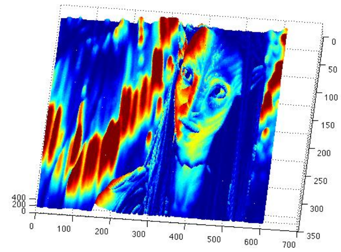
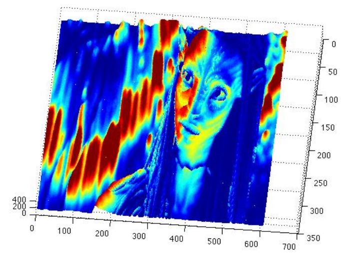
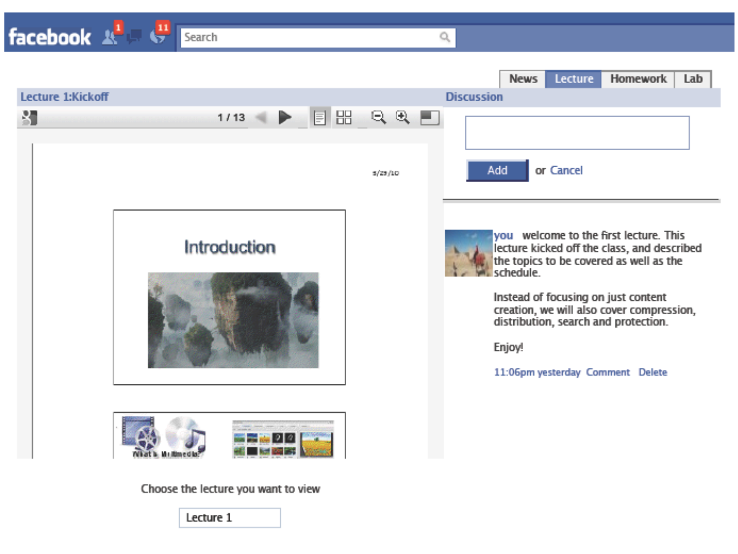




MEDIA FORMATS

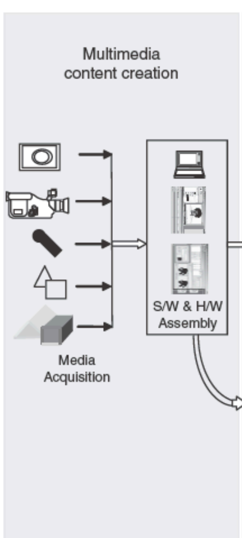
Some slides shamelessly taken from Yao Wang's Lecture Slides

- http://apps.facebook.com/classjournal/
- Register using your last name (low-case) and perm #

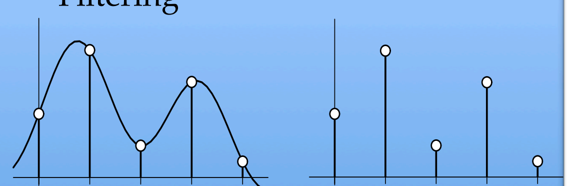


RECAP

A Multimedia System



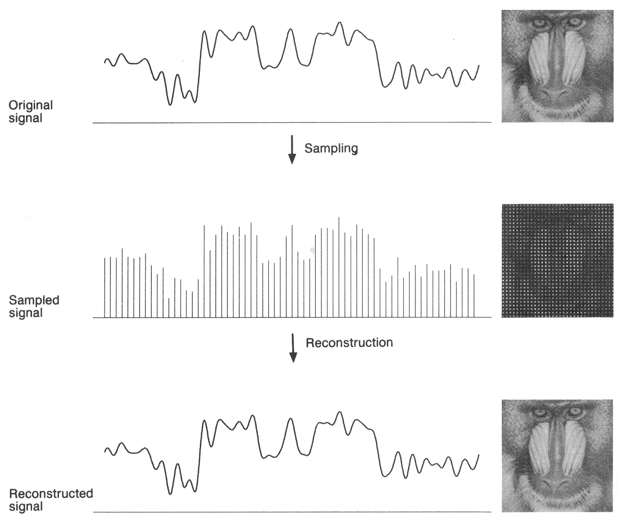
- Digital Data Acquisition
  - Analog-to-digital conversion
  - Sampling
  - Filtering



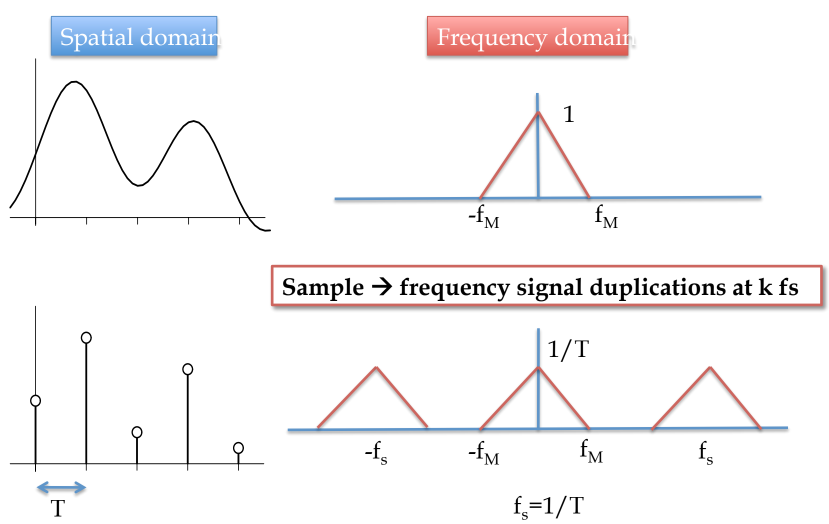
- Formatting
  - Image, video, audio
  - Adding colors

