IERG4190/IEMS5707 Multimedia Coding and Processing

Chapter 4: Multimedia Coding (3)

Bolei Zhou
Department of Information Engineering, CUHK
Spring 2021

Outline

1. Light, Color & Human Visual System Model
2. Human Hearing System
3. Subband Coding
4. Zonal Coding and Threshold Coding
5. JPEG
6. Video Coding

A Note about Learning Outcomes

- In early years you learn about basics and operations
  - Like in Kung Fu, you learn to punch and kick
- In final year/onward, at least, you should go beyond that
  - Using again Kung Fu analogy…
    - When to punch
    - Who to kick
    - Why you are kicking or punching
    - When not to use Kung Fu

Coding for Specific Media

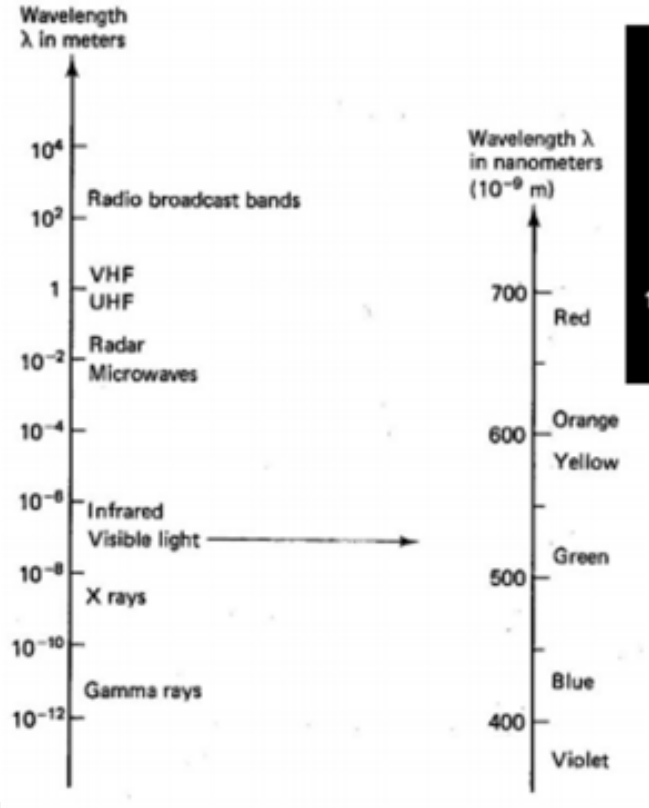
- We have looked at different parts of multimedia coding but we haven't looked into the media itself
  - **Image**
  - **Sound**
  - **Video**
- Image - Light, color, human visual system
- Audio - Human audio system
- Video - Going beyond images
- For coding to be successful, we need knowledge in different media

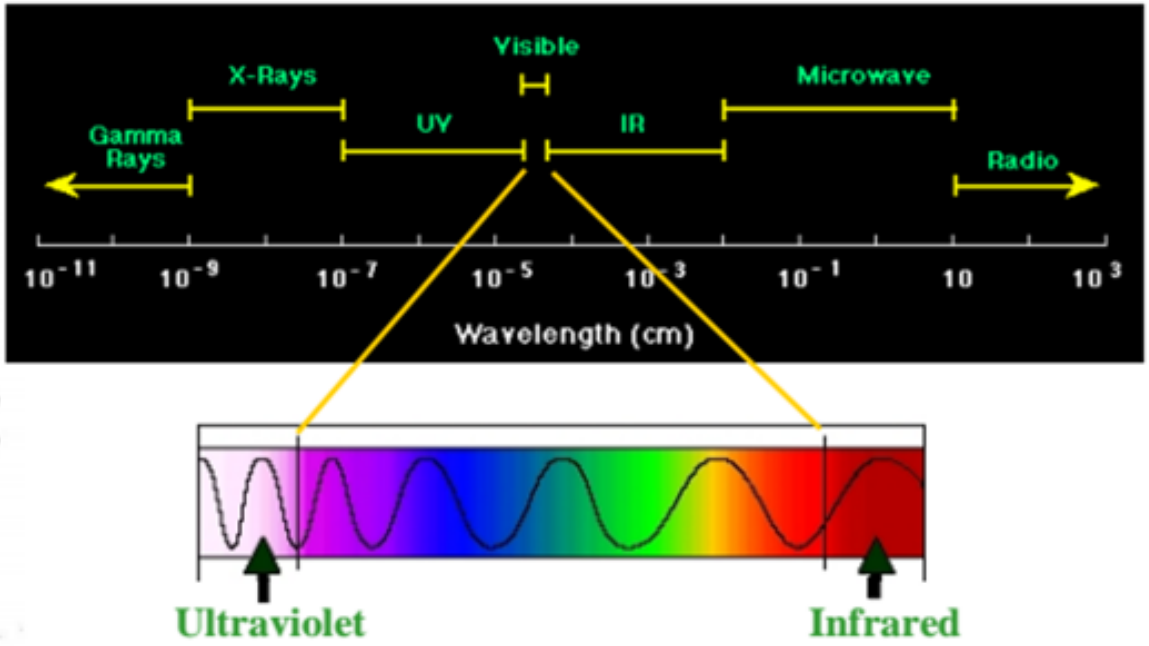
Physics Revisited: Light

Energy density of electromagnetic wave:

$$c(x, y, t, \lambda), \qquad joules/m^2 \cdot \sec \cdot m$$

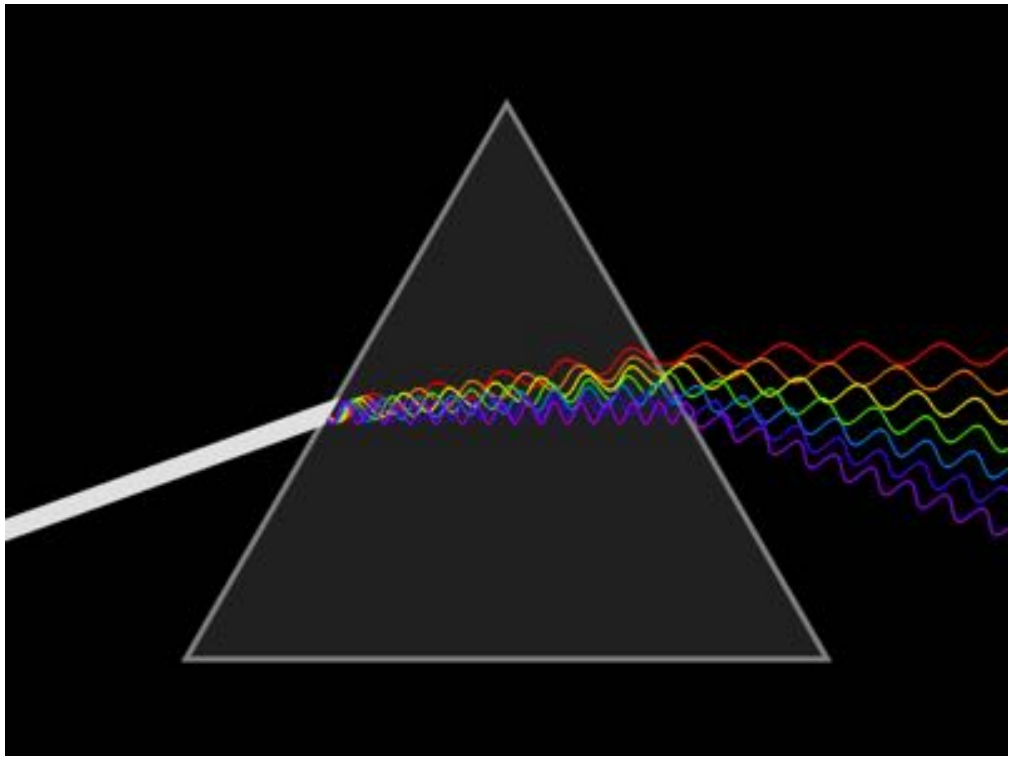
Light is the visible part of the spectrum





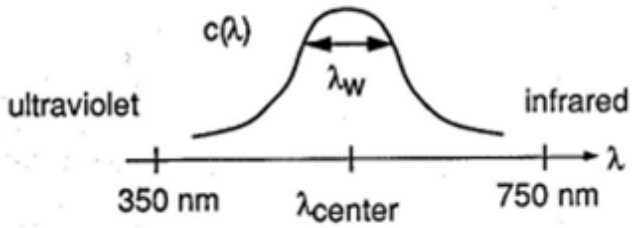
Physics Revisited: Dispersive prism

- A white light is split into a spectrum due to the different wavelengths of light



Light

At a specific x, y, t,



Physical Domain	very approximate	Perceptual Domain
$\int_{\lambda_{min}}^{\lambda_{max}} c(\lambda)s(\lambda)d\lambda$		Brightness
λ_{center}		Hue
λ_w		Saturation

s: Luminous efficiency function: average spectral sensitivity of human visual perception of brightness

Light

For black and white images

$$I = k \int_{\lambda=0}^{\infty} c(\lambda) s_{BW}(\lambda) d\lambda$$

I : intensity, luminance, gray level, usually scaled to 0 – 255.

$s_{BW}(\lambda)$: C.I.E. relative luminous efficiency function, measuring the spectral characteristics of the sensor.

C.I.E. (Commission Internationale de l'Eclairage): international body concerned with standards for light and color.

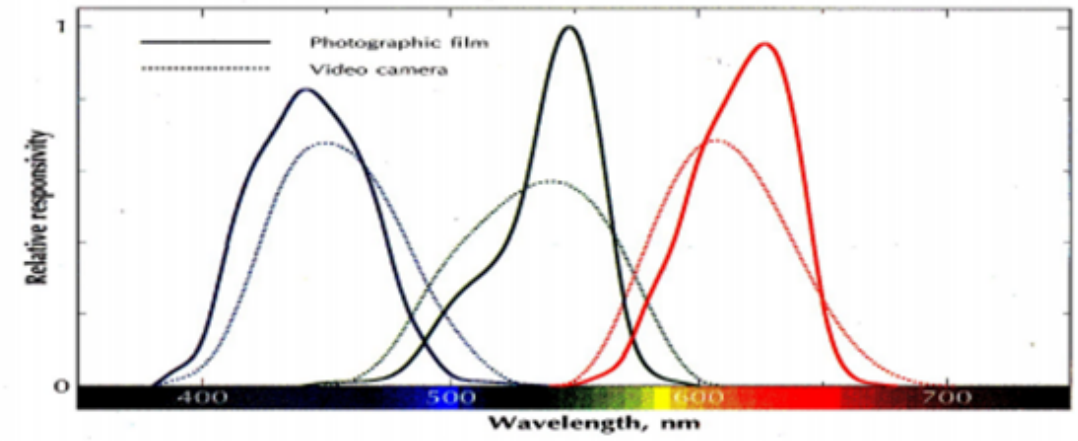
Light

Three primary colors (a set of tristimulus values):

Red component: $$I_R = \int_{\lambda=0}^{\infty} c(\lambda) s_R(\lambda) d\lambda$$

Green component: $$I_G = \int_{\lambda=0}^{\infty} c(\lambda) s_G(\lambda) d\lambda$$

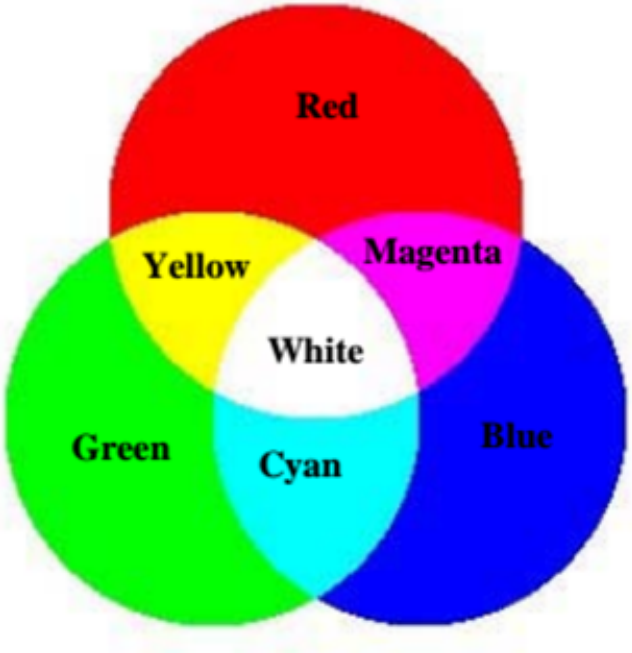
Blue component: $$I_B = \int_{\lambda=0}^{\infty} c(\lambda) s_B(\lambda) d\lambda$$



Color

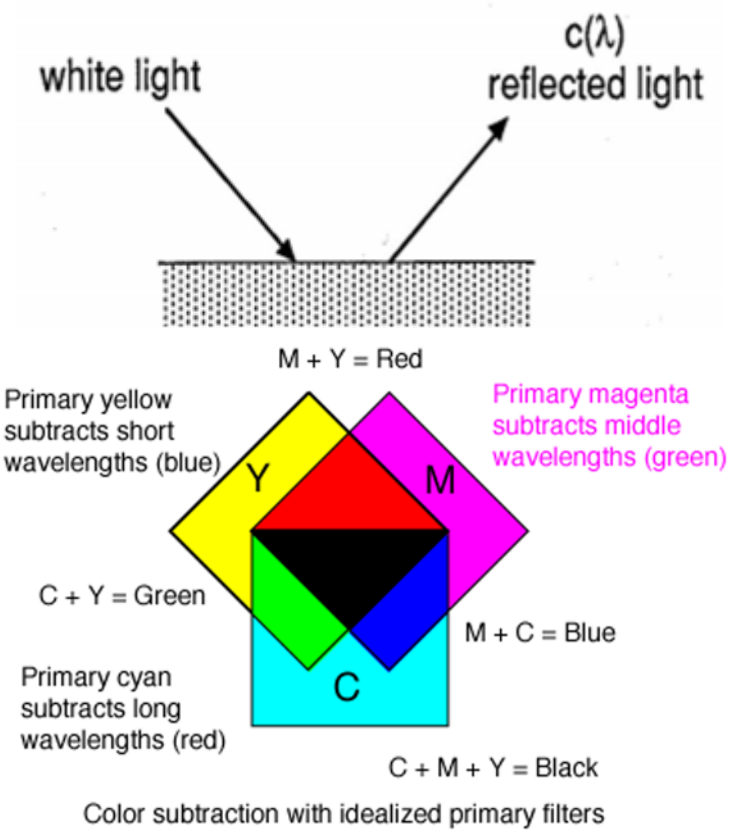
Additive color system: $c(\lambda) = c_1(\lambda) + c_2(\lambda)$

Examples:
Shine two lights with $c_1(\lambda)$ and $c_2(\lambda)$ on white screen;
Screen of a TV tube: red, green, blue, three primary colors.



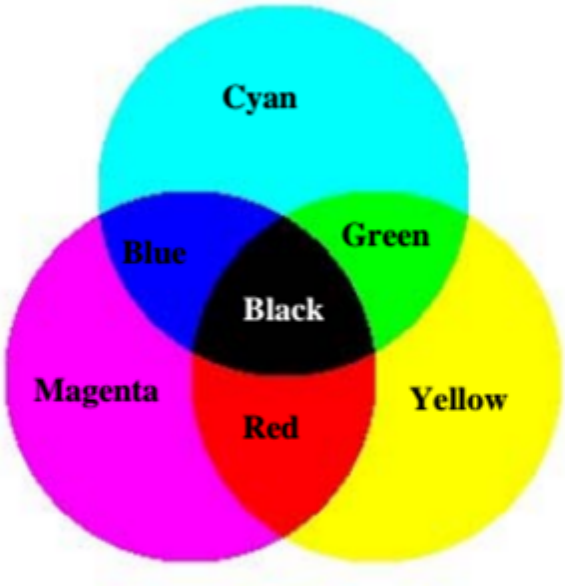
Color

Subtractive color system: yellow, cyan, magenta, three primary colors.



