11-755— Spring 2021
Large Scale Multimedia Processing

Lecture 4/6

Multimedia capture and storage

Rita Singh

Carnegie Mellon University

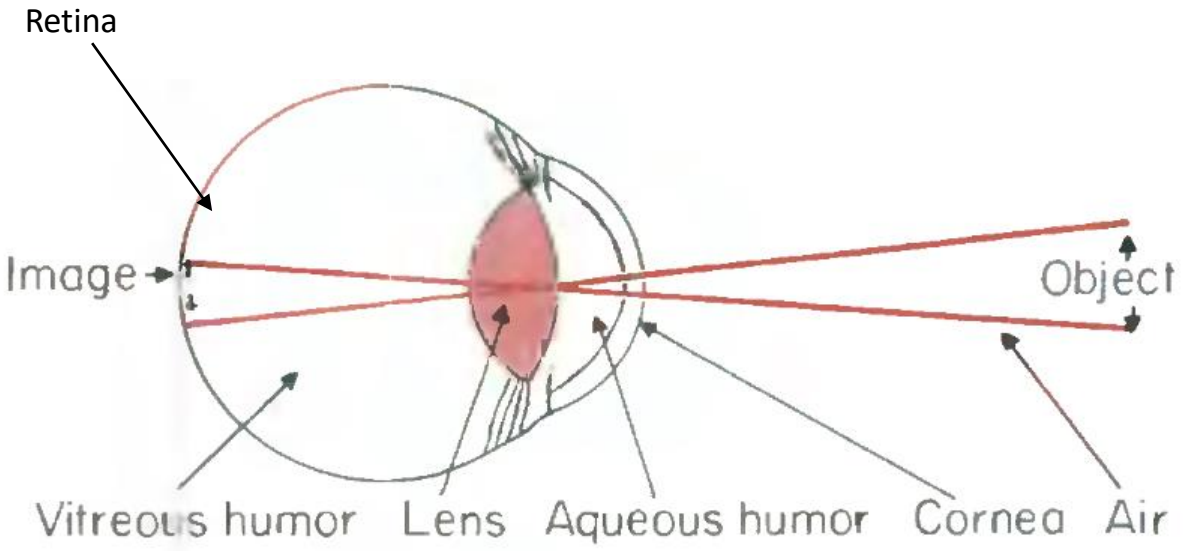
In this lecture

- Digital multimedia: Recording and devices
  - Audio
  - Images
  - Video
  - Text
- Digital multimedia: Processing
  - Audio processing
  - Two generic processing techniques

The human eye

The Eye

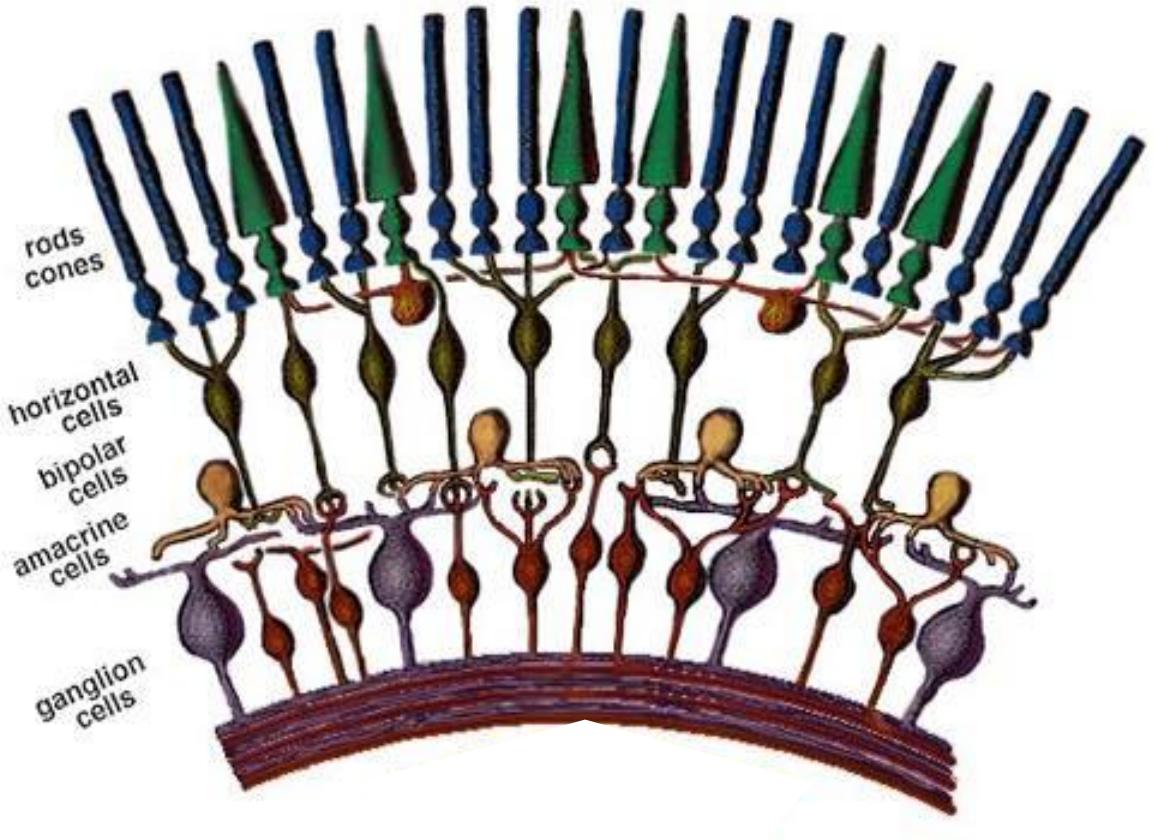
Images are formed on the retina



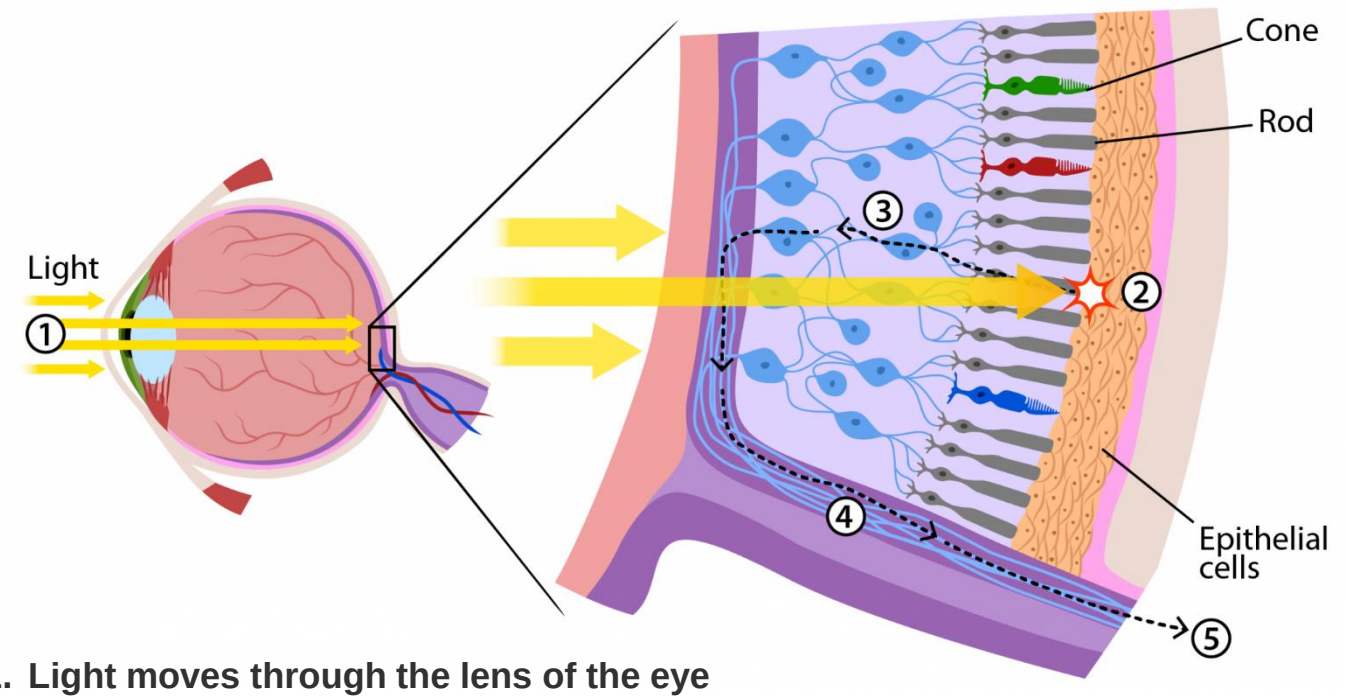
Basic Neuroscience: Anatomy and Physiology Arthur C. Guyton, M.D. 1987 W.B.Saunders Co.

The Retina

The Retina



http://www.brad.ac.uk/acad/lifesci/optometry/resources/modules/stage1/pvp1/Retina.html



1. Light moves through the lens of the eye
2. photons activate the cells. Rods can be activated in low light, but cones require much brighter light (many more photons). Most of the light not absorbed by the rods or cones is absorbed by the epithelial cells behind them.
3. When the signal reaches the inner end (left side) of the rods and cones, the signal is passed to sets of neural cells.
4. The signal moves through neural cells in the optic nerve.
5. The optic nerve will send this information to the brain, where separate signals can be processed so you see them as a complete image.

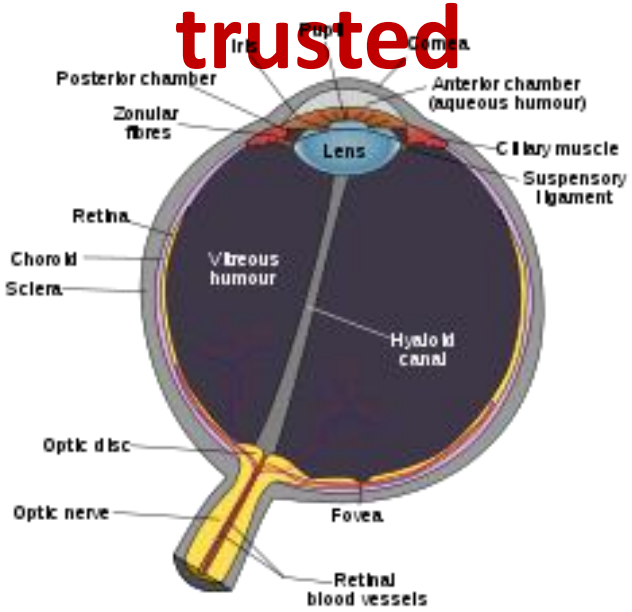
Rods and Cones

Separate Systems

- Rods
  - Fast
  - Sensitive
  - Grey scale
  - predominate in the periphery
- Cones
  - Slow
  - Not so sensitive
  - Fovea / Macula
  - COLOR!

Basic Neuroscience: Anatomy and Physiology Arthur C. Guyton, M.D. 1987 W.B.Saunders Co.

Why eyewitnesses cannot be trusted

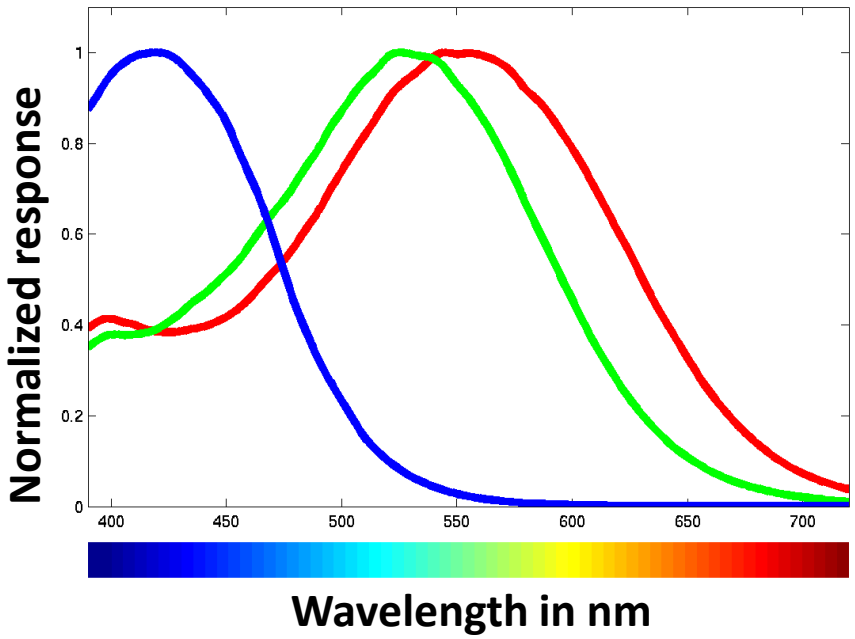


- The density of cones is highest at the fovea
  - The region immediately surrounding the fovea is the macula
    - The most important part of your eye: damage == blindness
- Peripheral vision is almost entirely black and white
- Eagles are bifoveate
- Dogs and cats have no fovea

Why?

