Images and Displays

CS4620 Lecture 11

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

Direct-view displays

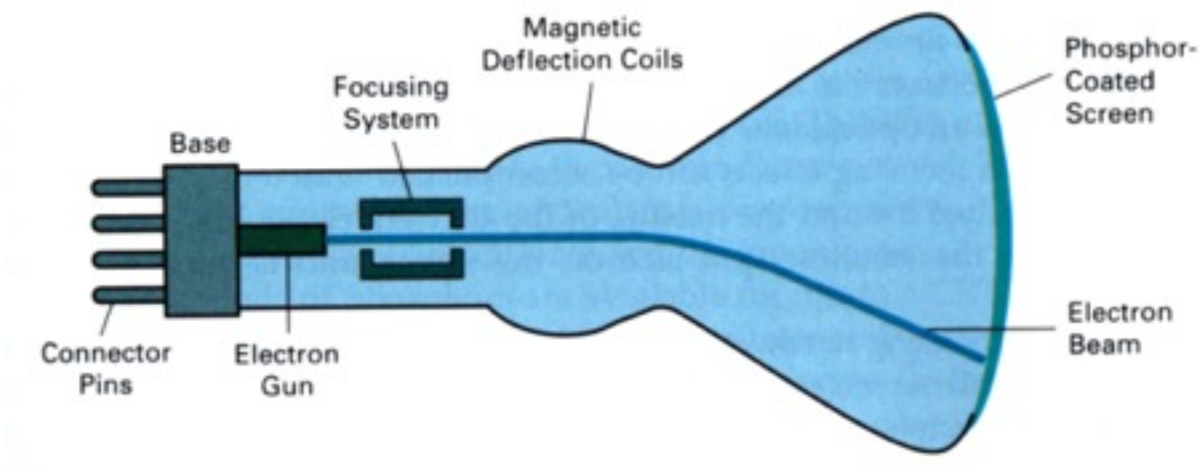
- Raster CRT display
- LCD display
- LED display

Printers

- Laser printer
- Inkjet printer

Cathode ray tube

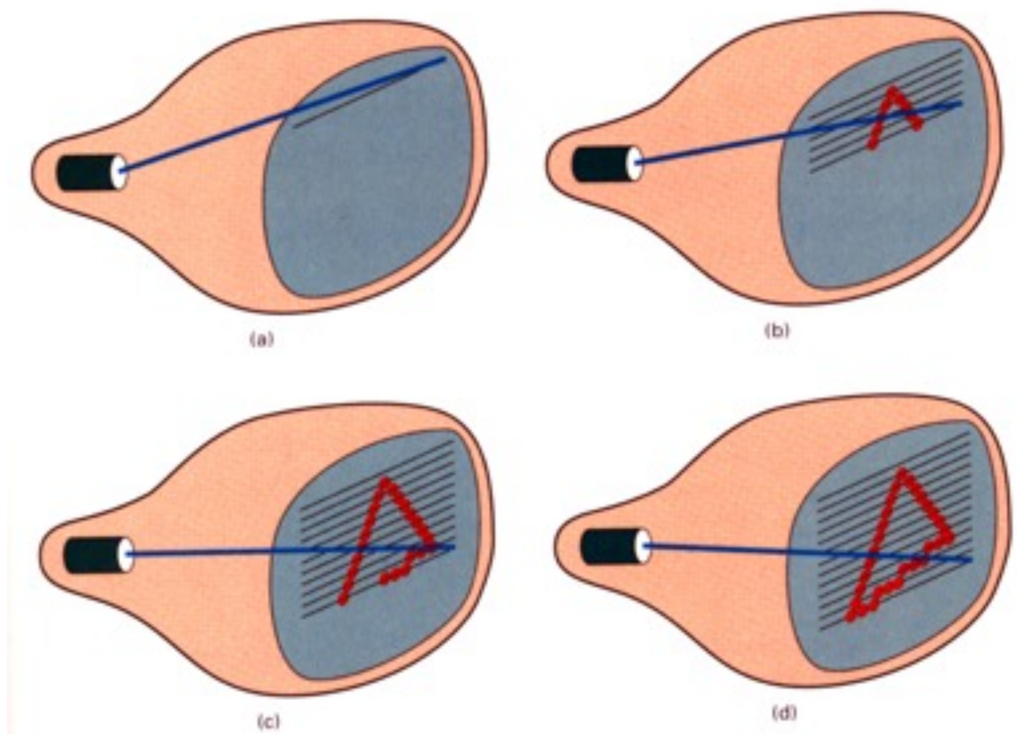
- First widely used electronic display
  - developed for TV in the 1920s–1930s



[H&B fig. 2-2]

Raster CRT display

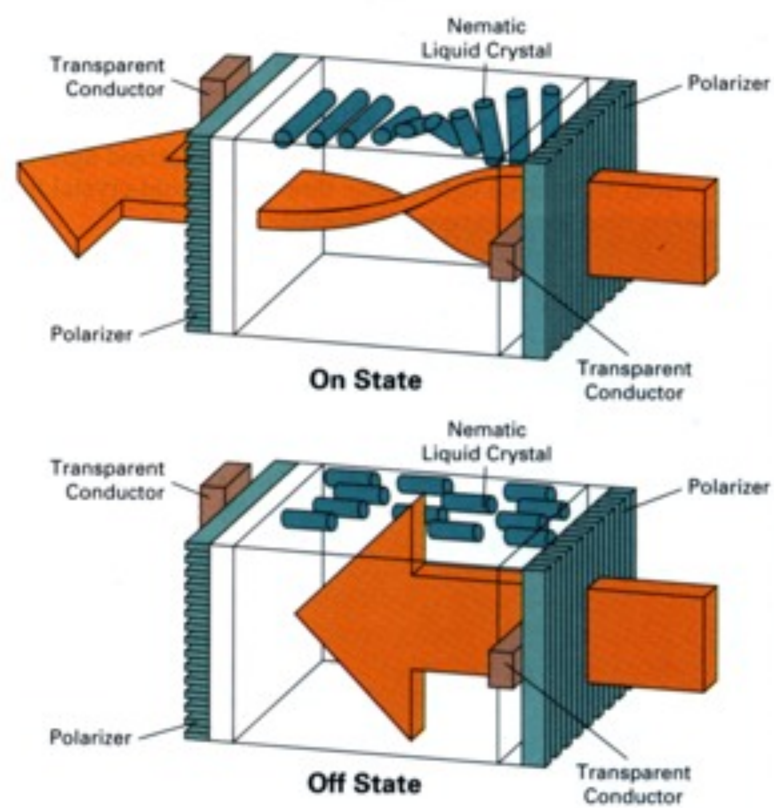
- Scan pattern fixed in display hardware
- Intensity modulated to produce image
- Originally for TV
  - (continuous analog signal)
- For computer, intensity determined by contents of *framebuffer*



[H&B fig. 2-7]

LCD flat panel display

- Principle: block or transmit light by twisting its polarization
- Illumination from backlight (either fluorescent or LED)
- Intermediate intensity levels possible by partial twist
- Fundamentally raster technology
- Fixed format



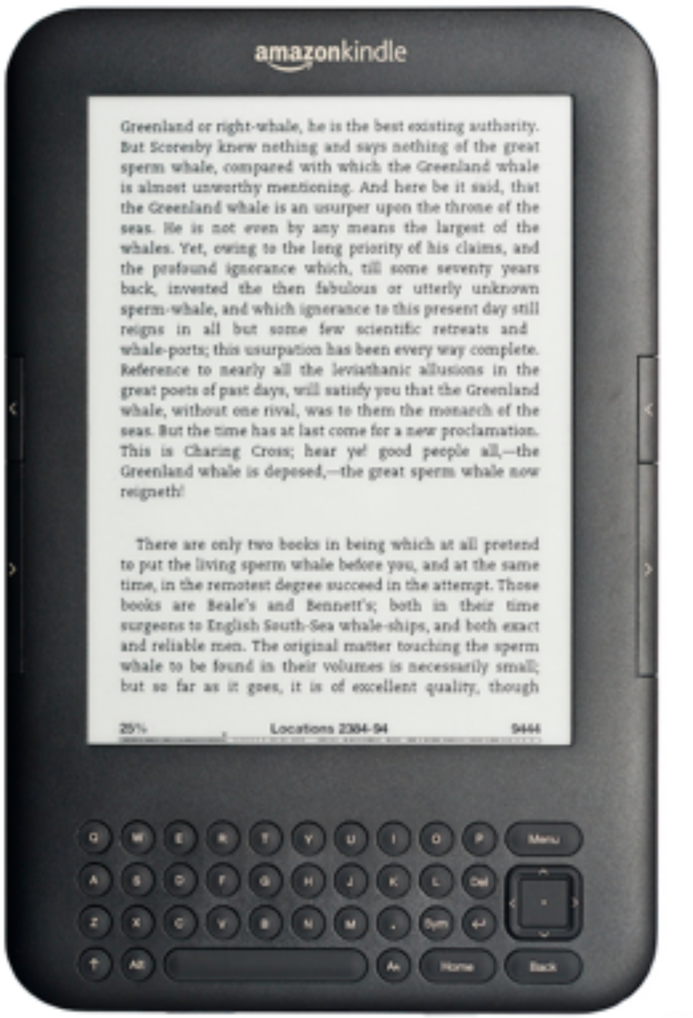
[H&B fig. 2-16]

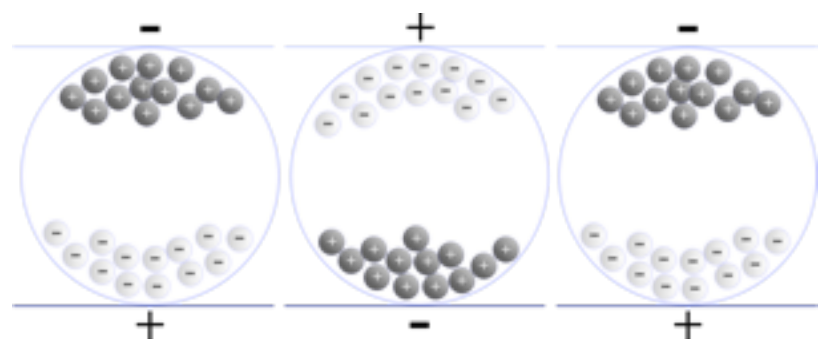
LED Displays

[Wikimedia Commons]

[Google—Nexus 4]

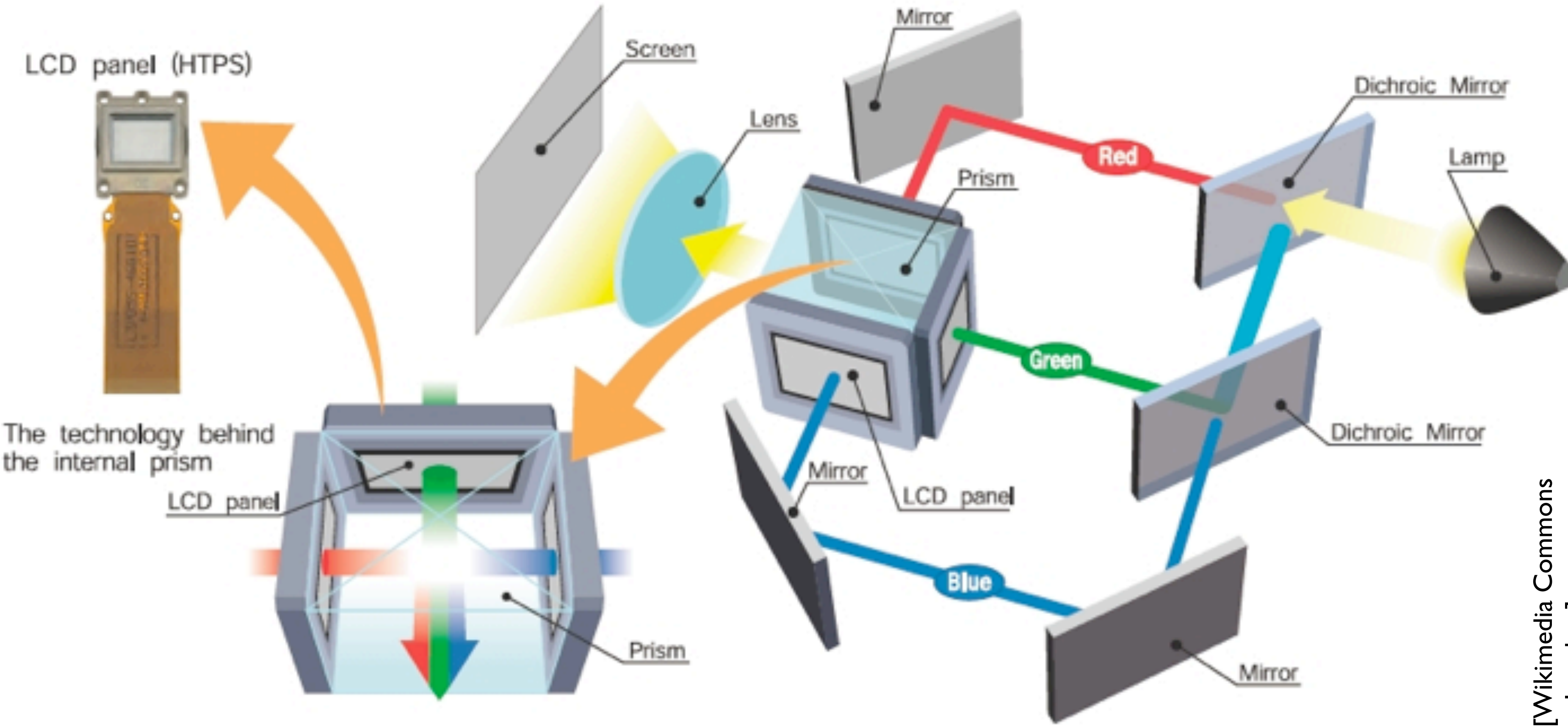
Electrophoretic (electronic ink)





