Credible repair of Sony main-sensor PDAF striping artifacts

Henry Gordon Dietz; Department of Electrical and Computer Engineering, University of Kentucky; Lexington, Kentucky

Abstract

Since the introduction of the Minolta Maxxum 9000 in 1985, PDAF (phase detect automatic focus) has been the standard way to achieve sharply-focused images of fast-moving action, such as professional sports. In a typical SLR (single lens reflex) camera, the image for the optical viewfinder is reflected up by the main mirror, while a secondary mirror and optics copy the image to the PDAF detector. However, such an arrangement is impractical for mirrorless digital cameras. Thus, there have been a variety of methods used to incorporate phase sensing on the main sensor – with various trade-offs. The current work discusses some of these trade-offs and then describes in detail a specific type of striping artifact introduced by the masked pixel structures used in Sony sensors. A computational method for credible repair of this artifact also is presented. The method described is quick and fully automatic; it has been implemented as KARWY-SR, an open source JavaScript version using a drag-and-drop interface to repair the artifact in Sony ARW raw files.

Introduction

Although pinholes can create images without focusing, the problem of focusing is as old as the concept of a lens. A lens with a relatively large aperture creates a much brighter image than a pinhole, but has limited depth of focus (DoF). For a point of light in the scene to be recorded as a point, the film, or image sensor, must be positioned the correct distance behind the lens. This distance depends on how far the subject is in front of the lens, thus it becomes necessary to move optical elements to compensate for changes in the distance to the subject – i.e., to focus.

Among the fastest and most reliable ways to manually focus a lens is an optical coincidence range finder, as shown in Figure 1. Light (blue) passes from the subject directly to the viewer through the beamsplitter (semi-transparent mirror), but also bounces there (yellow) off the mirror and beamsplitter. The angle of the mirror is manually adjusted so that the two images align, thus determining the angle shown in red and the distance to the subject is simply $arctan(Angle) \times Baseline$. Thus, in rangefinder cameras, the focus adjustment is mechanically coupled to the turning of the mirror so when images align, the lens is focussed at the correct distance. Figure 2 shows the misalignment of the direct view and yellow-tinted patch from the rangefinder's reflected path.

In 1985, the Minolta Maxxum 9000 SLR introduced the autofocus approach that became the standard for 135-film and digital single-lens reflex (SLR) cameras: phase detect automatic focus (PDAF). PDAF is essentially a rangefinder, but using a pair of viewpoints extracted from the image projected by the taking lens – the same idea also used by microprism and split image manual focus aids. The PDAF detector is essentially a number of lines

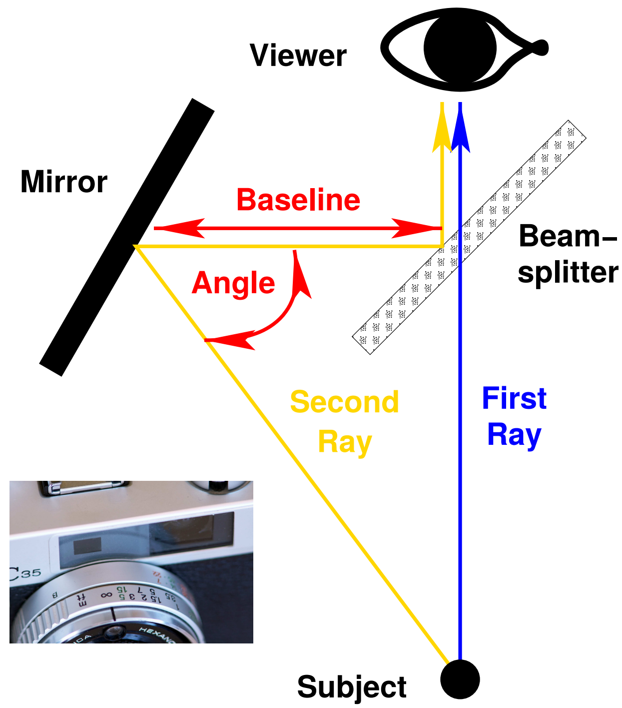

Figure 1. *Optical coincidence range finder, as in Konica C35*

Figure 2. *Rangefinder view through an out-of-focus Konica C35*

each consisting of pairs of point sensors accepting rays from opposite sides of the lens. The direction and distance needed to shift the line patterns from each side to align indicates the direction and relative distance by which the focus must be adjusted. The alignment is done computationally, rather than mechanically, so a focus drive motor controlled by the phase computation is used

Figure 3. *A7III image crop showing stripe artifacts*

Figure 4. *A7III image crop repaired by KARWY-SR*

to adjust the lens focus.