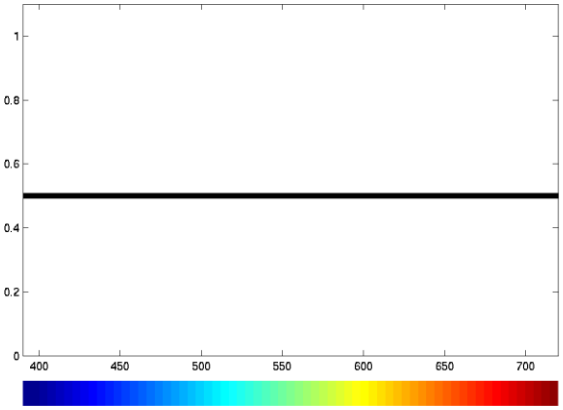
- How our eye perceives colors has a **lot** to do with how digital images are created and stored

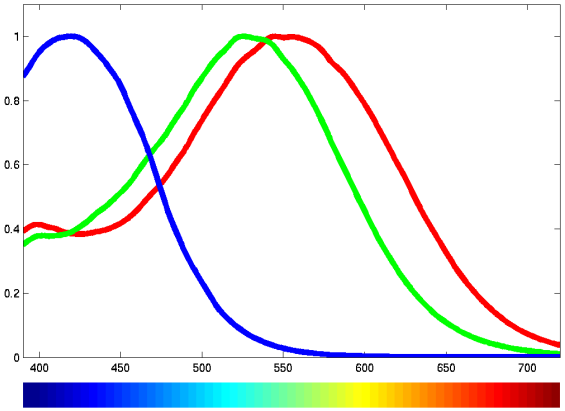
Trichromatic vision (three types of cones)

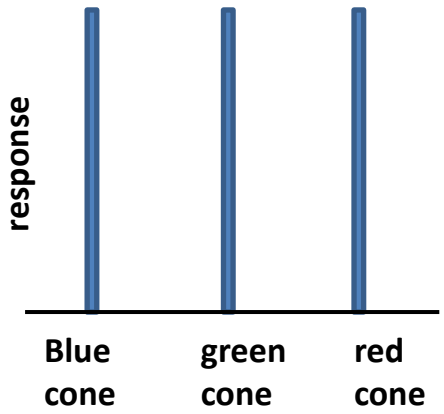


- Each **sensor** responds to an entire range of frequencies
  - E.g "blue" sensors also respond to the "green" and "red" wavelengths
- Difference in response of "green" and "red" sensors is small
  - Varies from person to person
    - Each person really sees the world in a different color
  - If curves are too close: color blindness
    - Ideally traffic lights should be red and blue

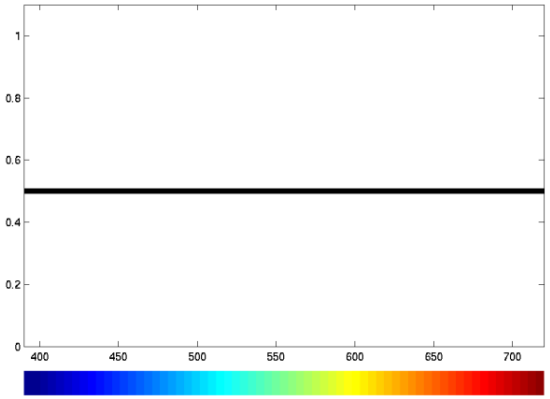
Response to White Light

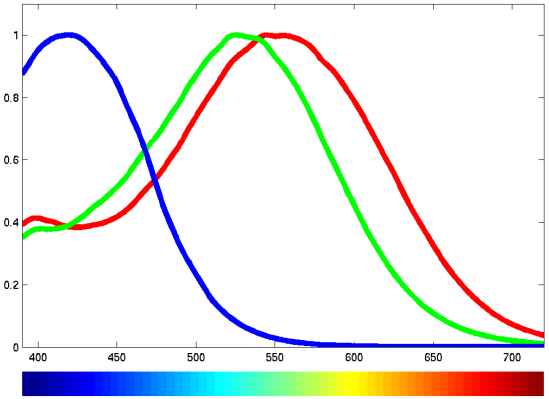


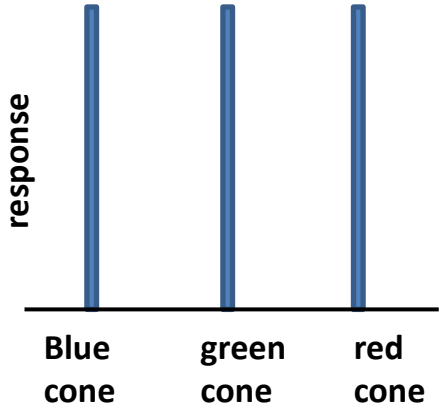


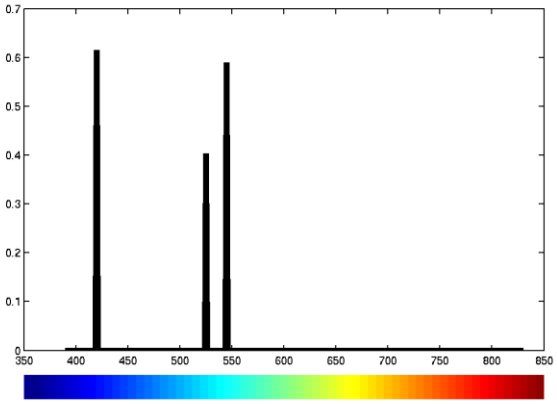


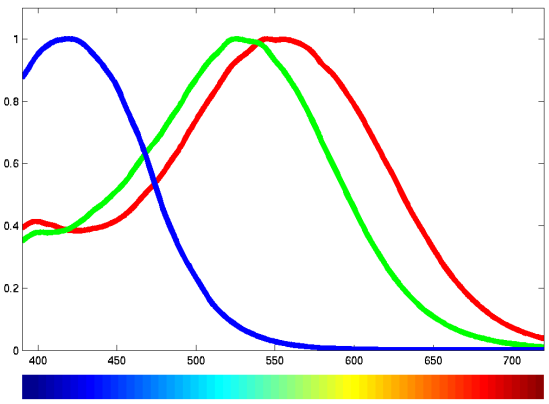
Response to Sparse Light





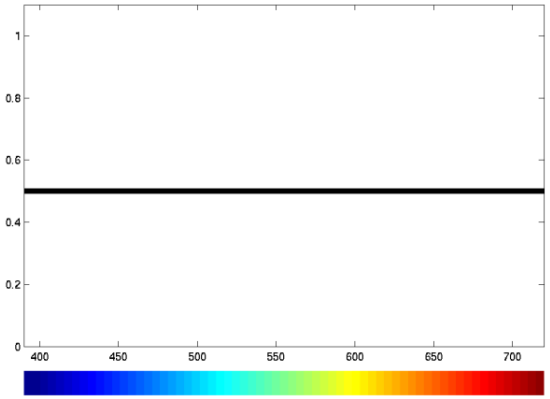


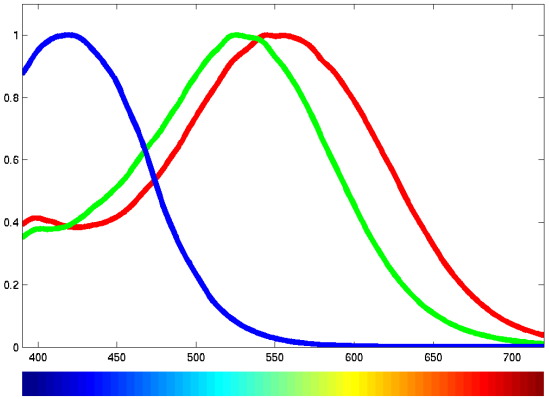


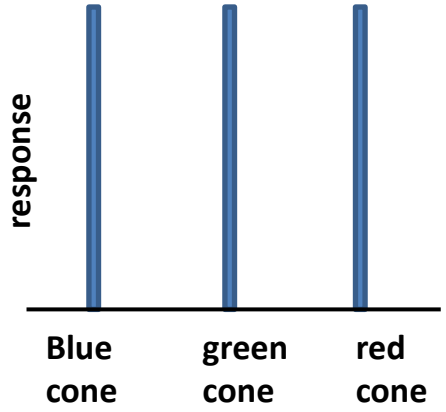


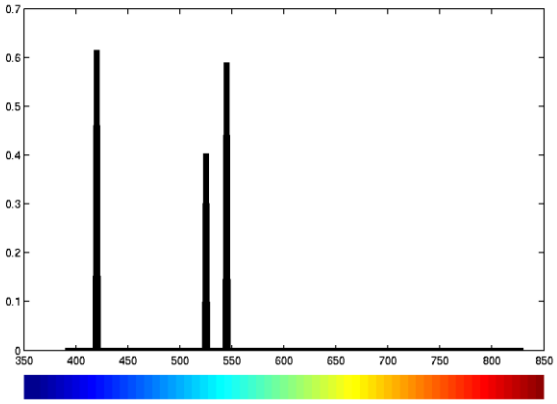
?

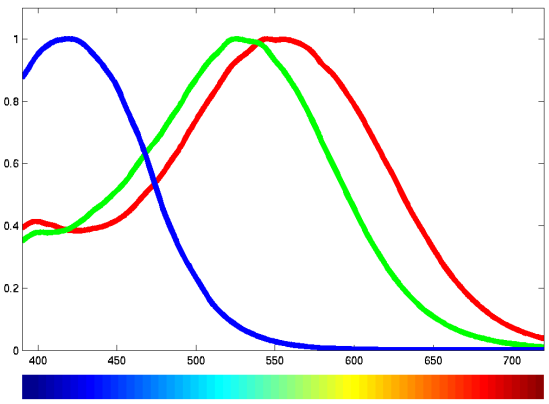
Response to Sparse Light

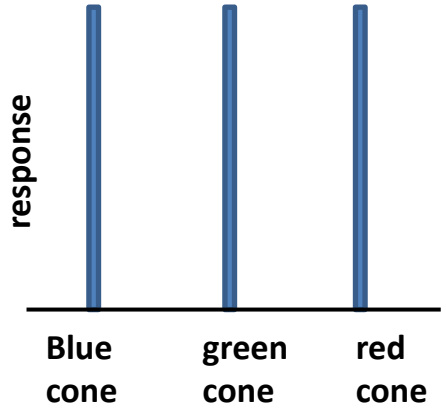




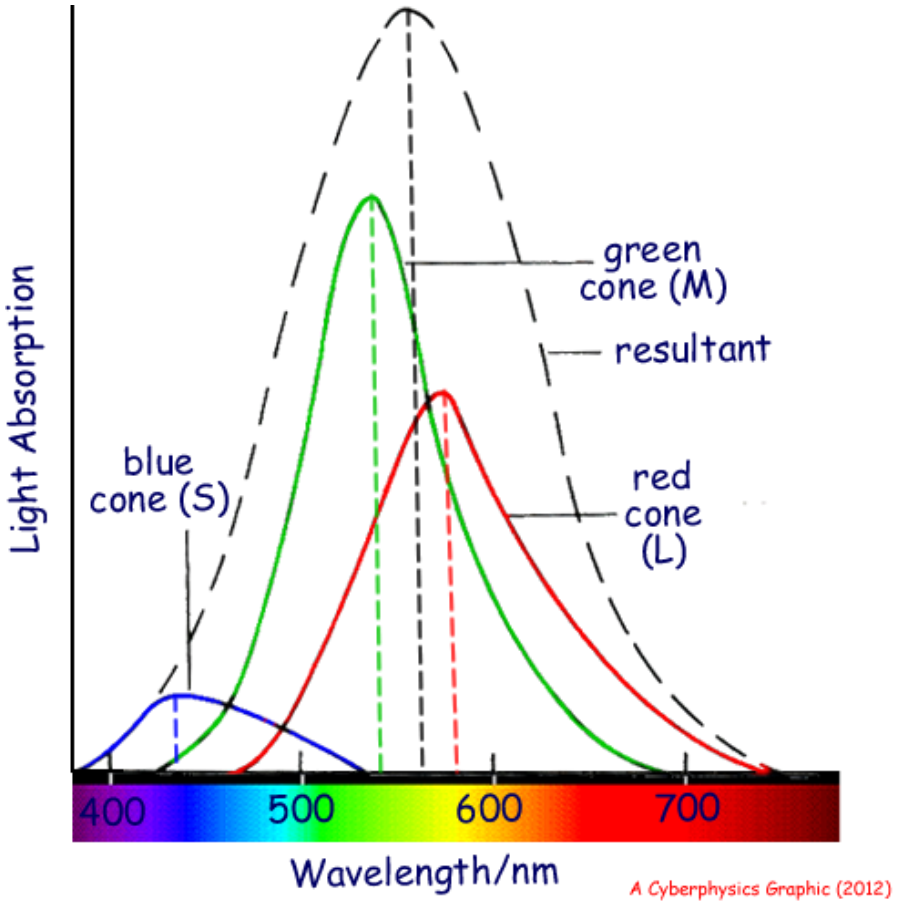




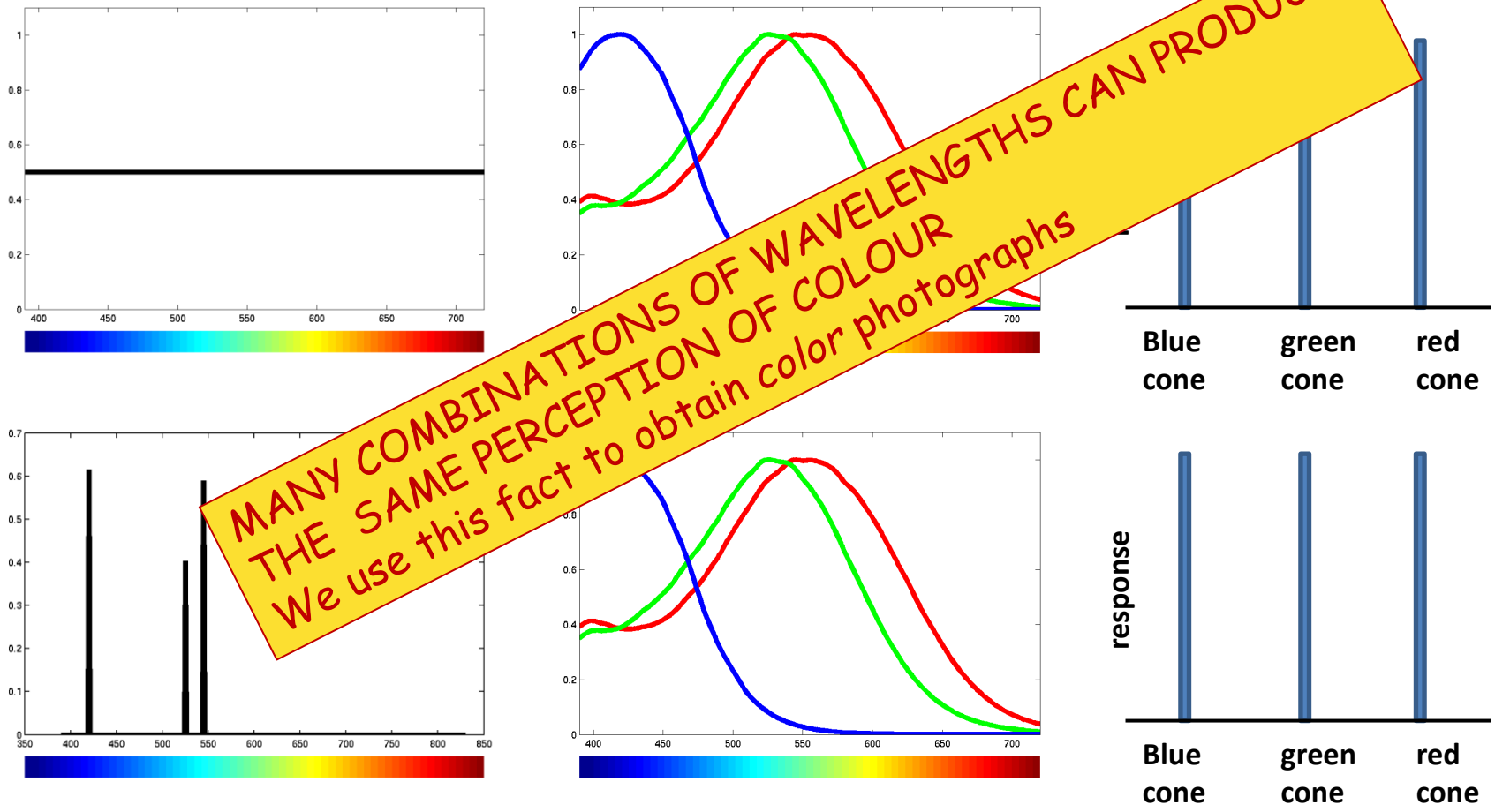




Why the different weights?

