The Canon sRaw and mRaw Output Formats

Douglas A. Kerr

Issue 1.1
March 7, 2010

ABSTRACT

Many digital cameras (especially in the digital SLR genre) offer an optional form of output file in which the data does not directly represent a "finished" image but rather is a more-or-less verbatim transcript of the digitized output of the collection of individual sensor photodetectors. This is referred to as the "Raw" output. Recent middle- and upper-tier Canon dSLR cameras offer an alternative to this output file type, described as the "sRaw" ("small Raw") output, and in current cameras another variation as well, called "mRaw" ("medium raw").

In this article we first discuss the principles of the sensor arrangement that leads to the use of the Raw output concept. Then we briefly review the "regular" Raw output, what we do with it, and the advantages of operating in that mode. Next we describe the concept of the sRaw format, its important technical details, and why it is said to be beneficial. Finally, we introduce the mRaw format, and compare it with sRaw.

BACKGROUND

The matter of "raw" output formats is heavily influenced by the nature of the CFA (color filter array) sensor used in most modern digital still cameras. We begin with some background on that.

Tristimulus pixel sensors

It would be desirable in a digital still camera to have a sensor in which, at each pixel site of the delivered image, there was a photodetector that would report the color (luminance and chromaticity, or the equivalent) of the light landing there. One way to do this is with a *tristimulus* sensor detector. Thus would comprise, in fact or in effect, three photodetectors, each provided with a filter that shaped its spectral response, perhaps mimicking the spectral response of the three "channels" of the human eye. These filters would have their response peaks at wavelengths we can think of as "red", "blue", and "green", and so we will refer to them as the R, G, and B filters (and the associated sensels accordingly).

The outputs of these three "sensor elements" (sensels) at a pixel location would describe the color of the light there in a way consistent with human perception.

This is actually done in some digital still cameras, notably the Sigma cameras that have the "Foveon" tristimulus sensor system. Sony also has a patent on a structure that will accomplish this.

Some video cameras take another approach to the same end. They have three full sensor arrays, each with a sensel per pixel. The three arrays have overall spectral filters (R, G, and B). The light that forms the image is split three ways by a system of semireflective mirrors. Thus, each pixel location is observed by three sensor elements, one on each array, with differing spectral sensitivity. As before, the outputs of the three detectors for each pixel describe the color of the light for that pixel.

The color filter array (CFA) sensor arrangement

Sadly, we don't have either of these arrangements on most digital still cameras. Rather, a clever compromise known as the *color filter array* (CFA) sensor is used.

Here, there is a sensor element (sensel) at each pixel site. Each sensel has a spectral filter, either R, G, or B. They are arranged in a repetitive pattern. In the most common pattern (called the Bayer pattern, after its inventor, Bruce E. Bayer of Kodak), each block of four sensels (2×2) has one sensel with an R filter, one with a B filter, and two with G filters.

Of course, such an arrangement cannot explicitly determine the color (in either luminance or chromaticity) of the light at any pixel site, where there is only one sensel with only a single output value (no matter how you cut it, three values are needed to indicate a color).

But a very clever algorithm, by examining the outputs of a number of sensels clustered about each pixel site, can **deduce the most likely color** (again, in terms of both luminance and chromaticity) for each pixel of the image. This operation is commonly called *demosaicing*.

DEVELOPMENT OF THE SENSEL OUTPUT DATA

Transformation of the suite of digitized sensel outputs (one numerical value for each) into a full color image is often described as *development* of the image, by parallel to the development of a film negative. In the basic mode of operation, this is done in the camera. The basic sequence of operations is as follows:

- The R, G, and B values are each scaled by a value (different but consistent ones for each "channel", R, G, and B) to accommodate the differing sensitivities of each group of detectors.

- The resulting values are further scaled by another set of three factors (a "color correction vector") to apply a chromaticity shift for "white balance color correction". The purpose of this is to compensate for the situation in which the illumination on the scene has a different chromaticity than contemplated by the "color model" under which the delivered image is described. The result of such a discrepancy is a "color cast" on the viewed image.

  In reality, these two sets of scaling factors are usually consolidated into a single three-value vector, and applied in a single step of multiplication. But it is important that we recognize the different implications of the two sets of factors.

- The data is demosaiced to give an explicit color description (in "RGB form") for each pixel of the emergent image. This is in terms of the spectral sensitivities of the three kinds of sensels. Those do not match the primaries of any of the standard color spaces.