Using viewpoints through the lens ensures that the distance measurement is directed to the desired subject, but significantly complicates the optical path in an SLR. The mirror that diverts the image projected by the lens onto the focus screen in a manual-focus SLR is needed for composing the shot at the same time the PDAF detector needs to do its rangefinding, so it becomes necessary to use a beamsplitter to share the image light between the viewfinder screen and PDAF detector. In fact, the optics needed include not only making the primary mirror a beamsplitter, but adding a secondary mirror behind the primary to direct light downward and then a third mirror and lens system to place the image on the PDAF detector, which is typically in the base of the SLR.

Starting with the Alpha 33 in 2010, Sony greatly simplified the physical construction in their single lens translucent (SLT) cameras by using a single fixed beamsplitter mirror to direct some light to the PDAF detector while letting most light pass to to main sensor. In effect, this allowed placing the separate PDAF detector where the optical viewfinder screen would have been, thus requiring the optical viewfinder to be replaced by live view from the main sensor in an electronic viewfinder.

However, mirrorless cameras do not have a mechanism for routing the image to a separate PDAF sensor. Although focus could be determined by measuring the contrast of the selected focus area on the main sensor, contrast measurements are only meaningful relative to contrast measurements made at other focus settings. In effect, contrast detect autofocus (CDAF) is depth from focus (DFF); systems must "hunt" for the maximum contrast. It is also possible to implement depth from defocus (DFD) by analyzing how contrast changes as focus is changed, and various Panasonic cameras have improved focus speed by implementing this[1].

As fast as DFD can be, PDAF offers the ability to estimate both direction and distance to focus in a single readout operation with relatively simple processing. So the question becomes how to integrate PDAF pixels in the main image sensor.

Masked-Pixel Main Sensor PDAF

In a typical digital camera, each pixel in the main sensor is covered by a microlens. Originally, the purpose of the microlens was to direct light from the entire pixel area into the light-sensitive region of the photodiode. However, like any lens, a microlens has the property that rays coming from different angles can be distinguished – sent to different areas within the pixel. Canon's dual pixel autofocus (DPAF)[2] places two photodiodes (a dual pixel) under every microlens: one sensitive to rays from the left, the other right. In that way, every pixel can be used for phase detection or left/right pairs can be summed to produce an image. Alternatively, by fabricating an opaque or reflective metal mask under the appropriate portion of a microlens[4], the photodiode can be made to sense only rays coming from relatively precisely determined angles – thus, masked pixels can perform better for autofocus than dual pixels. Unfortunately, the light not used is reflected away and a phase detect pixel does not accurately represent the full pixel value for forming an image. Variations on this type of masked-pixel approach are what Sony uses in their PDAF-capable sensors for mirrorless cameras.

There are several different mechanisms by which masked pixels can cause artifacts in captured images:

1. Masked pixels generally are lit by less light than unmasked ones, implying some loss of dynamic range for phase detect pixels. This might reduce autofocus performance in poor lighting.
2. Because each masked pixel does not sample all the light aimed at its position, it has a value which cannot be directly used as the value of the image pixel. This makes interpolation or other adjustments necessary to determine a value for the corresponding image pixel.
3. The light not sensed still has to go somewhere. This "stray" light could cause artifacts by interfering with nearby pixels.

In 2011, when main-sensor PDAF first started to appear in mirrorless cameras such as Nikon's 1 system, the expectation was that the second issue – interpolation over PDAF pixels – would cause visible artifacts, but this was quickly disproven. The reason is simply that the number of image pixel locations sacrificed to PDAF is small, not qualitatively worse than the number lost to

Figure 5. *RX100 V image with stripe artifacts, enlargement, and repair*

chip defects or dust removal. The most serious image quality issues are actually rooted in the handling of stray light.

Sony's PDAF Striping Artifact

As main-sensor PDAF became more common, there were isolated reports of strange line artifacts appearing in some images. When the Sony A9 camera was released in June 2017, severe line artifacts were observed in some images converted to black and white using extreme red filtering, and this drew the attention of staff member Rishi Sanyal and various technical forum members at Digital Photography Review (DPReview.com), especially Bill Claff, Jim Kasson, and Horshack. These forum members confirmed that the line artifacts did correspond to the rows of PDAF masked pixels, and it was noted that the Sony RX100 V and Fujifilm X-T2 both suffered similar artifacts (although Fuji's X-Trans sensor produces more of a weave pattern). These patterns were generally assumed to result from the interpolation over the green and blue pixels sacrificed for PDAF.