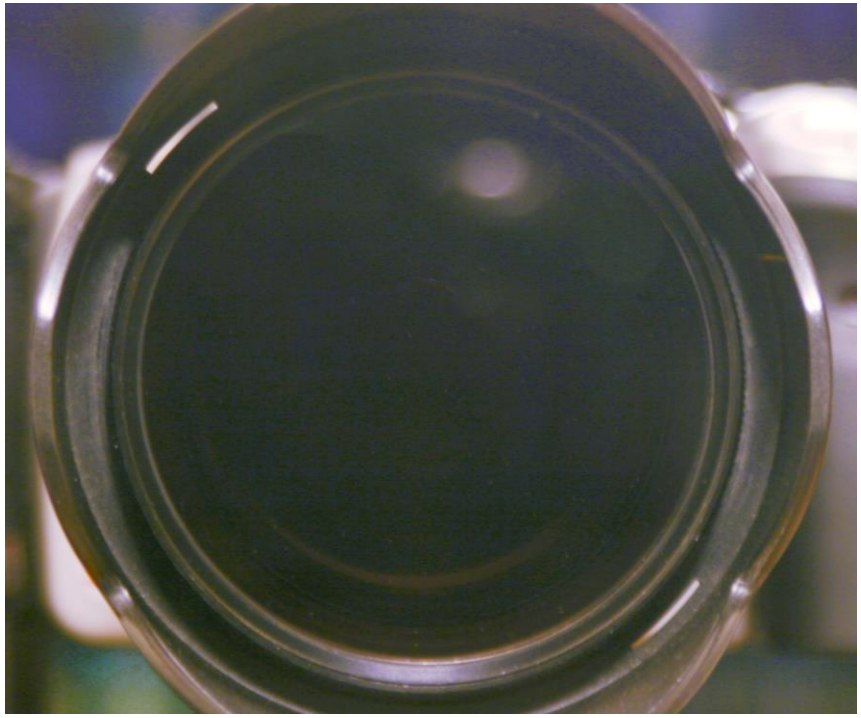


Response to Sparse Light

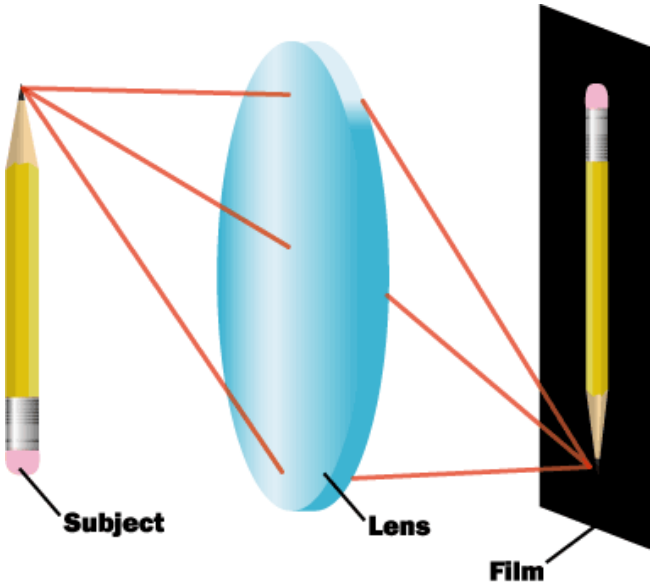


- "Absolute" color is a myth. "Color" is what we sense. There is no one-to-one correspondence between sensed color and wavelength.

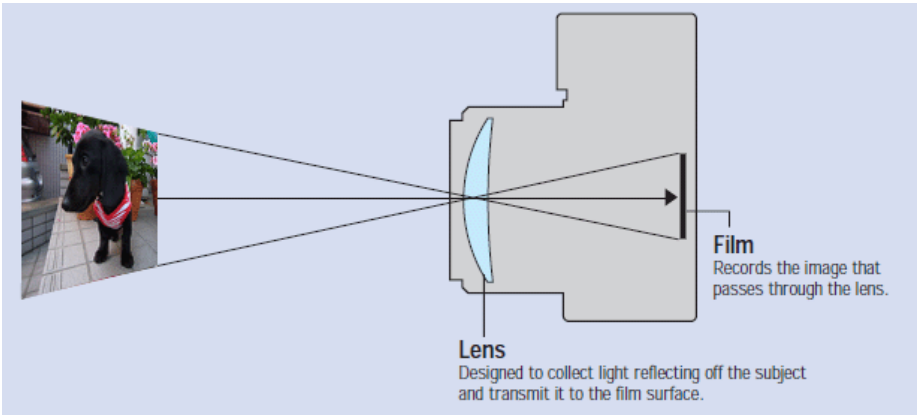
The camera mimics the eye



Photography: general concepts

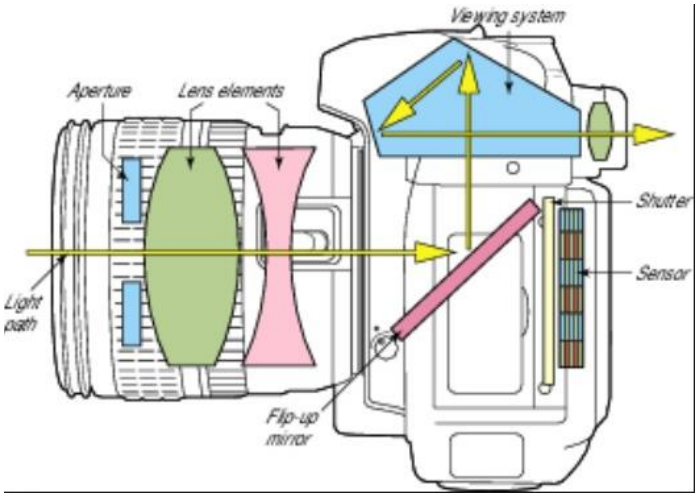


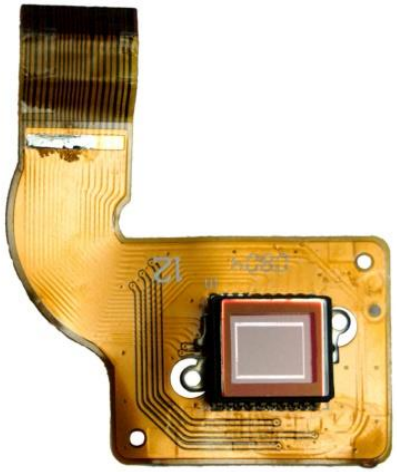
A film camera

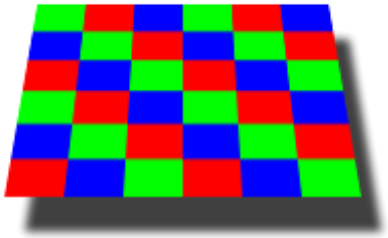


- **Basic principle:** light from the scene is focused onto an image plane by a convex lens
- The image is “captured” by chemicals on a film in the image plane

Digital camera

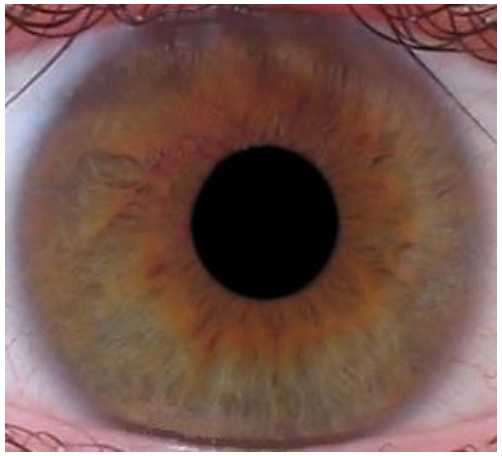


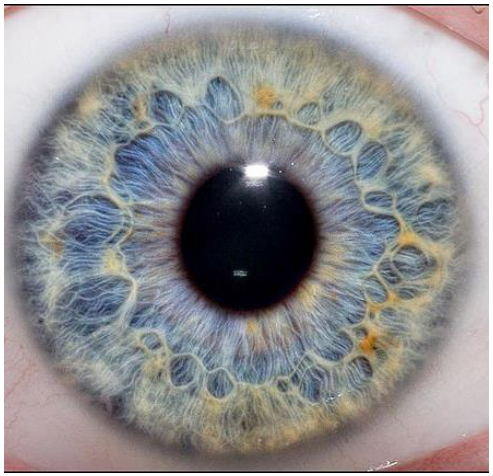


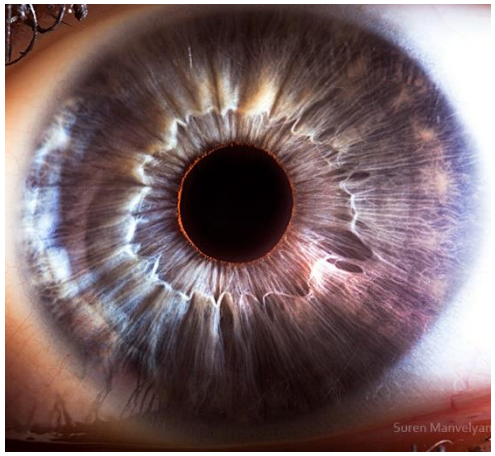


- Lens projects image on a ***digital sensor***
  - Typically **CCD** (charge-coupled device) or **CMOS** (complementary metal-oxide semiconductor)
- Sensor comprises sensing elements of 3 colors
  - Different strategies for arrangement of color sensors
- Limited number of sensing elements
  - 200-1200 ppi (pixels per inch)
  - Includes an anti-aliasing filter to eliminate aliasing in the image

Resolution of a digital camera





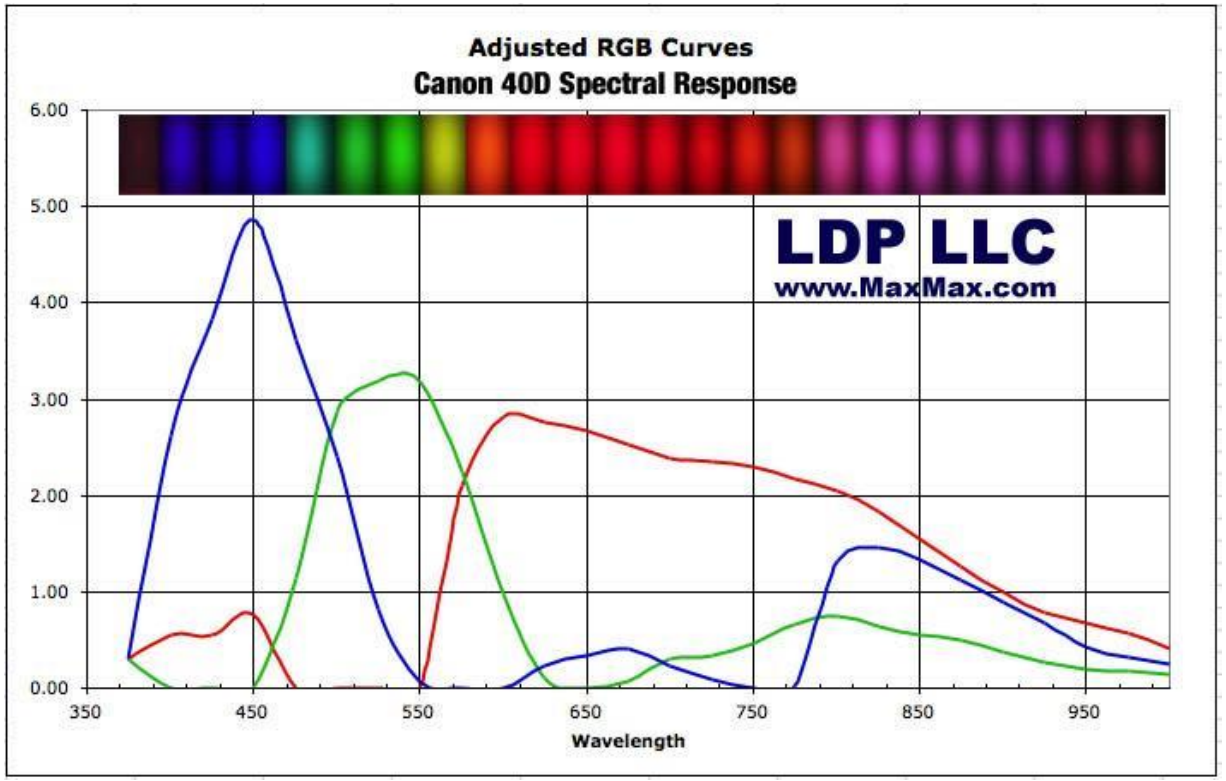


The human iris at different resolutions

Higher resolution = more ppi (pixels/inch)