  I believe that some sharpening is applied during this process.

- The three coordinates for each pixel are transformed by multiplication by a matrix to put the representation into terms of the primaries of the desired output color space (perhaps either sRGB or Adobe RGB).

  The two steps above may be consolidated, or may be spoken of collectively as "demosaicing".

- Such processes as (further) sharpening and tonal scale mapping are applied.

- This RGB representation is transformed into one (YCbCr) that describes colors in terms of luminance and chrominance (a factor closely related to chromaticity).

- The chrominance data for every other pixel is discarded.[1]

---

[1] This is based on the fact that the human eye is able to recognize finer variations in the luminance of the light from a scene than in its chrominance. Thus, we can reduce the amount of data to be carried by only representing chrominance on a "more coarse" basis.

• The resulting set of data is compressed using the JPEG compression algorithm, allowing the image to be described (fairly accurately) by a smaller amount of data.

• The compressed data is packed for delivery as a JPEG image file.

THE RAW OUTPUT

While this process can provide very useful images in a wide range of circumstances, in “difficult” situations, and when “very particular” results are desired, another approach can be beneficial. Here, the collection of digitized outputs of the entire array of sensels, described as “raw sensor data”, is delivered essentially verbatim in a special file format (called a Raw[2] file).

Now, the development process, essentially following the sequence described above, is performed not in the camera but rather in external software (sometimes called a “raw converter”). This has a number of advantages, including:

• If the exposure was not “ideal”, we can shift the mapping of sensel output values onto image luminance values, in effect discarding the part of the sensel output range that is “too high” or “two low”. (One can of course only correct for a certain amount of exposure misadventure this way.)

• A demosaicing algorithm that “performs better” than that used by the camera can be employed.

• A primary remapping can be used that “performs better” than that used by the camera can be employed. (The matter is not definitive; decisions need to be made regarding the “best” visual appearance under various circumstances.)

Often the two above matters are swept together as the “demosaicing algorithm”.

• White balance color correction can be applied at this stage, when we may be in a good position to ascertain what “correction vector” gives us the most desirable result (perhaps even on a “season to taste” basis, or after measuring in the tentatively-developed image the

---

[2] This is often written as “RAW”, inappropriate since it is neither an abbreviation nor an acronym. I use the form “Raw” here (rather than just “raw”) to make it clear that this is a format name, not just an adjective.

chromaticity of an object in the scene we know to have "neutral" reflectivity).

• The finished image can be saved in a form that allows the image to be more precisely reconstructed for viewing or printing than with a JPEG file. For example, the image may be recorded in a TIFF file, in which we are free from the errors introduced by JPEG compression and can, if we wish, have all the coordinates expressed to 16-bit precision rather than the 8-bit precision of the JPEG file.

There are two prices paid for this improved capability:

• the Raw data file is substantially larger than the file carrying the developed image in JPEG form (perhaps as much as 10 times the size of the JPEG file for the largest image size image file at the "medium" compression quality).

• there is another stage to be performed in the overall process of getting the image (perhaps one that is a bit complex to execute).

Nevertheless, the use of the Raw output, with "external development" of the finished image, is very popular among both professional and advanced amateur photographers. The "large file size" disadvantage has been greatly mitigated by the availability, at modest cost, of memory cards for cameras having quite large capacities.

And modern Raw development software packages include many features that, in many "not too difficult" cases, can facilitate the user's execution of the operation. Why, most of them even allow us to tell the software, "develop this one just as the camera would have" (the Raw file includes all the parameters the camera **would have used** to internally develop the image).

CANON'S sRaw FORMAT

The basics

In 2007, in their EOS 1D Mark III digital SLR camera, Canon introduced a new output format option, the "sRaw" ("small raw") format. It is the actual subject of this article. It is now offered in all subsequent middle- and upper-tier Canon dSLR's.

Here are its primary characteristics (with distinctions drawn with the Raw format):

• The pixel dimensions of the image represented in sRaw form are half the pixel dimensions of the largest JPEG image format of the camera model. This normally corresponds to the smallest of the available JPEG image formats for that model. [The Raw output represents an image whose pixel dimensions are the same as the largest available JPEG image.]