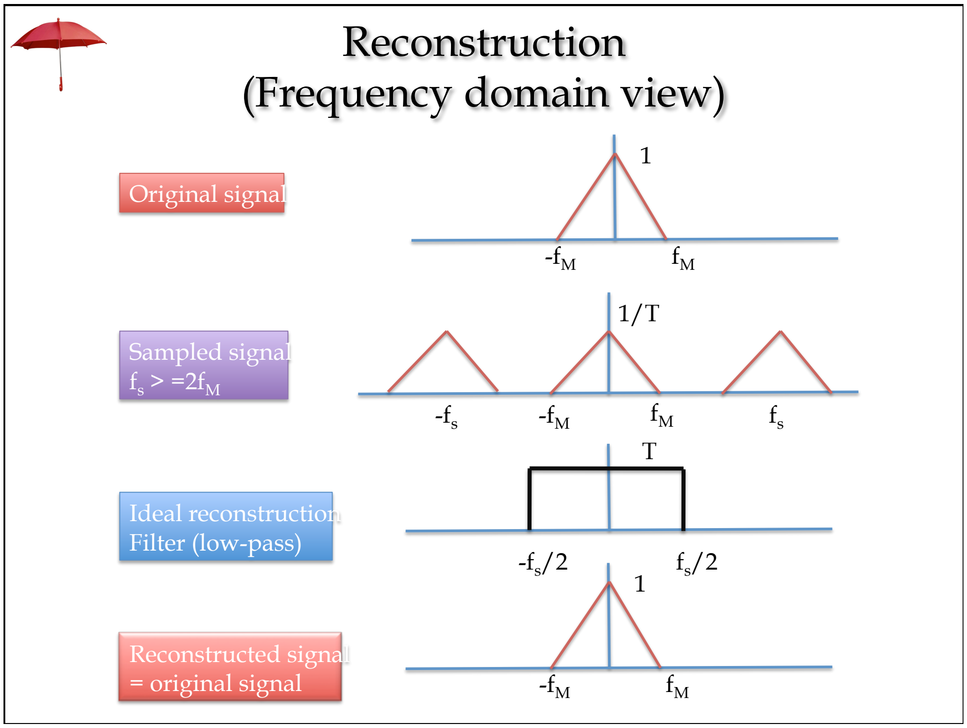
Figure 1-2 Components of a multimedia system today

