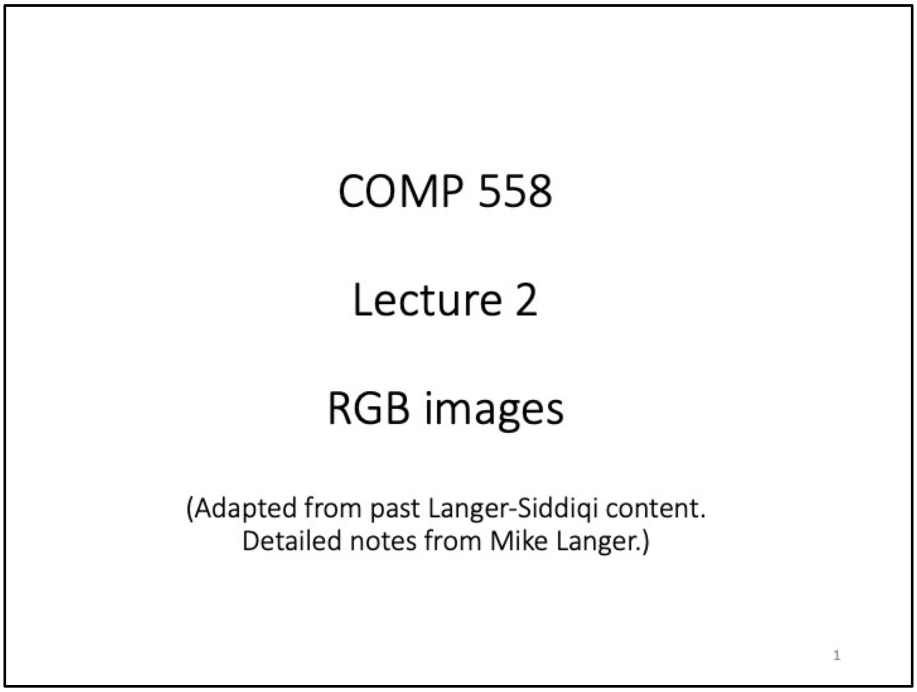


COMP 558

Lecture 2

RGB images

(Adapted from past Langer-Siddiqi content.
Detailed notes from Mike Langer.)

Hi everyone. Welcome to the first real lecture of 558 for this semester.
Today I'm going to talk about RGB images. We'll spend the whole semester talking about processing and interpreting images, so its good to start off by learning a bit about what images actually are.
I'm not talking about the images that you measure with your eye. Rather I'm talking about the images that are recorded by a camera and that we process with computers. We do "look" at these images when they are displayed on a screen, but today is not about us looking – rather, today is about the digital images themselves.

Cameras

Digital SLR

“point and shoot”

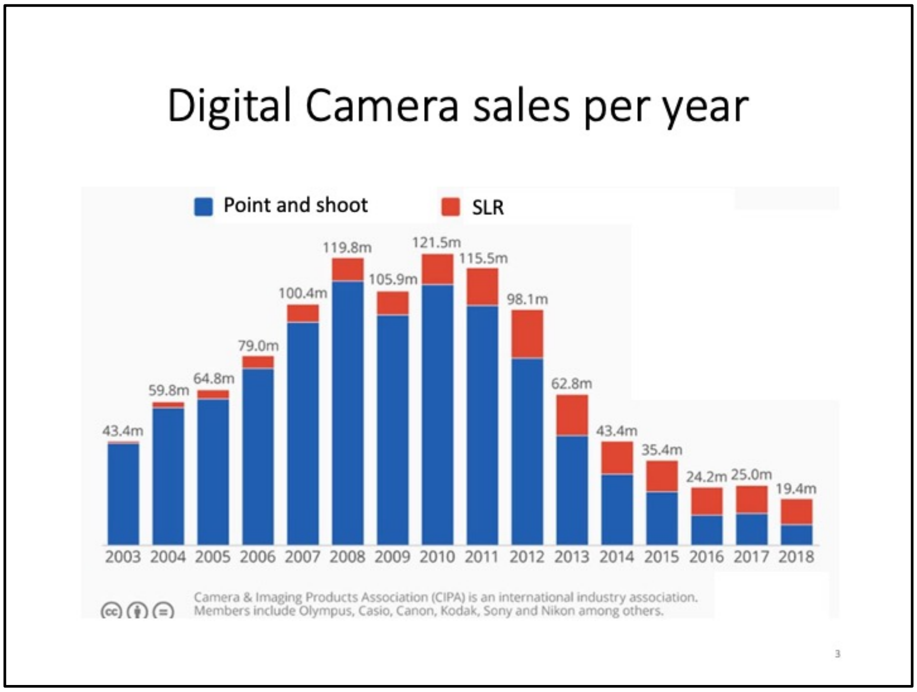
cell phone

2

I’m sure you all have some experience with making images, whether you are a hobby photographer or just someone who just likes to take pictures of your friends and family and pets, or yourself.

If you are a hobby photographer, then you probably own a camera such as the one on the left.   This is called a single lens reflex or SLR camera.   I wrote “digital” SLR camera on the slide, since before most of you were born there were non-digital SLR cameras, namely film cameras.   I know that some people have gone back to film, just like some music fans have gone back to vinyl.   But let’s just talk about digital cameras today.

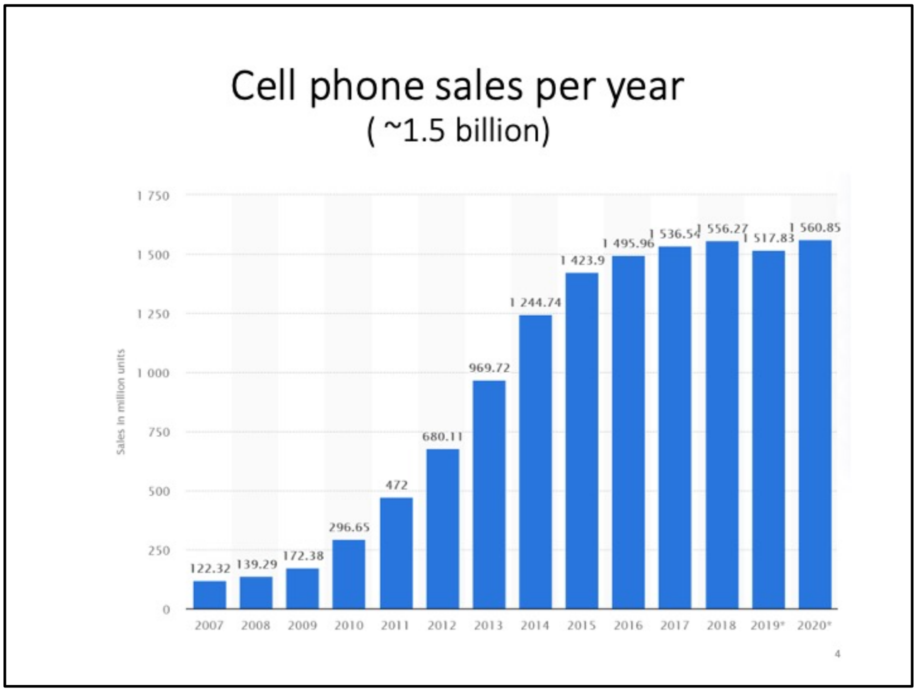
The camera in the middle is called a “point and shoot camera”.   It is typically smaller than an SLR.   The camera on the right is the one that many of you presumably own and use – the cell phone camera.



Digital Camera sales per year

Camera & Imaging Products Association (CIPA) is an international industry association.
Members include Olympus, Casio, Canon, Kodak, Sony and Nikon among others.

Digital cameras were invented several decades ago. But it was only in the past 20 years that they have become a commercial success. Sales of point and shoot cameras are shown in blue over a period of 15 years starting in 2003. As you can see, these sales were very big in the first decade after 2000, but their sales of point and shoot have died off since then. The sales of SLR cameras are shown in red. Sales of SLR cameras have been steady, around 10 million per year for the last decade.



Cell phone sales per year

(~1.5 billion)

The reason that point and shoot cameras don't sell so well is that cell phone cameras have taken over the market. There are about 1.5 BILLION cell phones sold every year, and most of these come with built in cameras. The image quality is not great. In particular, my 4 year old phone takes pretty BAD pictures. Its fine for most purposes, and its amazingly convenient. But if I want to make a high quality photo, then I'll use my digital SLR.

I should mention that the enormous increase in digital photography over the last two decades has had a profound impact on the field of computer vision. Techniques have become more *data driven*. I'll have more to say about this as the course goes along.

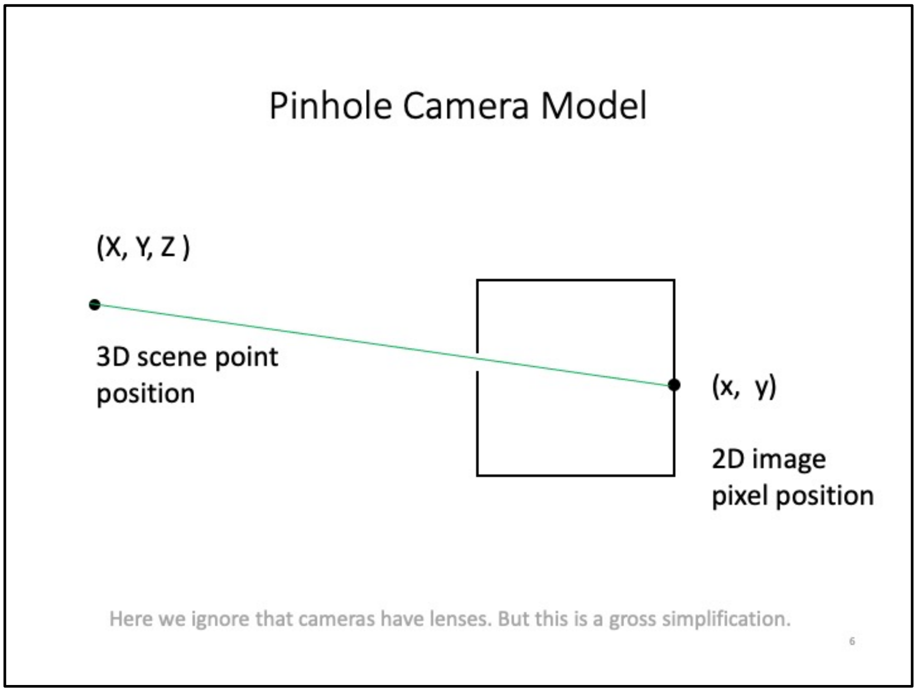
Real Digital Camera (SLR)

Today we will concentrate on the sensors and what is meant by RGB.

5

Real digital cameras are very complicated devices. Here we see the various parts of a digital SLR camera. There are lenses, mirrors, sensors and other elements.

We will concentrate on the sensors. In particular, we will discuss what pixels are and what RGB means.



Pinhole Camera Model

(X, Y, Z)

3D scene point position

(x, y)

2D image pixel position

Here we ignore that cameras have lenses. But this is a gross simplification.

6

We will ignore the optics for now.   Instead, let just work with the following simple model of image formation.   Consider some visible point out in the scene at 3D position XYZ. That point projects through a small opening in the camera, called the aperture.   If the aperture is small, we can think of it as a single point or “pin hole”.   In that case, the 3D point then projects uniquely to a 2D position (x,y) on the sensor.   The 3D point in the scene has some color, and the point in the image measures that color.

Think of the point in the scene as some point on a surface.   Maybe it is a blade of grass, or maybe it a point on someone’s green eye. Or green hair. I’ll have more to say about “color” as we go on today.