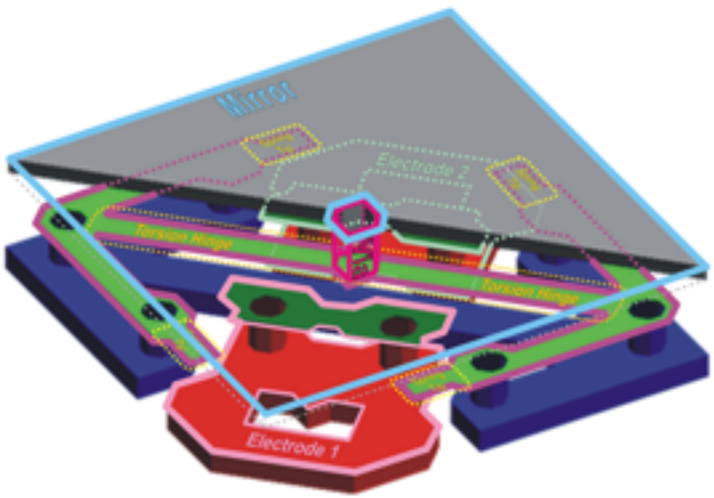
[Wikimedia Commons —Senarclens]

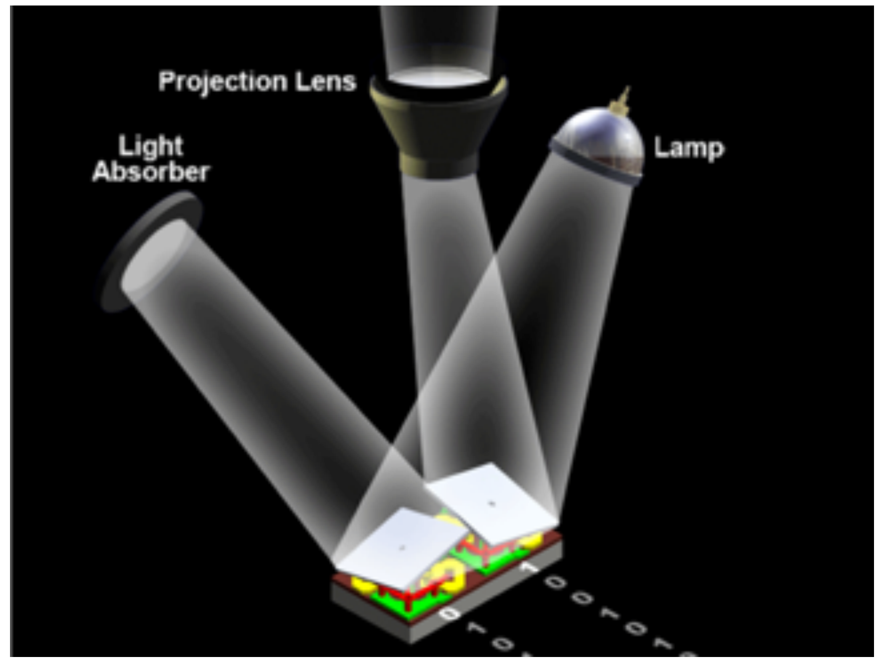
Projection displays: LCD



[Wikimedia Commons —Javachan]

Projection displays: DLP

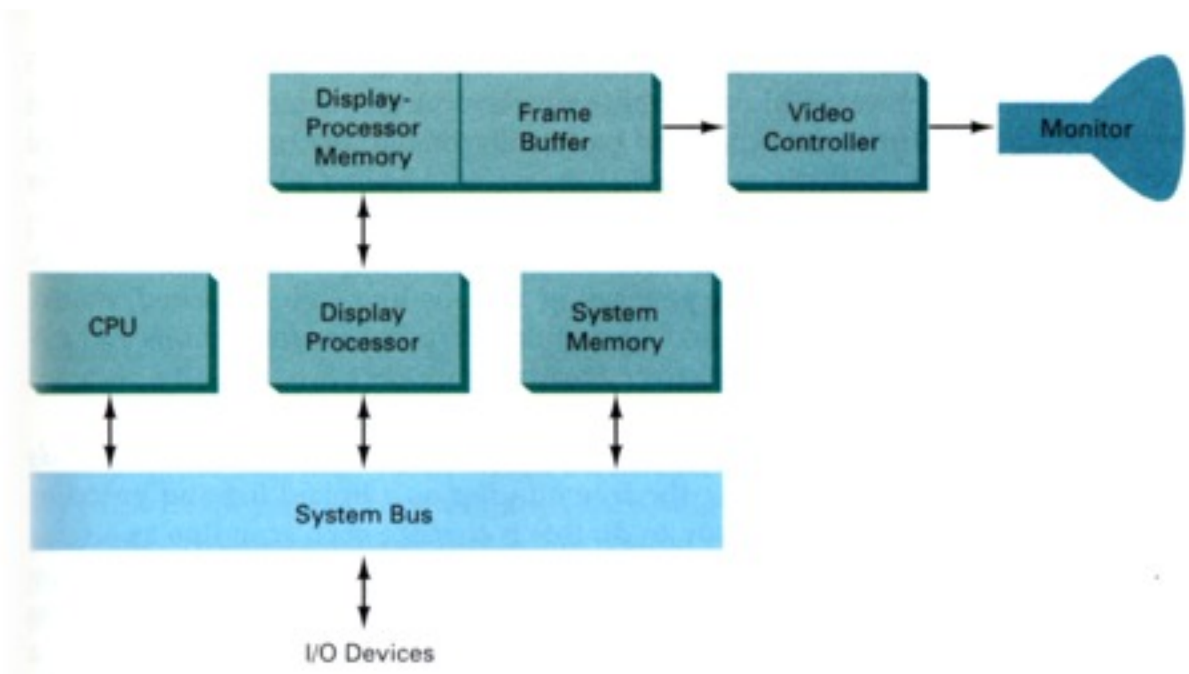




[Texas Instruments]

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 *frame buffer*



[H&B fig. 2-29]

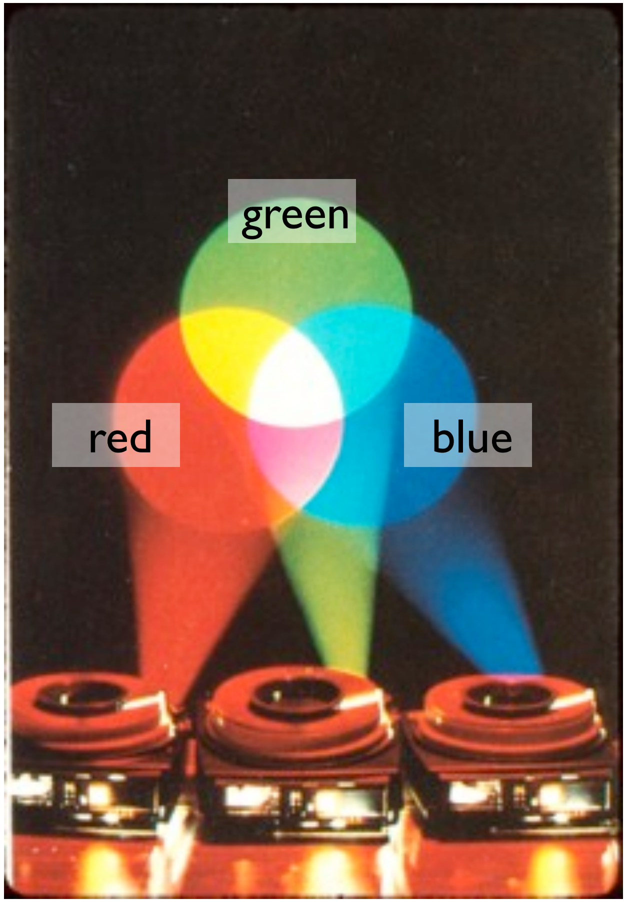
Color displays

- 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

[cs417 S02 slides]

Color displays

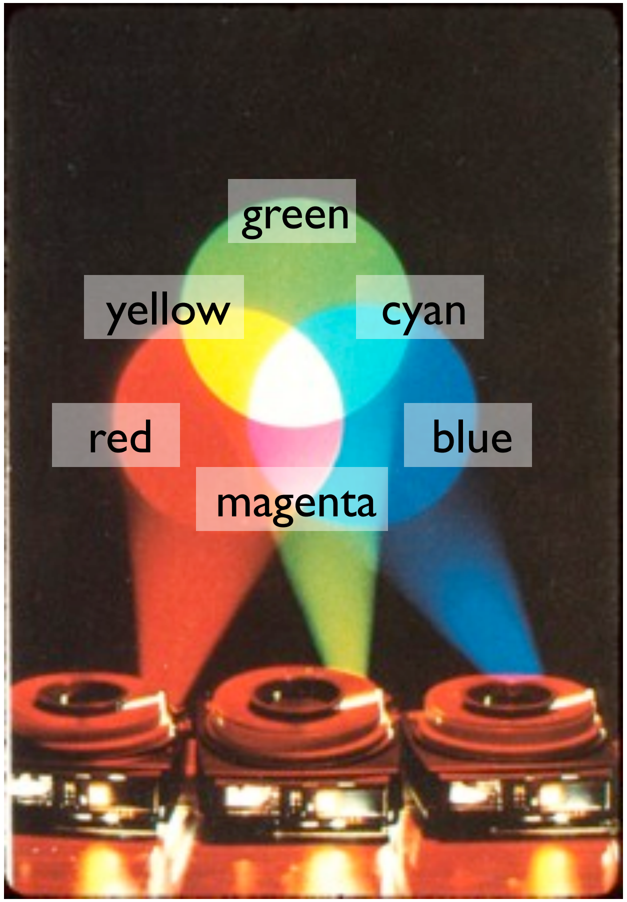
- Operating principle: humans are trichromatic
  - match any color with blend of three
  - therefore, problem reduces to producing 3 images and blending
- Additive color
  - blend images by sum
  - e.g. overlapping projection
  - e.g. unresolved dots
  - R, G, B make good primaries



[cs417 S02 slides]

Color displays

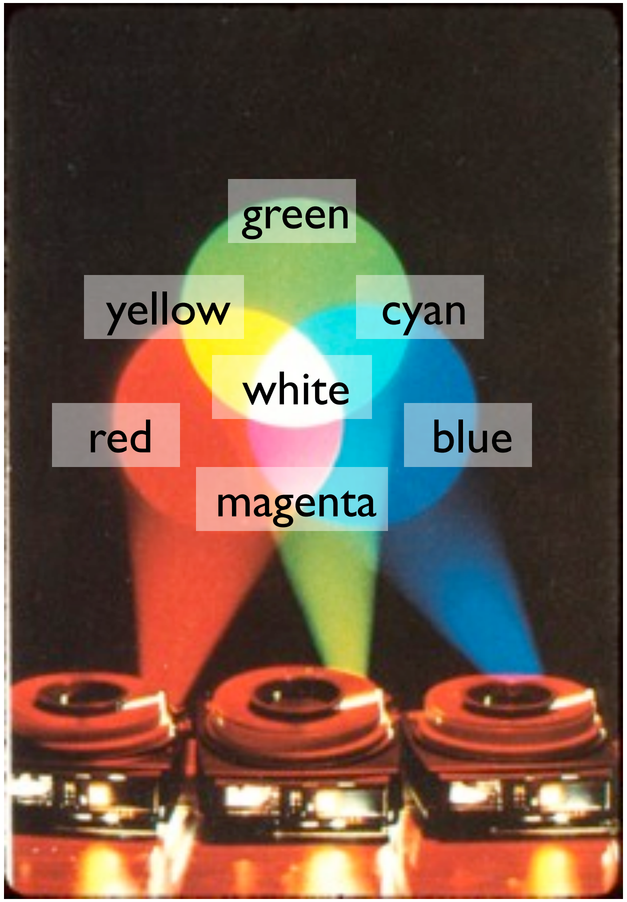
- Operating principle: humans are trichromatic
  - match any color with blend of three
  - therefore, problem reduces to producing 3 images and blending
- Additive color
  - blend images by sum
  - e.g. overlapping projection
  - e.g. unresolved dots
  - R, G, B make good primaries



