Images and Displays
or
Electronic Imaging 101

CS4620

Fall 2020

Images and Displays

1. What is an image?

CS4620

Fall 2020

What is an image?

- A photographic print
- 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

$$I : \mathbb{R}^2 \to \ldots$$

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

Images and Displays

2. Datatypes

CS4620

Fall 2020

Datatypes for raster images

- If an image is a function with image space as its domain:

$$I : \mathbb{R}^2 \to \ldots$$

- …then what is its range?
- What type of data do we store, and what does it mean?

Integer (fixed point) image datatypes

Integer (fixed point) image datatypes

- Bitmaps: `boolean` per pixel (1 bpp): $I : \mathbb{R}^2 \rightarrow \{0, 1\}$
  - interp. = black and white; e.g. fax

Integer (fixed point) image datatypes

- 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. X-ray, black-and-white photo
  - precision: usually `byte` (8 bpp); sometimes 10, 12, or 16 bpp

Integer (fixed point) image datatypes

- 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. X-ray, black-and-white photo
  – 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 photograph
  – precision: usually `byte[3]` (24 bpp)
  – sometimes 30 or 36 or 48 bpp

Integer (fixed point) image datatypes

- 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. X-ray, black-and-white photo
  - 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 photograph
  - precision: usually `byte[3]` (24 bpp)
  - sometimes 30 or 36 or 48 bpp
- Usually, think of these as fixed-point numbers in [0,1]
  - normally closed on both sides
  - pixel value n maps to real value $n / N = n / (2^b - 1)$
  - e.g. 8-bit to float is `x` = `n` / `255.0`
  - e.g. float to 8-bit is `n` = `round(clamp(x, 0.0, 1.0) * 255.0)`

Floating point image datatypes

- Floating point: $I : \mathbb{R}^2 \rightarrow \mathbb{R}_+$ or $I : \mathbb{R}^2 \rightarrow \mathbb{R}_+^3$
  - more abstract; no output device has infinite range
  - provides *high dynamic range* (HDR)
  - represent real scenes independent of display

image: Paul Debevec

exposure:
+0 stops

exposure:
-8 stops

image: Paul Debevec

exposure:
+6 stops

image: Paul Debevec

Floating point image datatypes

- Floating point: $I : \mathbb{R}^2 \to \mathbb{R}_+$ or $I : \mathbb{R}^2 \to \mathbb{R}^3_+$
  - provides *high dynamic range* (HDR)
  - represent real scenes independent of display
  - 16 (half precision) or 32 (single precision, $\mathtt{float}$) bpp for grayscale
  - …and therefore 48 or 96 bpp for RGB color

Floating point image datatypes

- Floating point: $I : \mathbb{R}^2 \rightarrow \mathbb{R}_+$ or $I : \mathbb{R}^2 \rightarrow \mathbb{R}_+^3$
  - provides *high dynamic range* (HDR)
  - represent real scenes independent of display
  - 16 (half precision) or 32 (single precision, $\mathtt{float}$) bpp for grayscale
  - …and therefore 48 or 96 bpp for RGB color
- …in HDR photography
  - maintain full dynamic range of camera sensor
  - build more range from multiple exposures