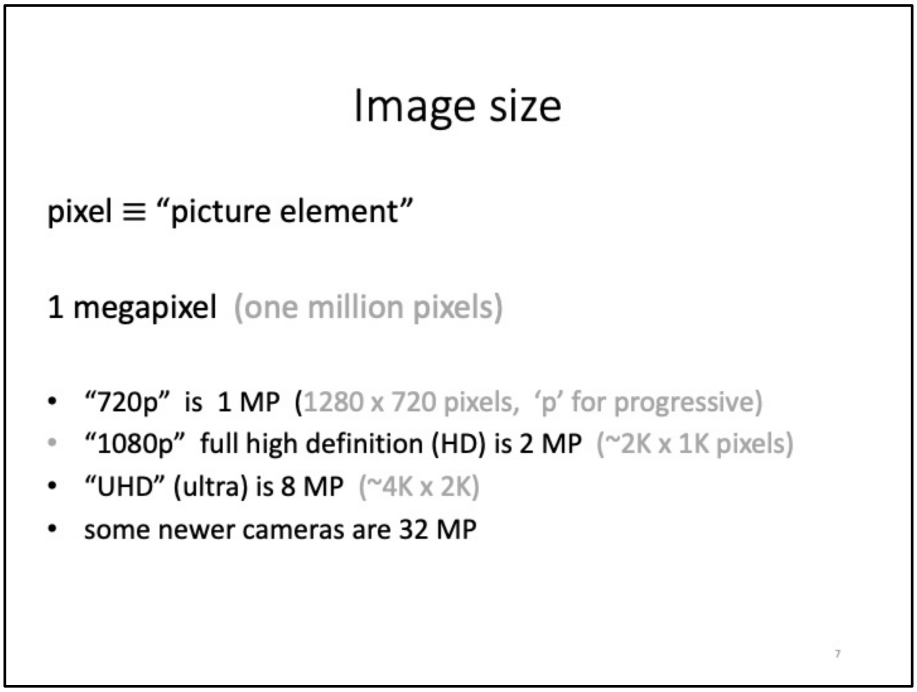


Image size

pixel ≡ “picture element”

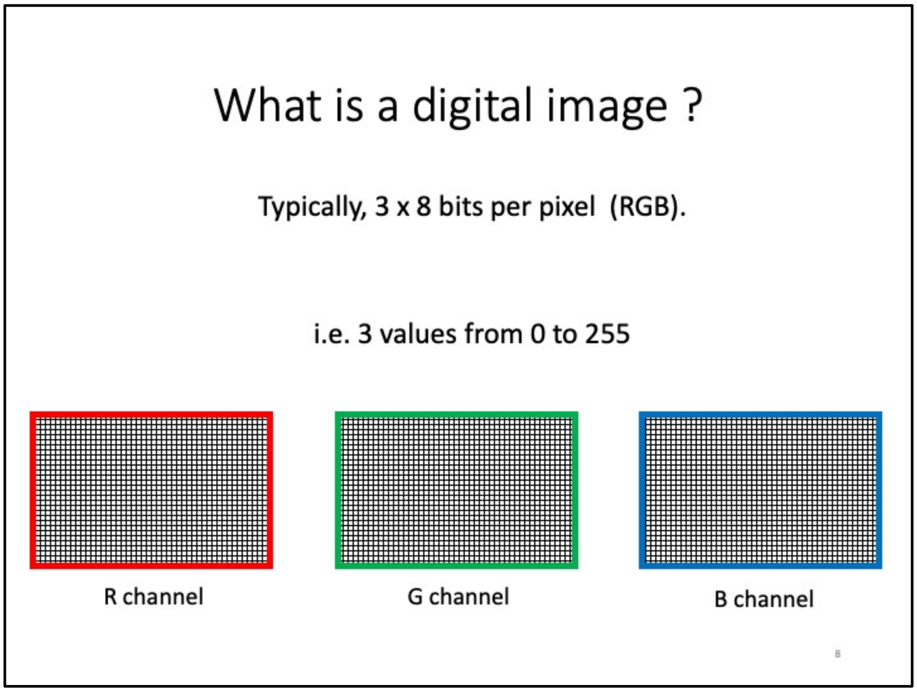
1 megapixel (one million pixels)

- “720p” is 1 MP (1280 x 720 pixels, ‘p’ for progressive)
- “1080p” full high definition (HD) is 2 MP (~2K x 1K pixels)
- “UHD” (ultra) is 8 MP (~4K x 2K)
- some newer cameras are 32 MP

When we take pictures and work with images, as we’ll be doing in the assignments in this course, one of the first issues we need to deal with is the size of the image.

We say that an image consists of “pixels”. The word “pixel” means a picture element. A one megapixel image has one million pixels. An example would be an image with 1000 rows x 1000 columns: 1000*1000 = one million.
There are other image sizes you may have heard of. 720p is also about one megapixel. It is a standard with 1280 columns and 720 rows. (ASIDE: ‘P’ is for progressive scan, which is in contrast to the “i” format which stands for “interlaced”. Old televisions and old display standards were interlaced, namely they only showed half the lines in each frame, and alternated which half from frame to frame.) We’ll only be talking and thinking about progressive scan, which just means non-interlaced.

HD or high definition is 1080p. It is a standard with about 1000 rows (1080 to be precise) and about 2000 columns. So it has about 2 million pixels or 2 megapixels. There are more recently standards such as ultraHD which have 8 megapixels, and new cameras I’ve seen advertised have 32 megapixels.

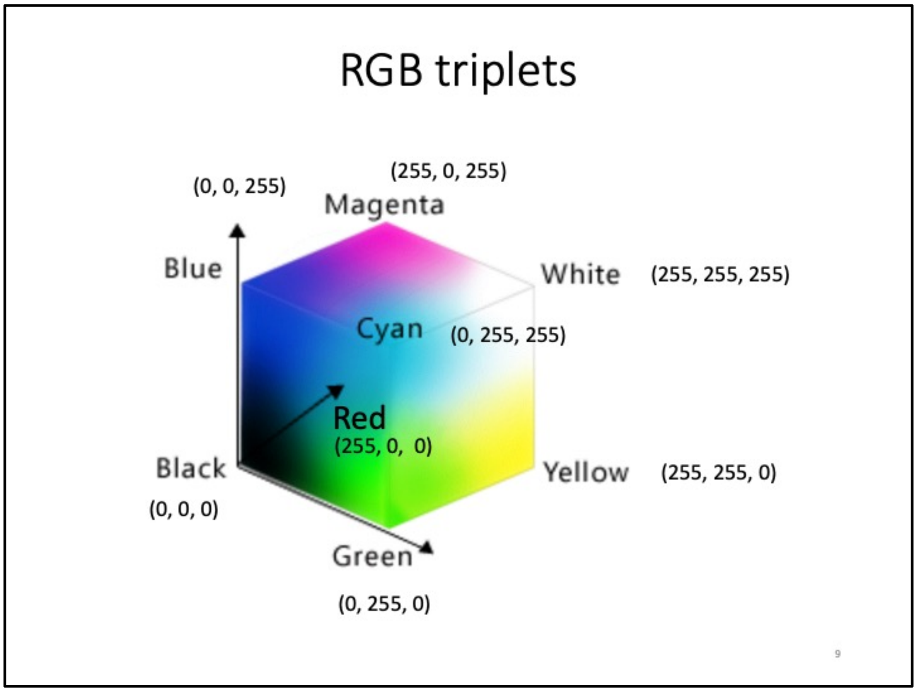


What is a digital image ?

Typically, 3 x 8 bits per pixel (RGB).

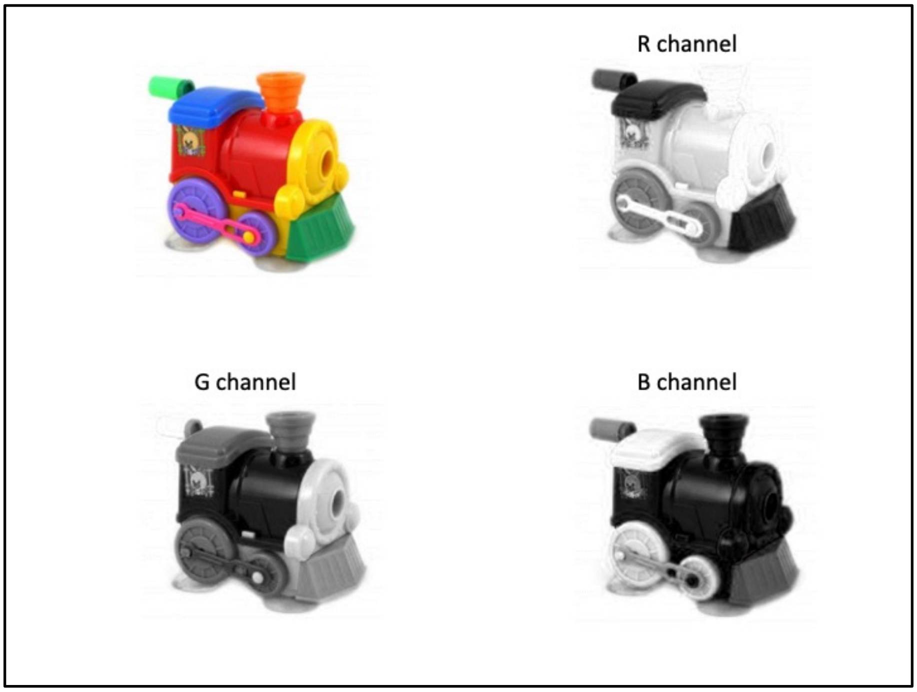
i.e. 3 values from 0 to 255

We’ll say quite a bit today about what a pixel is. First of all, a pixel has three values associated with it: R, G, and B, or red, green and blue. We sometimes talk about the RGB “channels” of an image. Often these are 8 bit values, so we can think of them as numbers from 0 to 255. There are no physical units associated with these numbers though. Rather, they are just a way of ordering “intensities” where 0 is darkest and 255 is brightest. When there are 8 bits per R, G, and B value at each pixel, we have 24 bits or 3 bytes each pixel.



RGB triplets

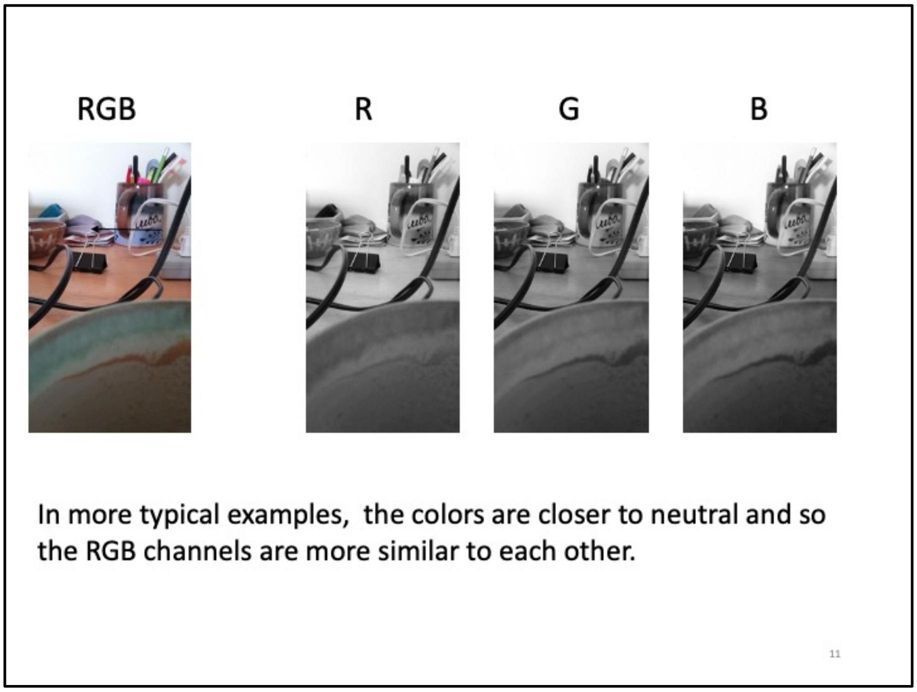
3 bytes for each RGB value means we can think of each pixel as a point in a 3D space. The corners of this space are the points where each RGB value has either its minimum or maximum, namely 0 or 255 respectively. These corner points have color names that you are familiar with. Black is (0,0,0) and it is at the bottom left corner. White is (255, 255, 255) and it is at the top right corner. Red, green and blue have values that are 0 in two channels, and 255 is one channel. Green is (0, 255, 0) and it lies at the bottom of this figure. Blue is (0, 0, 255) and it lies at the upper left. Red is (255, 0, 0). Where is it? It is actually hidden in this figure; its on the far side.
There are three more corners, and these are less obvious. Yellow is at the bottom right and it is (255, 255, 0). You can think of yellow light as the sum of red and green light. (Its more complicated than that, but this is not a human vision course.) Magenta is at the top and it is the sum of red and blue. There is one corner left. Where is it? Do you see it? It is there in the top middle: cyan. It is the sum of blue and green.
Apart from these corners, there are many other color combinations, namely all the different combinations of R, G, B values.