• The data has seemingly already been demosaiced. That is, it does not report the outputs of sensels (of differing spectral sensitivities). Rather, it reports the recorded color for each pixel of the image associated with the sRaw mode. (But see below for a discussion as to why this should perhaps not be considered demosaicing.) By "recorded color", I mean in terms of the three sensel spectral sensitivities, not as transformed into terms of the primaries of a certain color space.

• The recorded color for each pixel is represented in linear luminance-chrominance form (essentially, a "linear" version of the YCbCr representation). [In the regular raw file, the output of each sensel is reported with a single number, as if it were a pixel detector in a "monochrome" camera.]

• The luminance data is decimated by a factor of 2 ("chrominance subsampling"). That is, only the even-numbered pixels in each row have both luminance and chrominance properties; the odd-numbered pixels only have luminance properties.

• The resulting suite of data is compressed with a "reversible" (often called "lossless") form of JPEG encoding and saved in a file. This means that when the data is "decompressed", the original data will be reconstructed, bit-for bit, without compression error. [The same is true of the regular raw data file"]

• The sRaw file size is roughly half the size of the raw file. (Recall that from it we can develop an image with one-fourth the number of pixels of the image potentially that can be developed from the raw file.)

The point of this?

Before we proceed with the technical specifics of the sRaw format, we might ask just what it does for us.

The format was first introduced in the Canon EOS 1D Mark II digital SLR, a top-end digital camera. The "White Paper" produced by Canon USA for this camera mentions the availability of this new output format, and expresses its value this way (to paraphrase):

> Photographers often require an image of lesser pixel dimensions than the maximum size for the camera (especially if the image is to be turned into deliverable prints of modest size). The use of

> the sRaw output file brings to such work the flexibility and control of image development given by use of the Raw file, but with a file size half the size, thus easing storage space requirements.

For later models, the Canon USA White Papers generally mention the availability of the sRaw output without any discussion of its advantages.

UNDER THE HOOD

Reverse engineering to the n^{th} power

Canon has not released any technical description of the sRaw format. What we know is largely do to the dogged work of various “reverse engineers” in the photographic community, including several who have developed “raw converters” that will process data in that form.

The details I will describe here are largely based on reports from workers who have “reverse-engineered” these reverse-engineered products, often with the assistance of a third layer of “reverse engineers”, all of whose results I have (yes, I’m afraid so), “reverse-engineered”.

Less from more

It is tempting to describe what happens to the original set of sensel outputs on its way to becoming the set of data in the sRaw file as comprising *demosaicing* and *downsampling*.

Before we examine that outlook more critically, let’s review what those mean.

Demosaicing

In a CFA array sensor, at each pixel site there is a sensor element, equipped with a certain spectral filter (from a repertoire of three). The outputs of this array of sensels does not directly tell us the color of the light at any pixel location, neither in luminance nor chromaticity.

But an ingenious algorithm, by examining the outputs of the sensels in a cluster around each pixel location, can deduce the “most probable” color of the light at that pixel. This is reported as the color of the pixel.

The process is called *demosaicing*.

Downsampling

If we have a data set that describes an image of certain pixel dimensions, and want a data set describing the same "image" but at a lower spatial resolution (smaller pixel dimensions), we say that we need to *downsample* it.

For example, if we want the new image representation to have exactly half the pixel dimensions of the course (in each direction, a total pixel count one-forth that of the original), we might just take clusters of four pixels of the original image (2 × 2), average their color coordinates, and use that average for the single pixel occupying the same real estate in the "downsampled" image.

But if the pixel dimensions of the desired new image do not have that handy relationship, something more subtle is needed. In effect, we must examine a cluster of the pixels of original image centered about the location of a pixel of the new image, and with an "interpolation" algorithm, determine the "most appropriate" color for that new pixel. There are many such algorithms, some extremely complex.

In sRaw

In the sRaw basic data set, the data describes the color of pixels of an image, not the "monochrome" outputs of an array of sensels with differing spectral sensitivity. So it seems that demosaicing has occurred. And that image has exactly half the pixel dimensions as the sensel dimensions of the sensor array. So it seems that downsampling has occurred as well.

But it may be a type of downsampling that is inherent in a primitive form of demosaicing—which isn't really demosaicing at all.