[cs417 S02 slides]

Color displays

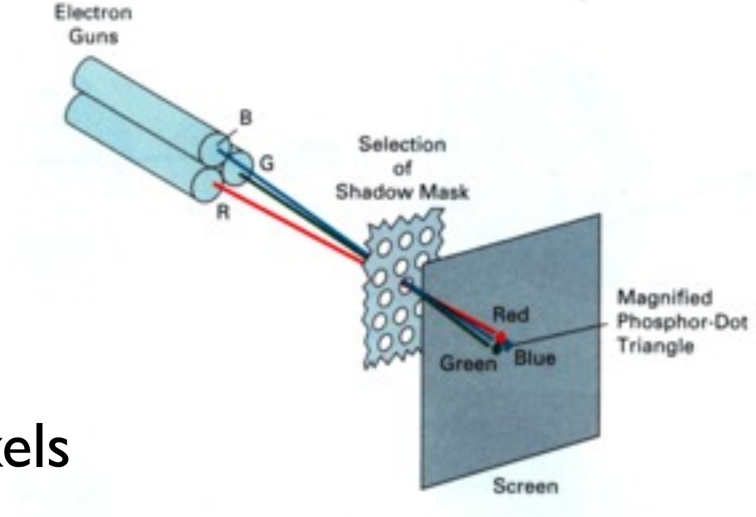
- Operating principle: humans are trichromatic
  - match any color with blend of three
  - therefore, problem reduces to producing 3 images and blending
- Additive color
  - blend images by sum
  - e.g. overlapping projection
  - e.g. unresolved dots
  - R, G, B make good primaries



[cs417 S02 slides]

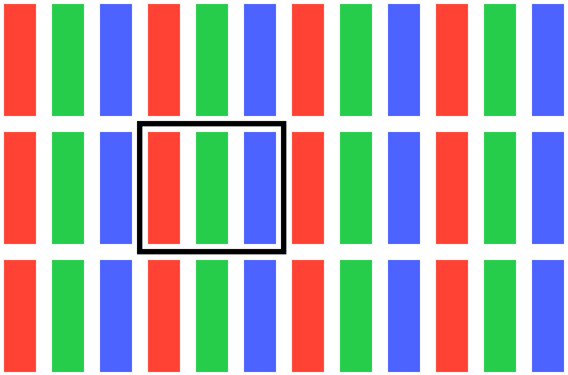
Color displays

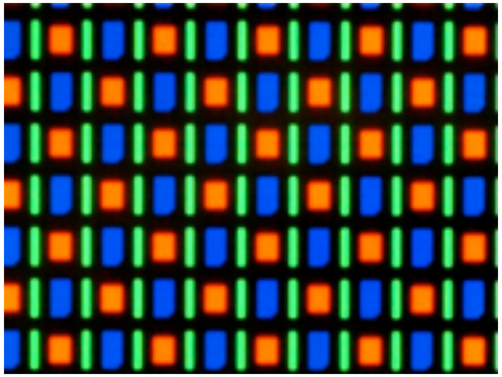
- CRT: phosphor dot pattern to produce finely interleaved color images



[H&B fig. 2-10]

- LCD, LED: interleaved R,G,B pixels

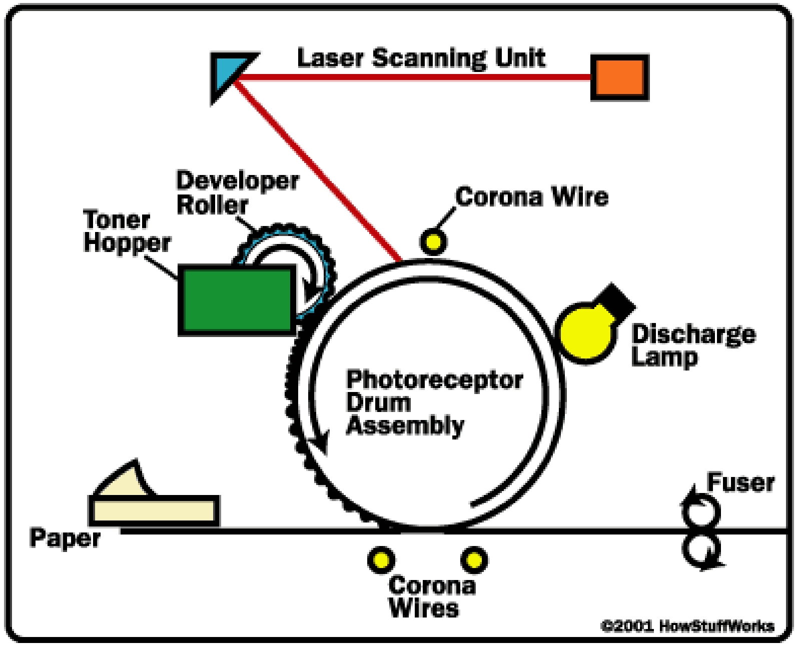




[Wikimedia Commons]

Laser printer

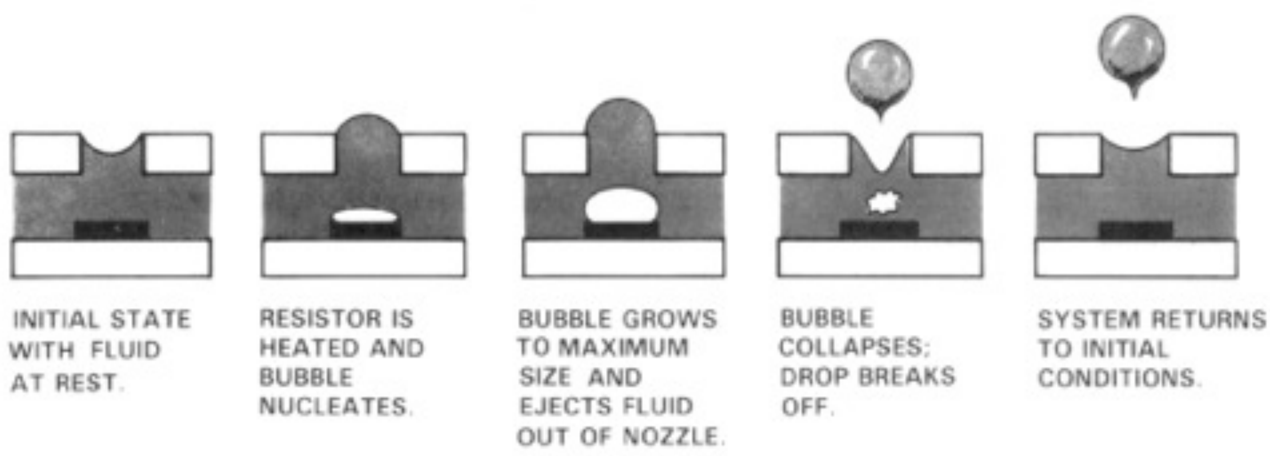
- Xerographic process
- Like a photocopier but with laser-scanned raster as source image
- Key characteristics
  - image is binary
  - resolution is high
  - very small, isolated dots are not possible



[howstuffworks.com]

Inkjet printer

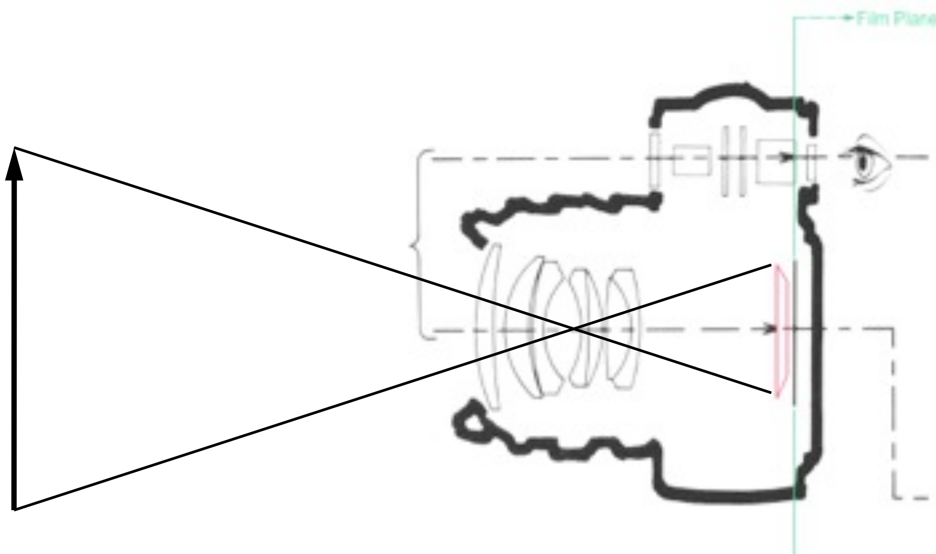
- Liquid ink sprayed in small drops
  - very small—measured in picoliters
- Head with many jets scans across paper
- Key characteristics:
  - image is binary (drop or no drop; no partial drops)
  - isolated dots are reproduced well



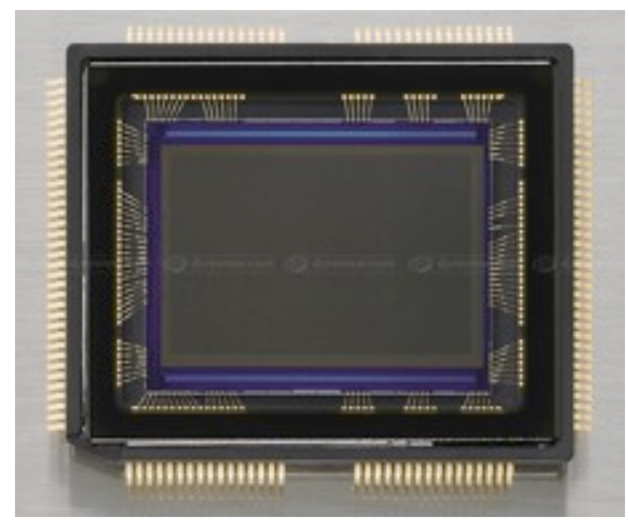
[cs417 S02 slides]

Digital camera

- A raster input device
- Image sensor contains 2D array of photosensors



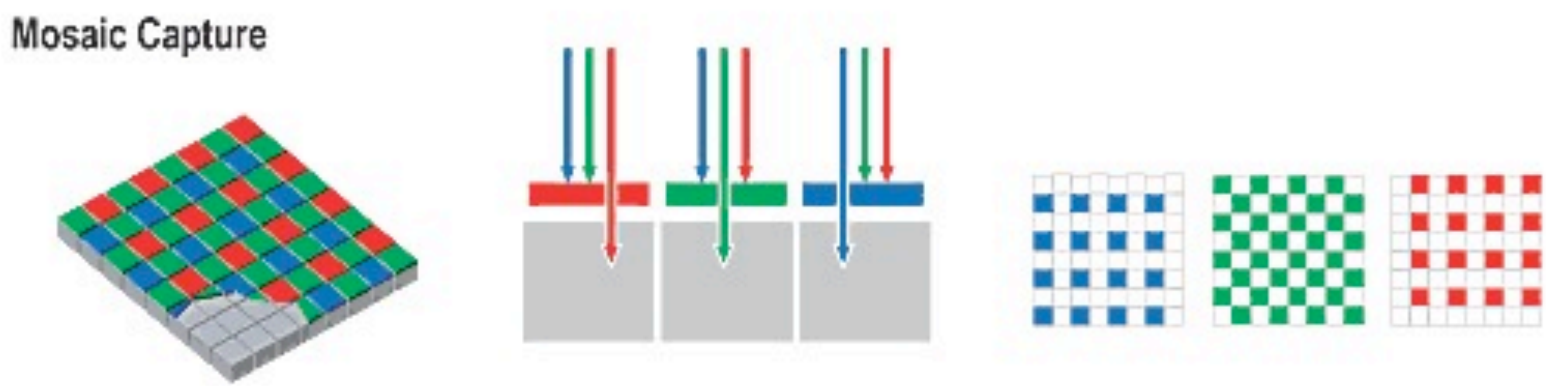
[CS 417 Spring 2002]



[dpreview.com]

Digital camera

- Color typically captured using color mosaic



[Foveon]

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!)