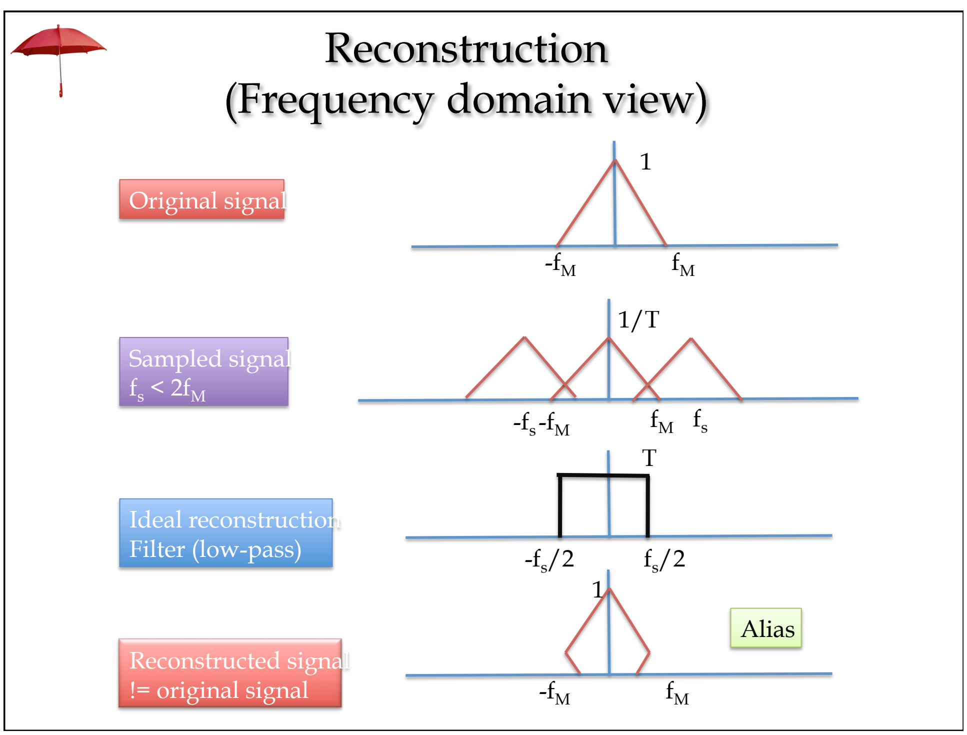
Sampling & Reconstruction

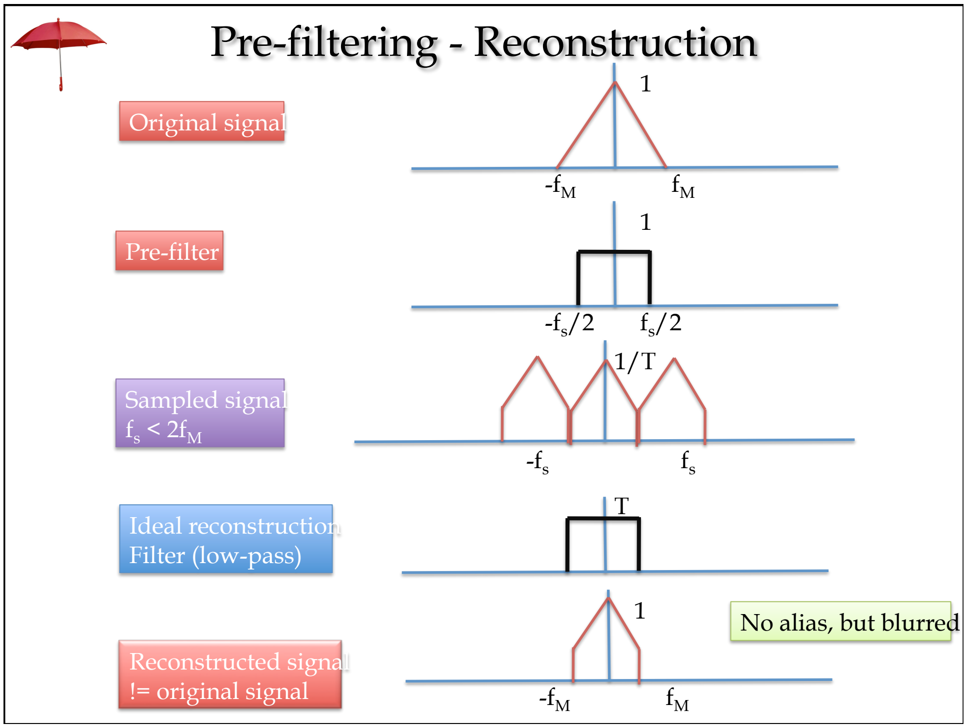


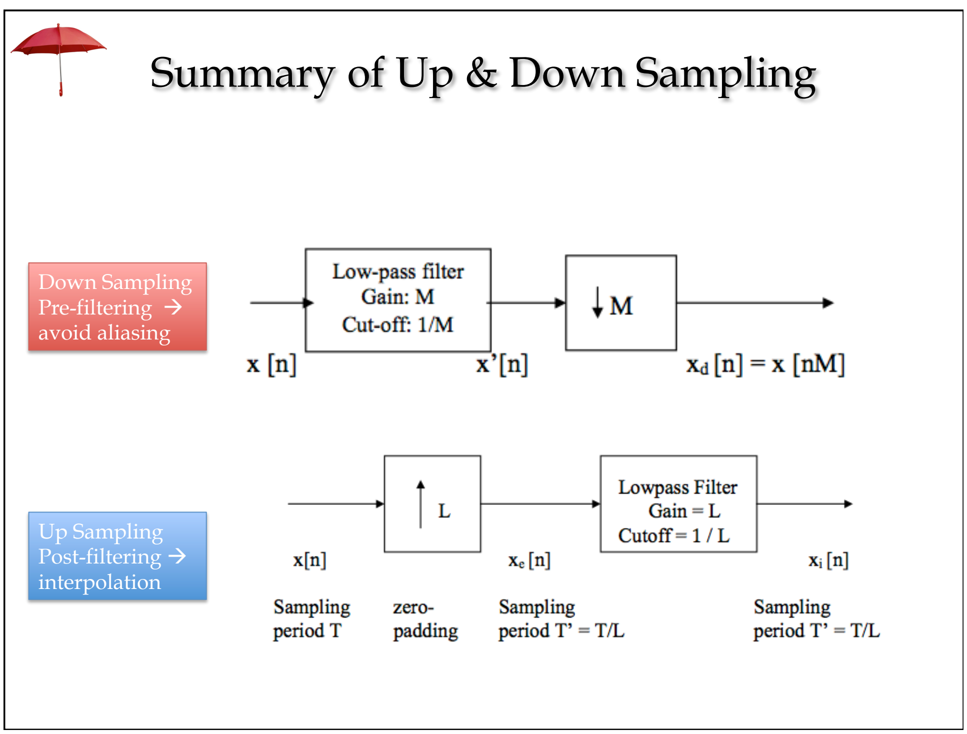
Before and After Sampling




Reconstruction
(Frequency domain view)
Original signal
1
-fM
fM
Sampled signal
fs > =2fM
1/T
-fs
-fM
fM
fs
Ideal reconstruction
Filter (low-pass)
T
-fs/2
fs/2
Reconstructed signal
= original signal
1
-fM
fM


Reconstruction
(Frequency domain view)
Original signal
1
-fM
fM
Sampled signal
fs < 2fM
1/T
-fs -fM
fM fs
Ideal reconstruction
Filter (low-pass)
T
-fs/2
fs/2
Reconstructed signal
!= original signal
1
Alias
-fM
fM


Pre-filtering - Reconstruction
Original signal
1
-fM
fM
Pre-filter
1
-fs/2
fs/2
Sampled signal
fs < 2fM
1/T
-fs
fs
Ideal reconstruction
Filter (low-pass)
T
Reconstructed signal
!= original signal
1
-fM
fM
No alias, but blurred


