Machine Learning for Signal Processing

Lecture 1: Signal Representations

Class 1. 29 August 2013

Instructor: Bhiksha Raj

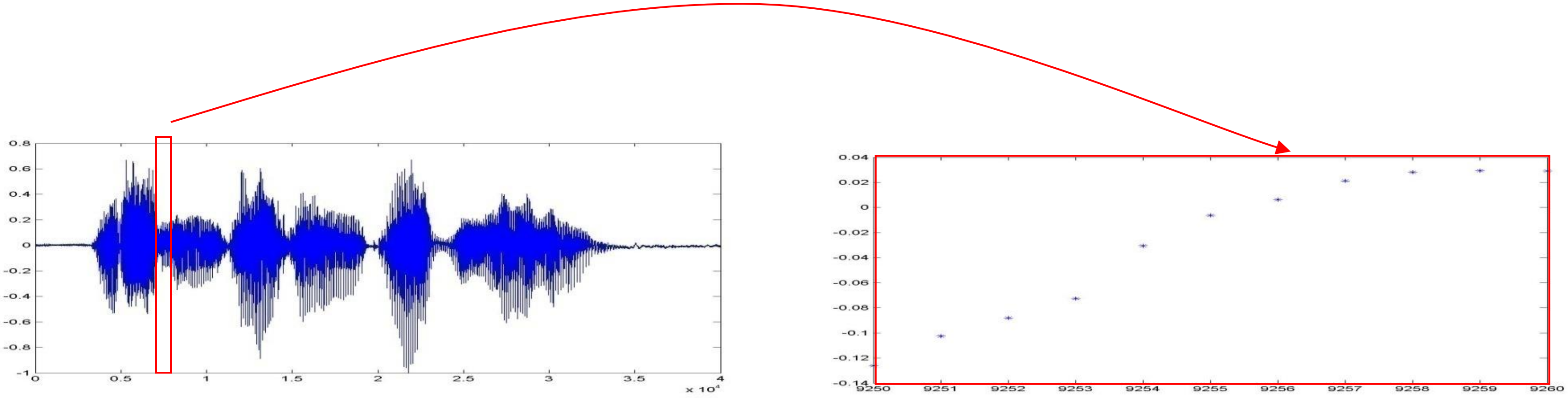
What is a signal

- A mechanism for conveying information
  - Semaphores, gestures, traffic lights..

- Electrical engineering: currents, voltages

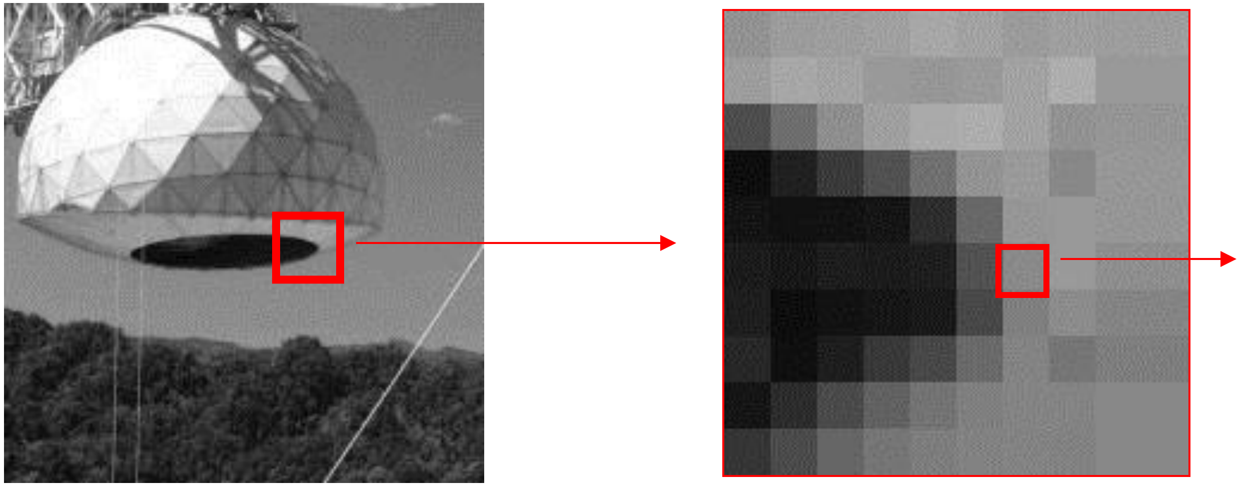
- Digital signals: Ordered collections of numbers that convey information
  - from a source to a destination
  - about a real world phenomenon
    - Sounds, images

Signal Examples: Audio



- A sequence of numbers
  - [n_1 n_2 n_3 n_4 …]
  - The order in which the numbers occur is important
    - Ordered
    - In this case, a *time series*
  - Represent a perceivable sound

Example: Images



Pixel = 0.5

- A rectangular arrangement (matrix) of numbers
  - Or sets of numbers (for color images)
- Each pixel represents a visual representation of one of these numbers
  - 0 is minimum / black, 1 is maximum / white
  - Position / order is important

What is Signal Processing

- Acquisition, Analysis, Interpretation, and Manipulation of signals.
  - Acquisition: Sampling, sensing
  - Decomposition: Fourier transforms, wavelet transforms, dictionary-based representations
  - Denoising signals
  - Coding: GSM, Jpeg, Mpeg, Ogg Vorbis
  - Detection: Radars, Sonars
  - Pattern matching: Biometrics, Iris recognition, finger print recognition
  - Etc.

What is Machine Learning

- The science that deals with the development of algorithms that can learn from data
  - Learning patterns in data
    - Automatic categorization of text into categories; Market basket analysis
  - Learning to classify between different kinds of data
    - Spam filtering: Valid email or junk?
  - Learning to predict data
    - Weather prediction, movie recommendation
- Statistical analysis and pattern recognition when performed by a computer scientist..

MLSP

- Application of Machine Learning techniques to the analysis of signals
  - Such as audio, images, video, etc.

- *Data driven* analysis of signals
  - *Characterizing* signals
    - What are they composed of?
  - *Detecting* signals
    - Radars. Face detection. Speaker verification
  - *Recognize* signals
    - Face recognition. Speech recognition.
  - *Predict* signals
  - Etc..

In this course

- Jetting through fundamentals:
  - Linear Algebra, Signal Processing, Probability
- Machine learning concepts
  - Methods of modelling, estimation, classification, prediction
- Applications:
  - Sounds:
    - Characterizing sounds, Denoising speech, Synthesizing speech, Separating sounds in mixtures, Music retrieval
  - Images:
    - Characterization, Object detection and recognition, Biometrics
  - Other forms of data
  - Representation
  - Sensing and recovery.
- Topics covered are representative
- Actual list to be covered may change, depending on how the course progresses

Recommended Background

- DSP
  - Fourier transforms, linear systems, basic statistical signal processing
- Linear Algebra
  - Definitions, vectors, matrices, operations, properties
- Probability
  - Basics: what is an random variable, probability distributions, functions of a random variable
- Machine learning
  - Learning, modelling and classification techniques

Guest Lectures

- Fernando de la Torre
  - Component Analysis
- Roger Dannenberg
  - Music Understanding
- Aswin Sankarnarayanan
  - Compressive Sensing
- Marios Savvides
  - Visual biometrics
- Ajay Diwakaran
  - Multimedia analysis
- Yaser Sheikh
  - Structure from motion

Travels..

- I will be travelling in Oct/Nov:
  - 28 Oct – 1 Nov: Lisbon
  - 2 Nov – 6 Nov: Berlin

- We will have four guest lectures in this period

Schedule of Other Lectures

- Tentative Schedule on Website
- http://mlsp.cs.cmu.edu/courses/fall2013

Grading

- Homework assignments : 50%
  - Mini projects
  - Will be assigned during course
  - Minimum 3, Maximum 4
  - You *will not catch up* if you slack on any homework
    - Those who didn’t slack will also do the next homework

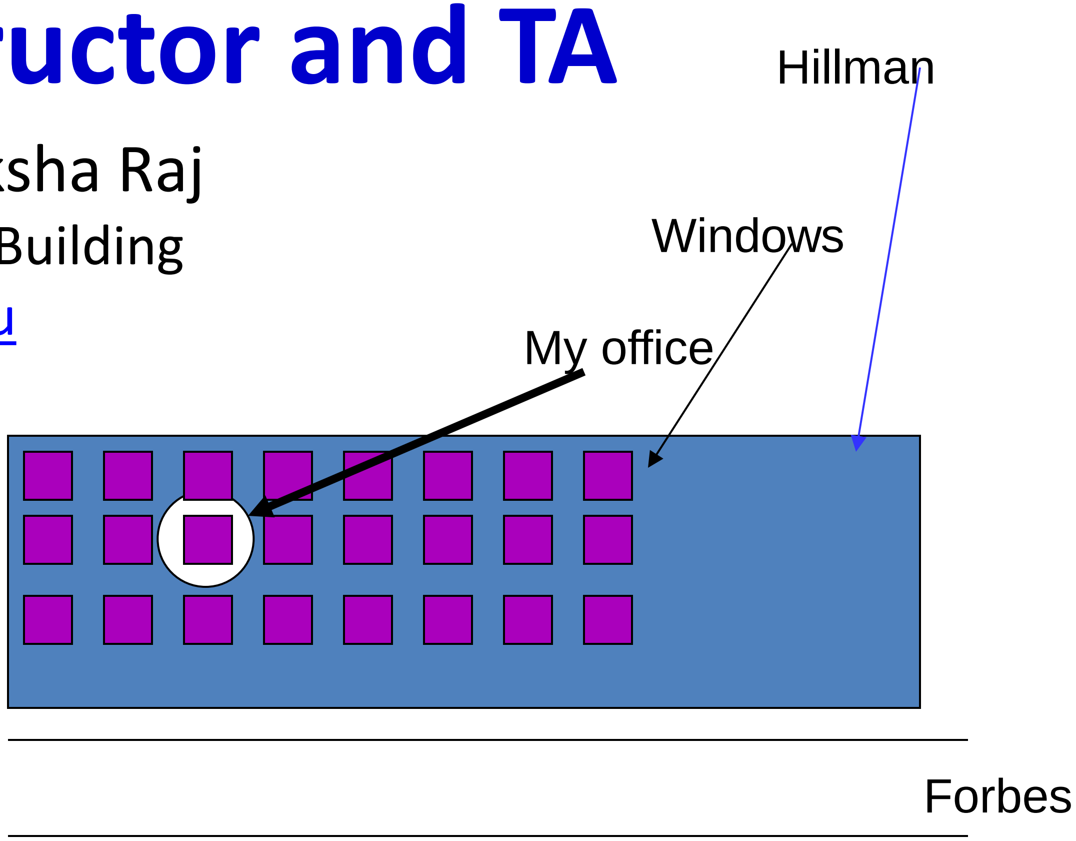
- Final project: 50%
  - Will be assigned early in course
  - Dec 5: Poster presentation for all projects, with demos (if possible)
    - Partially graded by visitors to the poster

Projects

- Previous projects (partially) accessible from web pages for prior years
- Expect significant supervision
- Outcomes from previous years
  - 10+ papers
  - 2 best paper awards
  - 1 PhD thesis
  - Several masters' theses

Instructor and TA

- Instructor: Prof. Bhiksha Raj
  - Room 6705 Hillman Building
  - bhiksha@cs.cmu.edu
  - 412 268 9826



- TAs:
  - James Ding
    - dingyingjian@gmail.com
  - Varun Gupta
    - vgupta1@andrew.cmu.edu
- Office Hours:
  - Bhiksha Raj: Wed 3:30-4.30
  - TA: TBD

Additional Administrivia

- Website:
  - http://mlsp.cs.cmu.edu/courses/fall2013/
  - Lecture material will be posted on the day of each class on the website
  - Reading material and pointers to additional information will be on the website

- Mailing list: Use blackboard
  - All notices will be posted there

Additional Administrivia

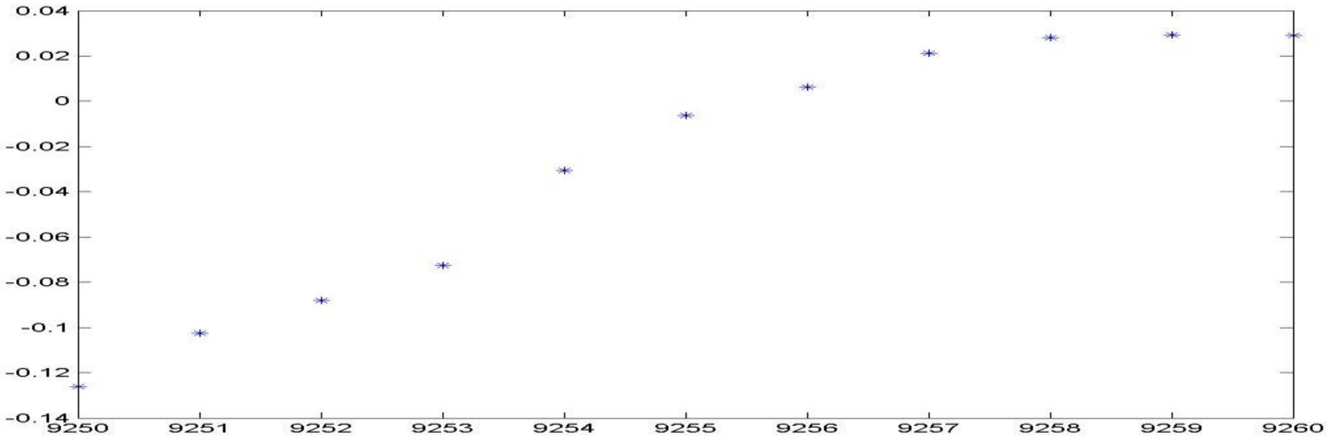
- How many on waitlist?

Representing Data

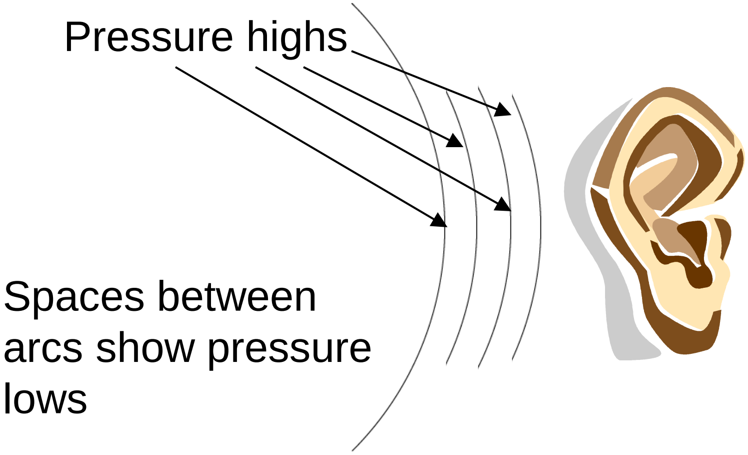
- Audio
- Images
  - Video
- Other types of signals
  - In a manner similar to one of the above

What is an audio signal

- A typical digital audio signal
  - It’s a sequence of points



Where do these numbers come from?

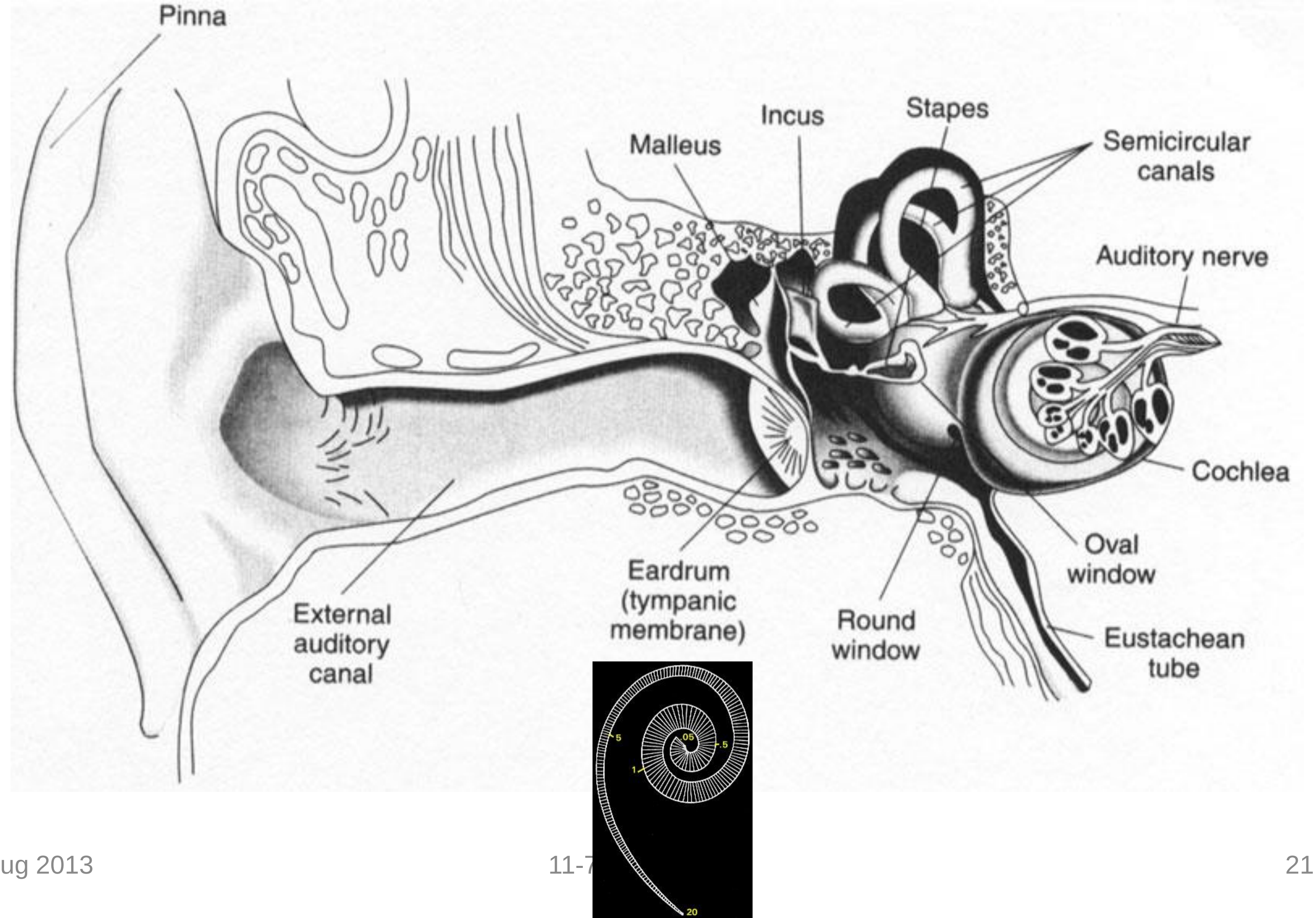


Pressure highs

Spaces between arcs show pressure lows

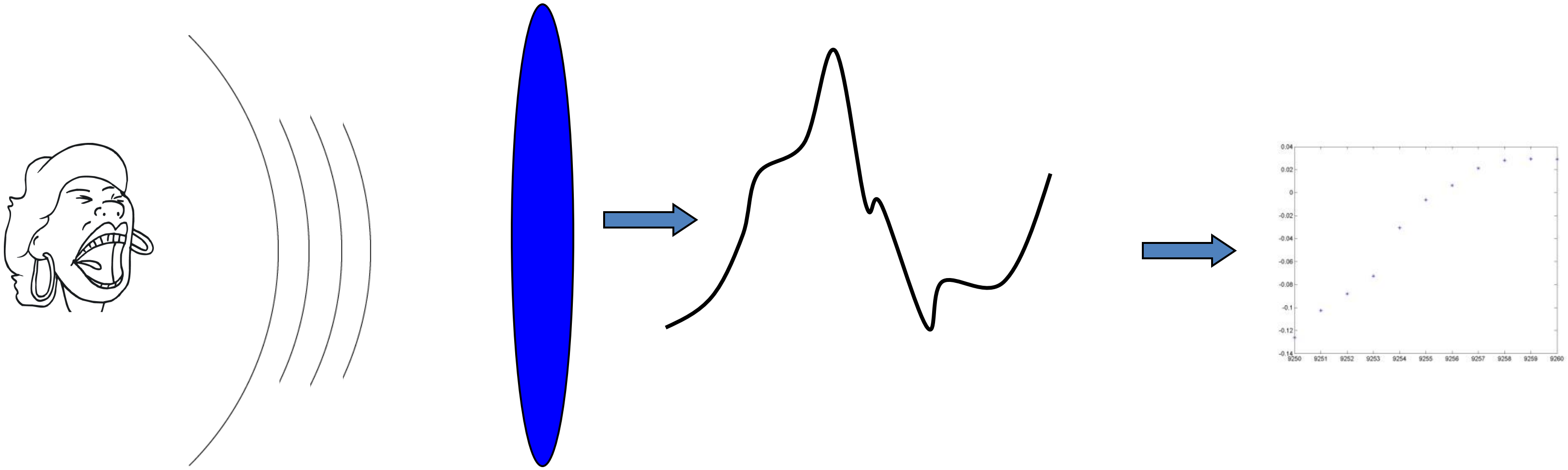
- Any sound is a pressure wave: alternating highs and lows of air pressure moving through the air
- When we speak, we produce these pressure waves
  - Essentially by producing puff after puff of air
  - Any sound producing mechanism actually produces pressure waves
- These pressure waves move the eardrum
  - Highs push it in, lows suck it out
  - We sense these motions of our eardrum as “sound”

SOUND PERCEPTION



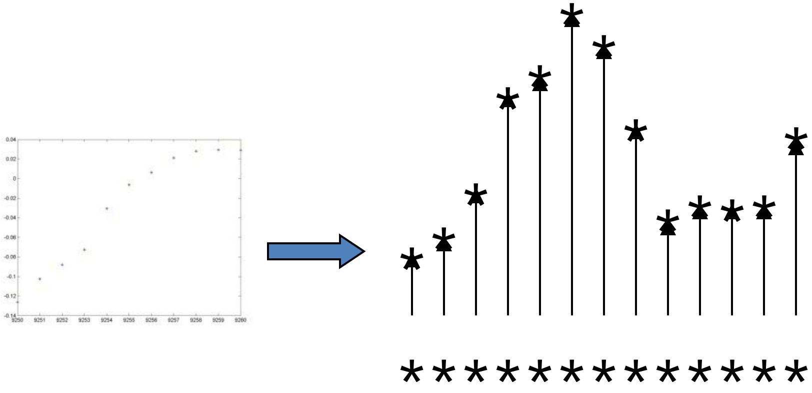
Storing pressure waves on a computer

- The pressure wave moves a diaphragm
  - On the microphone
- The motion of the diaphragm is converted to continuous variations of an electrical signal
  - Many ways to do this
- A "sampler" samples the continuous signal at regular intervals of time and stores the numbers



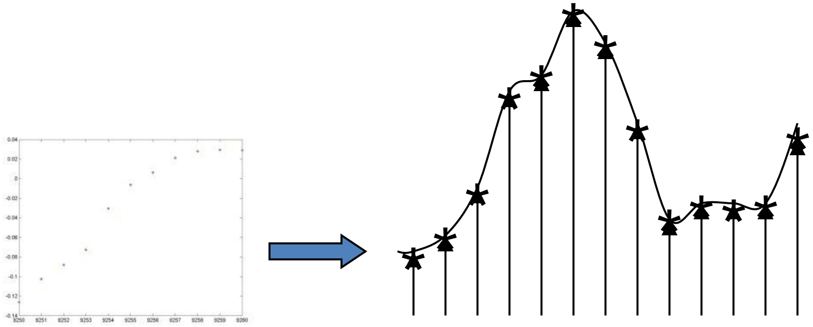
Are these numbers sound?

- How do we even know that the numbers we store on the computer have anything to do with the recorded sound really?
  - Recreate the sense of sound
- The numbers are used to control the levels of an electrical signal



Are these numbers sound?

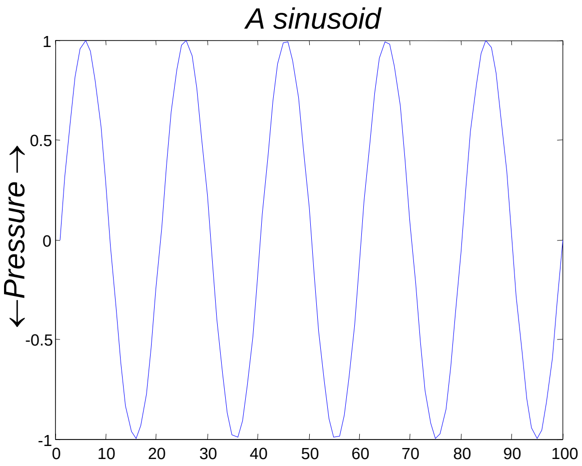
- How do we even know that the numbers we store on the computer have anything to do with the recorded sound really?
  - Recreate the sense of sound
- The numbers are used to control the levels of an electrical signal
- The electrical signal moves a diaphragm back and forth to produce a pressure wave
  - That we sense as sound

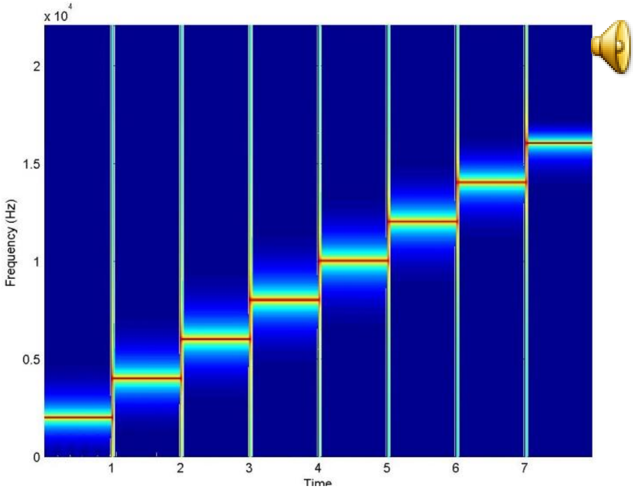


****

How many samples a second

- Convenient to think of sound in terms of sinusoids with frequency
- Sounds may be modelled as the sum of many sinusoids of different frequencies
  - Frequency is a physically motivated unit
  - Each hair cell in our inner ear is tuned to specific frequency
- Any sound has many frequency components
  - We can hear frequencies up to 16000Hz
    - Frequency components above 16000Hz can be heard by children and some young adults
    - Nearly nobody can hear over 20000Hz.