[Philip Greenspun]

Meaning of a raster image

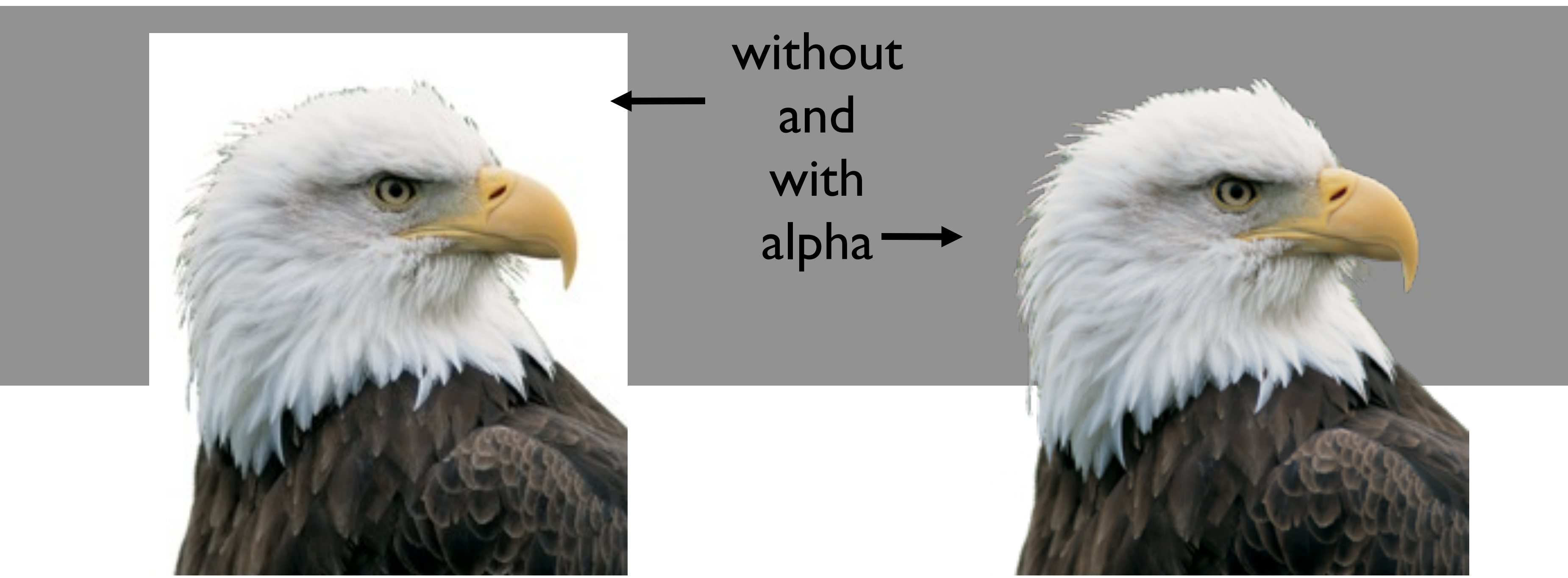
- Meaning of a given array is a function on 2D
- Define meaning of array = result of output device?
  - that is, piecewise constant for LCD, blurry for CRT
  - but: we don't have just one output device
  - but: want to define images we can't display (e.g. too big)
- Abstracting from device, problem is 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 later

Datatypes for raster images

- Bitmaps: `boolean` per pixel (1 bpp): $I : \mathbb{R}^2 \to \{0, 1\}$
  - interp. = black and white; e.g. fax
- Grayscale: integer per pixel: $I : \mathbb{R}^2 \to [0, 1]$
  - interp. = shades of gray; e.g. black-and-white print
  - precision: usually `byte` (8 bpp); sometimes 10, 12, or 16 bpp
- Color: 3 integers per pixel: $I : \mathbb{R}^2 \to [0, 1]^3$
  - 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 or 48 bpp
- Floating point: $I : \mathbb{R}^2 \to \mathbb{R}_+$ or $I : \mathbb{R}^2 \to \mathbb{R}_+^3$
  - more abstract, because no output device has infinite range
  - provides *high dynamic range* (HDR)
  - represent real scenes independent of display
  - becoming the standard intermediate format in graphics processor

Datatypes for raster images

- For color or grayscale, sometimes add *alpha* channel
  - describes transparency of images
  - more on this in a few lectures



[Adobe Photoshop sample]

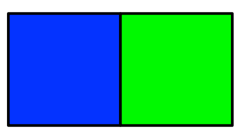
Storage requirements for images

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

Converting pixel formats

- Color to gray
  - could take one channel (blue, say)
    - leads to odd choices of gray value
  - combination of channels is better
    - but different colors contribute differently to lightness
    - which is lighter, full blue or full green?
    - good choice: gray = 0.2 R + 0.7 G + 0.1 B
    - more on this in color, later on

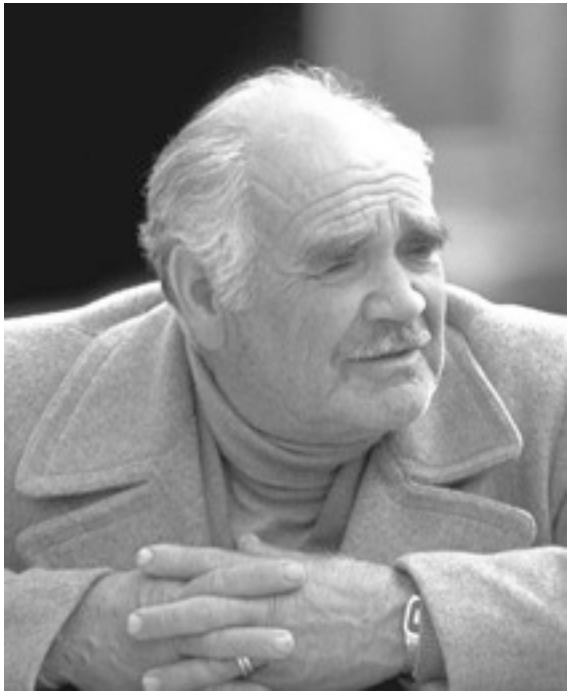
Same pixel values.



Same luminance?

Converting pixel precision

- Up is easy; down loses information—be careful

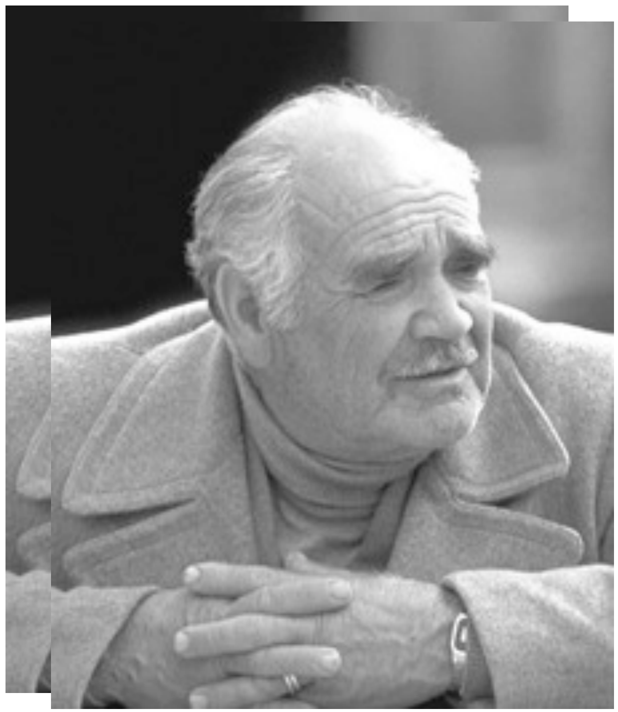


8 bpp (256 grays)

[photo: Philip Greenspun]

Converting pixel precision

- Up is easy; down loses information—be careful

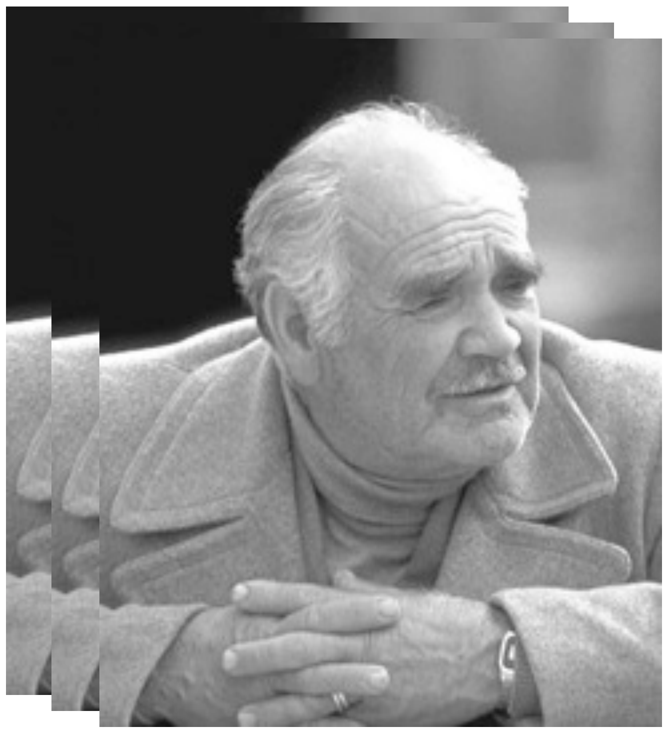


7 bpp (128 grays)

[photo: Philip Greenspun]

Converting pixel precision

- Up is easy; down loses information—be careful

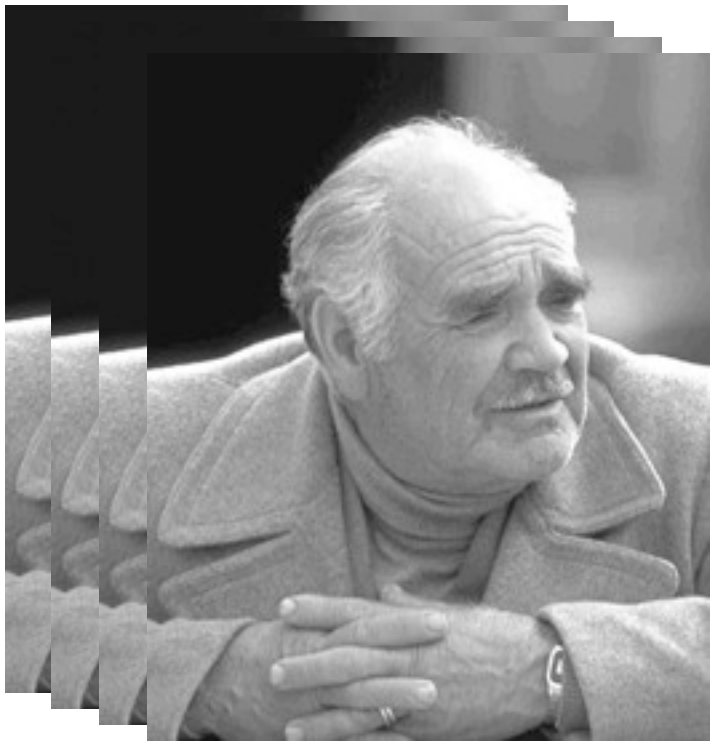


6 bpp (64 grays)

[photo: Philip Greenspun]

Converting pixel precision

- Up is easy; down loses information—be careful

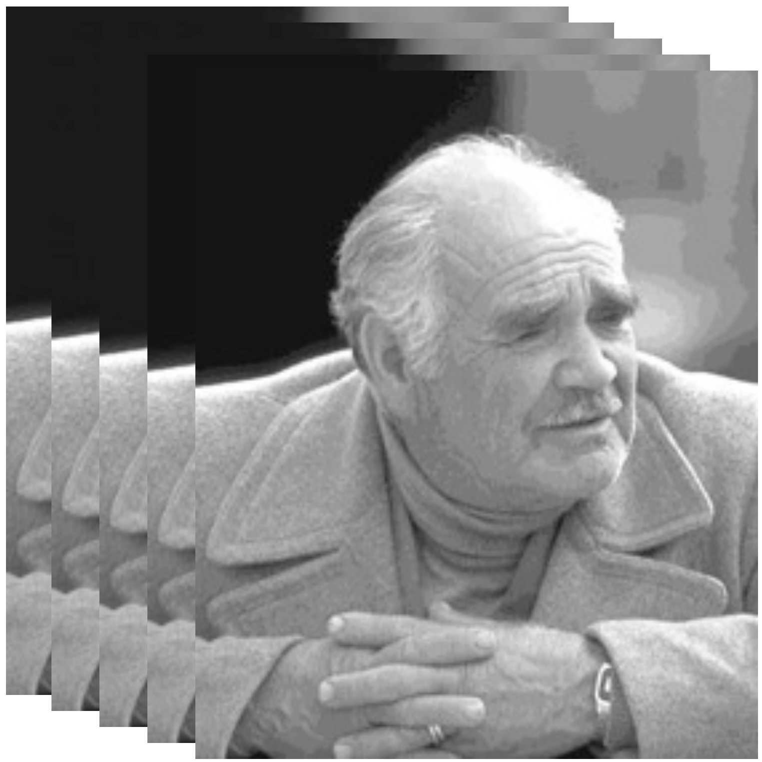


5 bpp (32 grays)

