Multimedia Networking

#4 Coding and Compression

Semester Ganjil 2012

PTIIK Universitas Brawijaya

Schedule of Class Meeting

1. Introduction
2. Applications of MN
3. Requirements of MN
4. **Coding and Compression**
5. RTP
6. IP Multicast
7. IP Multicast (cont'd)
8. Overlay Multicast
9. CDN: Solutions
10. CDN: Case Studies
11. QoS on the Internet: Constraints
12. QoS on the Internet: Solutions
13. Discussion
14. Summary

Today's Outline

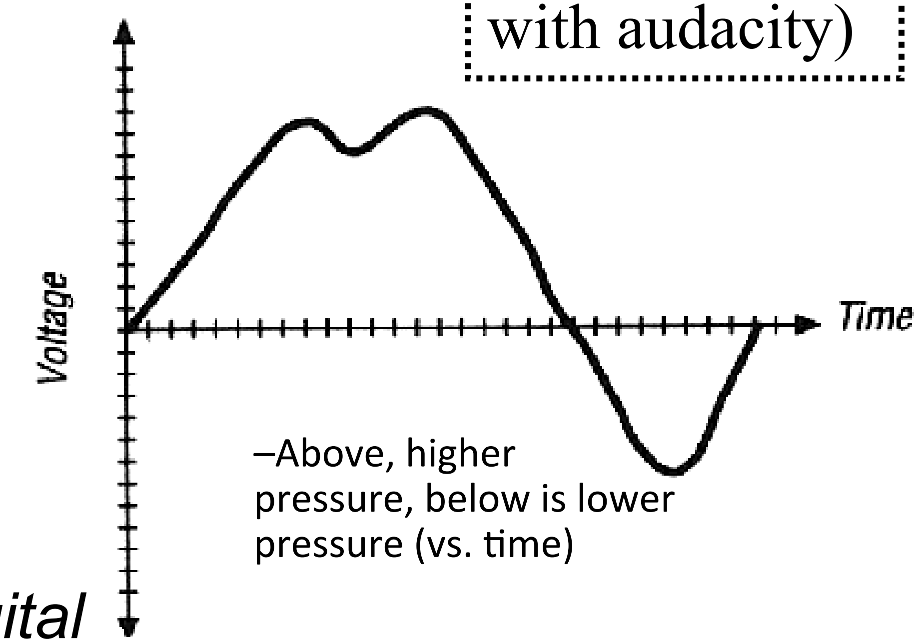
Coding and Compression

- **Digital Audio**
- Digital Video

Digital Audio

- Sound produced by variations in air pressure
  - Can take any continuous value
  - *Analog* component

(Show samples with audacity)



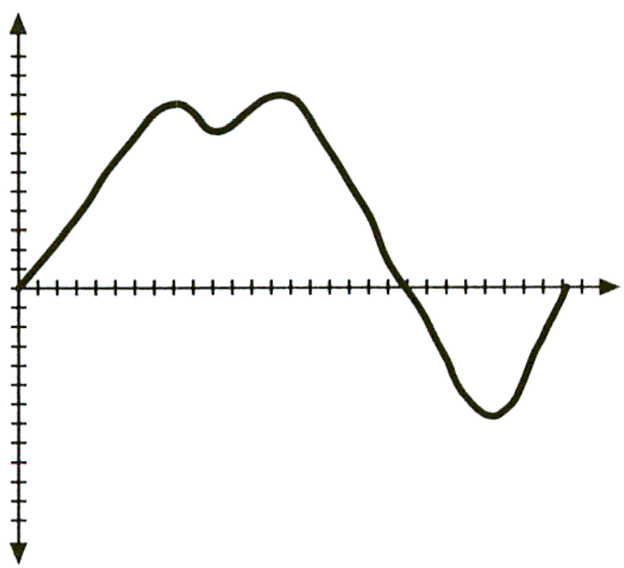
–Above, higher pressure, below is lower pressure (vs. time)

- Computers work with *digital*
  - Must convert analog to digital
  - Use *sampling* to get discrete values

