

Steve on Image Processing

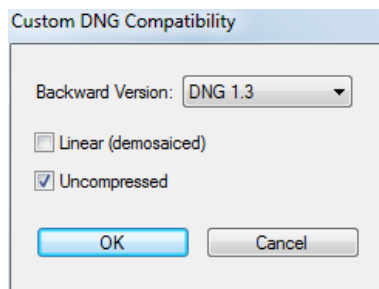
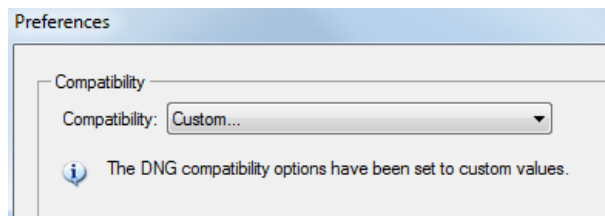
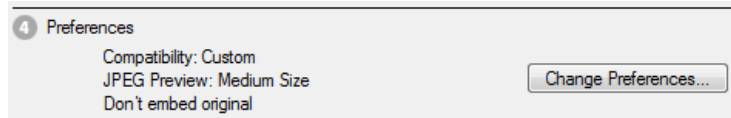
March 8th, 2011

Tips for reading a camera raw file into MATLAB

An academic colleague asked me recently how to read the sensor data (in [Bayer](#) or color filter array form) from a Nikon raw camera file (an NEF file) into MATLAB. We figured out a way to do it, and I thought I'd pass it along.

I should caution you, though, that there are multiple steps involved, some of which are "advanced maneuvers."

First, I suggested that we try using the free [Adobe DNG Converter](#) program to convert the NEF file to a DNG (Digital Negative) file. After some trial-and-error we found that it's necessary to tell DNG Converter to do the conversion "uncompressed." Here are some screenshots of the relevant dialogs.



Under the covers, a DNG file is a very specialized kind of TIFF file. However, we can't read the raw sensor data using `imread`. If you call `imread` on the DNG file, it will just give you back the thumbnail image. Instead, we have to make use of the MATLAB [Tiff class](#). This class offers low-level access to the "guts" of a TIFF file. To use it successfully, it helps to have some familiarity with how TIFF files work.

But first let's poke at a sample DNG file using `imfinfo`.

```
info = imfinfo('books.dng')
```

```
info =
```

```

      Filename:
      '\mathworks\home\eddins\files\I\ipblog_material\2011\books.dng'
      FileModDate: '08-Mar-2011 12:01:20'
      FileSize: 25079002
      Format: 'tif'
      FormatVersion: []
      Width: 256
      Height: 170
      BitDepth: 24
      ColorType: 'truecolor'
```

```

    ColorType: 'truecolor'
    FormatSignature: [73 73 42 0]
    ByteOrder: 'little-endian'
    NewSubFileType: 1
    BitsPerSample: [8 8 8]
    Compression: 'Uncompressed'
    PhotometricInterpretation: 'RGB'
    StripOffsets: 133234
    SamplesPerPixel: 3
    RowsPerStrip: 170
    StripByteCounts: 130560
    XResolution: []
    YResolution: []
    ResolutionUnit: 'None'
    Colormap: []
    PlanarConfiguration: 'Chunky'
    TileWidth: []
    TileLength: []
    TileOffsets: []
    TileByteCounts: []
    Orientation: 1
    FillOrder: 1
    GrayResponseUnit: 0.0100
    MaxSampleValue: [255 255 255]
    MinSampleValue: 0
    Thresholding: 1
    Offset: 8
    Make: 'NIKON CORPORATION '
    Model: 'NIKON D90 '
    Software: 'Ver.1.00 '
    DateTime: '2011:03:06 14:45:58 '
    SubIFDs: {2x1 cell}
    XMP: [1x7309 char]
    DigitalCamera: [1x1 struct]
    DNGVersion: [4x1 double]

    DNGBackwardVersion: [4x1 double]
    UniqueCameraModel: 'Nikon D90 '
    ColorMatrix1: [9x1 double]
    ColorMatrix2: [9x1 double]
    AnalogBalance: [3x1 double]
    AsShotNeutral: [3x1 double]
    BaselineExposure: 0.2500
    BaselineNoise: 1
    BaselineSharpness: 1
    LinearResponseUnit: 1
    CameraSerialNumber: '3276464 '
    LensInfo: [4x1 double]
    ShadowScale: 1
    DNGPrivateData: [121588x1 double]
    CalibrationIlluminant1: 17
    CalibrationIlluminant2: 21
    AliasLayerMetadata: [16x1 double]
    OriginalRawFileName: 'books.NEF '
    UnknownTags: [15x1 struct]