If images of different resolution are patched, shows up in intensity histograms

The sensitivity of digital cameras



- Each “color” sensor responds to a wide range of wavelengths

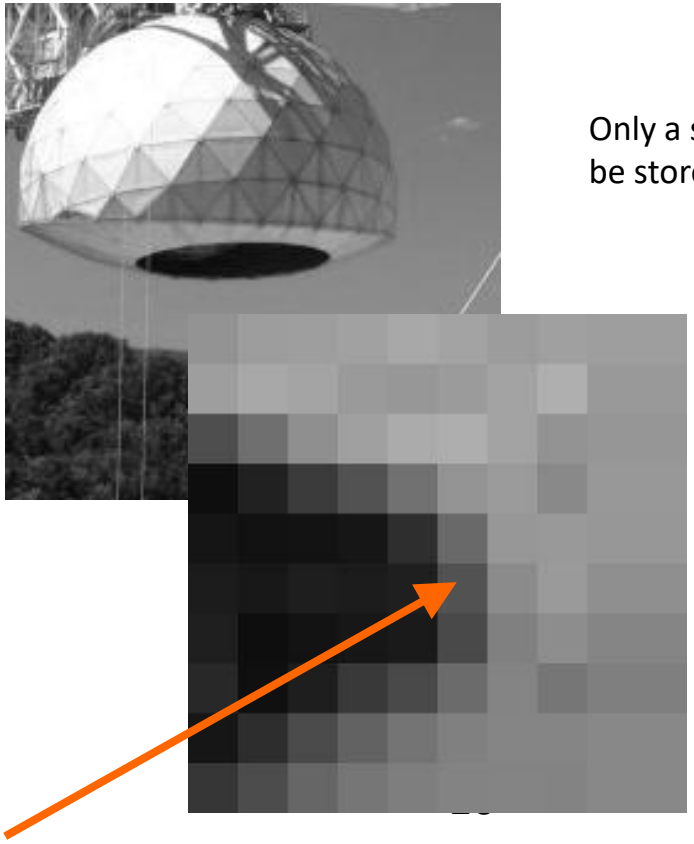
What is in a digital image?

- A digital image is merely a set of matrices of numbers

- Greyscale: a single matrix of numbers
  - Each number represents the **intensity** of the image at a specific location in the image
    - Interpreted as the shade of grey at that location
  - Greyscale images are typically derived from color images

Digital computer Images: Grey Scale

What we see



Only a single number need be stored per pixel

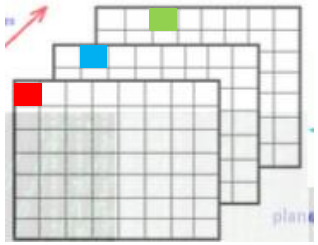
What the computer "sees"

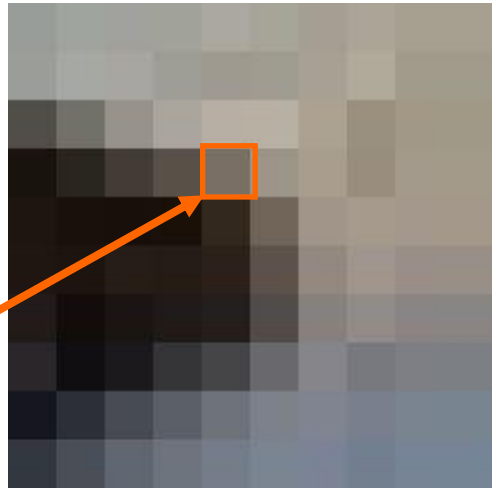
147	157	158	161	168	164	156	159	158	158
157	166	163	158	154	155	162	170	154	154
78	111	143	162	171	173	163	149	150	150
13	36	60	82	116	145	155	136	151	151
21	18	19	22	46	106	151	152	151	151
27	25	29	28	34	87	141	151	143	143
31	11	19	22	26	72	128	141	133	133
40	15	29	53	72	108	131	124	128	128
22	46	75	99	116	128	133	133	136	136
55	79	103	122	125	130	132	132	136	136

Picture Element (PIXEL)

Position & gray value (scalar)

Color Images





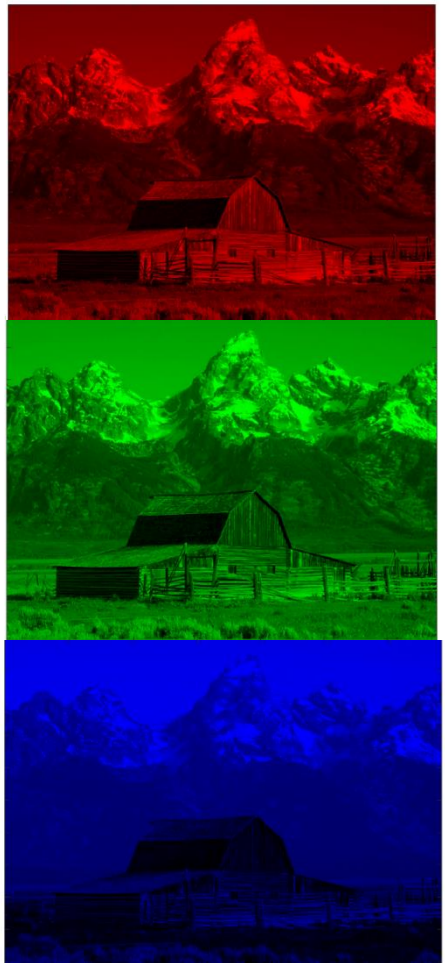
Each pixel represented by a triad of numbers

Picture Element (PIXEL)

Position & color value (red, green, blue)

RGB colorspace representation of an image

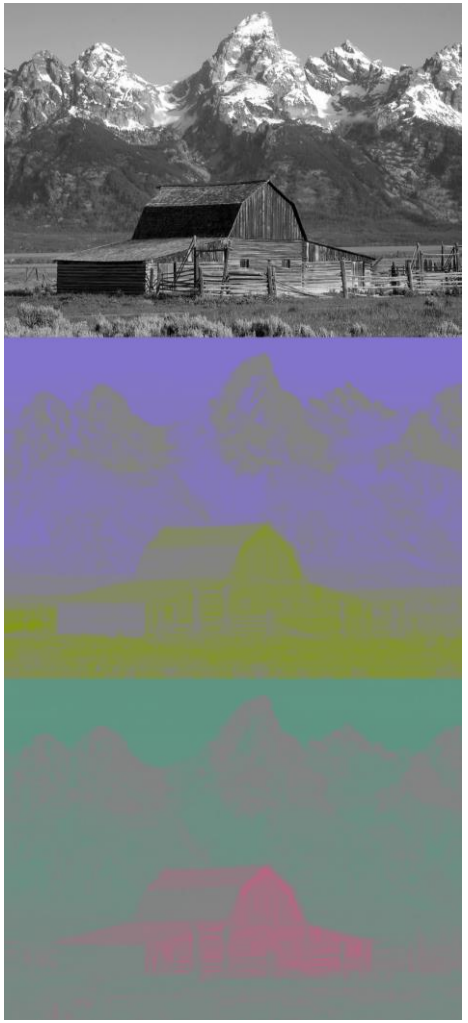
Example from wikipedia



RGB colorspace:

The three matrices represent R, G and B

Y Cb Cr colorspace

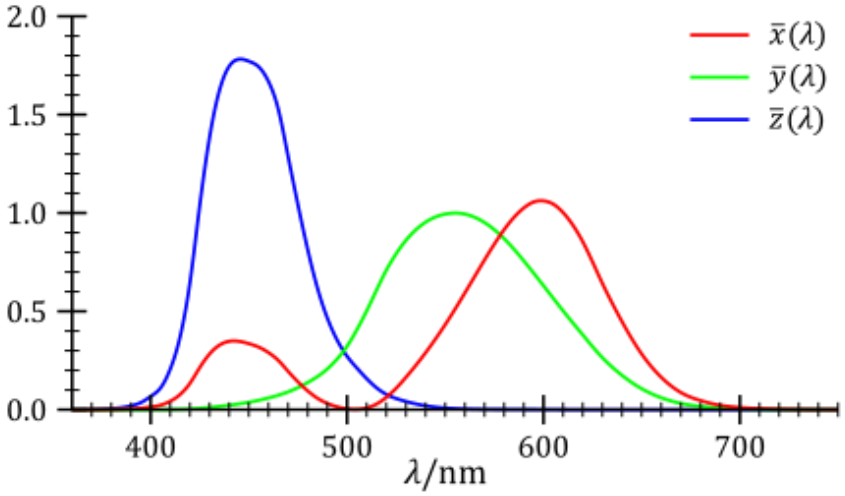


- **Y** carries overall intensity information
- **Cb** carries blue-green information
  - **a.Red – b.Y**
- **Cr** carries red-green information
  - **c.Blue – d.Y**

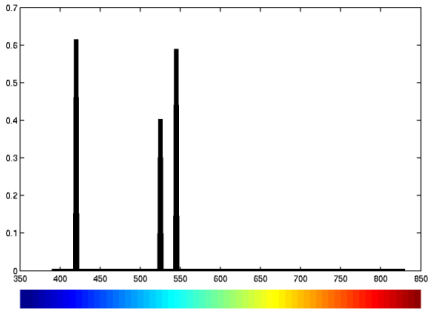
Interesting consequences

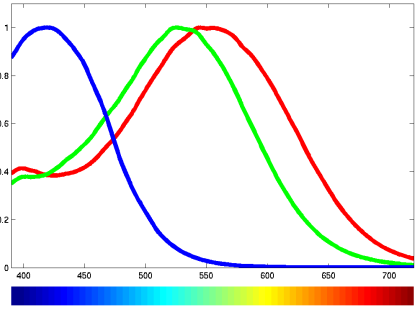
- Images are shown using only 3 colors
  - Red, Green, Blue
  - OR their complements, Cyan, Magenta, Yellow (for prints)
- **This results in limitations**
  - **Image depends on what can be achieved through RGB only!!**

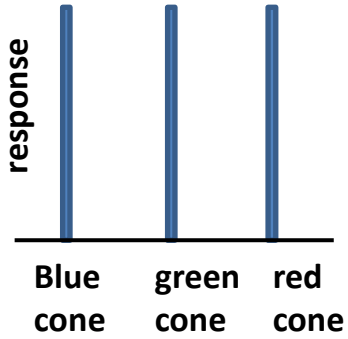
Limitations of RGB displays



- The same intensity of monochromatic light will result in different *perceived intensity* (brightness at) different wavelengths

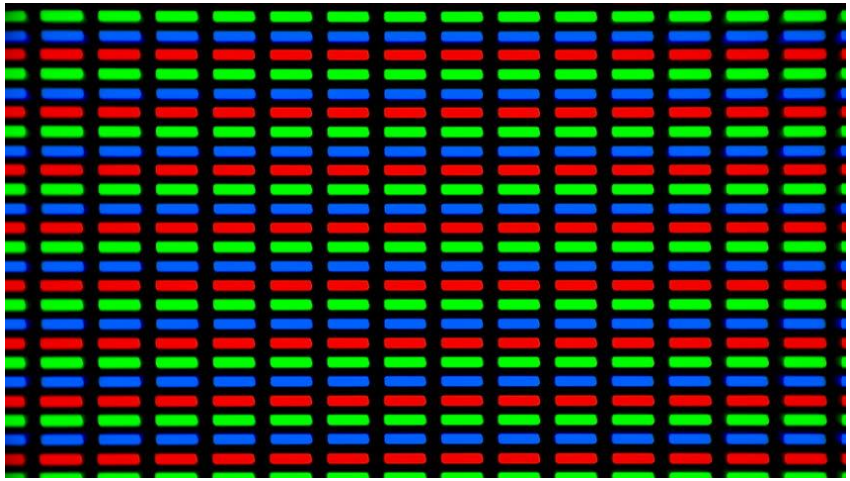






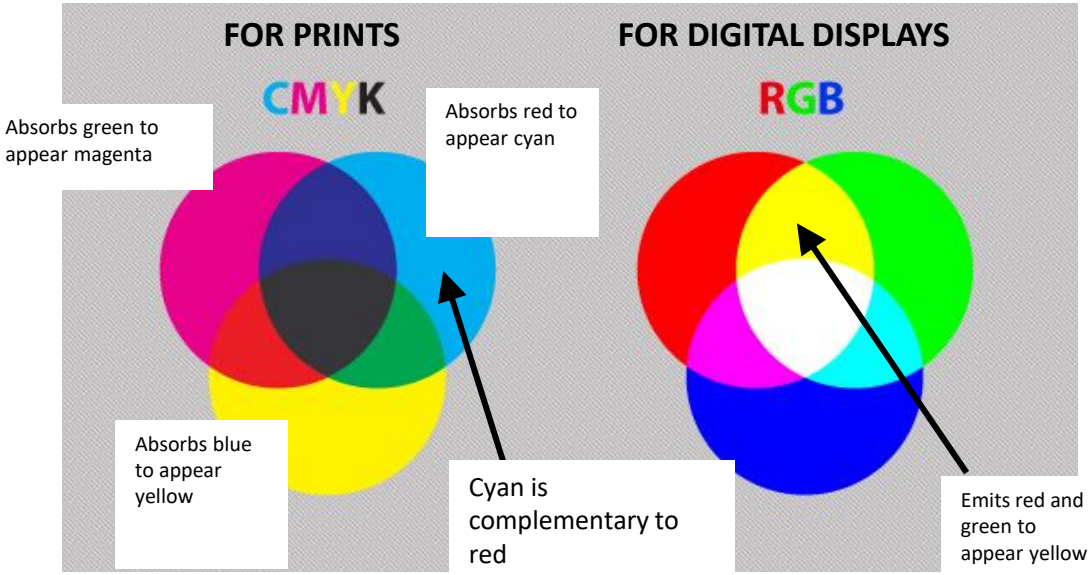
- Many combinations of wavelengths can produce the *same* sensation of color.
  - **Yet humans can distinguish ~1 *million* colors**