[photo: Philip Greenspun]

Converting pixel precision

- Up is easy; down loses information—be careful

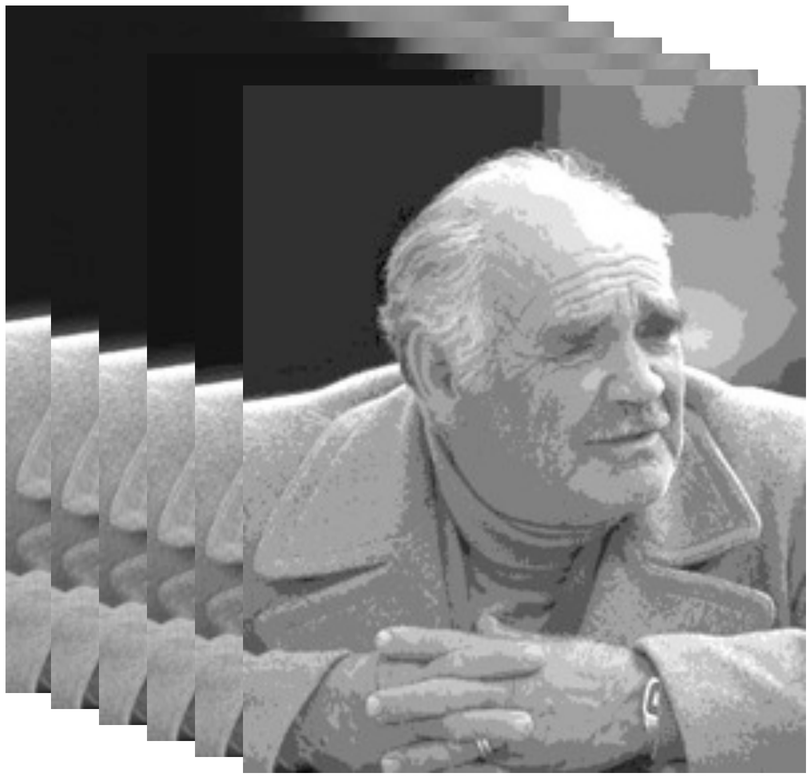


4 bpp (16 grays)

[photo: Philip Greenspun]

Converting pixel precision

- Up is easy; down loses information—be careful



3 bpp (8 grays)

[photo: Philip Greenspun]

Converting pixel precision

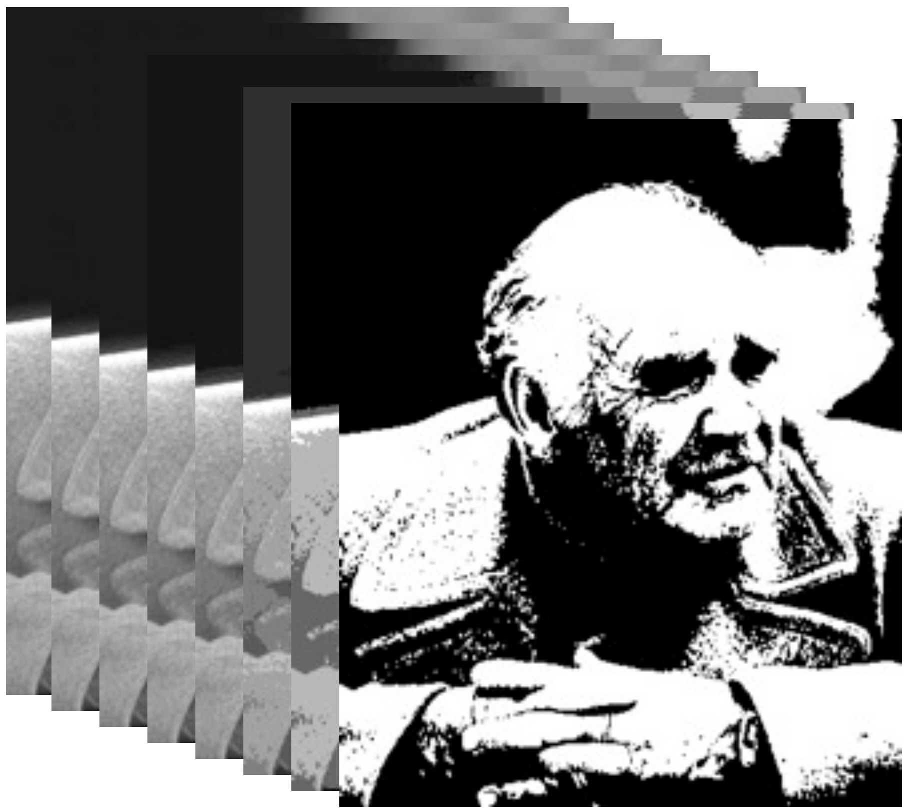
- Up is easy; down loses information—be careful

2 bpp (4 grays)

[photo: Philip Greenspun]

Converting pixel precision

- Up is easy; down loses information—be careful



1 bpp (2 grays)

[photo: Philip Greenspun]

Dithering

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

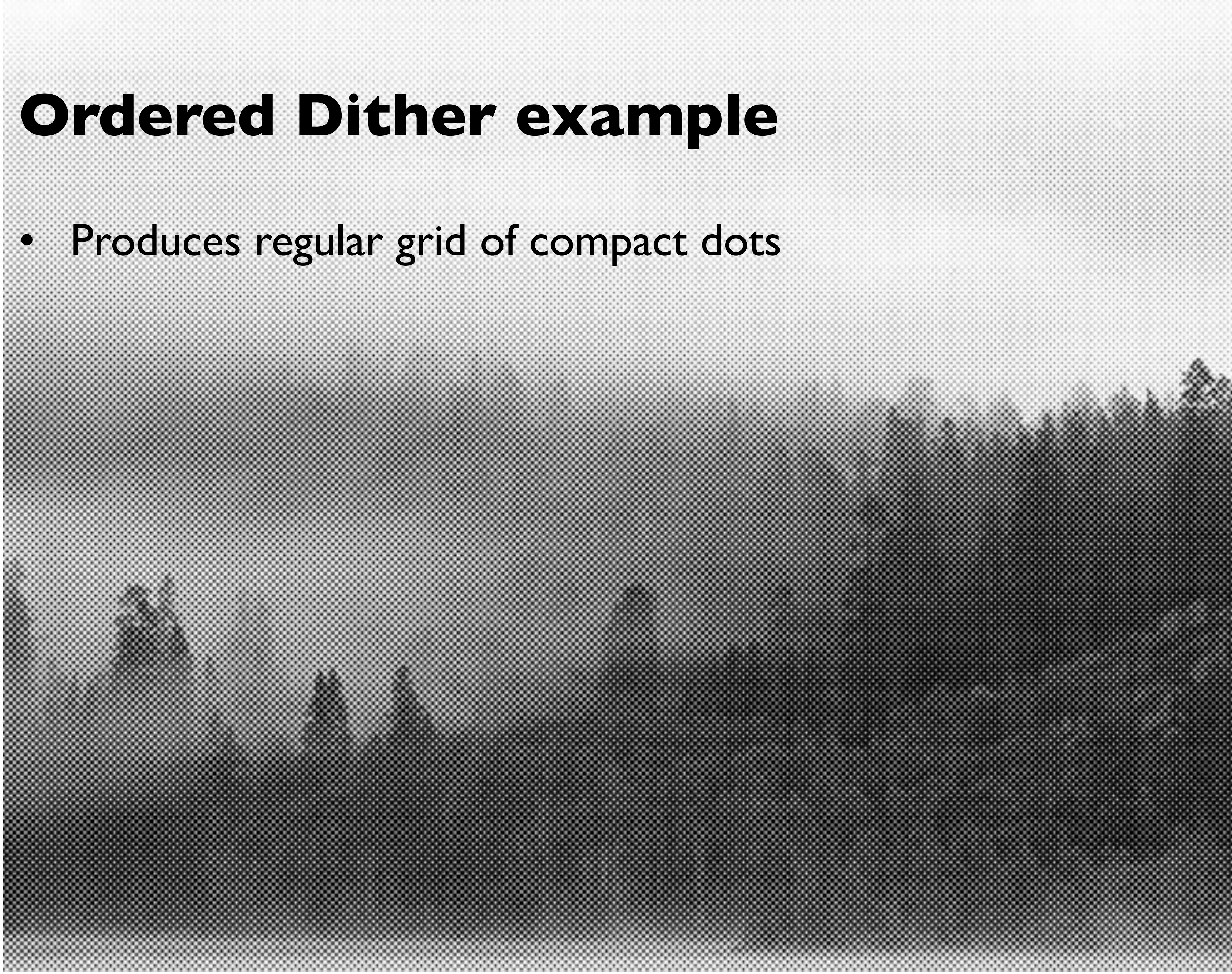
Dithering methods

- Ordered dither
  - based on traditional, optically produced halftones
  - produces larger dots
- Diffusion dither
  - takes advantage of devices that can reproduce isolated dots
  - the modern winner for desktop printing

[Philip Greenspun]

Ordered Dither example

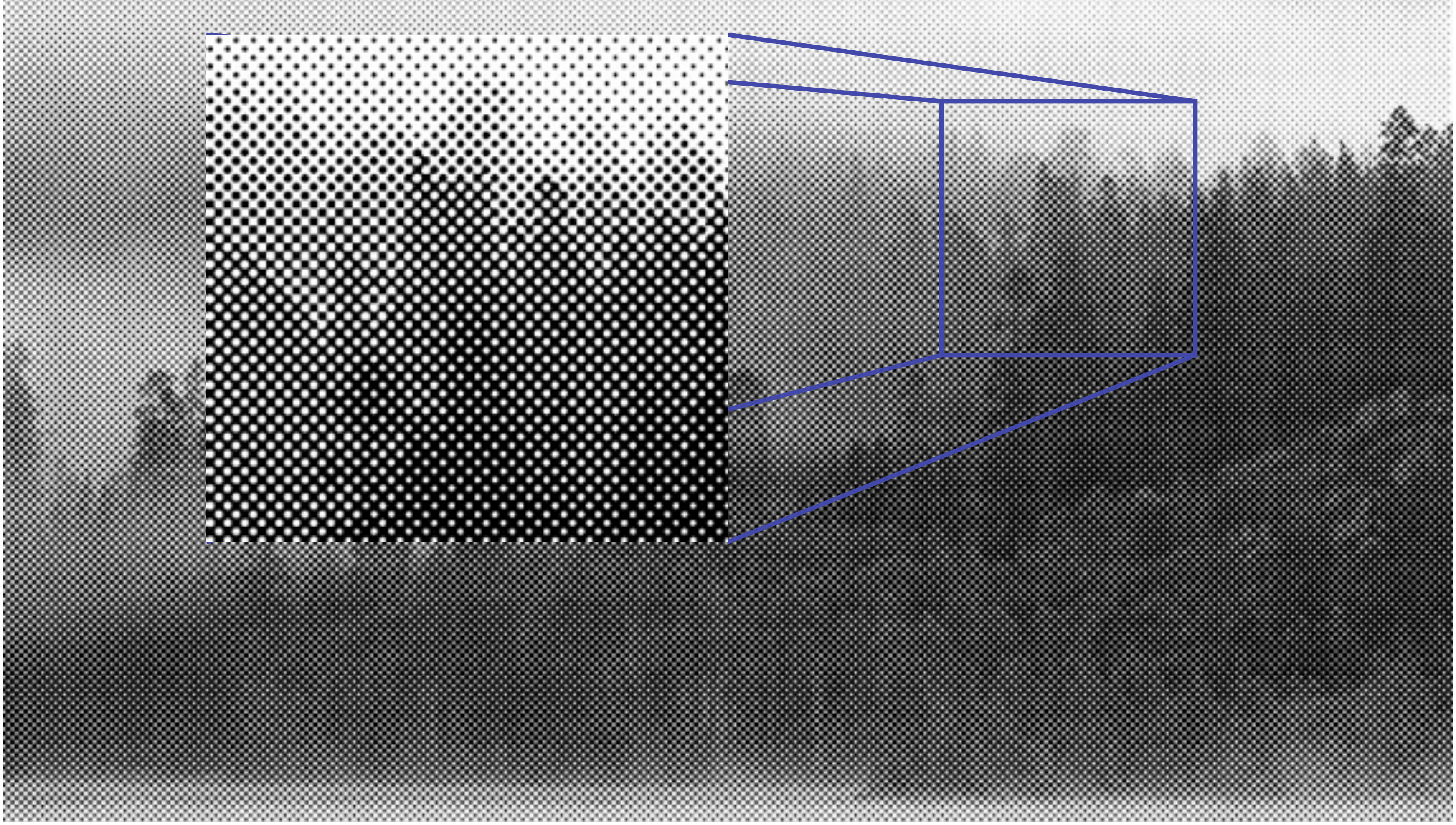
- Produces regular grid of compact dots



[photo: Philip Greenspun]