Summary of Up & Down Sampling
Down Sampling
Pre-filtering →
avoid aliasing
Low-pass filter
Gain: M
Cut-off: 1/M
↓M
x [n]
x'[n]
xd [n] = x [nM]
Up Sampling
Post-filtering →
interpolation
↑ L
Lowpass Filter
Gain = L
Cutoff = 1 / L
x[n]
xe [n]
xi [n]
Sampling
period T
zero-
padding
Sampling
period T' = T/L
Sampling
period T' = T/L

Summary

- Sampling:
  - Derive the minimally required sampling rate:
    - $f_s > 2f_{max}$,$Ts < T_{min}/2$
    - Can estimate T_{min} from signal waveform
  - Can plot the spectrum of a sampled signal
    - The sampled signal spectrum contains the original spectrum and its replicas (aliases) at k f_s, k=+/- 1,2,....
    - Can determine whether the sampled signal suffers from aliasing
  - Understand why do we need a prefilter when sampling a signal
    - To avoid alising
    - Ideally, the filter should be a lowpass filter with cutoff frequency at $f_s/2$.
    - Can show the aliasing phenomenon

Summary

- Interpolation
  - Can illustrate sample-and-hold and linear interpolation from samples.
  - Understand why the ideal interpolation filter is a lowpass filter with cutoff frequency at fs /2.
- Sampling Rate Conversion:
  - Know the meaning of down-sampling and upsampling
  - Understand the need for prefiltering before down-sampling
    - To avoid aliasing
    - Know how to apply simple averaging filter for downsampling
  - Can illustrate up-sampling by sample-and-hold and linear interpolation
- Filtering concept
  - Know how to apply filtering in the frequency domain
  - Can interpret the function of a filter based on its frequency response

This Week

- Media Formats (today and Wed)
- More about Colors (Wed)

IMAGE

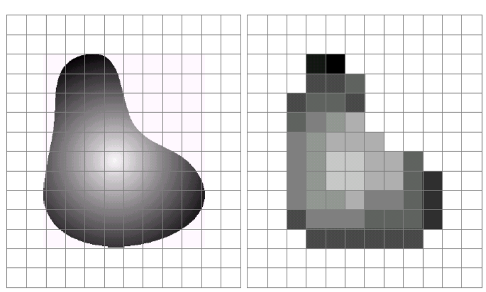
Definition of An Image

- Think an **image** as a function, *f*
  - *f* (*x, y*) gives the **intensity** at position (*x, y*)
  - Realistically, we expect the image only to be defined over a rectangle, with a finite range:
    - f: $[a,b]\times[c,d] \rightarrow [0,1]$
- A color image is just three functions pasted together
  - (R, G, B) components

$$f(x,y) = \begin{bmatrix} r(x,y) \\ g(x,y) \\ b(x,y) \end{bmatrix}$$

Sampling + Quantization

- We usually operate on **digital** (**discrete**) images:
  - **Sample** the 2D space on a regular grid → **Pixel**
  - **Quantize** each sample (round to nearest integer)



a b

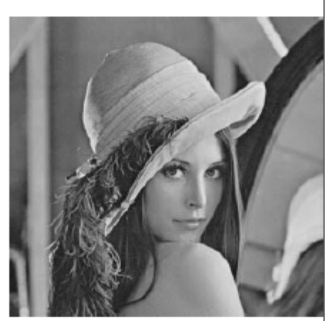
FIGURE 2.17 (a) Continuos image projected onto a sensor array. (b) Result of image sampling and quantization.

1-bit Image

- Each pixel is stored as a single bit (0 or 1), so also referred to as **binary image**.
- Such an image is also called a 1-bit **monochrome** image or a pure black/ white image since it contains no color.
- We show a sample 1-bit monochrome image "Lena"
  - A standard image used to illustrate many algorithms

8-bit Grayscale Image

- Each pixel has a gray-value between 0 and 255.
  - A dark pixel might have a value of 10, and a bright one might be 230.
- Each pixel is represented by a single byte;
- **Image resolution** refers to the number of pixels in a digital image
  - Higher resolution always yields better quality
  - Fairly high resolution for such an image might be 1600x1200, whereas lower resolution might be 640x480.
- Without any compression, a raw image's size = # of pixels x byte per pixel



24-bit Colored Image

- Each pixel is represented by **three bytes**, usually representing RGB
  - one byte for each R, G, B component
  - 256x256x256 possible combined colors, or a total of 16,777,216 possible colors.
  - However such flexibility does result in a storage penalty: A 640x480 24-bit color image would require 921.6 kB of storage without any compression.

x=imread('Filename')

imshow(x)

image(x)

imwrite('Filename',x)

y=rgb2gray(x)

PROCESSING IMAGES IN MATLAB

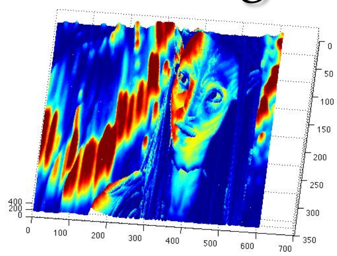
Read An Image From A File

- A = imread(filename, *fmt*)
  - Read a grayscale or color image from the file specified by the string filename.
  - The text string fmt specifies the format of the file by its standard file extension. For example, specify 'gif' for Graphics Interchange Format files. If imread cannot find a file named filename, it looks for a file named filename.fmt.
  - The return value A is an array containing the image data.
    - If the file contains a grayscale image, A is an M-by-N array.
    - If the file contains a truecolor image, A is an M-by-N-by-3 array.
    - For TIFF files containing color images that use the CMYK color space, A is an M-by-N-by-4 array.