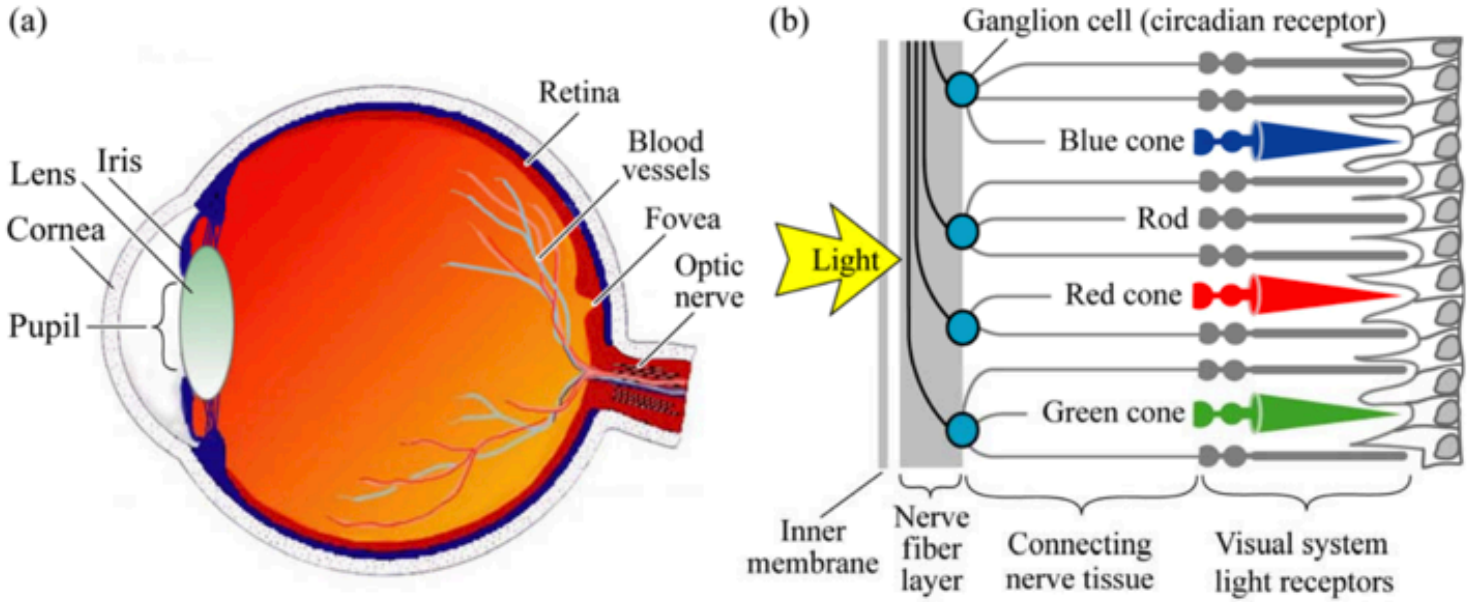
Color subtraction with idealized primary filters

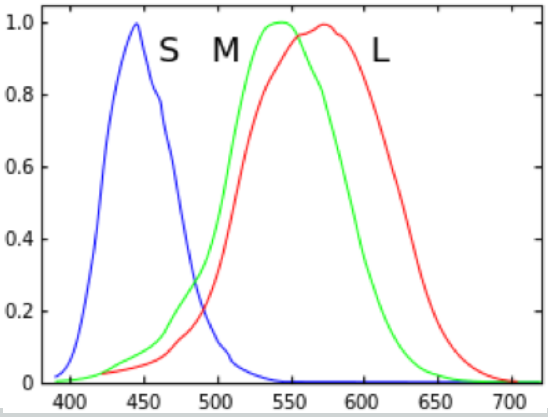
http://hyperphysics.phy-astr.gsu.edu/hbase/vision/subcol.html



Do you know its main application?

Human Visual System: Eye

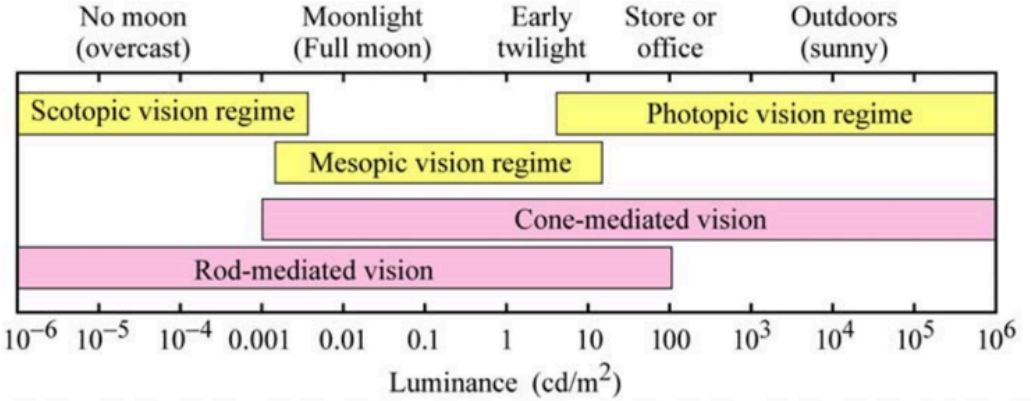




Spectral sensitivities of human cone cells, S, M, and L types

Human Visual System: Eye

- Rod cells and cone cells



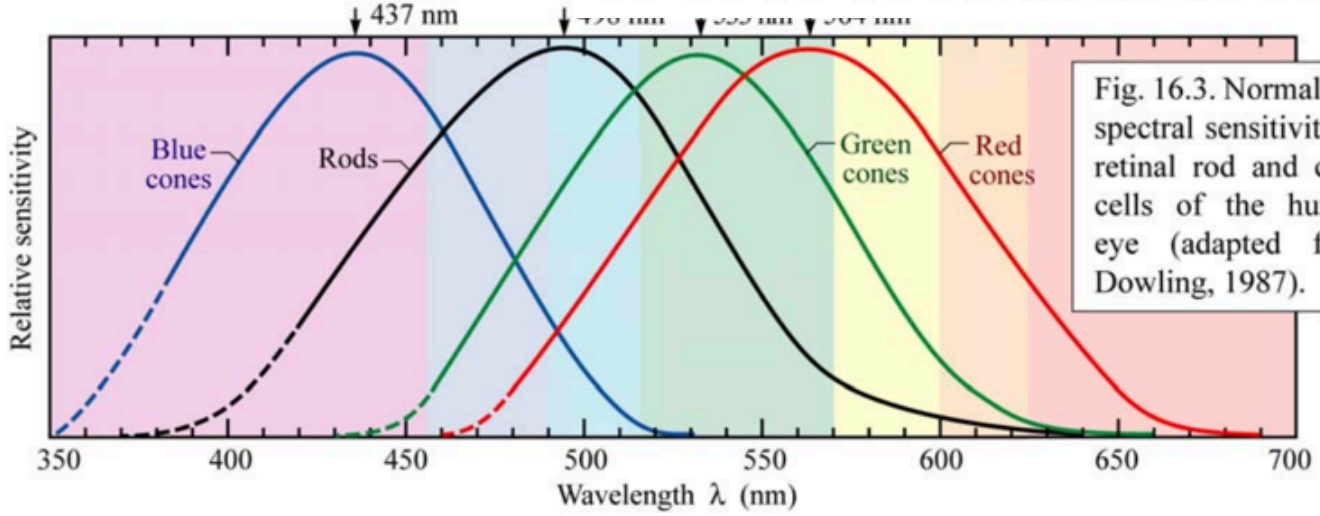
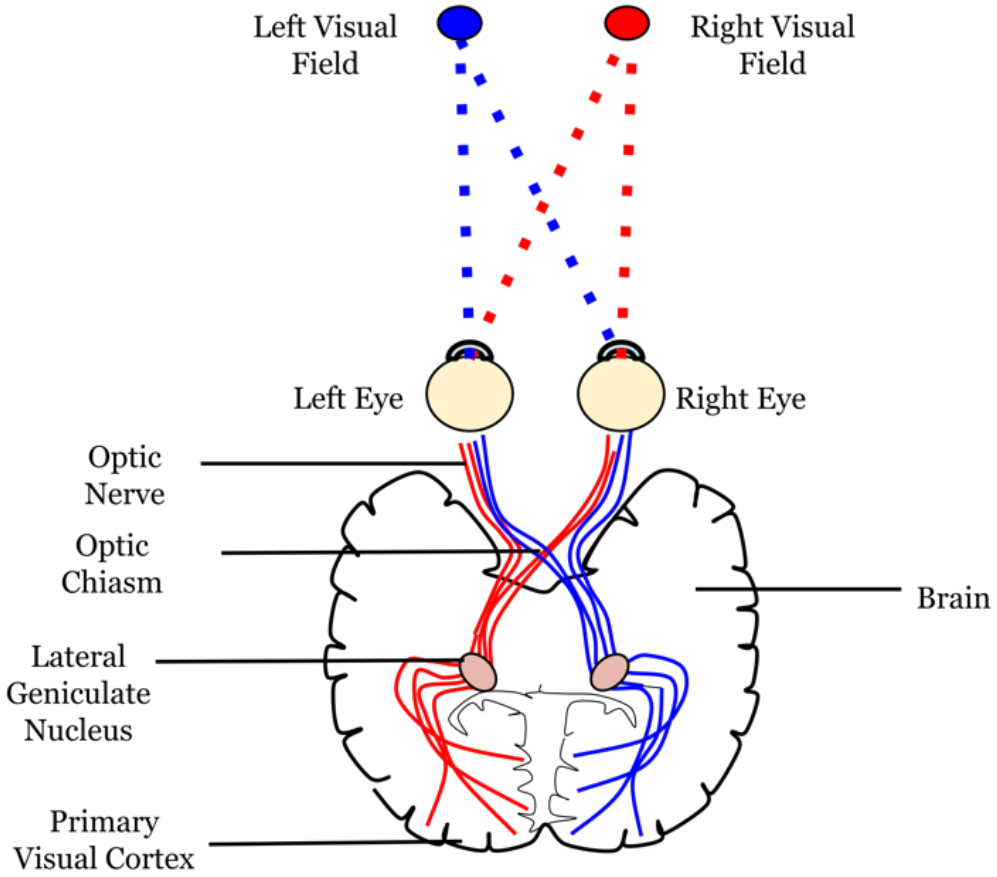
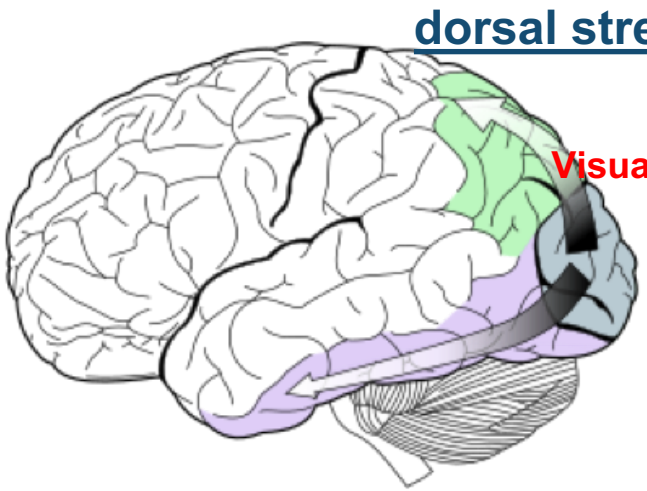


Fig. 16.3. Normalized spectral sensitivity of retinal rod and cone cells of the human eye (adapted from Dowling, 1987).

