Scope

In AXIOM cameras a simplistic raw image format was invented for storing raw sensor data plus contextual metadata. This format was first called “raw16” as it stored the data in 16 bit per pixel blocks.

Raw16

Format Specifications

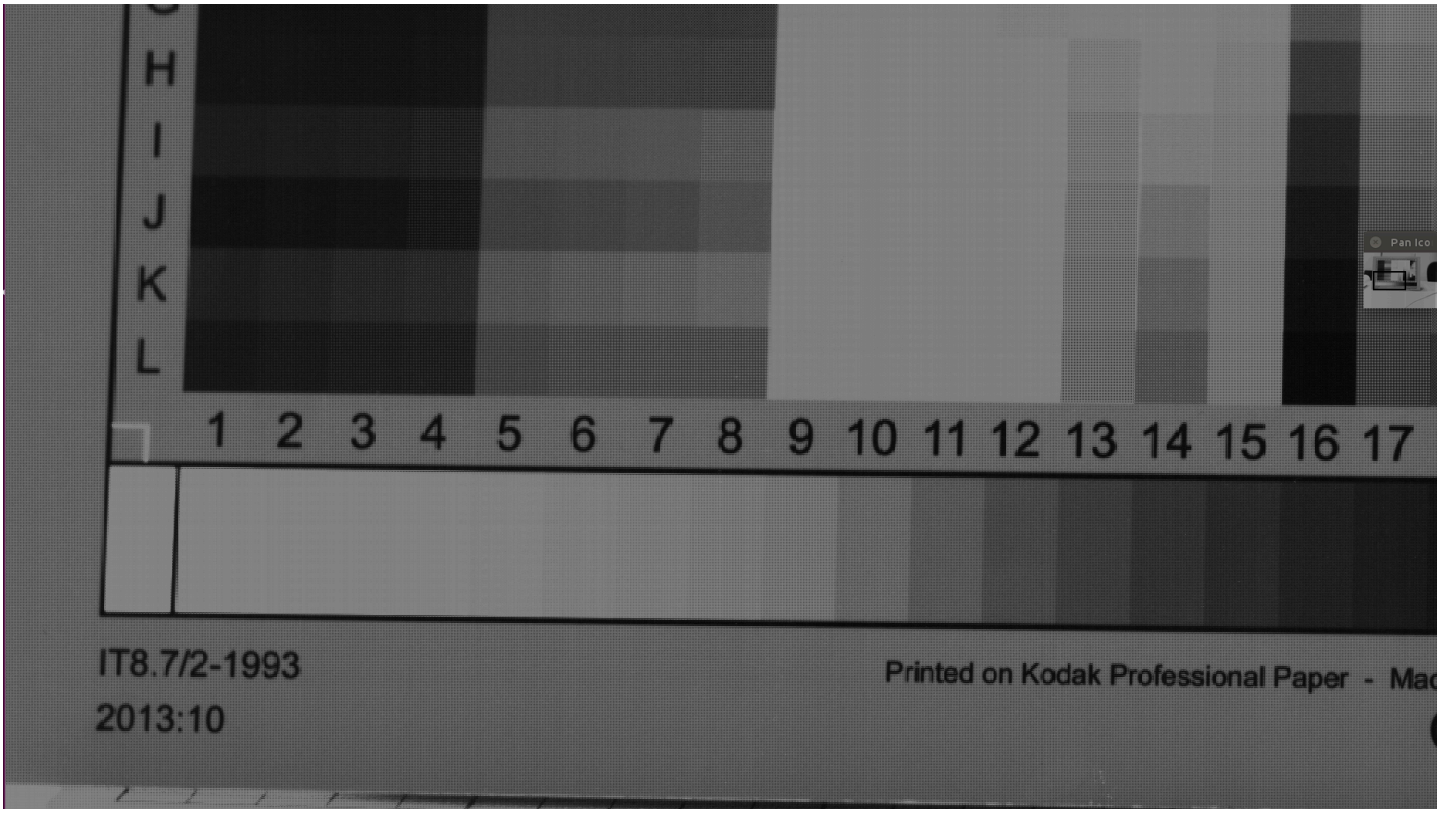
- no header, little endian, 16bit (12msb padded with 4lsb zeroes)
- starting at the top left, with 3072 rows of 4096 columns
- the data is in bayer pattern RG/GB
- image sensor registers dump (128 x 16bit, big endian) appended

Raw12

To save space a derived format that leaves out the 4 lsb zeros was created.