Matlab Example #1

```
I=imread('avatar1.jpg');
size(I)
imshow(I);
I(1:10,1:10,:)
J=double(I);  %convert I from integer to double format
mesh(J(:,:,1))
colormap(Gray)
```

Colored Image



Blue-layer Pixel Strength

ans(:,:,1) =

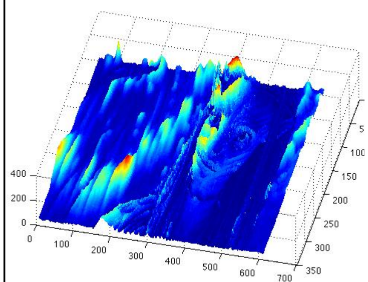
17	16	14	15	17	19	19	18	13	12
17	15	14	14	16	17	17	16	14	13
19	18	16	16	16	17	17	16	15	14
21	20	19	18	18	17	17	16	16	15
17	17	16	15	16	14	14	13	15	14
16	17	17	17	17	15	15	15	14	13
14	16	17	17	16	15	15	15	13	12
11	13	15	15	14	13	13	14	12	11
15	14	14	13	13	11	11	10	13	11
16	16	15	15	15	14	14	14	10	8

ans(:,:,2) =

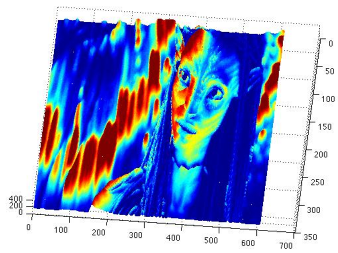
10	9	7	8	10	12	11	10	5	4
10	8	7	7	9	10	9	8	6	5
14	13	11	11	11	12	9	8	7	6
16	15	14	13	13	12	9	8	8	7
14	14	13	12	10	8	8	7	9	8
13	14	14	14	11	9	9	9	8	7
13	15	16	16	13	12	12	12	7	6
10	12	14	14	11	10	10	11	6	5
15	14	14	13	11	9	8	7	10	8
16	16	15	15	13	12	11	11	7	5

ans(:,:,3) =

17	16	14	15	18	20	22	21	18	17
17	15	14	14	17	18	20	19	19	18
20	19	17	17	18	19	20	19	20	19
22	21	20	19	20	19	20	19	21	20
21	21	20	19	20	18	18	17	21	20
20	21	21	21	21	19	19	19	20	19
21	23	24	24	22	21	23	23	19	18
18	20	22	22	20	19	21	22	18	17
25	24	24	23	22	20	19	18	21	19
26	26	25	25	24	23	22	22	18	16



Red-layer Pixel Strength



Green-layer Pixel Strength

Matlab Example #2

```
W=I;
W(1:2:end,1:2:end,:)=0;
figure; imshow(W);
R=W;
R(1:2:end,1:2:end,:)=W(2:2:end,2:2:end,:);
figure; imshow(R);
```

original image first 10x10

ans(:,:,1) =

17	16	14	15	17	19	19	18	13	12
17	15	14	14	16	17	17	16	14	13
19	18	16	16	16	17	17	16	15	14
21	20	19	18	18	17	17	16	16	15
17	17	16	15	16	14	14	13	15	14
16	17	17	17	17	15	15	15	14	13
14	16	17	17	16	15	15	15	13	12
11	13	15	15	14	13	13	14	12	11
15	14	14	13	13	11	11	10	13	11
16	16	15	15	15	14	14	14	10	8

ans(:,:,2) =

10	9	7	8	10	12	11	10	5	4
10	8	7	7	9	10	9	8	6	5
14	13	11	11	11	12	9	8	7	6
16	15	14	13	13	12	9	8	8	7
14	14	13	12	10	8	8	7	9	8
13	14	14	14	11	9	9	9	8	7
13	15	16	16	13	12	12	12	7	6
10	12	14	14	11	10	10	11	6	5
15	14	14	13	11	9	8	7	10	8
16	16	15	15	13	12	11	11	7	5

ans(:,:,3) =

17	16	14	15	18	20	22	21	18	17
17	15	14	14	17	18	20	19	19	18
20	19	17	17	18	19	20	19	20	19
22	21	20	19	20	19	20	19	21	20
21	21	20	19	20	18	18	17	21	20
20	21	21	21	21	19	19	19	20	19
21	23	24	24	22	21	23	23	19	18
18	20	22	22	20	19	21	22	18	17
25	24	24	23	22	20	19	18	21	19
26	26	25	25	24	23	22	22	18	16