```

You can see that `imfinfo` thinks that this file is a 256-by-170 truecolor image. But as I mentioned above, that's just a thumbnail image. The data we want is hiding in something called a "SubIFD."

```
info.SubIFDs{1}
```

```
ans =
```

```

    Filename:
'\\mathworks\home\eddins\files\I\ipblog_material\2011\books.dng'
    FileModDate: '08-Mar-2011 12:01:20'
    FileSize: 25079002
    Format: 'tif'
    FormatVersion: []
    Width: 4310
    Height: 2868
    BitDepth: 16
    ColorType: 'CFA'
    FormatSignature: [73 73 42 0]
    ByteOrder: 'little-endian'
    NewSubFileType: 0
    BitsPerSample: 16

```

```

    Compression: 'Uncompressed'
    PhotometricInterpretation: 'CFA'
    StripOffsets: 356842
    SamplesPerPixel: 1
    RowsPerStrip: 2868
    StripByteCounts: 24722160
    XResolution: []
    YResolution: []
    ResolutionUnit: 'None'
    Colormap: []
    PlanarConfiguration: 'Chunky'
    TileWidth: []

    TileLength: []
    TileOffsets: []
    TileByteCounts: []
    Orientation: 1
    FillOrder: 1
    GrayResponseUnit: 0.0100
    MaxSampleValue: 65535
    MinSampleValue: 0
    Thresholding: 1
    Offset: 130252
    CFAPlaneColor: [3x1 double]
    CFALayout: 1
    LinearizationTable: [772x1 double]
    BlackLevelRepeatDim: [2x1 double]
    BlackLevel: 0
    WhiteLevel: 3767
    DefaultScale: [2x1 double]
    DefaultCropOrigin: [2x1 double]
    DefaultCropSize: [2x1 double]
    BayerGreenSplit: 0
    AntiAliasStrength: 1
    BestQualityScale: 1
    ActiveArea: [4x1 double]
    UnknownTags: [2x1 struct]

```

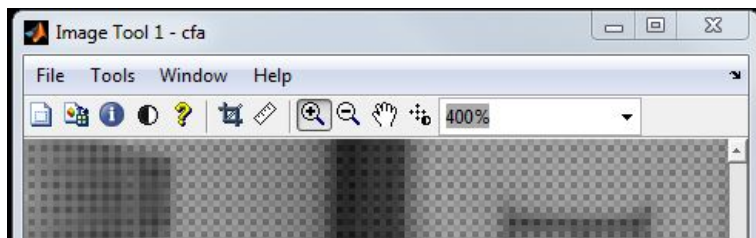
That's the good stuff! It's a 16-bit-per-sample 4310-by-2868 color filter array (CFA). Here's how to read it. I'm just going to give the steps without explanation and refer you to the [documentation for the Tiff class](#) for more information.

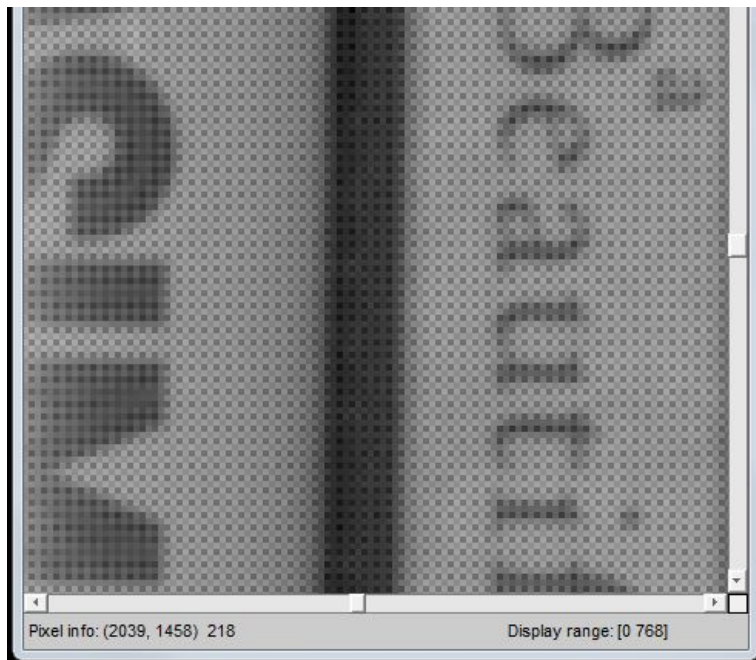
```

warning off MATLAB:tifflib:TIFReadDirectory:libraryWarning
t = Tiff('books.dng','r');
offsets = getTag(t,'SubIFD');
setSubDirectory(t,offsets(1));
cfa = read(t);
close(t);