These drive display technology



- **DISPLAY:** uses RGB, colors are directly emitted, e.g. red is obtained by emitting red, other colors are obtained by emitting the appropriate primary colors, e.g. cyan is obtained by emitting blue and green
- **PRINT:** uses CMYK, **colors are printed directly on paper**. Colors are created through properties of ink to ABSORB primary colors
- Color cameras use RGB
- Color **photographs use CMYK**

These drive display technology



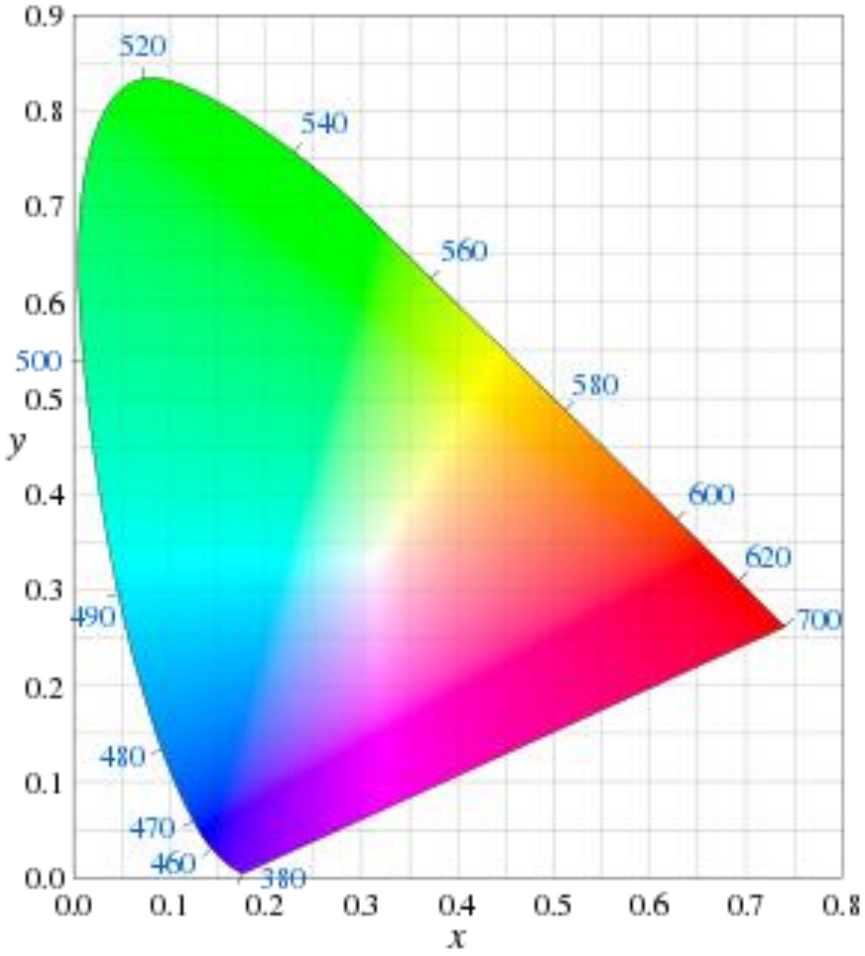
- **PRINT:** uses CMYK, **colors are printed directly on paper**. Property of ink: magenta ink appears magenta because it ABSORBS green, which is its complement
  - Black is directly inked, other wise would use 3 times as much ink (could be obtained by combining CMY). White is obtained by applying no ink. Most economical. More than RGB.

Displaying Digital Images

- To represent images, digital display technology utilizes trichromatic nature of human vision
  - Sufficient to trigger the three cone types to produce the sensation of the desired color
- "Chosen" colors are red (650nm), green (510nm) and blue (475nm)
  - By appropriate combinations of these colors, the cones can be excited to produce a very large set of colors
  - Which is still a small fraction of what we can actually see
- **How many colors are produced in actual displays?** …

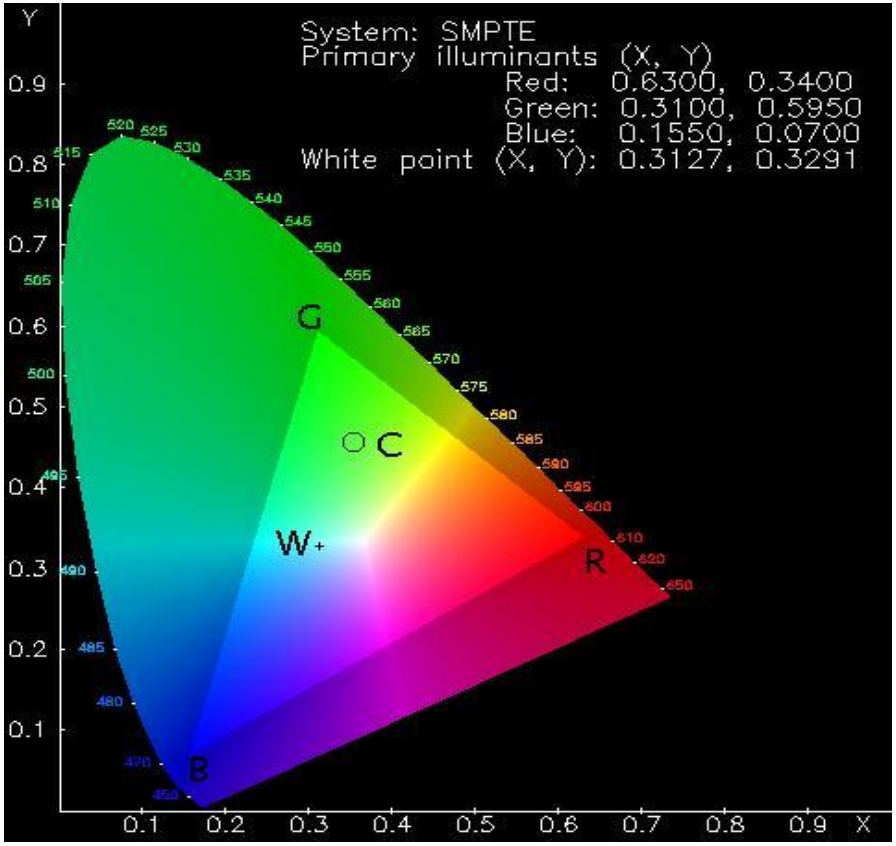
Understanding display limitations: The "CIE" color space

- From 1920 experiments by David Wright and John Guild, (updated 1976)
  - Partition between subject's eyes
  - 2 knobs, x for intensity of red ; y for intensity of green ; z = blue
  - x+y+z = 1, (i.e. net intensity is normalized)
  - Subjects adjusted x, y on the right of partition to match a color on the left
  - Outer curve represents monochromatic light
  - Subjects could only see the x-y area shown
    - Defines the range of human color vision

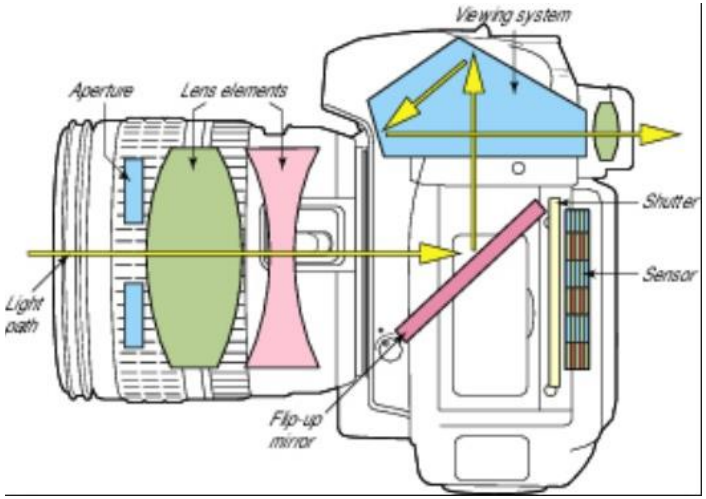


What is displayed (or captured)

- The RGB triangle
  - All combinations of R G B in a display or camera sensor fall within the triangle shown
  - Each corner represents the (X,Y,Z) coordinate of one of the three “primary” colors used in images
  - Colors outside this area cannot be matched by additively combining the 3 colors used
  - Any other set of monochromatic colors would have a differently restricted area
  - TV images can never be like the real world
  - In reality, this represents a very *tiny fraction* of our visual acuity
  - Also affected by the quantization of levels of the colors



What happens after capture?



Additional processing

Store

- The digital image is not stored in its raw form
- Image is processed in multiple stages before it is stored

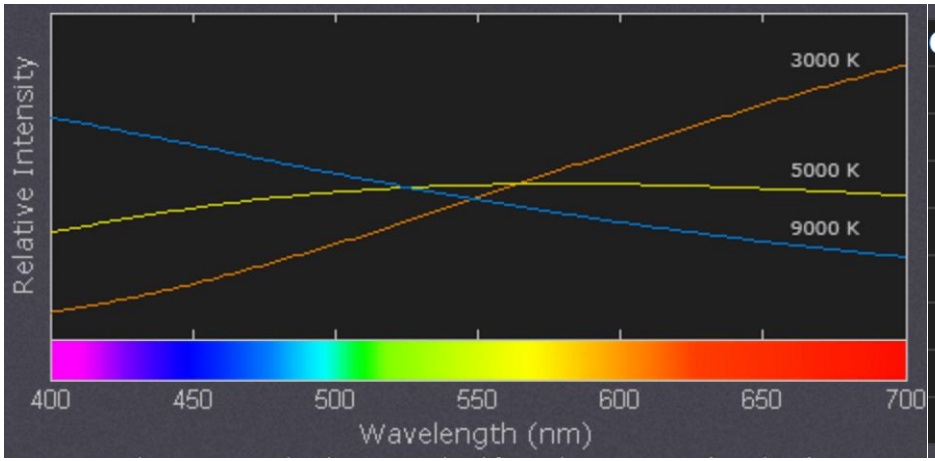
Additional Processing Steps

- **White balancing**
- **Demosaicing**
  - Interpolation for colors
  - Also called demosaicking or debayering
- **Gamma correction**
- And other processing steps

White balancing

- Ever wonder why the picture you saw on the screen when taking a photo is not like the photograph you finally got?
- Its probably because of **automatic white balancing**

The effect of light source



Color Temperature	Light Source
1000-2000 K	Candlelight
2500-3500 K	Tungsten Bulb (household variety)
3000-4000 K	Sunrise/Sunset (clear sky)
4000-5000 K	Fluorescent Lamps
5000-5500 K	Electronic Flash
5000-6500 K	Daylight with Clear Sky (sun overhead)
6500-8000 K	Moderately Overcast Sky
9000-10000 K	Shade or Heavily Overcast Sky

- Different light sources have different light spectra
  - Quantified through “temperature”: the temperature of an ideal black body that emits the same spectrum
  - E.g. a black body at 3000 K has a high intensity of red (left), corresponding to sunrise/sunset color intensities

The effect of light source

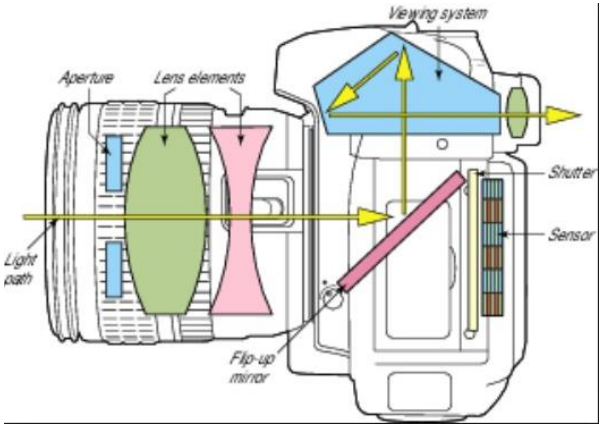
- Pictures of the same scene taken under different light conditions (color temperatures) look different
- Our eyes normalize out these differences
- The camera does not
  - You see this “raw” image on the screen

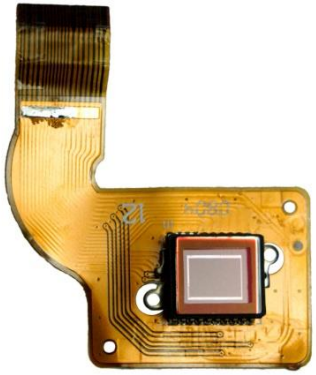
White balancing

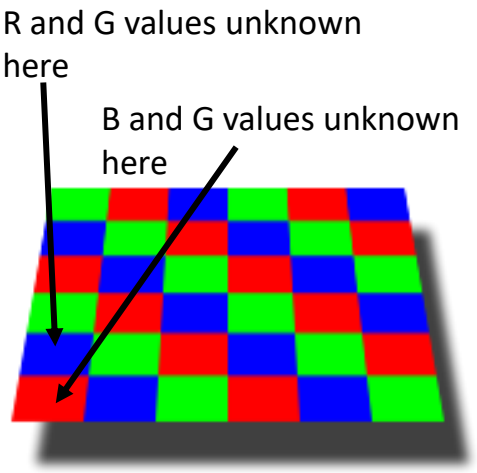
Camera automatically *adjusts* individual colors to give you a more “neutral” image

- Closer to an image taken under white light
  - Called “white balancing”

