation
  \[ \hat{y} = Hx \]
- Image with incomplete/missing CFA pattern is corrupted by an additive residual \( \epsilon \)
  \[ y = Hx + \epsilon \]

CFA synthesis

- Find a possible sensor signal \( x \) such that
- Least squares solution
  \[ ||\epsilon|| = ||y - \hat{y}|| \rightarrow \min \]
  \[ x = (H'H)^{-1}H'y \]

CFA re-interpolation not from the signal itself, but from a pre-filtered version
Structure of $H$

- For $N$ pixels per channel and $M \leq N/2$ genuine sensor samples, a direct implementation of the LS solution is impossible for typical image sizes.

- Matrix $H$ has dimension $N \times M$.

- Cubic complexity: $x = (H'H)^{-1}H'y$

  - Inversion $O(M^3)$
  - Multiplication $O(M^2N)$
Structure of $H$

- for $N$ pixels per channel and $M \leq N/2$ genuine sensor samples, a direct implementation of the LS solution is impossible for typical image sizes

- matrix $H$ has dimension $N \times M$

- cubic complexity: $x = (H'H)^{-1}H'y$

Efficiency improvements

- matrix $H$ is typically sparse (interpolation kernels have finite support) and has a regular structure (Bayer pattern)
Red Channel
Partitioning $\mathbf{H}$

- columns partition $\mathbf{H}$ into repeating blocks $\mathbf{A}$, $\mathbf{B}$ with $\mathbf{B} = \frac{1}{2} \mathbf{A}$
- $\mathbf{H} = \mathbf{A} \otimes \mathbf{A}$
- $\mathbf{A}$ has only dimension $\sqrt{N} \times \sqrt{N}/2 + 1$
Partitioning $\mathbf{H}$

- Columns partition $\mathbf{H}$ into repeating blocks $\mathbf{A}, \mathbf{B}$ with $\mathbf{B} = \frac{1}{2} \mathbf{A}$
- $\mathbf{H} = \mathbf{A} \otimes \mathbf{A}$
- $\mathbf{A}$ has only dimension $\sqrt{N} \times \sqrt{N}/2 + 1$

**Kronecker tweaks**

\[ \mathbf{x} = (\mathbf{H}'\mathbf{H})^{-1} \mathbf{H}' \mathbf{y} \]

With $\mathbf{H}^\times = \mathbf{H}'\mathbf{H}$:

- $(\mathbf{H}^\times)^{-1} = (\mathbf{A}^\times)^{-1} \otimes (\mathbf{A}^\times)^{-1}$

With $\mathbf{H}^+ = (\mathbf{H}^\times)^{-1} \mathbf{H}'$:

- $\mathbf{H}^+ = \mathbf{A}^+ \otimes \mathbf{A}^+$ (pseudo-inverse)
Analytical inversion $\Phi = (A^\times)^{-1}$

$A^\times = \begin{bmatrix}
\frac{5}{4} & \frac{1}{4} & 0 & 0 \\
\frac{1}{4} & \frac{3}{2} & \frac{1}{4} & 0 \\
\frac{1}{4} & \frac{3}{2} & \frac{1}{4} & 0 \\
\frac{1}{4} & \frac{3}{2} & \frac{1}{4} & 0 \\
0 & 0 & 0 & 0 \\
\end{bmatrix}$

- method by Huang & McColl (1997)
- second order linear recurrences:
  \[ \zeta_i = \frac{3}{2} \zeta_{i-1} - \left(\frac{1}{4}\right)^2 \zeta_{i-2} \]
  \[ \upsilon_j = \frac{3}{2} \upsilon_{j+1} - \left(\frac{1}{4}\right)^2 \upsilon_{j+2} \]
- and ratios:
  \[ \xi_i = \frac{\zeta_i}{\zeta_{i-1}} \quad \text{and} \quad \gamma_i = \frac{\upsilon_i}{\upsilon_{i+1}} \]

$A^\times$ is tridiagonal symmetric
Analytical inversion $\Phi = (A \times)^{-1}$

- method by Huang & McColl (1997)
- second order linear recurrences:
  - $\zeta_i = \frac{3}{2} \zeta_{i-1} - \left(\frac{1}{4}\right)^2 \zeta_{i-2}$
  - $\upsilon_j = \frac{3}{2} \upsilon_{j+1} - \left(\frac{1}{4}\right)^2 \upsilon_{j+2}$
- and ratios:
  - $\xi_i = \frac{\zeta_i}{\zeta_{i-1}}$ and $\gamma_i = \frac{\upsilon_i}{\upsilon_{i+1}}$
- inversion has complexity $O(N/4)$
Red channel approximate solution

Infinite image

- $\Phi$ is asymptotically symmetric

Toeplitz

$$
\Phi_{j,j} \to \Phi_D \\
\Phi_{i,j} \to \left( \frac{-1/4}{q} \right)^{|i-j|} \Phi_D
$$
Red channel approximate solution

Infinite image

- $\Phi$ is asymptotically symmetric Toeplitz

$$\Phi_{j,j} \to \Phi_D$$

$$\Phi_{i,j} \to \left(\frac{-1/4}{q}\right)^{|i-j|} \Phi_D$$

$$< 1$$

- off-diagonal elements decay exponentially

Asymptotic kernel
Green Channel
Green channel in a nutshell

Additional border pixels

- avoid special interpolation kernel for margin pixels

Block structure

- columns partition $H$ into repeating blocks $A_1$, $A_2$, and $B$
- but: no trivial decomposition
Green channel in a nutshell

Additional border pixels

- avoid special interpolation kernel for margin pixels

Block structure

- columns partition $H$ into repeating blocks $A_1$, $A_2$, and $B$
- but: no trivial decomposition