Working with raw12 files under Linux using imagemagick:

```
cat IT8-chart-15ms.raw12 | display -size 4096x3072 -depth 12 gray:-
```



Note that there is no color-filter-array interpretation (debayering) with this method.

DNG Converter

As the digital color negative format (DNG) is the only open standard universal format for storing raw data in a file that is widely supported in processing libraries as well as graphics and video image editing suites this is the primary choice for converting raw data to with the AXIOM.

The software that takes care of these operations has its source code stored on Github:

https://github.com/apertus-open-source-cinema/misc-tools-utilities/tree/master/raw2dng

```
DNG converter for Apertus .raw12 files

Usage:
  ./raw2dng input.raw12 [input2.raw12] [options]
  cat input.raw12 | ./raw2dng output.dng [options]

Flat field correction:
 - for each gain (N=1,2,3,4), you may use the following reference images:
 - darkframe-xN.pgm will be subtracted (data is x8 + 1024)
 - dcnuframe-xN.pgm will be multiplied by exposure and subtracted (x8192 + 8192)
 - gainframe-xN.pgm will be multiplied (1.0 = 16384)
 - clipframe-xN.pgm will be subtracted from highlights (x8)
 - reference images are 16-bit PGM, in the current directory
 - they are optional, but gain/clip frames require a dark frame
 - black ref columns will also be subtracted if you use a dark frame.

Creating reference images:
 - dark frames: average as many as practical, for each gain setting,
   with exposures ranging from around 1ms to 50ms:
        raw2dng --calc-darkframe *-gainx1-*.raw12
 - DCNU (dark current nonuniformity) frames: similar to dark frames,
   just take a lot more images to get a good fit (use 256 as a starting point):
        raw2dng --calc-dcnuframe *-gainx1-*.raw12
   (note: the above will compute BOTH a dark frame and a dark current frame)
 - gain frames: average as many as practical, for each gain setting,
   with a normally exposed blank OOF wall as target, or without lens
   (currently used for pattern noise reduction only):
        raw2dng --calc-gainframe *-gainx1-*.raw12
 - clip frames: average as many as practical, for each gain setting,
   with a REALLY overexposed blank out-of-focus wall as target:
        raw2dng --calc-clipframe *-gainx1-*.raw12
 - Always compute these frames in the order listed here
   (dark/dcnu frames, then gain frames (optional), then clip frames (optional).

General options:
--black=%d          : Set black level (default: 128)
                      - negative values allowed
--white=%d          : Set white level (default: 4095)
                      - if too high, you may get pink highlights
                      - if too low, useful highlights may clip to white
--width=%d          : Set image width (default: 4096)
--height=%d         : Set image height
                      - default: autodetect from file size
                      - if input is stdin, default is 3072
--swap-lines        : Swap lines in the raw data
```

```
                      - workaround for an old Beta bug
--hdmi              : Assume the input is a memory dump
                      used for HDMI recording experiments
--pgm               : Expect 16-bit PGM input from stdin

--lut               : Use a 1D LUT (lut-xN.spi1d, N=gain, OCIO-like)

--totally-raw       : Copy the raw data without any manipulation
                      - metadata and pixel reordering are allowed.

Pattern noise correction:
--rnfilter=1        : FIR filter for row noise correction from black columns
--rnfilter=2        : FIR filter for row noise correction from black columns
                      and per-row median differences in green channels
--fixrn             : Fix row noise by image filtering (slow, guesswork)
--fixpn             : Fix row and column noise (SLOW, guesswork)
--fixrnt            : Temporal row noise fix (use with static backgrounds; recommended)
--fixpnt            : Temporal row/column noise fix (use with static backgrounds)
--no-blackcol-rn    : Disable row noise correction from black columns
                      (they are still used to correct static offsets)
--no-blackcol-ff    : Disable fixed frequency correction in black columns

Flat field correction:
--dchp              : Measure hot pixels to scale dark current frame
--no-darkframe      : Disable dark frame (if darkframe-xN.pgm is present)
--no-dcnuframe      : Disable dark current frame (if dcnuframe-xN.pgm is present)
--no-gainframe      : Disable gain frame (if gainframe-xN.pgm is present)
--no-clipframe      : Disable clip frame (if clipframe-xN.pgm is present)
--no-blackcol       : Disable black reference column subtraction
                      - enabled by default if a dark frame is used
                      - reduces row noise and black level variations
--calc-darkframe    : Average a dark frame from all input files
--calc-dcnuframe    : Fit a dark frame (constant offset) and a dark current frame
                      (exposure-dependent offset) from files with different exposures
                      (starting point: 256 frames with exposures from 1 to 50 ms)
--calc-gainframe    : Average a gain frame (aka flat field frame)
--calc-clipframe    : Average a clip (overexposed) frame
--check-darkframe   : Check image quality indicators on a dark frame

Debug options:
--dump-regs         : Dump sensor registers from metadata block (no output DNG)
--fixpn-dbg-denoised: Pattern noise: show denoised image
--fixpn-dbg-noise   : Pattern noise: show noise image (original - denoised)
--fixpn-dbg-mask    : Pattern noise: show masked areas (edges and highlights)
--fixpn-dbg-col     : Pattern noise: debug columns (default: rows)
--export-rownoise   : Export row noise data to octave (rownoise_data.m)
```