Demosaicing

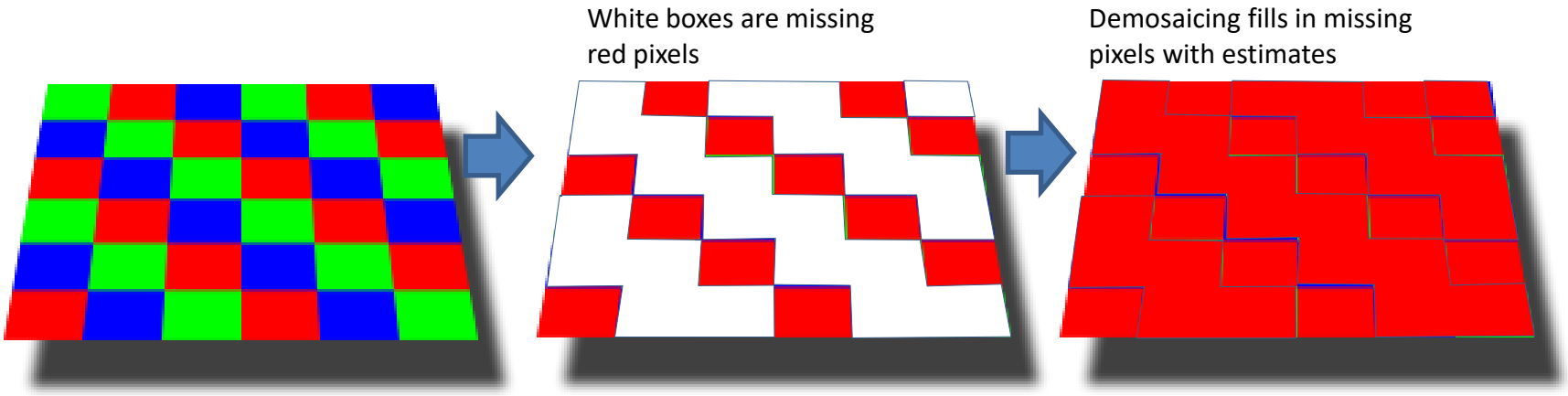






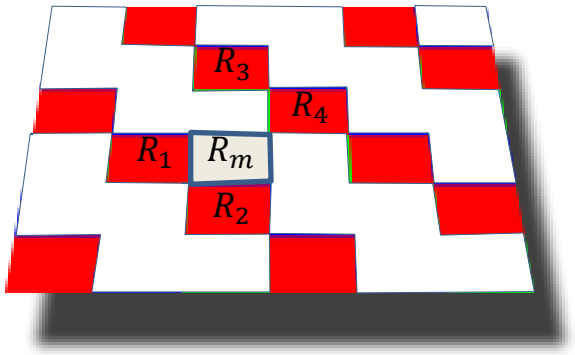
- An NxM digital camera does not actually capture NxM pixels for each of the three colors
- R, G and B sensors alternate
- In any pixel, the value of only one of the colors is known, the other two are unknown

Demosaicing



- For each color, the *missing* pixels must be filled in
  - Done by *estimating* them from their neighbors
  - This process is called *demosaicing*

Typical Demosaicing algo



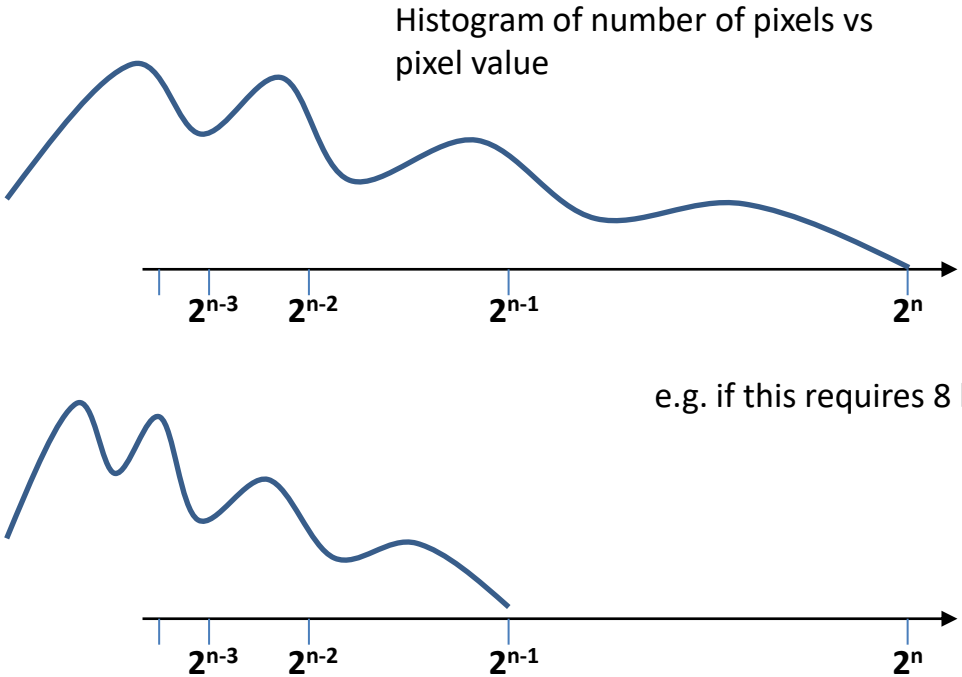
$$R_m \approx w_1 R_1 + w_2 R_2 + w_3 R_3 + w_4 R_4$$

- Simple demosaicing algorithm: estimate each missing pixel as a linear combination of the nearest present pixels
- More advanced algorithms use splines and other functions
- The actual algorithm used is generally proprietary

Gamma correction

- Pixel values (of all colors) are compressed through a non-linearity
  - Typically $P_{compressed}(i,j) = A\big(P(i,j)\big)^{\gamma}, \quad \gamma < 1$
- This reduces the dynamic range of pixel values
  - Can be stored using fewer bits
- Image must be uncompressed for final display
  - $P(i,j) = A^{-1}\big(P_{compressed}\big)^{\frac{1}{\gamma}}$

Gamma correction



- Compression reduces dynamic range of pixel values
- Requires fewer bits to store

Quantization and Saturation

- Captured images are typically quantized to N-bits
- Standard value: 8 bits per color (24 bits per pixel)
  - 2^8 (or 256) distinct intensity values per color
  - Very coarse
  - Humans can easily distinguish 100,000 intensity values
  - Would require at least 17 bits per color
- And most cameras will give you 6 real bits anyway…
  - Only 64 distinct intensity values per color
  - The final two bits are rarely used after gamma correction

And finally.. compression

- The final raw picture is typically too large
  - 15 megabytes for a 5 mega-pixel photo
    - 3 bytes per pixel (one per color) for 8-bit resolution

- It is *compressed* prior to storage
- Usual compression used: **JPEG** compression
  - "JPEG" is an acronym for the **J**oint **P**hotographic **E**xperts **G**roup, which created the standard.

Image compression

JPEG compression

- Uses properties of images to compress them
  - Understanding this is important for forensic reasons

- Relies on two key concepts:
  - The Discrete Cosine Transform
  - Quantization tables

Representing images


aboard **Apollo space** capsule.
1038 x 1280 - 142k
LIFE


Apollo Xi
1280 x 1255 - 226k
LIFE


aboard **Apollo space** capsule.
1029 x 1280 - 128k
LIFE


Building **Apollo space** ship.
1280 x 1257 - 114k
LIFE


aboard **Apollo space** capsule.
1017 x 1280 - 130k
LIFE


Apollo Xi
1228 x 1280 - 181k
LIFE


Apollo 10 space ship, w.
1280 x 853 - 72k
LIFE


Splashdown of **Apollo** XI mission.
1280 x 866 - 184k
LIFE


Earth seen from **space** during the
1280 x 839 - 60k
LIFE


Apollo Xi
844 x 1280 - 123k
LIFE


Apollo 8
1278 x 1280 - 74k
LIFE


working on **Apollo space** project.
1280 x 956 - 117k
LIFE


the moon as seen from **Apollo** 8
1223 x 1280 - 214k
LIFE


Apollo 11
1280 x 1277 - 142k
LIFE


Apollo 8 Crew
968 x 1280 - 125k
LIFE

- How do we store an image?
  - Easy answer – simply specify every pixel value
    - Problem – for a 5 MP camera, all 5 million pixel values (per image) must be specified
  - Is there a more efficient way?
    - Can we specify based on some (smaller number of) standard **"image elements"**?

Representing images


aboard **Apollo space** capsule.
1038 x 1280 - 142k
LIFE


Apollo Xi
1280 x 1255 - 226k
LIFE


aboard **Apollo space** capsule.
1029 x 1280 - 128k
LIFE


Building **Apollo space** ship.
1280 x 1257 - 114k
LIFE


aboard **Apollo space** capsule.
1017 x 1280 - 130k
LIFE


Apollo Xi
1228 x 1280 - 181k
LIFE


Apollo 10 space ship, w.
1280 x 853 - 72k
LIFE


Splashdown of **Apollo** XI mission.
1280 x 866 - 184k
LIFE


Earth seen from **space** during the
1280 x 839 - 60k
LIFE


Apollo Xi
844 x 1280 - 123k
LIFE


Apollo 8
1278 x 1280 - 74k
LIFE


working on **Apollo space** project.
1280 x 956 - 117k
LIFE

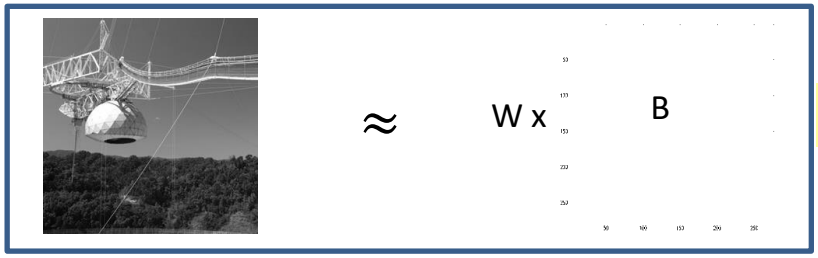

the moon as seen from **Apollo** 8
1223 x 1280 - 214k
LIFE


Apollo 11
1280 x 1277 - 142k
LIFE


Apollo 8 Crew
968 x 1280 - 125k
LIFE

- *We could use content!!*
- Observation: The most common element in the image: background
  - Or rather large regions of relatively featureless shading
  - Uniform sequences of numbers

Representing images using an “elementary” image

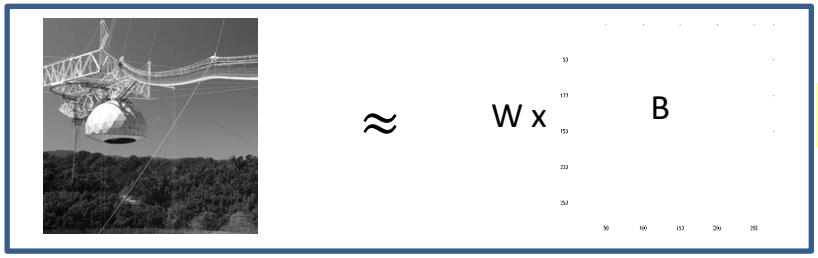


$$\text{Image} = \begin{bmatrix} pixel\,1 \\ pixel\,2 \\ . \\ pixel\,N \end{bmatrix} \qquad B = \frac{1}{\sqrt{N}} \begin{bmatrix} 1 \\ 1 \\ . \\ 1 \end{bmatrix}$$

- Most of the figure is a more-or-less uniform shade
  - Dumb approximation – a image is a block of uniform **shade**
    - Will be mostly right!
- Easy to compute the scalar weight W if length of B is 1
  - Only need to store W to store the image

$$W = B^T Image$$
$$Image = BW$$

Representing images using an "elementary" image

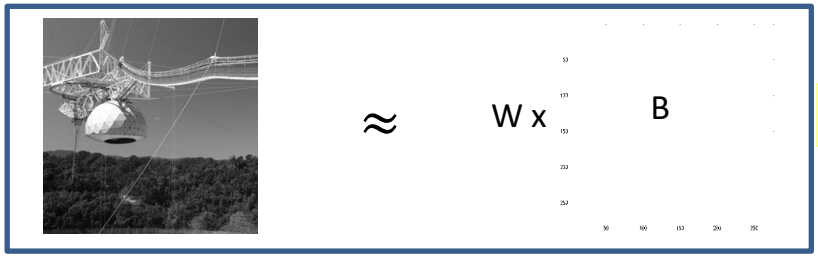


$$\text{Image} = \begin{bmatrix} pixel\,1 \\ pixel\,2 \\ . \\ pixel\,N \end{bmatrix} \qquad B = \frac{1}{\sqrt{N}} \begin{bmatrix} 1 \\ 1 \\ . \\ 1 \end{bmatrix}$$

- Most of the figure is a more-or-less uniform shade
  - Dumb approximation – a image is a block of uniform **shade**
    - Will be mostly right!
- Easy to compute the scalar weight W if length of B is 1
  - Only need to store W to store the image

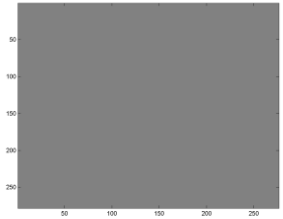
$$W = B^T Image$$
$$Image = BW$$

Representing images using a “plain” image



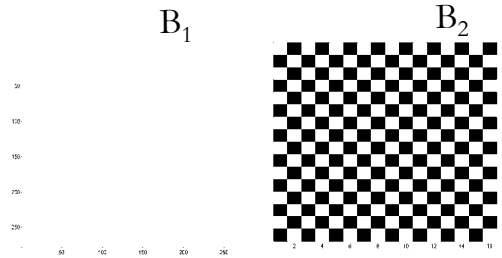
$$\text{Image} = \begin{bmatrix} pixel\,1 \\ pixel\,2 \\ . \\ pixel\,N \end{bmatrix} \qquad B = \frac{1}{\sqrt{N}} \begin{bmatrix} 1 \\ 1 \\ . \\ 1 \end{bmatrix}$$

$$W = B^T Image$$
$$Image = BW$$



Problem: Not a terribly good approximation
Must improve

Adding more *bases*



$$B = \frac{1}{\sqrt{N}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \\ 1 & 1 \\ 1 & -1 \\ 1 & 1 \end{bmatrix}$$