Re-ordering: $\tilde{H} = \begin{bmatrix} I \\ A \end{bmatrix}$
Experimental Results
Evaluation of attacks against digital image forensics should always be benchmarked against (at least) two criteria (Kirchner & Böhme, 2008):

**(Un)detectability**
- state-of-the-art detector can not distinguish between original and synthesized CFA images

**Visual quality**
- higher image quality than naive re-interpolation
Tamper hiding performance measures

Evaluation of attacks against digital image forensics should always be benchmarked against (at least) two criteria (Kirchner & Böhme, 2008):

**(Un)detectability**
- state-of-the-art detector can not distinguish between original and synthesized CFA images
- fast version of Popescu and Farid’s detector (Popescu & Farid, 2005; Kirchner, 2008)

**Visual quality**
- higher image quality than naive re-interpolation
- \[ \triangle \text{PSNR}(y_1, y_2; y_0) = \text{PSNR}(y_1, y_0) - \text{PSNR}(y_2, y_0) \]
Detectability

- periodic p-map and strong high-frequency interpolation peaks
- post-processing destroys CFA pattern
- CFA pattern synthesis re-introduces typical artifacts
Detectability, quantitative results

Histograms from 1000 images

CFA indicator (Popescu & Farid, 2005)
Detectability, quantitative results

Histograms from 1000 images

$9 \times 9$ median

RAW

CFA synthesis

CFA indicator (Popescu & Farid, 2005)
Detectability, quantitative results

Histograms from 1000 images

- 9 × 9 median
- RAW
- CFA synthesis

CFA interpolation not the last in-camera processing step!
### Image quality

#### Quartiles from 1000 images

<table>
<thead>
<tr>
<th>Synthesis gain [dB]</th>
<th>Q_{25}</th>
<th>Q_{50}</th>
<th>Q_{75}</th>
<th>IQR</th>
<th>Q_{25}</th>
<th>Q_{50}</th>
<th>Q_{75}</th>
<th>IQR</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1.07</td>
<td>1.18</td>
<td>1.28</td>
<td>0.21</td>
<td>0.89</td>
<td>0.94</td>
<td>0.99</td>
<td>0.10</td>
</tr>
</tbody>
</table>

LS approach yields better visual quality after re-interpolation.
Image quality
Quartiles from 1000 images

<table>
<thead>
<tr>
<th>synthesis gain [dB]</th>
</tr>
</thead>
<tbody>
<tr>
<td>( Q_{25} )</td>
</tr>
<tr>
<td>1.07</td>
</tr>
</tbody>
</table>

LS approach yields better visual quality after re-interpolation.

Linear filter approximation equivalent to exact solution for reasonable filter dimensions.
Conclusion
Concluding Remarks

Results in a nutshell

- CFA synthesis is important building block for tamper hiding techniques.
- Minimal distortion CFA synthesis can be formulated as least squares problem.
- Special structure allows efficient implementation; near-optimal approximate solution is only of linear complexity.

Further research and limitations

- More sophisticated (and signal-adaptive) interpolation algorithms?
- Discrete optimum?
- CFA interpolation not the last step in the in-camera processing chain!
Thanks for your attention

Questions?

Matthias Kirchner and Rainer Böhme
{matthias.kirchner,rainer.boehme}@inf.tu-dresden.de

The first author gratefully receives a doctorate scholarship from Deutsche Telekom Stiftung, Bonn, Germany.
Red channel explicit solution

\[ x = H^+ y = (A^+ \otimes A^+) y \]

\[ x_i = \sum_{j=1}^{N} \left( A^+_{r,s} \cdot A^+_{u,v} \right) y_j \]

- Elements \( A^+_{i,j} \) can be derived efficiently from \( \Phi = (\mathbf{A} \times)^{-1} \).
Red channel explicit solution

\[ x = H^+ y = (A^+ \otimes A^+) y \]

\[ x_i = \sum_{j=1}^{N} (A^+_{r,s} \cdot A^+_{u,v}) y_j \]

- Elements \( A^+_{i,j} \) can be derived efficiently from \( \Phi = (A \times)^{-1} \).

- Indices \((r, u)\) and \((s, v)\) are the 2D coordinates of pixels \(x_i\) and \(y_j\) in the subsampled genuine image and input image, respectively.
Green channel in a nutshell (cont’d)

Explicit solution

\[ \Phi = (\tilde{H}^\times)^{-1} \]

\[ x = \Phi \tilde{y}_G + \Phi A' \tilde{y}_{CFA} \]

Analytical inversion of \( \tilde{H}^\times \)

\[ \Phi = I - A'(I + AA')^{-1} A \]

\[ I + AA' \text{ is block tridiagonal Toeplitz} \]

Huang & McColl (1997):
second order matrix recurrences

Approximate filter kernel

\[
\begin{pmatrix}
-0.001 & 0.001 & 0.001 & 0.001 & -0.001 \\
0.001 & 0.001 & 0.003 & 0.005 & -0.004 \\
-0.001 & 0.003 & 0.009 & -0.022 & -0.029 \\
0.001 & 0.005 & -0.022 & -0.072 & 0.165 \\
0.001 & -0.004 & -0.029 & 0.165 & 0.835 \\
0.001 & 0.005 & -0.022 & -0.072 & 0.165 \\
-0.001 & 0.003 & 0.009 & -0.022 & -0.029 \\
-0.001 & 0.003 & 0.005 & -0.004 & 0.005 \\
-0.001 & 0.001 & 0.001 & 0.001 & -0.001
\end{pmatrix}
\]