In the previous slide, we were just thinking about the RGB values at one pixel.   We can instead think about the RGB *channels* of the image.   On the upper left is shown a photograph of a children's toy. Look at the red part of the toy.   What is its intensity in the R, G, and B channels? As expected, it has a large intensity in the R channel on the upper right.

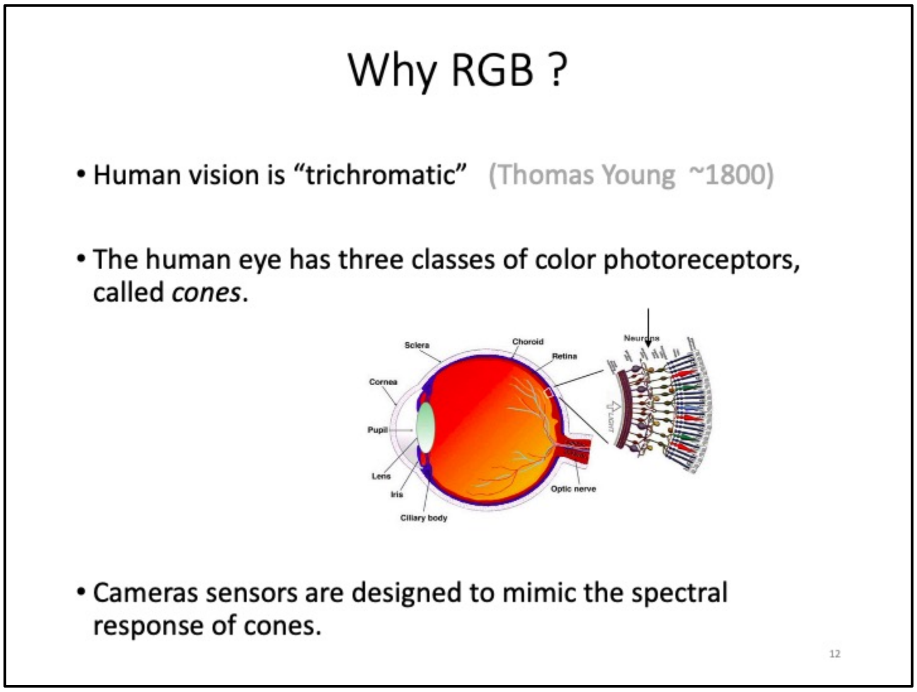
But it has a very low intensity in the G and B channels.   Similarly, if you look at the green part - -the front bumper of the train – then you see it has a large intensity in the G channel but it has a lower intensity in the R channel and B channel.

The colors in this image and children's toys in general tend to be closer to the RGB and Cyan/magenta/yellow corners of the color cube shown on the previous slide.   We say that these colors are very "saturated". For many natural images, however, the colors are often more neutral (black, grey, white).   Let's take another example on the next slide.



In more typical examples, the colors are closer to neutral and so the RGB channels are more similar to each other.

Here is an image taken with Mike Langer’s cell phone camera. It is his desk top, and the thing in the foreground is my coffee mug. The image on the left obviously does have SOME color. But the colors are relatively neutral. The R, G, B channels have some differences, but the intensity differences across the R, G, and B are barely noticeable to you when you look at them – unlike the example of children’s toys with saturated colors.



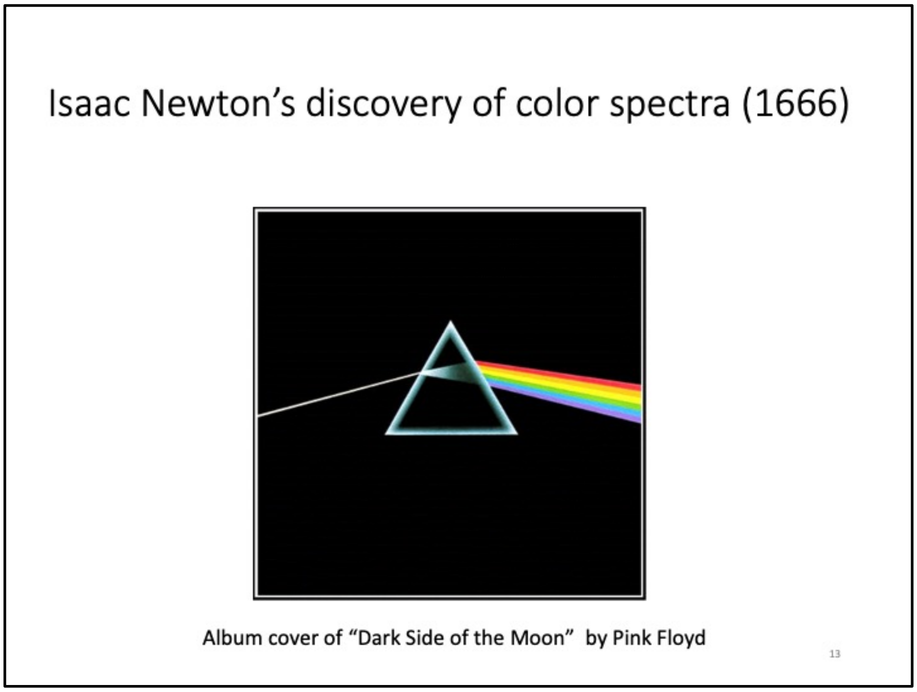
Why RGB ?

- Human vision is "trichromatic" (Thomas Young ~1800)
- The human eye has three classes of color photoreceptors, called *cones*.

Sclera
Choroid
Retina
Neurons
Cornea
Pupil
Lens
Iris
Ciliary body
Optic nerve
LIGHT

- Cameras sensors are designed to mimic the spectral response of cones.

Why do images have three components per pixel? Why RGB? It has been known for 200 years that there is something three dimensional or "tri-chromatic" about color. It has nothing to do with space that live in being three dimension. Rather it has to do with properties of the sensory nerve cells in our eyes. These cells are called "photoreceptors". These cells have a conical shape and are thus called cones. The three different types of cones have different sensitivities, which I'll describe over the next few slides. These sensitivities are important since camera sensors are – in a certain technical sense -- designed to mimic the sensitivities of cones.

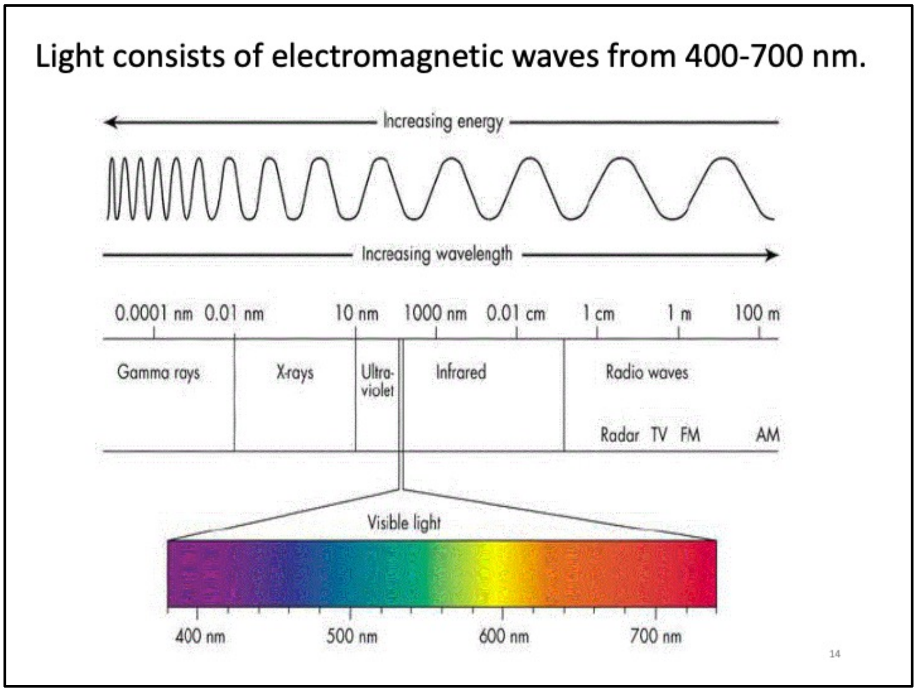


Isaac Newton’s discovery of color spectra (1666)

Album cover of “Dark Side of the Moon” by Pink Floyd

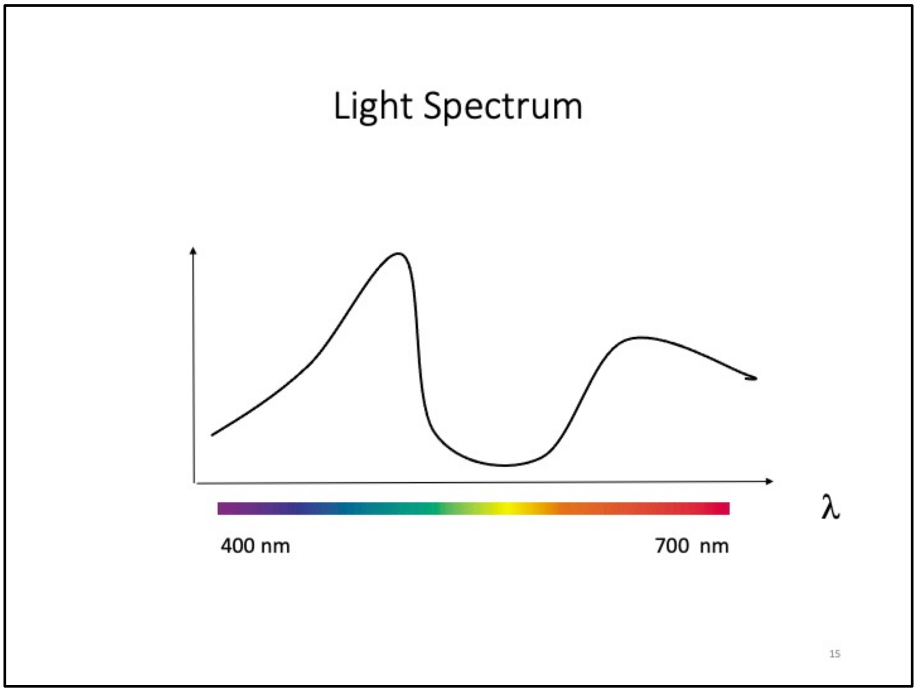
13

To understand how RGB is related to light that a camera measures or that an eye measures, we need to think about the basics of what light is. As Isaac Newton discovered over 350 years ago, a beam of white light consists of superimposed beams of “colored” light. Newton showed this by separating these beams with a prism. You have hopefully all learned about Newton’s discovery at some point in your science education.