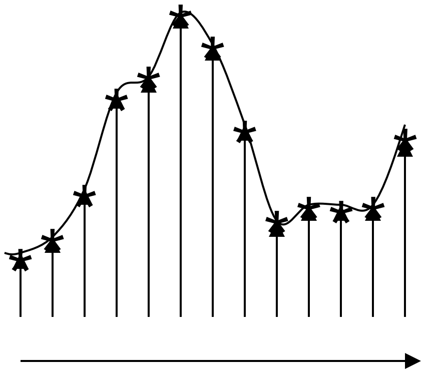


Signal representation - Sampling

- *Sampling frequency* (or *sampling* rate) refers to the number of samples taken a second

- Sampling rate is measured in Hz
  - We need a sample rate twice as high as the highest frequency we want to represent (Nyquist freq)

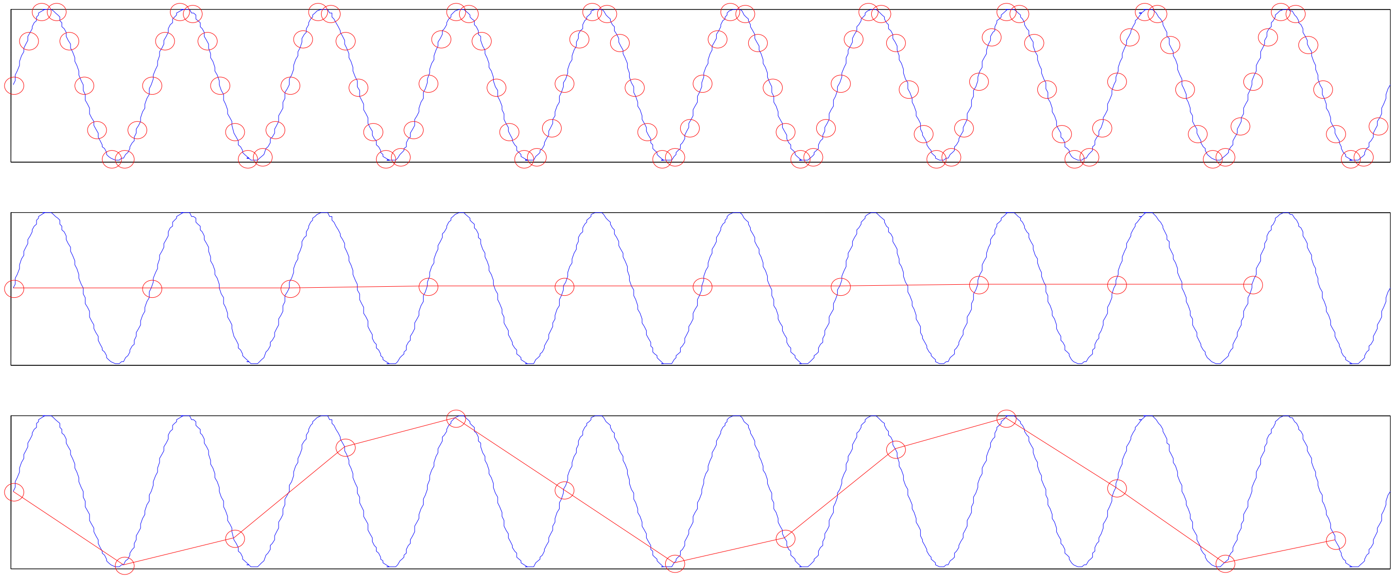
- For our ears this means a sample rate of at least 40kHz
  - Because we hear up to 20kHz



Time in secs.

Aliasing

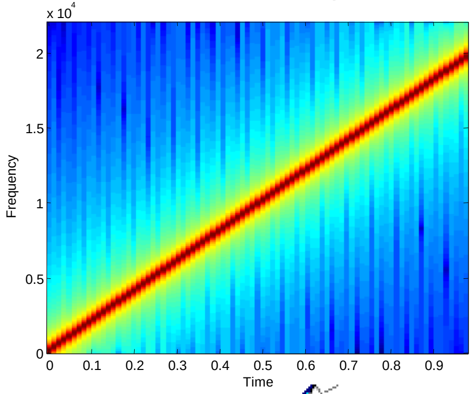
- Low sample rates result in *aliasing*
  - High frequencies are misrepresented
  - Frequency f_1 will become (sample rate – f_1)
  - In video also when you see wheels go backwards



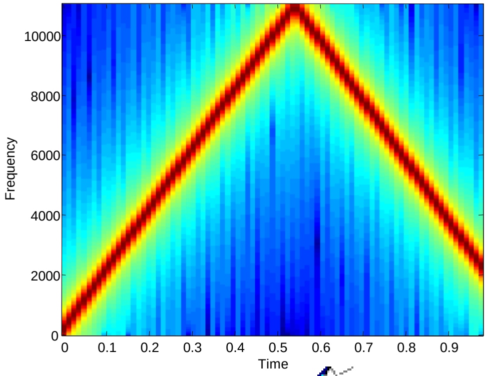
Aliasing examples

Sinusoid sweeping from 0Hz to 20kHz

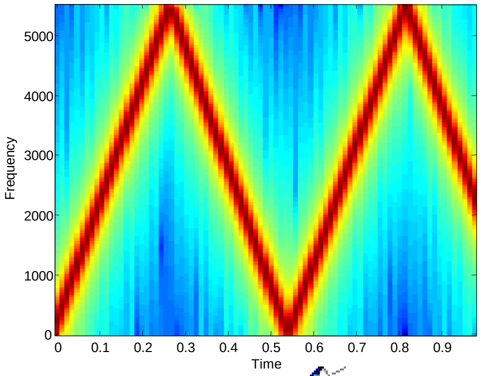
44.1kHz SR, is ok



22kHz SR, aliasing!



11kHz SR, double aliasing!

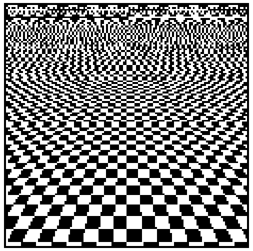


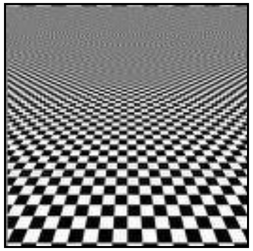
On real sounds

at 44kHz *at 11kHz* *at 4kHz*

at 22kHz *at 5kHz* *at 3kHz*

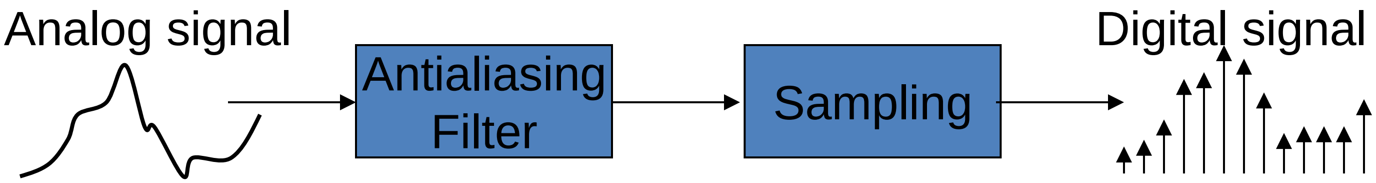
On images





On video

Avoiding Aliasing



- Sound naturally has all perceivable frequencies
  - And then some
  - Cannot control the rate of variation of pressure waves in nature

- Sampling at *any* rate *will* result in aliasing
- Solution: *Filter the electrical signal* before sampling it
  - Cut off all frequencies above sampling.frequency/2
  - E.g., to sample at 44.1Khz, filter the signal to eliminate all frequencies above 22050 Hz

Typical Sampling Rates

- Common sample rates
  - For speech 8kHz to 16kHz
  - For music 32kHz to 44.1kHz
  - Pro-equipment 96kHz

Storing numbers on the Computer

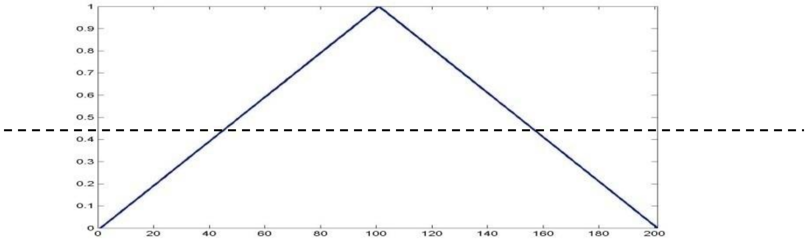
- Sound is the outcome of a continuous range of variations
  - The pressure wave can take any value (within limits)
  - The diaphragm can also move continuously
  - The electrical signal from the diaphragm has continuous variations

- A computer has finite resolution
  - Numbers can only be stored to finite resolution
  - E.g. a 16-bit number can store only 65536 values, while a 4-bit number can store only 16 values
  - To store the sound wave on the computer, the continuous variation must be “mapped” on to the discrete set of numbers we can store

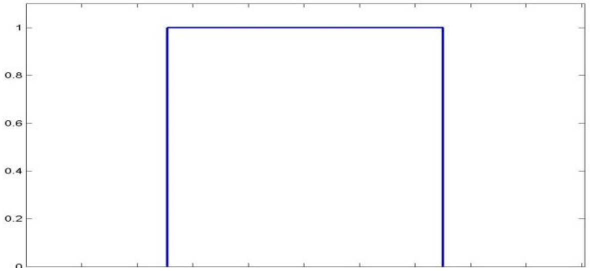
Mapping signals into bits

- Example of 1-bit sampling table

Signal Value	Bit sequence	Mapped to
S > 2.5v	1	1 * const
S <=2.5v	0	0



Original Signal

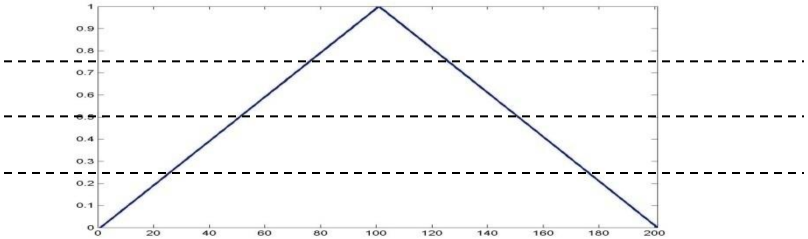


Quantized approximation

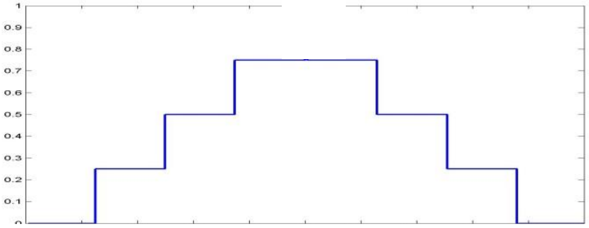
Mapping signals into bits

- Example of 2-bit sampling table

Signal Value	Bit sequence	Mapped to
S >= 3.75v	11	3 * const
3.75v > S >= 2.5v	10	2 * const
2.5v > S >= 1.25v	01	1 * const
1.25v > S >= 0v	0	0



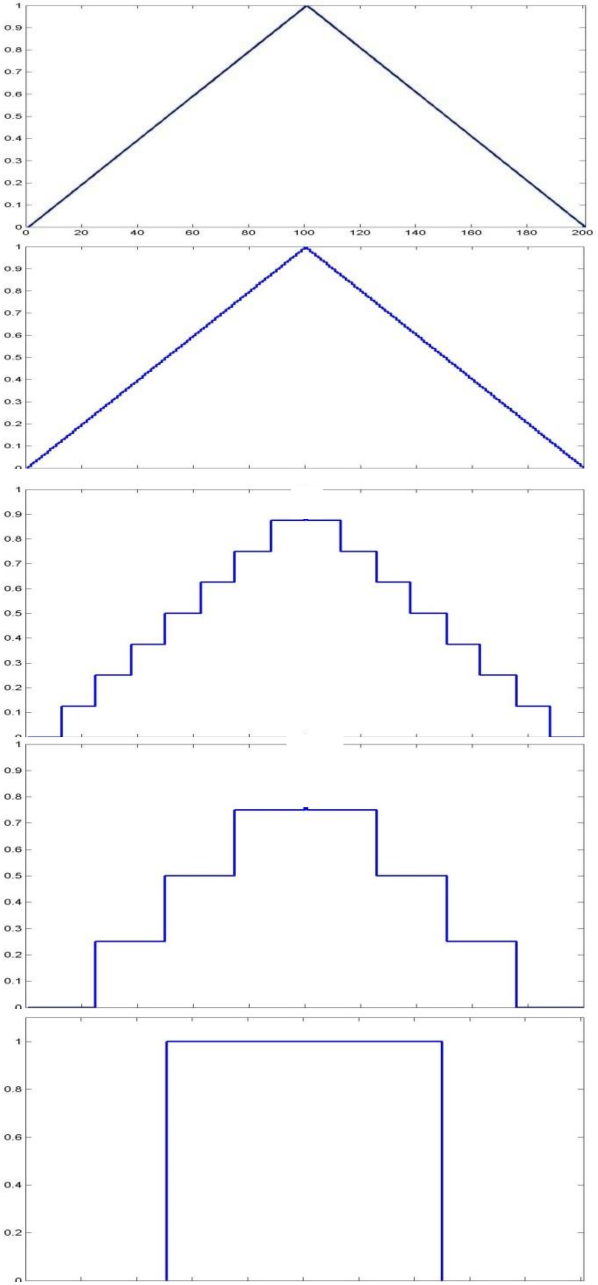
Original Signal



Quantized approximation

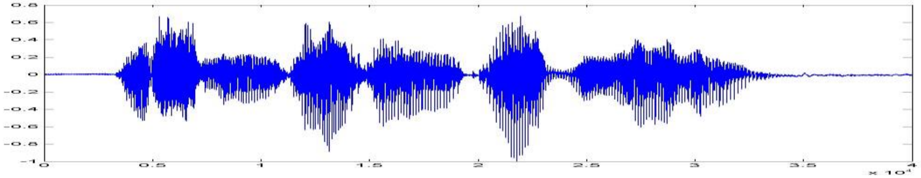
Storing the signal on a computer

- The original signal
- 8 bit quantization
- 3 bit quantization
- 2 bit quantization
- 1 bit quantization

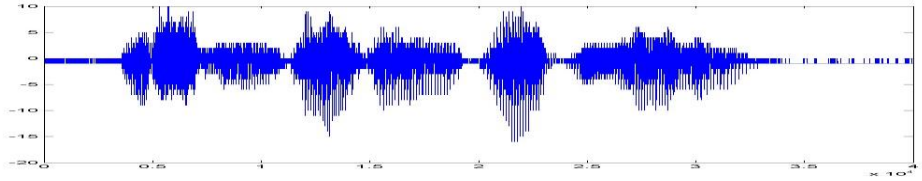


Tom Sullivan Says his Name

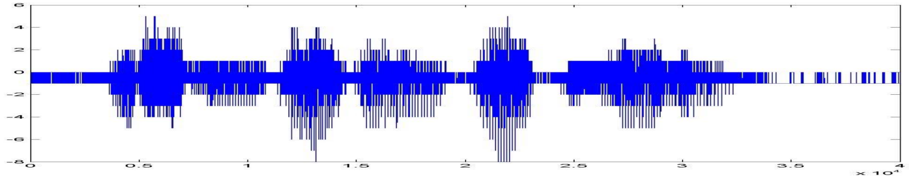
- 16 bit sampling



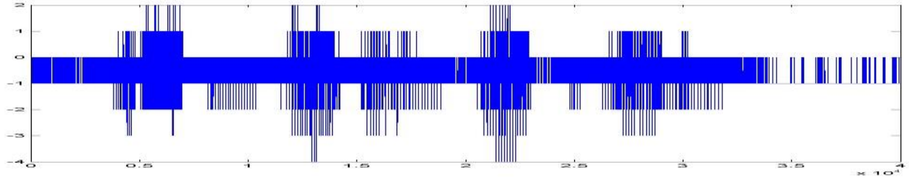
- 5 bit sampling



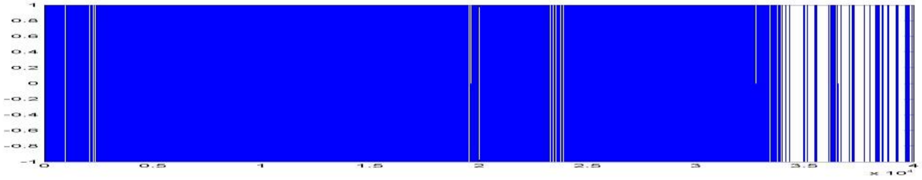
- 4 bit sampling



- 3 bit sampling

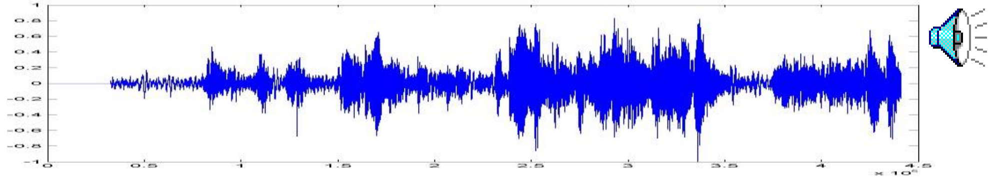


- 1 bit sampling

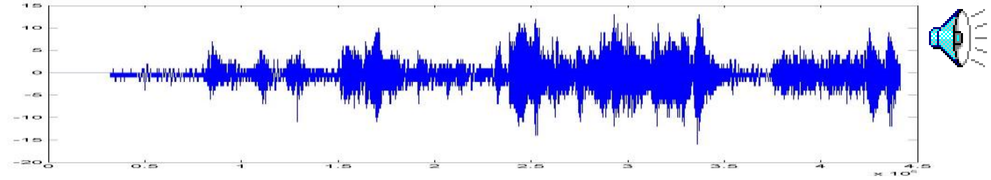


A Schubert Piece

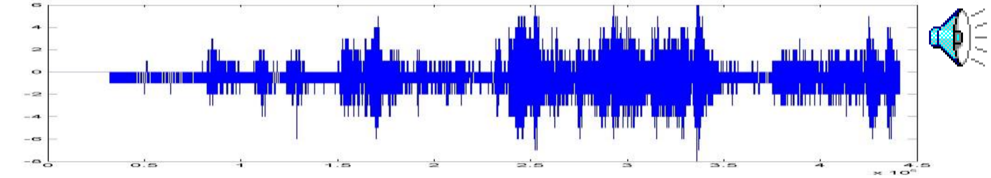
- 16 bit sampling



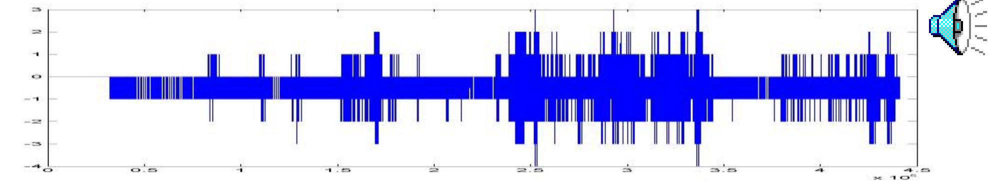
- 5 bit sampling



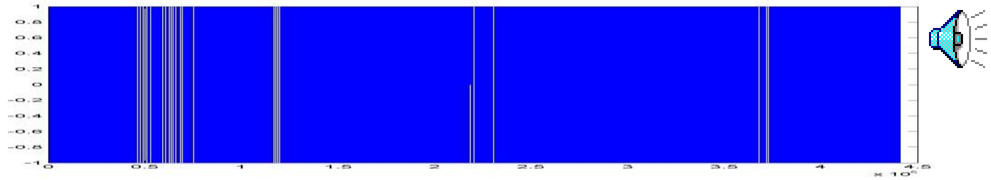
- 4 bit sampling



- 3 bit sampling



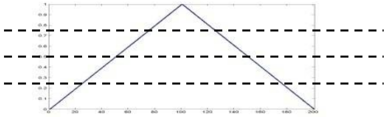
- 1 bit sampling

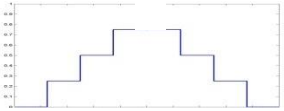


Quantization Formats

- Sampling can be uniform
  - Sample values equally spaced out

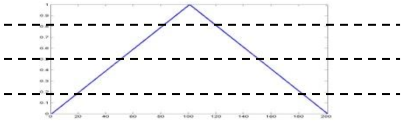
Signal Value	Bits	Mapped to
S >= 3.75v	11	3 * const
3.75v > S >= 2.5v	10	2 * const
2.5v > S >= 1.25v	01	1 * const
1.25v > S >= 0v	0	0

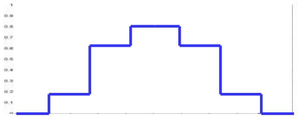




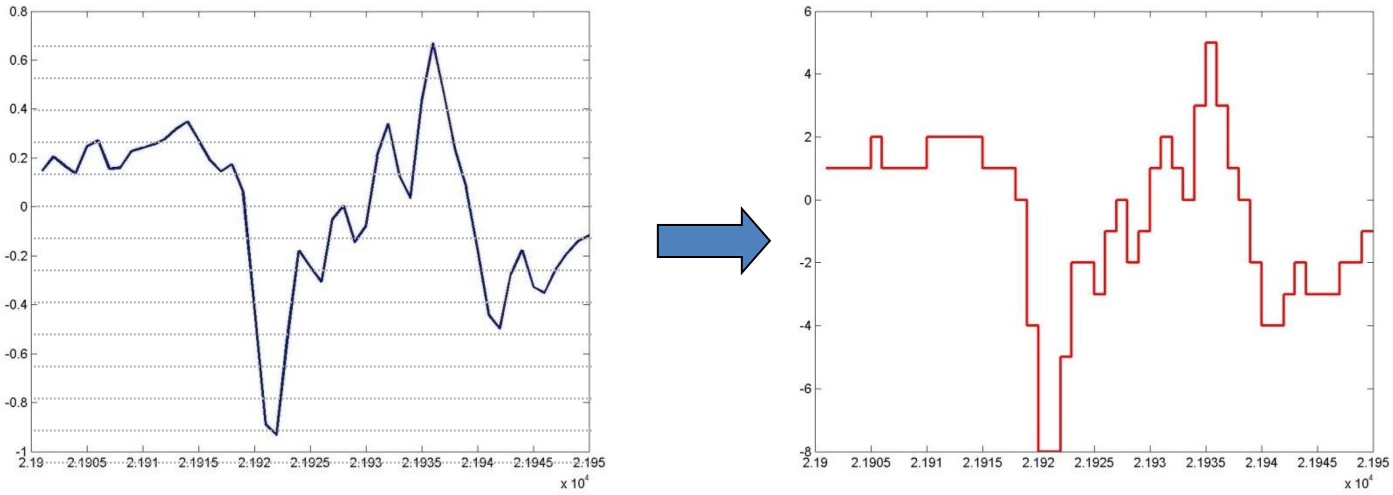
- Or nonuniform

Signal Value	Bits	Mapped to
S >= 4v	11	4.5 * const
4v > S >= 2.5v	10	3.25 * const
2.5v > S >= 1v	01	1.25 * const
1.0v > S >= 0v	0	0.5 * const