Imagine than we take a 2 × 2 block of sensels (which includes one R sensel, two G sensels, and one B sensel) and consider it to be a tristimulus photodetector for an image pixel whose size is 2 × 2 sensels. For openers, we can just take the R sensel output, the average of the two G sensel outputs, and the B sensel output, each scaled by the factor that reflects the differing sensitivities of the three kinds of sensels (and in addition takes care of white balance color correction) and declare that to be the "linear RGB" description of the color of that "fat" pixel.

Now, perhaps ideally, we would capture average color description over the entire "pixel area", or perhaps over a small area in the center of the pixel area. But in this case we capture the R aspect at one corner and the B aspect at an opposite corner. This is probably not of great consequence.

Now, has “downsampling” actually taken place? No. We did not start with a set of pixels at sensel resolution, and from it get a set of pixels at half that resolution.

Nor did we start with a set of sensel outputs (R, G1, G2, B) at sensel resolution, and get from it another set of corresponding values (R, G1, G2, B) at half that resolution.

So we cannot speak of the “downsampling” of any consistent type of data. Perhaps all we have is a very simple, direct way to develop a pixel image from the suite of sensel data—one that requires no “guessing” as to pixel colors.

Thus, we can see that perhaps what happens to the Raw data on its way to the sRaw format does not, rigorously, involve either *demosaicing* or *downsampling*.

Luminance-chrominance encoding

One of the genres of color models (schemes of coordinates used to represent the color of light) is the *luminance-chrominance* genre.

We are familiar with the concept of luminance, which we can think of as an indicator of the “brightness” of the light. (In an image, this is on a “relative” scale.)

Chrominance sounds a little like chromaticity, and in fact the two terms are related to the same concept, but they are different in an important way.

Chromaticity can be thought of, qualitatively, as the property that describes the difference between red and blue, and the difference between red and pink, in a way that applies to any value of luminance. Chromaticity is a two-dimensional property (in the mathematical, not geometric, sense): two values are needed to specify it. There are a number of sub-genres here, in which different pairs of coordinates are used to quantify chromaticity. Two recognized forms are:

• Hue-saturation. Hue is the property, within chromaticity, that distinguishes red from blue (or pink from pastel blue). Saturation is the property, within chromaticity, that distinguishes red from pink, or blue from pastel blue. There are different ways that these two properties are quantified.

• CIE x and y coordinates. These two values together specify chromaticity in a way that is derived from the "CIE XYZ tristimulus color space", in which a color is thought to be made up of a described amount of three "primary illuminants", all of which are imaginary and, by themselves, invisible. The familiar "horseshoe" representation of the colors of the spectrum is most often seen on a chart in which the axes are the CIE coordinates x and y.

Chrominance fits in with a color model in which we describe a color of light in terms of starting with a certain amount of "white" light (of course, that can mean several things, and we must be certain about which is meant) to which we add a "colorant light dose" (whose description is said to indicate the chrominance of the color). The "amount" of white light is indicated by the luminance value.

Typically, the "colorant dose" is described in a "two ingredient" form, each ingredient specified by a number than can be positive or negative. One ingredient always has a certain chromaticity for positive values, and, for negative values, the "opposite" chromaticity (that is, on the "opposite" side of the chromaticity diagram, if we travel through the point that represents white). The other ingredient always has a certain different chromaticity for positive values, and again negative values imply the opposite chromaticity.

For a color whose chromaticity is at the "white point" of the color space in use, the chroma is zero.

Now, let's look at a very significant difference between chromaticity and chrominance. Imagine that we have a floodlight with two lamps behind a certain "gel" (filter), such that the light from it has a certain color; that is, a certain luminance and a certain chromaticity. Now imagine that we kill one of the two lamps. What happens to the luminance, the chromaticity and the chrominance?

• The luminance falls to half of its original value.

• The chromaticity is unchanged (if the light was fully-saturated red, it still is—just not so bright).

• The magnitude of both ingredients of the chrominance falls to half.

The latter can be thought of in a very homey way. If we are going to mix not a gallon of paint, but rather a half gallon, we need to add half as much of each colorant ingredient to get the same appearance. (Think of the quantity of paint as the analog of luminance—I know it's a rather mixed metaphor.)

Where do we encounter a luminance-chrominance description of color, before we heard about sRaw?

To be precise, we do not encounter a true luminance-chrominance representation in any common photographic image situation. (It may seem that we do, but this is only because of imprecision of terminology.)