Light consists of electromagnetic waves from 400-700 nm.

Light consists of electromagnetic waves from 400-700 nm. That is, the photoreceptors in our eyes only absorb and respond to electromagnetic waves over this tiny interval. If the short waves are seen in isolation then we see them as violet or blue. If the long waves are seen in isolation then we see them as red. For intermediate wavelengths seen in isolation, we see them as other specific colors ranging from green to yellow to orange.

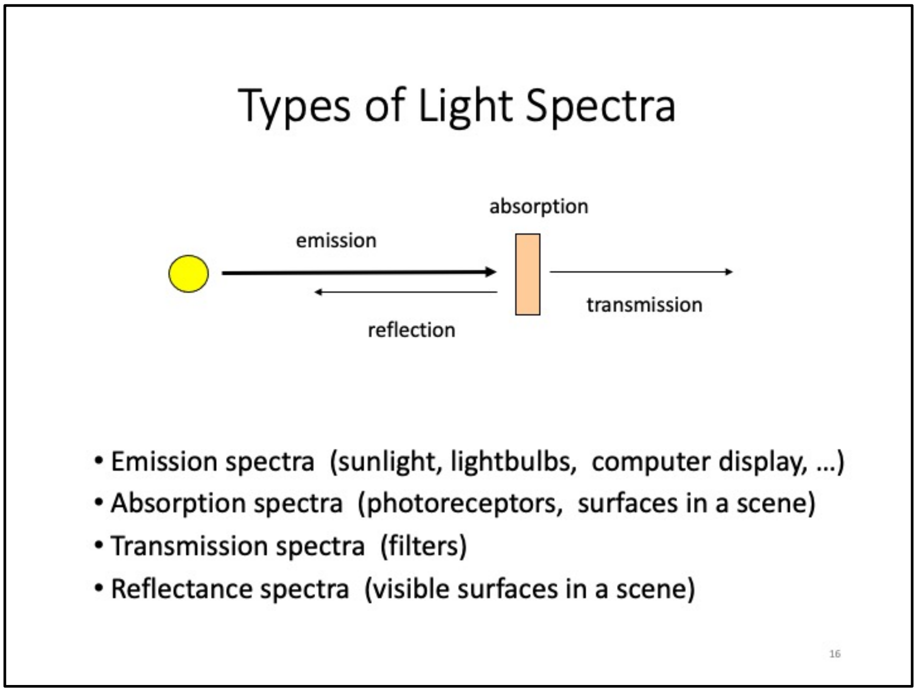


Light Spectrum

Natural light sources such as the sun or a burning filament of a tungsten light bulb do not emit light of single wavelengths in isolation though.   Rather, they emit light over a range of frequencies.   This light is then reflected off of surfaces around us, and some of it is eventually measured by eyes or cameras.

When we talk about a *spectrum* of light, we mean some function of wavelength between the range of visible light.

Here in this figure, I have not labelled the vertical axis. That’s because a spectrum is a general idea: there are several different types of light spectrum, as I’ll mention on the next slide.   For the case of an emission spectrum, the vertical axis would say be the intensity of light at each wavelength.



Types of Light Spectra

- Emission spectra (sunlight, lightbulbs, computer display, …)
- Absorption spectra (photoreceptors, surfaces in a scene)
- Transmission spectra (filters)
- Reflectance spectra (visible surfaces in a scene)

Several types of light spectra will come up today, so its worthwhile to distinguish them. To give these spectra names, we need to review how light travels through a scene and is eventually measured in an image.

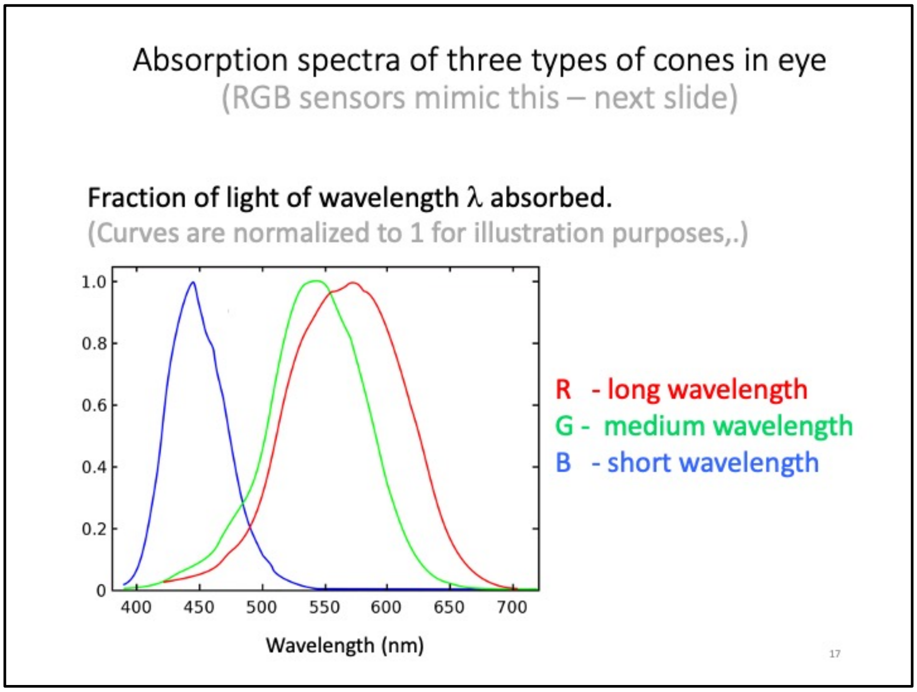
Light is first emitted from a light source. It then arrives at some surface or volume. There it is either absorbed, transmitted through, or reflected. The only thing that may be new to you here is to think of this process wavelength-by-wavelength. How much is emitted/absorbed/transmitted/reflected at each wavelength? That's what we mean by a *spectrum*.

So, an emission spectrum indicates the distribution of the power of the light emitted by a light source at each wavelength. This could be light emitted from the sun, or from a lamp, or a computer display.

An absorption spectrum indicates the fraction of light that arrives at a surface or volume and is absorbed, again as a function of wavelength. This spectrum will depend on the material of the surface or volume.

Any light that is not absorbed is either transmitted through the surface or volume or is reflected. A transmission spectrum indicates the fraction of incoming light that is transmitted.

A reflectance spectrum indicates the fraction of light that is reflected back. All of these spectra are, by definition, functions of wavelength.



Absorption spectra of three types of cones in eye

(RGB sensors mimic this – next slide)

Fraction of light of wavelength λ absorbed.

(Curves are normalized to 1 for illustration purposes,.)

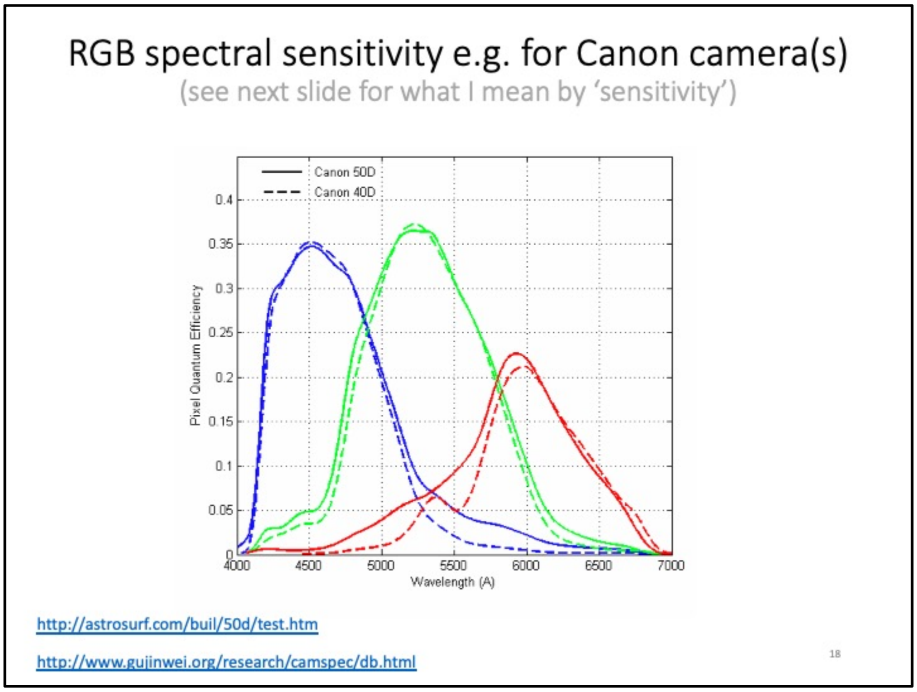
R - long wavelength

G - medium wavelength

B - short wavelength

I could show you examples of all four types of spectra. But let's just consider absorption spectra, since this lecture is not really about what happens out there in a scene as light bounces around between surfaces, but rather it is about RGB images.

In human vision, the three types of photoreceptor cones in the eye have different absorption spectra, which are shown here. (ASIDE: the difference is due to having different pigment proteins in the cells.) We refer to the three types of cones as long, medium, and short wavelength cones. Notice that they have been normalized to 1 here. Note that each of the spectra peak at different wavelengths. By definition, the short, medium, and long wavelength cones have absorption spectra peaks at short, medium and long wavelengths, respectively.
These three different absorption spectra are the reason why color vision is tri-chromatic. The fact that the peaks are found at wavelengths whose pure light appears blue, green, and red is the reason why we talk about RGB.



RGB spectral sensitivity e.g. for Canon camera(s)

(see next slide for what I mean by 'sensitivity')

http://astrosurf.com/buil/50d/test.htm

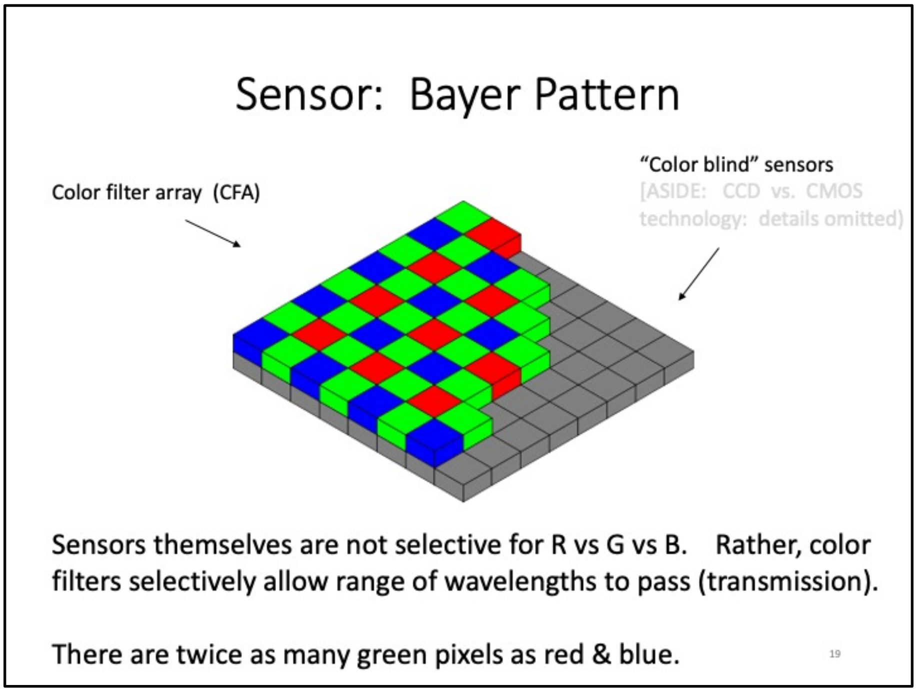
http://www.gujinwei.org/research/camspec/db.html

18

Camera sensors are designed to roughly to mimic human cone absorption spectra. Here we see examples for two Canon cameras. Different cameras will have different absorption spectra.