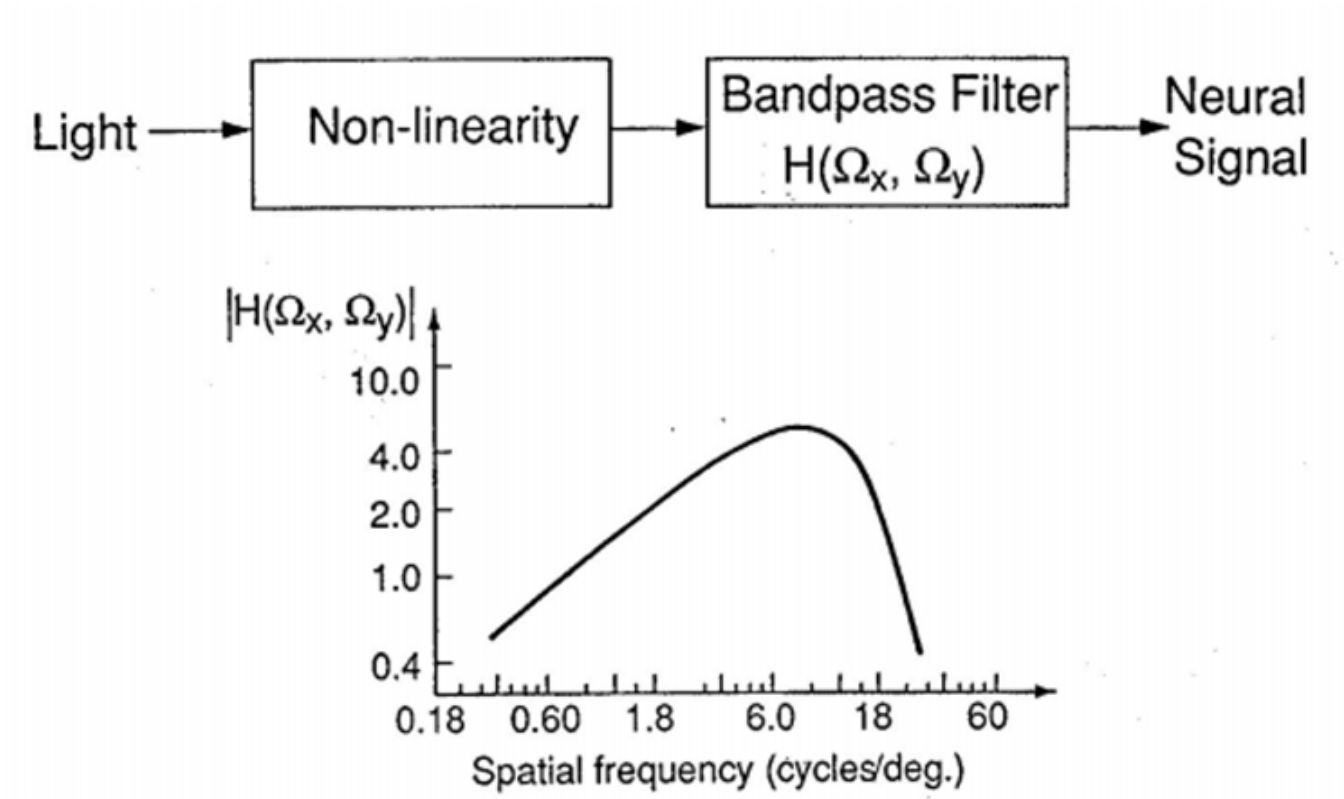
Human Visual System





Human Visual System

One simple model of the visual system

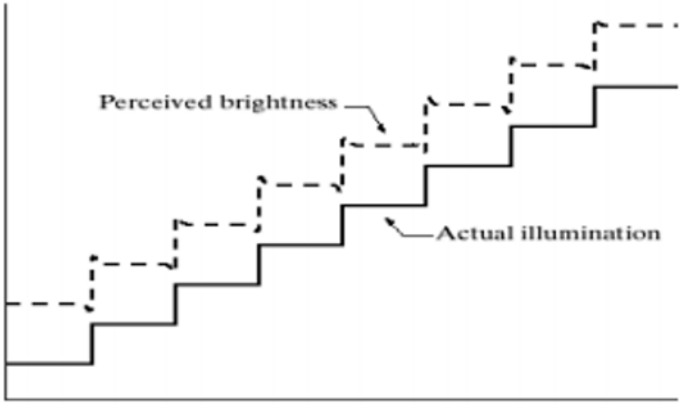


Human Visual System

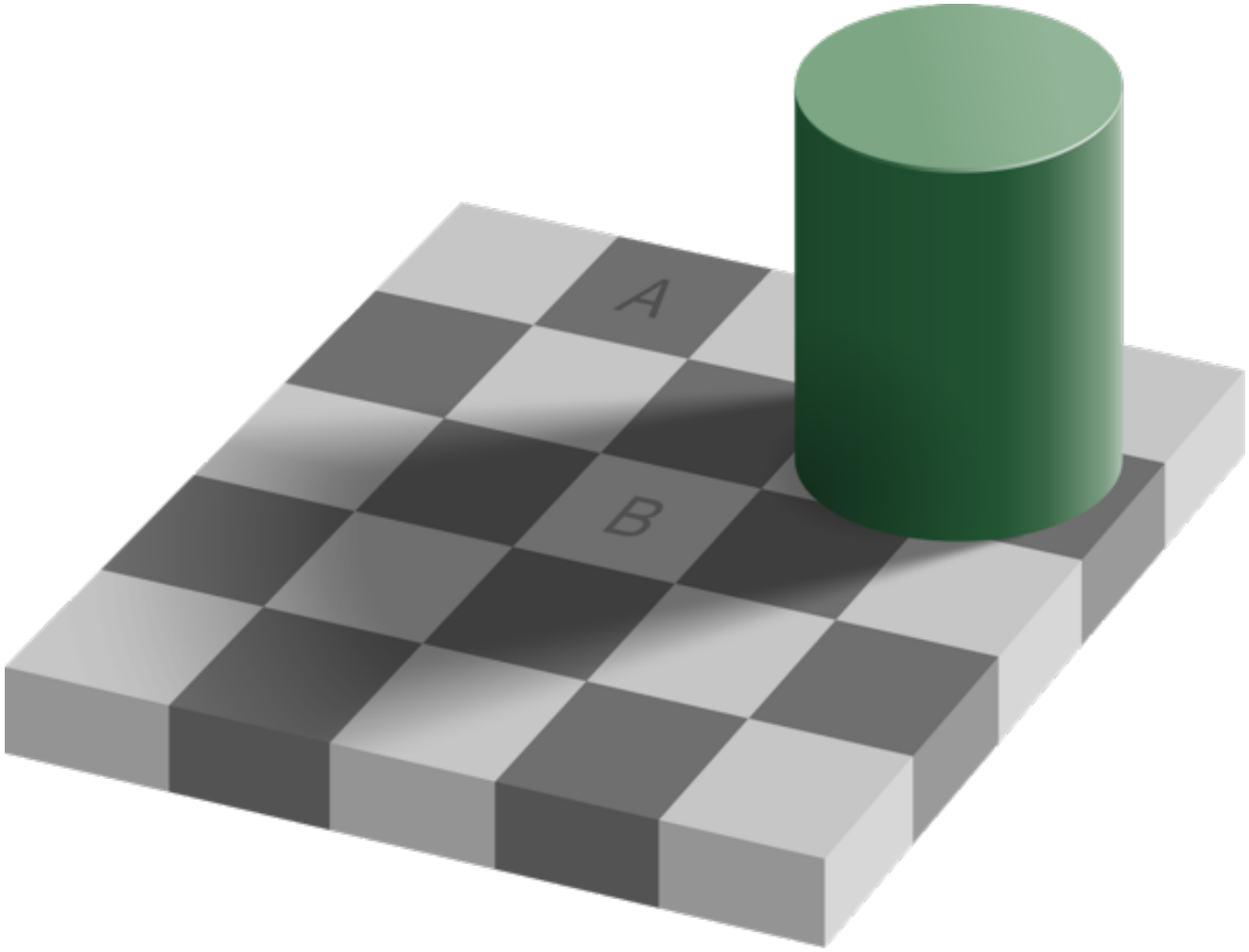
- Researchers have an interest in human perception of light and color for a long time...

Mach Band Effect (in 1865)

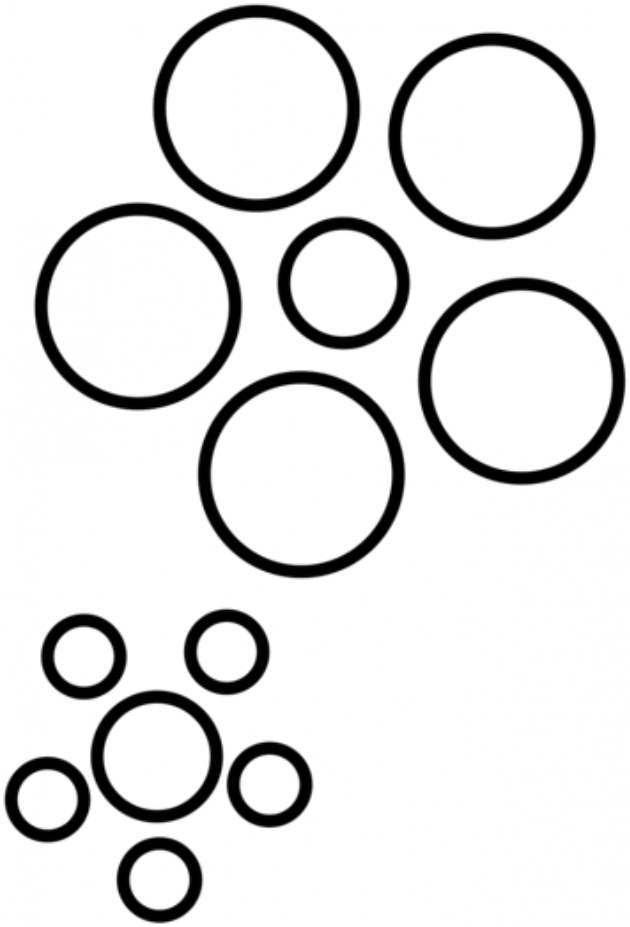
- **undershoots** and **overshoots** around the boundary of regions of different intensities

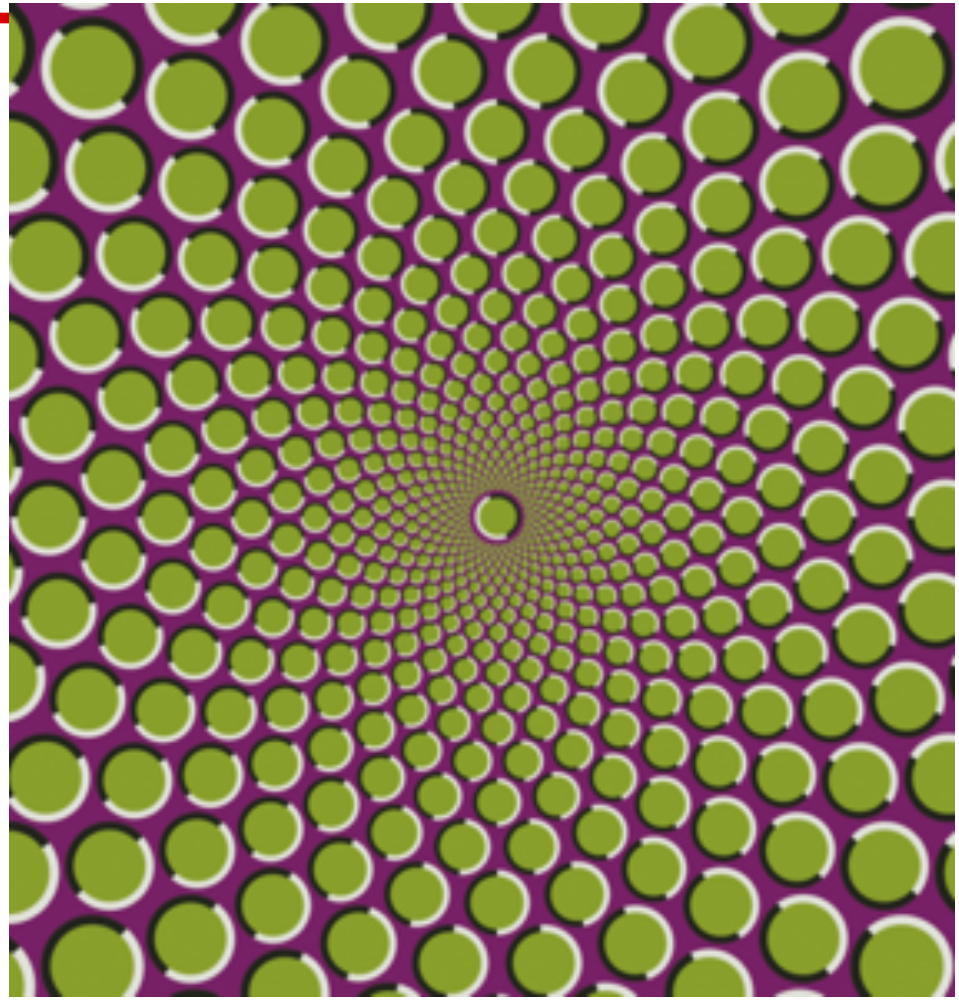


Human Visual System: Visual Illusion



Human Visual System: Visual Illusion





https://interestingengineering.com/11-puzzling-optical-illusions-and-how-they-work

Human Visual System Model

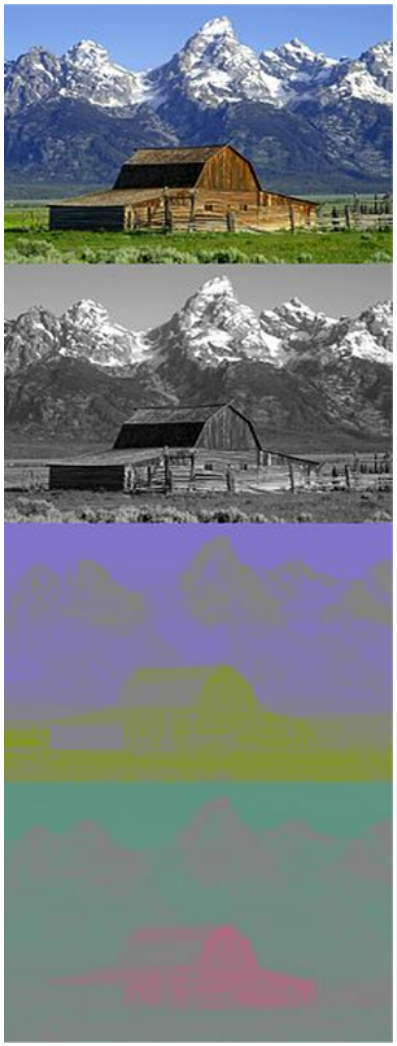
- Human visual system is far too complicated
- To simplify our understanding, we utilize a Human Visual System (HVS) Model
  - Gets updated whenever there are significant new knowledges
  - Exploited in many image coding and processing applications and standards

Human Visual System Model

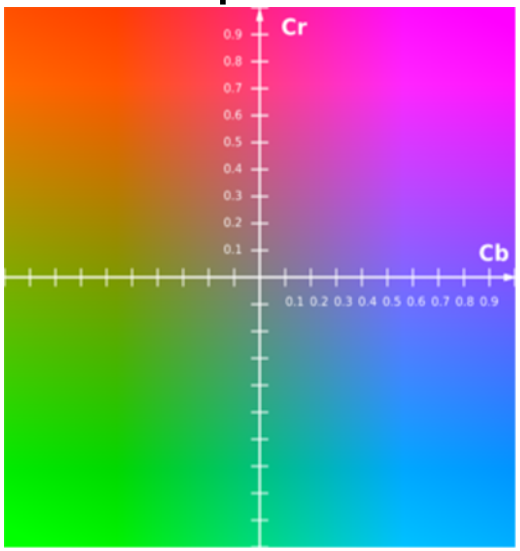
Examples:

- Frame-per-second (FPS) - dependent on how humans perceive continuous motion
  - Minimum needed: 25-30 fps
  - For high brightness case: 60 fps
- Chrominance less sensitive than luminance
  - Human has lower resolution for color than pure brightness
  - Let's use lower resolution for the color information! ("Chroma Subsampling")
    - Requires conversion from RGB to another color space like YUV/YCbCr

YCbCr



CbCr plane



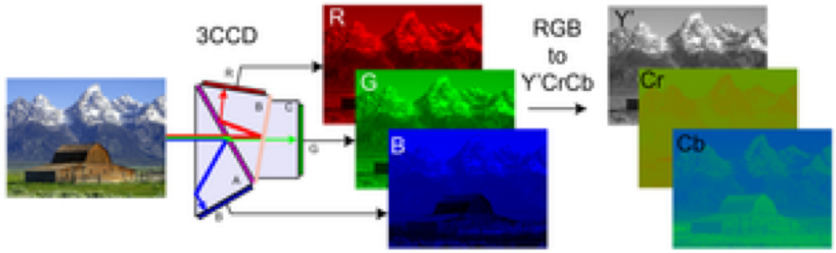
Conversion between YCbCr and RGB

The equivalent matrix manipulation is often referred to as the "color matrix":

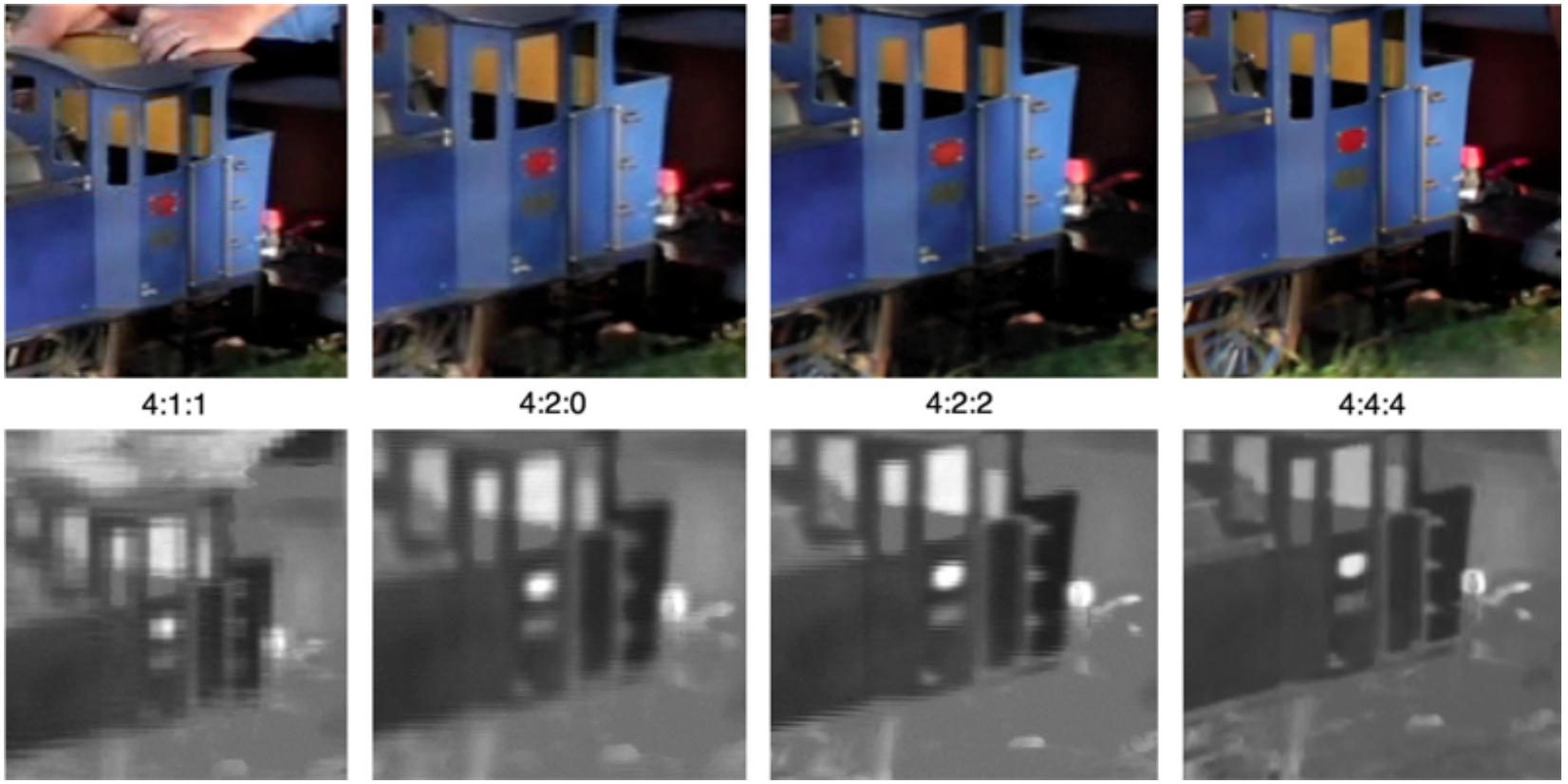
$$\begin{bmatrix} Y' \\ P_B \\ P_R \end{bmatrix} = \begin{bmatrix} K_R & K_G & K_B \\ -\frac{1}{2} \cdot \frac{K_R}{1-K_B} & -\frac{1}{2} \cdot \frac{K_G}{1-K_B} & \frac{1}{2} \\ \frac{1}{2} & -\frac{1}{2} \cdot \frac{K_G}{1-K_R} & -\frac{1}{2} \cdot \frac{K_B}{1-K_R} \end{bmatrix} \begin{bmatrix} R' \\ G' \\ B' \end{bmatrix}$$