Floating point image datatypes

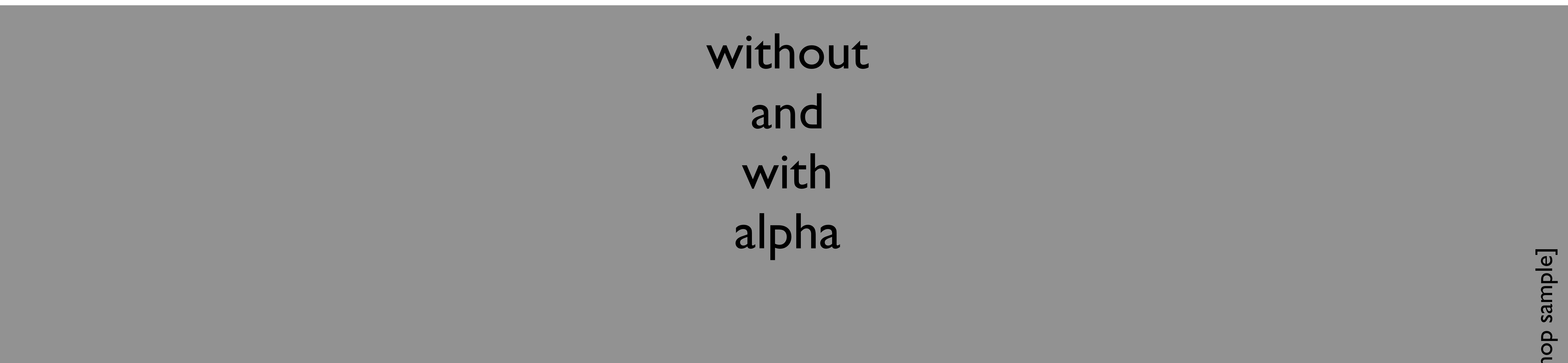
- Floating point: $I : \mathbb{R}^2 \rightarrow \mathbb{R}_+$ or $I : \mathbb{R}^2 \rightarrow \mathbb{R}_+^3$
  - provides *high dynamic range* (HDR)
  - represent real scenes independent of display
  - 16 (half precision) or 32 (single precision, `float`) bpp for grayscale
  - …and therefore 48 or 96 bpp for RGB color
- …in HDR photography
  - maintain full dynamic range of camera sensor
  - build more range from multiple exposures
- …in high-end realistic rendering
  - compute images with 32 bit floats in memory
  - store more compact formats in image files

Floating point image datatypes

- Floating point: $I : \mathbb{R}^2 \rightarrow \mathbb{R}_+$ or $I : \mathbb{R}^2 \rightarrow \mathbb{R}_+^3$
  - provides *high dynamic range* (HDR)
  - represent real scenes independent of display
  - 16 (half precision) or 32 (single precision, `float`) bpp for grayscale
  - …and therefore 48 or 96 bpp for RGB color
- …in HDR photography
  - maintain full dynamic range of camera sensor
  - build more range from multiple exposures
- …in high-end realistic rendering
  - compute images with 32 bit floats in memory
  - store more compact formats in image files
- …in real-time rendering
  - half-precision is a common intermediate format in GPUs
  - allows for image processing without clipping until the final result

Datatypes for raster images

- For color or grayscale, sometimes add *alpha* channel
  - describes transparency of images
  - more on this a bit later



[Adobe Photoshop sample]