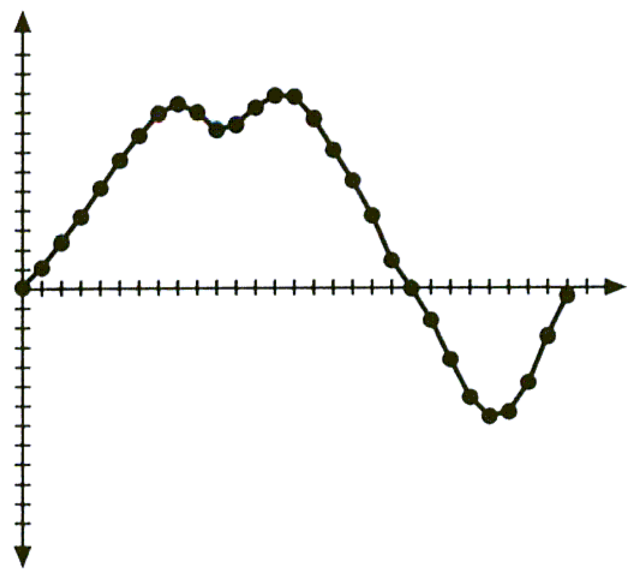
Digital Sampling

- *Sample rate* determines number of discrete values

a. *Original Analog Waveform*

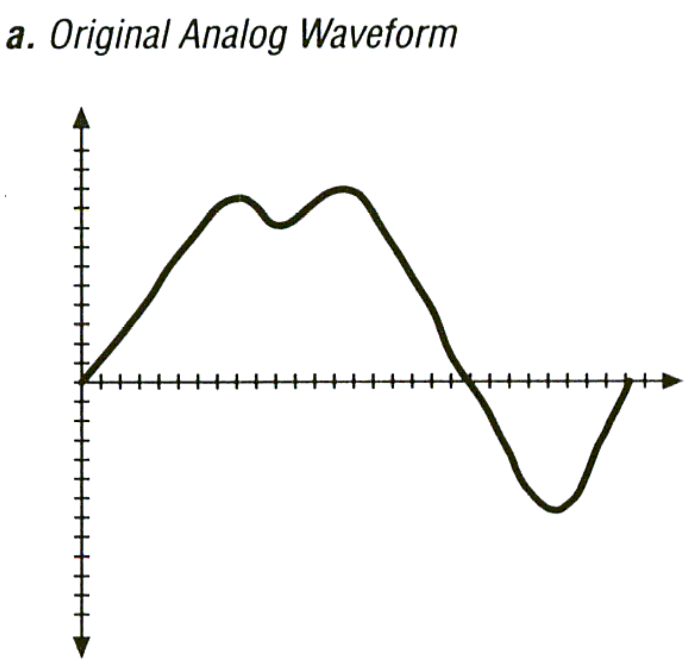


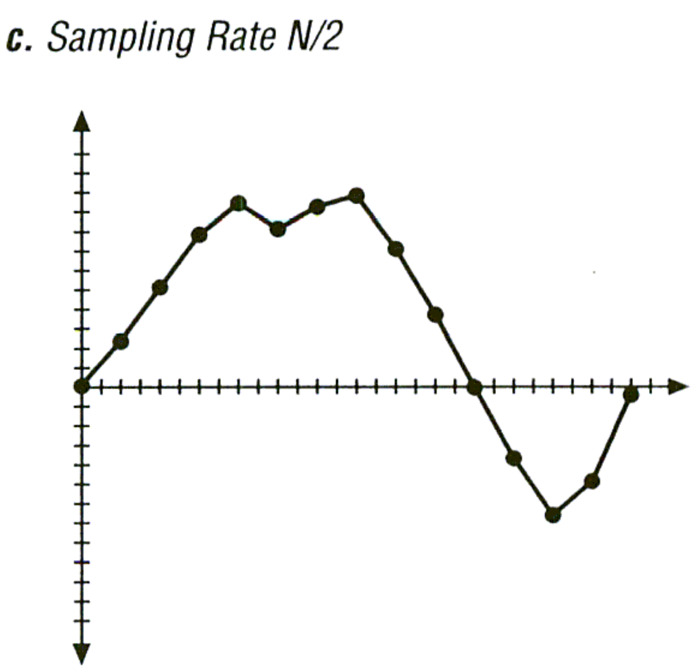
b. *Sampling Rate N*



Digital Sampling

- Half the sample rate

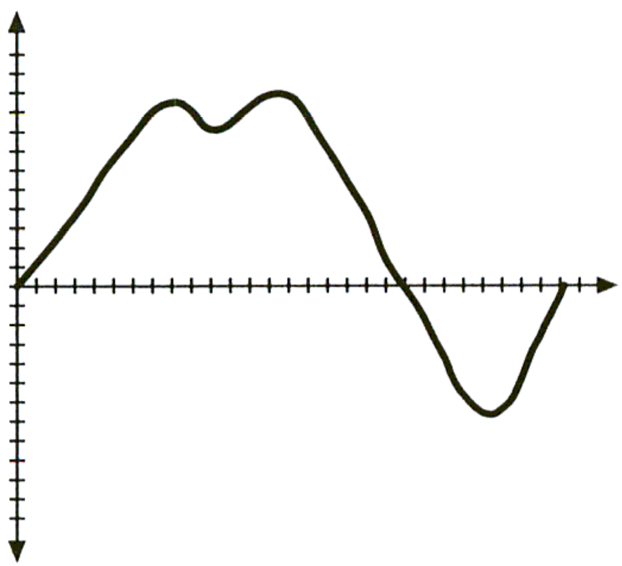




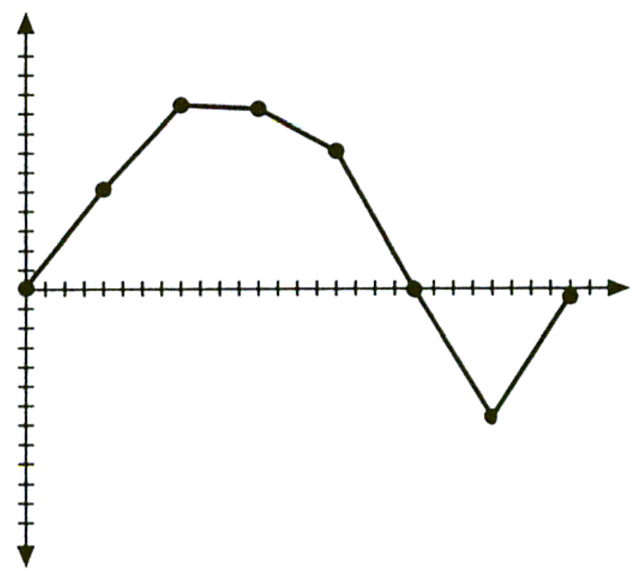
Digital Sampling

- Quarter the sample rate

a. Original Analog Waveform

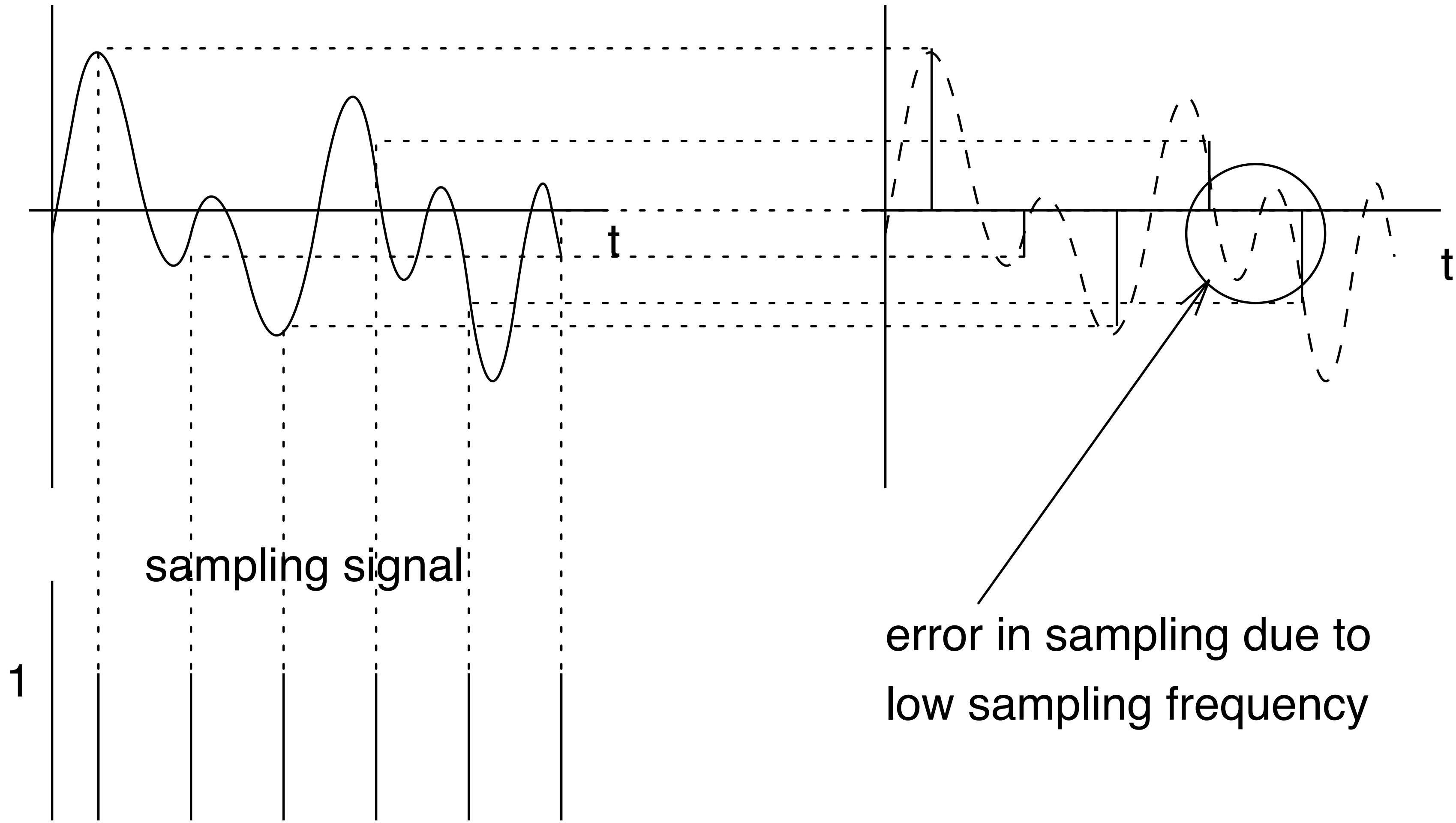


d. Sampling Rate N/4

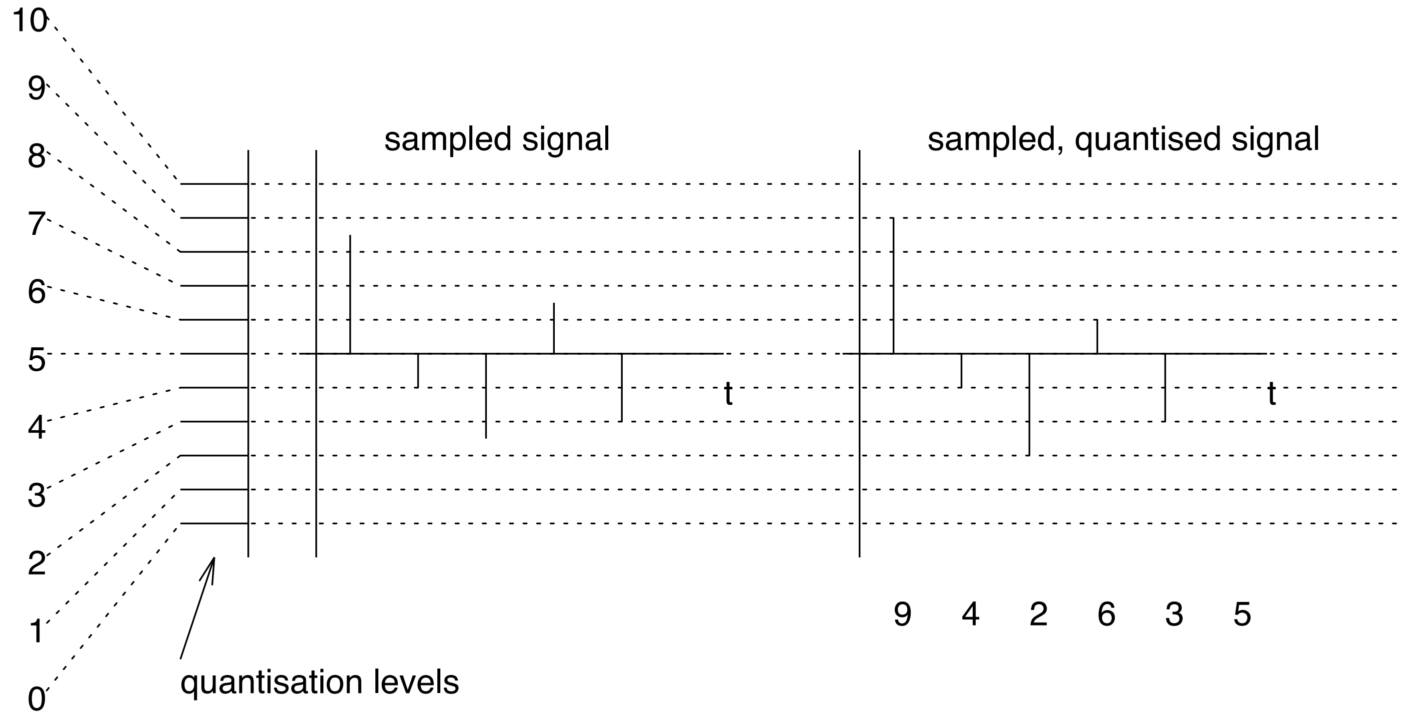


(How often to sample to reproduce curve?)

Digital Sampling



Digital Sampling



Two factors which determine the accuracy of digital sampling (analog to digital):

- *sampling rate*
- *quantization*

Sample Rate

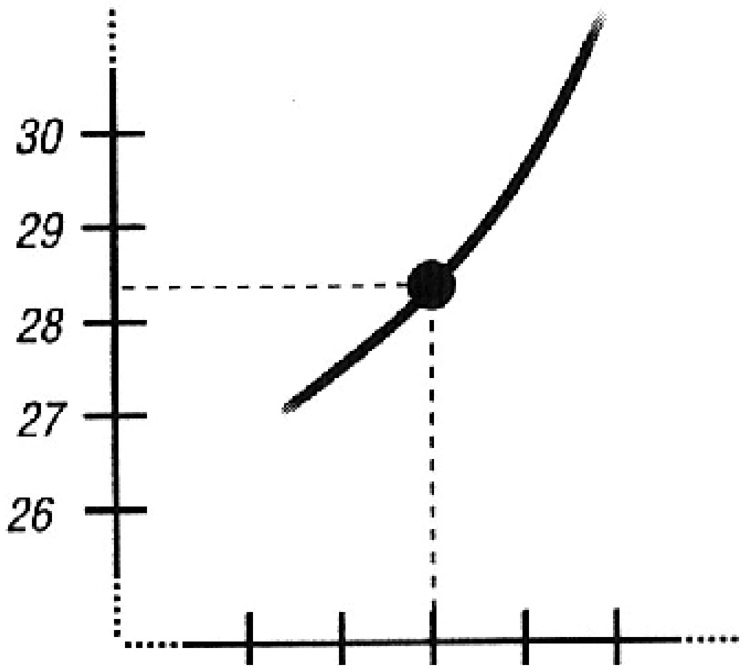
- Shannon’s Theorem: to accurately reproduce signal, must sample at twice the highest frequency
- Why not always use high sampling rate?

Sample Rate