As with human cones, the photoreceptors are more sensitive to either short, medium, or long wavelengths. The curves are not exactly the same as the those of the human cones, but that's ok. The three curves still have peaks at the short, medium, and long wavelengths.

Note that I use the word "sensitivity" in the title, rather than absorption spectrum. To understand why, we need to know a bit more about how the sensors work.



Sensor: Bayer Pattern

Color filter array (CFA)

“Color blind” sensors
[ASIDE: CCD vs. CMOS technology: details omitted)

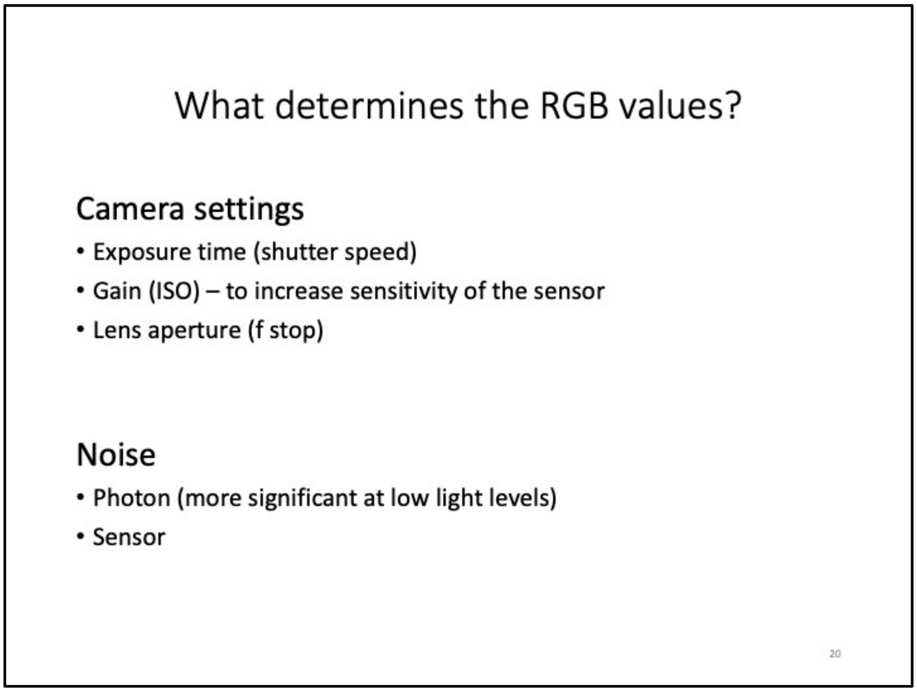
Sensors themselves are not selective for R vs G vs B. Rather, color filters selectively allow range of wavelengths to pass (transmission).

There are twice as many green pixels as red & blue.

19

The sensors themselves form a 2D array. The sensors are the gray squares shown in the figure, below the colored tiles. The sensors themselves are color blind. The color sensitivity comes from a “color filter array (CFA)” that is placed above the sensor. These red, green, and blue filters allow different wavelengths to selectively transmit the light to the sensors.

You’ll also note that there are twice as many green color filter array elements as there are red or blue ones. This design is called a Bayer pattern, after a person named Bayer who first proposed it. The reason for having more for green is quite technical and has to do with properties of human color, so I’ll skip the details. But I will have more to say about the Bayer pattern later this lecture, and how a camera treats the interleaved R, G, and B channels of an image.



What determines the RGB values?

Camera settings

- Exposure time (shutter speed)
- Gain (ISO) – to increase sensitivity of the sensor
- Lens aperture (f stop)

Noise

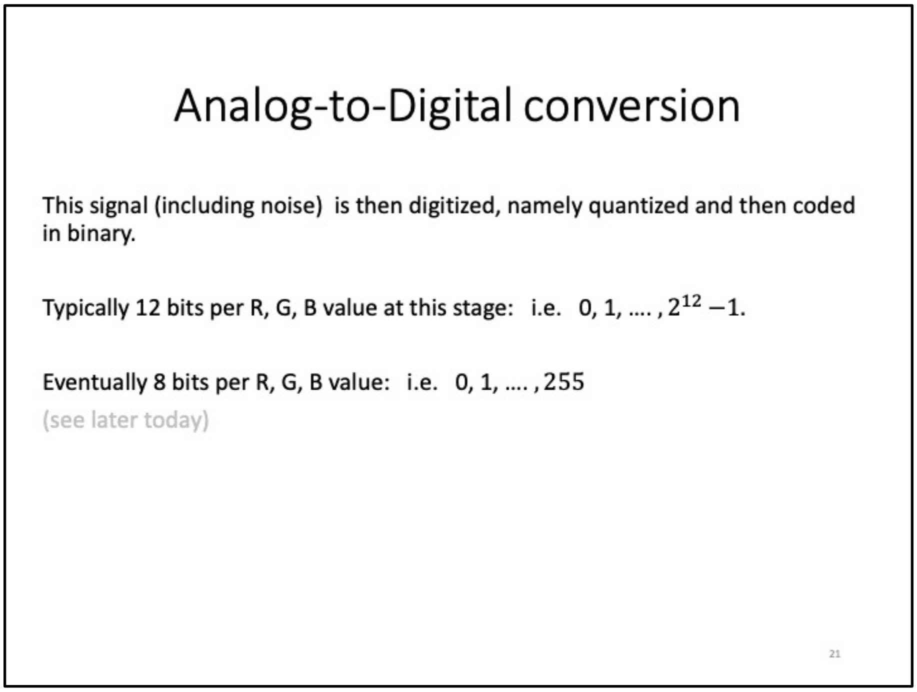
- Photon (more significant at low light levels)
- Sensor

20

So, some light spectrum arrives at the camera from each direction. Then light gets absorbed by the sensors and we get an RGB value at each pixel. What determines the RGB values? There is a lot to say about this. We'll come back to these details in November, but let me just mention a few details here – especially for those of you who have some familiarity with images.

First of all, the camera has certain settings and, in particular, users of an SLR camera need to know these settings.
The user can modify the exposure time of an image, which is the amount of time that the sensor is exposed. On a typical SLR, this can range from several seconds down to 1/1000 of a second. Obviously, the more time the sensor is exposed, the more light is absorbed and higher is the intensity value recorded.
The user can also vary the sensitivity of the sensor – called the gain or "ISO". If you are shooting a scene under dim lighting conditions, you want to increase the gain.
The user can also vary the size of the lens aperture. This is like the pupil in your eye. Under dim conditions, your pupil opens more to let more light in and under bright light it closes down. Similarly with photography, by increasing or decreasing the size of the aperture, we increase or decrease the amount of light arriving at the sensor. But as we'll see in November, the aperture also affects how well the images are focused.

Images also have noise. The noise can either come from the light source itself, namely light sources emit photons randomly. At low light levels, this variation in intensity can be significant. There is also noise in the sensor, so that even if there were no noise in the light sources and if you would take two images one after the other and nothing has changed in the scene, you would typically get slightly different images. I'll say a bit more about noise later in the lecture.



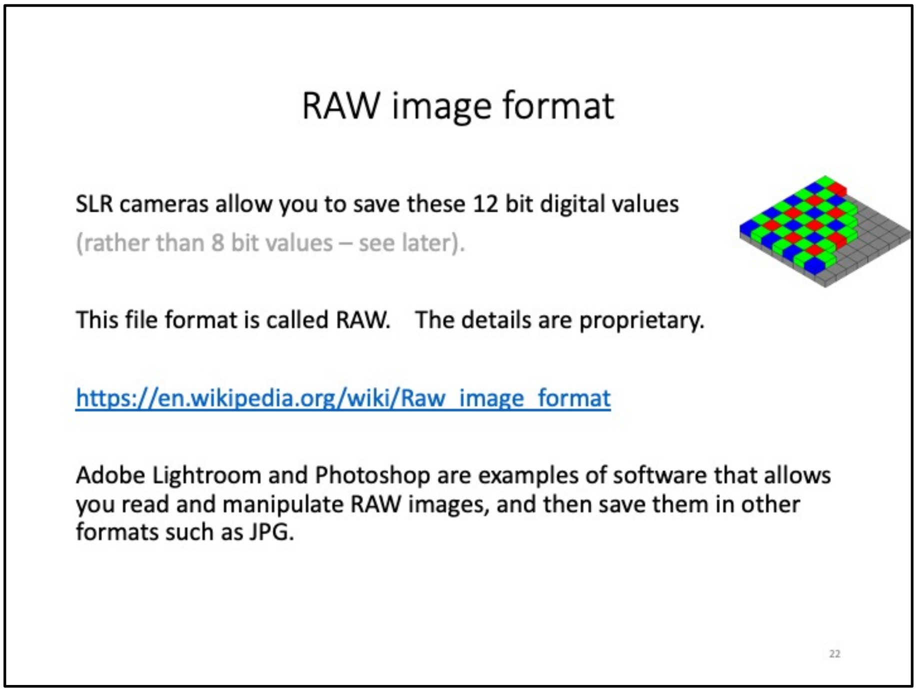
Analog-to-Digital conversion

This signal (including noise) is then digitized, namely quantized and then coded in binary.

Typically 12 bits per R, G, B value at this stage: i.e. 0, 1, …. , $2^{12} - 1$.

Eventually 8 bits per R, G, B value: i.e. 0, 1, …. , 255
(see later today)

So, for any RGB image, some amount of light energy (or number of photons) is measured at each RGB pixel element. We say this is an analogue signal since we think of it as a real number intensity. This analogue signal is then digitized, namely it is quantized into one of a limited set of values. This is called Analog-to-Digital conversion.
Typically the signal is stored as a 12 bit number, so we have 2^12 possible values. So there are 12 bits initially stored for each R,G, or B tile on the sensor. As we will see in coming slides, this values later on become 8 bits.



RAW image format

SLR cameras allow you to save these 12 bit digital values (rather than 8 bit values – see later).

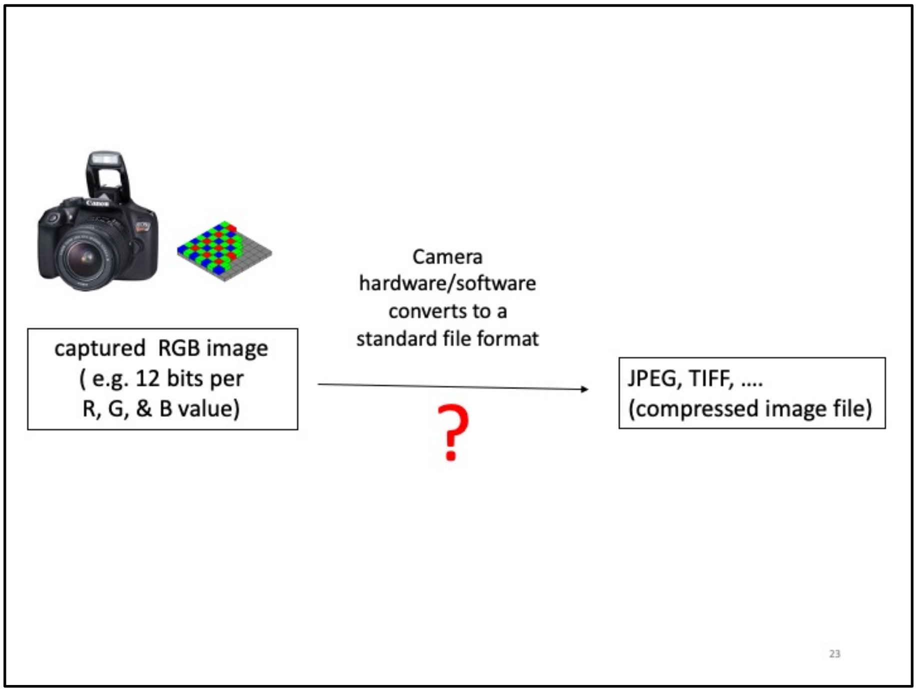
This file format is called RAW. The details are proprietary.

https://en.wikipedia.org/wiki/Raw_image_format

Adobe Lightroom and Photoshop are examples of software that allows you read and manipulate RAW images, and then save them in other formats such as JPG.