- Lets improve the approximation
- Images have some fast varying regions
  - Dramatic changes
  - Add a second picture that has very fast changes
    - **A checkerboard where every other pixel is black and the rest are white**

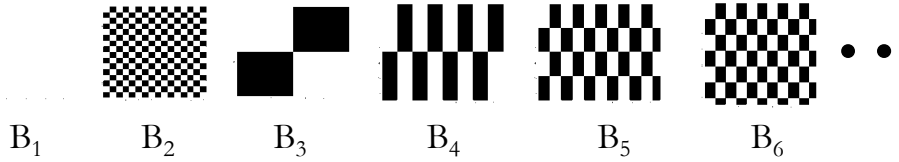
$$Image \approx w_1 B_1 + w_2 B_2$$

$$W = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \qquad B = [B_1 \;\; B_2]$$

$$W = B^T Image$$
$$Image = BW$$

Problem: Still not a terribly good approximation
Must improve

Adding still more bases

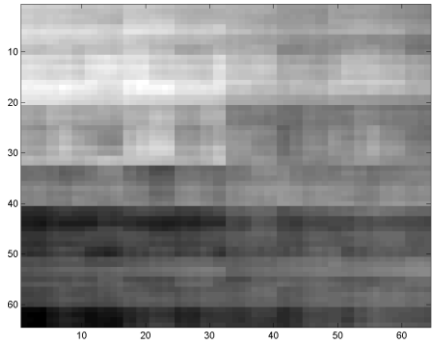


- Regions that change with different "speeds"

$$Image \approx w_1 B_1 + w_2 B_2 + w_3 B_3 + \ldots$$

$$W = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ . \\ . \end{bmatrix} \qquad B = [B_1 \;\; B_2 \;\; B_3]$$

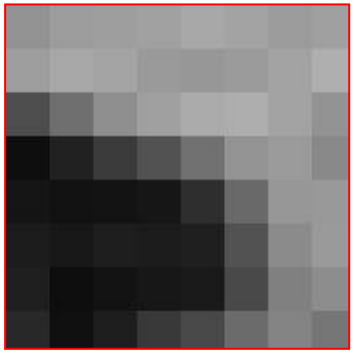
$$W = B^T Image$$
$$Image = BW$$

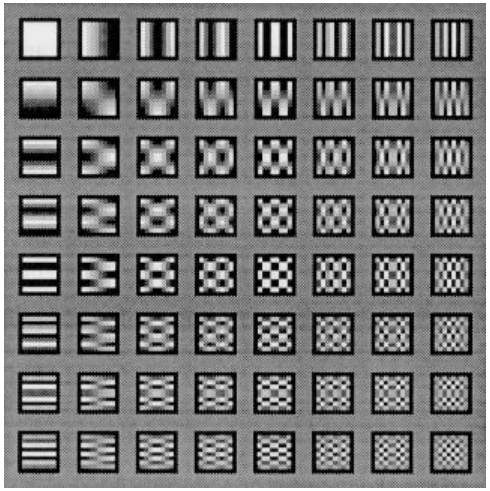


Getting closer at 625 bases!

Gets better
Eventually perfect when number of bases = no. of pixels

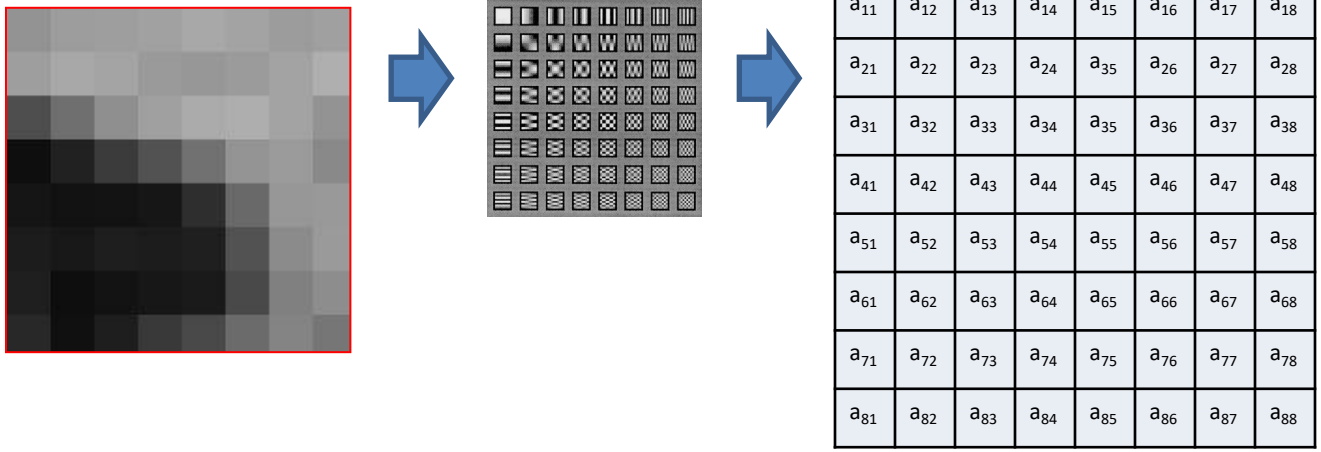
The Discrete Cosine Transform





- To fully describe an MxN image we need MxN bases
  - Which can be arranged as an MxN grid
  - The Discrete Cosine transform uses “two-dimensional cosines” as the bases, rather than checker-board patterns

The Discrete Cosine Transform



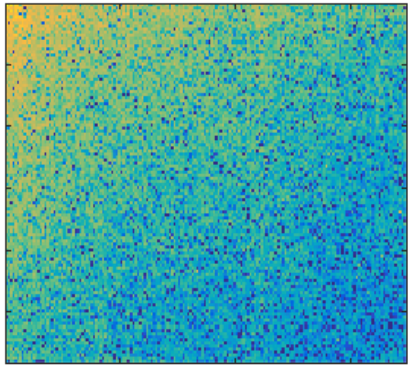
a_{11}	a_{12}	a_{13}	a_{14}	a_{15}	a_{16}	a_{17}	a_{18}
a_{21}	a_{22}	a_{23}	a_{24}	a_{35}	a_{26}	a_{27}	a_{28}
a_{31}	a_{32}	a_{33}	a_{34}	a_{35}	a_{36}	a_{37}	a_{38}
a_{41}	a_{42}	a_{43}	a_{44}	a_{45}	a_{46}	a_{47}	a_{48}
a_{51}	a_{52}	a_{53}	a_{54}	a_{55}	a_{56}	a_{57}	a_{58}
a_{61}	a_{62}	a_{63}	a_{64}	a_{65}	a_{66}	a_{67}	a_{68}
a_{71}	a_{72}	a_{73}	a_{74}	a_{75}	a_{76}	a_{77}	a_{78}
a_{81}	a_{82}	a_{83}	a_{84}	a_{85}	a_{86}	a_{87}	a_{88}

- An 8 x 8 image requires 8 x 8 (64) DCT bases
- The image itself is represented by the 8x8 weights with which the bases must be combined to compose it
- No gain so far! Its an identity transform….

Storing the DCT

Original image

DCT coefficients

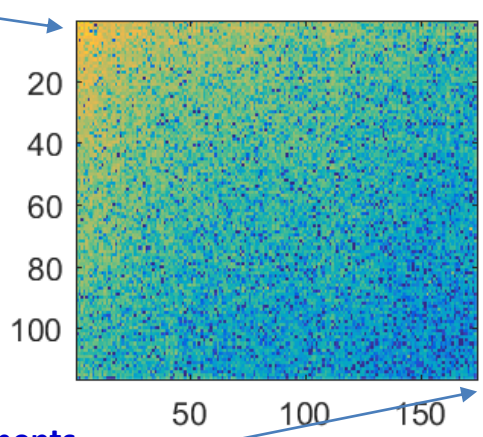


Recovered from DCT

- Instead of storing the *image,* store the DCT coefficients
  - Image can be recovered from DCT through inverse DCT

Property of the DCT

High-information components, high resolution required



Low-information components, low resolution sufficient

- Most of the useful information is in the upper left corner of the DCT
  - These numbers must be represented with high accuracy
- The bottom right corner is less informative
  - Can make errors in storing these numbers without affecting image too much

Quantization

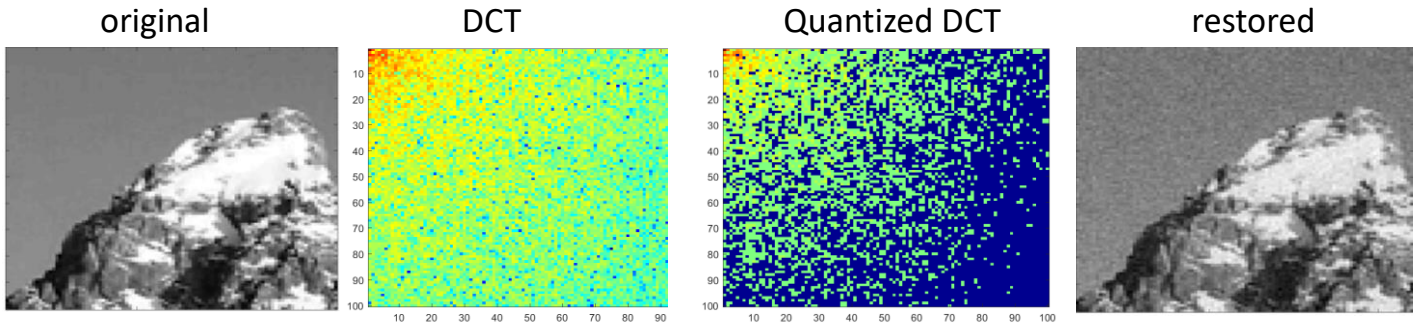
- **Quantization:** The process of reducing the number of bits used to represent a number
- E.g.
  - An integer X in the range 0-255 requires 8 bits to store
  - Consider Y = X/2 [integer division]
  - Y is an integer in the range 0-127
    - Requires only 7 bits to store
  - Y is a *quantized* version of X requiring lesser storage than X

Quantization: 2

- Quantizing X to Y:
  - $Y = X/K$
  - Y requires $\log(\text{Range}(X)) - \log(K)$ bits of storage
  - Only store Y rather than X
    - X can be partially recovered by Y so long as we know K
    - K = quantization factor

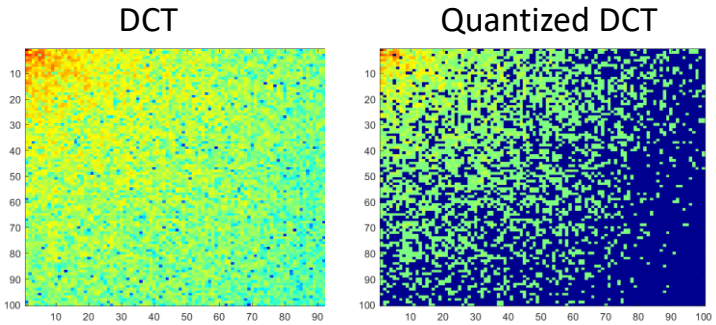
- Recovering X from Y
  - $X \approx KY$
  - This is an *approximation* because the remainder of X/K is lost
    - Real value of X: $X = KY + R$,
    - But since R is not stored we cannot recover X
  - **The larger the quantization factor K, the greater the *loss* due to approximation**
    - **A quantization factor of $K = 1$ represents zero loss**

Jpeg and Quantization



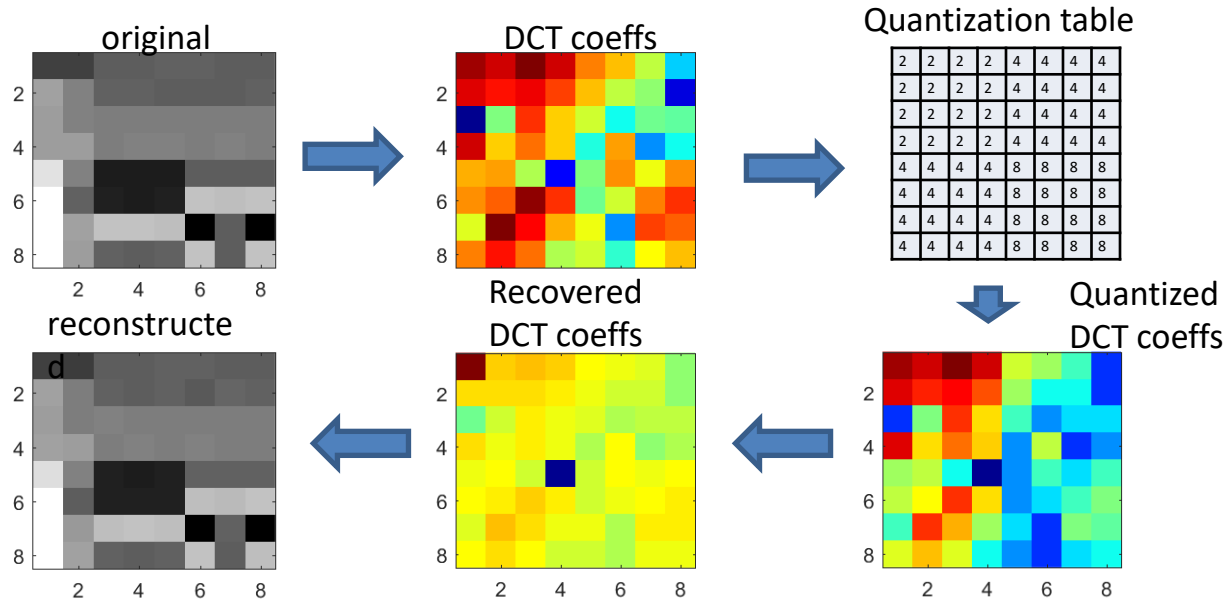
- *Quantize* the DCT coefficients prior to storage
  - Each quantized coefficient will require fewer bits, reducing overall storage requirement for the image
- Illustrated by figure: "Quantized DCT" : K=32 (or 2^5), requires only 3 bits per pixel color, vs. 8 for original image
  - This is *NOT* high-quality quantization; we can do better