W=original image
W(odd x, odd y, 1:3)=0

ans(:,:,1) =

0	16	0	15	0	19	0	18	0	12
17	15	14	14	16	17	17	16	14	13
0	18	0	16	0	17	0	16	0	14
21	20	19	18	18	17	17	16	16	15
0	17	0	15	0	14	0	13	0	14
16	17	17	17	17	15	15	15	14	13
0	16	0	17	0	15	0	15	0	12
11	13	15	15	14	13	13	14	12	11
0	14	0	13	0	11	0	10	0	11
16	16	15	15	15	14	14	14	10	8

ans(:,:,2) =

0	9	0	8	0	12	0	10	0	4
10	8	7	7	9	10	9	8	6	5
0	13	0	11	0	12	0	8	0	6
16	15	14	13	13	12	9	8	8	7
0	14	0	12	0	8	0	7	0	8
13	14	14	14	11	9	9	9	8	7
0	15	0	16	0	12	0	12	0	6
10	12	14	14	11	10	10	11	6	5
0	14	0	13	0	9	0	7	0	8
16	16	15	15	13	12	11	11	7	5

ans(:,:,3) =

0	16	0	15	0	20	0	21	0	17
17	15	14	14	17	18	20	19	19	18
0	19	0	17	0	19	0	19	0	19
22	21	20	19	20	19	20	19	21	20
0	21	0	19	0	18	0	17	0	20
20	21	21	21	21	19	19	19	20	19
0	23	0	24	0	21	0	23	0	18
18	20	22	22	20	19	21	22	18	17
0	24	0	23	0	20	0	18	0	19
26	26	25	25	24	23	22	22	18	16

W(odd x, odd y, 1:3)=
W(even x, even y, 1:3)

ans(:,:,1) =

15	16	14	15	17	19	16	18	13	12
17	15	14	14	16	17	17	16	14	13
20	18	18	16	17	17	16	16	15	14
21	20	19	18	18	17	17	16	16	15
17	17	17	15	15	14	15	13	13	14
16	17	17	17	17	15	15	15	14	13
13	16	15	17	13	15	14	15	11	12
11	13	15	15	14	13	13	14	12	11
16	14	15	13	14	11	14	10	8	11
16	16	15	15	15	14	14	14	10	8

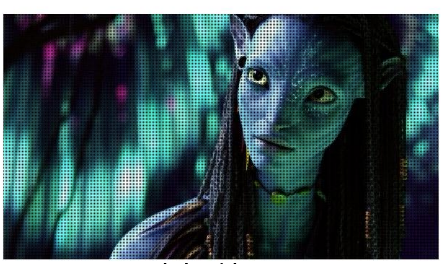
ans(:,:,2) =

8	9	7	8	10	12	8	10	5	4
10	8	7	7	9	10	9	8	6	5
15	13	13	11	12	12	8	8	7	6
16	15	14	13	13	12	9	8	8	7
14	14	14	12	9	8	9	7	7	8
13	14	14	14	11	9	9	9	8	7
12	15	14	16	10	12	11	12	5	6
10	12	14	14	11	10	10	11	6	5
16	14	15	13	12	9	11	7	5	8
16	16	15	15	13	12	11	11	7	5

ans(:,:,3) =

15	16	14	15	18	20	19	21	18	17
17	15	14	14	17	18	20	19	19	18
21	19	19	17	19	19	19	19	20	19
22	21	20	19	20	19	20	19	21	20
21	21	21	19	19	18	19	17	19	20
20	21	21	21	21	19	19	19	20	19
20	23	22	24	19	21	22	23	17	18
18	20	22	22	20	19	21	22	18	17
26	24	25	23	23	20	22	18	16	19
26	26	25	25	24	23	22	22	18	16

Sampling+ Reconstruction



W=original image
W(odd x, odd y, 1:3)=0

W(odd x, odd y, 1:3)=W(even x, even y, 1:3)

Write An Image To A File

- imwrite(A,filename,*fmt*) writes the image A to the file specified by filename in the format specified by *fmt*.
  - A can be an M-by-N (grayscale image) or M-by-N-by-3 (truecolor image) array, but it cannot be an empty array.
  - For TIFF files, A can be an M-by-N-by-4 array containing color data that uses the CMYK color space.
  - filename is a string that specifies the name of the output file.
  - fmt can be any of the text strings listed in the table in Supported Image Types.

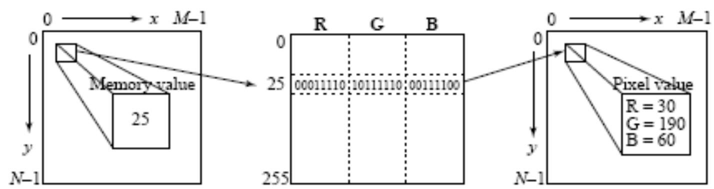
Converting Color to Grayscale

```
I=imread('avatar1.jpg');
J = rgb2gray(I);
figure, imshow(I);
figure, imshow(J);
size(I)
size(J)
J(1:10,1:10)
```

RGB2GRAY converts RGB images to grayscale by eliminating the hue and saturation information while retaining the luminance.

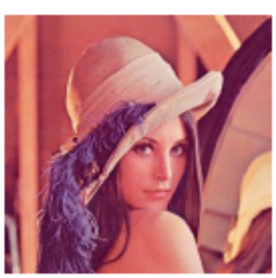
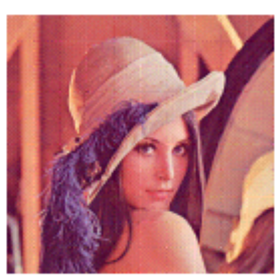
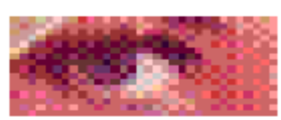
8-bit Colored Image

- Many systems can make use of 8 bits of color information (the so-called "**256 colors**") in producing a screen image.
- Such images use the concept of a **lookup table** to store color information.
  - Basically, the image stores not color, but instead just a set of single bytes, each of which is actually an index into a table of 3-byte values that specify the pixel color for that lookup table index.