Converting a raw12 file to DNG:

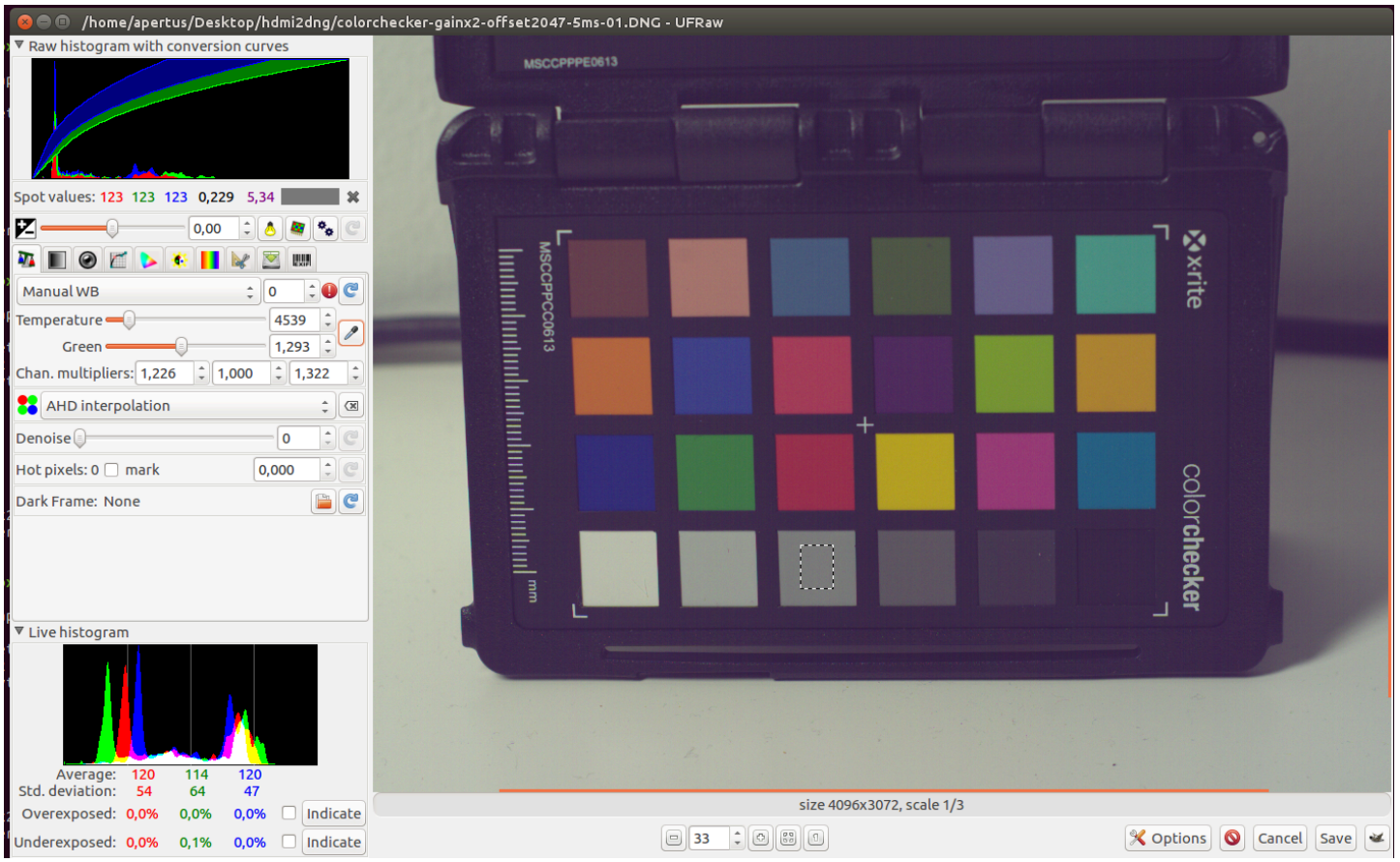
```
./raw2dng colorchecker-gainx2-offset2047-5ms-01.raw12 --swap-lines
```

