Credible Repair Of Sony Main-Sensor PDAF Striping Artifacts

Henry Dietz

PMII-585, Noon, January 15, 2019

University of Kentucky
Electrical & Computer Engineering
Focus Using An Optical Coincidence Rangefinder

- Two *different* points of view, superimposed by mirrors
- Aligned views $\Rightarrow$ in focus
- Misalignment shows both direction and distance to achieve focus
Phase-Detect AutoFocus (PDAF) Using An SLR

- **SLR** means **Single Lens Reflex**; a mirror deflects lens projection to optical viewfinder
- Two *different* points of view *within one lens* e.g., left vs. right sides (Maxxum 9000, 1985)
- Separate PDAF detector:
  - Main mirror beamsplits to another mirror and optics targeting the PDAF sensor
  - PDAF sensor has pairs of point sensors seeing rays from opposite sides of lens
  - Misalignment gives distance & direction
Mirrorless AutoFocus?

• Mirrorless cameras use electronic live view from the main sensor as viewfinder
• 2010 Sony A33 SLT beamsplitter to PDAF
• Use the main sensor for AF:
  – Contrast Detect AF (CDAF), aka Depth From Focus (DFF)
  – Depth From Defocus (DFD)
  – Dual Pixel AF (DPAF)
  – Masked Pixel AF
Masked Pixel AutoFocus?

- Better view separation than DPAF
- Used in many cameras
- Microlens + mask distinguishes views
- Many variants; shown is from Sony US20110063484A1
Problems With Masked Pixels?

- Masked pixels see less light, which can reduce low-light AF performance.
- Masked pixels don’t see all views thru lens; pixel value needs adjustment or is ignored and interpolated over.
- Light not sensed has to go somewhere; stray light could interfere with pixel values.
Sony’s Bad Choice Of Press Event Lighting, A7III Release

Crop from DPReview photo by Rishi Sanyal, April 2018
We Can Credibly Fix It!

Crop from DPReview photo by Rishi Sanyal, April 2018
In case you couldn’t see...
A Brief Timeline

- **6/2017**, Sony A9 line artifacts noted by Rishi Sanyal, Bill Claff, Jim Kasson, Horshack, etc.; correlation with PDAF lines established
- Similar defects noted in RX100V, Fuji X-T2... these artifacts had been around for years
- **4/2018**, A7III artifact analysis @DPRreview, pippo27 postprocessing, **KARWY-SR repair tool** (Kentucky’s ARW Stripe Removal)
- **12/2018**, RawTherapee 5.5 postprocessing
Characterizing The Problem

- Difficult because it affects many models
- Very difficult because it’s rare...
  had to create problematic circumstances:
  - Lenses with big rear elements, lighting flare
  - No lens at all; flare generated with penlight
  - Lens adapter with built-in flare light
## Characterizing The Problem

<table>
<thead>
<tr>
<th>Model</th>
<th>Stripes</th>
<th>Severity</th>
<th>Common?</th>
</tr>
</thead>
<tbody>
<tr>
<td>RX100IV</td>
<td>–</td>
<td>–</td>
<td>Never (no PDAF)</td>
</tr>
<tr>
<td>RX100V</td>
<td>Bright</td>
<td>Moderate</td>
<td>Yes</td>
</tr>
<tr>
<td>NEX-5</td>
<td>–</td>
<td>–</td>
<td>Never (no PDAF)</td>
</tr>
<tr>
<td>NEX-7</td>
<td>–</td>
<td>–</td>
<td>Never (no PDAF)</td>
</tr>
<tr>
<td>A6000</td>
<td>Bright</td>
<td>Very</td>
<td>Yes</td>
</tr>
<tr>
<td>A6500</td>
<td>Bright</td>
<td>Minor</td>
<td>Yes</td>
</tr>
<tr>
<td>A7</td>
<td>Dark</td>
<td>Minor</td>
<td>No</td>
</tr>
<tr>
<td>A7II</td>
<td>Dark</td>
<td>Moderate</td>
<td>No</td>
</tr>
<tr>
<td>A7RII</td>
<td>Bright</td>
<td>Very</td>
<td>No</td>
</tr>
<tr>
<td>A9</td>
<td>Bright</td>
<td>Very</td>
<td>Yes</td>
</tr>
<tr>
<td>A7RIII</td>
<td>Bright</td>
<td>Very</td>
<td>Somewhat</td>
</tr>
<tr>
<td>A7III</td>
<td>Bright</td>
<td>Very</td>
<td>Yes</td>
</tr>
</tbody>
</table>
KARWY-SR Design