And its inverse:

$$\begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} = \begin{bmatrix} 1 & 0 & 2-2 \cdot K_R \\ 1 & -\frac{K_B}{K_G} \cdot (2-2 \cdot K_B) & -\frac{K_R}{K_G} \cdot (2-2 \cdot K_R) \\ 1 & 2-2 \cdot K_B & 0 \end{bmatrix} \begin{bmatrix} Y' \\ P_B \\ P_R \end{bmatrix}$$



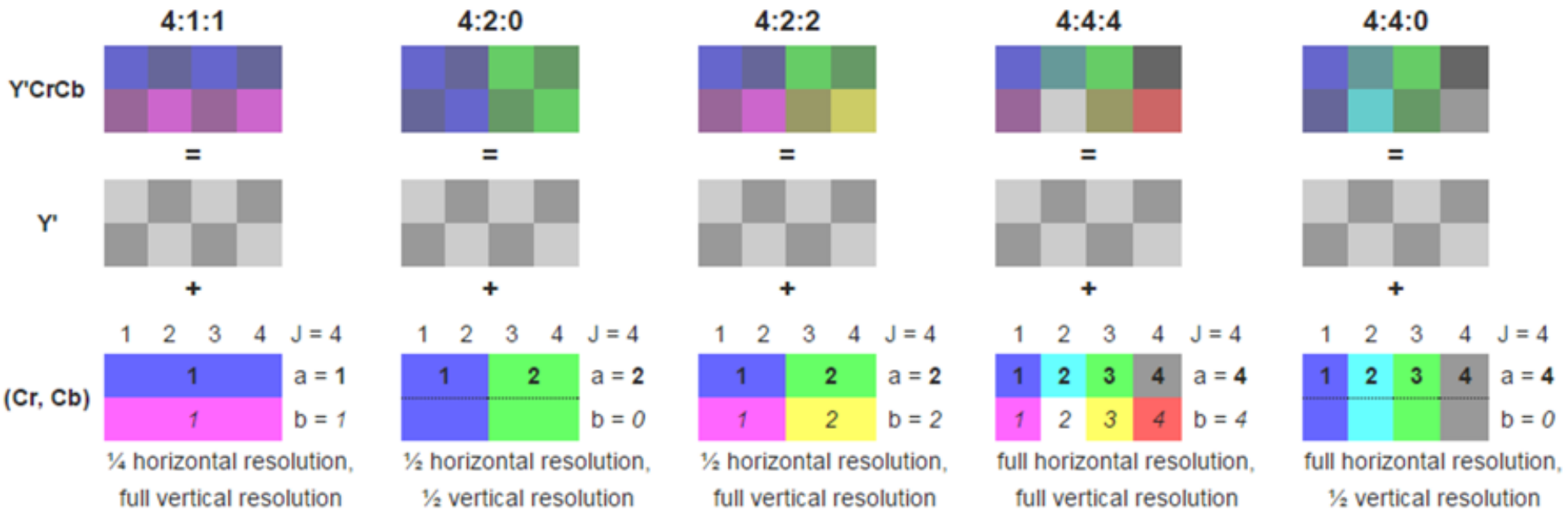
Chroma Subsampling



The bottom row indicates the resolution of color information for the corresponding image in the top row. Notice the lack of difference in the final color image.

source: en.wikipedia.org

Chroma Subsampling



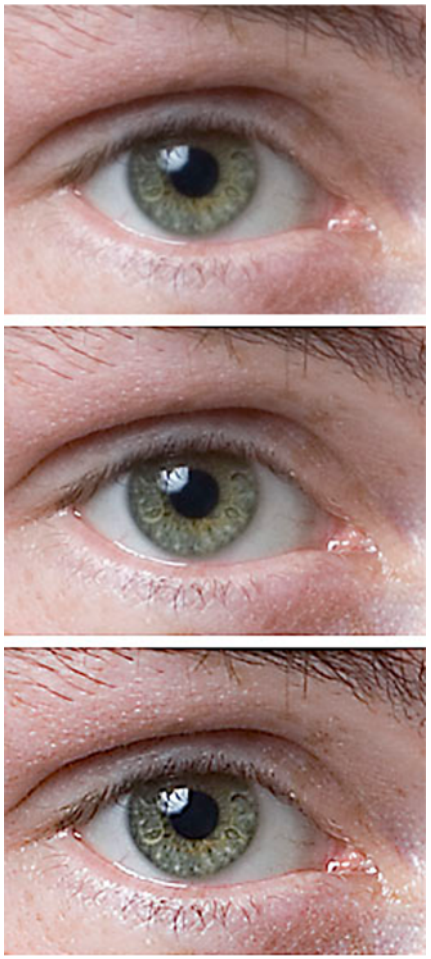
- *J*: horizontal sampling reference (width of the conceptual region). Usually, 4.
- *a*: number of chrominance samples (Cr, Cb) in the first row of *J* pixels.
- *b*: number of changes of chrominance samples (Cr, Cb) between first and second row of *J* pixels.

source: en.wikipedia.org

Human Visual System Model

- Mach Bands - used in creating a "sharpening" illusion

(Image Processing to be covered later)



Human Visual System Model

More Examples

- "Inverted face problem"
- "Grandmother neuron"
  - To be explained in computer vision
- Less sensitive to higher spatial resolutions
  - To be used in our JPEG image coding standard

Human Hearing System

Basic structure of human hearing system:

Sound propagates through air as pressure waves
Sound velocity = 344 meters/second in air at 20 degree
The hearing range of human beings: 20 Hz ~ 20k Hz

Power level in decibels (dB):

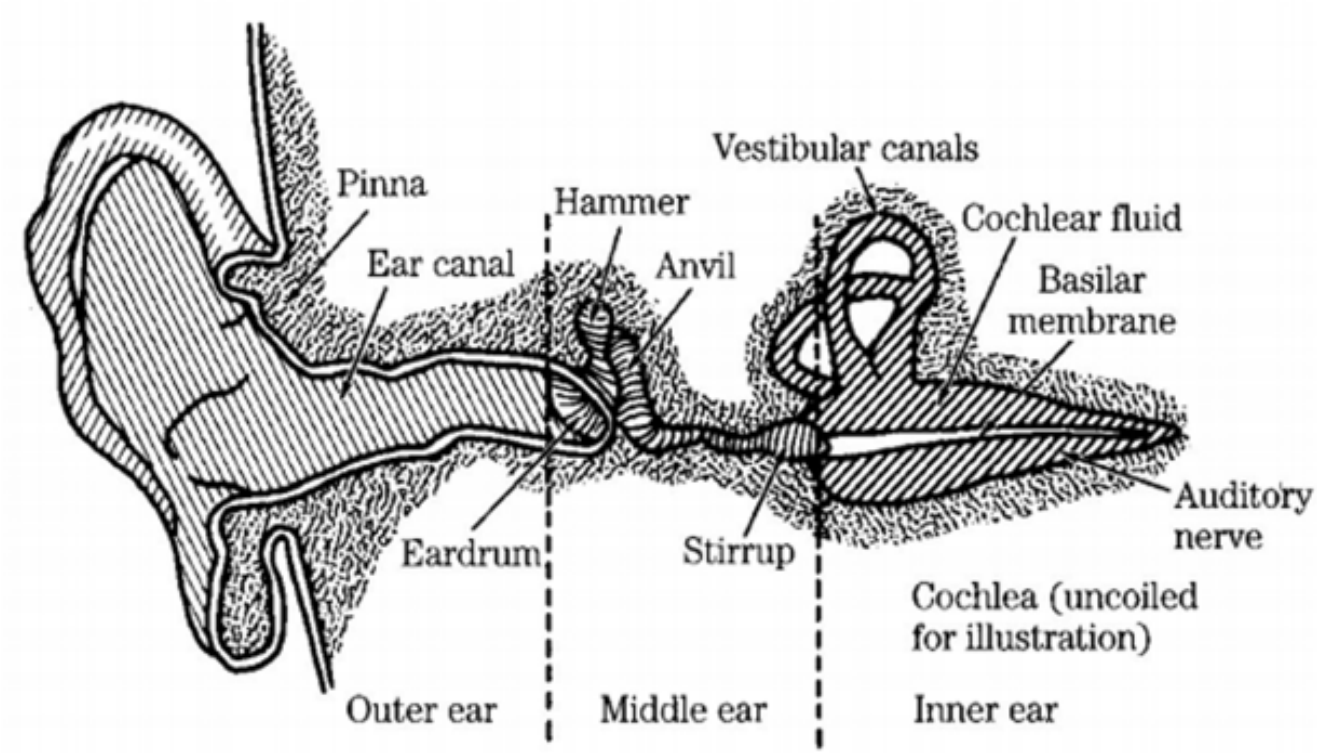
$$\mathrm{PWL} = 10\log_{10}\left(\frac{\text{sound power output}}{10^{-12}\ \text{watts}}\right)$$

Human shout has PWL of 70 dB, large rocket motor 190 dB.

The air pressure wave vibrates our eardrum. Then the pressure is transposed to the inner ear, which consists of fluid-filled structure. In this structure, mechanical pressure is transformed into electrochemical signals in frequency domain.

Human Hearing System

The electrochemical signals are then transported to the auditory systems in the brain.



Human Hearing System

- Challenge of a hearing system
  - In vision, different signals tend to occupy different areas on the retina; eyeball movements further allow us to select signals
  - In hearing, we can't choose much - they come from everywhere and we have ears fixed on both sides!
- A hearing system must handle simultaneous signals well

Critical Bands

Critical Bands

They relate to the ability of our hearing system to distinguish and separate two signals which are present simultaneously. Our ears only respond to the strongest stimulation in a local frequency region.

Critical bands are much narrower at low frequencies than at high frequencies. It can be approximated as,

$$\text{Critical bandwidth in hertz} = 24.7(4.37F + 1)$$

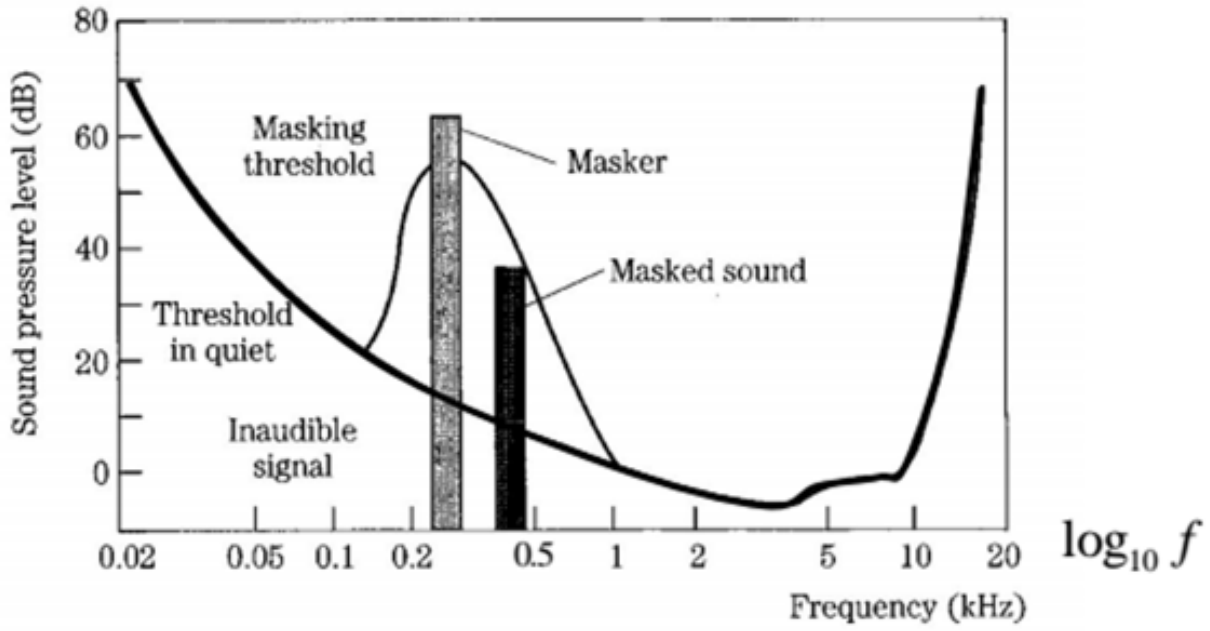
F is the center frequency in kHz. There are approximately 25 critical bands within the human hearing frequency range (see Table 11-1).