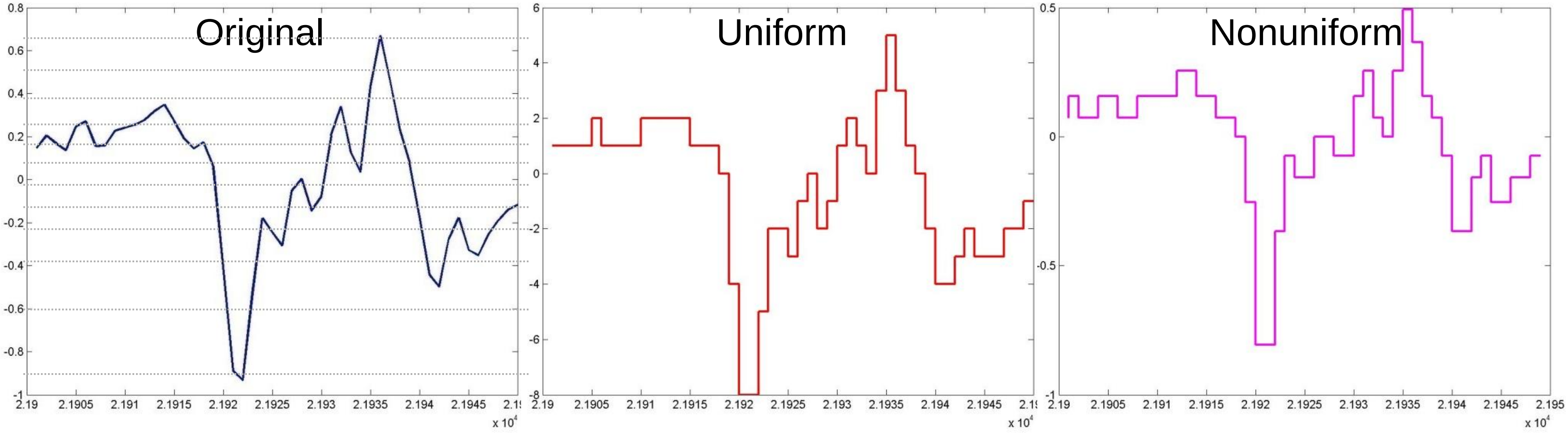


Uniform Quantization



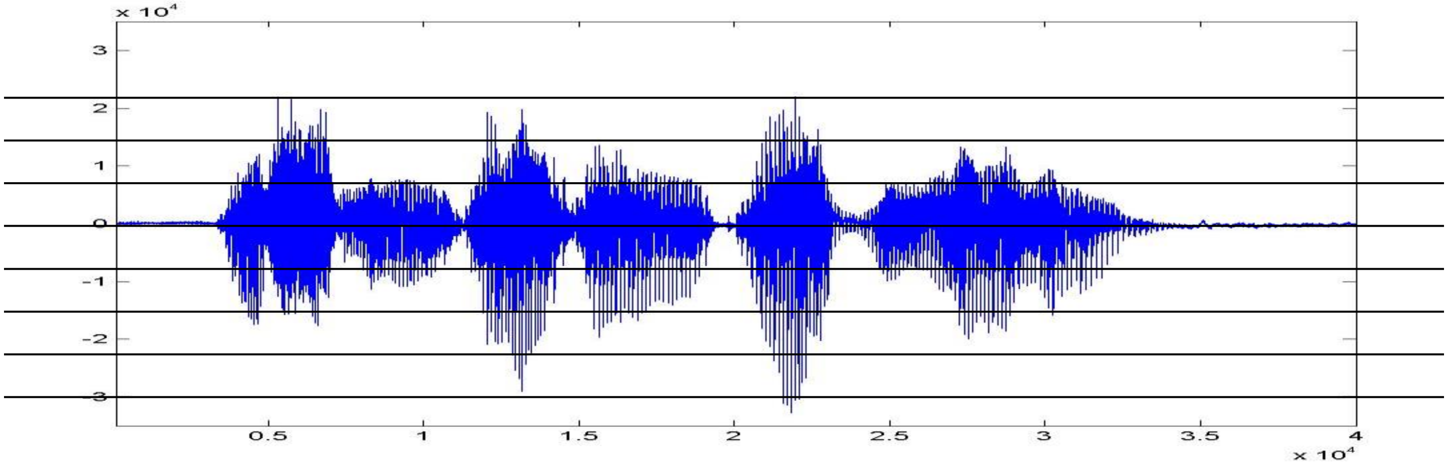
- At the sampling instant, the actual value of the waveform is rounded off to the nearest level permitted by the quantization
- Values entirely outside the range are quantized to either the highest or lowest values

Non-uniform Sampling

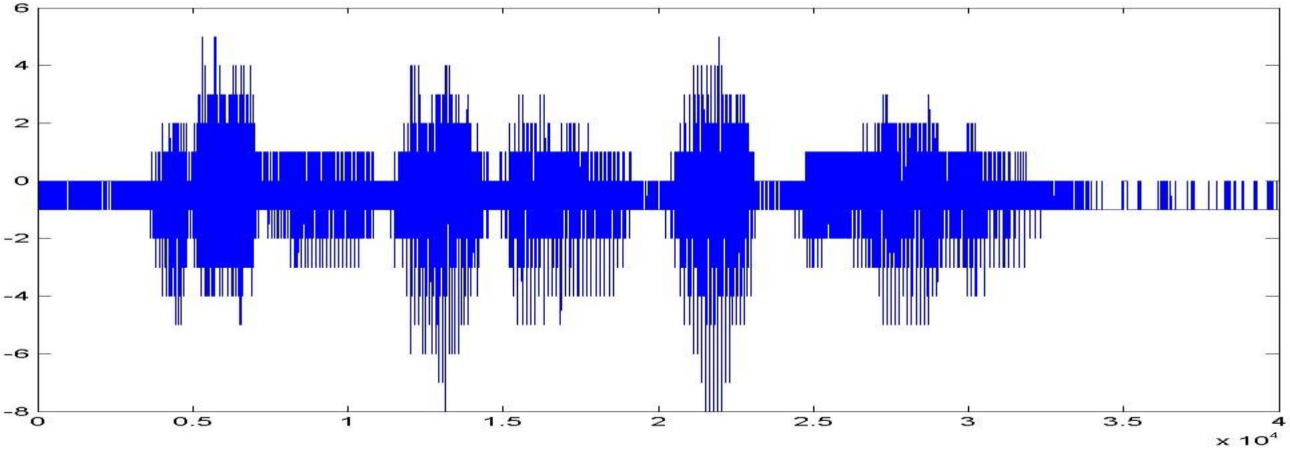


- Quantization levels are non-uniformly spaced
- At the sampling instant, the actual value of the waveform is rounded off to the nearest level permitted by the quantization
- Values entirely outside the range are quantized to either the highest or lowest values

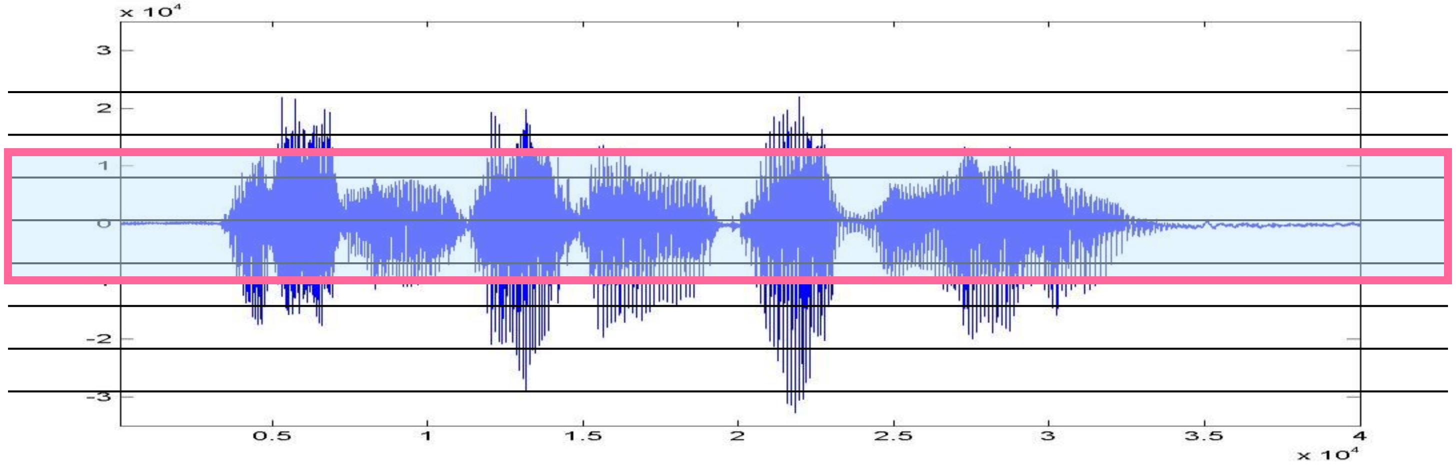
Uniform Quantization



UPON BEING SAMPLED AT ONLY 3 BITS (8 LEVELS)

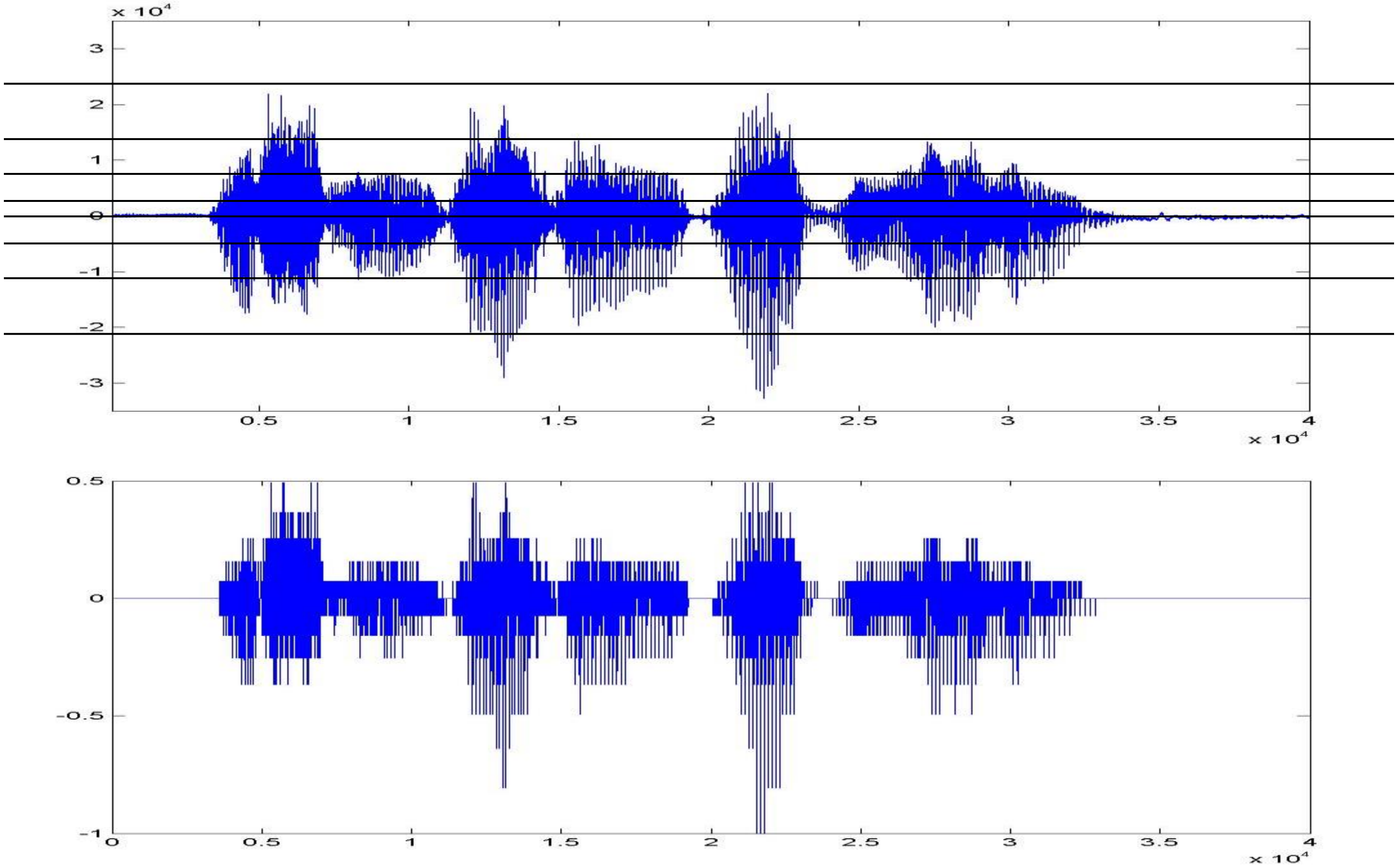


Uniform Quantization



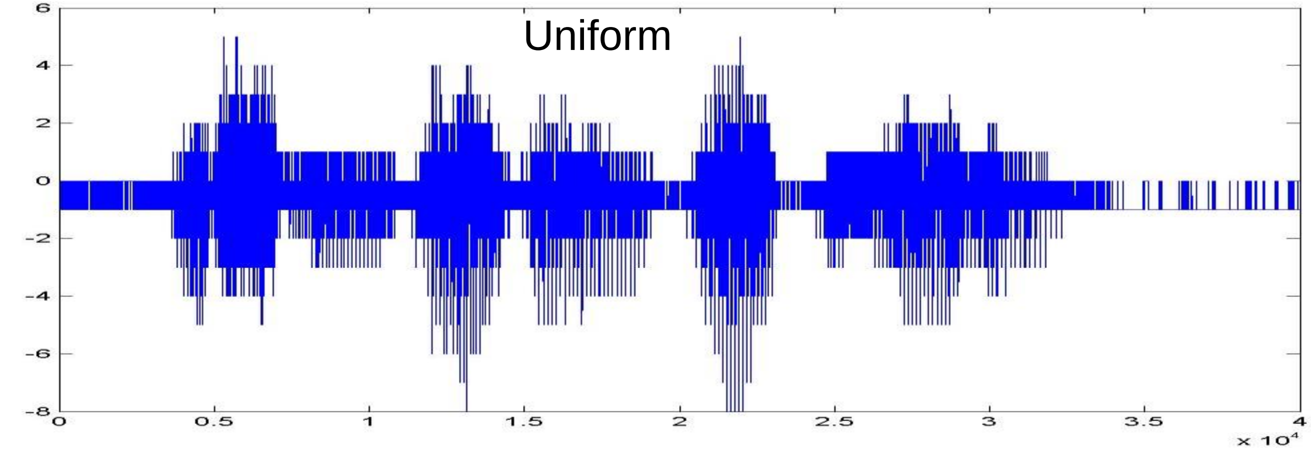
- There is a lot more action in the central region than outside.
- Assigning only four levels to the busy central region and four entire levels to the sparse outer region is inefficient
- Assigning more levels to the central region and less to the outer region can give better fidelity
  - for the same storage

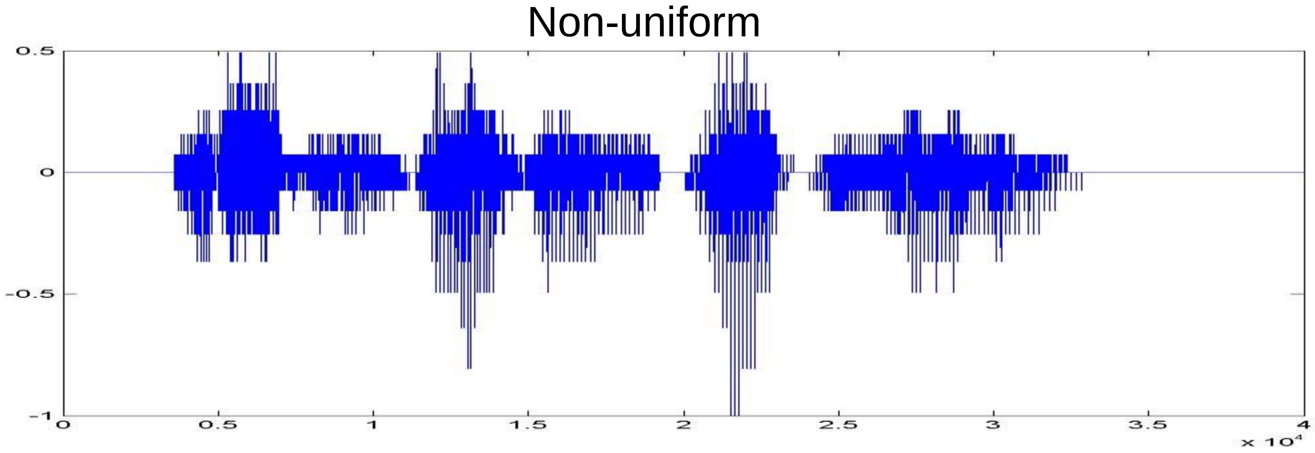
Non-uniform Quantization



- Assigning more levels to the central region and less to the outer region can give better fidelity for the same storage

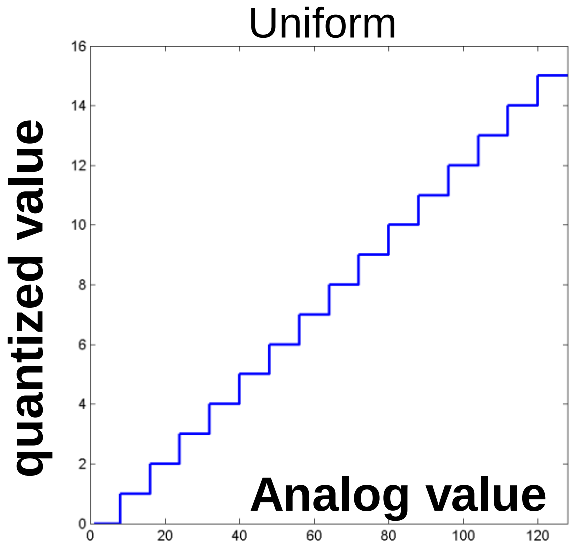
Non-uniform Quantization

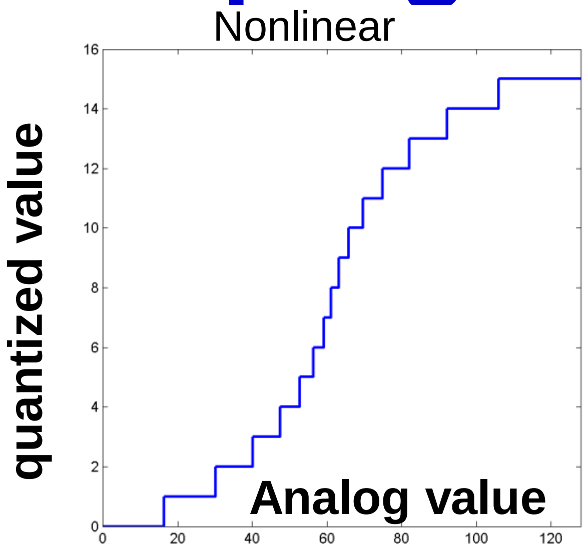

Uniform


Non-uniform

- Assigning more levels to the central region and less to the outer region can give better fidelity for the same storage

Non-uniform Sampling





- Uniform sampling maps uniform widths of the analog signal to units steps of the quantized signal
- In “standard” *non-uniform* sampling the step sizes are smaller near 0 and wider farther away
  - – The curve that the steps are drawn on follow a logarithmic law:
    - Mu Law: Y = C. log(1 + mX/C)/(1+m)
    - A Law: Y = C. (1 + log(a.X)/C)/(1+a)
- One can get the same perceptual effect with 8bits of non-uniform sampling as 12bits of uniform sampling

Dealing with audio

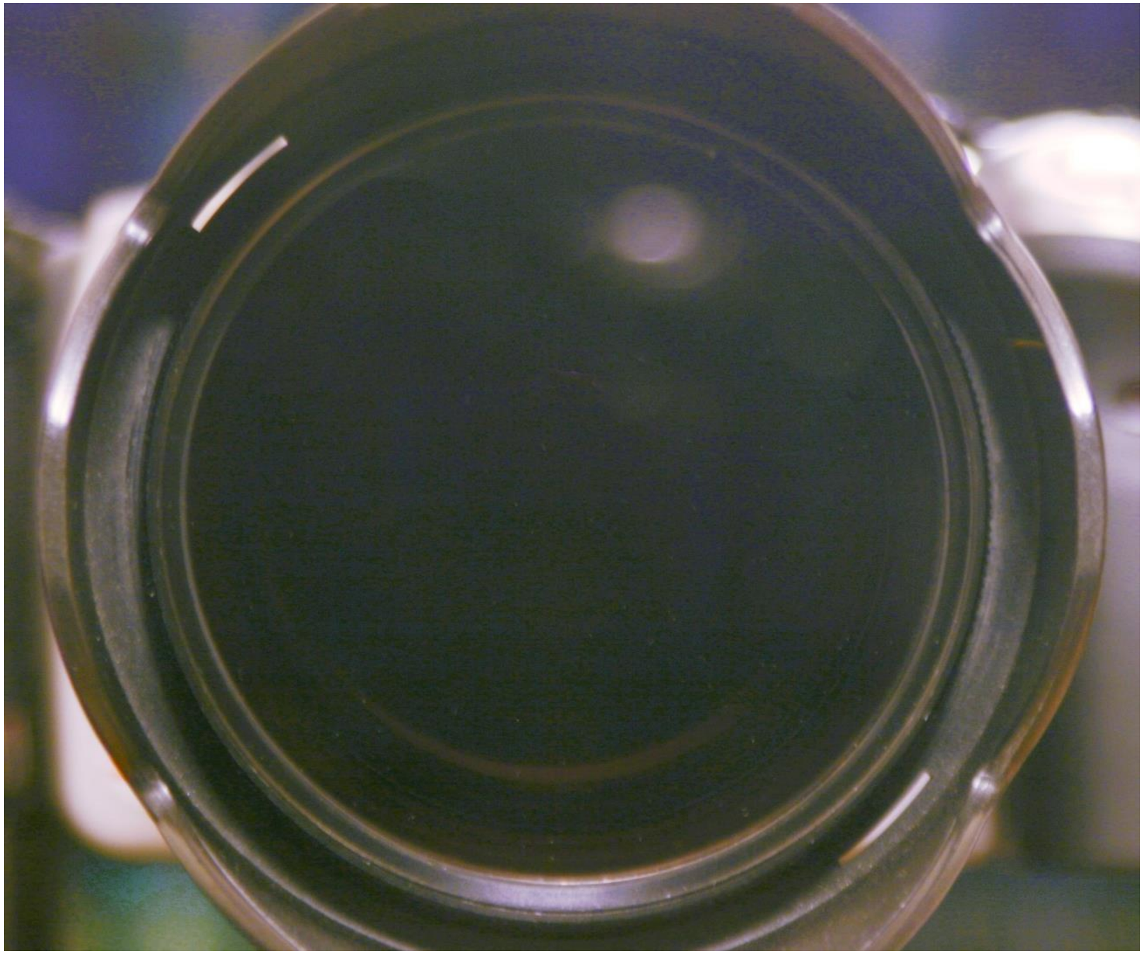
Signal Value	Bits	Mapped to
S >= 3.75v	11	3
3.75v > S >= 2.5v	10	2
2.5v > S >= 1.25v	01	1
1.25v > S >= 0v	0	0

Signal Value	Bits	Mapped to
S >= 4v	11	4.5
4v > S >= 2.5v	10	3.25
2.5v > S >= 1v	01	1.25
1.0v > S >= 0v	0	0.5

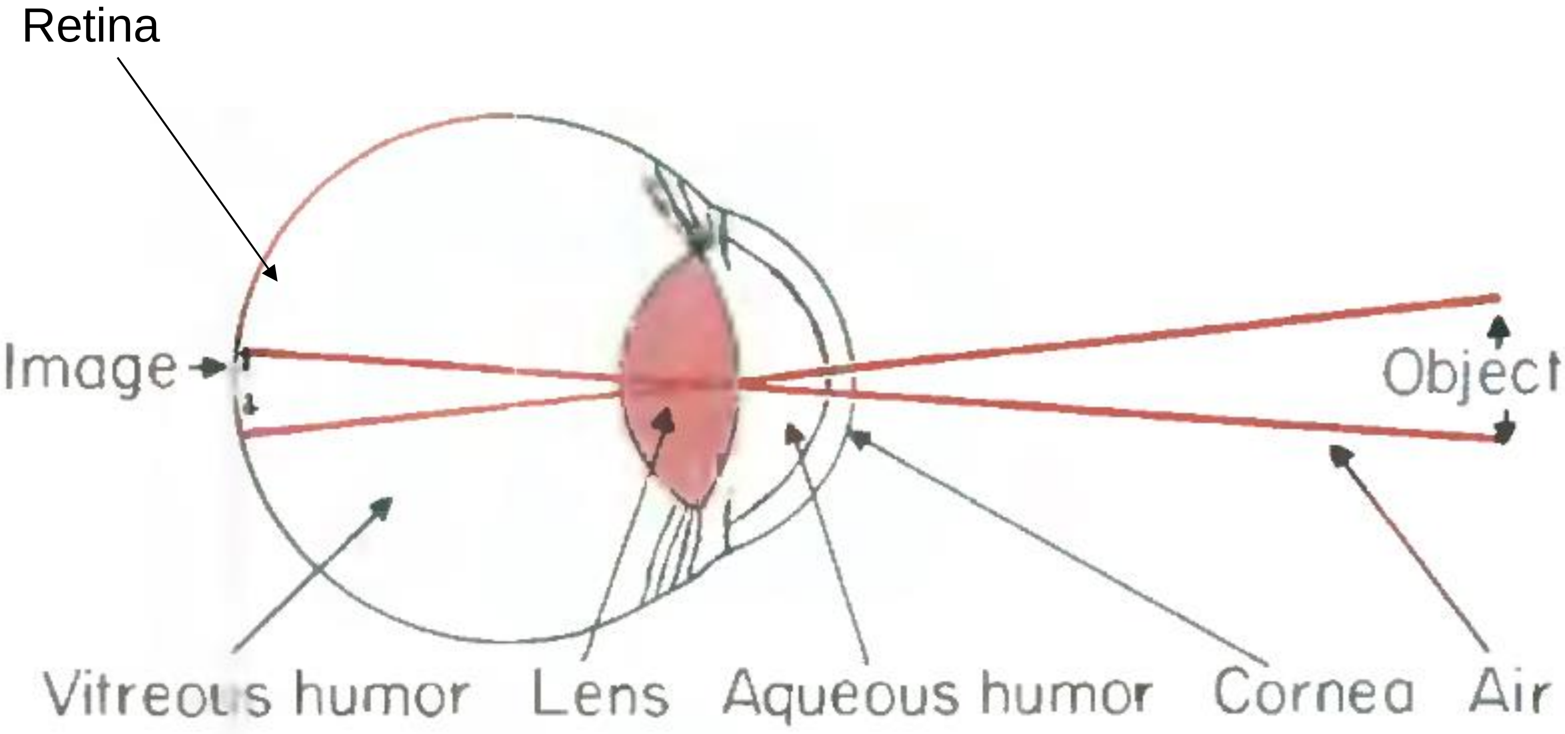
- Capture / read audio in the format provided by the file or hardware
  - Linear PCM, Mu-law, A-law,
- Convert to 16-bit PCM value
  - I.e. map the bits onto the number on the right column
  - This mapping is typically provided by a table computed from the sample compression function
  - No lookup for data stored in PCM
- Conversion from Mu law:
  - http://www.speech.cs.cmu.edu/comp.speech/Section2/Q2.7.html