- *Nyquist's theorem*: to accurately reproduce signal, must sample at twice the highest frequency (*mathematical proof provided by Claude Shannon*)
- Why not always use high sampling rate?
  - Requires more storage
  - Complexity and cost of analog to digital hardware
  - Human's can't always perceive
    - Dog whistle
  - Typically want an "*adequate*" sampling rate
    - "Adequate" depends upon use of reconstructed signal

Sample Size

- Samples have discrete values



- How many possible values?
  - *Sample Size*
  - Say, 256 values from 8 bits

Sample Size

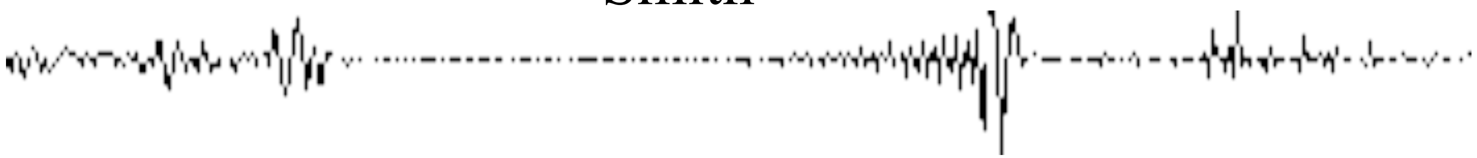
- *Quantization error* from rounding
  - Ex: 28.3 rounded to 28
- Why not always have large sample size?

Sample Size

- *Quantization error* from rounding
  - Ex: 28.3 rounded to 28
- Why not always have large sample size?
  - Storage increases per sample
  - Analog to digital hardware becomes more expensive

Audio

- Encode/decode devices are called *codecs*
  - Compression is the complicated part
- For voice compression, can take advantage of speech:



- Many similarities between adjacent samples
  - Send differences (ADPCM)
- Use understanding of speech
  - Can 'predict' (CELP)

Audio by People

- Sound by breathing air past vocal cords
  - Use mouth and tongue to shape vocal tract
- Speech made up of phonemes
  - Smallest unit of distinguishable sound
  - Language specific
- Majority of speech sound from 60-8000 Hz
  - Music up to 20,000 Hz
- Hearing sensitive to about 20,000 Hz
  - Stereo important, especially at high frequency
  - Lose frequency sensitivity with age

Typical Encoding of Voice

- Today, telephones carry digitized voice
- 8000 samples per second
  - Adequate for most voice communication
- 8-bit sample size
- For 10 seconds of speech:
  - 10 sec x 8000 samp/sec x 8 bits/samp
    = 640,000 bits or 80 Kbytes
  - Fit 3 minutes of speech on a floppy disk
  - Fit 8 weeks of sound on typical hard disk
- Ok for voice (but Skype better), but what about music?