But we do encounter something very similar. In a JPEG image file, the data that is submitted to the JPEG compression algorithm does not represent image pixels in "RGB" form, as is often but incorrectly assumed. Rather, the data has been transformed into the "YCbCr" form (actually a specific type called "sYCC", for "standard YCbCr), a "pseudo luminance-chrominance" form.

At first glance, it may seem that Y represents the luminance of the color of a pixel, and Cb and Cr together represent the two ingredients of its chrominance. But in fact this is not so.

True luminance can be computed as a weighted combination of the amounts of three primary light components expressed in linear form (that is, twice the potency of an component has twice the number). Thus true (relative[3]) luminance (for which I will use the symbol Y_{tr}) can be calculated thus:

$$Y_{tr} = \mathrm{j}r + \mathrm{k}g + \mathrm{l}b \qquad (1)$$

where *r*, *g*, and *b* are the relative amounts of the different primaries (described on a linear basis) and j, k, and l are constants reflecting the relative sensitivity of the human eye to light having the chromaticities of the three primaries. (The exact values of j, k, and l depend on which set of primaries are involved, which depends on the particular "RGB" family color space involved.)

But, in our RGB color representation, we do not use r, g, and b, the "linear" representation of the amounts of the three primaries. Rather, we use nonlinear representations (a concept sometimes described, somewhat imprecisely, as "gamma precompensation"), the nonlinear variables being labeled *R*, *G*, and *B*.

The exact way this is done varies between different specific color spaces in the RGB family. A typical relationship is this:

---

[3] Relative in that it is not given in actual luminance units, such as cd/m^2, but rather on a relative scale compared to the maximum luminance that can be represented by the color model in the current presentation.

$$R = r^{0.45} \quad G = g^{0.45} \quad B = b^{0.45} \tag{2}$$

This corresponds to a value of "gamma" of 2.2 (1/2.2 = 0.45). In the sRGB color space, a slightly more complicated transform is used.

Now, in the SYCC YCbCr color space, the three coordinates are defined thus:

$$Y = 0.299R + 0.587G + 0.114B \tag{3}$$

$$Cb = 0.564(B - Y) \tag{4}$$

$$Cr = 0.713(R - Y) \tag{5}$$

We see that the coefficients of R, G, and B in the first equation add to 1, a mandate for conventional YCbCr representations. That being the case, then in the situation where R = G = B (which implies a color whose chromaticity is the reference white chromaticity of the color space in use) then *Cb* and *Cr* will be zero (no chroma).

Note that Y is not luminance, since it is not based on r, g, and b, but rather on their nonlinear transforms R, G, and B. It is not even a nonlinear form of luminance. (I will spare the reader the proof of this.)

In analog color television practice, where a form of color representation based on this same principle (including the gamma precompensation) is used, it became common to call Y "luma". To some, this was just a handy short form (true). To others, it had the significance that what was being referred to was a specific electrical signal, rather than just an abstract number (true). But a third significance is that this term, used instead of luminance, reminds us that this is nor really luminance, but what we might charitably call a "pseudo-luminance".

In a similar way, the two-dimensional property represented by Cb and Cr is not really "chrominance" either. And in television practice, the two components of this "pseudo-chrominance" (there defined in a different way, and called "I" and "Q") are said together to represent "chroma" (with the three same implications).

Thus, in discussion of such color spaces as YCbCr, I describe the two aspects as *luma* and *chroma*.

In any event, the positive direction of Cb implies a contribution of the first chroma ingredient equivalent to the blue primary; a negative value implies a contribution in the opposite direction, toward a chromaticity we can think of as yellow.

The positive direction of Cr implies a contribution of the second chroma ingredient equivalent to the red primary; a negative value implies a contribution in the opposite direction, toward a chromaticity we can think of as cyan.

Numeric precision

Before we move on, note that in the YCbCr representation used in our standard JPEG files (sYCC), the three coordinates are represented to a precision of 8 bits. The digitized outputs of the sensels in cameras supporting the sRaw format are to a precision of almost 14 bits. We must, however, be cautious about attempting to directly compare these precision values, owing to the non-linear nature of the YCbCr representation and many other considerations.

Nevertheless, the YCbCr representation does not fully support the potential image precision growing from the original precision of the sensel outputs. (For example, other image file formats, such as the 16-bit form of the TIFF format, do support that image precision potential.)