Threshold of Hearing

Threshold of Hearing

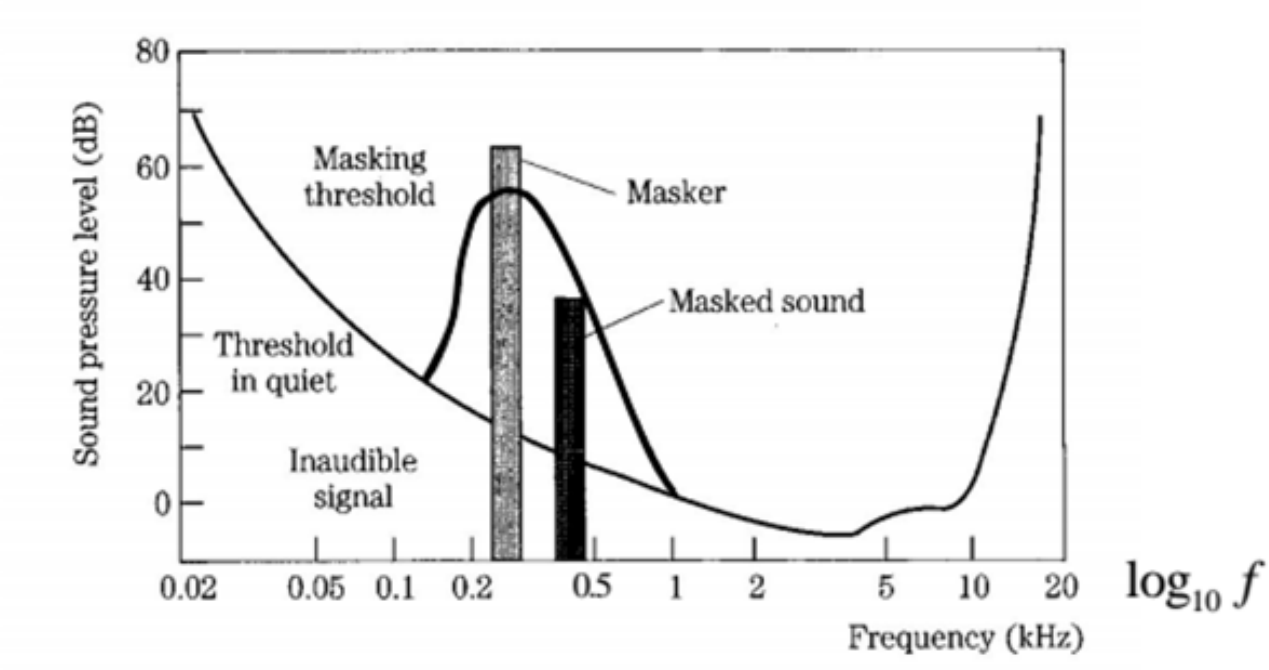
The hearing range of a human being: 20 Hz ~ 20k Hz.

The sensitivity of our hearing system is not equal over the total frequency range. A sound at a given frequency has a minimum level (threshold) in order to be heard.



Amplitude Masking

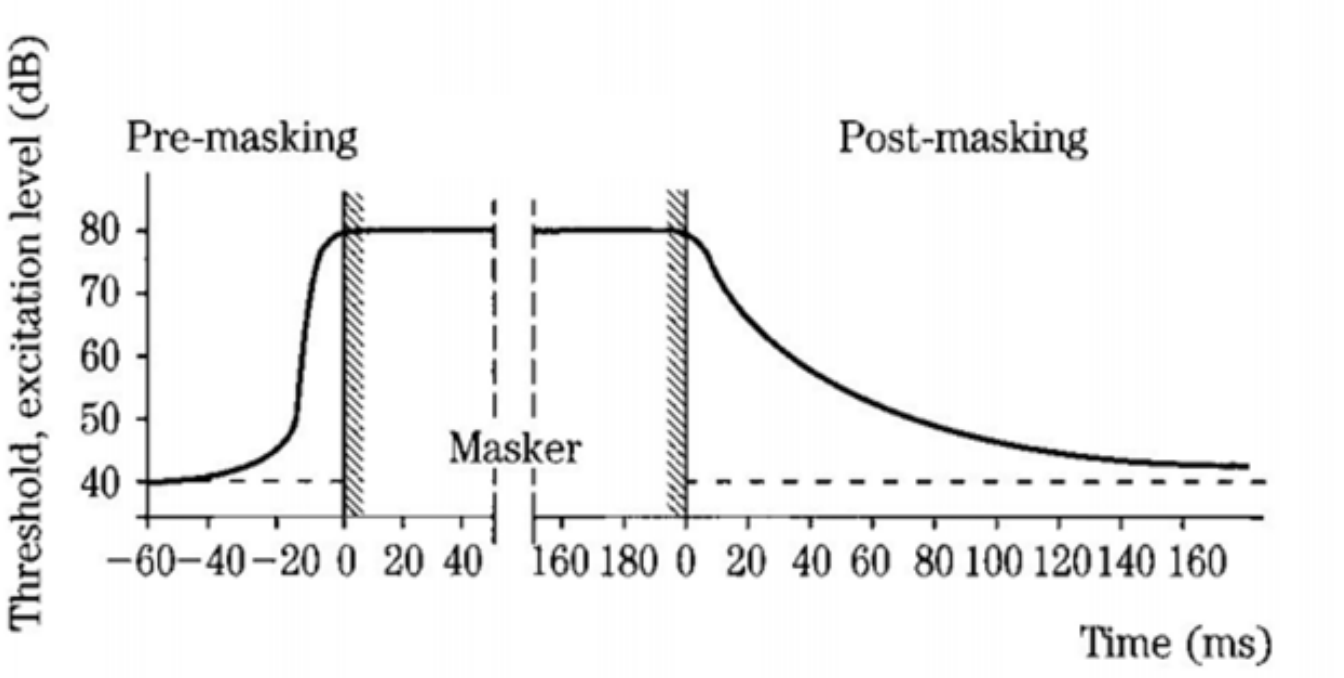
One frequency, with higher level, can make another frequency with a lower level inaudible.



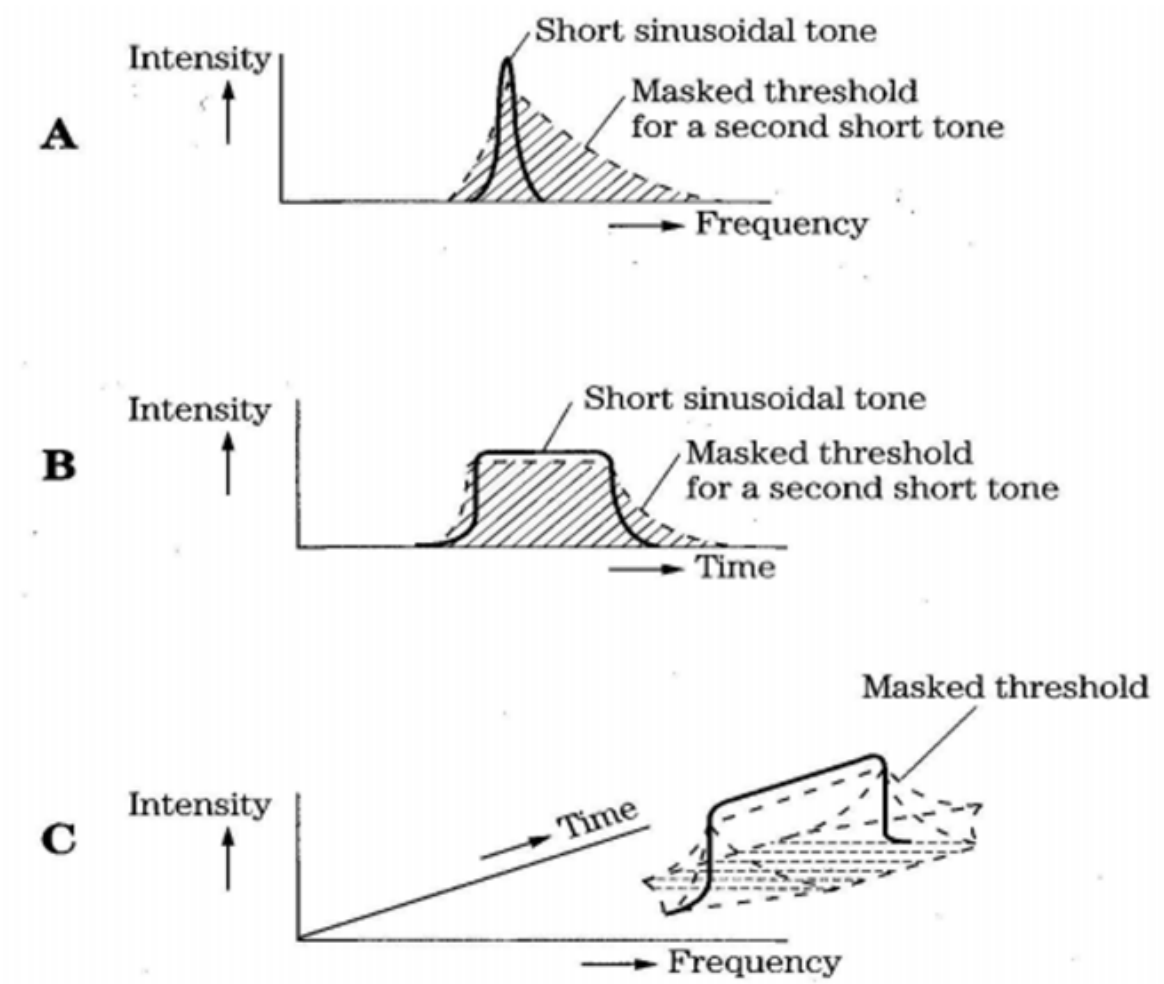
Temporal Masking

Temporal Masking

Besides simultaneous masking, the forward-masking (pre-masking) and backward-masking (post-masking) mean that a higher level sound can mask a lower level sound which is preceding or following.



Masking Combined

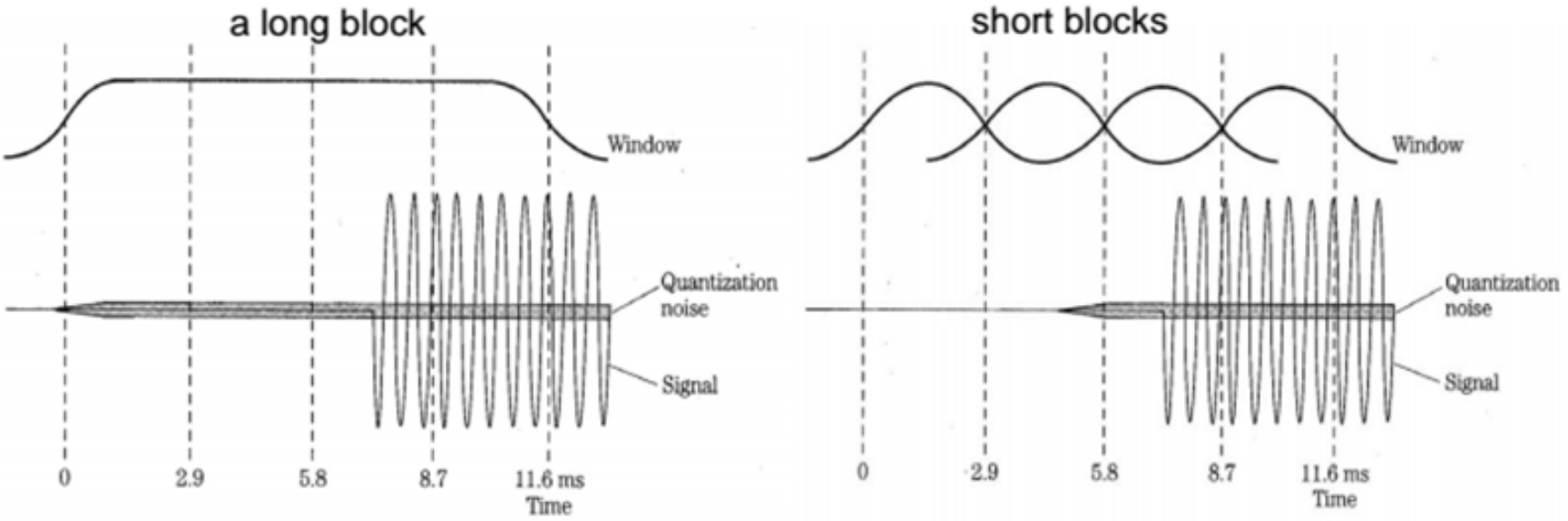


Subband Coding

A good coder should use the psychoacoustic models to adaptively quantize only the perceptually significant parts of the signal. Other parts of the signal masked by the significant parts are judged to be inaudible and are not coded.

Subband Coding

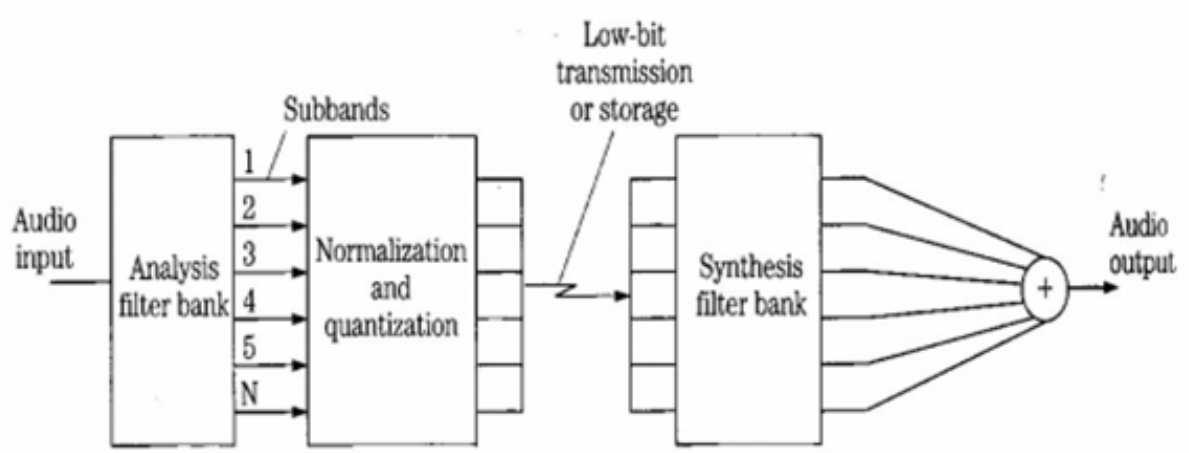
Temporal masking is calculated by dividing the audio signal into blocks of samples (typically several milliseconds in length) and analyzing each block for transients which will act as temporal maskers. Most systems vary the length of the block to take advantage of both backwards and forwards masking.



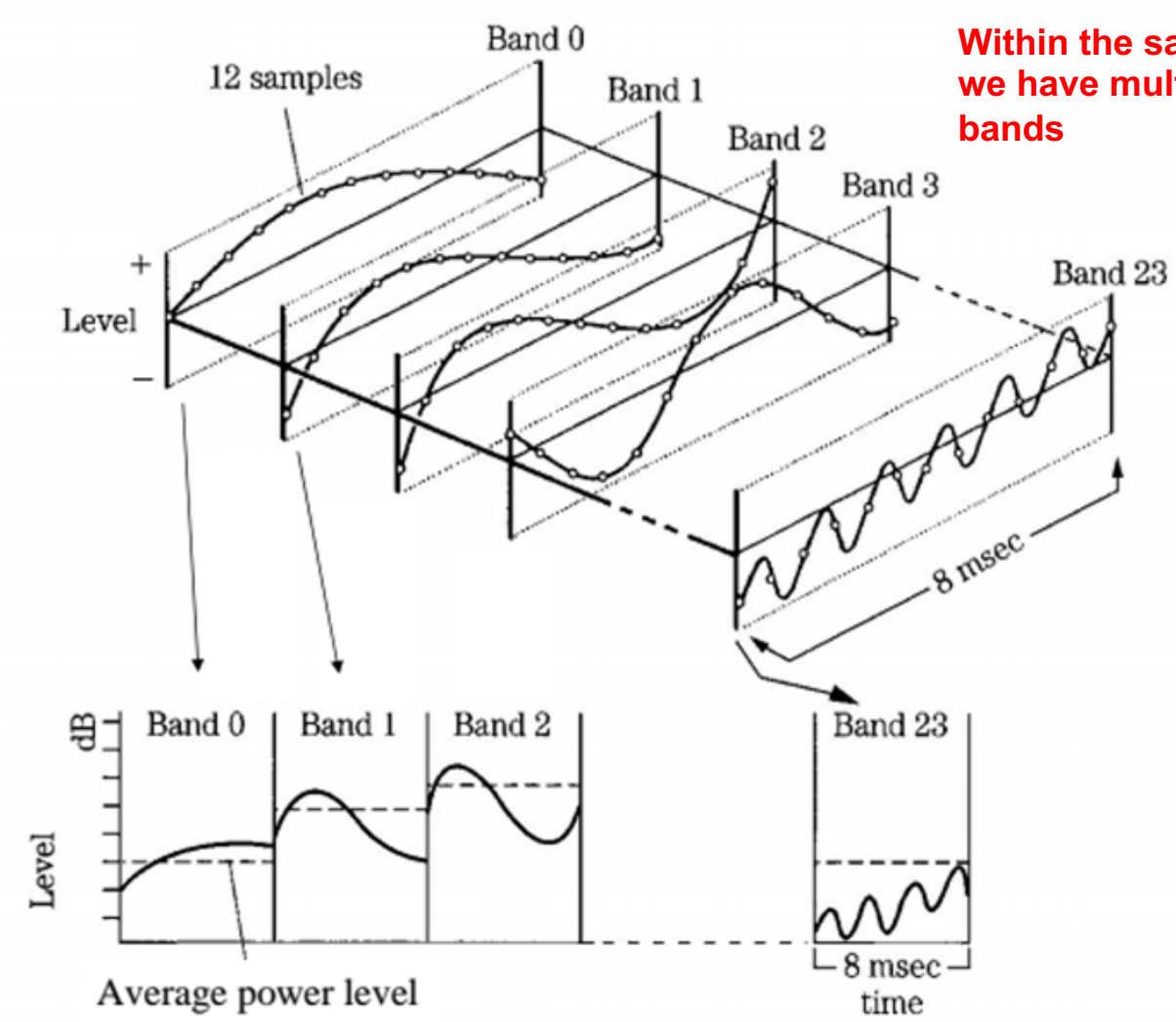
Subband Coding

A bank of filters divides the signal into many narrow bands (typically 32 or more) prior to processing, and a psychoacoustic model then analyzes the spectral content of the audio to determine which elements are likely to be completely masked (i.e. non-significant) and can therefore be discarded.