Typical Encoding of Audio

- Can only represent 4 KHz frequencies (why?)
- Human ear can perceive 10-20 KHz
  - Full range used in music
- CD quality audio:
  - sample rate of 44,100 samples/sec
  - sample size of 16-bits
  - 60 min x 60 secs/min x 44100 samp/sec
    x 2 bytes/samples x 2 channels
    = 635,040,000, about 600 Mbytes (typical CD)
- Can use *compression* to reduce
  - mp3 ("as it sounds)", RealAudio
  - 10x compression rate, same audible quality

Sound File Formats

- Raw data has samples (interleaved w/stereo)
- Need way to ‘parse’ raw audio file
- Typically a header
  - Sample rate
  - Sample size
  - Number of channels
  - Coding format
  - …
- Examples:
  - .au for Sun µ-law, .wav for IBM/Microsoft
  - .mp3 for MPEG-layer 3

Homework

- Think of as many uses of computer audio as you can
- Which require a high sample rate and large sample size? Which do not? Why?

- Summarize the article of Paul Sellars:
  - Behind the Mask - Perceptual Coding: How MP3 Compression Works
  - http://www.soundonsound.com/sos/may00/articles/mp3.htm
  - The summary should not more than 300 words.

Today’s Outline

Coding and Compression

- Digital Audio
- **Digital Video**

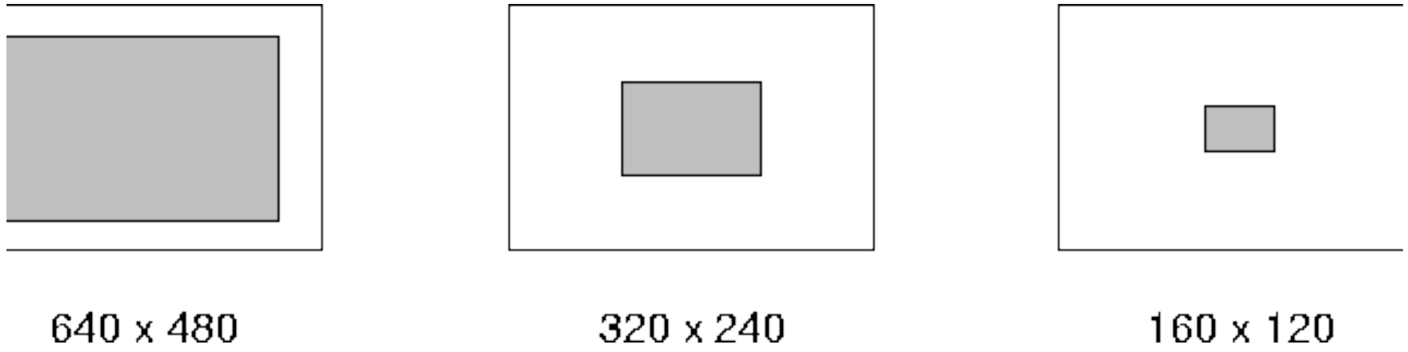
Graphics and Video

“A Picture is Worth a Thousand Words”

- People are visual by nature
- Many concepts hard to explain or draw
- Pictures to the rescue!
- Sequences of pictures can depict motion
  - Video!

Video Images

- Traditional television is 646x486 (NTSC)
- HDTV is 1920x1080 (1080p), 1280x720 (720p), 852x480 (480p)
- Digital video smaller
  - 352x288 (H.261), 176x144 (QCIF)



- Monitors higher resolution than traditional TV
- Computer video often called "Postage Stamp"

Moving Video Images

- Assume 30 fps uncompressed

Time:Size	640x480	320x240	160x120
1sec	27Mb	6.75Mb	1.68Mb
1min	1.6Gb	400Mb	100Mb
1hour	97Gb	24Gb	6Gb
1000hours	97Tb	24Tb	6Tb

Uncompressed video is enormous!

