Lecture 3 – Sampled Images

Michael Gleicher
CS559-2007
Notes not for display – for notes only

Overview

- Last Time
  - Light -> Images
  - Sensing on the Image plane
  - Eyes
- Unanswered questions
  - Interlace
- What does it mean to have a sampled representation?

Today

- Intensity, Quantization, Gamma Correction
- Image Representation
- Sampling for imaging
- Sampling for display
- Issues in Sampling
- Aliasing
- Sampling, Reconstruction, Resampling, Aliasing

The image plane

- 2D (surface, even if not flat)
- Distribution of light – can measure anywhere
  - F(x,y)
- Discrete set of points
  - Measurements taken
  - Measurements displayed
- Represent this discrete set of measurements
  - Regular "lattice" of measurements
  - Often a grid

How sensitive is the eye?

- Amazing range!
  - Night vision – when eyes adjusted, camping
  - Bright daylight
    - `Sunlight 10000.`
    - `Twilight 10.`
    - `Starlight 0.001`
- Catch: at any given time, can't see this range
  - Adaptation – bright light, iris closes, lets in less light, …
- At any given time, about 100:1 contrast ratio
  - This is a lot more than most displays
  - Better displays = more constrast
    - Often by blacker blacks

High Dynamic Range Imagery

- Most sensors/displays have less range than eye
  - Certainly less range than scenes do
- What happens?
  - Bright areas – all white (no details)
  - Dark (shadow) areas – all black (no details)
- What to do?
  - Adjust exposure (time, aperature, sensitivity) to get the most important stuff
  - Acquire "High Dynamic Range" Imagery
    - Special sensors
    - Multiple exposures (at different settings) – cool thing to do
  - HDR later in the course

Perception of intensity

- Eye senses relative differences
  - Equivalent differences 50:100 20:40
  - Hard to tell absolute differences directly
    - Adaptation to current setting
- Can sense 1% differences
- At any given time 100:1 contrast ratio
- How many levels can you see in an image?
  - 1.01 ^ 463 = 100.2 (e.g. 463 1% differences = 100:1)
  - This is about 8 bits of precision (less than 9)
  - But its VERY non linear 1, 1.01, …. , 99.2, 100.2

Non-linearity of intensity

- Non-linear mapping from "amount of light" to perceived brightness
- Want uniform mapping of intensities -> perception
  - Level 1, 2, 3, …. 255 -> 1, 1.01, 1.02, … 99, 100
- Worse: displays are non-linear too
  - Voltage -> amount of light is non-linear
  - Different displays are different
- Want to linearize the system
  - Intensity levels map nicely to perceived levels

Gamma correction

- Idea: put a non-linear function between intensity and output
  - Done as the last step (usually) – after all computations
- Could create arbitrary functions for mapping
  - Too cumbersome
- Exponential is a good approximate model
  - Exponential non-linearity of perception
  - Exponential power laws in CRTs

Modeling a display device

- 5/2 power law (five-halves)
  - Models physics of a CRT
  - Real CRTs are close, LCDs designed to be similar
- $L = M (i+\varepsilon)$^γ
  - i = input intensity value
  - L = amount of light
  - ε = since zero isn't really black
  - M = maximum intensity
  - γ = specific property of display

Linearizing the display

- Define a function **g** that corrects for non-linearity
- $L = M (g(i))$^γ (ignoring ε)
  - $G = 1/\gamma$
- Where do we get γ from?
  - Pick it so things look right
- Note: 1^{st} order approximation (very simple)
  - Only 1 parameter to specify (γ), many factors

Gamma correction

- Want value 0 = minimum intensity
- Want value max (1 or 255) = maximum intensity

  --- those 2 are easy to get
- Pick one more point
  - Midpoint should be 50%
  - Easy – show 50% black white + 50% gray
  - Adjust gamma until it looks the same
- All this happens "behind the scenes"
- Everything gets harder when we deal with color

What to store in the frame buffer?

- Frame Buffer = rectangular chunk of memory
- Intensity measurements
  - Deal with color later, basically store multiple monochrome
- Continuous range of intensities
  - 8-9 bits of precision ideally
    - More since can't get exactly right (10-12 bits)
    - More since want more dynamic range (12-14 bits)
    - More since want linear space to make math easy (16-32 bits)
- Discrete set of choices – **QUANTIZATION**
  - Inks, palettes, color tables, …
  - Less storage cost + Color table animation

The image plane

- 2D (surface, even if not flat)
- Distribution of light – can measure anywhere
  - F(x,y)
- Discrete set of points
  - Measurements taken
  - Measurements displayed
- Represent this discrete set of measurements
  - Regular “lattice” of measurements
  - Often a grid

Raster Scanning

- 2D rectangular grid (common, easy, …)
- Serialize (1,1) (1,2) (1,3) … (1,w), (2,1), …
- Useful for storing in memory
- Useful for transmission
  - Send samples in order

How (analog) TV works

- CRT beam lights up phosphors
- Scans across (zig-zag pattern)
  - Horizontal, vertical retrace time
- Send images as a signal
  - Rows easy (although band-limited)
- 1930s radio technology limited rate
  - Couldn’t achieve 60hz full image
  - Interlace (avoid flicker, but get full resolution)

What is a pixel

- One of these finite measurements
- At a particular position
- Point sample – value at a specific place
  - Infinitesimally small place
- Finite region of constant value (little square)
  - Doesn’t actually model things better (inconsistent)
  - Mathematically less convenient
  - Useful for some thought experiments later on

A pixel is not a little square!

- Sensors average over region
  - Doesn’t mean its really peicewise constant
  - Don’t really know what went on in the square
- Point Samples (paradoxically) fit better with the finite case (the buckets, screen dots)
  - Sensing – estimation of what happens at the point from the neighborhood
  - Display – neighborhood is created based on the points inside of it (splats, bleeding, …)

Dealing with discretization

- Sampling
  - Understand what information we are throwing away
- Reconstruction
  - Recreate as well as possible from the samples
- Re-Sampling
  - Sample a sampled image
  - Transform the image
- Signal Processing / Image Processing
- Consider the 1D case first since its easier

Point Sampling Has Problems

- Miss small things
- Problem: discretization throws away information
- Don't know what happens between samples
- Sampling loses information – you cannot get back the information once its lost!

Aliasing

- Technical term for sampling problems
- If you lose information and "make it up" wrong, you get weird effects

Point Sampling Has Problems

- Miss small things
- Problem: discretization throws away information
- Don't know what happens between samples
- Sampling loses information – you cannot get back the information once its lost!

Dealing with discretization

- Sampling
  - Understand what information we are throwing away
- Reconstruction
  - Recreate as well as possible from the samples
- Re-Sampling
  - Use a sensor on a sampled representation
  - Transform the image
- Signal Processing / Image Processing
- Consider the 1D case first since its easier