In April 2018, DPReview posted a gallery of images taken at the press event releasing the Sony A7 III. The unusual lighting in Sony's release event resulted in some of the photos containing a number of bright line artifacts that came to be known as "PDAF striping." A crop from one of these images is shown in Figure 3. This was not merely an interpolation artifact, but apparently the result of the third potential problem described above: stray light reflecting off the PDAF masked pixels.

The goal of the work reported in this paper was to attempt to understand the cause of the striping artifact, characterize it, and effect a credible, low computational cost, repair. This was accomplished, although the precise cause has not yet been confirmed by Sony.

Characterizing The Defect

To begin, the striping artifact was induced and characterized on a variety of cameras – including multiple different models using Sony sensors. In all cases where stripes were found, the stripes are actually dotted/dashed, not continuous lines. The artifact was surprisingly difficult to induce, requiring strong lighting at very specific angles. Those angles were unobtainable with some lenses. Moving around bright light sources held at sharp angles to the open lens mount proved to be the most reliable way to trigger the artifacts until a 3D-printed adapter was built that contained a LED light positioned to cause the artifacts. The artifacts were also found to be dependent on the color of the light; purplish (magenta) flare was present in nearly all examples.

Clearly, Sony has used a variety of somewhat different masking structures in different camera models. However, many aspects are the same in all models that have main-sensor PDAF. For example, the stripes are always in the long dimension of the sensor; the vertical stripes in Figure 3, the first example, were because the camera was held in the portrait orientation. (Note that this also explains why PDAF performance can be very different if the camera is held in portrait orientation instead on landscape.) The stripes are not evenly spaced, but follow a repeating pattern. Depending on sensor model, the stripes arise from either reflections or shading caused by light hitting the masked pixels at unusual angles.

The following briefly summarizes the striping artifacts for various Sony models tested:

Sony RX100 IV: This 1" sensor does not have PDAF pixels and no stripe artifacts were observed.

Sony RX100 V: Moderately bright stripes were observed and they were easy to induce. This sensor differs from that in the RX100 IV primarily in the addition of 315 phase detection points; the non-removable lens is identical. Figure 5 shows an example from the RX100 V. The affected region, the upper right, is shown in the contrast-enhanced enlarged crops to the right, first the original and then the KARWY-SR repaired version.

Sony NEX-5: No stripe artifacts were observed. This was expected because the 14MP APS-C sensor supports only CDAF, not PDAF.

Sony NEX-7: No stripe artifacts were observed. Although this APS-C sensor holds 24MP, like the NEX-5, it does not have any PDAF pixels. This camera was tested particularly thoroughly, including using all the lighting and lenses that produced artifacts on the other cameras tested.

Sony A6000: Very bright stripes were easy to induce. This 24MP APS-C sensor differs from the one in the NEX-7 primarily in having a hybrid AF system with 25 contrast-detect and 179 phase-detect points.

Sony A6500: Relatively dim bright striping was easy to induce. The 24MP APS-C sensor has a hybrid AF system with 169 contrast-detect and 425 phase-detect points.

Sony A7: Very slightly dark stripes were very difficult to induce; only red pixels were darker. This is a 24MP full-frame sensor with 25 contrast-detect and 117 phase-detect points.

Sony A7 II: Moderately dark stripes were very difficult to induce; both red and green pixels were darker. This is a 24MP full-frame sensor with 25 contrast-detect and 117 phase-detect points, very similar to the sensor in the A7, but apparently not the same.

Sony A7R II: Very bright stripes were observed, but it was very difficult to induce them. The 42MP full-frame sensor contains 25 contrast-detect and 399 phase-detect points.

Sony A9, Sony A7R III, and Sony A7 III: Although not personally tested, posted examples show very bright stripes that were easy to induce on the A9 and A7 III. It seems Sony has been making fewer changes to the masking structure in their most recent models.

Credible Repair

Given the above understanding of the phenomena, there is a clear need for software that can repair these artifacts. They may be rare, but it is not acceptable to have a critical image ruined by artifacts that are difficult to see using the rear LCD to check captures in the field. We say our processing implements *credible repair* because there is no way to recover the actual data lost by PDAF masking; we are merely synthesizing image data that is statistically more consistent with the un-artifacted image data.