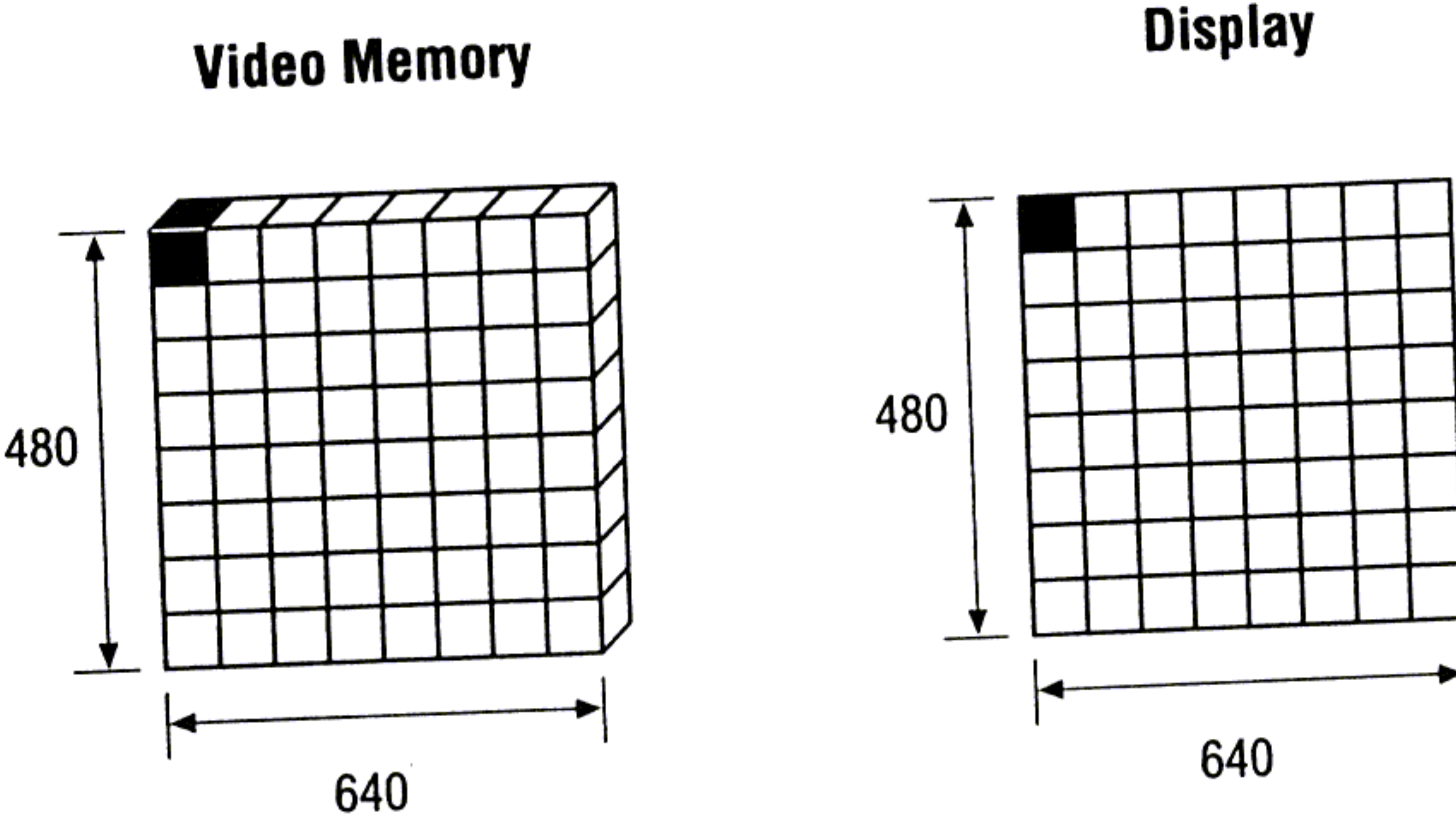
Video Image Components

- Luminance (Y) and Chrominance: Hue (U) and Intensity (V) - *YUV*
  – Human eye less sensitive to color than luminance, so those sampled at less resolution
- YUV has backward compatibility with BW televisions (only had Luminance)
  – Monitors are typically Red Green Blue (RGB)
  – (Why are primary colors Red Yellow Blue?)

Graphics Basics

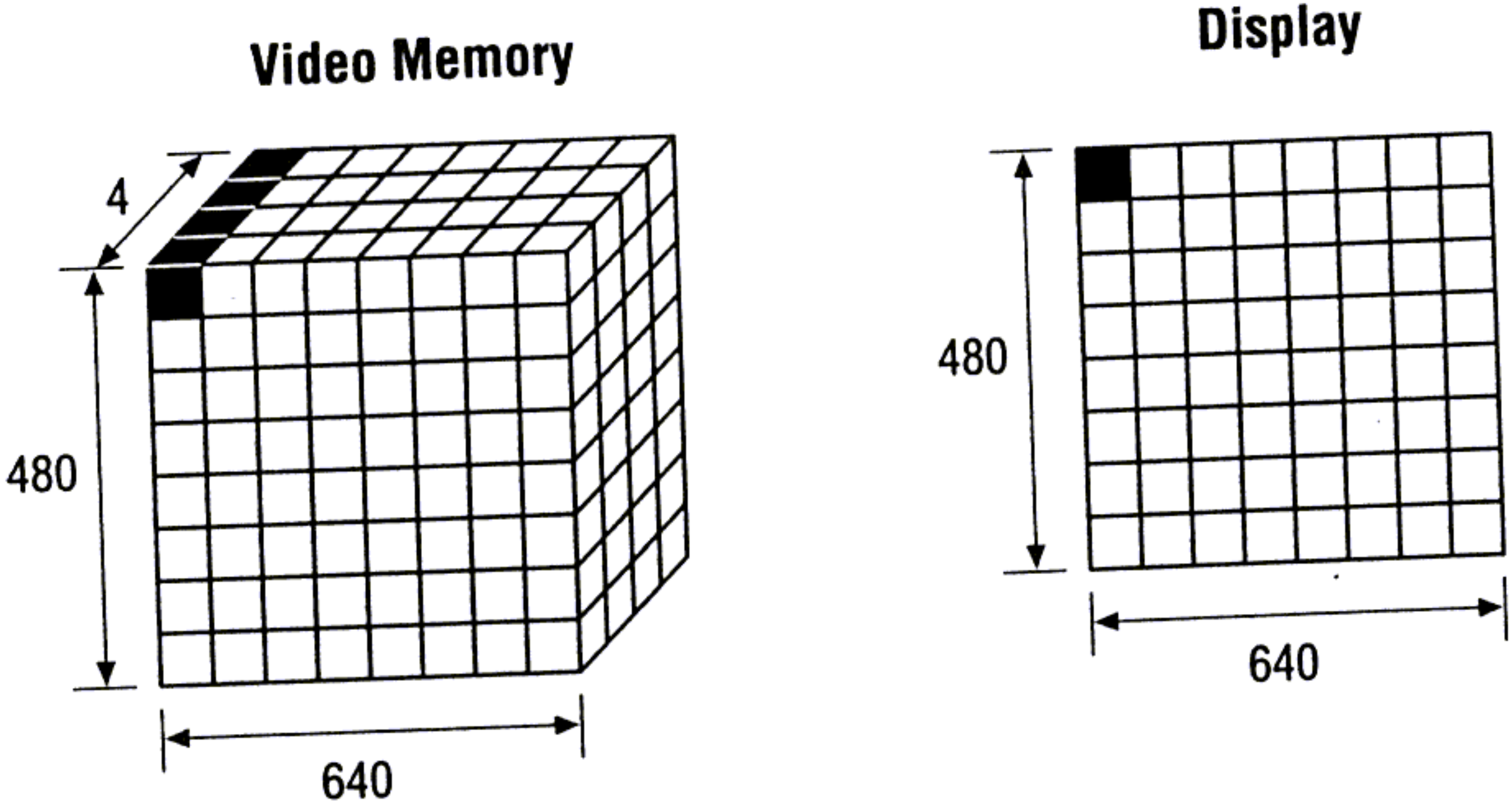
- Display images with graphics hardware
- Computer graphics (pictures) made up of pixels
  - Each pixel corresponds to region of memory
  - Called *video memory* or *frame buffer*
- Write to video memory
  - Traditional monitor displays with raster cannon
  - LCD monitors align crystals with electrodes