Ordered Dither example

- Produces regular grid of compact dots



[photo: Philip Greenspun]

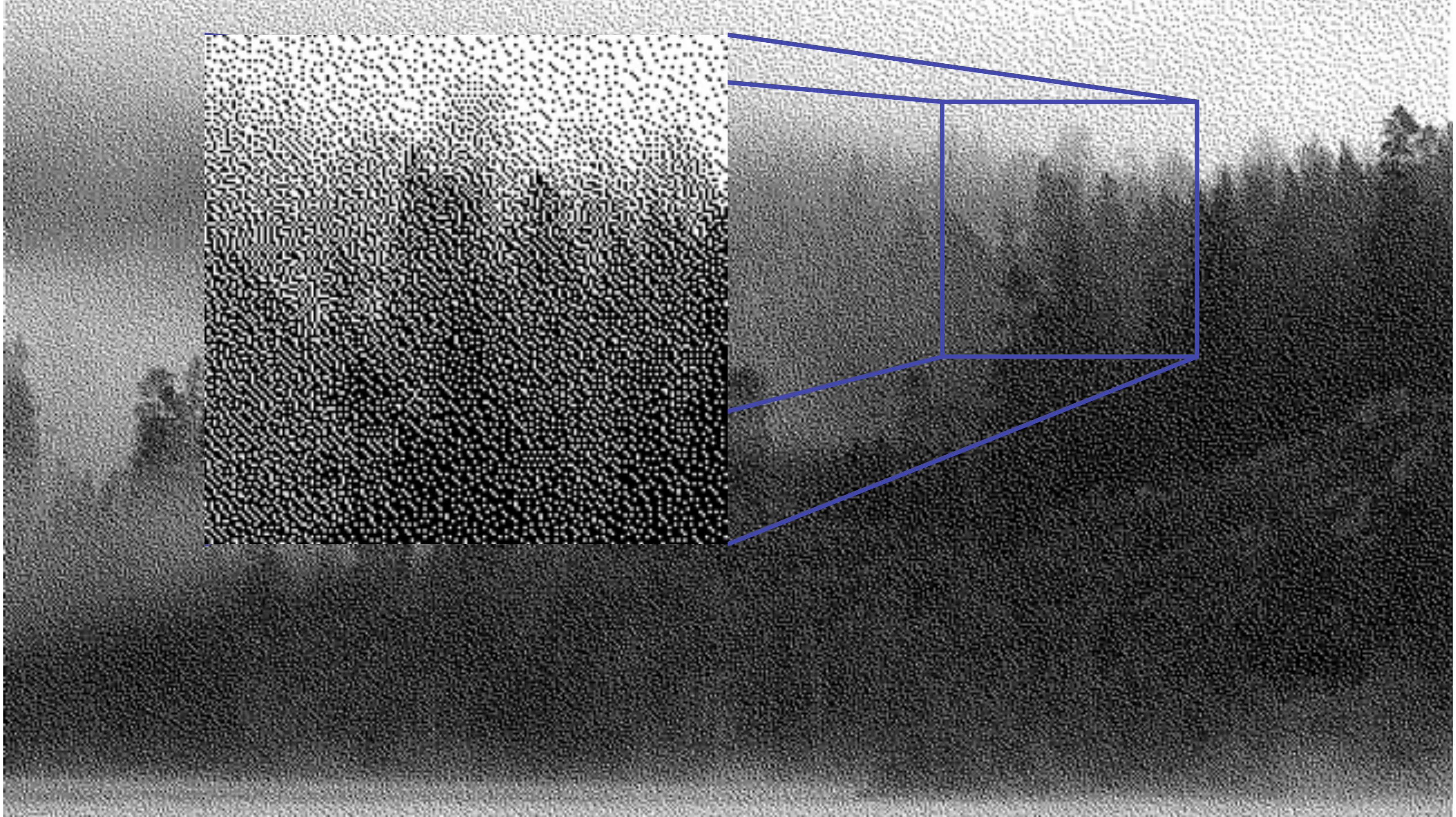
Diffusion dither

- Produces scattered dots with the right local density

[photo: Philip Greenspun]

Diffusion dither

- Produces scattered dots with the right local density



[photo: Philip Greenspun]

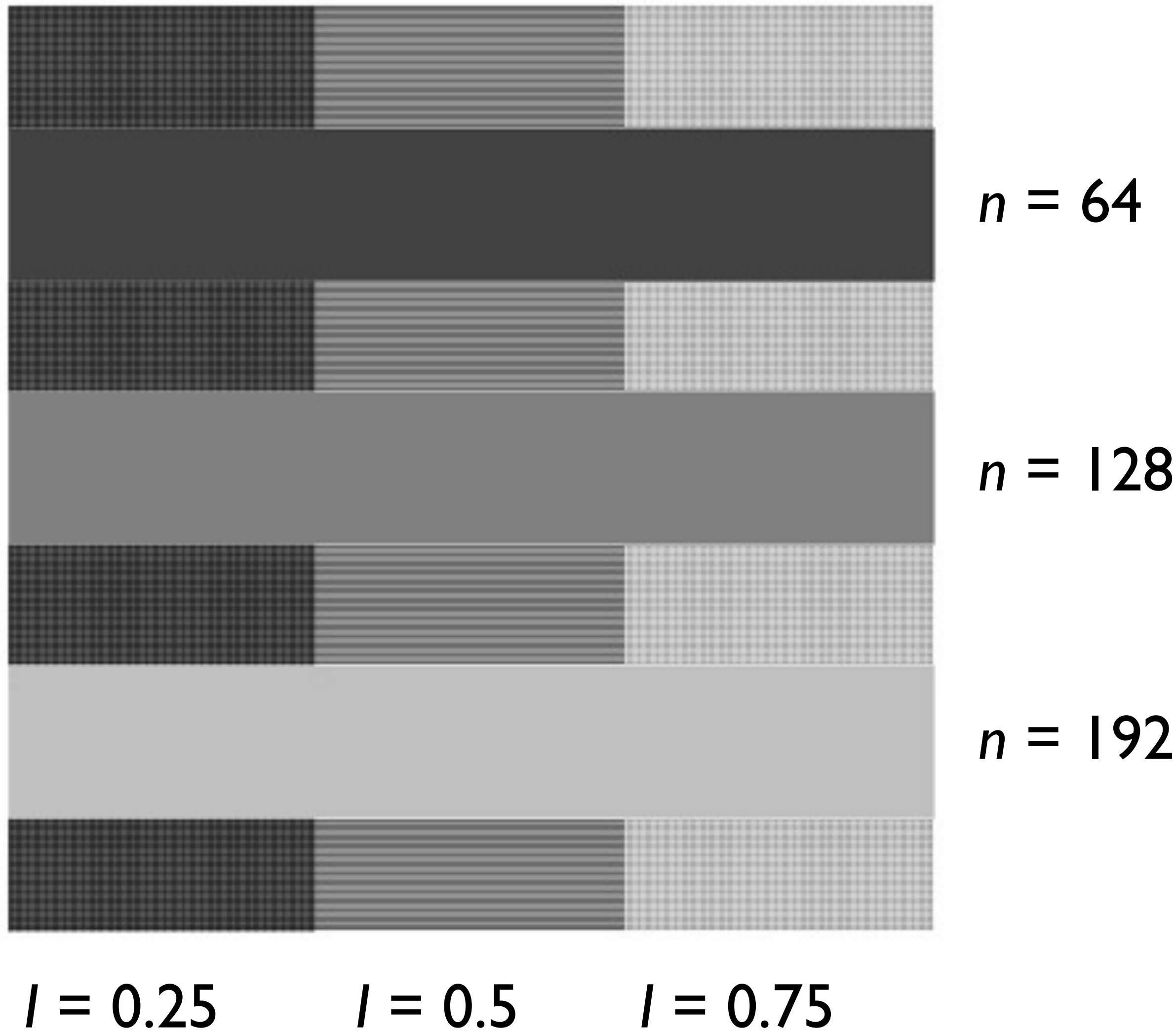
Intensity encoding in images

- What do the numbers in images (pixel values) mean?
  - they determine how bright that pixel is
  - bigger numbers are (usually) brighter
  - for floating point pixels, they directly give the intensity (in some units) — they are *linearly* related to the intensity
  - for pixels encoded in integers, this mapping is **not direct**
- *Transfer function*: function that maps input pixel value to luminance of displayed image

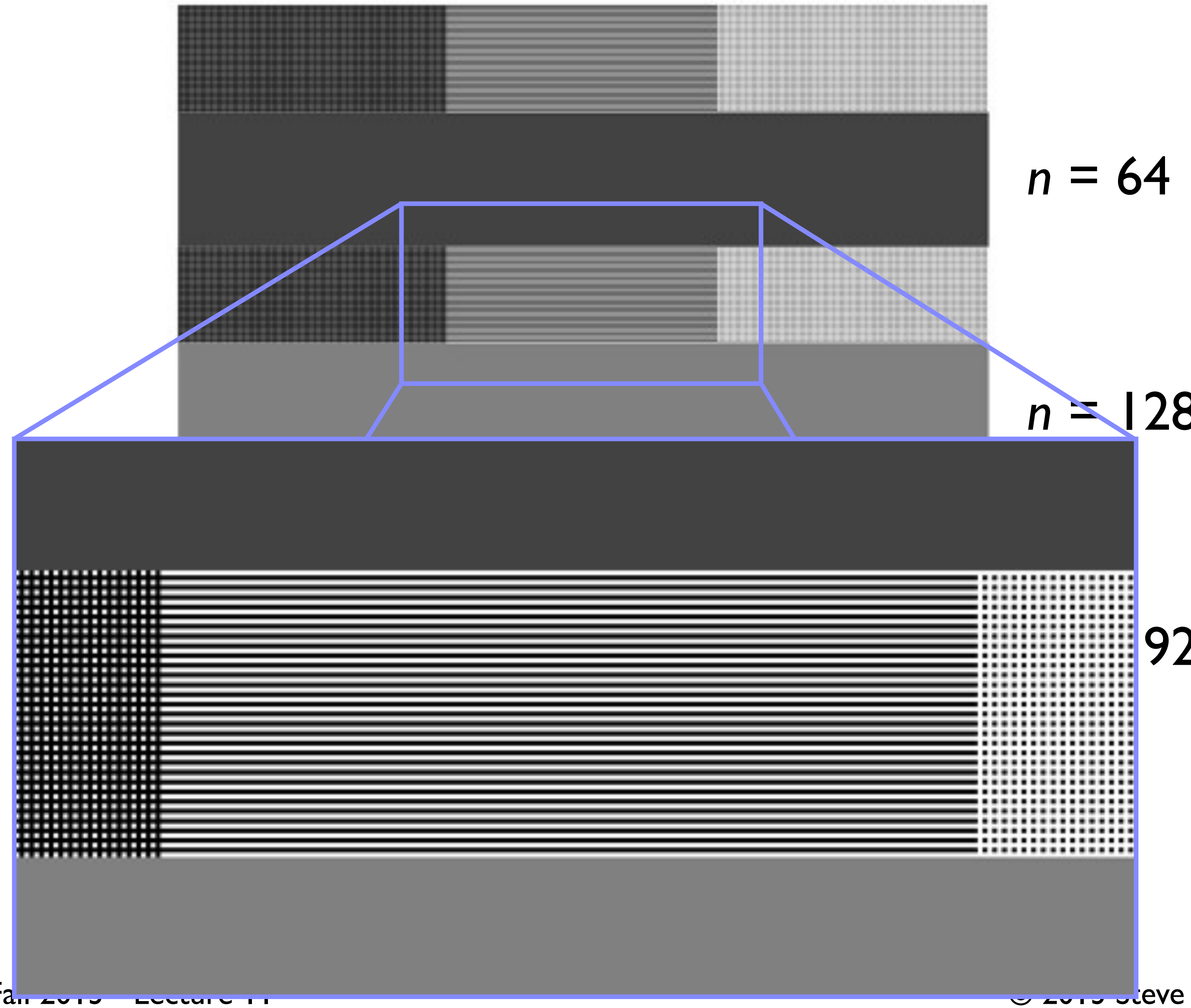
$$I = f(n) \quad f : [0, N] \to [I_{\min}, I_{\max}]$$

- What determines this function?
  - physical constraints of device or medium
  - desired visual characteristics