SLR cameras allow you to save these 12 bit digital values. This file format is called RAW. The details of the file format are *not* standard. Rather, they depend on the camera. So they are proprietary.

Image manipulation software such as Adobe Lightroom and Photoshop allows you read in RAW images, manipulate them, and then save them in other formats such as JPG. Since you'll be dealing with JPG files in this course, let's talk a bit now about how this works.



captured RGB image
(e.g. 12 bits per
R, G, & B value)

Camera
hardware/software
converts to a
standard file format

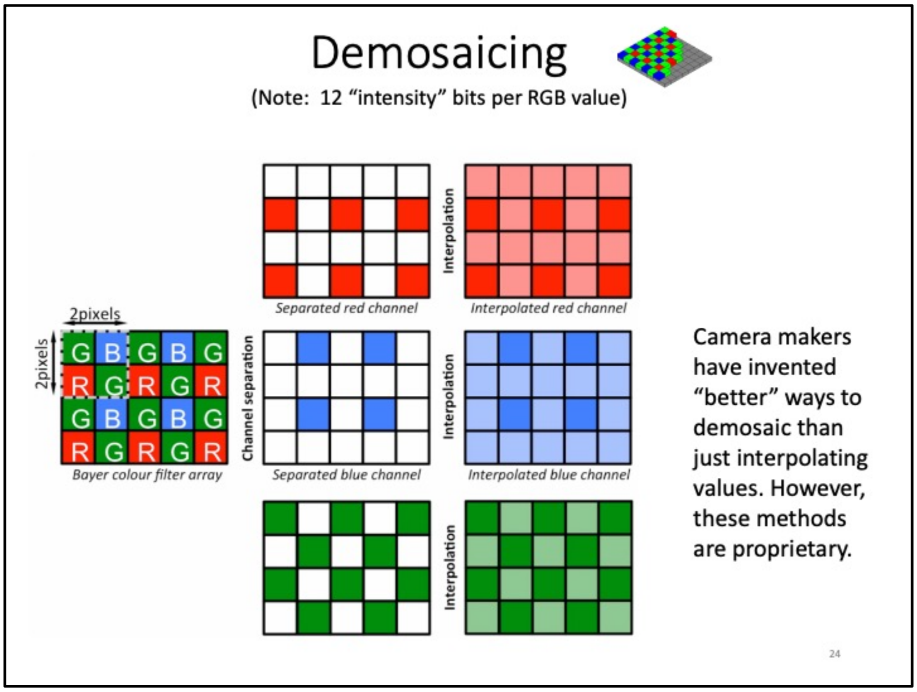
?

JPEG, TIFF, ….
(compressed image file)

23

What happens when a camera captures an RGB image? Here I’m showing an SLR camera, but the basic pipeline I’m about to describe is basically the same for point and shoot cameras and for cell phone cameras too.

The camera converts the captured RGB image to a standard file format such as JPEG or TIFF which is stored in memory, for example, on an SD card. It may also store it as RAW, but let’s say the images are stored as JPEG or TIFF. These are compressed files, so they take fewer bits than the raw files. We’re not going to talk about the JPEG compression standard here. Instead we’re going to talk about the what needs to happen *before* the image gets compressed into JPEG.



Demosaicing

(Note: 12 "intensity" bits per RGB value)

Camera makers have invented "better" ways to demosaic than just interpolating values. However, these methods are proprietary.

24

The first step is to deal with the Bayer pattern. Recall that there are twice as many sensors with the green filter as there are for the red or blue. Each 2x2 square of sensors has 2 greens and 1 blue and 1 red.

Now you are wondering: how many pixels are defined on a square sensor grid? Surprisingly, each sensor element corresponds to one pixel! So the 4x5 grid on the left corresponds to 20 pixels.

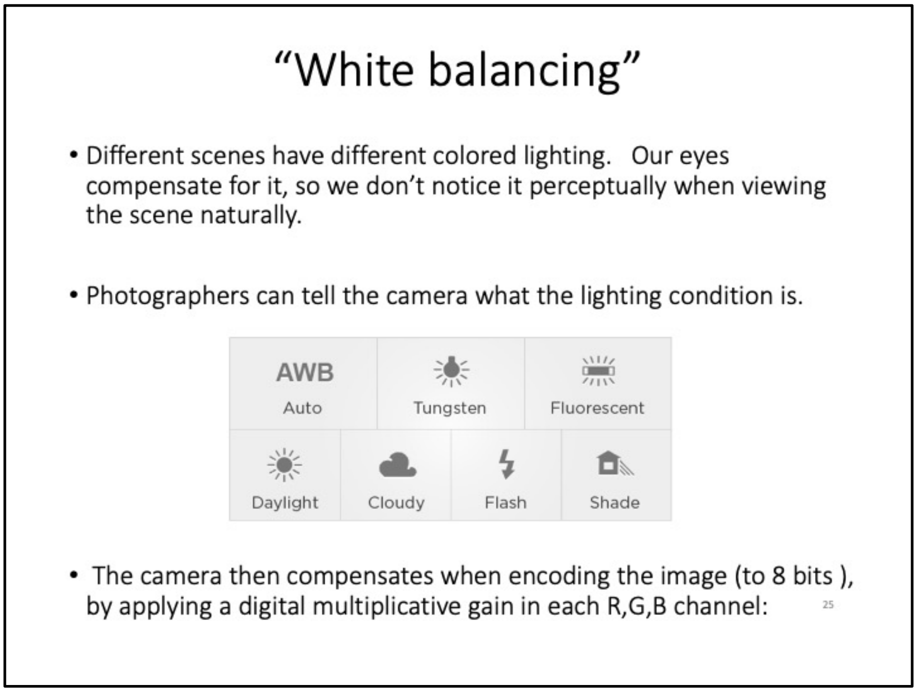
The Bayer pattern is also sometimes called a mosaic, and so the RGB tiles need to be "de"-mosaiced. That is what is illustrated in the second column. The RGB are simply pulled out and the values are written into corresponding locations in three new grids.

Then, the camera interpolates between the values within each of these grids. In this figure, one ends up with a 4x5 RGB image, as I wrote above

How does this interpolation work? This is proprietary. It isn't simply a matter of taking the local average. To appreciate why, consider that the image may have noise in it, so the camera might try to reduce this noise at an early stage.

Anyhow, let's say we have 12 bits per RGB value at this stage. Think of this as the demosaiced RAW image.

[If you would like to know more about demosaicing and see some examples, see https://en.wikipedia.org/wiki/Demosaicing.]



“White balancing”

- Different scenes have different colored lighting. Our eyes compensate for it, so we don’t notice it perceptually when viewing the scene naturally.
- Photographers can tell the camera what the lighting condition is.

AWB
Auto
Tungsten
Fluorescent
Daylight
Cloudy
Flash
Shade

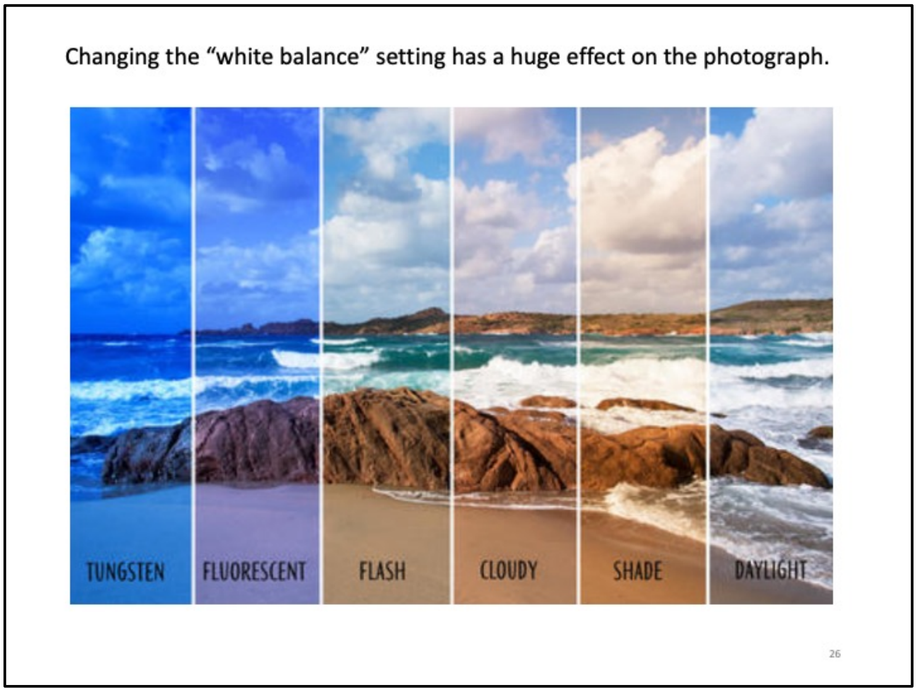
- The camera then compensates when encoding the image (to 8 bits), by applying a digital multiplicative gain in each R,G,B channel:

As I mentioned earlier, different light sources have different spectra. You are not generally aware of this, unless you happen to be professionally trained. Rather, your eyes generally compensate for the light source color, just as they compensate for the overall light level to some extent. However, the spectrum of the light source has a huge effect on the RGB values that are stored in a raw image.

For example, if you have a reddish light source, then your R values are going to be much larger than your G and B values. In particular if you have a white (or grey) object in the scene – and by that I mean an object whose surface material reflects the same percentage of light at all wavelengths– then that object’s image will have larger R values than G or B, if the light source is red. That’s generally not what you want. You want white or gray objects to have roughly the same RGB values.

When you are taking pictures, you can tell the camera what the lighting conditions are. (This is assuming that you have time to adjust this setting, obviously.) Then, you take the picture and the camera encodes the image as a JPG, and it takes account of the dominant lighting literally by scaling the values in the R, G, B channels. For example, if you tell the camera that the light source is a tungsten light bulb (which emits reddish light), then the camera would multiply all the red values by some number that is *less than 1*, in order to compensate for the reddish light. Each of the seven settings shown in the slide has a particular scale factor for R, G ,and B channels.