Images

Images

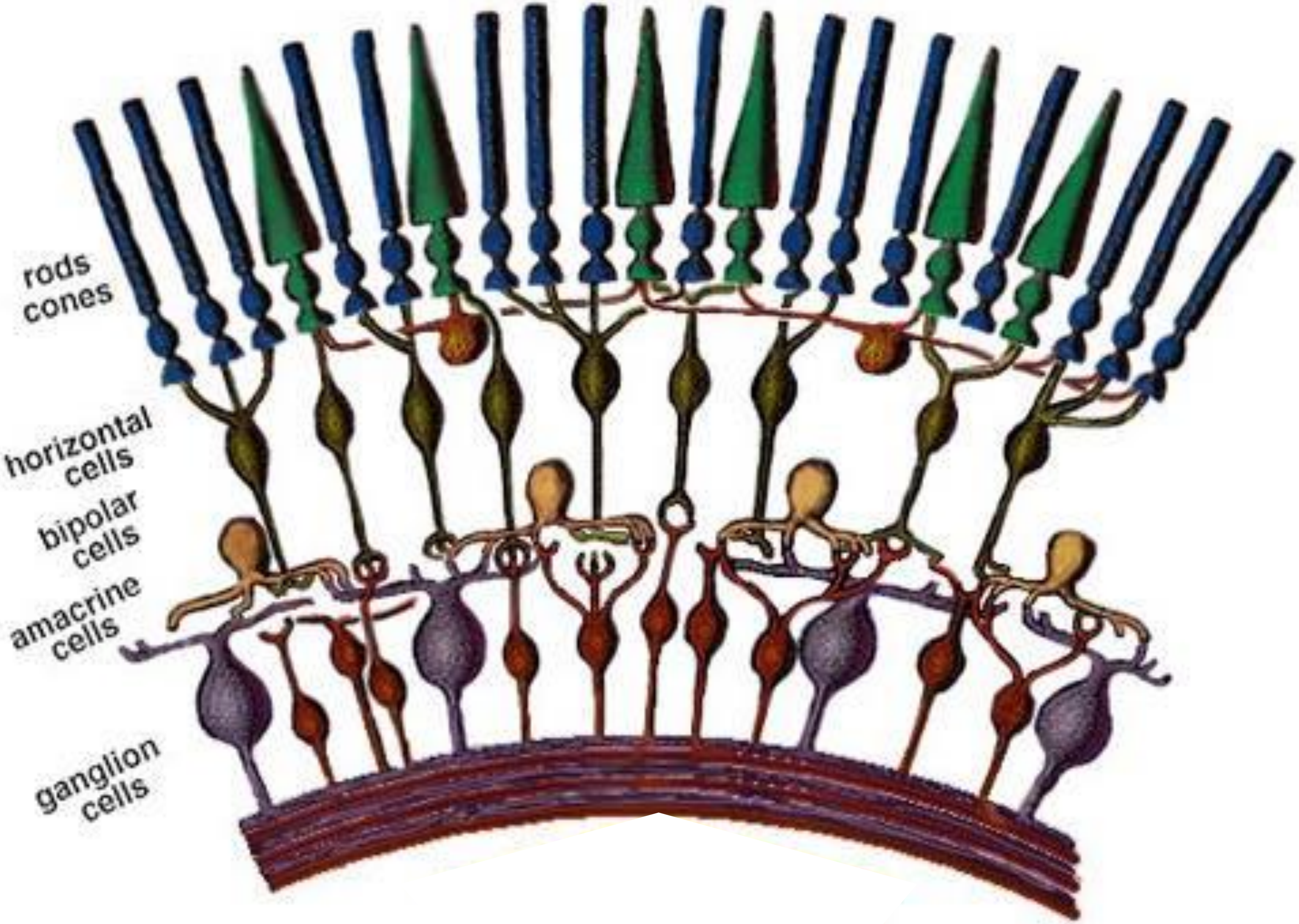


The Eye



Retina

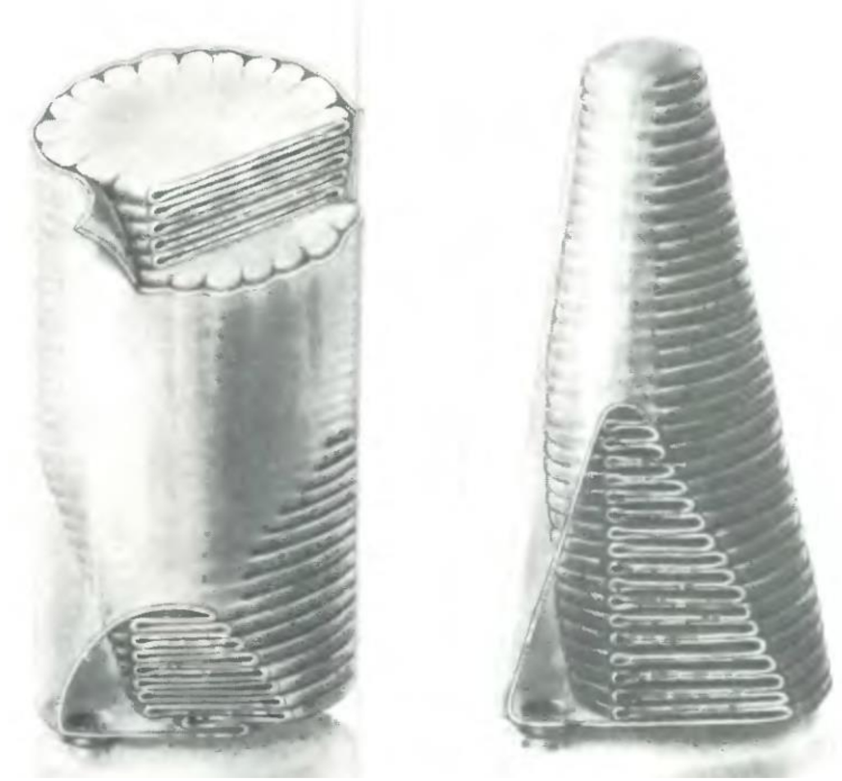
Basic Neuroscience: Anatomy and Physiology Arthur C. Guyton, M.D. 1987 W.B.Saunders Co.



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

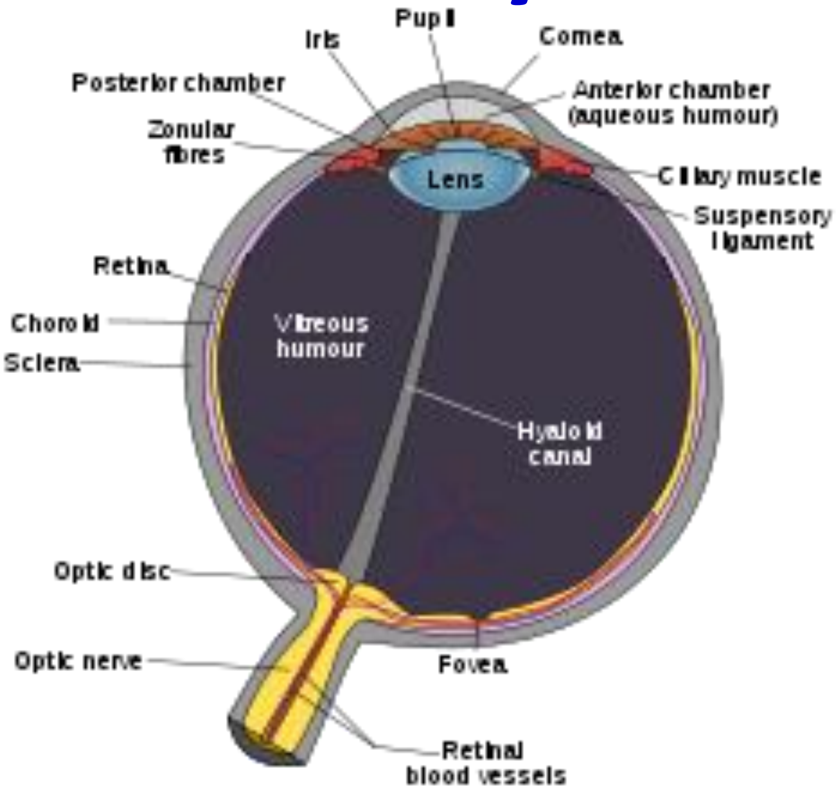
Rods and Cones

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



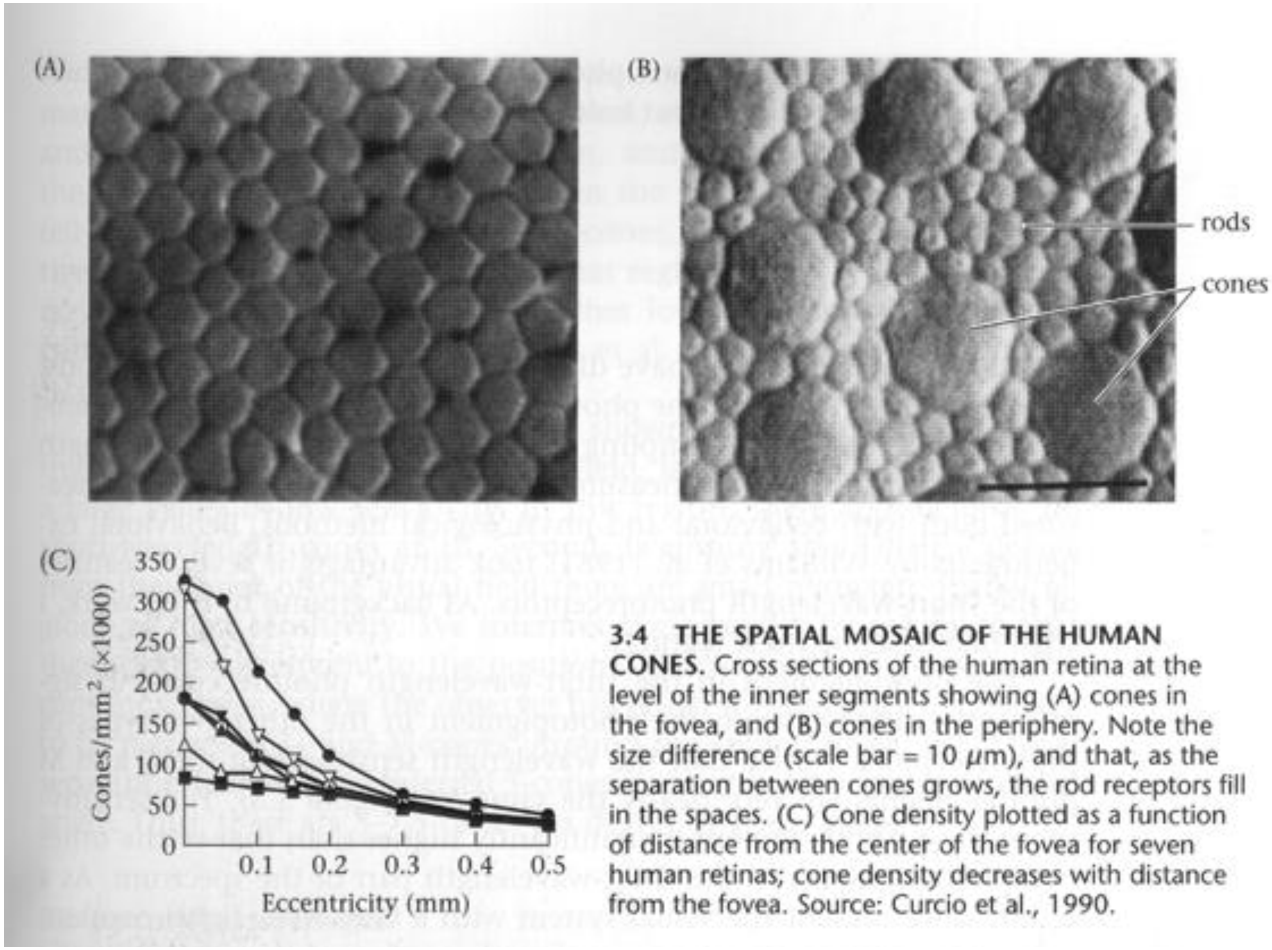
Basic Neuroscience: Anatomy and Physiology Arthur C. Guyton, M.D. 1987 W.B.Saunders Co.

The Eye



- 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, instead they have an elongated slit

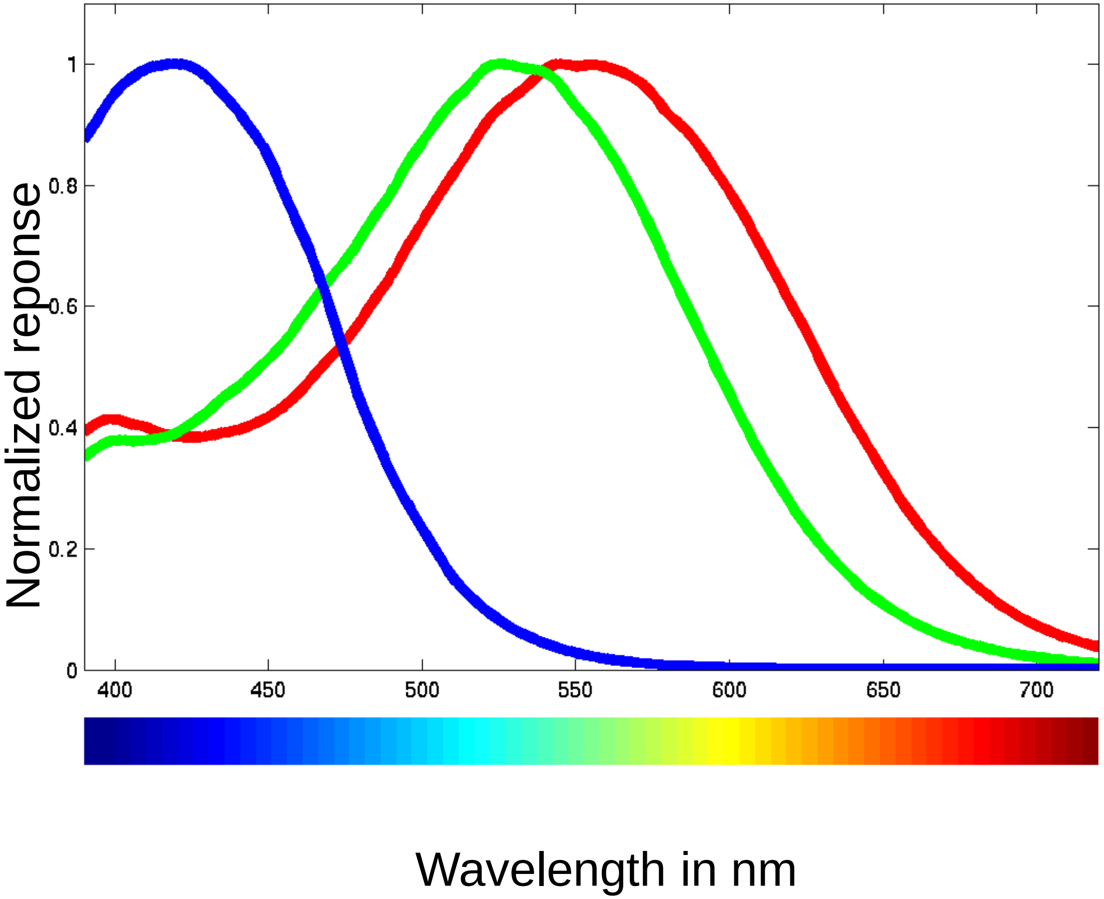
Spatial Arrangement of the Retina



3.4 THE SPATIAL MOSAIC OF THE HUMAN CONES. Cross sections of the human retina at the level of the inner segments showing (A) cones in the fovea, and (B) cones in the periphery. Note the size difference (scale bar = 10 μm), and that, as the separation between cones grows, the rod receptors fill in the spaces. (C) Cone density plotted as a function of distance from the center of the fovea for seven human retinas; cone density decreases with distance from the fovea. Source: Curcio et al., 1990.

(From Foundations of Vision, by Brian Wandell, Sinauer Assoc.)

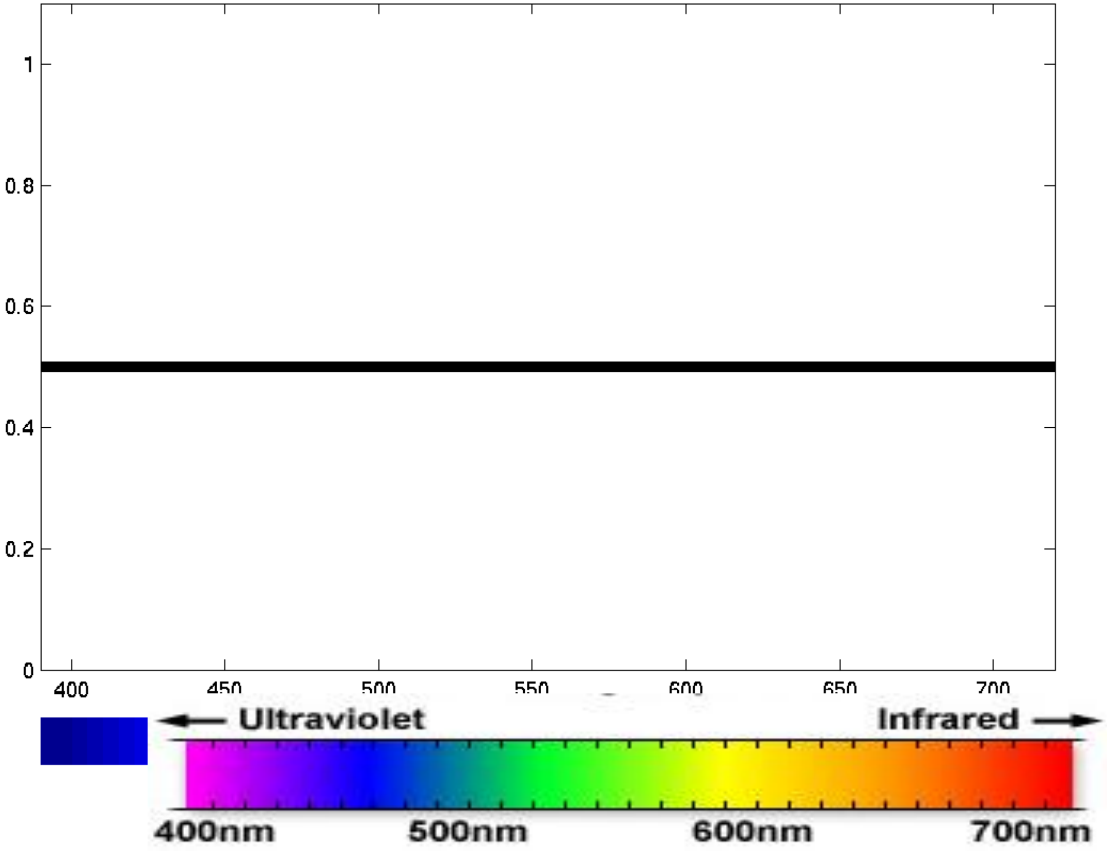
Three Types of Cones (trichromatic vision)



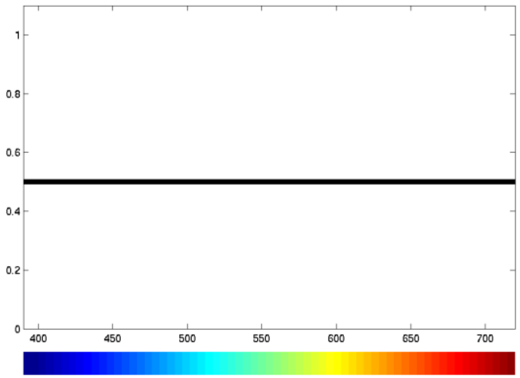
Trichromatic Vision

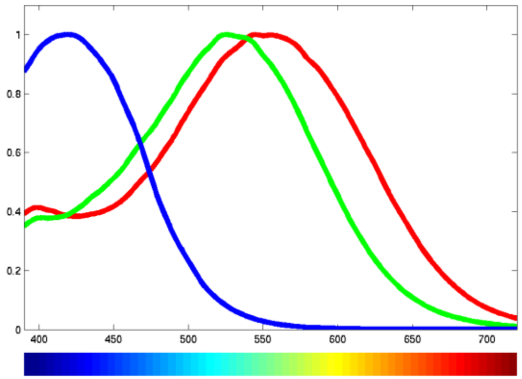
- So-called “blue” light sensors respond to an entire range of frequencies
  - Including in the so-called “green” and “red” regions
- The difference in response of “green” and “red” sensors is small
  - Varies from person to person
    - Each person really sees the world in a different color
  - If the two curves get too close, we have color blindness
    - Ideally traffic lights should be red and blue

White Light


1
0.8
0.6
0.4
0.2
0
400
450
500
550
600
650
700
Ultraviolet
Infrared
400nm
500nm
600nm
700nm

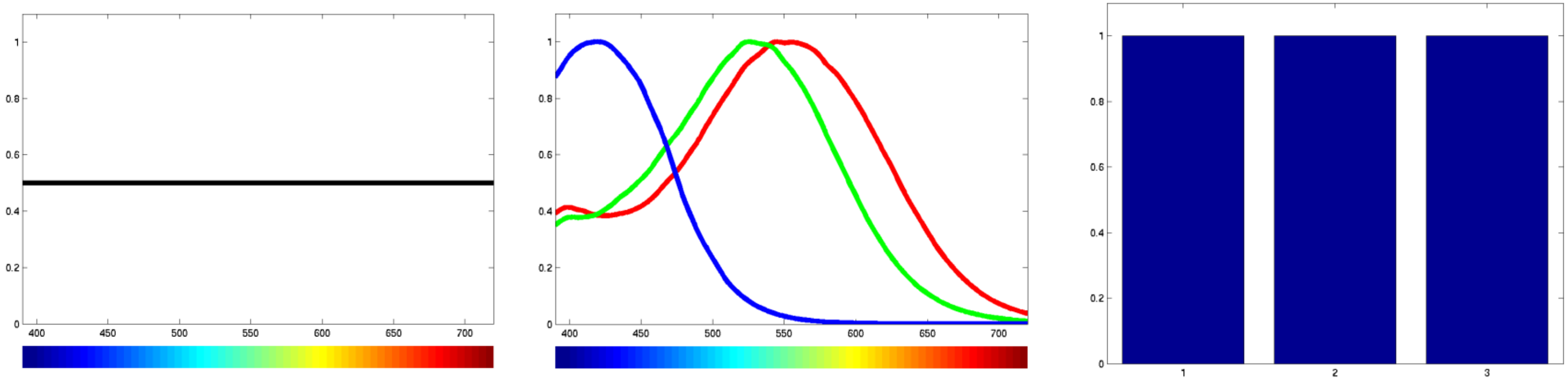
Response to White Light



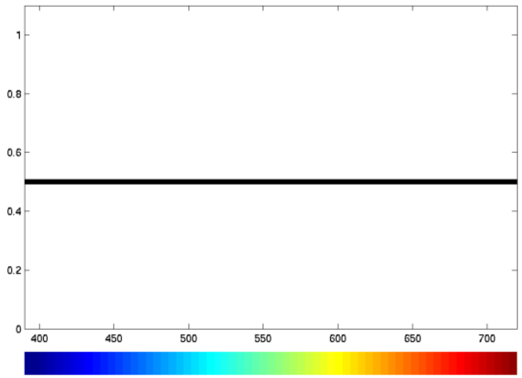


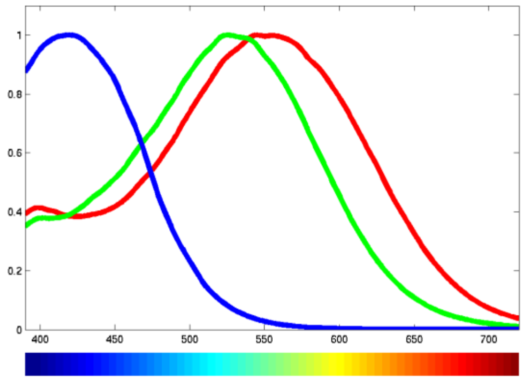
?