Because dark stripes were rarer and less severe, we decided to repair only bright striping. It was further decided not to rely on prior knowledge of where the PDAF pixels are. Sony has varied that parameter across models, so any repair tool depending on that information would need a regularly-updated table of the location data for each model. Certainly, we recommend that if Sony incorporates a fix in their camera firmware, it should take advantage of the row information to speed the processing and minimize potential for introduction of repair artifacts. In fact, DPReview forum member pippo27 produced that type of fix and integrated it in RawTherapee, which in version 5.5[7] can apply a similar repair as part of the demosaicing process for certain Sony cameras and the Nikon Z6 and Z7. These new Nikon cameras apparently also use Sony sensors, but Nikon seems to have implemented an overly aggressive repair of the raw data in-camera, creating smeared line patterns.

Directly Editing Compressed ARW2

In 2015, we developed a tool called KARWY[5] that credibly repairs compression artifacts in Sony AWR2 raw image files. Although the artifact and processing are very different here, it was that experience which enabled us to build the new KARWY-SR tool to perform stripe removal. Unlike KARWY, which requires Adobe DNG Converter as a helper application and runs on a server, KARWY-SR (KARWY stripe removal) is self-contained JavaScript source code, and can be run locally within a web browser using a drag-and-drop GUI shown in Figure 10. Also unique to KARWY-SR is the fact that the output is a compressed Sony ARW raw file, just like the input to KARWY-SR, so the repair preserves all other attributes of the raw file.

The basic decoding of lossy-compressed ARW2 files was initially understood by examining code in dcraw[3]. Sony's raw file format is fairly complex, but has the highly desirable property that it obtains a fixed level of compression for image data. Thus, changes can be made to the raw image data in place without affecting any other contents of the file (e.g., EXIF data). The image data compression is explained in more detail in the original paper about KARWY[5], but for the current work it suffices to understand it as encoding into 16-byte blocks, each representing 16 consecutive same-color-channel pixel values. Depending on camera model and mode, the original pixel readings are either 12-bit or 14-bit linear values, which are then converted to 11-bit values implementing a piecewise linear approximation to log encoding. The minimum and maximum 11-bit values are directly recorded along with their 4-bit positions within the pixel sequence. The remaining 14 pixel values are each then encoded as a 7-bit delta scaled by the difference between the maximum and minimum for the block. The process is fully understood, and hence KARWY-SR can both decode and encode pixel data.

The graphical user interface to KARWY-SR was written in JavaScript, but the core processing was actually implemented in C. Emscripten[6] was used to compile the C code into the `asm.js` subset of JavaScript.

KARWY-SR Algorithm

When an file is dropped into the KARWY-SR interface, the file contents are copied into a local temporary file. This temporary file will be modified in-place and returned as the final result.

The algorithm to recognize artifacted pixels is:

1. Confirm that the file is a compressed ARW2. If not, no further processing is done – the input image is essentially ignored.
2. Process the file to extract an uncorrected preview image as a JavaScript blob and display it. The preview image is also labeled with the name of the camera model that was apparently used to capture the image.
3. The compressed ARW image is decoded to create several temporary data structures. Most significantly, bounds are computed for the original linear value of each pixel (accounting for the lossy compression). An array of linearized 16-bit raw pixel values is produced by picking a deliberately random value within the bounds computed for each pixel value. Adding random noise in this way would not change the values encoded, but avoids introducing roundoff-correlated artifacts in the later processing. This code is derived from the original KARWY.
4. The linearized pixel array is scanned to mark where sequences of 3 green pixels have at least 2 pixels brighter than the nearest green pixels in the +/-1 and +/-2 rows. In addition, red/blue pixels are marked where at least 2 of 3 same-color pixels are brighter than +/-2 rows and either there is a small difference between this and neighboring green pixels or the neighboring green isn't brighter than its +/-2 row neighbors.
5. Mark pixels where the pixel is brighter than the +/-2 rows neighbor and both the pixel +/-1 column were already marked. This bridges single-pixel detection gaps.

If one wanted to make use of knowledge of the particular pixel positions where PDAF pixels reside on the sensor of the particular camera model that was used to capture the image, steps 4 and 5 could be restricted to only consider the known PDAF pixel locations. Although execution is already fast enough, this would dramatically speed-up the marking. More significantly, it would ensure that very few non-artifact pixels would be marked. False-positive markings are rare, but the blonde hair in the same orientation as stripe artifacts did result in a few false positives in Figure 4. Of course, because KARWY-SR does not require knowledge of PDAF pixel locations, the tool can repair images from a wider range of camera models.