The Raw output preserves the image precision potential (since it comprises the sensor outputs, verbatim). In Raw conversion software, we can, if we wish, preserve the resulting precision of the developed image by delivering it not in JPEG form but in, for example, 16-bit TIFF form.

Back to sRaw

Now that I have prepared us to understand that YCbCr, which we have had in hand for some while, is not really a luminance-chrominance representation, I'm pleased to tell us that the data in sRaw is presented in a bona fide luminance-chrominance form, although one very reminiscent of YCbCr.

In fact, some writers use the symbols Y, Cb, and Cr to represent the three coordinates used in the sRaw format. I do not, for fear that inappropriate equivalences be drawn. Rather I adopt a distinct, but still mnemonically-useful, notation, using lower-case versions of the familiar symbols—y, c_b, and c_r.

A long trail of reverse engineering, running thorough messrs. Coffin, Clévy, Yang, and Kerr (see the Acknowledgements section at the end) leads to the conclusion that the actual arrangement is:

$$y = 0.296r + 0.592g + 0.114b \tag{6}$$

$$c_b = b - y \tag{7}$$

$$c_r = r - y \tag{8}$$

where *r*, *g*, and *b* are the linear "tristimulus fat pixel" sensor output equivalents.

Now, does *y* actually represent bona fide luminance? Well for that to be so, the coefficients of the three terms in the equation for y must be appropriately chosen in the face of:

- A certain set of assumed primaries (as might be defined for a particular color space).
- A certain reference white chromaticity (as might be defined for a particular color space).

In fact, the coefficients seen above are almost precisely those that would apply for the NTSC RGB color space, a color space used in digital television encoding. It assumes a set of primaries corresponding to the phosphor colors for typical color TV CRT displays, and assumes for the white point "illuminant C", a particular chromaticity that does not really match any "likely" real illumination source, is not assumed by many working color spaces, but is for historical reasons often used as a reference in theoretical color work.

Recall that Raw (and sRaw) data is not really defined in terms of an "output" color space, such as sRGB or Adobe RGB; that particularization comes later in the process. So NTSC RGB is as good an arbitrary color space as any to use as the premise for the development of "luminance" in the yc_bc_r representation.

Oh, yes: the values y, c_b, and c_r are stated in 14 bits.

Chrominance subsampling

During the development of the first standard US color TV format[4] (analog, of course), attention was paid to earlier research that showed

[4] Called the *NTSC* format, recognizing its development by the National Television Systems Committee. Later, after different encoding systems had been developed for use in other countries (naturally starting with a advanced body of literature, including much experience gained with the use of the NTSC system), it became popular for advocates of these other systems (arguably more consistent in performance) to say pejoratively that NTSC stood for "never twice the same color."

that the human eye could distinguish changes in luminance at a finer pitch (that is, with finer acuity) than for changes in chromaticity.

Thus, in that format, the chroma was transmitted with less resolution than the chroma (thus requiring a lesser bandwidth ration). In fact, refinement of that research had showed that the human "chrominance acuity" was greatest along a certain axis in the x-y chromaticity diagram. Based on this, the two ingredients of chroma were transmitted with different bandwidth (both of them less than for luma) to give the overall "biggest bang for the hertz".

In modern digital image representation, we commonly follow that principle (usually just the first part of it, lesser resolution for chroma than for luma, the two chroma subcomponents being at equal resolution).

In particular, in the YCbCr representation used as the underpinning of a JPEG image file, we may have only one chroma value (that is, one Cb and Cr) for every two or four pixels (each of which has a Y value).

Now, in sRaw, the same principle is followed: there is only one chroma value (that is, one c_b and c_r) for each two pixels (each of which has a y value).

This chrominance subsampling arrangement is formally designated "4:2:2" (don't ask[5]).

When the sRaw file is loaded into raw conversion software, the pixels that do not bring a chrominance value (c_b, c_r) with them are provided with one developed by interpolation between the c_b and c_r values for the pixels to the left and right.

WHAT EXACTLY DO WE HAVE?

Before we proceed to the bigger cousin of sRaw, mRaw, let's reflect on the bottom line of sRaw.