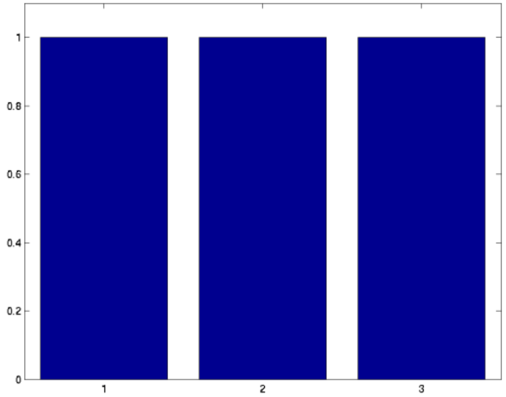
Response to White Light

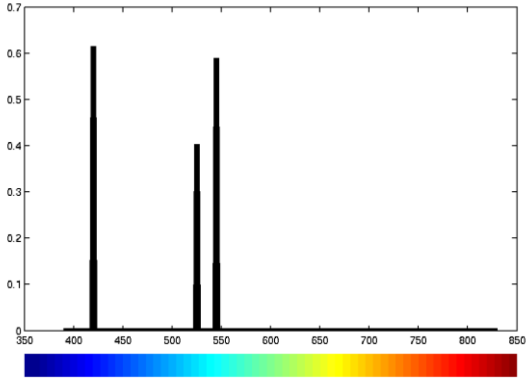


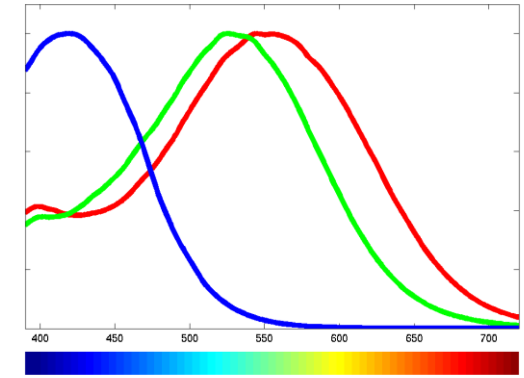
Response to Sparse Light





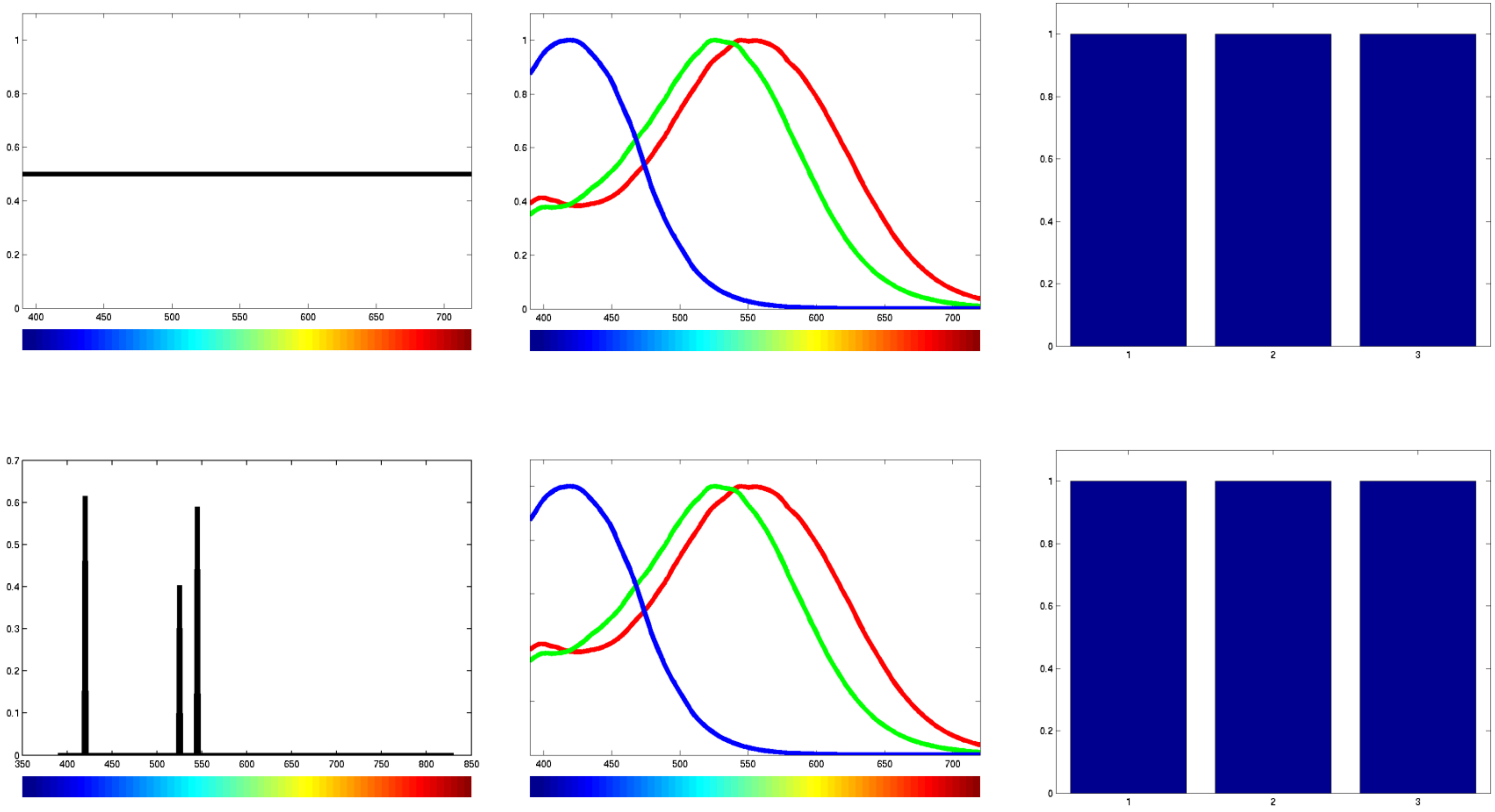




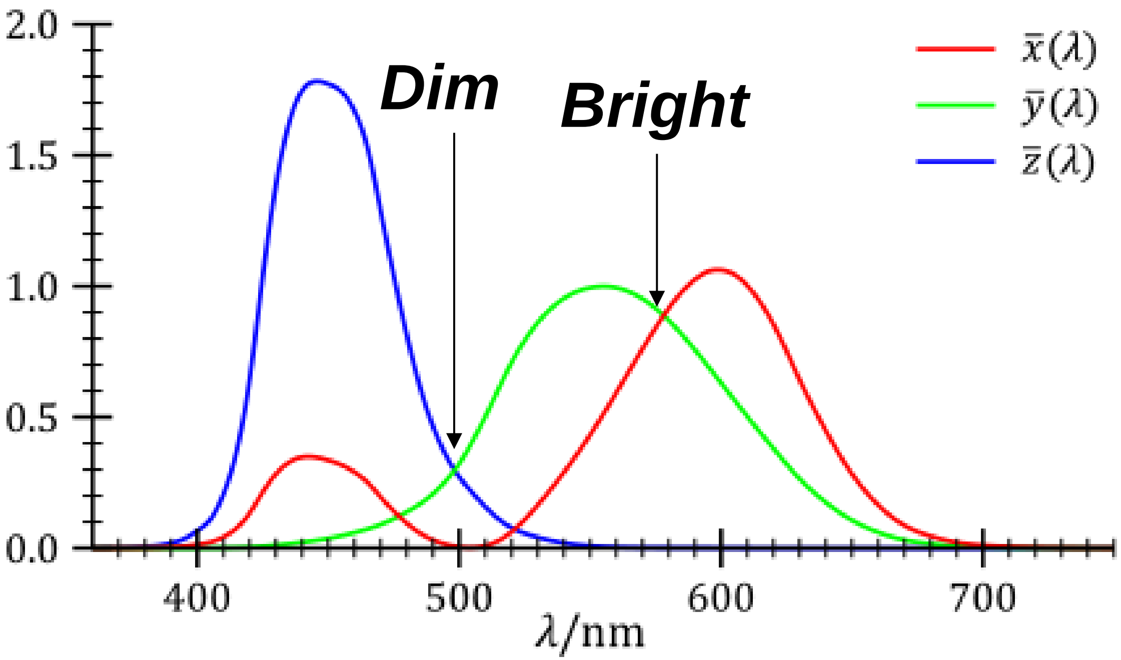


?

Response to Sparse Light

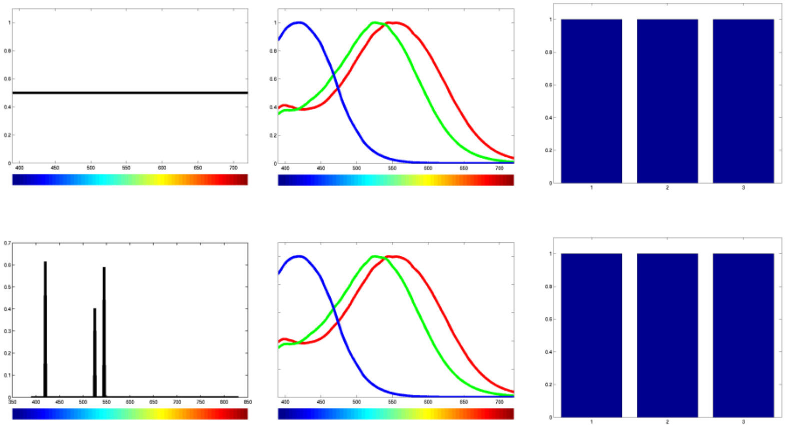


Human perception anomalies



- The same intensity of monochromatic light will result in different *perceived* brightness at different wavelengths
- Many combinations of wavelengths can produce the same sensation of colour.
- Yet humans can distinguish 10 *million* colours

Representing Images



- Utilize trichromatic nature of human vision
  - Sufficient to trigger each of the three cone types in a manner that produces the sensation of the desired color
    - A *tetrachromatic* animal would be very confused by our computer images
      - Some new-world monkeys are tetrachromatic
- The three “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 colours
    - Which is still a small fraction of what we can actually see
  - How many colours? …

The “CIE” colour space

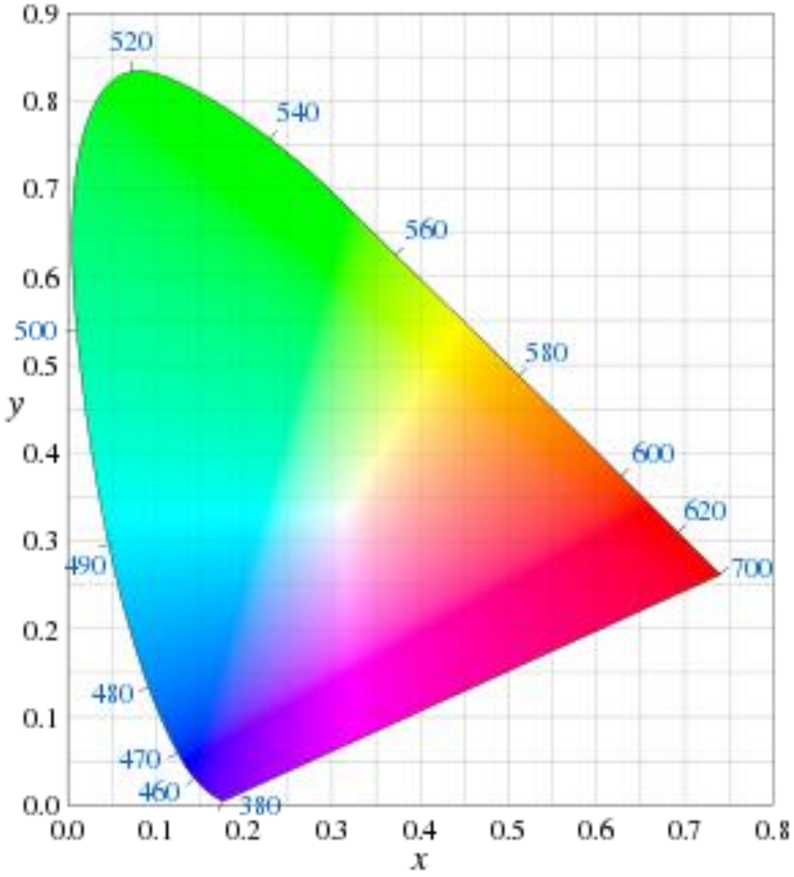
- From experiments done in the 1920s by W. David Wright and John Guild
  - Subjects adjusted x,y,and z on the right of a circular screen to match a colour on the left

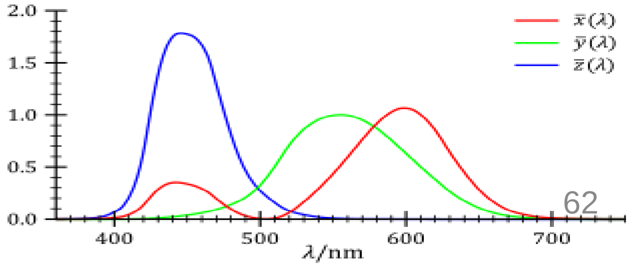
- X, Y and Z are normalized responses of the three sensors
  - X + Y + Z is 1.0
    - Normalized to have to total net intensity

- The image represents all colours we can see
  - The outer curve represents monochromatic light
    - X,Y and Z as a function of λ
  - The lower line is the line of purples
    - End of visual spectrum

- The CIE chart was updated in 1960 and 1976
  - The newer charts are less popular

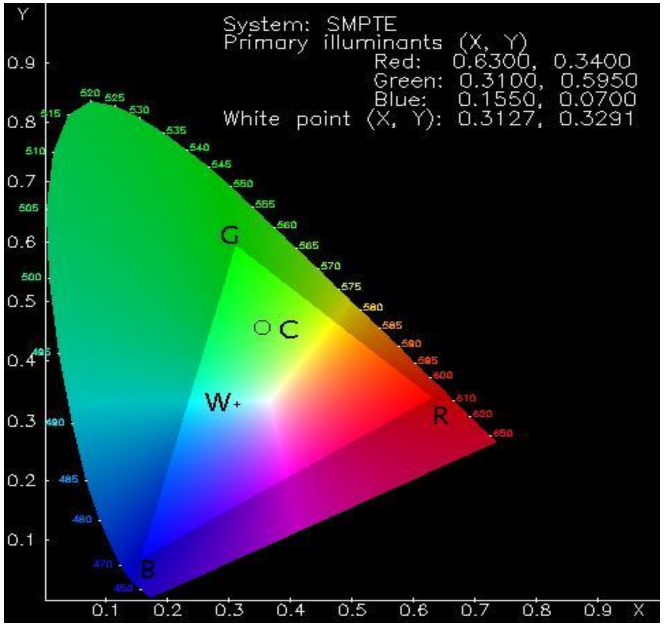
International council on illumination, 1931





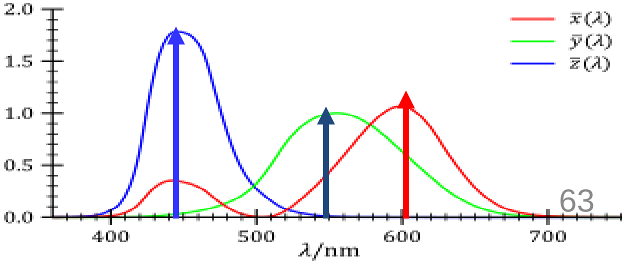
What is displayed

- The RGB triangle
  - Colours outside this area cannot be matched by additively combining only 3 colours
    - Any other set of monochromatic colours would have a differently restricted area
    - TV images can never be like the real world



- Each corner represents the (X,Y,Z) coordinate of one of the three "primary" colours used in images

- In reality, this represents a very tiny fraction of our visual acuity
  - Also affected by the quantization of levels of the colours



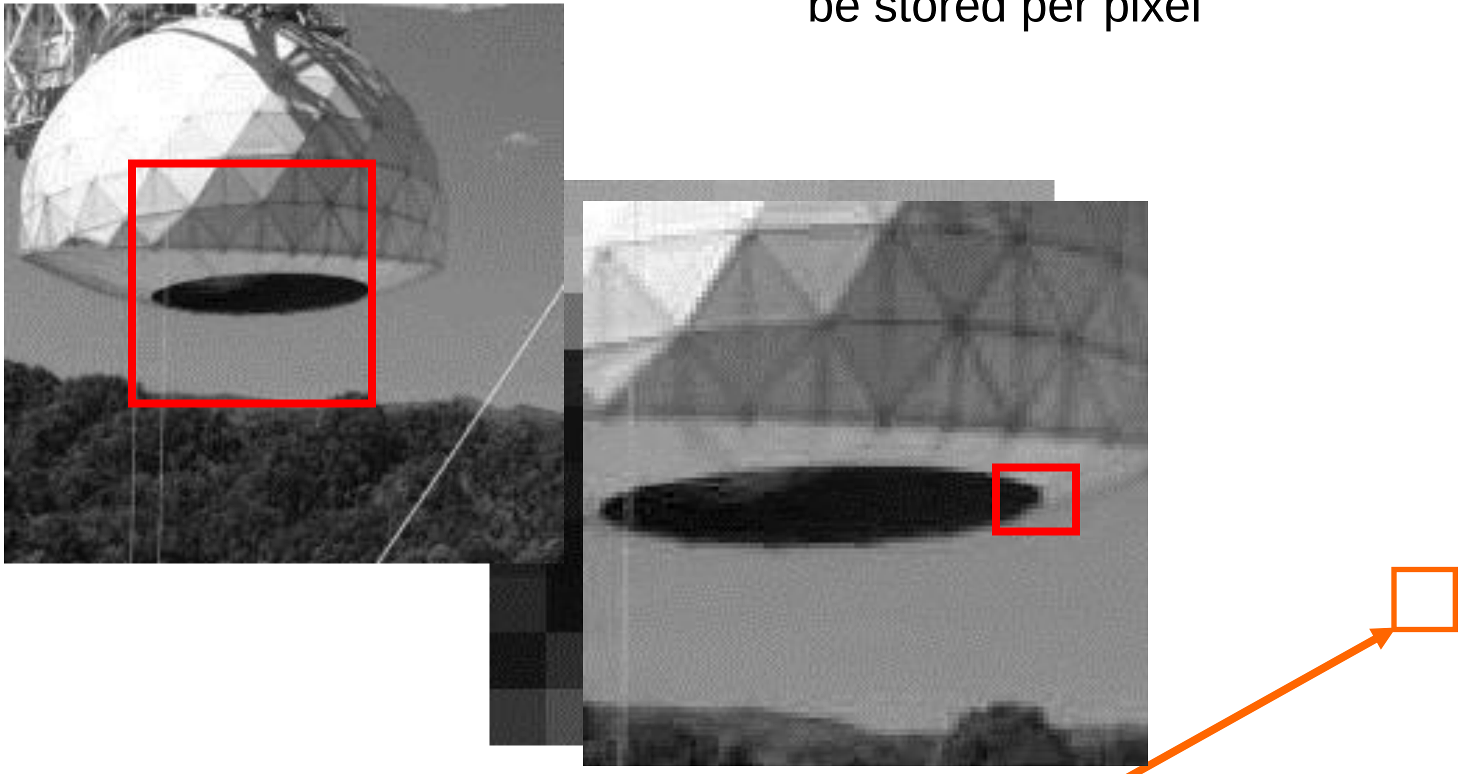
Representing Images on Computers

- Greyscale: a single matrix of numbers
  - Each number represents the intensity of the image at a specific location in the image
  - Implicitly, R = G = B at all locations

- Color: 3 matrices of numbers
  - The matrices represent different things in different representations
  - RGB Colorspace: Matrices represent intensity of Red, Green and Blue
  - CMYK Colorspace: Cyan, Magenta, Yellow
  - YIQ Colorspace..
  - HSV Colorspace..

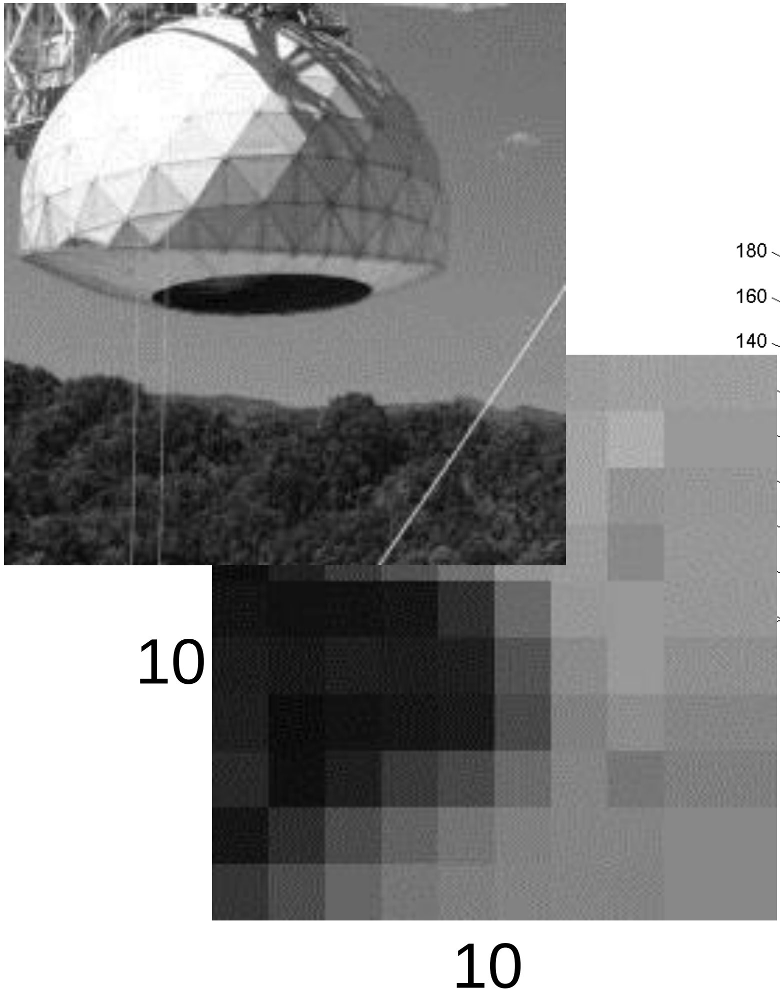
Computer Images: Grey Scale

R = G = B. Only a single number need be stored per pixel



Picture Element (PIXEL)
Position & gray value (scalar)

What we see



What the computer “sees”

