Multimedia

Week 7
LBSC 690
Information Technology
Agenda

• Questions
• Images
• Video
• Audio
• Streaming
• SMILe
• XML?
• (HCI)
Nothing new…

Georges Seurat, A Sunday Afternoon on the Island of La Grande Jatte
Visual Perception

• Closely spaced dots appear solid
  – But irregularities in diagonal lines can stand out

• Any color can be produced from just three
  – Red, Blue and Green: “additive” primary colors

• High frame rates produce apparent motion
  – Smooth motion requires about 24 frames/sec

• Visual acuity varies markedly across features
  – Discontinuities easily seen, absolutes less crucial
Basic Image Coding

• Raster of picture elements (pixels)
  – Each pixel has a “color”
    • Binary - black/white (1 bit)
    • Grayscale (8 bits)
    • Color (3 colors, 8 bits each)
      – Red, green, blue

• Screen
  – A 1024x768 image requires 2.4 MB
    • So a picture is worth 400,000 words!
Monitor Characteristics

• Technology (CRT, Flat panel)
• Size (15, 17, 19, 21 inch)
  – Measured diagonally
  – For CRT, key figure is “viewable area”
• Resolution
  – 640x480, 800x600, 1024x768, 1280x1024, …
• Layout (three dot, lines)
• Dot pitch (0.26, 0.28)
• Refresh rate (60, 72, 80 Hz)
Pop Quiz

• How many images can a 1 GB SD store?
  – But mine holds about 500. How?
Compression

• Goal: reduce redundancy
  – Send the same information using fewer bits

• Originally developed for fax transmission
  – Send high quality documents in short calls

• Two basic strategies:
  – Lossless: can reconstruct exactly
  – Lossy: can’t reconstruct, but looks the same
Palette Selection

• Opportunity:
  – No picture uses all 16 million colors
  – Human eye does not see small differences

• Approach:
  – Select a palette of 256 colors
  – Indicate which palette entry to use for each pixel
  – Look up each color in the palette

“The rain in Spain falls mainly in the plain”
→ [*=ain, ^=in] “The r* ^ Sp* falls m*ly ^ the pl***”
Run-Length Encoding

• Opportunity:
  – Large regions of a single color are common

• Approach:
  – Record # of consecutive pixels for each color

Sheep go baaaaaaaaaa and cows go mooooooooo
→ Sheep go ba<10> and cows go mo<10>

• An example of lossless encoding
GIF

• Palette selection, then lossless compression

• Opportunity:
  – Common colors are sent more often

• Approach:
  – Use fewer bits to represent common colors

  • 1  Blue  75%  75x1= 75  75x2=150
  • 01  White  20%  20x2= 40  20x2= 40
  • 001  Red  5%  5x3= 15  5x2= 10
    130  200
JPEG

• Opportunity:
  – Eye sees sharp lines better than subtle shading

• Approach:
  – Retain detail only for the most important parts
  – Accomplished with Discrete Cosine Transform
    • Allows user-selectable fidelity

• Results:
  – Typical compression 20:1
Variable Compression in JPEG

37 kB (20%)  4 kB (95%)
Vector Graphics
Vector Graphics

• Raster images ("bitmap graphics")
  – Actually describe the contents of the image
  – Good for natural scenes

• Vector images
  – Mathematically describe how to draw the image
  – Rescalable without loss of resolution
Discussion Point:
Selecting an Image Format

• Should I use GIF, JPEG, or vector graphics for …
  • Color photos?
  • Scanned black & white text?
  • Line drawings?
Basic Video Coding

• Display a sequence of images
  – Fast enough for smooth motion and no flicker

• NTSC Video
  – 60 “interlaced” half-frames/sec, 512x486

• HDTV
  – 30 “progressive” full-frames/sec, 1280x720
Video Data Rates

- “NTSC” Quality Computer Display
  - 640 X 480 pixel image
  - 3 bytes per pixel (red, green, blue)
  - 30 Frames per Second

- Bandwidth
  - 26.4 MB/second
  - Exceeds bandwidth of most disk drives

- Storage
  - CD-ROM would hold 25 seconds worth
  - 30 minutes would require 46.3 GB
Video Compression