```

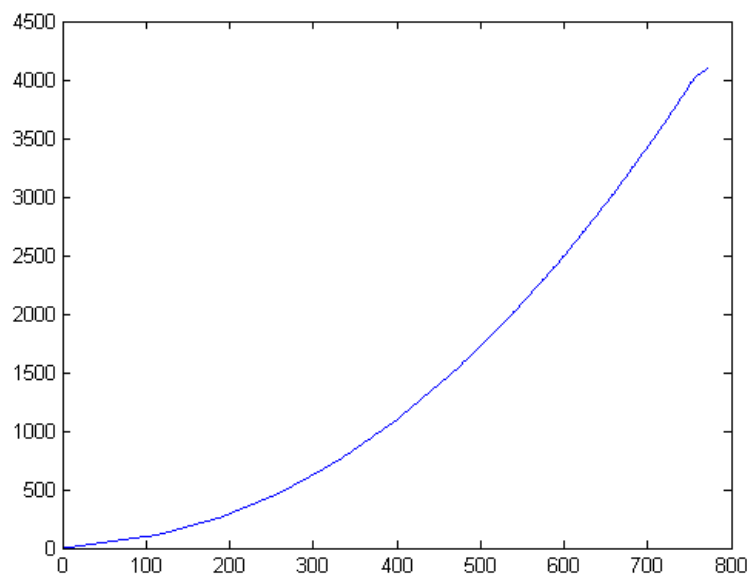
Here's a screen shot of `imtool` showing the resulting color filter array at 400% magnification. You can clearly see the Bayer array pattern.





We wondered why the maximum sensor value was 768. After some investigation, we found that Nikon appears to be storing a nonlinear quantization off the original 12-bit values. The linearization curve can be found in the DNG file as follows:

```
curve = info.SubIFDs{1}.LinearizationTable;  
plot(curve)
```



So if you were going to do calculations based on this raw data, you'd probably want to linearize it back to the original 12-bit range first using this curve.

done by posting a comment here.

[Get the MATLAB code](#)

Published with MATLAB® 7.11



By Steve Eddins

23:29 UTC | Posted in [Uncategorized](#) | [Permalink](#) | [19 Comments »](#)

You can follow any responses to this entry through the [RSS 2.0](#) feed. Both comments and pings are currently closed.

19 Responses to “Tips for reading a camera raw file into MATLAB”

1. *Joel Trussell* replied on [March 9th, 2011 at 01:39 UTC](#) :

Thanks for the most informative information. I'm sure it will help many – including many of my own students

2. *Craig* replied on [March 10th, 2011 at 00:23 UTC](#) :

Storing 'raw' 12-bit data as 9-10 bit with a curve to convert back to 12-bit seems to somewhat defeat the purpose of using raw data; conversion to 10-bit and back again will lose information. I wonder, does Nikon really do that or is this some short-cut introduced by Adobe's PNG (DNG, -steve) converter?

Does this also mean that converting Nikon NEF files to DNG does not reproduce the original linear output of the CCD/CMOS, without further processing? Or is DNG converter applying a gamma function during conversion?

Thanks for the blog.

Craig.

3. *Steve* replied on [March 10th, 2011 at 12:56 UTC](#) :

Craig—The DNG converter is a product of Adobe, and I have no idea what it's doing inside. My semi-educated guess is that the converter is faithfully reproducing the original data from the NEF file. Information on [this web page](#) suggests that Nikon is doing the quantization and gives a possible explanation.

4. *Craig* replied on [March 13th, 2011 at 01:36 UTC](#) :

I did some checking and Nikon's consumer-grade DSLRs (D90 and below) do indeed compress the raw sensor data from 12 bits to 9-10 bits in NEF format using curves of the type above. The higher-end DSLRs have the choice of compressed or uncompressed NEF.

Thanks for drawing this to our attention and suggesting a workflow to recover the linear sensor information. It is very important for anyone considering using consumer-grade cameras for scientific purposes to be aware of this.

Craig.

5. *Craig* replied on [March 13th, 2011 at 08:50 UTC](#) :

I hope this isn't a dumb question: is there somewhere in the subIFD information that details the sensor alignment for use in the demosaic function in MATLAB?

6. *Steve* replied on [March 14th, 2011 at 12:31 UTC](#) :

Craig—I don't know. I didn't see anything obvious.