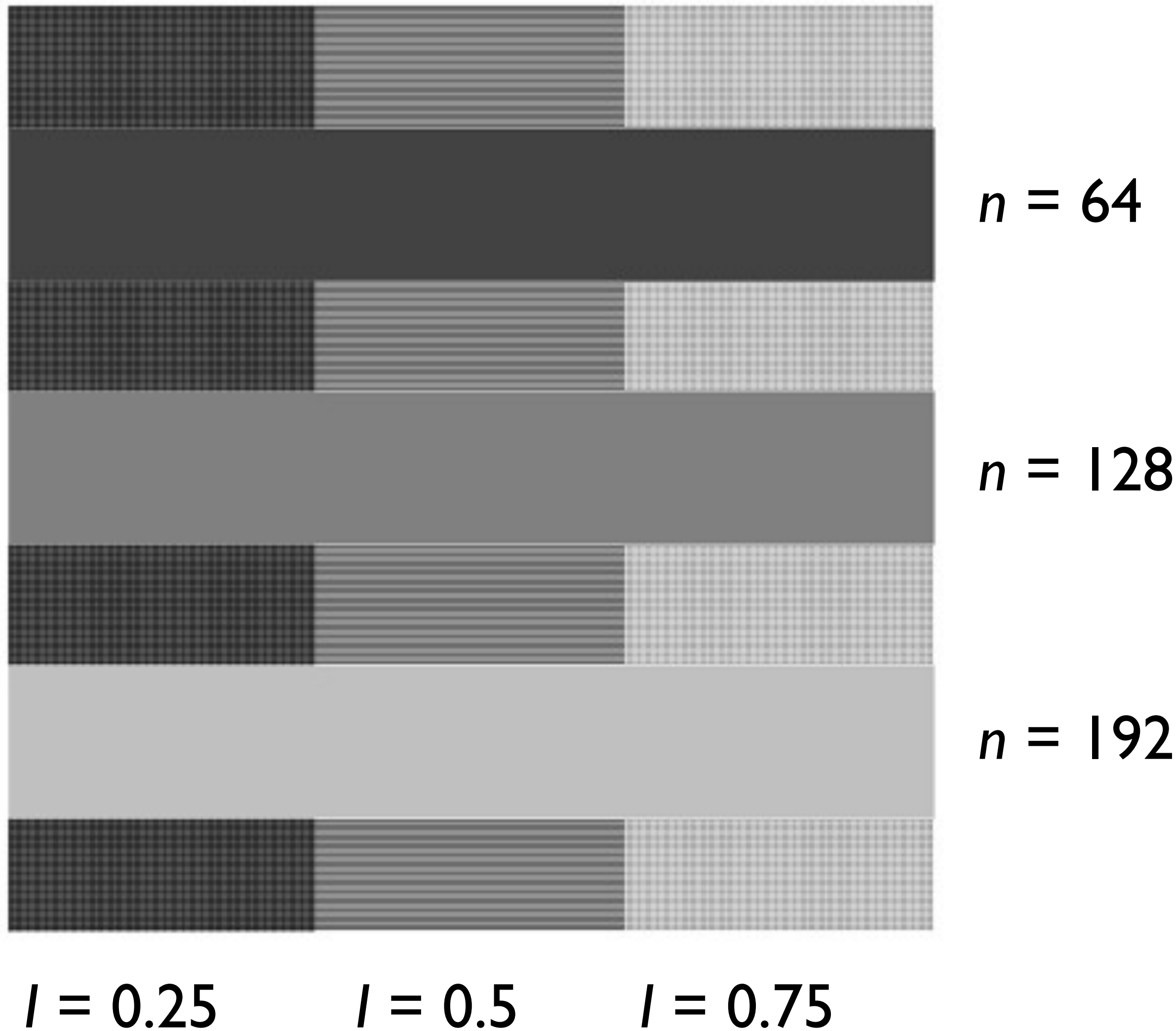
What this projector does



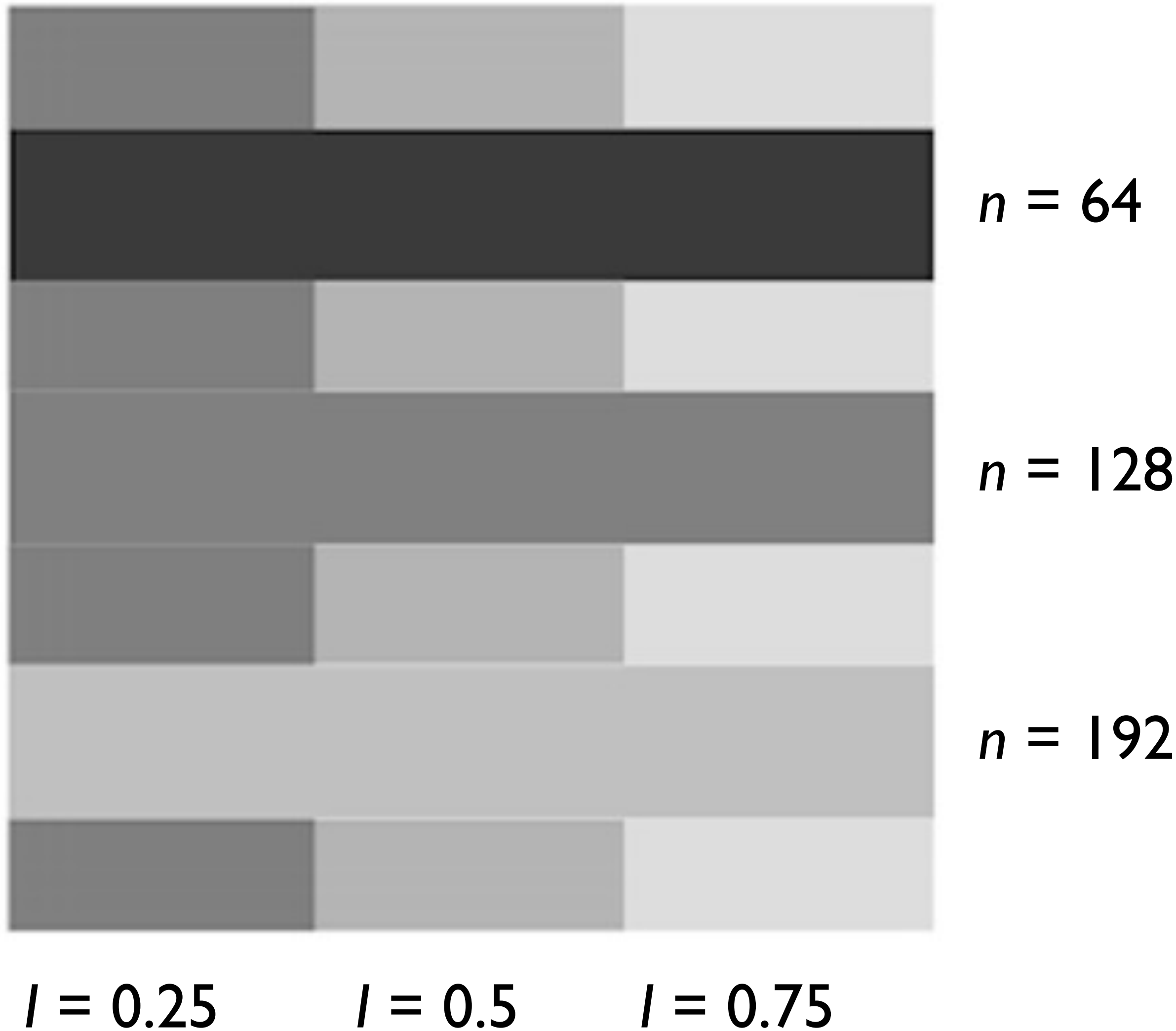
What this projector does



What this projector does

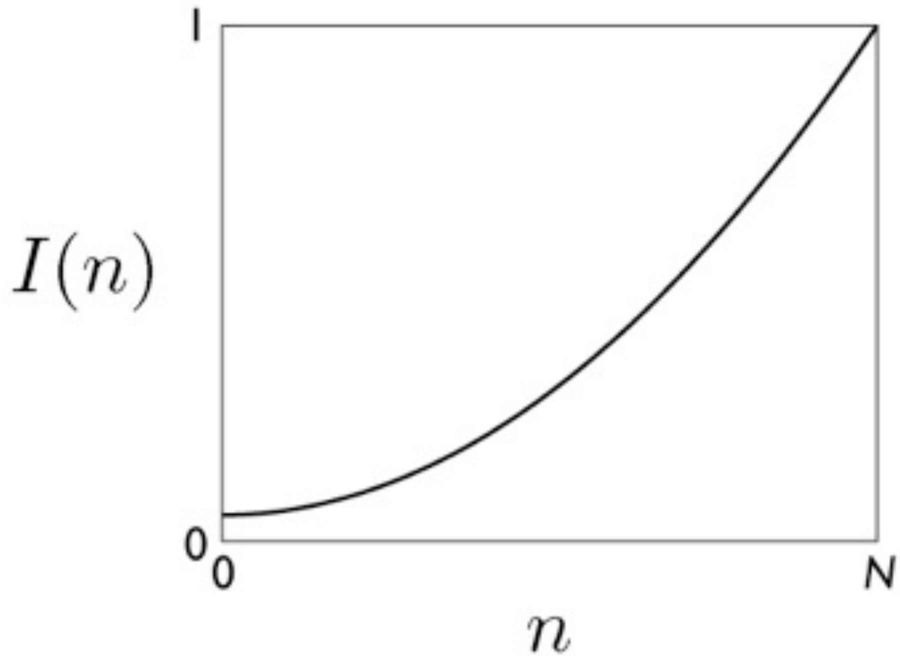


What this projector does (simulated)



What this projector does

- Something like this:



Constraints on transfer function

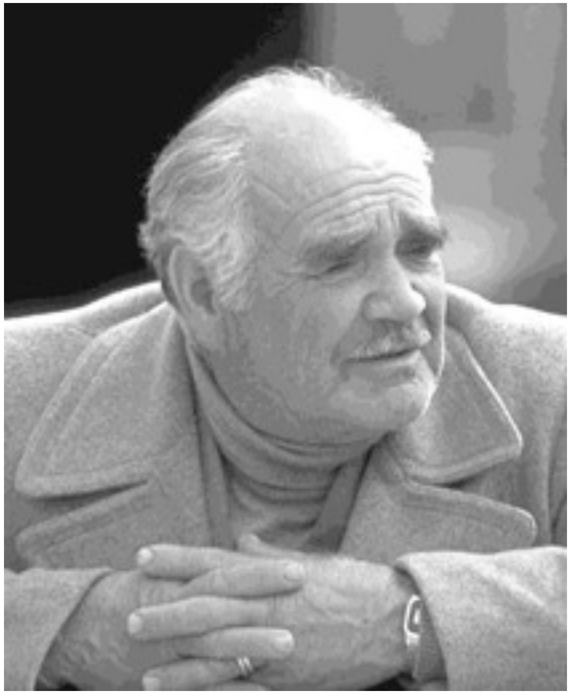
- Maximum displayable intensity, I_{max}
  - how much power can be channeled into a pixel?
    - LCD: backlight intensity, transmission efficiency (<10%)
    - projector: lamp power, efficiency of imager and optics
- Minimum displayable intensity, I_{min}
  - light emitted by the display in its "off" state
    - *e.g.* stray electron flux in CRT, polarizer quality in LCD
- Viewing flare, *k*: light reflected by the display
  - very important factor determining image contrast in practice
    - 5% of I_{max} is typical in a normal office environment [sRGB spec]
    - much effort to make very black CRT and LCD screens
    - all-black decor in movie theaters

Dynamic range

- Dynamic range $R_d = I_{\max} / I_{\min}$, or $(I_{\max} + k) / (I_{\min} + k)$
  - determines the degree of image contrast that can be achieved
  - a major factor in image quality
- Ballpark values
  - Desktop display in typical conditions: 20:1
  - Photographic print: 30:1
  - Desktop display in good conditions: 100:1
  - High-end display under ideal conditions: 1000:1
  - Digital cinema projection: 1000:1
  - Photographic transparency (directly viewed): 1000:1
  - High dynamic range display: 10,000:1

Transfer function shape

- Desirable property: the change from one pixel value to the next highest pixel value should not produce a visible contrast
  - otherwise smooth areas of images will show visible bands
- What contrasts are visible?
  - rule of thumb: under good conditions we can notice a 2% change in intensity
  - therefore we generally need smaller quantization steps in the darker tones than in the lighter tones
  - most efficient quantization is logarithmic



an image with severe *banding*

[Philip Greenspun]

How many levels are needed?