Monochrome Display



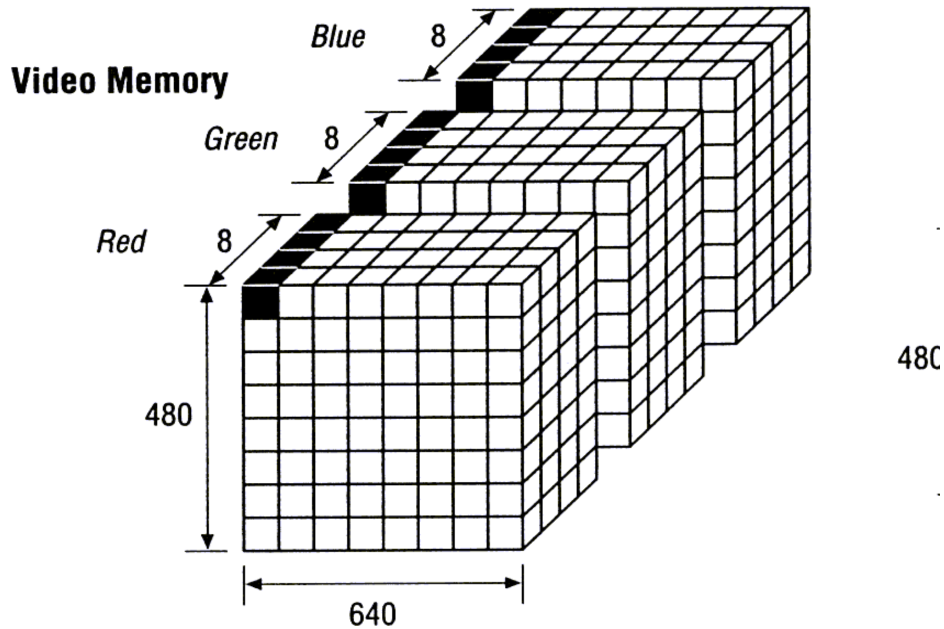
- Pixels are on (black) or off (white)
  - *Dithering* can make area appear gray

Grayscale Display



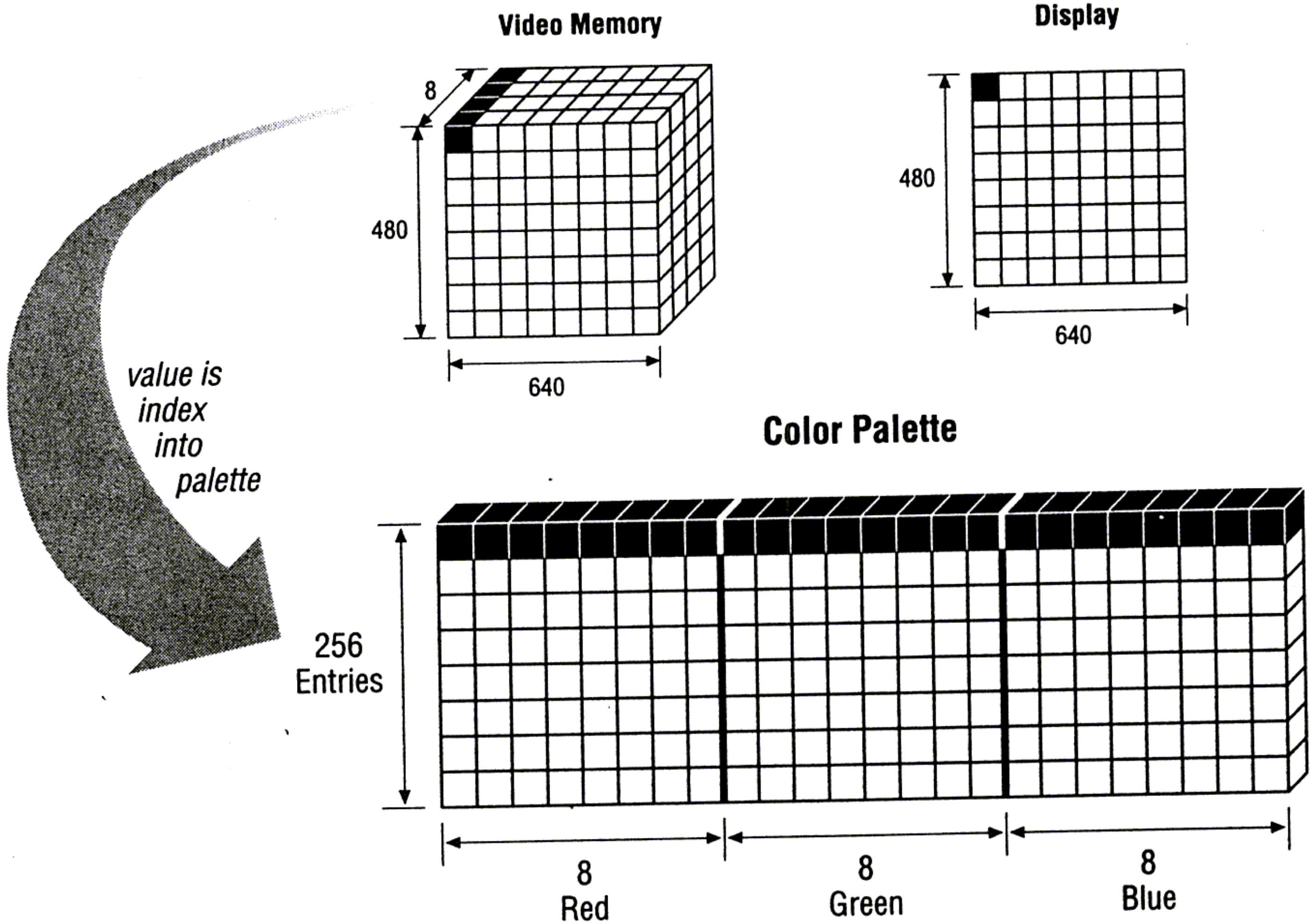
- *Bit-planes*
  - 4 bits per pixel, $2^4 = 16$ gray levels