Jpeg and Quantization



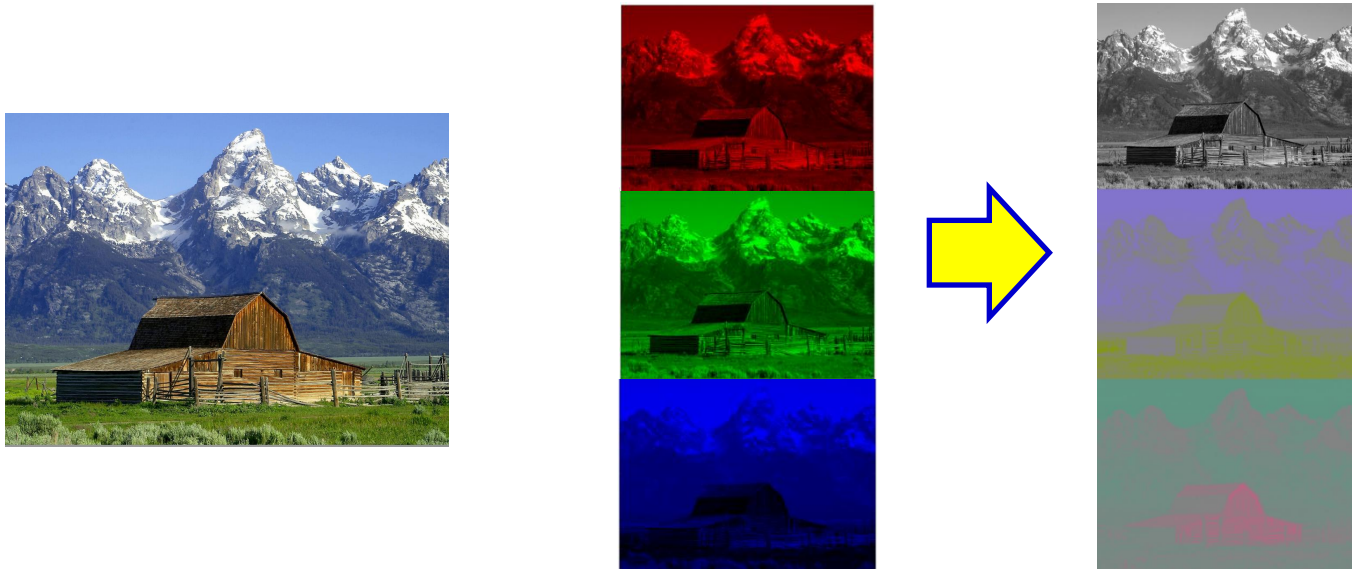
- In the example *ALL* coefficients have been quantized by the same quantization factor (32)
- In practice we can use a different quantization factor for each coefficient
  - Resulting in both higher quality recovery and better compression

Quantization Tables



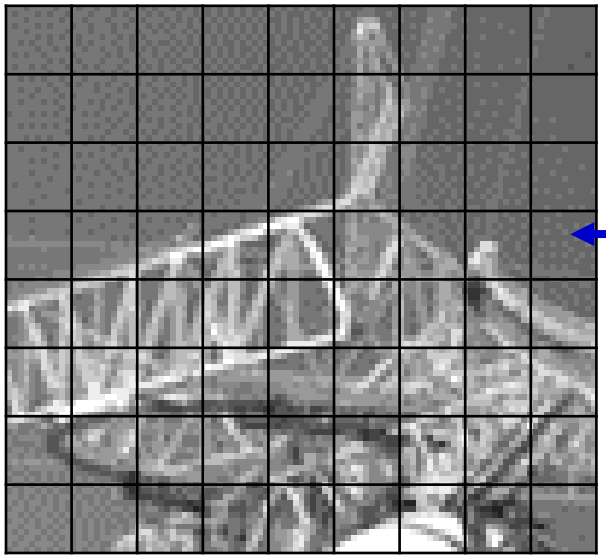
- Example shows an 8x8 DCT coefficient matrix, an 8x8 quantization table matrix and the quantized DCT coefficients
  - Each DCT coefficient is quantized using the quantization factor in the corresponding location of the quantization table

Jpeg Compression 1: Conversion



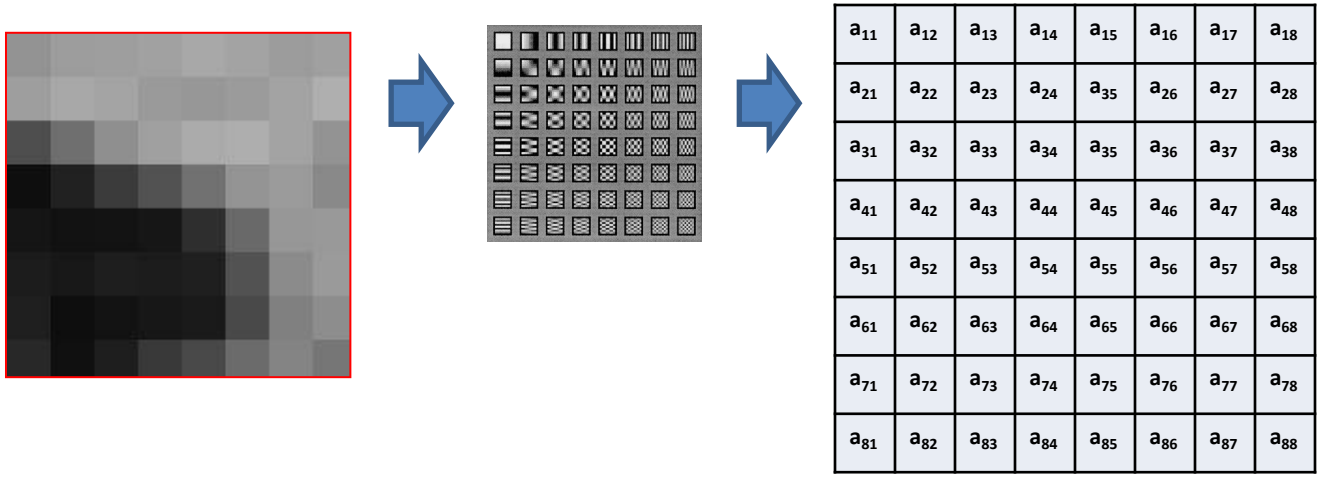
- As a first step, images are converted to Y Cb Cr representation
  - Enables high levels of lossy compression of Cb and Cr channels without significant degradation of the picture

JPEG compression: Blocking



- Segment the image into non-overlapping 8x8 boxes
  - Pad the borders with blank pixels, if the size of the image is not a multiple of 8 in any dimension

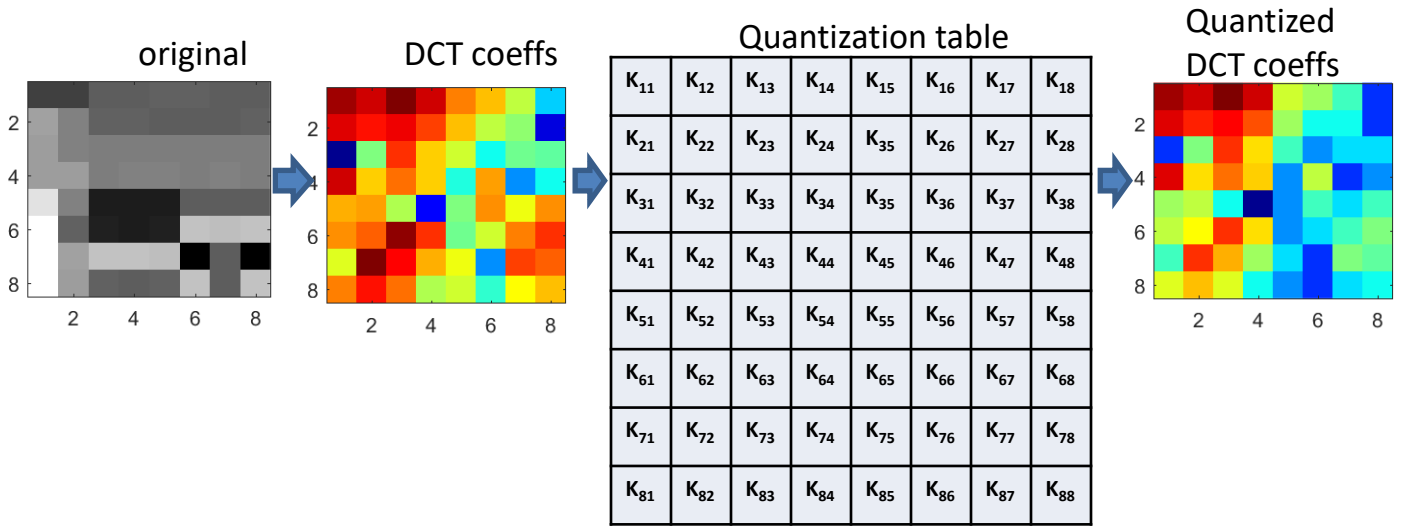
JPEG compression: 2



a_{11}	a_{12}	a_{13}	a_{14}	a_{15}	a_{16}	a_{17}	a_{18}
a_{21}	a_{22}	a_{23}	a_{24}	a_{35}	a_{26}	a_{27}	a_{28}
a_{31}	a_{32}	a_{33}	a_{34}	a_{35}	a_{36}	a_{37}	a_{38}
a_{41}	a_{42}	a_{43}	a_{44}	a_{45}	a_{46}	a_{47}	a_{48}
a_{51}	a_{52}	a_{53}	a_{54}	a_{55}	a_{56}	a_{57}	a_{58}
a_{61}	a_{62}	a_{63}	a_{64}	a_{65}	a_{66}	a_{67}	a_{68}
a_{71}	a_{72}	a_{73}	a_{74}	a_{75}	a_{76}	a_{77}	a_{78}
a_{81}	a_{82}	a_{83}	a_{84}	a_{85}	a_{86}	a_{87}	a_{88}

- Compute the DCT of each 8x8 box in the image
  - To obtain an 8x8 DCT coefficients
  - Each 8x8 box is thus converted to 64 DCT coefficients

Jpeg Compression



K_{11}	K_{12}	K_{13}	K_{14}	K_{15}	K_{16}	K_{17}	K_{18}
K_{21}	K_{22}	K_{23}	K_{24}	K_{35}	K_{26}	K_{27}	K_{28}
K_{31}	K_{32}	K_{33}	K_{34}	K_{35}	K_{36}	K_{37}	K_{38}
K_{41}	K_{42}	K_{43}	K_{44}	K_{45}	K_{46}	K_{47}	K_{48}
K_{51}	K_{52}	K_{53}	K_{54}	K_{55}	K_{56}	K_{57}	K_{58}
K_{61}	K_{62}	K_{63}	K_{64}	K_{65}	K_{66}	K_{67}	K_{68}
K_{71}	K_{72}	K_{73}	K_{74}	K_{75}	K_{76}	K_{77}	K_{78}
K_{81}	K_{82}	K_{83}	K_{84}	K_{85}	K_{86}	K_{87}	K_{88}

- Quantize each 8x8 grid using an appropriate quantization table
  - More on this in the next slide
- Store quantized DCT coefficients
  - The individual grid cells and the entire image can be recovered from the quantized DCT coefficients as explained earlier

The Jpeg Quantization tables

- Separate quantization tables are computed for the Y, Cb and Cr channels
  - Cb and Cr can tolerate greater quantization than Y
- The quantization tables must be stored along with the compressed image
  - Otherwise we cannot reconstruct the image
- Appropriate quantization tables are critical for good compression and reconstruction
  - A “quality factor” Q may be used to decide the degree of quantization
- Most cameras will use customized quantization tables that are optimized for the individual images
  - Using proprietary algorithms which are company specific
  - This has forensic implications!

Storing the Image

- The stored image includes quantized DCT coefficients and the quantization tables

- It also includes a lot of additional meta data. This is generally stored in the form of an "**EXIF**" table

Storage: The Exchangeable Image Format (EXIF)

- A standardized format for images
  - Agreed upon by a number of standardization bodies
- Specifies formats for images, sounds, and other tags used by digital image recorders (cameras, scanners, etc.)
- Standard includes format specifications for
  - JPEG images
  - TIFF images (Tagged Image File Format)
  - RIFF wav for audio files (Resource Interchange File Format)
  - Appended metadata

EXIF example

from wikipedia

Tag	Value
Manufacturer	CASIO
Model	QV-4000
Orientation (rotation)	top-left [8 possible values[20]]
Software	Ver1.01
Date and time	2003:08:11 16:45:32
YCbCr positioning	centered
Compression	JPEG compression
X resolution	72.00
Y resolution	72.00
Resolution unit	Inch
Exposure time	1/659 s
F-number	f/4.0
Exposure program	Normal program
Exif version	Exif version 2.1
Date and time (original)	2003:08:11 16:45:32
Date and time (digitized)	2003:08:11 16:45:32
Components configuration	Y Cb Cr –
Compressed bits per pixel	4.01
Exposure bias	0.0
Max. aperture value	2.00
Metering mode	Pattern
Flash	Flash did not fire
Focal length	20.1 mm
MakerNote	432 bytes unknown data
FlashPix version	FlashPix version 1.0
Color space	sRGB
Pixel X dimension	2240
Pixel Y dimension	1680
File source	DSC
Interoperability index	R98
Interoperability version	(null)

- Typical tags for images (note the detail)

In the next lecture

- Digital multimedia: Recording and devices
  - Audio
  - Images
  - Video
  - Text
- Digital multimedia: Processing
  - Audio processing
  - Two generic processing techniques