Color Lookup Table

- A 24-bit color image of "Lena", and the same image reduced to only 5 bits via color lookup and dithering. A detail of the left eye is shown.

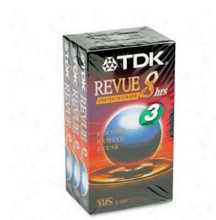
VIDEO

You Tube
Broadcast Yourself

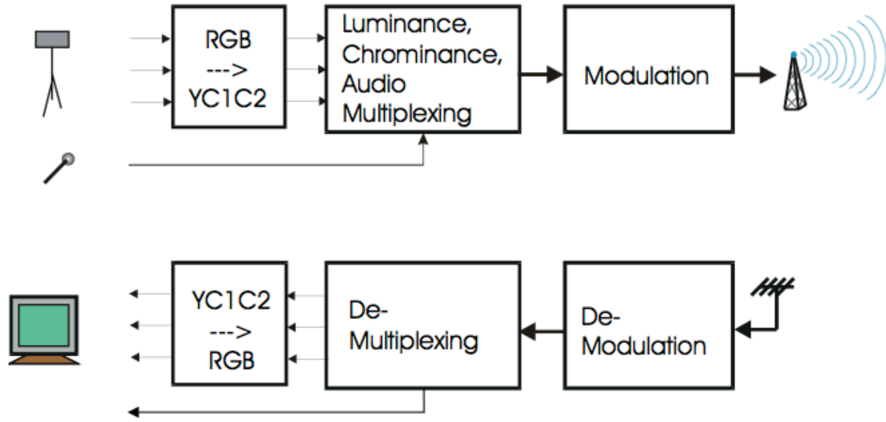
http://www.technobuzz.net/wp-content/uploads/2009/09/blog-iphone-video.jpg

Video: Roadmap

- Video= "I see" in Latin
- Digital
  - Blu-ray disc, DVD, QuickTime, MPEG-4
  - MPEG-4 AFX (3D-video)
  - Cable TV
- Analog
  - VHS videotapes
  - Analog video/TV



Analog TV

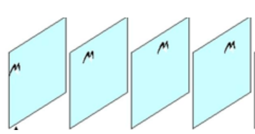


Video Raster

- Real-world scene is a continuously varying 3-D signal (temporal, horizontal, vertical)

- Analog video is captured and stored in the raster format
  - Sampling in time: consecutive sets of frames
  - Sampling in vertical direction: successive scan lines in one frame
  - Video-raster = **1-D signal** consisting of scan lines from successive frames

- Video is displayed in the raster form
  - Display successive frames
  - Display successive lines per frame
  - To enable the display to recognize the beginning of each frame and each line, special sync signals are inserted



Progressive vs. Interlaced Frames

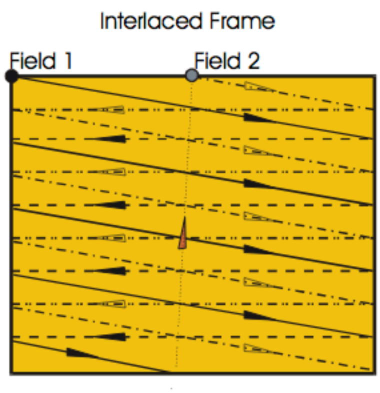
• Used in standard television formats (NTSC, PAL, and SECAM)

• Displays only half of the horizontal lines at a time

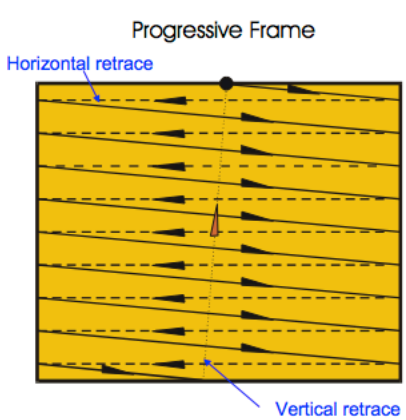
• The first *field* , containing the odd-numbered lines, is displayed, followed by the second field, containing the even-numbered lines

• Good: A high refresh rate (50 or 60 Hz) can be achieved with only half the bandwidth.

• Bad: The horizontal resolution is essentially cut in half.



Progressive vs. Interlaced Frames



• Used in CRT, LCD, DTV, HDTV
  • 720p, 1024p

• Displays all the horizontal lines at a time

• Good: Higher resolution at the same refresh rate, no blurring

• Bad: Higher bandwidth

Bandwidth of 1920x1080 (1080**i**60)
= Bandwidth of 1280x720(720**p**60)

Interlaced scan is developed to provide a trade-off between temporal and vertical resolution, for a given, fixed data rate (number of line/sec).

Interlaced Frames on Progressive TV without Compensation

How Many Lines?