7. *Jack Hogan* replied on [April 10th, 2011 at 13:08 UTC](#) :

Nikon's 'lossy' NEF compression stores all of the information in the underlying image data, thereby saving space and time, thanks to clever use of information theory: there is virtually no additional information in the uncompressed NEF. Emil Martinec explains it this way in the following link (scroll down to the NEF compression section): <http://theory.uchicago.edu/~ejm/pix/20d/tests/noise/noise-p3.html>

8. *Steve* replied on [April 10th, 2011 at 13:34 UTC](#) :

Jack—Thanks. That link is also given in my comment #3 above.

9. *David* replied on [April 12th, 2011 at 14:31 UTC](#) :

Hello, and thank you very much for this code. It has proven extremely useful. However, I am very new to

image processing and have a question that may seem basic but please bare with me.

I used your processing on a .cr2 (cannon raw) file and was able to display it using imtool and it is a BW image like your above with values ranging from 0 to 65535 which leads me to believe that the tone curve is already linearized for my 16 bit camera.

My question how does that 16 bit value correspond to the RGB values of the sensor? Is this camera specific and is there a way to find out?

I am trying to do white balance using this raw image in matlab. Thank you for any help!

10. Steve replied on [April 13th, 2011 at 12:08 UTC](#) :

David—Camera raw files are generally proprietary, and I don't know much at all about them. I'm afraid I don't know the answers to your questions.

11. *Ins* replied on [October 10th, 2011 at 06:47 UTC](#) :

Hi,

I wanna read the bayer image into matlab. Could you please tell me how to read the Bayer image?

Also How to obtain bayer image from Camera sensors?

12. Steve replied on [October 10th, 2011 at 13:28 UTC](#) :

Ins—That's what this blog post is about.

13. *Henrique* replied on [October 17th, 2011 at 00:10 UTC](#) :

Steve, I'm currently trying to read a 10 bit (per pixel) raw image from a Logitech UVC Pro 9000 camera in MATLAB using imaqttool.

Logitech provided as application to enable RAW mode in UVC cameras in both 8 and 10 bits.

For 8 bits per pixel Bayer I was able to acquire and work with the image but NOT in 10 bit mode.

I used the information from the following link: <http://www.quickcamteam.net/documentation/how-to/how-to-enable-raw-streaming-on-logitech-webcams>

Are you aware of any way to read 10 bpp RAW mode images from Logitech UVC cameras ?

It seems the data is not being read from MATLAB in the way Logitech imagined for 10 bit. I developed an .m code to convert the frames but the input seems not to be OK.

14. Steve replied on [October 17th, 2011 at 12:11 UTC](#) :

Henrique—No, I don't know anything about that.

15. *Stephen Nuske* replied on [December 20th, 2011 at 13:47 UTC](#) :

Hi Steve,

Could you list which versions of Matlab and Adobe DNG you use for the above steps? I am using MatlabR2011B and Adobe DNG Converter 6.5 and have followed your steps but do not have the LinearizationTable field inside info.SubIFDs{1}.

Thanks,

-Stephen

16. Steve *Eddins* replied on [December 20th, 2011 at 13:53 UTC](#) :

Stephen—MATLAB R2010b and Adobe Digital Negative Converter 6.3.0.79.

17. *Stephen Nuske* replied on [December 20th, 2011 at 18:57 UTC](#) :

Thanks for that info Steve,

I wonder if this LinearizationTable is camera specific?

Because it appears that there was no LinearizationTable applied by my NikonD300S to the NEF. After reading the cfa from the Tiff, as you listed above, the values go up past 768 which is unlike the NEF you tested with. The values in my NEF went all the way to the full 12bit range (4096).

I did a quick test using static camera and static lighting and plotting the mean cfa values against varying exposure time there is a linear relationship, no curve. So for my NikonD300S imaging setup it looks like I don't need to worry about the LinearizationTable.

Thanks,

Stephen

18. Steve *Eddins* replied on [December 21st, 2011 at 07:11 UTC](#) :

18. Steve Ludwig replied on [December 21st, 2011 at 07:11 UTC](#) :

Stephen—It probably is camera-specific. See Craig's comment #4 above.

19. *Stephen Nuske* replied on [December 23rd, 2011 at 11:33 UTC](#) :

Ah thanks, whoops sorry I missed that comment!

-Stephen

These postings are the author's and don't necessarily represent the opinions of MathWorks.

Problems? Suggestions? Contact us at files@mathworks.com

© 1994-2013 The MathWorks, Inc.

[Patents](#) | [Trademarks](#) | [Privacy Policy](#) | [Preventing Piracy](#) | [Terms of Use](#)