This process of rescaling the RGB channels is called “white balancing”. The idea here is that we want whites object to appear white, rather than to have its RGB values biased by the color of the light source.   (Same for gray objects…)



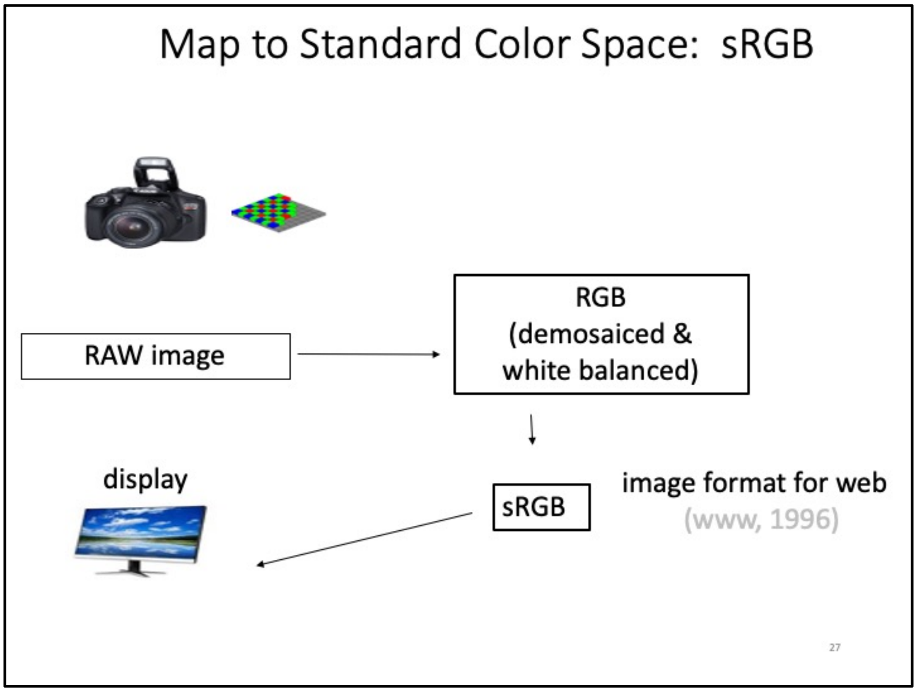
Changing the “white balance” setting has a huge effect on the photograph.

26

Changing the white balance has a huge effect on the photograph. Here we have one scene and the figure illustrates how changing the settings of the white balance affects the result. If you set the camera to tungsten lighting as on the left, then it will compensate by multiplying the R channel by a smaller number to cancel the light source effect, which is assumed to be dominated by long wavelengths (red). This makes the image appear very blue. Clearly this is not what you want for the scene shown.

Whenever you take a picture with a digital camera, the camera will do some sort of white balance (either automatic white balance or some WB setting by the photographer).

Also note that, even if a camera leaves the RGB channels exactly as they were recorded (quantizing from 12 bits to 8 per channel, by chopping off the least significant 4 bits), you should not expect the RGB values to be identical to the physical spectrum. The reason is that the color filters themselves implicitly apply different weights to different parts of the spectrum. This has a similar effect of “white balancing” but it is not white balancing per se since it happens prior to the image being captured by the sensors.



Map to Standard Color Space: sRGB

RAW image

RGB
(demosaiced &
white balanced)

sRGB

image format for web
(www, 1996)

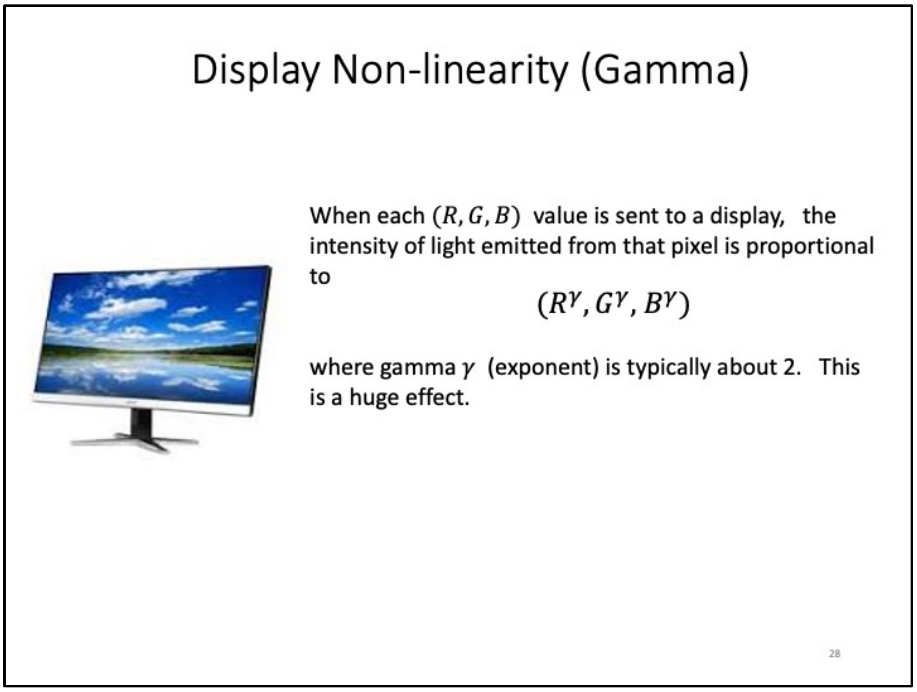
display

27

So now we’ve taken the RAW image, demosaiced and white balanced. So far so good. However, it turns out this is not good enough. The RGB image will be displayed on a device that doesn’t know about the camera sensor’s properties. So it turns out (and most of the details are omitted here) that, despite the white balancing, the RGB values still need to be transformed again. They need to be transformed to a *standardized* color space. One common color space is called sRGB. This was a standard invented in the 1990’s when the world wide web was created, and images needed to be displayed in web browsers.

There are two aspects to sRGB. First, the RGB values are transformed by reweighting each RGB value to some (arbitrary) standard. The second aspect, which is more interesting, is that a non-linear mapping is applied to the RGB values. To understand this non-linear map, I need to tell you about “gamma”.

In the box that says RGB (demosaiced and white balanced), you can still think of this as a 12 bit representation. The conversion to 8 bits only needs to come when we want the RGB values in a standard image format e.g. sRGB.



Display Non-linearity (Gamma)

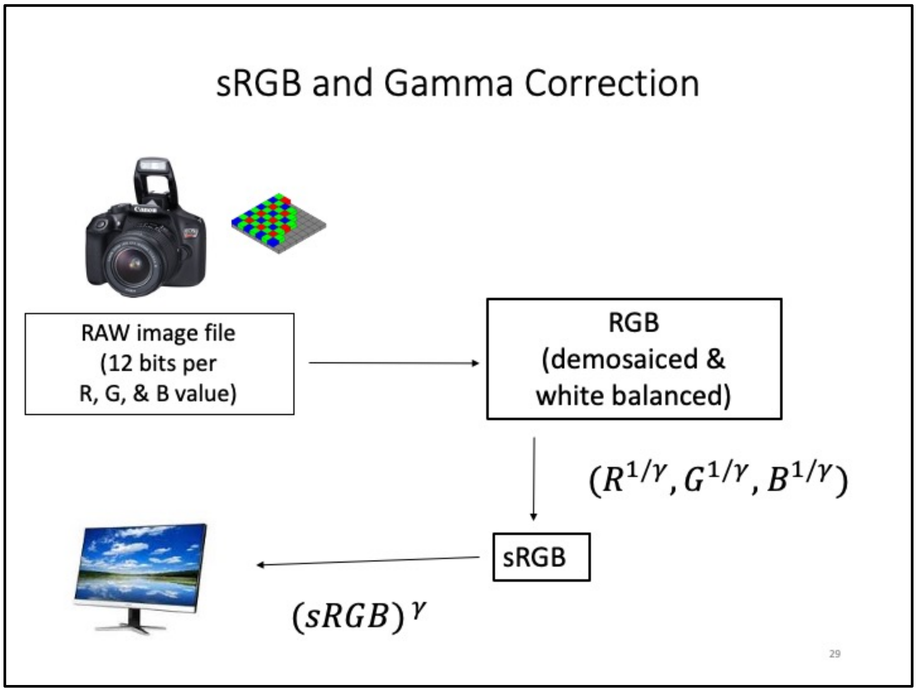
When each (R, G, B) value is sent to a display, the intensity of light emitted from that pixel is proportional to

$$(R^{\gamma}, G^{\gamma}, B^{\gamma})$$

where gamma γ (exponent) is typically about 2. This is a huge effect.

When each RGB value in the image is sent to a display, such as your computer screen, the values are sent to RGB emission pixels on the display. The RGB emitters have their own emission spectra. The intensity (a scale factor) for the spectrum emitted at each display pixel is surprisingly not proportional to the RGB values. Rather, the intensity is proportional to the RGB values raised to some power (called the “gamma” of the display). The gamma is typically about 2, but it varies for different manufacturers.

This gamma transformation has been used in electronic displays since the first televisions in the 1940s. There are good reasons for it which have to do with some important properties of human vision. (Details omitted.)



sRGB and Gamma Correction

RAW image file (12 bits per R, G, & B value)

RGB (demosaiced & white balanced)

$(R^{1/\gamma}, G^{1/\gamma}, B^{1/\gamma})$

sRGB

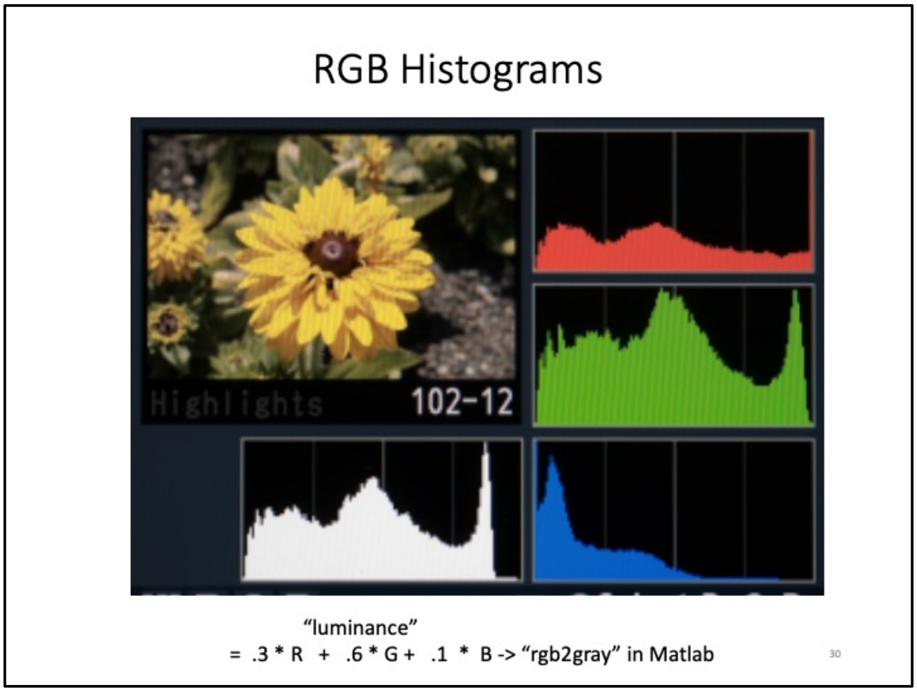
$(sRGB)^{\gamma}$

29

To correct for the gamma effect of the display, one needs to mathematically invert it, prior to sending the values to the display. This inversion is called gamma correction, and it is part of the sRGB standard. So, after the camera processor performs the demosaicing and white balancing, it then raises the RGB values to the power "1/gamma". That way, when the display subsequently raises the values of the sRGB image to the power gamma, the intensities that actually are displayed are proportional to the demosaiced and white balanced RGB values. For example, (R^(1/gamma))^gamma = R, and similarly for G and B.

Now you will ask, why bother having a gamma in the display if you are just going to correct for it? The answer is that the RGB image and display doesn't deal with continuous intensities, but rather it only deals with 256 possible values of R, G, and B. The gamma trick essentially allows you to represent the lower intensity values with more precision than the upper intensity values. And this, it turns out, is better for displaying to the human vision which has higher precision at lower intensity values than upper intensity values. If you would like to read up on this, you could see https://www.cambridgeincolour.com/tutorials/gamma-correction.htm.

Bottom line: the image in sRGB form has RGB values that have been raised to 1/gamma, as defined by the sRGB standard. So these RGB digital values that you would work with in a computer vision application – and that we'll be working with in this course when we deal with images that have been captured with a camera-- are quite different from the values in the RAW file that was captured.