Datatypes for raster images

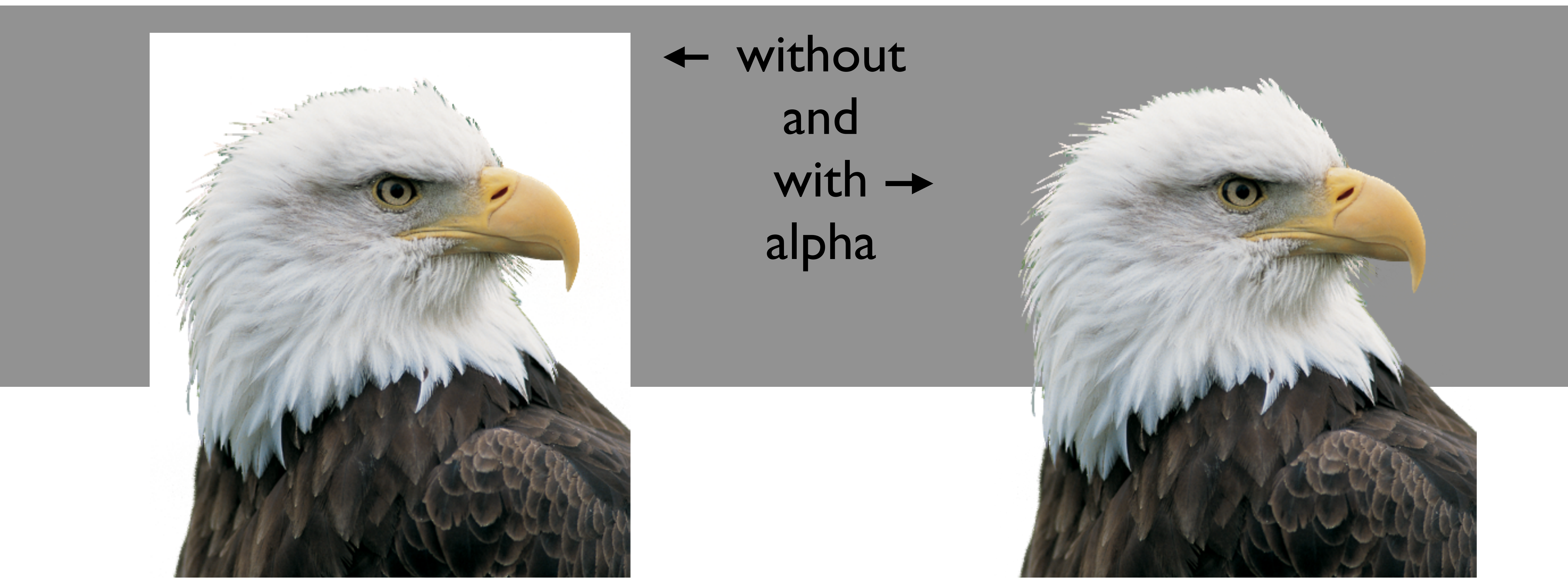
- For color or grayscale, sometimes add *alpha* channel
  - describes transparency of images
  - more on this a bit later

← without
and
with
alpha

[Adobe Photoshop sample]

Datatypes for raster images

- For color or grayscale, sometimes add *alpha* channel
  - describes transparency of images
  - more on this a bit later



[Adobe Photoshop sample]

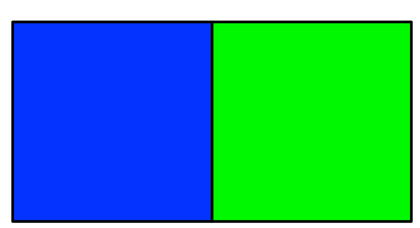
Storage requirements for images

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

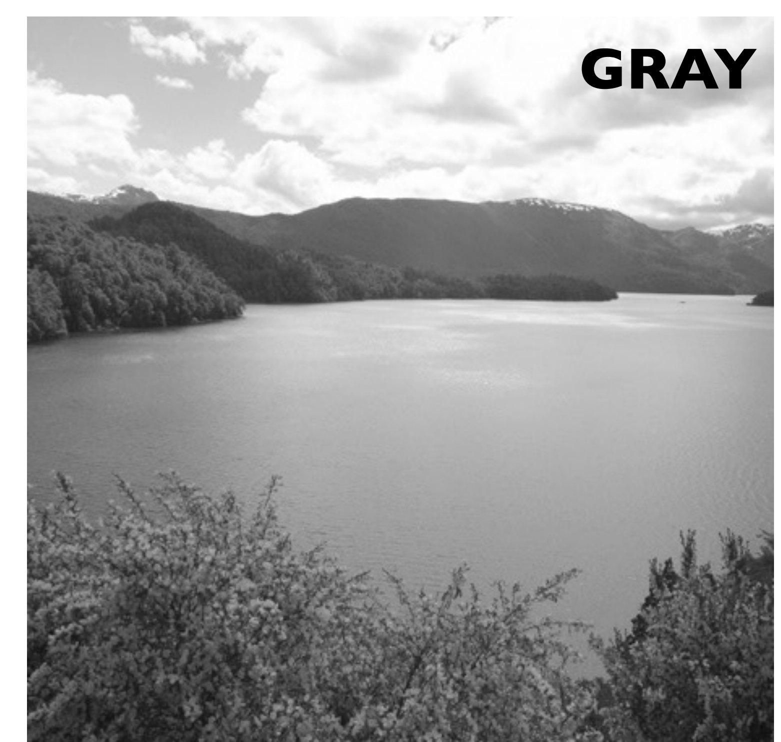
Converting pixel formats

- 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.

