**Images and Displays**

CS465 Lecture 2

---

**What is an image?**

- A photographic print
- A photographic negative?
- This projection screen
- Some numbers in RAM?

---

**An image is:**

- A 2D distribution of intensity or color
- A function defined on a two-dimensional plane

\[ I : \mathbb{R}^2 \rightarrow \ldots \]

- Note: no mention of pixels yet
- To do graphics, must:
  - represent images—encode them numerically
  - display images—realize them as actual intensity distributions

---

**Representative display technologies**

**Computer displays**

- Raster CRT display
- LCD display

**Printers**

- Laser printer
- Inkjet printer

---

**Cathode ray tube**

- First widely used electronic display
  - developed for TV in the 1920s–1930s

---

**Vector CRT display**

- Beam steered under program control
- Image representation = command stream
**Raster CRT display**

- Scan pattern fixed in display hardware
- Intensity modulated to produce image
- Originally for TV
  - (continuous analog signal)
- For computer, intensity determined by contents of framebuffer

**LCD flat panel or projection display**

- Principle: block or transmit light by twisting its polarization
- Intermediate intensity levels possible by partial twist
- Fundamentally raster technology
- Fixed format

**Raster display system**

- Screen image defined by a 2D array in RAM
  - for CRT, read out and convert to analog in sync with scan
- In most systems today, it’s in a separate memory
- The memory area that maps to the screen is called the framebuffer

**Color displays**

- Operating principle: humans are trichromatic
  - match any color with blend of three
  - therefore, problem reduces to producing 3 images and blending
- Additive color
  - blend images by sum
  - e.g. overlapping projection
  - e.g. unresolved dots
  - R, G, B make good primaries

**Color displays**

- Operating principle: humans are trichromatic
  - match any color with blend of three
  - therefore, problem reduces to producing 3 images and blending
- Additive color
  - blend images by sum
  - e.g. overlapping projection
  - e.g. unresolved dots
  - R, G, B make good primaries
Color displays

- Operating principle: humans are trichromatic
  - match any color with blend of three
  - therefore, problem reduces to producing 3 images and blending
- Additive color
  - blend images by sum
  - e.g. overlapping projection
  - e.g. unresolved dots
  - R, G, B make good primaries

Color displays

- CRT: phosphor dot pattern to produce finely interleaved color images
- LCD: interleaved R,G,B pixels

Laser printer

- Xerographic process
- Like a copier but with laser-scanned raster as source image
- Key characteristics
  - image is binary
  - resolution is high
  - very small dots are not possible

Inkjet printer

- Liquid ink sprayed in small drops
  - very small—measured in picoliters
- Head with many jets scans across paper
- Key characteristics:
  - image is binary (drop or no drop; no partial drops)
  - isolated dots are reproduced well

Raster image representation

- All these devices suggest 2D arrays of numbers
- Big advantage: represent arbitrary images
  - approximate arbitrary functions with increasing resolution
  - works because memory is cheap (brute force approach!)

Meaning of a raster image

- Meaning of a given array is a function on 2D
- Define meaning of array = result of output device?
  - that is, piecewise constant for LCD, blurry for CRT
  - but: we don’t have just one output device
  - but: want to use images we can’t display (e.g. too big)
- Abstracting from device, problem is reconstruction
  - image is a sampled representation
  - pixel means “this is the intensity around here”
    - LCD: intensity is constant over square regions
    - CRT: intensity varies smoothly across pixel grid
  - will discuss specifics of reconstruction next week
Datatypes for raster images

- Bitmaps: boolean per pixel (1 bpp): $I : \mathbb{R}^2 \rightarrow \{0, 1\}$
  - interp. = black and white; e.g., fax
- Grayscale: integer per pixel: $I : \mathbb{R}^2 \rightarrow [0, 1]$
  - interp. = shades of gray; e.g., black-and-white print
  - precision: usually byte (8 bpp); sometimes 10 or 12 bpp
- Color: 3 integers per pixel: $I : \mathbb{R}^2 \rightarrow [0, 1]^3$
  - interp. = full range of displayable color; e.g., color print
  - precision: usually byte (3) (24 bpp)
  - sometimes 16 (5+6+5) or 30 or 36 bpp
  - indexed color: a fading idea

- Floating point images: $I : \mathbb{R}^2 \rightarrow \mathbb{R}_+$
  - more abstract, because no output device has infinite range
  - provides high dynamic range (HDR)
  - represent real scenes independent of display
  - represent lighting in realistic scenes
  - about to become standard intermediate format in graphics hardware

- For color or grayscale, sometimes add alpha channel
  - describes transparency of images
  - more on this next lecture
Storage requirements for images

- 1024x1024 image (1 megapixel)
  - bitmap: 128KB
  - grayscale 8bpp: 1MB
  - grayscale 16bpp: 2MB
  - color 24bpp: 3MB
  - floating-point HDR color: 12MB

Converting pixel formats

- Up is easy; down loses information—be careful

8 bpp (256 grays)

---

Converting pixel formats

- Up is easy; down loses information—be careful

7 bpp (128 grays)

---

Converting pixel formats

- Up is easy; down loses information—be careful

6 bpp (64 grays)

---

Converting pixel formats

- Up is easy; down loses information—be careful

5 bpp (32 grays)

---

Converting pixel formats

- Up is easy; down loses information—be careful

4 bpp (16 grays)
Converting pixel formats

- Up is easy; down loses information—be careful

3 bpp (8 grays)

2 bpp (4 grays)

Dithering

- When decreasing bpp, we quantize
- Make choices consistently: banding
- Instead, be inconsistent—dither
  - turn on some pixels but not others in gray regions
  - choose pattern based on output device
  - laser, offset: clumped dots required (halftone)
  - inkjet, screen: dispersed dots can be used

Dither matrix

![Dither matrix](image)

- Fill in pixels numbered < pixel value

Larger ordered dither

![Larger ordered dither](image)
Error diffusion

- Avoid regular patterns
- Every time we make a choice:
  - keep track of error
  - propagate to neighboring pixels

![Diagram showing error diffusion]

Intensity units in images

- Say pixel value is 123
  - this means the intensity is 123. 123 what?
  - look to devices to motivate definition
    - transfer function of a display

![Graph showing intensity units in images]
Display transfer function

Intensity units in images

- Say pixel value is 123
  - this means the intensity is 123. 123 what?
  - look to devices to motivate definition
    - transfer function of a display

Why nonlinear intensity?

- Original reason: CRTs are like that
  - intensity on screen is proportional to voltage\(^2\)
- Continuing reason: perceptual uniformity
  - our eyes are sensitive to relative intensity differences
  - this means we can see smaller steps in darker areas
  - therefore we want to concentrate the available quantization levels towards the dark end of the scale

Intensity units in images

- Say pixel value is 123
  - this means the intensity is 123. 123 what?
  - look to devices to motivate definition
    - transfer function of a display

Why nonlinear intensity?

- Closer to ideal perceptually uniform exponential
Gamma correction

- Sometimes (often, in graphics) we have computed intensities that we want to display linearly.
- In the case of an ideal monitor with zero black level,

\[ I(n) = (n/N)^\gamma \]

so if we define

\[ n = Na^{\frac{1}{\gamma}} \]

we will get linear intensity out:

\[ I(n) = (Na^{\frac{1}{\gamma}}/N)^\gamma = a \]