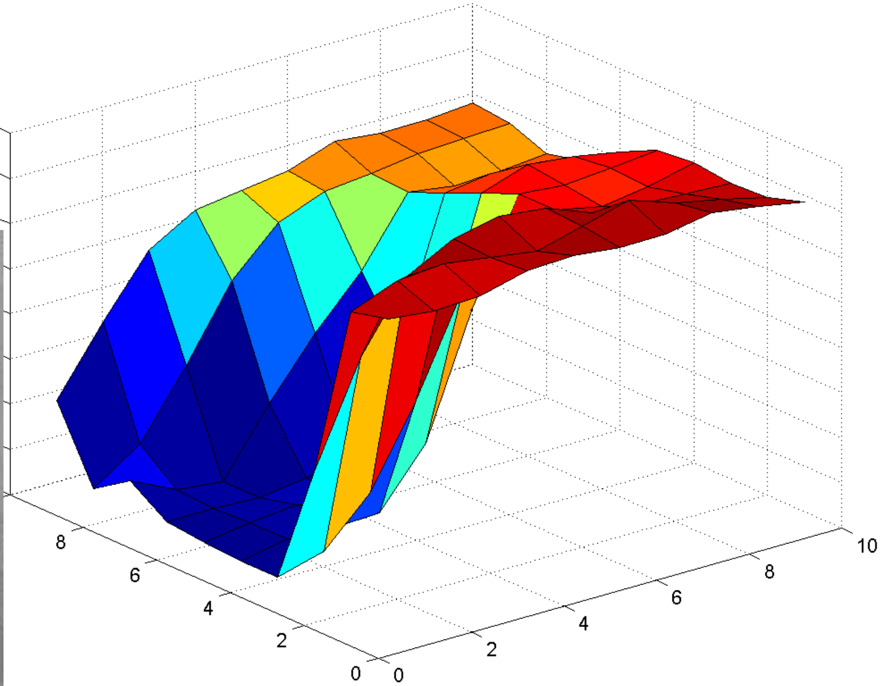
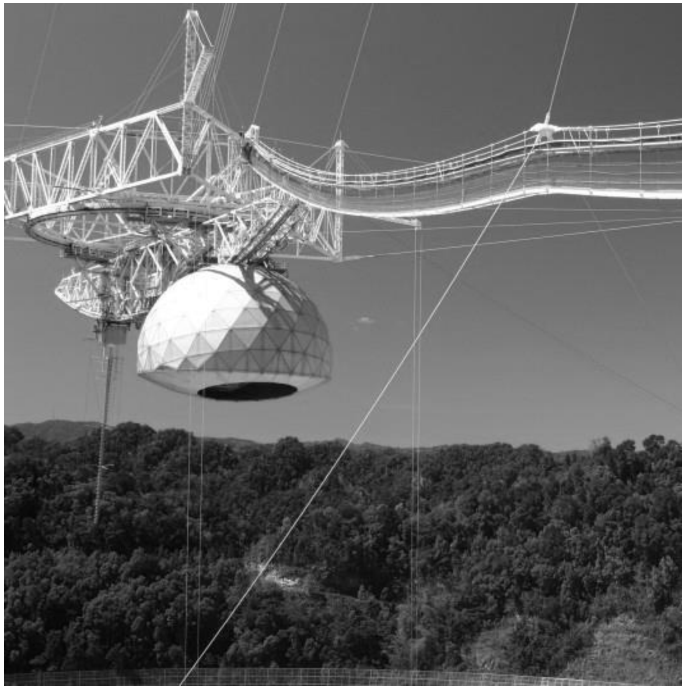
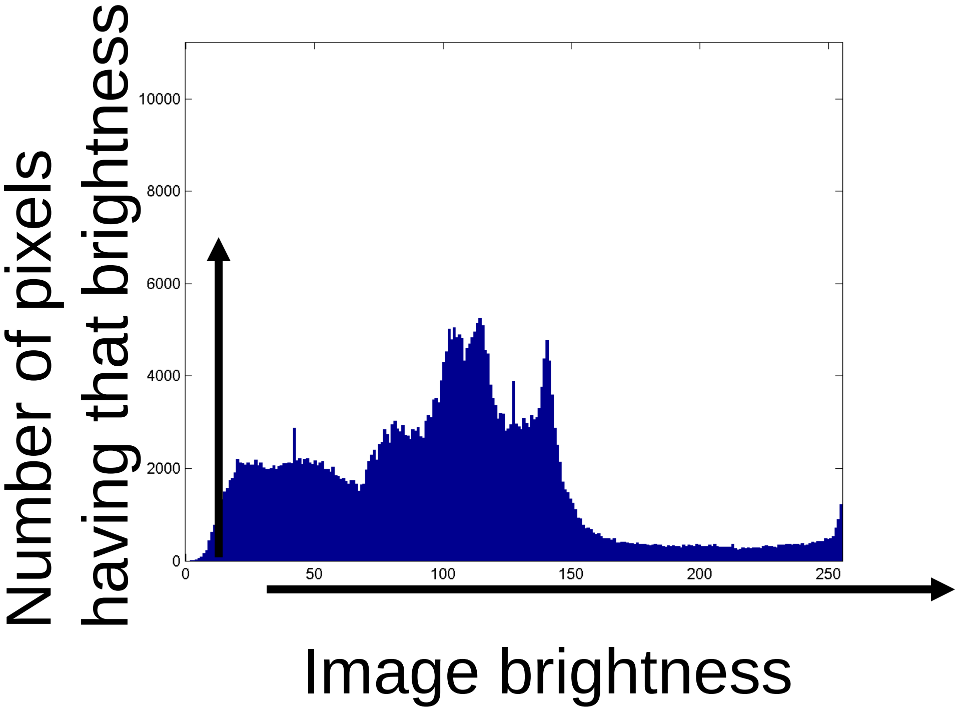


Image Histograms





Example histograms

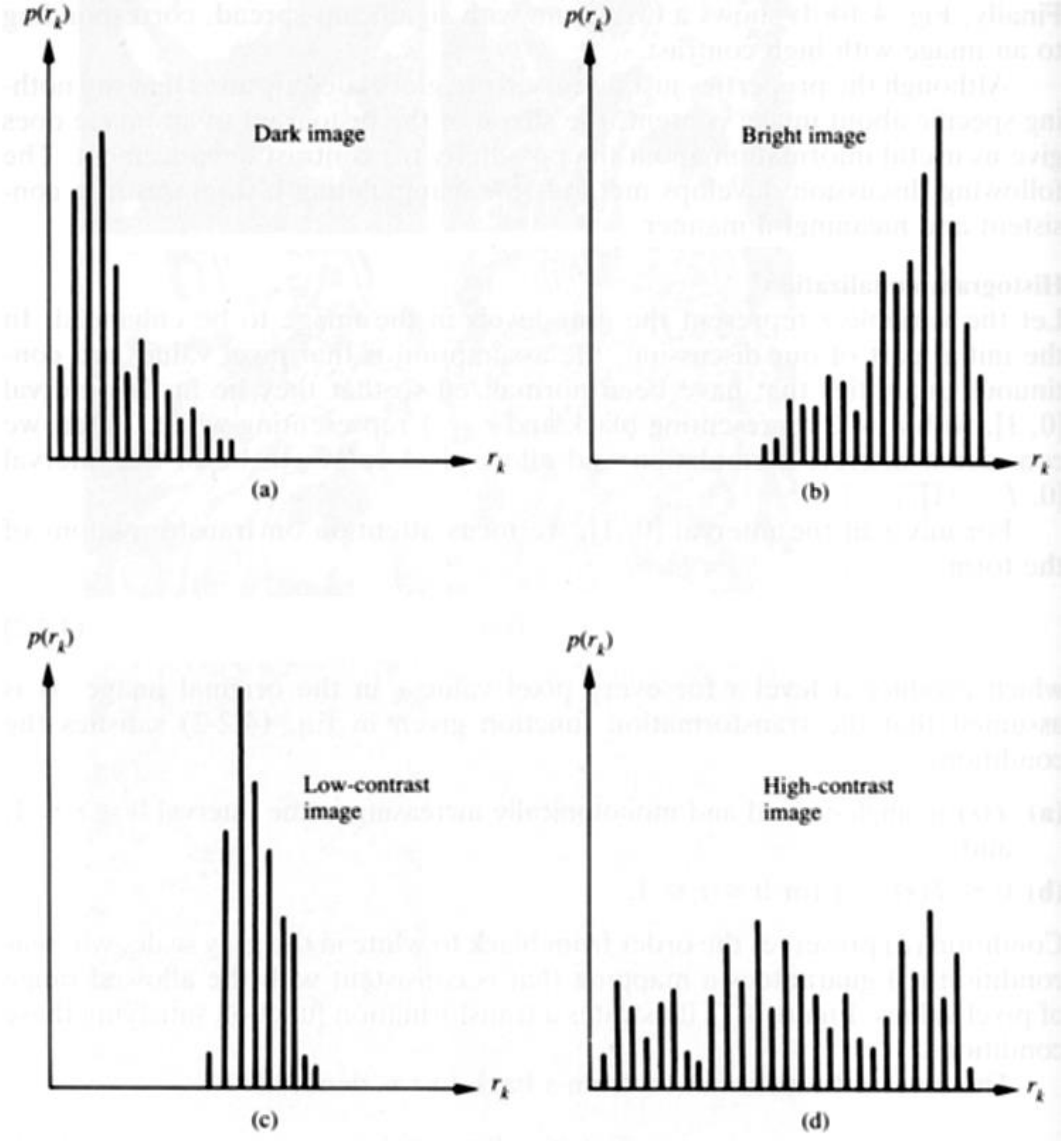
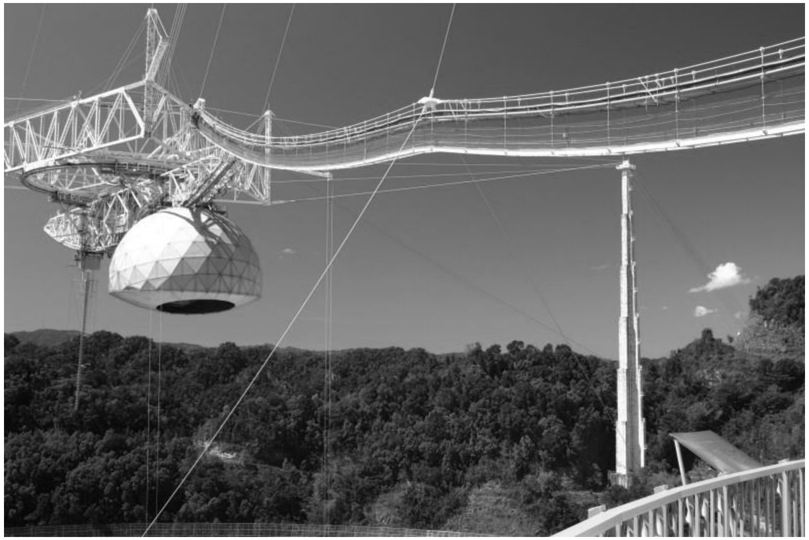


Figure 4.10 *Histograms corresponding to four basic image types.*

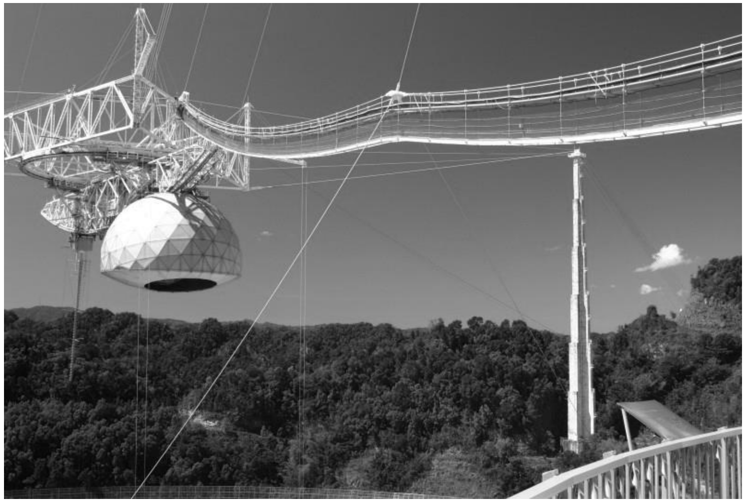
From: Digital Image Processing,
by Gonzales and Woods, Addison Wesley, 1992

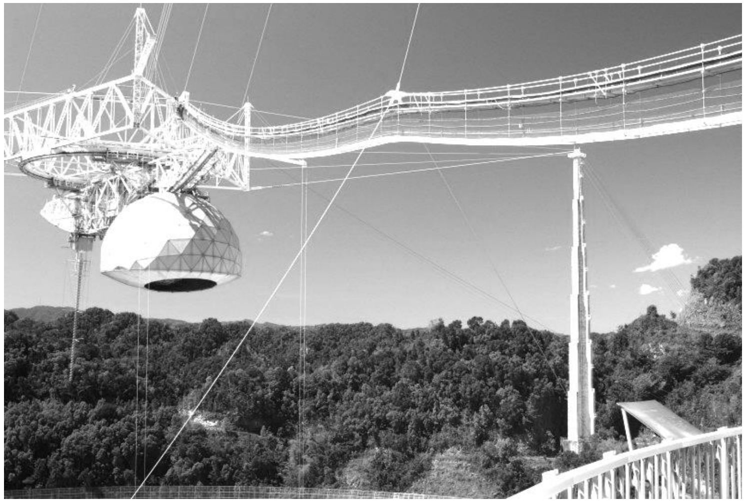
Pixel operations

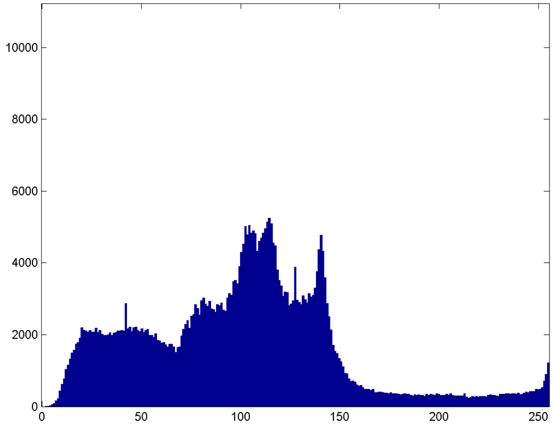
- New value is a function of the old value
  - Tonescale to change image brightness
  - Threshold to reduce the information in an image
  - Colorspace operations

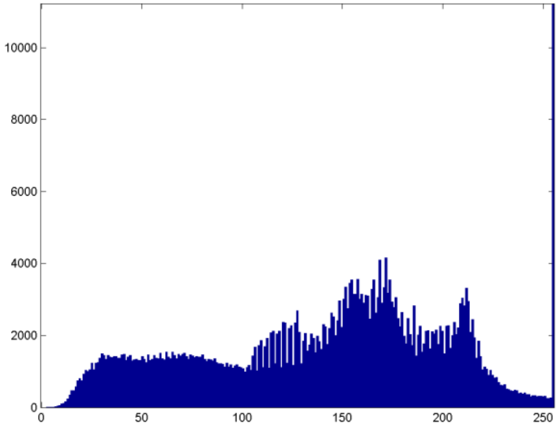


J=1.5*I

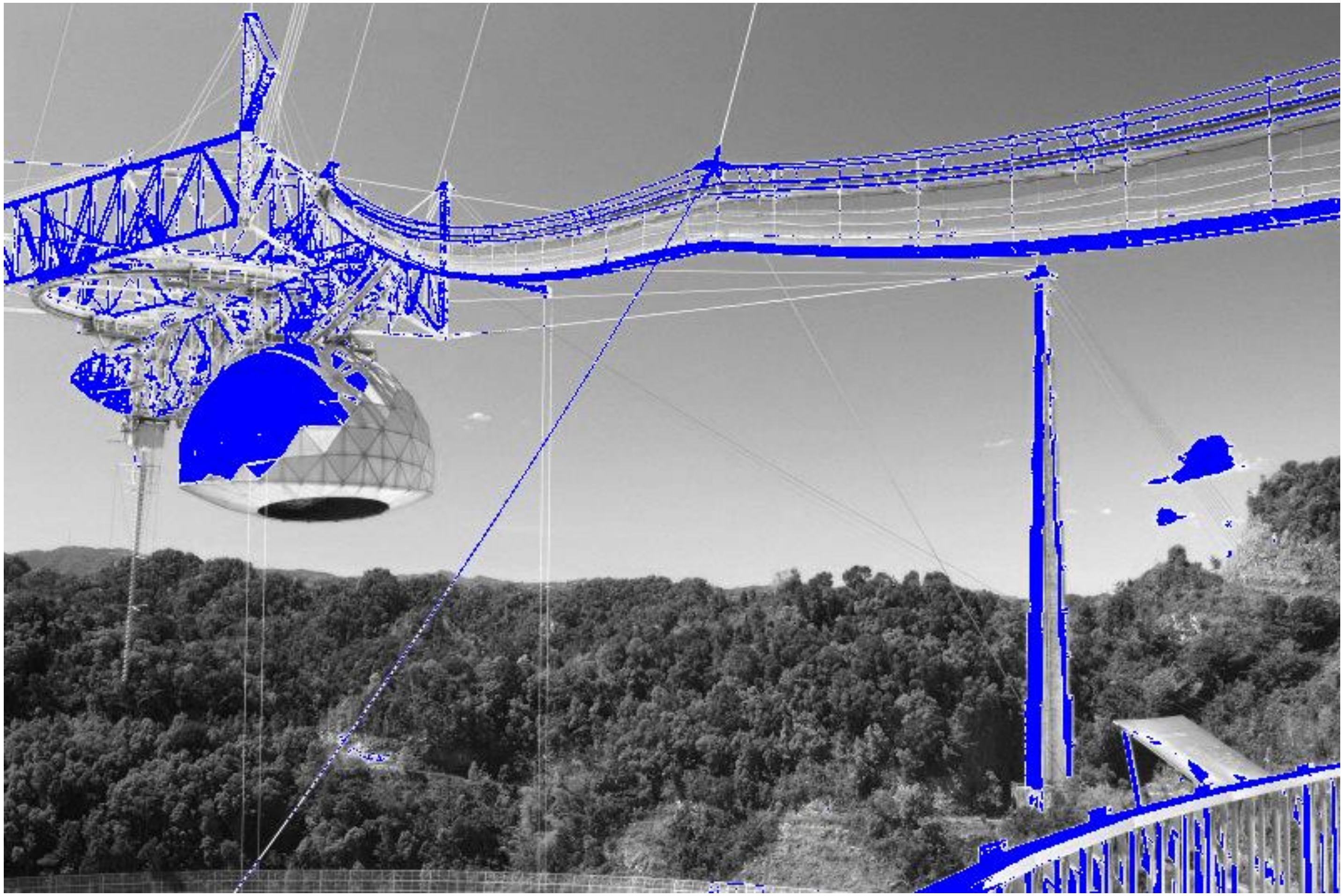




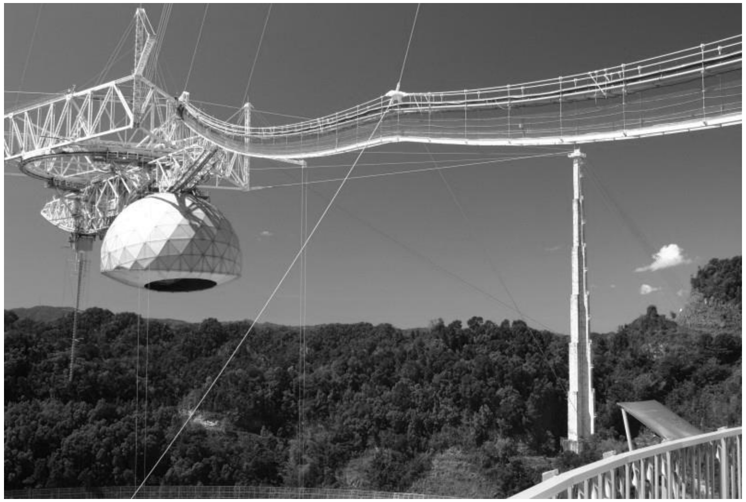

10000
8000
6000
4000
2000
0
0
50
100
150
200
250

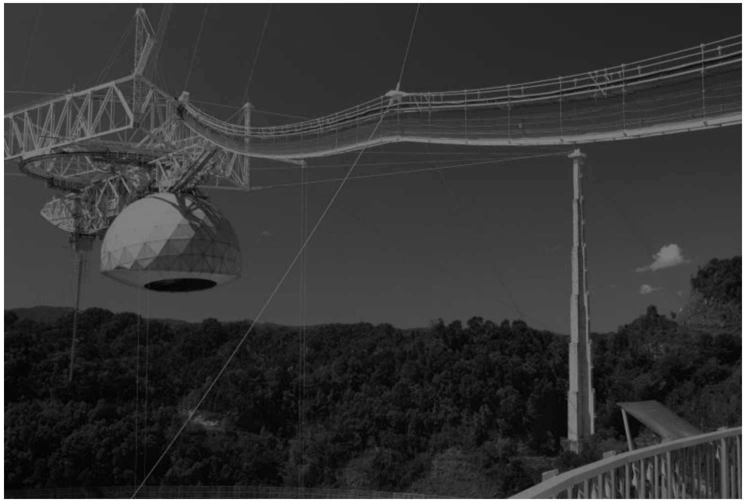

10000
8000
6000
4000
2000
0
0
50
100
150
200
250

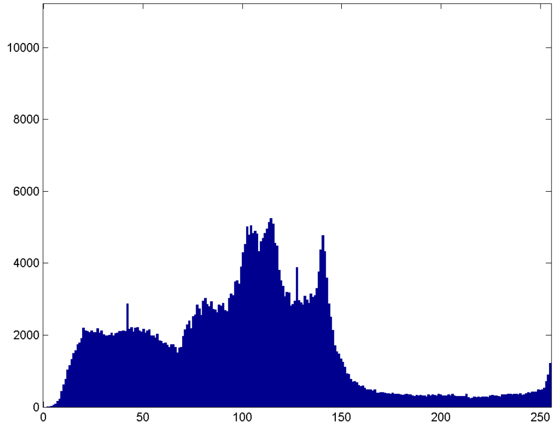
Saturation

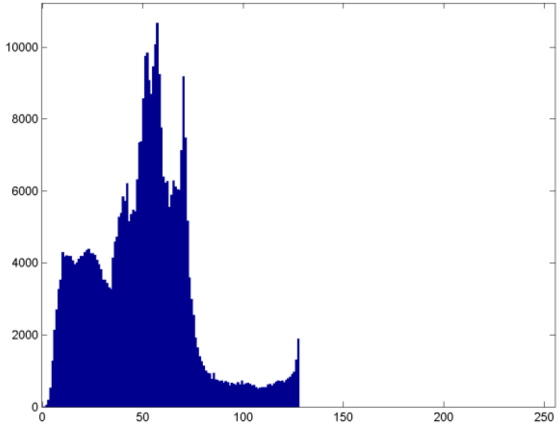


J=0.5*I

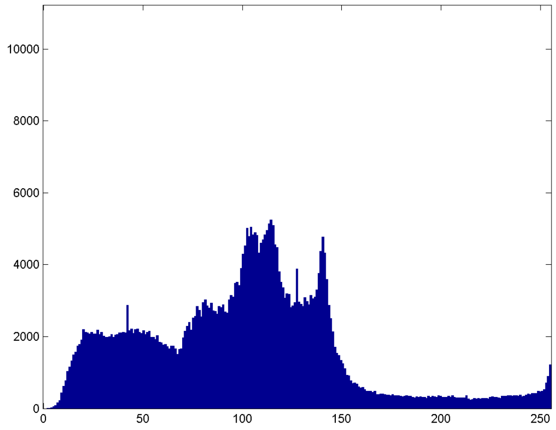


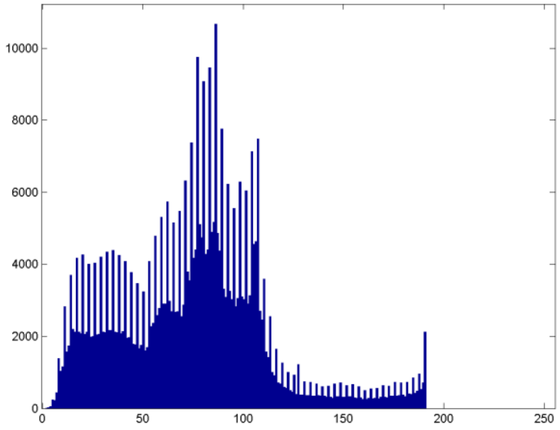



10000
8000
6000
4000
2000
0
0
50
100
150
200
250

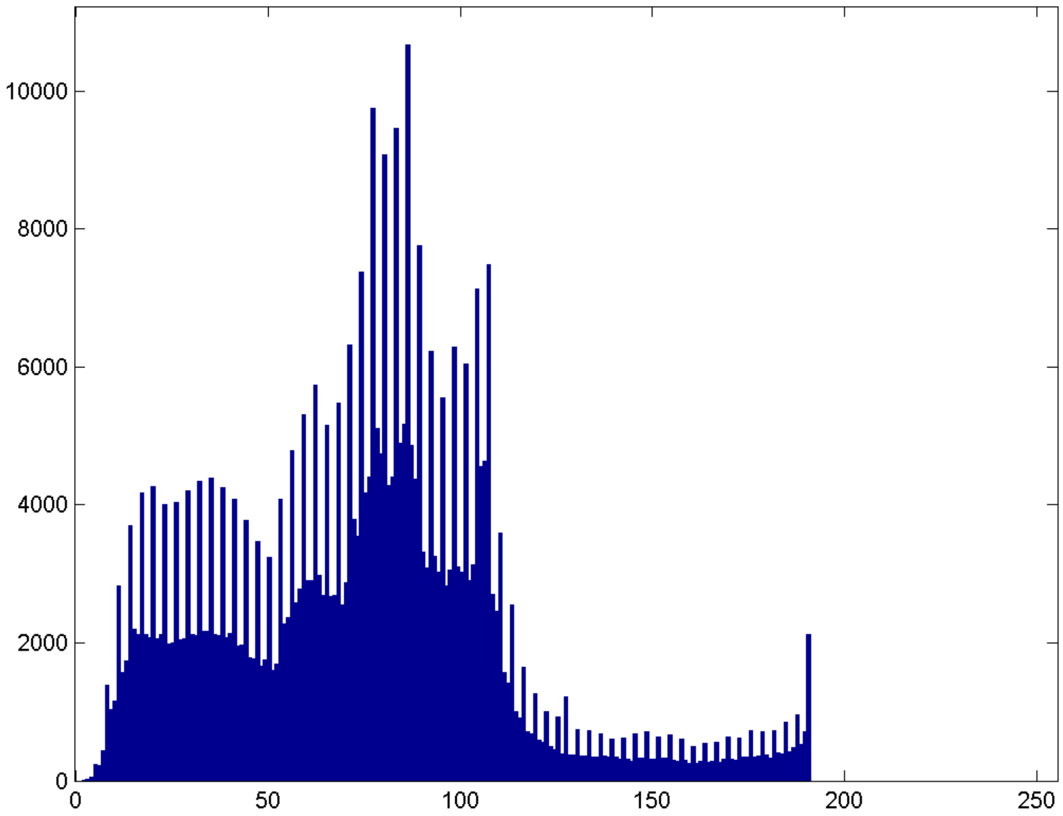

10000
8000
6000
4000
2000
0
0
50
100
150
200
250

J=uint8(0.75*I)

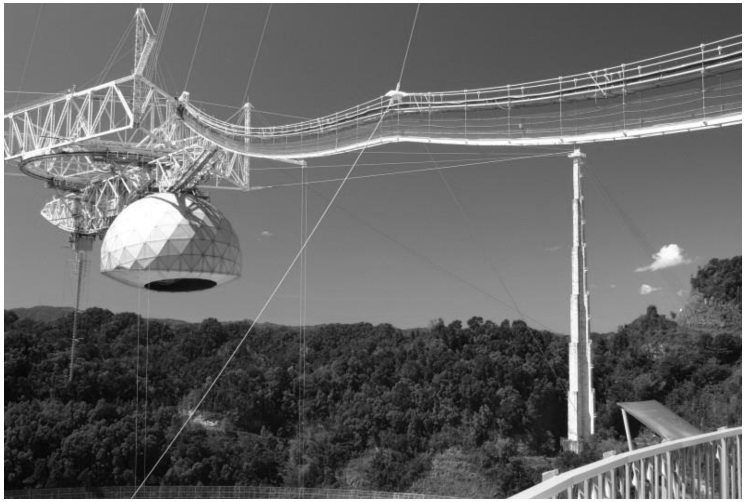

10000
8000
6000
4000
2000
0
0
50
100
150
200
250

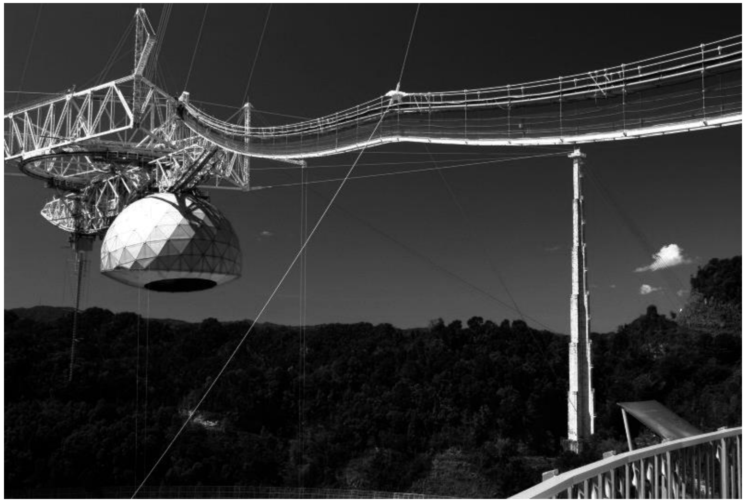

10000
8000
6000
4000
2000
0
0
50
100
150
200
250

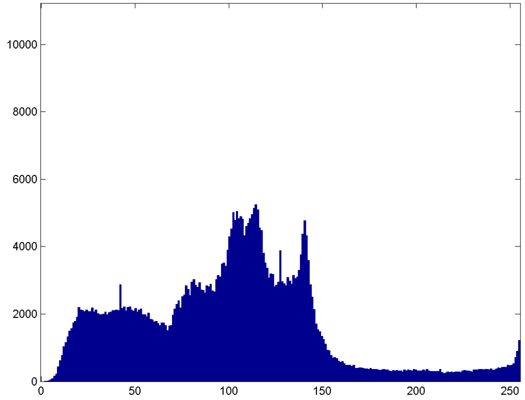
What’s this?