RGB Histograms

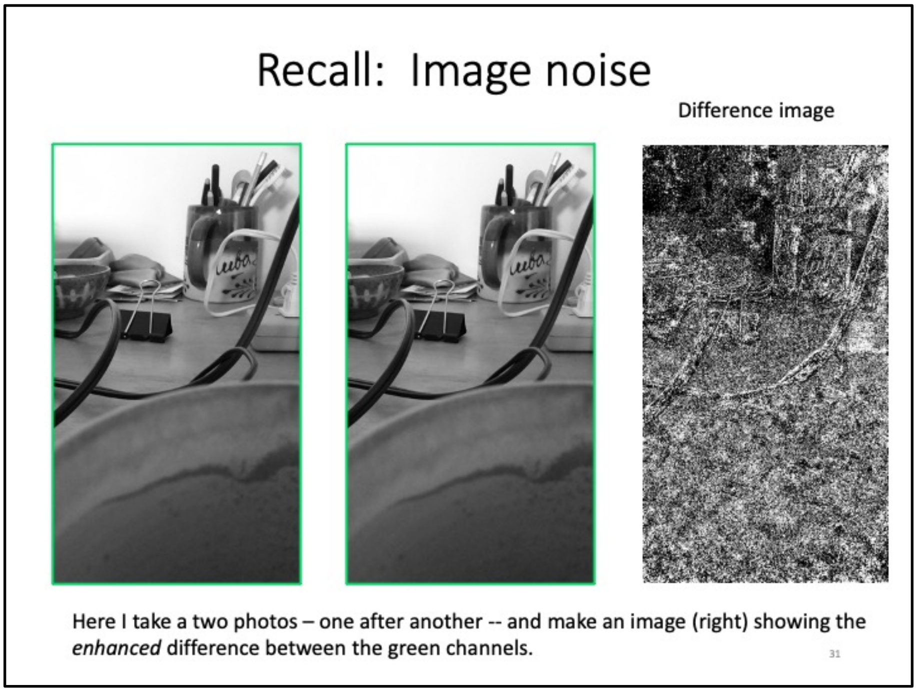
"luminance"
= .3 * R + .6 * G + .1 * B -> "rgb2gray" in Matlab

30

We will use the word "histogram" many times in this course. An RGB histogram shows the frequencies of each R, G, or B values in an image, that is, the number of pixels that have each value of R, each value of G, each value of B. See the right three panels on the slide. This is what is shown in the LCD panel on the back of a typical SLR cameras. Note there is also a histogram that takes the weighted average of the RGB values at each point. This average of RGB values at a pixel is sometimes called the luminance. A typical formula that cameras use to compute luminance values is shown at the bottom of the slide.

Note that if the image is a raw image, then the R,G, and B histogram shows the frequencies of the raw values. However, if the image has already been coded as sRGB, then histogram will show these encoded values, so the values will have been raised to the power 1/gamma already.

As mentioned on the previous slide, in a computer vision application, you will often be using images that have been encoded as sRGB which means that they have this non-linearity built in. Many computer vision researchers are unaware of this or are aware but just ignore it.



Recall: Image noise

Difference image

Here I take a two photos – one after another -- and make an image (right) showing the *enhanced* difference between the green channels.

31

Earlier I mentioned that images contain noise. So let’s come back to that now briefly.

I have taken two photos – one after another – and pulled out the green channels. (I have indicated that these are the green channels by giving them a green frame.) I have then made a new image at the right which is the difference between the green channel images at each pixel. The image on the right is just the first image minus the second image. Since the G intensity differences are relatively small and since I cannot make an image with negative values, I have scaled the differences in intensities and shifted them so that they have values in the range 0 to 255. White is a positive difference, black is a negative difference. Observe that the values change rapidly from point to point.

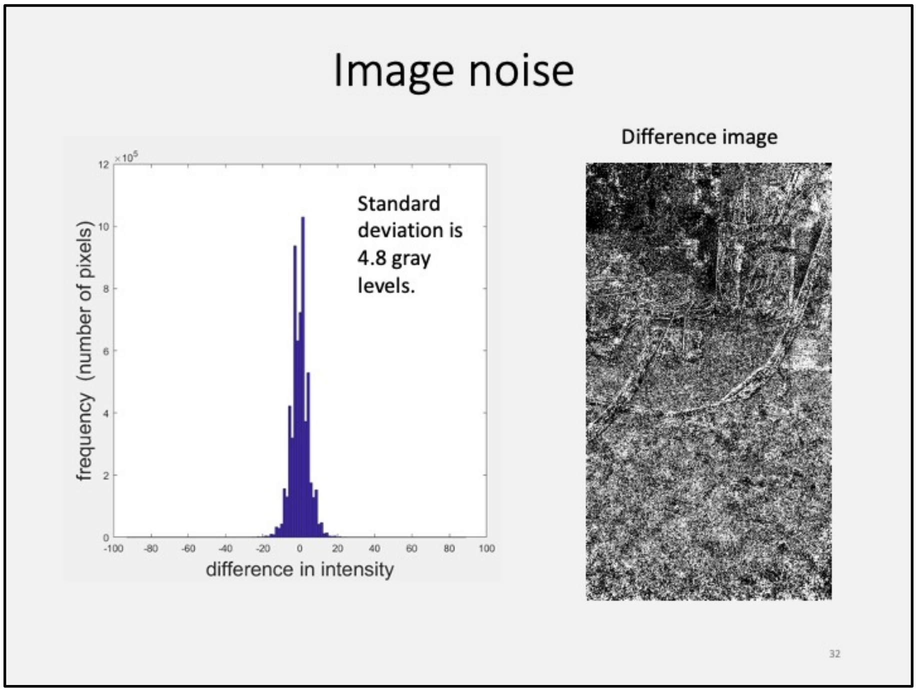
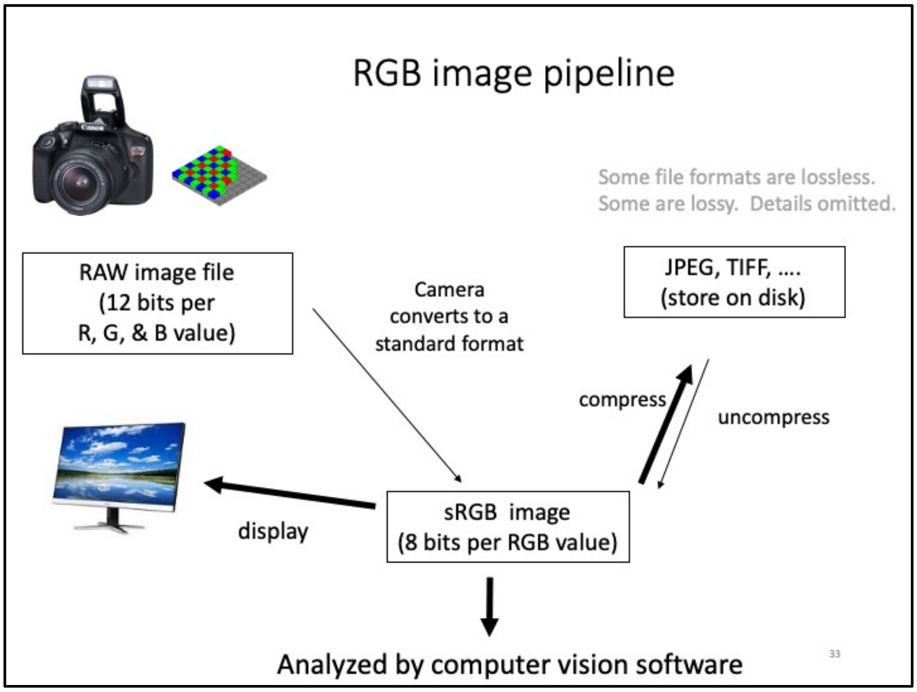


Image noise

Difference image

Standard deviation is 4.8 gray levels.

Here I make a histogram of the image noise values. I plot the number of pixels that have a certain difference in intensity. The peak is roughly at the value 0. But the values at most pixels are non-zero and they are non negligible. The standard deviation is 4.8 gray levels out of 255. That’s really quite a lot. We will discuss noise again next lecture.



RGB image pipeline

Some file formats are lossless.
Some are lossy. Details omitted.

RAW image file
(12 bits per
R, G, & B value)

Camera
converts to a
standard format

JPEG, TIFF, ….
(store on disk)

compress

uncompress

display

sRGB image
(8 bits per RGB value)

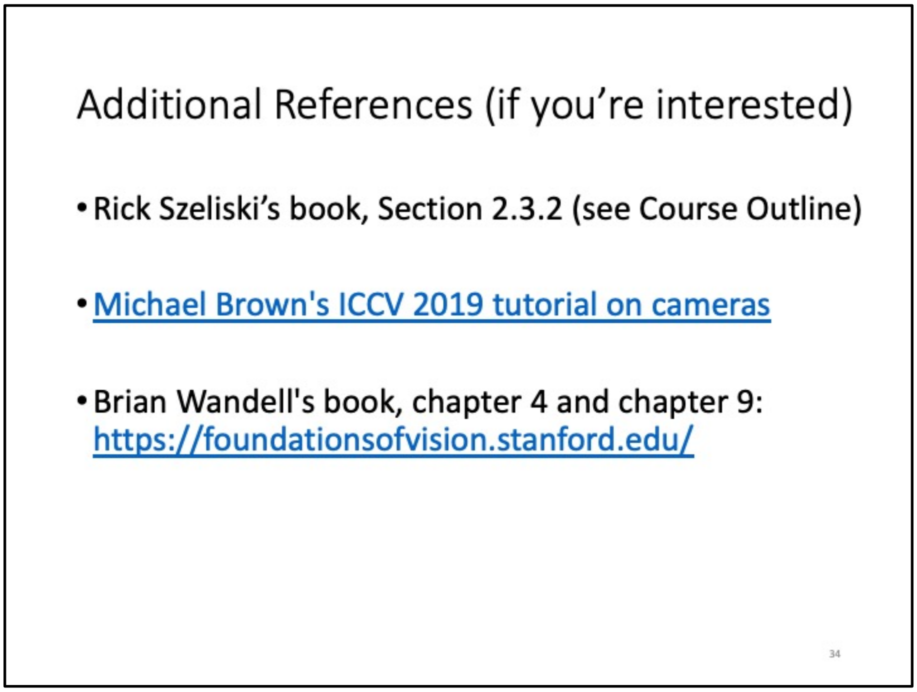
Analyzed by computer vision software

33

Ok, so let’s wrap up. I’ve called this slide the RGB image pipeline. At the upper left is the image capture. The sensors record measurements of light, which are typically 12 bit values at each pixel, arranged in a Bayer pattern. The camera then converts this to a standard image format like sRGB. This requires demosaicing, white balancing, and gamma correction. This image might be displayed on a screen such as your laptop or your cell phone.

The image also will be stored on disk as a JPEG file. I haven’t talked about JPEG compression, and frankly I don’t need to for our purposes. The point is that the RGB values get encoded– typically using far fewer bits in total than the original image. When the image needs to be displayed, it is decoded or decompressed back into sRGB.

Note that the sRGB image also might be analyzed by computer software, namely by computer vision. That’s the arrow shown at the bottom of the slide. In fact, most of the course from here on is about that arrow.



Additional References (if you're interested)

- Rick Szeliski's book, Section 2.3.2 (see Course Outline)
- Michael Brown's ICCV 2019 tutorial on cameras
- Brian Wandell's book, chapter 4 and chapter 9: https://foundationsofvision.stanford.edu/

If you would like to read more about this, then here are a few references.

In particular, some of today's slides were inspired by Michael Brown's ICCV 2019 tutorial on cameras

https://foundationsofvision.stanford.edu/

That's all for now!