Given the markings, the repair algorithm is:

1. The initial repair replaces each suspect pixel value with a statistically-biased value interpolated from the +/-2 row neighbors. The value is almost an average, but deliberately

Figure 6. *A6500 image with extensive striping (enhanced)*

Figure 7. *A6500 image crop*

Figure 8. *A6500 image crop repaired by RawTherapee*

Figure 9. *A6500 image crop repaired by KARWY-SR*

injects random noise consistent with the pixel noise estimates. Using the average would change local noise statistics – producing smeared banding similar to that observed in images from the Nikon Z6 and Z7.

2. In the pure C-code version, these initial repairs are refined using the same texture synthesis logic employed in the original KARWY to further enhance the quality. In effect, the interpolated values are adjusted to be closer to the average value of nearby pixels that have similar-value neighbors. Unfortunately, the texture synthesis typically examines hundreds of patterns per pixel repaired, and thus was too slow to enable in the JavaScript version.
3. Any 16-pixel block in which some pixel changed its value must be recompressed and used to overwrite the corresponding original compressed data block. As described above, compressed ARW2 encoding involves packing 11, 4, and 7-bit fields, which is not particularly quick in JavaScript, so avoiding recompression of unchanged blocks yields a significant speedup.
4. A link to the corrected-in-place file is output.

Running as JavaScript within a WWW browser, a typical 24MP ARW2 file is processed in less than 5 seconds. All results shown were created using the JavaScript version.

Results

The improvement seen in Figures 4 and 5 is substantial. However, since RawTherapee version 5.5[7] can apply a similar repair while demosaicing a raw image, it is useful to compare the quality of the repairs. In fact, all the images shown here were rendered as JPEGs using RawTherapee, so a direct comparison of repair quality is possible. Figures 6 and 7 respectively show an A6500 image with extensive striping and a crop of an affected area. Figure 8 shows the excellent result of applying RawTherapee's PDAF stripe repair. However, careful examination of Figure 9, the repaired ARW2 file created by KARWY-SR, reveals minor differences that all favor this repair over the one implemented in RawTherapee. For example, RawTherapee failed to remove the line artifact that runs just under the stem of the flower after the leaf. The noise model in KARWY-SR also resulted in slightly more even statistical properties for the repaired areas.

Conclusion

Use of masked pixels for main-sensor phase-detect autofocus currently appears to be the scheme able to deliver the fastest, most precise, autofocus for mirrorless cameras. However, the rare striping artifacts it can cause seem extremely difficult to remove at the hardware source: they have merely changed form as Sony has gone through several generations of main-sensor PDAF