- Depends on dynamic range
  - 2% steps are most efficient:

$$0 \mapsto I_{\min}; 1 \mapsto 1.02 I_{\min}; 2 \mapsto (1.02)^2 I_{\min}; \ldots$$

  - log 1.02 is about 1/120, so 120 steps per decade of dynamic range
    - 240 for desktop display
    - 360 to print to film
    - 480 to drive HDR display
- If we want to use linear quantization (equal steps)
  - one step must be < 2% (1/50) of $I_{\min}$
  - need to get from ~0 to $I_{\min} \bullet R_d$ so need about 50 R_d levels
    - 1500 for a print; 5000 for desktop display; 500,000 for HDR display
- Moral: 8 bits is just barely enough for low-end applications
  - but only if we are careful about quantization

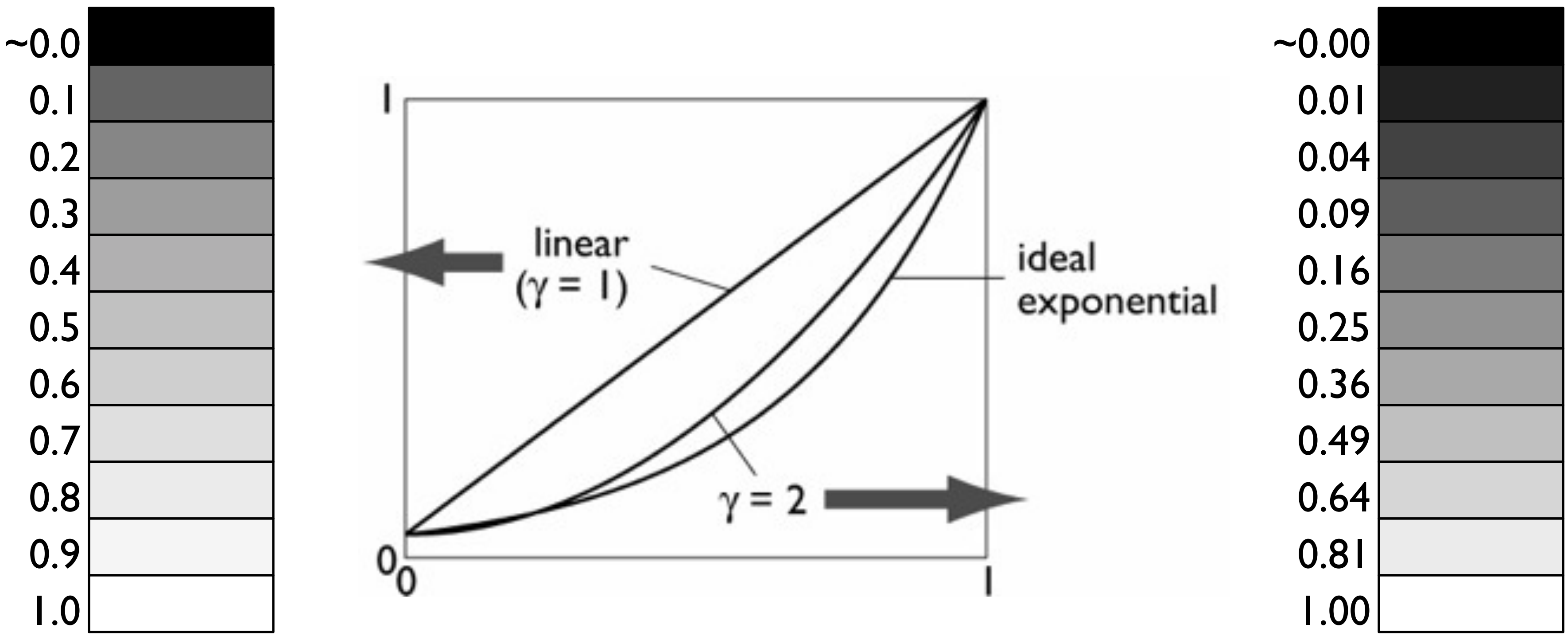
Intensity quantization in practice

- Option 1: linear quantization $I(n) = (n/N)\, I_{\max}$
  - pro: simple, convenient, amenable to arithmetic
  - con: requires more steps (wastes memory)
  - need 12 bits for any useful purpose; more than 16 for HDR
- Option 2: power-law quantization $I(n) = (n/N)^\gamma\, I_{\max}$
  - pro: fairly simple, approximates ideal exponential quantization
  - con: need to linearize before doing pixel arithmetic
  - con: need to agree on exponent
  - 8 bits are OK for many applications; 12 for more critical ones
- Option 2: floating-point quantization $I(x) = (x/w)\, I_{\max}$
  - pro: close to exponential; no parameters; amenable to arithmetic
  - con: definitely takes more than 8 bits
  - 16–bit "half precision" format is becoming popular

Why gamma?

- Power-law quantization, or *gamma correction* is most popular
- Original reason: CRTs are like that
  - intensity on screen is proportional to (roughly) voltage^2
- Continuing reason: inertia + memory savings
  - inertia: gamma correction is close enough to logarithmic that there's no sense in changing
  - memory: gamma correction makes 8 bits per pixel an acceptable option

Gamma quantization



- Close enough to ideal perceptually uniform exponential

Gamma correction

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

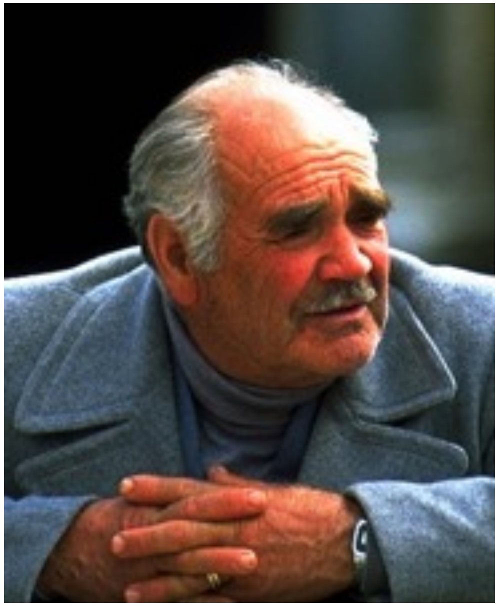
$$I(n) = (n/N)^\gamma$$

(where $N = 2^n - 1$ in *n* bits). Solving for *n*:

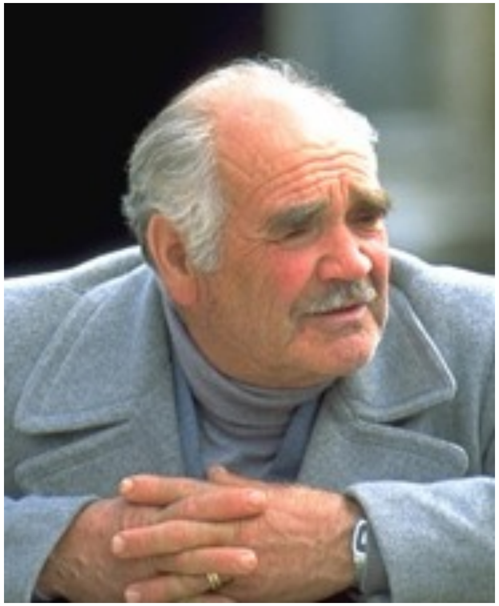
$$n(I) = NI^{\frac{1}{\gamma}}$$

- This is the "gamma correction" recipe that has to be applied when computed values are converted to 8 bits for output
  - failing to do this (implicitly assuming gamma = 1) results in dark, oversaturated images

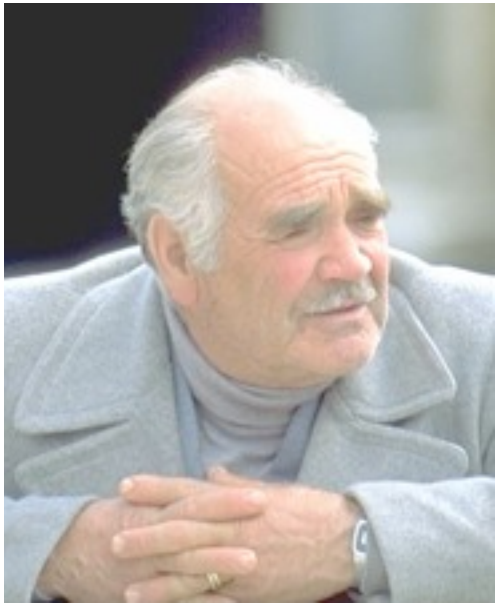
Gamma correction



corrected for
γ lower than
display



OK



corrected for
γ higher than
display

[Philip Greenspun]

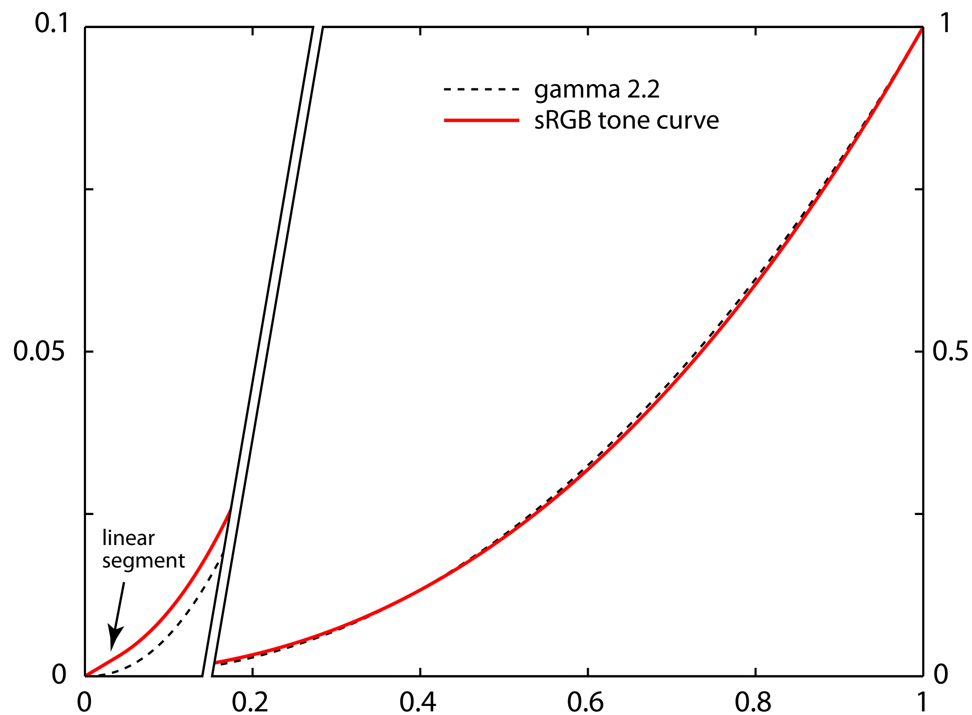
sRGB quantization curve

- The predominant standard for "casual color" in computer displays
  - consistent with older typical practice
  - designed to work well under imperfect conditions
  - these days all monitors are calibrated to sRGB by default
  - in practice, usually defines what your pixel values mean

$$I(C) = \begin{cases} \frac{C}{12.92}, & C \leq 0.04045 \\ \left(\frac{C+a}{1+a}\right)^{2.4}, & C > 0.04045 \end{cases}$$

$$C = n/N$$

$$a = 0.055$$



[derived from a figure by Dick Lyon]

Converting from HDR to LDR

- "High dynamic range" — pixels can be arbitrarily bright or dark
- "Low dynamic range" — there are limits on the min and max
- Simplest solution: just scale and clamp

$$I_{\mathrm{LDR}} = \min(1, aI) I_{\max}$$

- More flexible: introduce a contrast control

$$I_{\mathrm{LDR}} = \min(1, aI^{\gamma}) I_{\max}$$

- Scale factor *a* is "exposure"
  - often quoted on a power-of-2 scale

image: Paul Debevec

exposure:
-8 stops

exposure:
+0 stops

image: Paul Debevec

exposure:
+6 stops

image: Paul Debevec

Transfer functions for LDR display

- Not a new problem at all; photography has been dealing with this for a century
- In film it is the "D log E" curve: density vs. log exposure

the first to state the relation between density and exposure. The underexposure part (below log $E = 8.9$) is called the "toe." The overexposure region (above log $E = 1.1$) is sometimes referred to as the "shoulder."

Since, in the correct representation of the light and shade of the subject photographed, the opacity of the negative should be proportional to the quantity of light coming from the subject, it

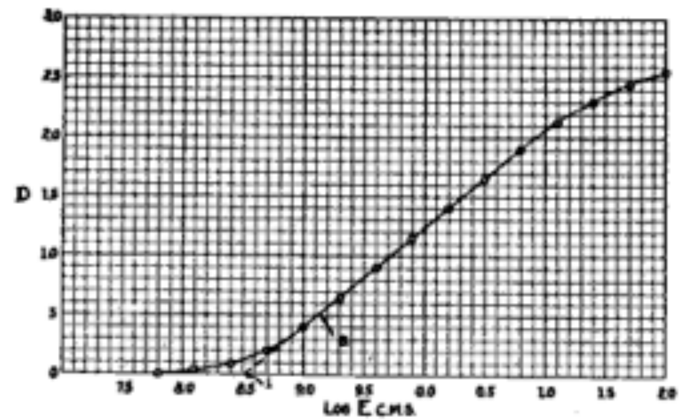


FIG. 6—*Density-exposure curve showing toe, straightline, and shoulder*

follows that the time of exposure should be such as to give densities on the plate which lie on the straight-line portion of the density-exposure curve. It is found that if the exposure is too short, there is no detail in the shadows, although there may be a

[National Bureau of Standards, 1922]

Ward | Fattal | LCIS (Tumblin)