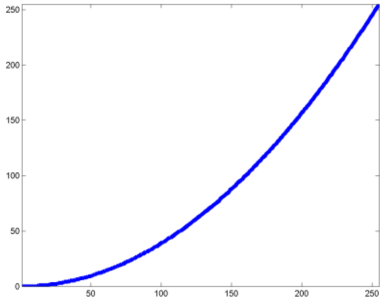


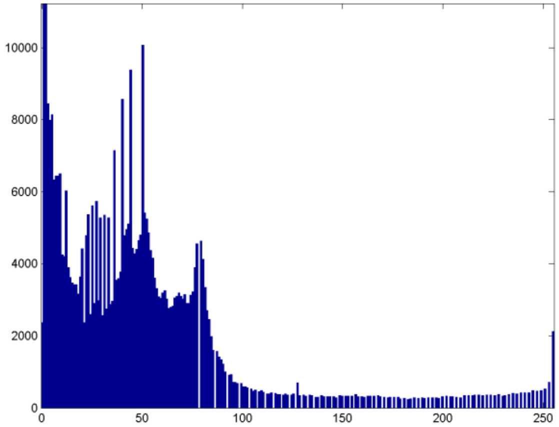
Non-Linear Darken



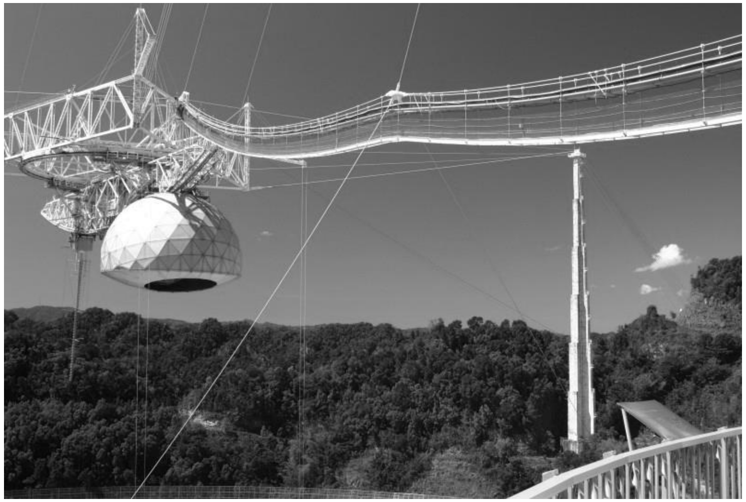


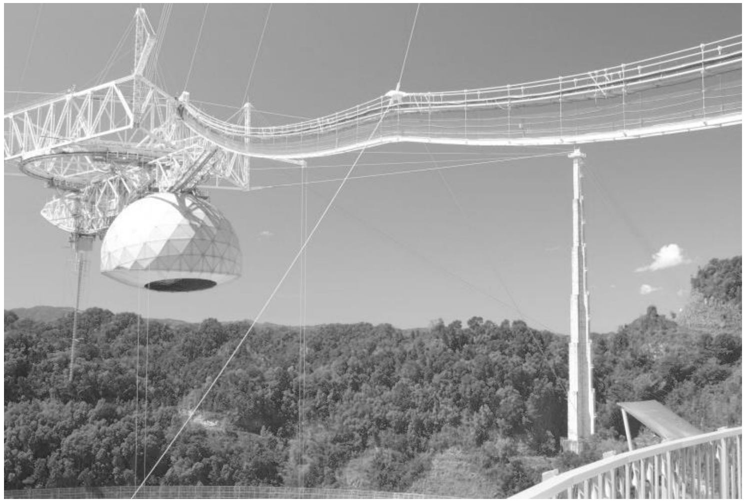


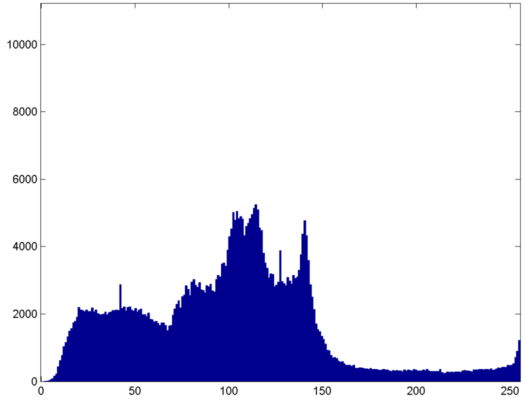


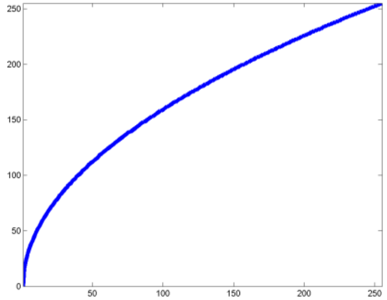


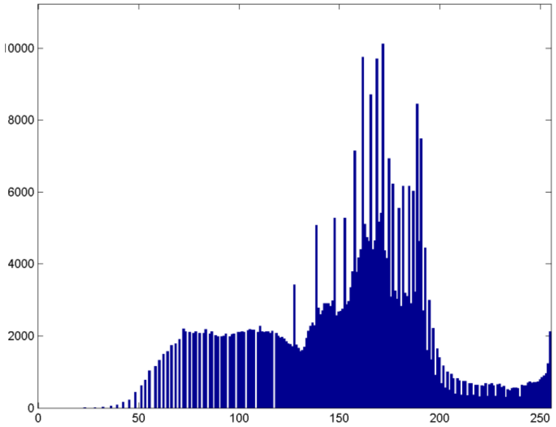
Non-Linear Lighten



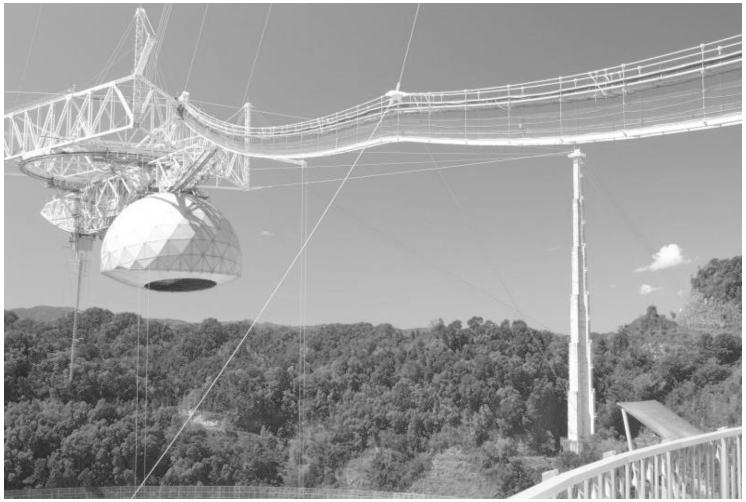


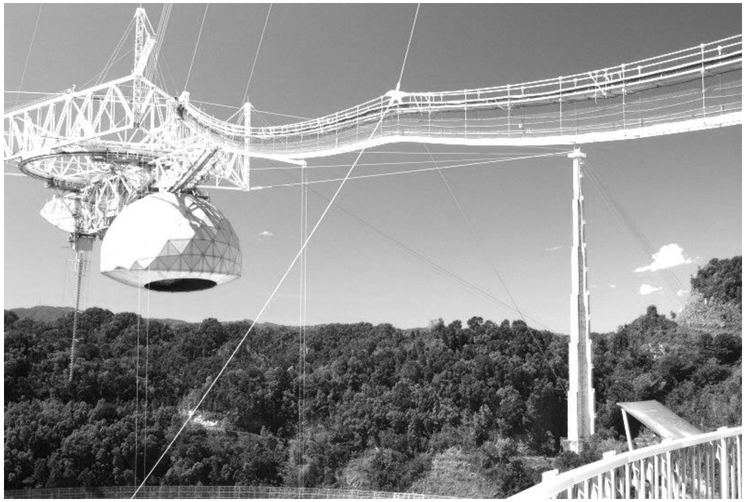


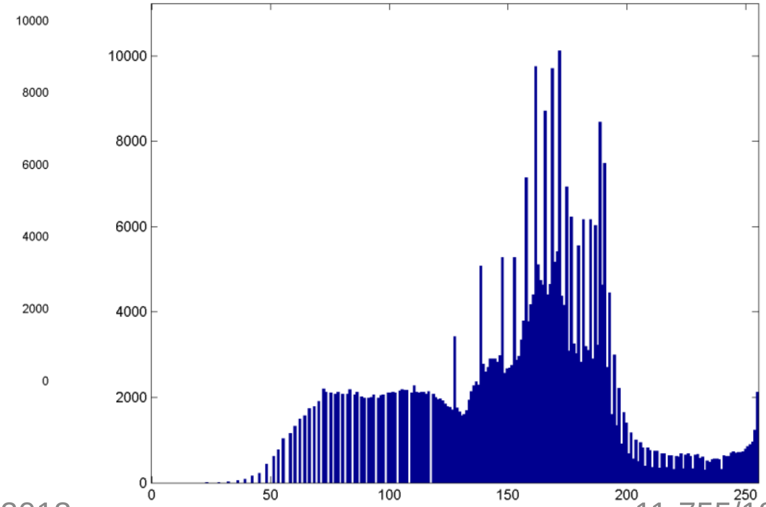




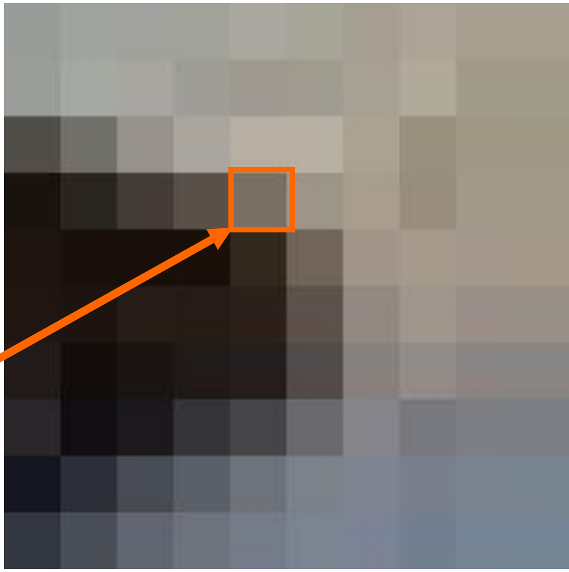
Linear vs. Non-Linear







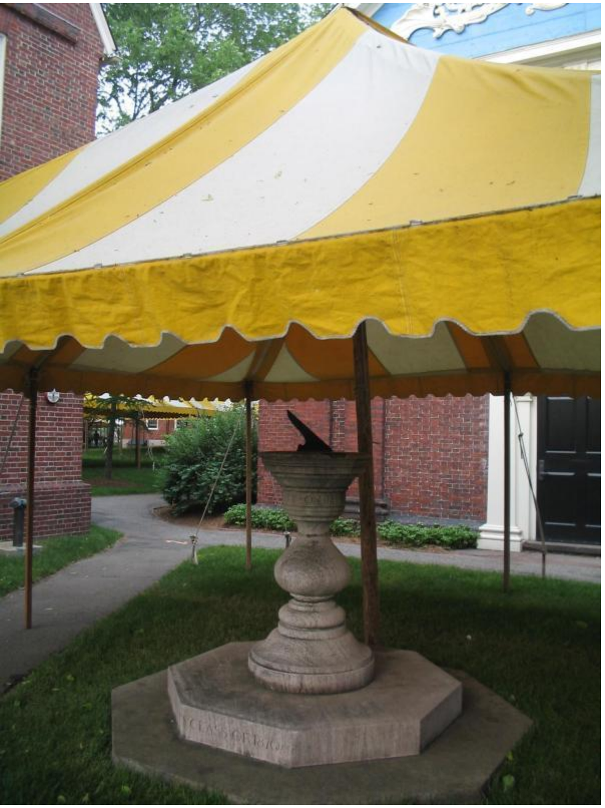
Color Images



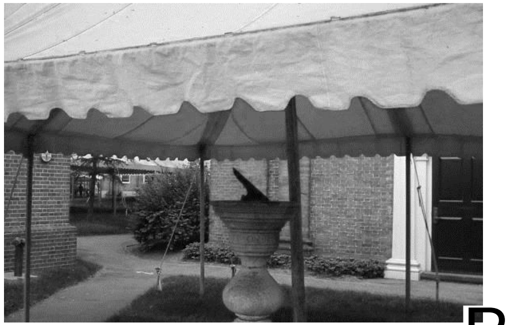
Picture Element (PIXEL)

Position & color value (red, green, blue)

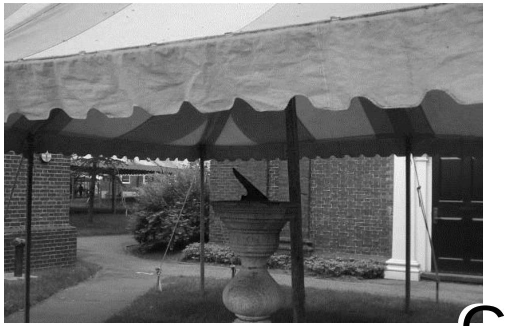
RGB Representation



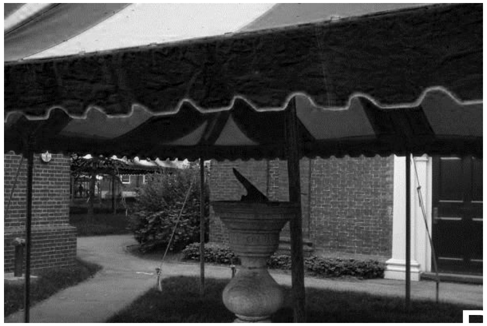
original



R

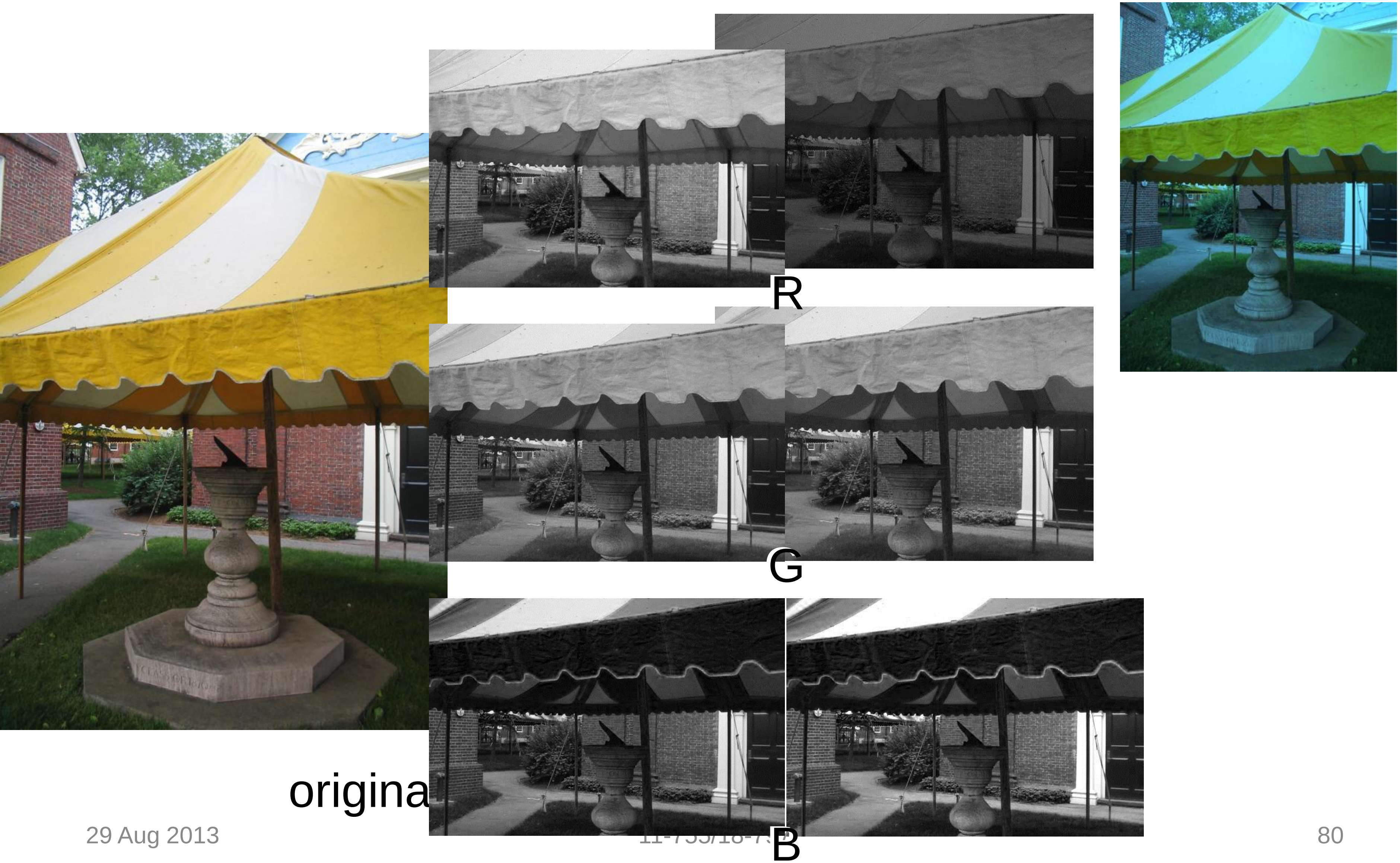


G

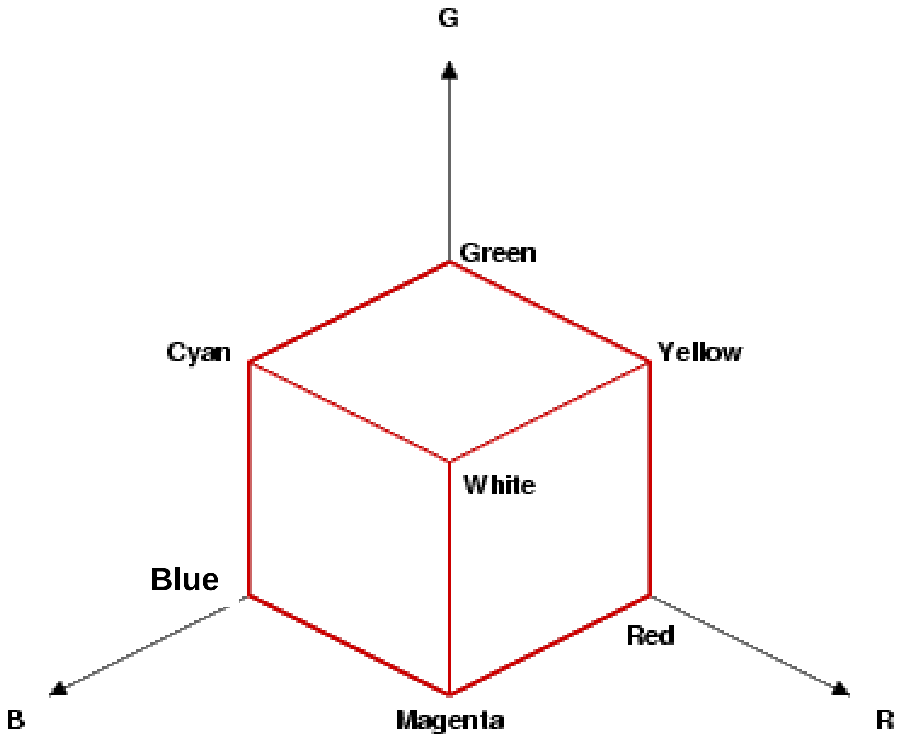


B

RGB Manipulation Example: Color Balance

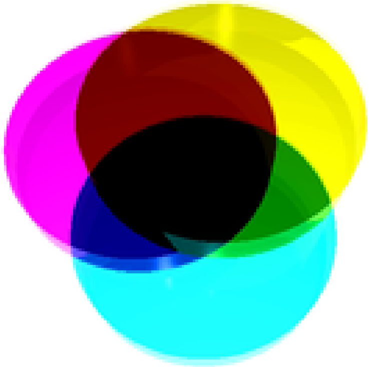


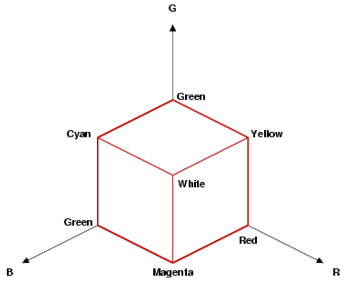
The CMYK color space



- Represent colors in terms of cyan, magenta, and yellow
  - The “K” stands for “Key”, not “black”

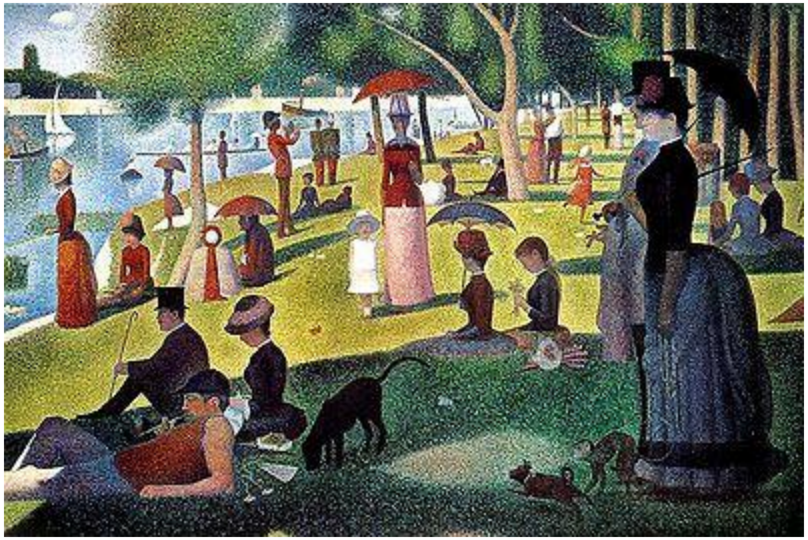
CMYK is a *subtractive* representation





- RGB is based on *composition*, i.e. it is an additive representation
  - Adding equal parts of red, green and blue creates white
- What happens when you mix red, green and blue paint?
  - Clue – paint colouring is subtractive..
- CMYK is based on *masking,* i.e. it is subtractive
  - The base is white
  - Masking it with equal parts of C, M and Y creates Black
  - Masking it with C and Y creates Green
    - Yellow masks blue
  - Masking it with M and Y creates Red
    - Magenta masks green
  - Masking it with M and C creates Blue
    - Cyan masks green
  - Designed specifically for *printing*
    - As opposed to rendering

An Interesting Aside



- Paints create subtractive coloring
  - Each paint masks out some colours
  - Mixing paint subtracts combinations of colors
  - Paintings represent subtractive colour masks

- In the 1880s Georges-Pierre Seurat pioneered an *additive-colour* technique for painting based on “pointilism”
  - How do you think he did it?