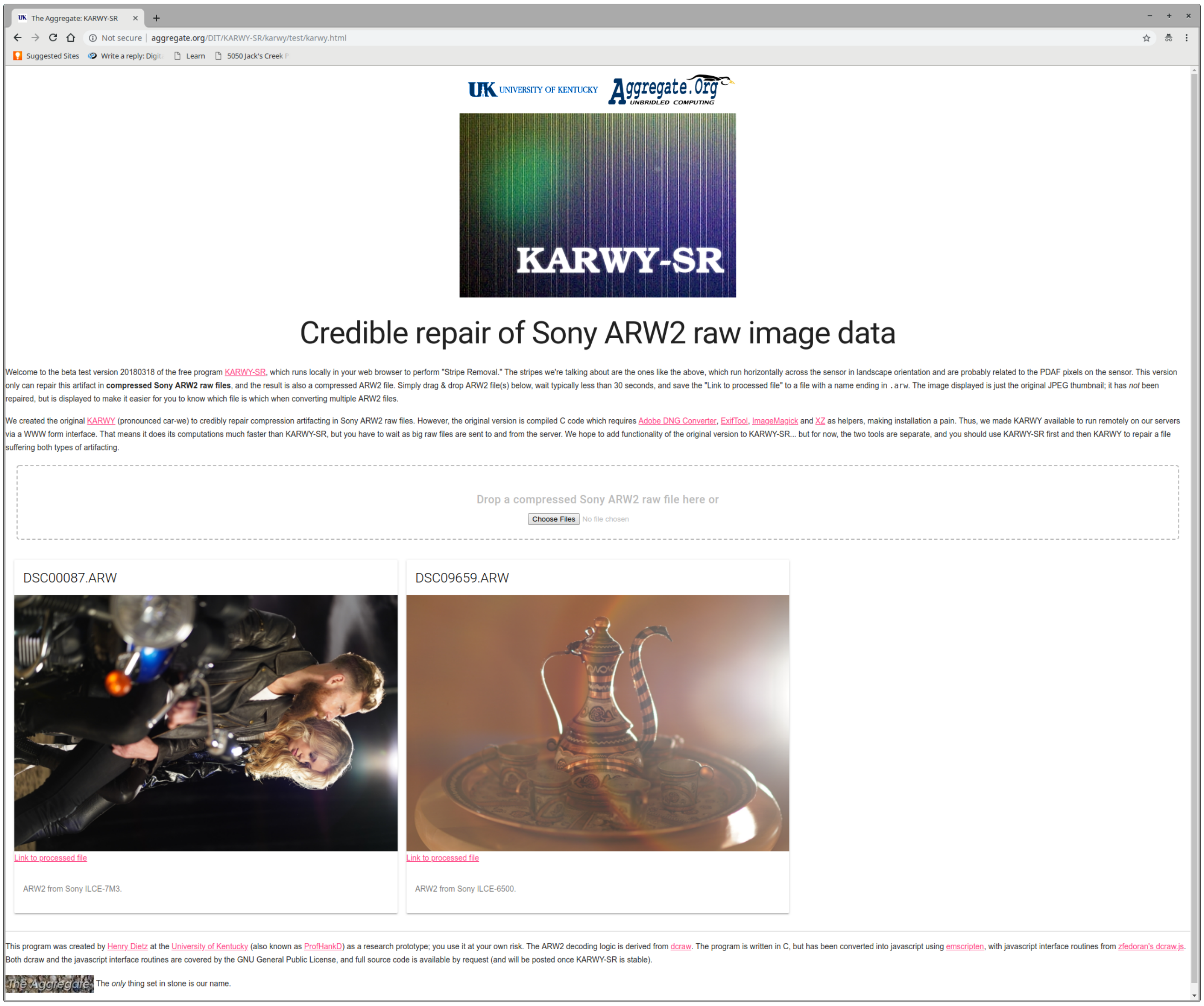

Figure 10. *KARWY-SR interface (JavaScript running in a browser)*

implementations. Thus, it is important to have a postprocessing fix that can rescue any affected images.

KARWY-SR, `http://aggregate.org/DIT/KARWY-SR/`, credibly repairs these artifacts using a simple algorithm to edit the ARW2 raw image file. Feedback on this open-source tool from both users and DPReview[8] has been very positive.

References

[1] S. Yasugi, Khang Nguyen, K. Ezawa, T. Kawamura, *Depth from Defocus applied to Auto Focus*, 2014 IEEE 3rd Global Conference on Consumer Electronics (GCCE), doi: 10.1109/GCCE.2014.7031237 (2014): pp. 171-173.

[2] Ichiro Onuki, Akihiro Nishio, Koichi Fukuda, Yoshihito Tamaki, and Yuki Yoshimura, *Image capture apparatus that performs focus detection with a phase difference detection method*, US Patent US9167148B2 (2012).

[3] Dave Coffin, Decoding raw digital photos in Linux, *http://www.cybercom.net/~dcoffin/dcraw/* (2015).

[4] Shinichi Fujii, Yasutoshi Katsuda, and Genta Yagyu, *Image Pickup Device And Image Pickup Element*, United States Patent Application 20110063484 (2011).

[5] Henry Gordon Dietz, *Sony ARW2 Compression: Artifacts And Credible Repair*, Electronic Imaging, Visual Information Processing and Communication VII, pp. 1-10, (2016). doi:10.2352/ISSN.2470-1173.2016.2.VIPC-227

[6] Alon Zakai, *Emscripten: An LLVM-to-JavaScript Compiler*, Proceedings of the ACM International Conference Companion on Object Oriented Programming Systems Languages and Applications Companion, pp. 301-312, (2011). doi:10.1145/2048147.2048224

[7] Raw Therapee 5.5 Released, *http://rawtherapee.com/* (December 17, 2018).

[8] Rishi Sanyal and Carey Rose, Sony 'striping': here's the fix, *https://www.dpreview.com/news/6974141509/sony-striping-heres-the-fix* (April 18, 2018).