- **Credible repair** of compressed ARW2 raw files
- Leverage algorithms from **KARWY**
- Fast without prior info about PDAF locations
- Run locally in a WWW browser *(JavaScript)*
Welcome to the beta test version 20180318 of the free program KARWY-SR which runs locally in your web browser to perform "Stripe Removal." The stripes we’re talking about are the ones like the above, which run horizontally across the sensor in landscape orientation and are probably related to the P5AF pixels on the sensor. This version only can repair the artifact in compressed Sony ARW2 raw files, and the result is also a compressed ARW2 file. Simply drag & drop ARW2 file(s) below, wait typically less than 30 seconds, and save the “Link to processed file” to a file with a name ending in .arw. The image displayed is just the original JPEG thumbnail, it has not been repaired, but is displayed to make it easier for you to know which file is which when connecting multiple ARW2 files.

We created the original KARWY (pronounced kar we) to credibly repair compression artifacts in Sony ARW2 raw files. However, the original version is compiled C code which requires Make, PNG-Converter, DelTools, Inkscape and zic as helpers, making installation a pain. Thus, we made KARWY available to run remotely on our servers via a Web form interface. That means it does not compress much faster than KARWY-SR, but you have to wait as big raw files are sent to and from the server. We hope to add functionality of the original version to KARWY-SR. But for now, the two tools are separate, and you should use KARWY-SR first and then KARWY to repair a file suffering both types of artifacting.

This program was created by Henry Ong at the University of Kentucky (also known as ProfOng) as a research prototype; you use it at your own risk. The ARW2 decoding logic is derived from donax. The program is written in C, but has been converted into javascript using assemble.js with javascript interface routines from eldorano’s donax.js. Both donax and the javascript interface routines are covered by the GNU General Public License, and full source code is available by request (and will be posted once KARWY-SR is stable).
Credible Repair Algorithm

• Mark artifac ted pixels:
  1. Confirm compressed ARW2 input
  2. Create preview image, extract camera info
  3. Decode ARW2 into linear 16-bit samples with noise bounds & random noise added
  4. Scan to mark “brighter line” sequences (could just check PDAF locations...)
  5. Mark single-pixels between marked pixels
Credible Repair Algorithm

1. Replace marked pixels with statistically-biased value interpolated from +/-2 rows (Nikon Z6/7 banding lacks randomness)
2. In C version, refine using texture synthesis (slow: hundreds of patterns per pixel)
3. Recompress & overwrite changed 16-pixel ARW2 blocks (saves packing operations)
4. Return link to corrected-in-place file

• Typical 24MP ARW2 repair <5s
Most Repairs Are Very Good

A6000 original image crop (left), tighter crop, KARWY-SR repair
RawTherapee 5.5 vs. KARWY-SR Repairs

Original (top left), crop (bottom left), RawTherapee (top right), KARWY-SR
RawTherapee 5.5 vs. KARWY-SR Repairs (Enhanced)

Enhanced original crop (left), RawTherapee (middle), KARWY-SR (right)
Repair Quality Evaluation

- They’re *credible*, right?
- **Zero** user complaints
- DPReview evaluation ⇒
- Repair is **not perfect**:
  - Blonde hair fooled it
  - Some hint of lines is visible in solid areas
- Does not fix dark lines
- Compressed ARW2 only
Conclusions

- Masked pixel PDAF is & will be common for mirrorless, video, and even cell phones
- **Can artifacts be prevented?**
- Credible repair is cheap and effective
- Repair requires matching noise statistics