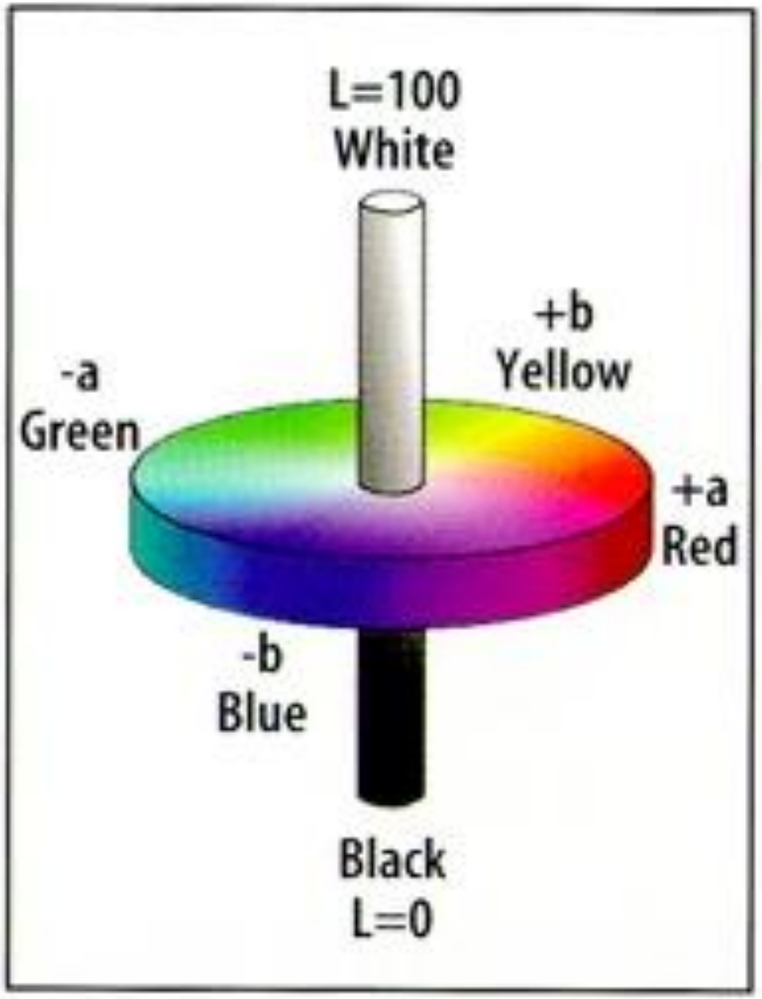
NTSC color components

Y = “luminance”

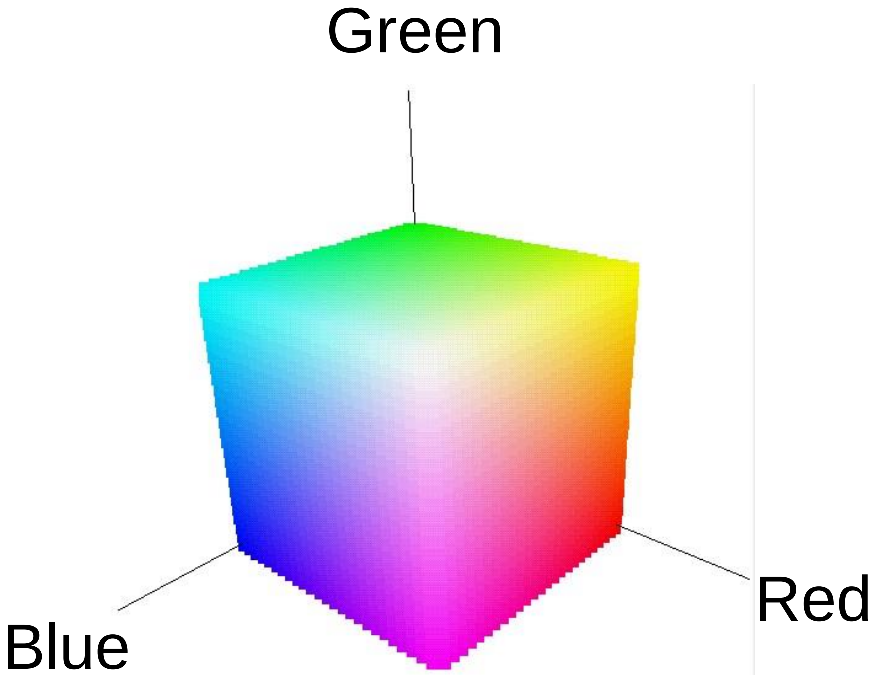
I = “red-green”

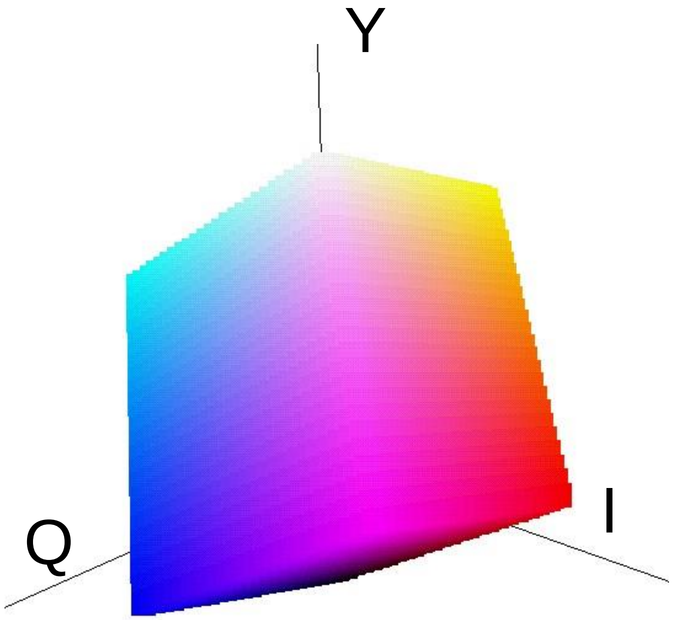
Q = “blue-yellow”

a.k.a. YUV although YUV is actually the color specification for PAL video



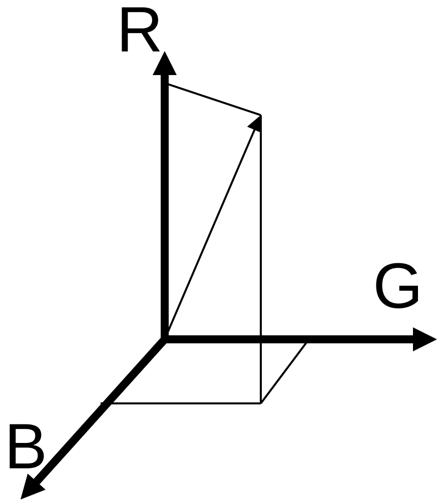
YIQ Color Space

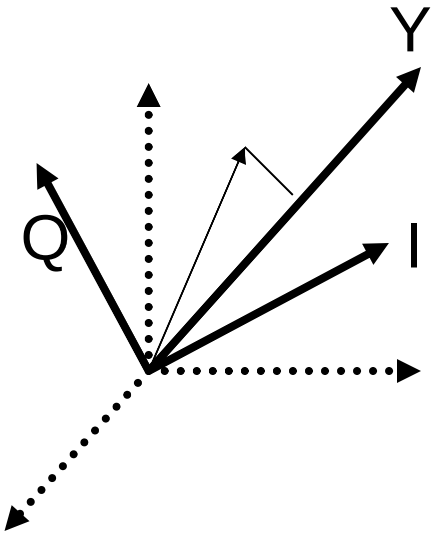




$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} .299 & .587 & .114 \\ .596 & -.275 & -.321 \\ .212 & -.523 & .311 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

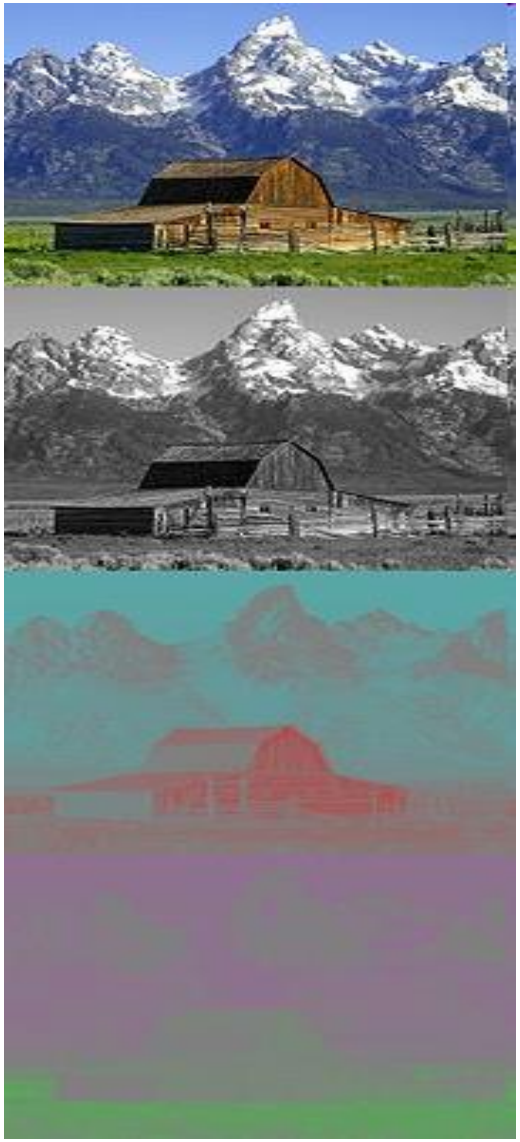
Color Representations





- Y value lies in the same range as R,G,B ([0,1])
- I is to [-0.59 0.59]
- Q is limited to [-0.52 0.52]
- Takes advantage of lower human sensitivity to I and Q axes

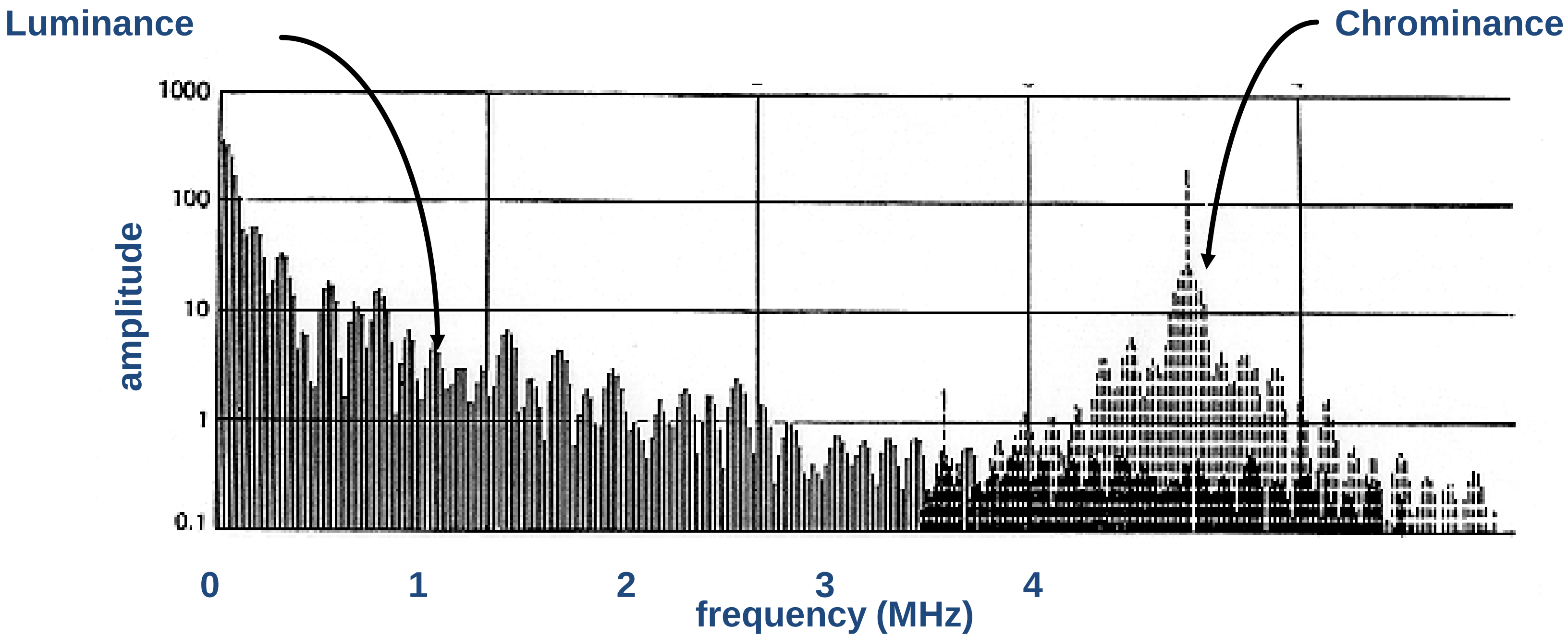
YIQ



- Top: Original image
- Second: Y
- Third: I (displayed as red-cyan)
- Fourth: Q (displayed as green-magenta)
  - From http://wikipedia.org/

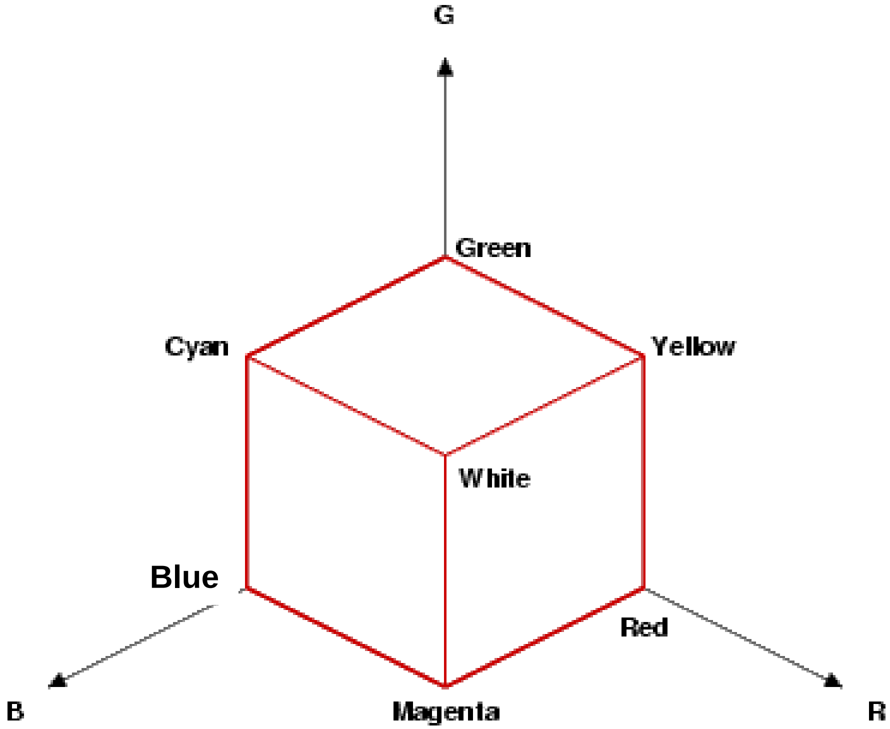
- Processing (e.g. histogram equalization) only needed on Y
  - In RGB must be done on all three colors. Can distort image colors
  - A black and white TV only needs Y

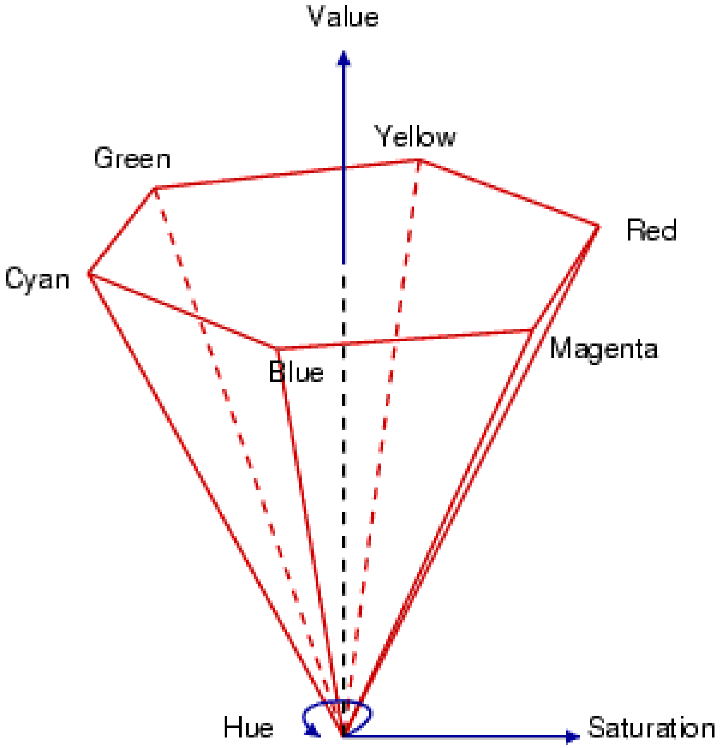
Bandwidth (transmission resources) for the components of the television signal



Understanding image perception allowed NTSC to add color to the black and white television signal. The eye is more sensitive to I than Q, so lesser bandwidth is needed for Q. Both together used much less than Y, allowing for color to be added for minimal increase in transmission bandwidth.

Hue, Saturation, Value



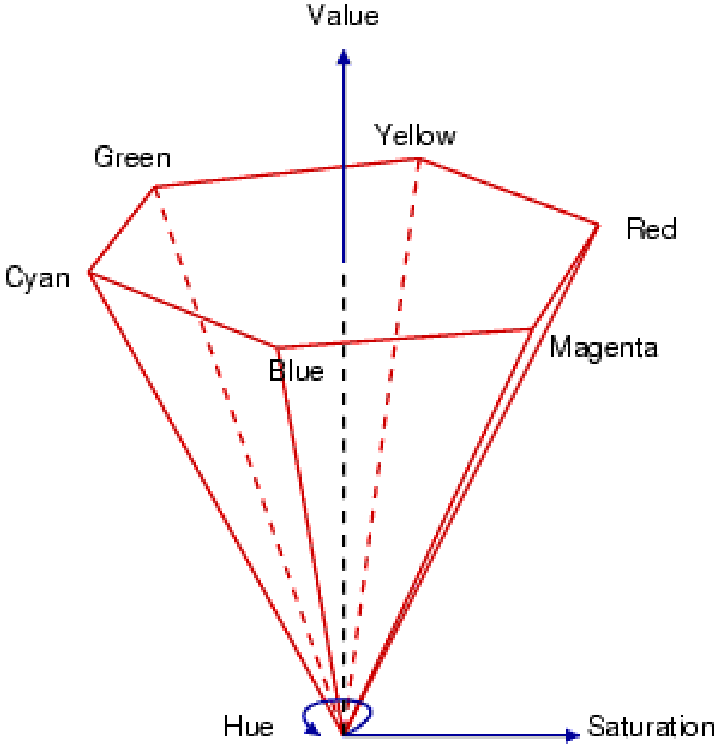


V = [0,1], S = [0,1]
H = [0,360]

The HSV Colour Model By Mark Roberts
http://www.cs.bham.ac.uk/~mer/colour/hsv.html

HSV

- V = Intensity
  - 0 = Black
  - 1 = Max (white at S = 0)
- S = 1:
  - As H goes from 0 (Red) to 360, it represents a different combinations of 2 colors
- As S->0, the color components from the opposite side of the polygon increase



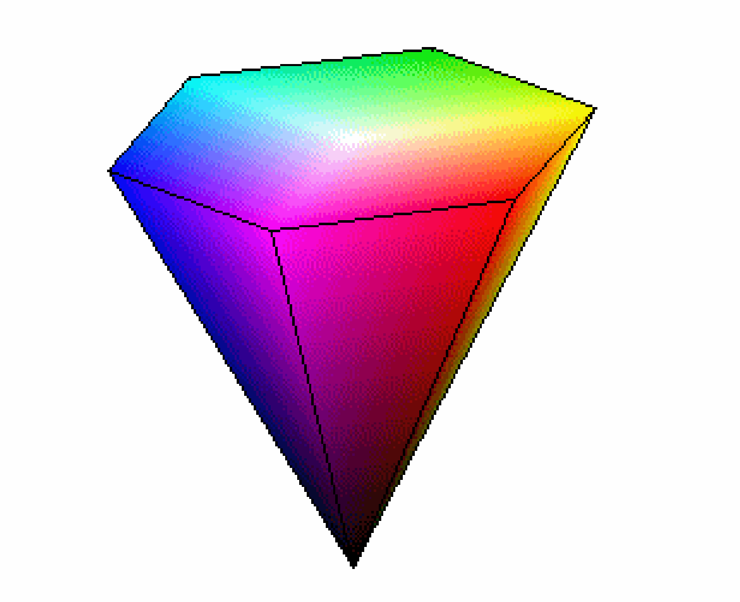
V = [0,1], S = [0,1]
H = [0,360]

Hue, Saturation, Value

$$h = \begin{cases} 0, & \text{if } \max = \min \\ (60^\circ \times \frac{g-b}{\max - \min} + 360^\circ) \bmod 360^\circ, & \text{if } \max = r \\ 60^\circ \times \frac{b-r}{\max - \min} + 120^\circ, & \text{if } \max = g \\ 60^\circ \times \frac{r-g}{\max - \min} + 240^\circ, & \text{if } \max = b \end{cases}$$

$$s = \begin{cases} 0, & \text{if } \max = 0 \\ \frac{\max - \min}{\max} = 1 - \frac{\min}{\max}, & \text{otherwise} \end{cases}$$

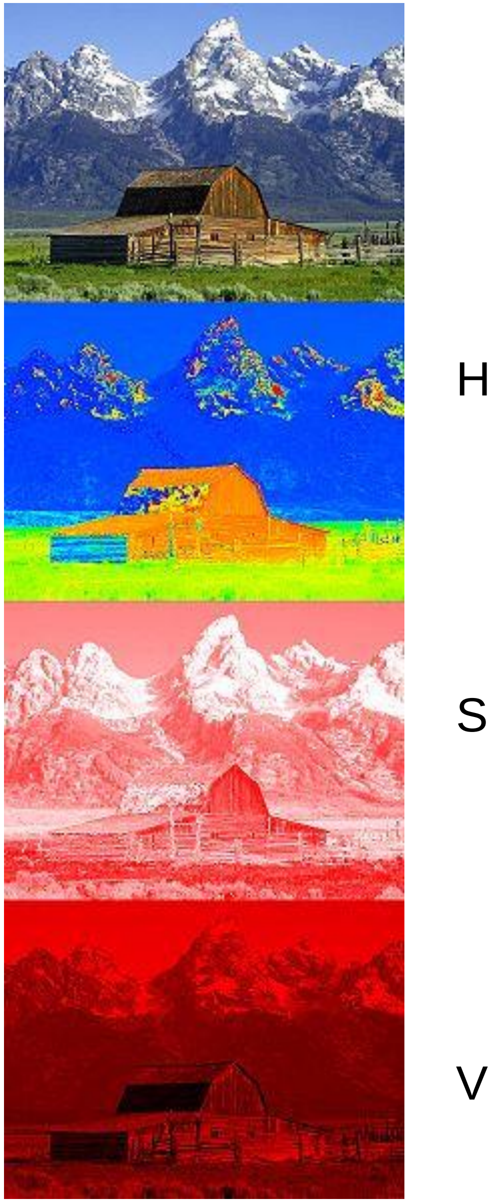
$$v = \max$$



Max is the maximum of (R,G,B)
Min is the minimum of (R,G,B)

HSV

- Top: Original image
- Second H (assuming S = 1, V = 1)
- Third S (H=0, V=1)
- Fourth V (H=0, S=1)



Quantization and Saturation

- Captured images are typically quantized to N-bits
- Standard value: 8 bits
- 8-bits is not very much < 1000:1
- Humans can easily accept 100,000:1
- And most cameras will give you 6-bits anyway…

Processing Colour Images

- Typically work only on the Grey Scale image
  - Decode image from whatever representation to RGB
  - GS = R + G + B
- The Y of YIQ may also be used
  - Y is a linear combination of R,G and B
- For specific algorithms that deal with colour, individual colours may be maintained
  - Or any linear combination that makes sense may be maintained.

Other Signals

- Direct measurement (like sound):
  - ECG, EMG, EKG
- Indirect measurement (through a transform)
  - MRI
    - Takes measurements in the *Fourier domain*

The General Theory of Sensing

- Actual signal : $y(j)$
  - j may be time, position, etc..
  - Usually continuously valued
- Captured value:
  - $y(J)=\int_{\Theta} y(j)K(j-J)dj \;\; ; \;\; \Theta$ is the space of all j
  - $K(j)$ is a measurement kernel
  - Ideally a *delta* (which takes non-zero value only at the desired j)
    - Captures *actual* snapshots
  - But in reality not
    - More on this later..

Next Class..

- Review of linear algebra..