- Ideally we want the rate to be as high as possible to get best possible quality
- But higher rates mean the capture and display devices must work with very high data rate, and transmission of TV signals would take significant amount of bandwidth
- Human eye does not perceive separate lines/frames when the rate is sufficiently high
- Use just enough frame/line rate at which the eye perceives a continuous video
  - 60Hz frame rate
  - 400-600 lines per frame

Connecting Frames in Time

- Persistence of vision: the eye (or the brain rather) can retain the sensation of an image for a short time even after the actual image is removed
  - Allows the display of a video as successive frames
  - As long as the frame interval is shorter than the persistence period, the eye sees a continuously varying image in time
  - When the frame interval is too long, the eye observes frame flicker
- The minimal frame rate (frames/second or fps or Hz) required to prevent frame flicker depends on display brightness, viewing distance.
  - For TV viewing: 50-60 fps
  - For Movie viewing: 24 fps
  - For computer monitor: > 70 fps

Digitizing a Raster Video

- Digitization = **Sampling + Quantization**
- Sample the raster waveform = Sample along the horizontal direction
- Apply the above sampling on Y,I,Q rasters separately
- Quantize Y,I,Q samples to 8 bits each
- How should we select the sampling rate?
  - Must be faster than the Nyquist rate (twice the highest frequency)
  - For the samples to be aligned vertically, the sampling rate should be multiples of the line rate
  - Horizontal sampling interval = vertical sampling interval (square pixel)
  - Total sampling rate equal among different systems (525/30 vs 625/25)
    - *fs = 858 fl (NTSC) = 864 fl (PAL/SECAM) =13.5 MHz*

Digital Video

Basic Metrics (from Wiki)

- **pixels per frame** = 640 * 480 = 307,200
- **bits per frame** = 307,200 * 24 = 7,372,800 = 7.37*Mbits*
- **bit rate (BR)** = 7.37 * 25 = 184.25***Mbits/sec***
- **1 hour video size (VS)** = 184*Mbits/sec* * 3600*sec* = 662,400*Mbits* = 82,800*Mbytes* = 82.8***Gbytes***

That is a lot of data!

Matlab Example

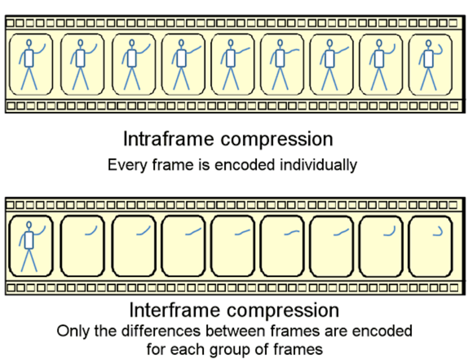
```
% load movie file in avi format
    mov = aviread('foreman.avi');
    imshow(mov(1).cdata)

% play the move
    movie(mov);
```

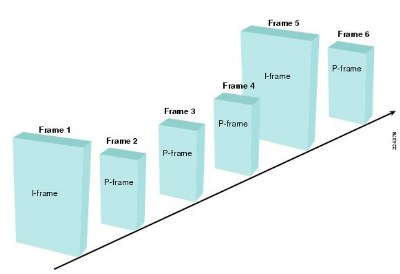
Audio Video Interleave (avi): Little or no compression

Digital Video Formats

Video Format	Y Size	Color Sampling	Frame Rate (Hz)	Raw Data Rate (Mbps)
HDTV Over air. cable, satellite, MPEG2 video, 20-45 Mbps				
SMPTE296M	1280x720	4:2:0	24P/30P/60P	265/332/664
SMPTE295M	1920x1080	4:2:0	24P/30P/60I	597/746/746
Video production, MPEG2, 15-50 Mbps				
BT.601	720x480/576	4:4:4	60I/50I	249
BT.601	720x480/576	4:2:2	60I/50I	166
High quality video distribution (DVD, SDTV), MPEG2, 4-10 Mbps				
BT.601	720x480/576	4:2:0	60I/50I	124
Intermediate quality video distribution (VCD, WWW), MPEG1, 1.5 Mbps				
SIF	352x240/288	4:2:0	30P/25P	30
Video conferencing over ISDN/Internet, H.261/H.263/MPEG4, 128-384 Kbps				
CIF	352x288	4:2:0	30P	37
Video telephony over wired/wireless modem, H.263/MPEG4, 20-64 Kbps				
QCIF	176x144	4:2:0	30P	9.1

Video Compression



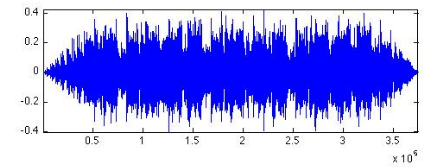


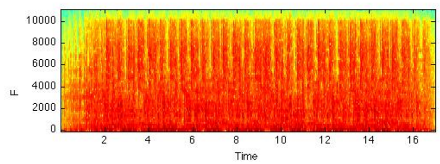
http://images.trustedreviews.com/images/article/inline/6317-Interintra.gif

http://www.cisco.com/en/US/i/200001-300000/220001-230000/224001-225000/224378.jpg

Digital Audio

- Audio= collection of waveforms
- Sampling + quantizatio





- Matlab demo
  - [audio,Fs]=wavread(file
  - sound(audio,Fs)
  - Can only read some .wav files

Things that you should know

- Image (binary, grayscale, color)
  - Can process them in matlab
- Temporal sampling → video frame rate
- Interlacing vs. progressive
  - NTSC, PAL use interlacing
  - Projectors, plasma TV: progressive scan
- Resolution: # of lines scanned
  - NTSC: 720/704/640 x 480**i**60
  - PAL: 768/720x576**i**50
  - HDTV: 1920x1080**p**60

Signup for Google group

http://groups.google.com/group/cs182ece160