The remaining audible signals are quantized with a low resolution just sufficient to put the quantizing noise below the masking threshold in each frequency band.



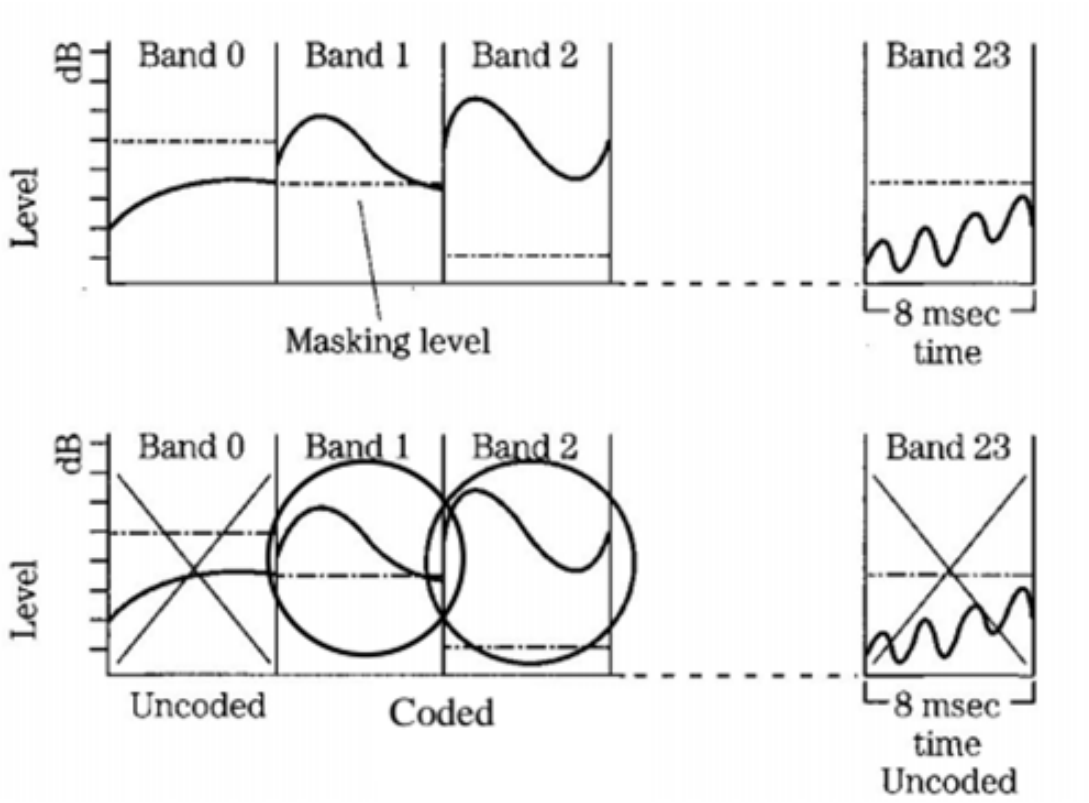
Subband Coding



Within the same time block, we have multiple frequency bands

Subband Coding

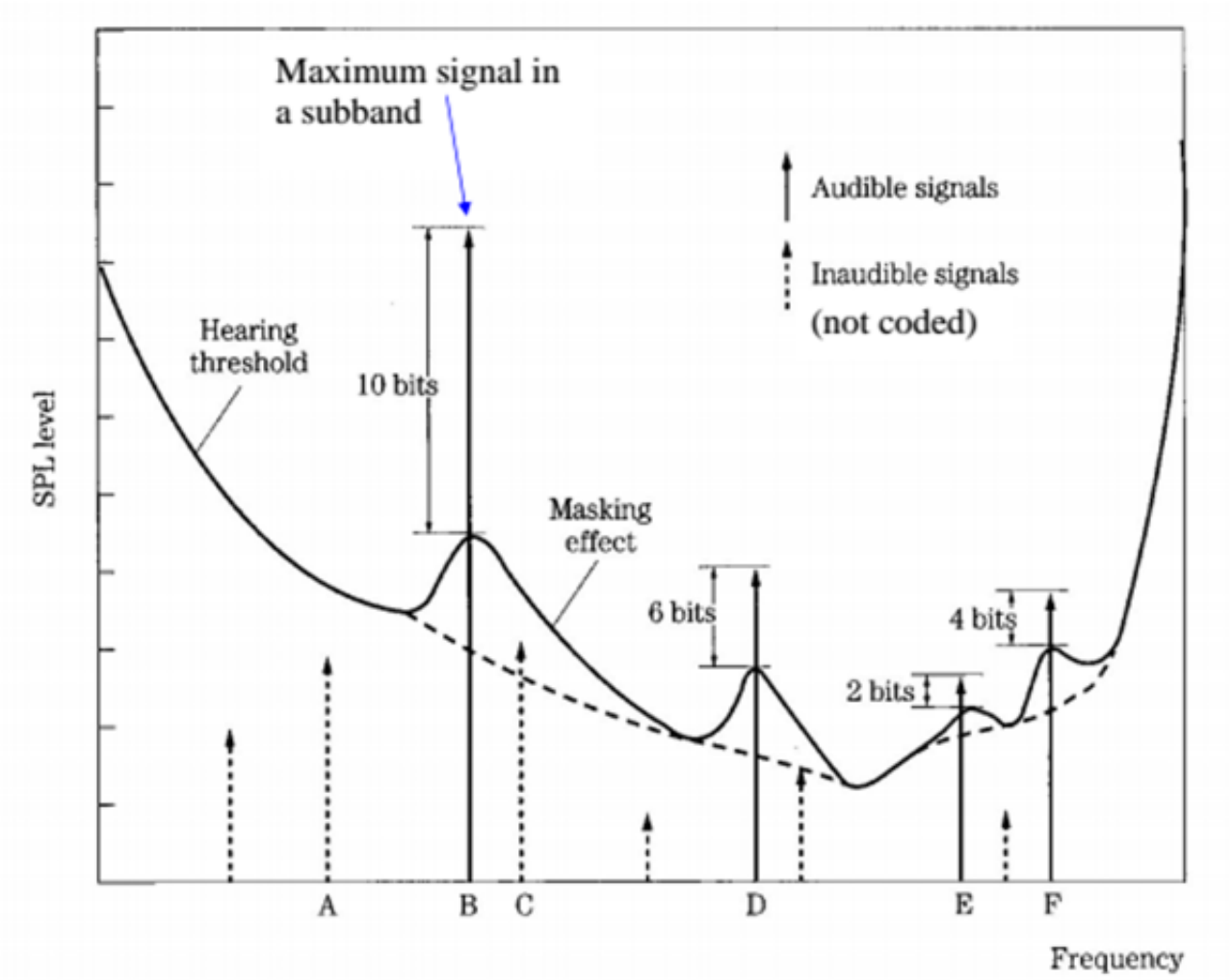
Masking levels obtained based on the psychoacoustic models, considering the average power level in each band, hearing threshold, and masking from signals in adjacent subbands.



Subband Coding

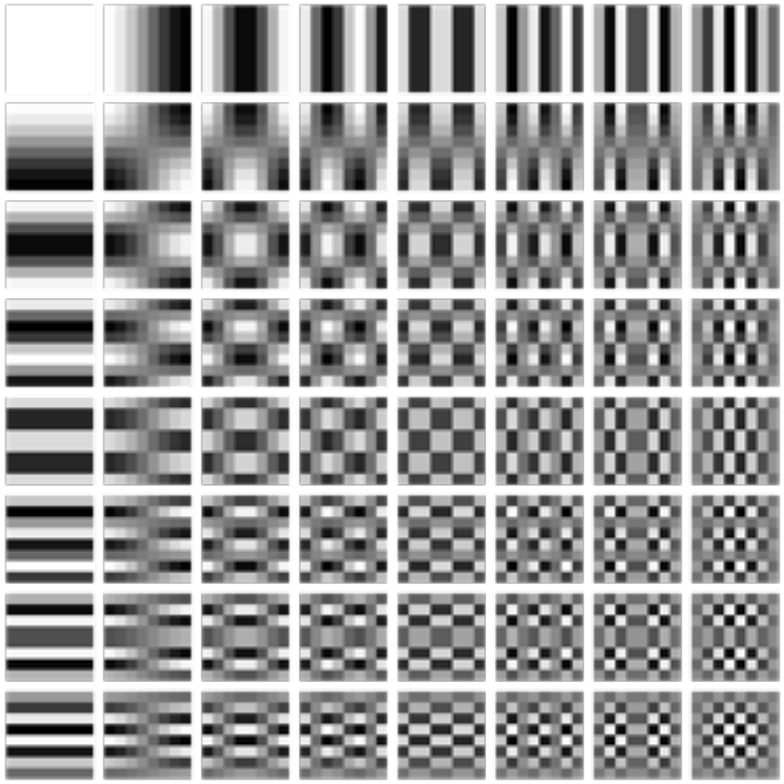
Bits allocation

More bits to significant subbands



DCT and Transform Coding

- So far, we have learned about DCT and its application in transform coding



DCT and Transform Coding

- Not feasible to transform whole image
- Divide into image blocks of 8 by 8 pixel or 16 by 16 pixel

Blocks clearly visible when JPEG quality is extremely low

DCT and Transform Coding

- We seek to retain transform coefficients that are most significant to our image blocks
- Apart from cropping insignificant transform coefficients, we may wish to encode the amplitude of the transform more carefully
  - Zonal Coding
  - Threshold Coding

Zonal Coding

- Assumption: transform coefficients of highest variances most important to image
- Calculate variances for each transform coefficients across each image block
- **Zonal coding masks** crop insignificant coefficients
- **Zonal bit allocation** further assigns more bits to coefficients with higher variances

Zonal Coding

1	1	1	1	0	0	0	0
1	1	1	0	0	0	0	0
1	1	0	0	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Example Zonal Coding Mask

8	6	4	2	0	0	0	0
6	4	2	0	0	0	0	0
4	2	0	0	0	0	0	0
2	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Example Zonal Bit Allocation

Zonal Coding

Basically means we care about this corner only

1	1	1	1	0	0	0	0
1	1	1	0	0	0	0	0
1	1	0	0	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Example Zonal Coding Mask

Clearly the more important ones will get more bits

8	6	4	2	0	0	0	0
6	4	2	0	0	0	0	0
4	2	0	0	0	0	0	0
2	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Example Zonal Bit Allocation

Threshold Coding

- Limits of Zonal Coding
  - Fixed mask over all blocks may not be optimal
  - Fixed bit allocation similarly suboptimal
- Let's be **adaptive**!
- Threshold Coding: transform coefficient with largest magnitude, most significant
  - Magnitude must exceed a certain threshold to be retained (retained coefficients thus different from block to block)
  - Then processed through a variable quantization matrix based on human visual system's response to luminance and chrominance

Threshold Coding

- Several options for threshold coding
  - Global threshold - same threshold for all coefficients; if lower than the global threshold, output is 0
  - Highest-N - consider all coefficients, retain only the highest N ones
  - Normalization - all coefficients are normalized by some normalization matrix before thresholding

Threshold Coding

This is an example of DCT coefficient matrix:

$$\begin{bmatrix} -415 & -33 & -58 & 35 & 58 & -51 & -15 & -12 \\ 5 & -34 & 49 & 18 & 27 & 1 & -5 & 3 \\ -46 & 14 & 80 & -35 & -50 & 19 & 7 & -18 \\ -53 & 21 & 34 & -20 & 2 & 34 & 36 & 12 \\ 9 & -2 & 9 & -5 & -32 & -15 & 45 & 37 \\ -8 & 15 & -16 & 7 & -8 & 11 & 4 & 7 \\ 19 & -28 & -2 & -26 & -2 & 7 & -44 & -21 \\ 18 & 25 & -12 & -44 & 35 & 48 & -37 & -3 \end{bmatrix}$$

A common quantization matrix is:

$$\begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}$$

Dividing the DCT coefficient matrix element-wise with this quantization matrix, and rounding to integers results in:

$$\begin{bmatrix} -26 & -3 & -6 & 2 & 2 & -1 & 0 & 0 \\ 0 & -3 & 4 & 1 & 1 & 0 & 0 & 0 \\ -3 & 1 & 5 & -1 & -1 & 0 & 0 & 0 \\ -4 & 1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