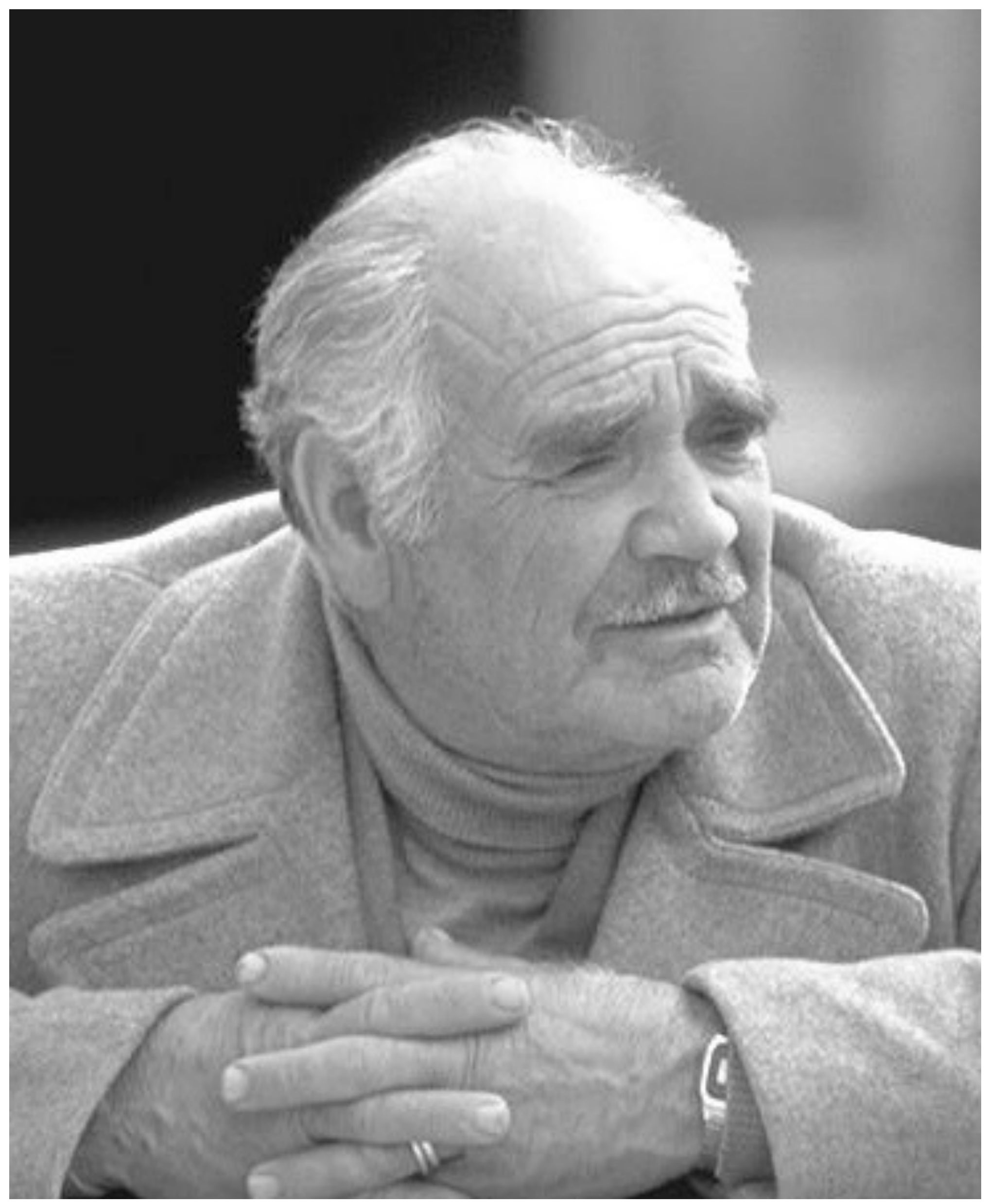


Same luminance?