Since the pixel image represented by sRaw has not had the "color correction" aspect of scaling applied (only the part that recognizes the differing sensitivities of the three classes of sensels), has not been transformed to recognize the primaries of the destination color space, and carries the data in a linear form, at 14-bit precision, we can use it in much the way we can use the raw data itself—assuming that we are willing to end up with an image of half the resolution of that which

[5] Or, if you are a masochist, see "Chrominance Subsampling", by this same author.

would have been developed from the Raw data. And we can do this with a file about half the size of the Raw file for the same shot.

There are many discussions about whether this is actually of any value. Some point out the compromises of the chrominance subsampling. Some point out that we no longer have the chance to apply a “superior” demosaicing algorithm, if we can find one. (However, as I pointed out above, demosaicing in this situation is essentially trivial—we really have a tristimulus pixel sensor array for this image size.)

Many comment that the designation—sRaw—is inappropriate, suggesting that the file contains “raw” data, which is usually considered to mean the original sensor outputs. Still, the image represented by the data in the sRaw file isn’t “fully done” yet—it is perhaps “precooked”, maybe “a little raw”.

Many point out that the seeming point of this—to give the benefit of greater flexibility in image development with a file half the size of a Raw file (for an image with one-fourth the number of pixels)—is greatly diluted by the availability today at modest cost of camera memory cards with very large storage capacity. (I won’t quote any prices—they will seem silly by the time anybody reads this.)

THE BIGGER BROTHER— mRaw

Not too long after the introduction of the sRaw format, Canon introduced a second format in much the same vein, offering both as options in their later cameras.

In its first outings, it was called “sRaw1”, with the version we have already described being called at that time “sRaw2”. Eventually, it became known as the mRaw format (“medium Raw”, I suppose), and the original format became again “sRaw”.[6]

The mRaw format differs from sRaw in these prominent ways:

- It represents an image having about one-fourth the pixel count of the image directly developable from a Raw file (usually, corresponding to the second-smallest JPEG output image for the particular camera model. (This image has about 71% the linear resolution of the “native” image.)

[6] Remember, this is the same company that, having previously offered a landmark digital SLR called the EOS D30, later brought out a wholly different one called the EOS 30D.

- The chroma is subsampled at a rate of 1:4 (that is, for every four pixels, there is only one set of c_b and c_r). This chrominance subsampling arrangement is formally designated "4:2:0" (don't ask).

When an mRaw file is loaded into raw conversion software, the three pixels of each block of four (2 × 2) that do not come with their own c_b and c_r values are provided with them through interpolation between the values of adjacent "fully specified" pixels, vertically or horizontally. The poor pixel in the corner opposite the fully specified one gets its c_b and c_r by interpolation between the c_b and c_r values for the pixels above and below, each of those being an already interpolated value (from the pixels to **their** left and right).

- The file size is approximately 3/4 the size of the Raw file, 1.5 times the size of the sRaw file.

In this case, the "demosaicing" cannot be trivial, as I conjectured for the sRaw file. The process should probably not be thought of as "demosaicing to a native size image followed by downsampling". Rather, it is most likely the development, in one stroke, of an image of this size by inference from the suite of sensel outputs. The algorithm may well be similar to that used to derive the "next-to-smallest size" developed image for JPEG output.

As to the advantage of this format over other possibilities available to the photographer, I would hate to volunteer an opinion.

ACKNOWLEDGEMENTS

Much of the credit for "our" knowledge of the Canon sRaw (and mRaw) formats is due to Dave Coffin, the developer of *dcraw*, an open-source program that decodes Raw files (of many types) and "develops" the images they describe. He has had to mostly do this based on some very difficult "reverse engineering".

Next in the food chain comes Laurent Clévy, who has "reverse engineered" the source code in *dcraw* for the Canon Raw files (including sRaw and mRaw), and has analyzed and described its implications. His input also included analysis of the various tags in the file structure using *exiftool*, developed by Phil Harvey, perhaps the dean of our metadata reverse engineers.

Joining the fray was Yang Gao, who, among other things, recognized the parallel between the encoding of the sRaw data and that of the YCbCr color space.

I came in late, and only had to figure out what it all means, and explain it.

Thanks also to Bruce Lindbloom, whose Web site is a treasure of information on color coordinates. It provided the coefficients of the expression for luminance for the NTSC-RGB color space.

Finally, thanks to my wife Carla for her keen and insightful copy editing of this difficult manuscript, not to mention assuring a stream of delicious and nourishing food to fuel the effort, and for creating an environment in my life within which this kind of investigation, analysis, and exposition can flourish.

#