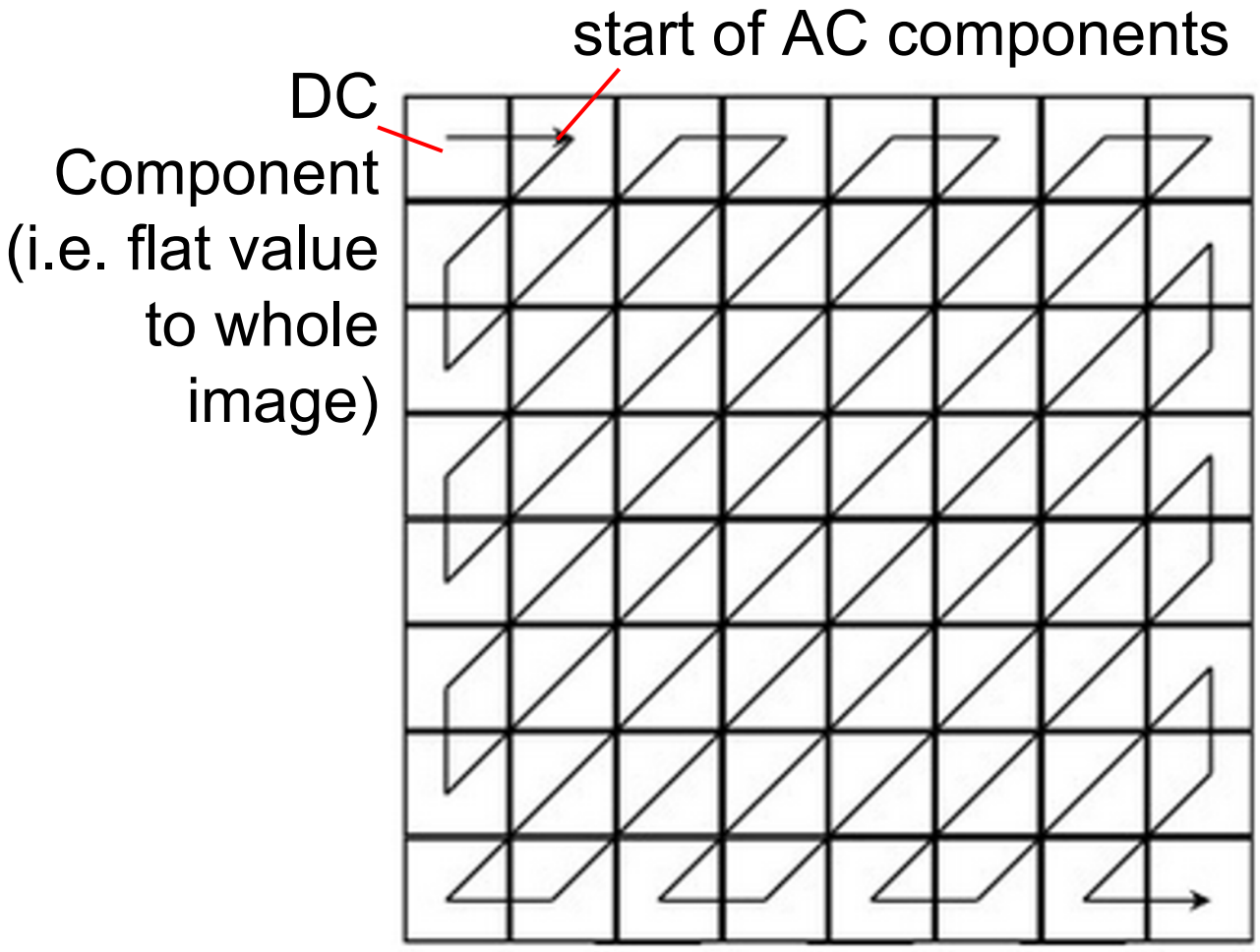
For example, using –415 (the DC coefficient) and rounding to the nearest integer

$$\operatorname{round}\left(\frac{-415}{16}\right) = \operatorname{round}\left(-25.9375\right) = -26$$

Threshold Coding

- Coefficients that are retained are few
  - How to encode with most efficiency?
- You may notice that retained transform coefficients tends to be at the top left corner
  - Zig-zag scanning with run-level encoding
- **Zig-zag scanning:** Starting from top-left corner and scan in a zig-zag manner
  - Most non-zeroes would be in the beginning of the scan (high-frequency coefficients much rarer)

Zig-Zag Scanning



Run-length encoding

- **Run-length encoding**: *Run-length* is the amount of zero encountered before a non-zero coefficient
  - Instead of encoding each coefficient one by one, we encoding **(run-length, level) pairs** where *level* is the first non-zero coefficient after the zeroes
  - Combined with threshold coding it's usually **(run-length, size, level)** where *size* is the number of bits used for representation of quantized coefficient level and *level* is the quantized coefficient
  - E.g. (5, 4, 3) would mean 5 zeroes, followed by a 4 bit quantized coefficient with value of 3

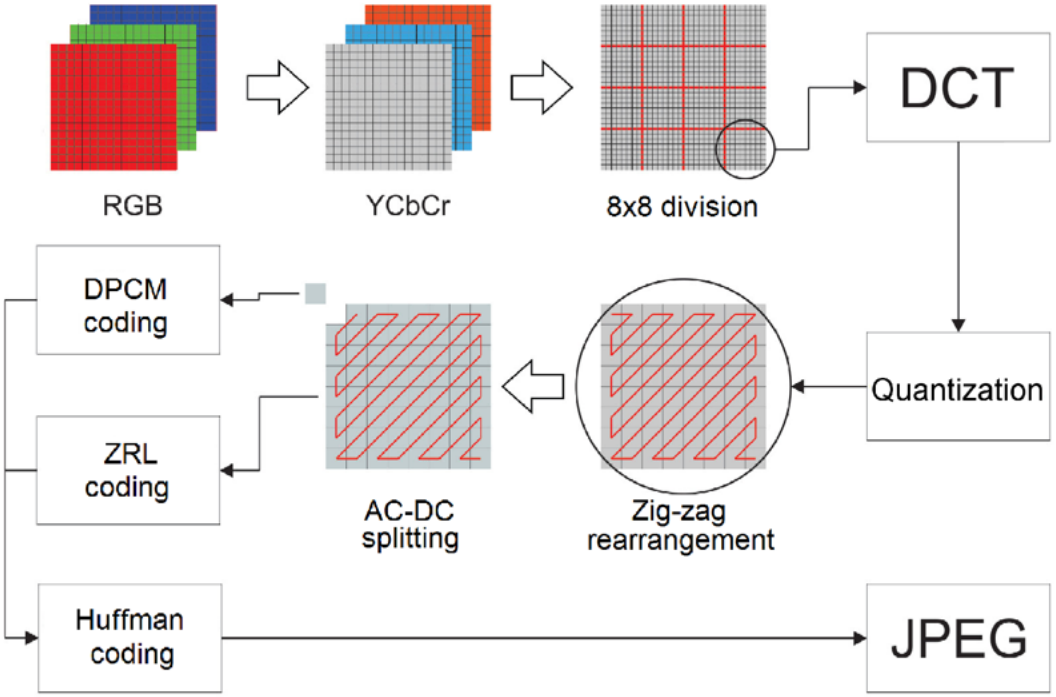
Run-length encoding

- Example:
  WWWWWWWWWWWWBWWWWWWWWW
  WWWWBBBWWWWWWWWWWWWWWW
  WWWWWWWWWWWBWWWWWWWWWW
  WWWW

Output: 12W1B12W3B24W1B14W

JPEG

- Joint Photographic Experts Group
- First standard issued 1992
  - Free code library *libjpeg* released in 1991



JPEG

- Obviously, an important image standard for Internet
  - Alongside with PNG as dominant Internet image coding standards
  - Similarly plagued by patent trolls, but JPEG has somewhat successfully defended the standard
- JPEG-2000, an improvement to JPEG, not widely used

JPEG

Highest quality

Q = 100

(2.6:1 compression)

JPEG

Medium quality

Q = 25

(23:1 compression)

JPEG

Low quality

Q = 10

(46:1 compression)

Video Coding

- Final achievement in multimedia coding
  - Combination of still images and sound (a lot of them)
  - Used in many appliances and devices (TV/HDTV, DVD, mobile etc.)
- Easily a course of its own - let's get to the basics
- Extra dimension: Time
  - Temporal redundancy not present in images

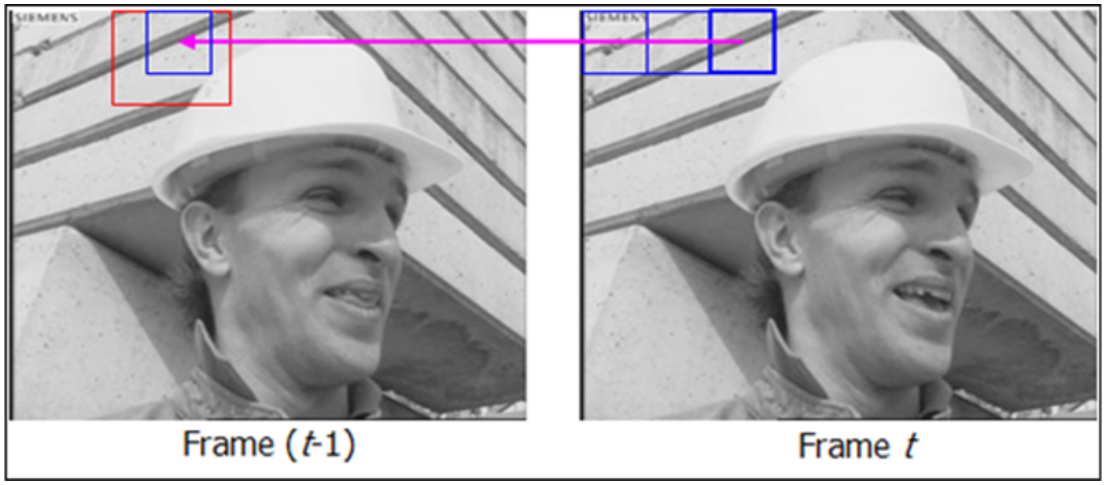
Video Coding

- First frame of video compressed as normal still image
  - **'Intraframe redundancy'**
- Second frame onward compressed exploiting two redundancies
  - Redundancy as an image
  - Redundancy with previous frames (**'Interframe redundancy'**)
- Why we need to exploit both redundancies?
  - Correct accumulated errors

Playback Artifacts

Motion Estimation

- Assume objects or parts of the objects in the scene has moved (translated)
  - Given short time between frames, also act as a rough estimation of scaling and rotation

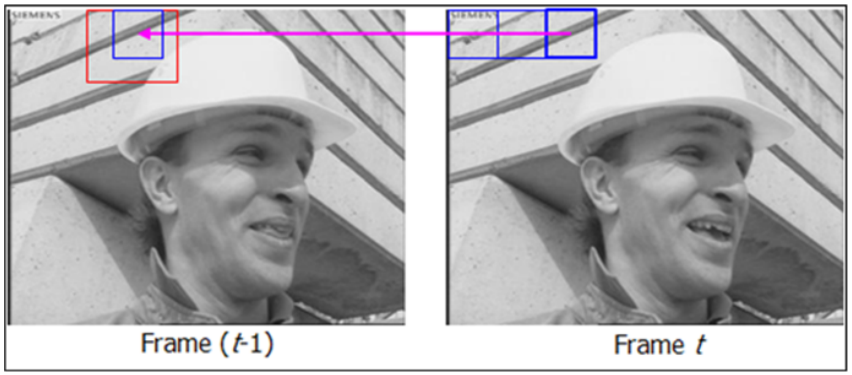


Motion Estimation

- Three possibilities:
  - **Backward estimation**: estimate motion change from last frame(s)
  - **Forward estimation:** estimation motion change from next frame(s)
    - It sounds weird but entirely possible in coding
  - **Bidirectional estimation:** estimate from both previous and subsequent frames

Motion Estimation

- Difficulties and concerns
  - Definitely cannot estimate by pixel



  - For each block we have to potentially "carpet sweep" the entire reference frame for a match
  - Also have to choose best match and avoid bad matches

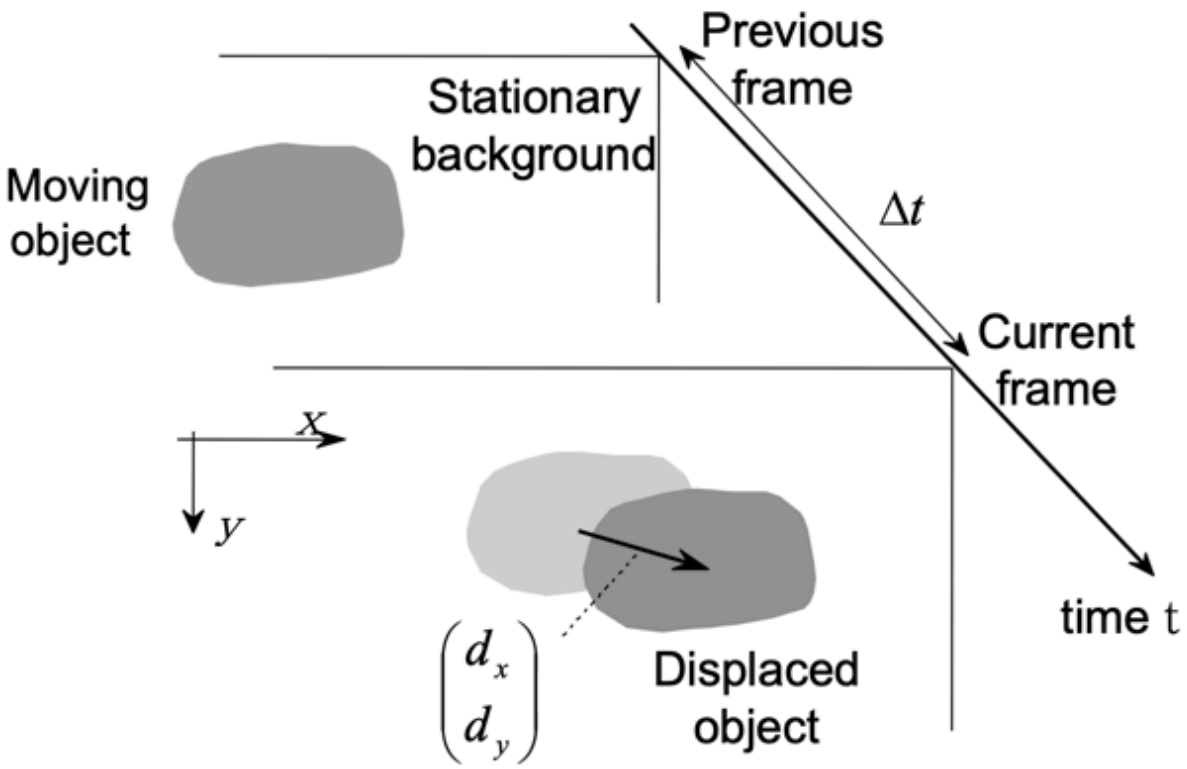
Motion Estimation

- Matching criteria
  - MSE (mean squared error)
  - MAD (mean absolute difference)
  - MPC (match pel count - number of pixels with difference within a threshold)
- Search algorithms
  - Full-search
  - Logarithmic search
    - Use tests locations and narrow down search window logarithmically
  - Many others...

Motion Estimation

- Motion vectors, the estimated motion (delta x and y) are encoded using DPCM
  - If you remember, DPCM predicts next sample using current value https://en.wikipedia.org/wiki/Differential_pulse-code_modulation
  - Motion vectors of adjacent blocks are usually similar
  - DPCM (2D) makes sense
- At the decoder, we reverse the process of motion estimation to reconstruct the frame
  - **"Motion compensation"**

Motion Compensation



Prediction for the luminance signal $s[x,y,t]$ within the moving object:

$$\hat{s}[x,y,t] = s'(x-d_x, y-d_y, t-\Delta t)$$

Motion Estimation

Frame 66

Frame 69

Absolute Difference w/o Motion Compensation

Absolute Difference with Motion Compensation

MPEG-1

- Three picture types
  - I-picture (keyframe)
    - Intraframe coded pictures
    - Coded with regular JPEG encoding
    - Support fast-forward, rewinding (random access) and error correction
  - P-picture
    - Interframe predicted pictures
    - Forward-predicted frame from the closest P-picture or I-picture
    - P-picture store only the *difference* in image
    - Much better compression than I-picture
    - Not for random access

MPEG-1

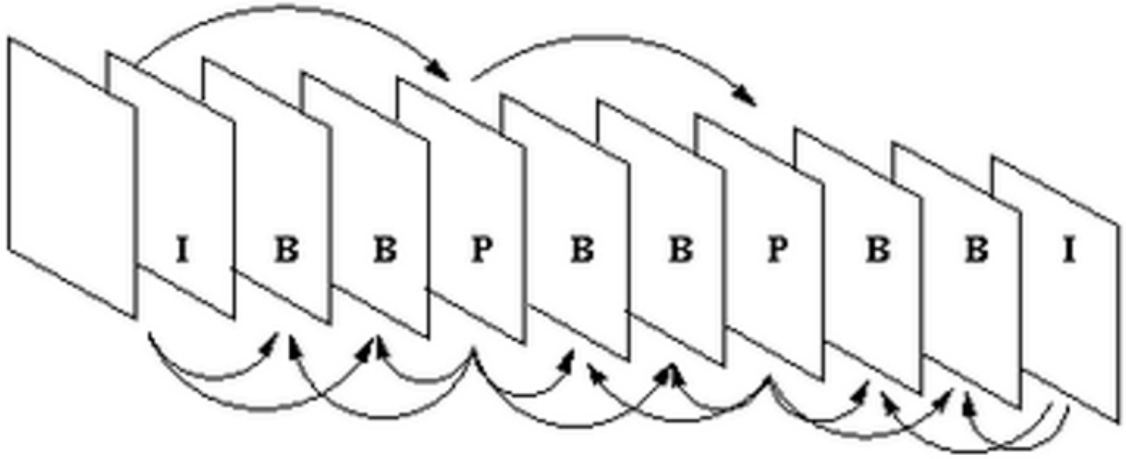
- Three picture types
  - B-picture
    - Bidirectionally predicted picture
    - Like P-picture, does not allow random access
    - Best compression performance
    - Most frames in video is B-picture
    - It is therefore necessary for the player to first decode the next I- or P- anchor frame sequentially after the B-frame, before the B-frame can be decoded and displayed

Stop and Think...