Color Displays



- Humans can perceive far more colors than grayscales
  - Cones (color) and Rods (gray) in eyes
- All colors seen as combo of **red**, **green** and **blue**
- Visual maximum needed
  - 24 bits/pixel, 2^{24} ~ 16 million colors (true color)
- Requires 3 bytes per pixel

Video Palettes



- Still have 16 million colors, only 256 at a time
- Complexity to lookup, color flashing
- Can dither for more colors, too

Graphics Summary

Display Type	Bits Per Pixel	Colors	Resolu-tion	Video Memory
monochrome	1	2 (black and white)	640x480	38 KB
grayscale	4	16 shades of gray	640x480	150 KB
color	24	16 million	640x480	900 KB
color with palette	8	256 from palette of 16 million	640x480	301 KB
monochrome	1	2 (black and white)	1024x768	96 KB
grayscale	4	16 shades of gray	1024x768	384 KB
color	24	16 million	1024x768	2.3 MB
color with palette	8	256 from palette of 16 million	1024x768	769 KB

- Linux: xdpyinfo, display→settings
- Windows: rt click desktop→display properties→settings
- Mac: apple→system preferences→displays

Moving Video Images (Guidelines)

- Series of frames with changes appear as motion (typically ~ 30 fps)
- Unit is Frames Per Second (fps)
  - 24-30 fps: full-motion video
  - 15 fps: full-motion video approximation
  - 7 fps: choppy
  - 3 fps: very choppy
  - Less than 3 fps: slide show

Video Compression

Time v. Scale	None	3:1	25:1 (JPEG)	100:1 (MPEG)
1 sec	27 Mb	9 Mb	1.1 Mb	270 Kb
1 min	1.6 Gb	540 Mb	65 Mb	16 Mb
1 hour	97 Gb	32 Gb	3.9 Gb	970 Mb

640x480

Time v. Scale	None	3:1	25:1 (JPEG)	100:1 (MPEG)
1 sec	6.75 Mb	2.25 Mb	270 Kb	68 Kb
1 min	400 Mb	133 Mb	16 Mb	4 Mb
1 hour	24 Gb	8 Gb	1 Gb	240 Mb

320x240

- Lossless or Lossy
- Intracoded or Intercoded
  - Take advantage of dependencies between frames
    - Motion

Video Compression

- Three characteristics of compressing algorithms:
  - *spatial* and/or *temporal*
    - *spatial* looks for similar patterns or repetitions within a still *frame.*
    - *temporal* looks for changes during *a sequence of frames*.
  - *lossy* or *lossless* (also in audio)
    - lossy: permanently eliminate certain (redundant) information
      - e.g., JPEG, (MPEG-4 & H.264 also adopt it)
    - lossless: data remains the same after uncompressed
      - e.g., PNG, TIFF
  - *symmetrical* or *asymmetrical*
    - times taken for compression and decompression with *asymmetrical is* different (e.g., MPEG & AVI)

Video Compression

spatial coding example: instead of sending *N* values of same color (all purple), send only two values: color value (*purple*) *and number of repeated values* (N)

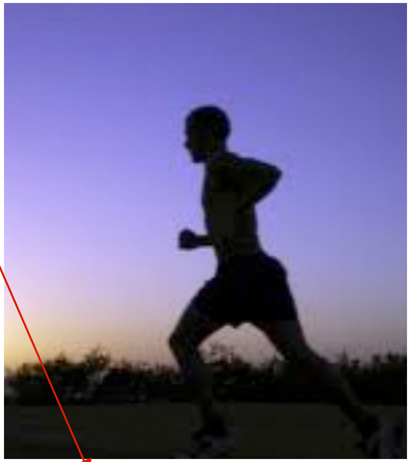
- video: sequence of images displayed at constant rate
  - e.g. 24 images/sec
- digital image: array of pixels
  - each pixel represented by bits
- coding: use redundancy within and between images to decrease # bits used to encode image
  - spatial (within image)
  - temporal (from one image to next)

Video Compression

- CBR: (constant bit rate): video encoding rate fixed
- VBR: (variable bit rate): video encoding rate changes as amount of spatial, temporal coding changes
- examples:
  - MPEG 1 (CD-ROM) 1.5 Mbps
  - MPEG2 (DVD) 3-6 Mbps
  - MPEG4 (often used in Internet, < 1 Mbps)

frame *i*

temporal coding example: instead of sending complete frame at i+1, send only differences from frame i



frame *i+1*