• Opportunity:
  – One frame looks very much like the next

• Approach:
  – Record only the pixels that change

• Standards:
  – MPEG-1: Web video (file download)
  – MPEG-2: HDTV and DVD
  – MPEG-4: Web video (streaming)
MPEG Encoding

Frame Types

- **I** Intra: Encode complete image, similar to JPEG
- **P** Forward Predicted: Motion relative to previous I and P’s
- **B** Backward Predicted: Motion relative to previous & future I’s & P’s
Frame Reconstruction

I frames provide complete image

P frames provide series of updates to most recent I frame
Frame Reconstruction

I₁, I₁+P₁, I₁+P₁+P₂, I₂

Interpolations

B₁, B₂, B₃, B₄, B₅, B₆, B₇, B₈, B₉

B frames interpolate between frames represented by I’s & P’s
Basic Audio Coding

• Sample at twice the highest frequency
  – 8 bits or 16 bits per sample

• Speech (0-4 kHz) requires 8 kB/s
  – Standard telephone channel (1-byte samples)

• Music (0-22 kHz) requires 172 kB/s
  – Standard for CD-quality audio (2-byte samples)
Music Compression

• Opportunity:
  – The human ear cannot hear all frequencies at once

• Approach:
  – Don’t represent “masked” frequencies

• Standard: MPEG-1 Layer 3 (.mp3)
Temporal Masking

If we hear a loud sound, then it stops, it takes a while until we can hear a soft tone at about the same frequency.

“Psychoacoustic compression”

- Eliminate sounds below threshold of hearing
- Eliminate sounds that are frequency masked
- Eliminate sounds that are temporally masked
- Eliminate stereo information for low frequencies
Speech Compression

• Opportunity:
  – Human voices vary in predictable ways

• Approach:
  – Predict what’s next, then send only any corrections

• Standards:
  – Rule of thumb: 1 kB/sec for (highly compressed) speech

• Demo at http://www.data-compression.com/speech.html
  – Scroll down to near the bottom
Narrated PowerPoint

• Create your slides

• Slide Show -> Record Slide Show
  – Set microphone level

• Record the narration
  – Slide transitions are automatically captured

• Narration plays automatically when displayed
The “Last Mile”

• Traditional modems
  – “56” kb/sec modems really move ~3 kB/sec

• Digital Subscriber Lines
  – 384 kb/sec downloads (~38 kB/sec)
  – 128 kb/sec uploads (~12 kB/sec)

• Cable modems
  – 10 Mb/sec downloads (~1 MB/sec)
  – 256 kb/sec uploads (~25kB/sec)
Multimedia on a Web Server

- Object stored in a file
- File transferred as an HTTP object:
  - Received entirely at the client
  - Passed to media player
Streaming

- Browser gets metafile over HTTP
  - Launches media player to interpret the metafile
- Media player contacts streaming server
Streaming Audio and Video

- Begin replay after only a portion received
- Buffer provides time to recover lost packets
- Interrupts replay when “rebuffering”
Client Buffering

- Client-side buffering:
  - Playout delay compensates for network delay
Playout Delay

• Receiver attempts to playout each chunk exactly $q$ ms after chunk was generated
  – Chunk has time stamp $t$: play out chunk at $t+q$
  – Arrives after $t+q$: too late for playout, data “lost”

• Tradeoff for $q$:
  – Large $q$: less packet loss
  – Small $q$: better interactive experience

• Easy to increase $q$ by inserting a pause
  – Decreasing $q$ requires skipping or accelerating
Lost Packets

• Network loss
  – Packets completely lost (e.g., due to collisions)

• Delay loss
  – Packets arrives too late for playout
    • Queueing; sender and receiver processing delays

• Loss tolerance
  – 1% to 10% packet loss may be tolerable
    • Some encoding schemes are more tolerant than others
Q: how to handle different client receive rate capabilities?
   – 28.8 Kbps dialup
   – 100Mbps Ethernet

A: server stores, transmits multiple copies of video, encoded at different rates
Internet Telephony

• Characteristics:
  – “Live” (<400 ms delay)
  – Alternating talk spurts
Before You Go!

• On a sheet of paper (no names), answer the following question:

What was the muddiest point in today’s class?