Question: Why P-picture and B-picture? Why not all B-picture?

(Try not to peek if you have printed out the notes)

MPEG-1



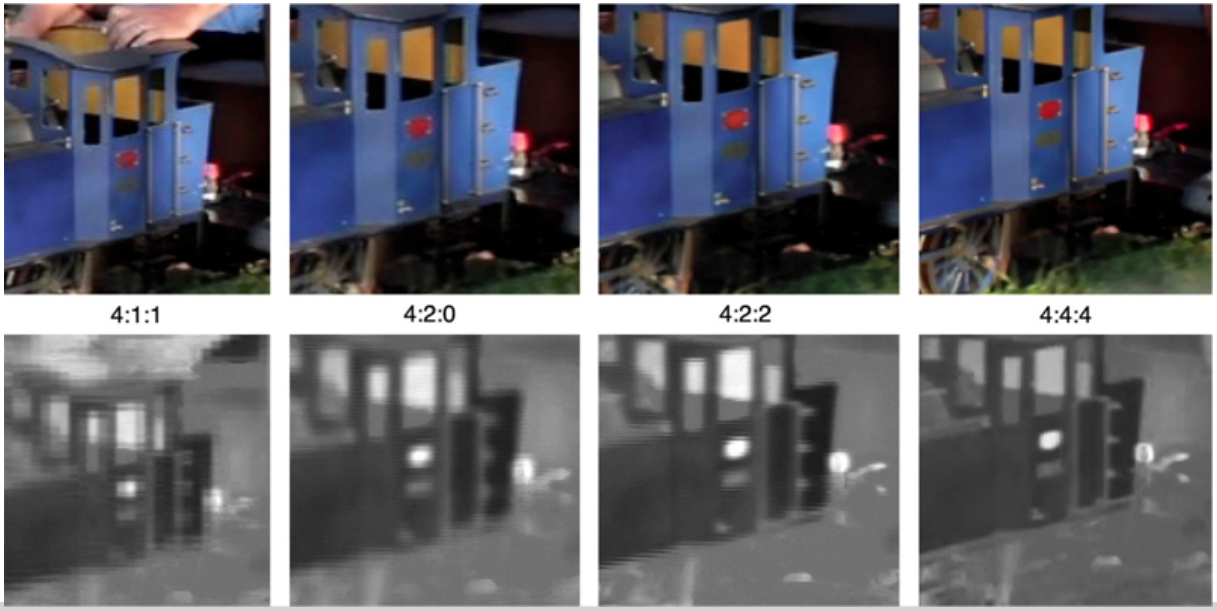
- P-picture based on previous I-picture or P-picture
- B-picture based on I-picture or P-picture around
- **No one depends on B-picture**

MPEG-1

- **Rationale:** If we want to reduce file size, we can reduce quality of B-pictures without causing problem
  - I-picture and P-picture, on the other hand, has to remain high-enough quality
- Fast-forward and rewinding to I-pictures only

MPEG-1

- Macroblocks
  - Chroma subsampling (as discussed): human visual system's lower acuity for color differences than for luminance.
  - Code color information with lower resolution

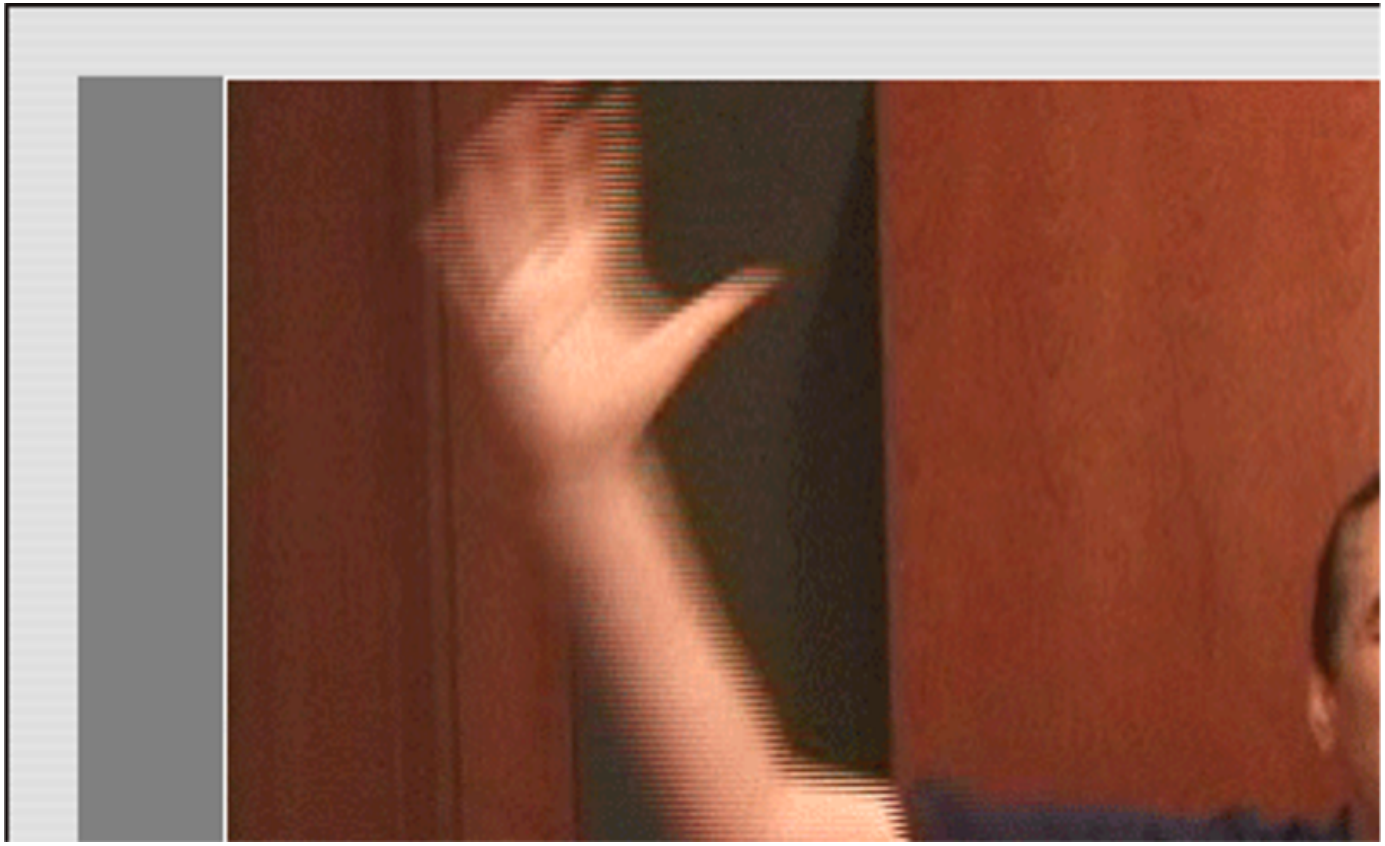


MPEG-2

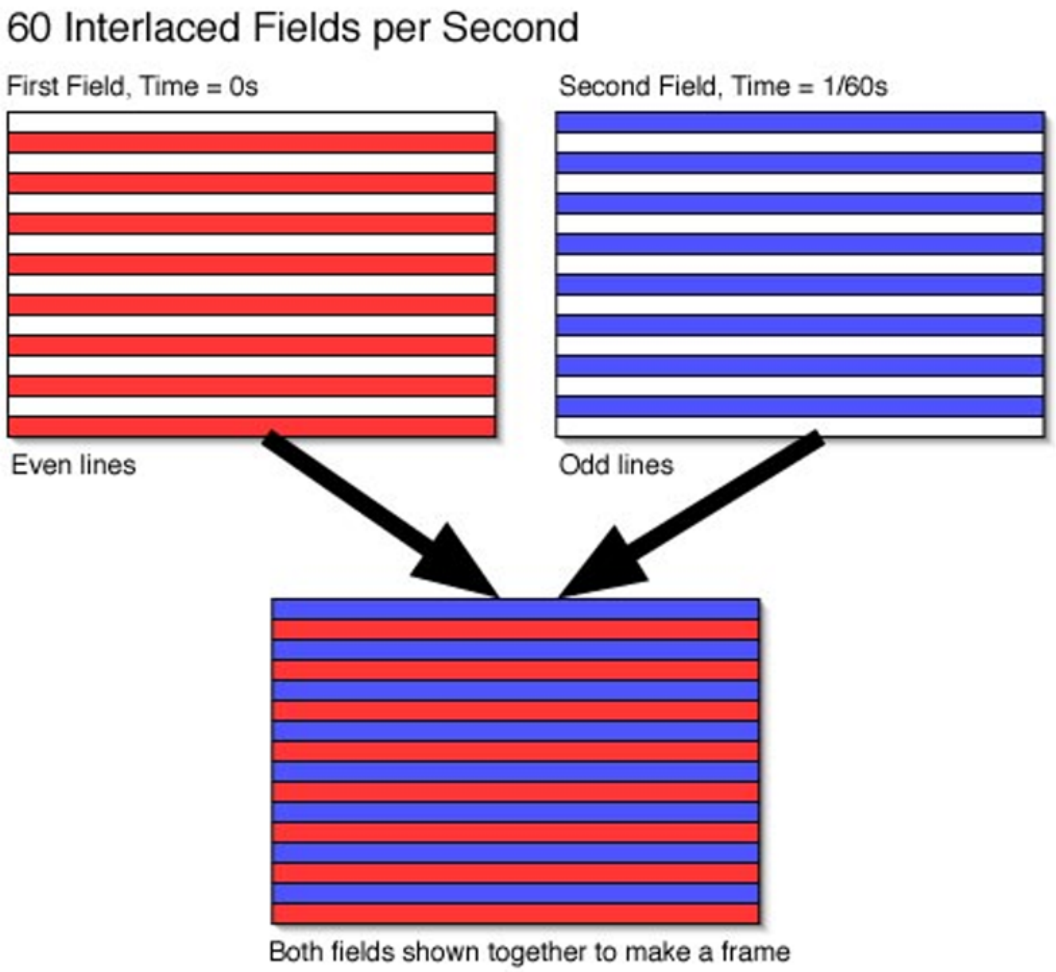
- Support for interlaced video
- **Interlacing:** a technique for doubling the perceived framerate of a video without extra bandwidth (wikipedia)
- **Fields:** odd/even lines of a frame
- Each time, odd fields are displayed first, then even fields are filled up, then odd fields…
  - Effectively doubles the frame-rate without additional frames

Interlacing

Interlacing



Interlacing

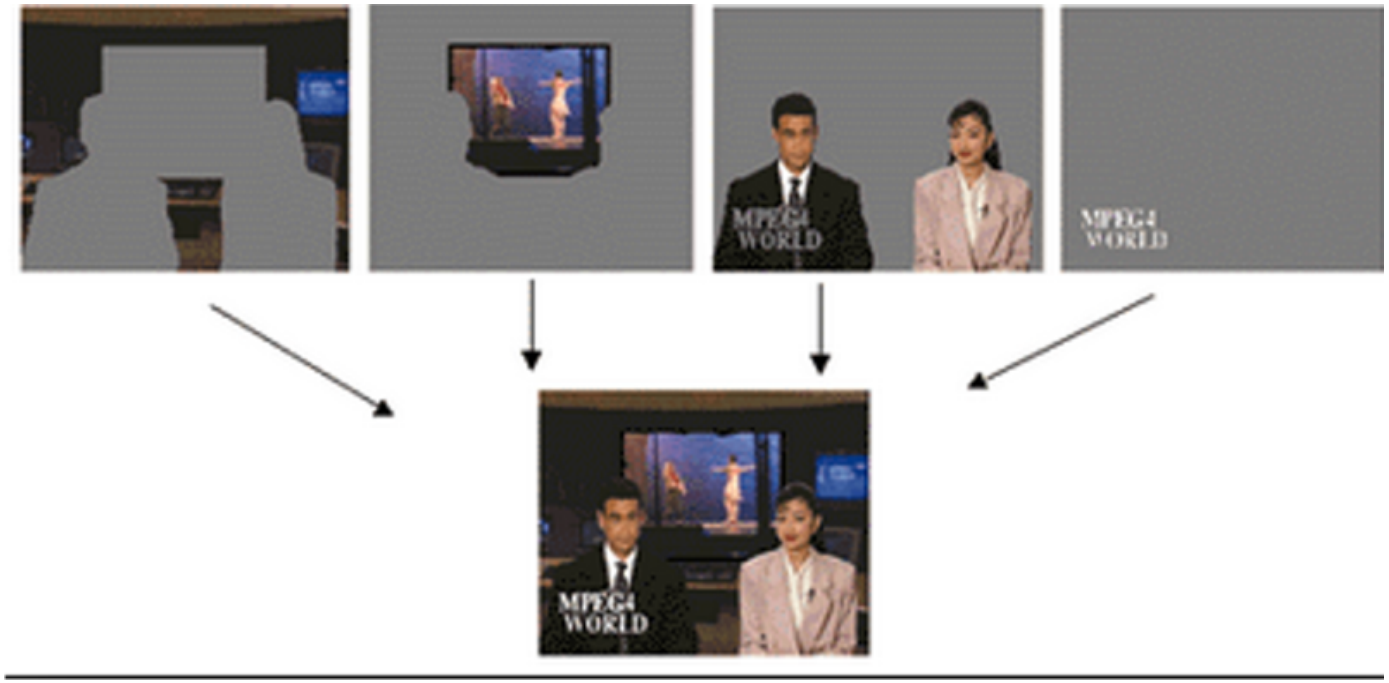


source: http://www.kenstone.net/fcp_homepage/24p_in_FCP_nattress.html

MPEG-4

- More ambitious and deviates significantly from previous standards
- Still evolving since 2000
  - Large number of "parts" (31 parts at the moment, much more than MPEG-1 and MPEG-2)
  - Large amount of features - up to developers to decide what to implement and what not via "Profiles" and "Levels"
    - For different formats, appliances and uses, e.g. Blu-ray v.s. web video
    - "Family of standards" - you don't just implement MPEG-4

Video Objects



Video Objects

- Different video objects (VO) are encoded and decoded individually
  - May use different encoding methods as they serve different purposes
  - Three things to encode: **shape, texture, motion**
  - May cover in our "Advanced Topics"

MPEG-4

- MPEG-4 Part 2
  - DCT-based, most similar to MPEG-2
  - DivX, Quicktime, H.263 compatible
- MPEG-4 Part 4
  - Jointly developed by VCEG in the name of Joint Video Team (JVT)
  - Also called H.264
  - Blu-ray, YouTube, iTunes, HDTV Broadcasts
  - Allows both lossy and lossless compression
  - Variable block size, more potent motion estimation and compression in general