Should generate an output similar to this:

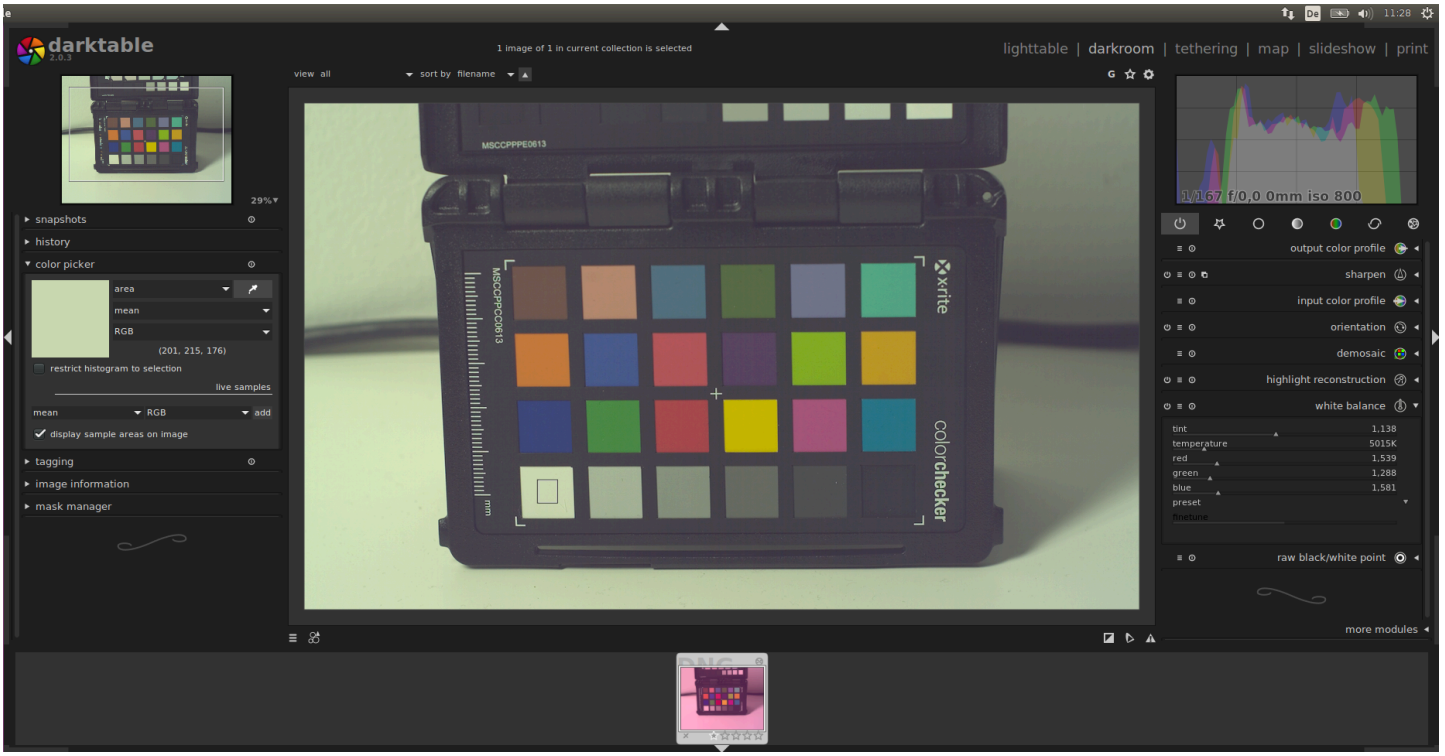
```
Active options:
--swap-lines        : Swap lines in the raw data

colorchecker-gainx2-offset2047-5ms-01.raw12
Resolution  : 4096 x 3072
Frame size  : 18874368 bytes
Bayer Order : GBRG
Exposure    : 5.99333 ms
Gain        : x2
Offset      : 2047
Black level : 128
White level : 4095
Line swap...
Not found   : darkframe-x2.pgm
Output file : colorchecker-gainx2-offset2047-5ms-01.DNG
Done.
```

