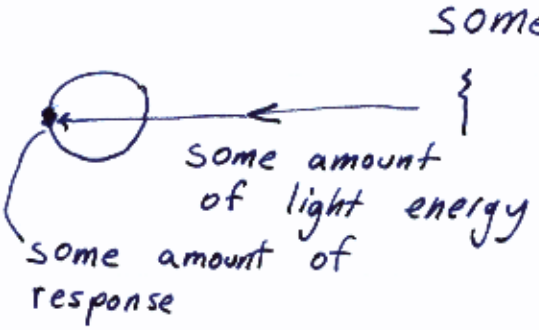
Day 6

Fix Demoprogram
Skeleton Driver

Program due today
Homework due Tuesday
Randy DL / UPL time

Today: Quantization and Color

Recap:



Eye sees a range of amounts
0 black ⟶ n (max)

differentiate 1% changes

1 unit of energy ⇒ 2 units of energy
Big change
200 → 201
~~1~~ half a percent - imperceptible

Linear increase in energy (luminence)
does not lead to linear increment in brightness (perceived amount)

Perception vs. Displays

Image -
what do numbers mean
black and white (monochrome)
for now

We percieve "ratios"

Brightness differences of 1% are noticeable, but very similar
(exact number depends on person, but its actually around 1%)

So if we measure "lightness units"

100	to	101	is	1%
25	to	26	is	4% - very noticeable
200	to	201	is	½% - not visible

Makes a logarithmic curve

100:1 = 463 levels

But, how do we specify "lightness"

amount of photons coming out of monitor?

control amount of electricity to monitor

non-linear

5/2 power law

if we want "brightness" or "pixel values" to be linear with light output (assuming value ≈ voltage) must account for this non-linearity

gamma correction

end-to-end argument

what do the numbers mean?

need to correct for everything....
- black level
- monitor transfer
- perception

Why do we need to get this right?

Color precision

How to get it right?

- A different way to get lightness: halftone
  - black and white dots
  - half black / half white = 50% gray
  - turns out that this is complicated:
    - reflectance, bleed, ...
    - ↳ printing issues ≅ gamma

Gives us a way to find gamma (or define "correct")

- 50% grey should = 128
- Good systems let you adjust gamma to get this right
  - hardware, or software (Photoshop)
- Actually, black level may be more important to control
  - $(I+\epsilon)^{1/\gamma}$ 0 is not truly black.

Halftone reading (along with color) for next time

- simple halftoning (resolution reducing)
- screens (change dot size) ← printing
- repetition
- artistic screens
- not just bi-level
- relationship to sampling

1.
1.01 1%

100:1 →

Is 8-bits for intensity enough?

- linear intensity - no! difference between 25 and 26 = 4%
  - easy to see (3-4 is probably too dark to see)
- even though can only see ≅ 250 different levels, need 12 or 16 bit coding (or non-linear, which makes math harder)

How to do this ?

Halftone screen or mask matrix

4 values
0 2
3 1

Use screen without recentering

example

More generally -

region around pixel must average to value

One way - error diffusion

Minimize pixel value - Average over area

Floyd - Steinberg Algorithm

start at upper left -
set pixel to 0 or 1
push error down and to right

$e = f - \bar{f}$	$\frac{3}{8}e$
$\frac{3}{8}e$	$\frac{1}{4}e$

or

	e	$\frac{7}{16}$
$\frac{3}{16}$	$\frac{5}{16}$	$\frac{1}{16}$

.5 .5 .5 .5
.5 .5 .5 .5
.5 .5 .5 .5

Thresh = > .5, 0 1

8 8 8
8 8 8
8 8 8

thresh > 8 0 16
err = 4

$8 + \frac{12}{8} = 9\frac{1}{2} > 8$
$-16 \quad 9\frac{1}{2} = -6\frac{1}{2}$

problem: directional artifacts (variants address)

good news: generalizes (colors, multiple levels, ...)