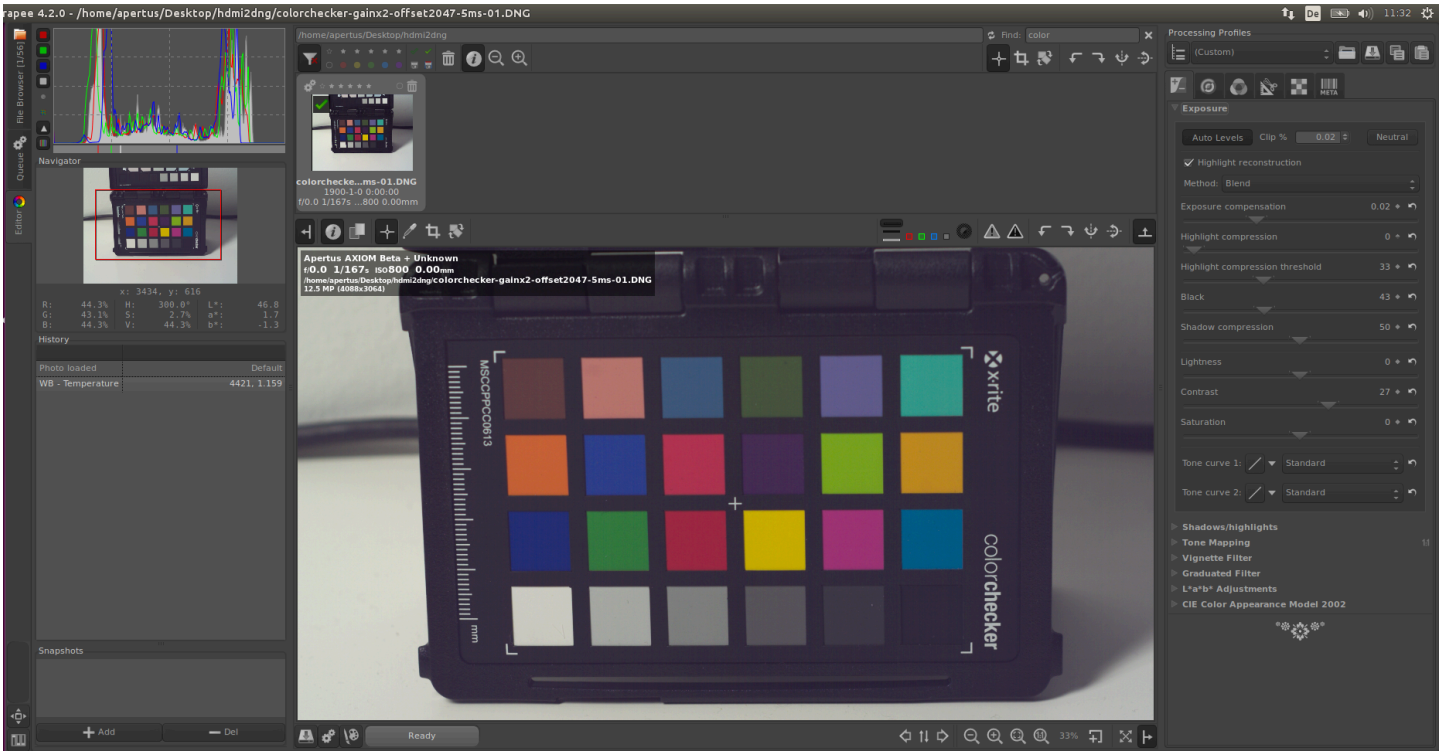
Afterwards the DNG can be opened in any raw image processing tool like:



Ufraw



Darktable



RawTherapee

Note that camera default color temperature and color matrix are not yet embedded in the DNG metadata. Therefore the white balancing has to be done manually.

Black Calibration

Finding the sensor output value, per pixel, in the absence of any illumination.
It covers:

- dark frame subtraction
- dark current compensation
- using black reference columns (called "optical black" by other manufacturers) to find the black level and fine-tune static offsets.

Note: black reference columns can be used to reduce row noise as well.
This is a basic technique: take a picture with the lens cap on, and subtract it from your image. To make really sure no light is reaching the sensor, also cover the entire camera with something.

How many dark frames?

Problem: if you take only one dark frame, it will also contain read noise (assumed to be Gaussian and uncorrelated with the read noise from other frames). Therefore, subtracting only one image will actually increase the noise in the final output, by sqrt(2).
Solution: use a *master* dark frame, averaged from many images.

Currently we capture 64 dark frames to create one averaged master darkframe.

Before:

After:

Note: The visible line noise is dynamic and compensated separately.