Converting pixel precision

- Up is easy; down loses information—be careful

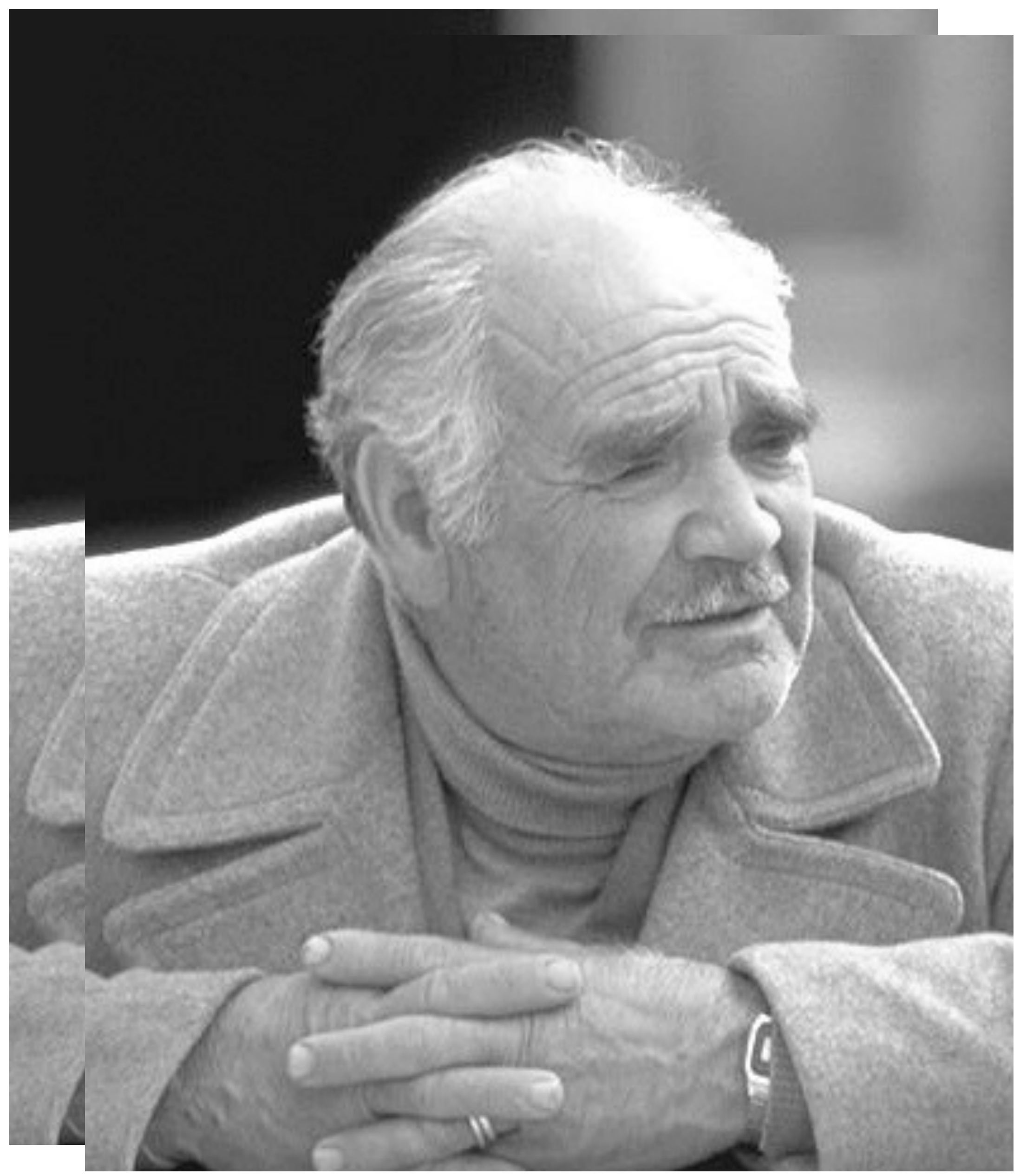


8 bpp (256 grays)

[photo: Philip Greenspun]

Converting pixel precision

- Up is easy; down loses information—be careful

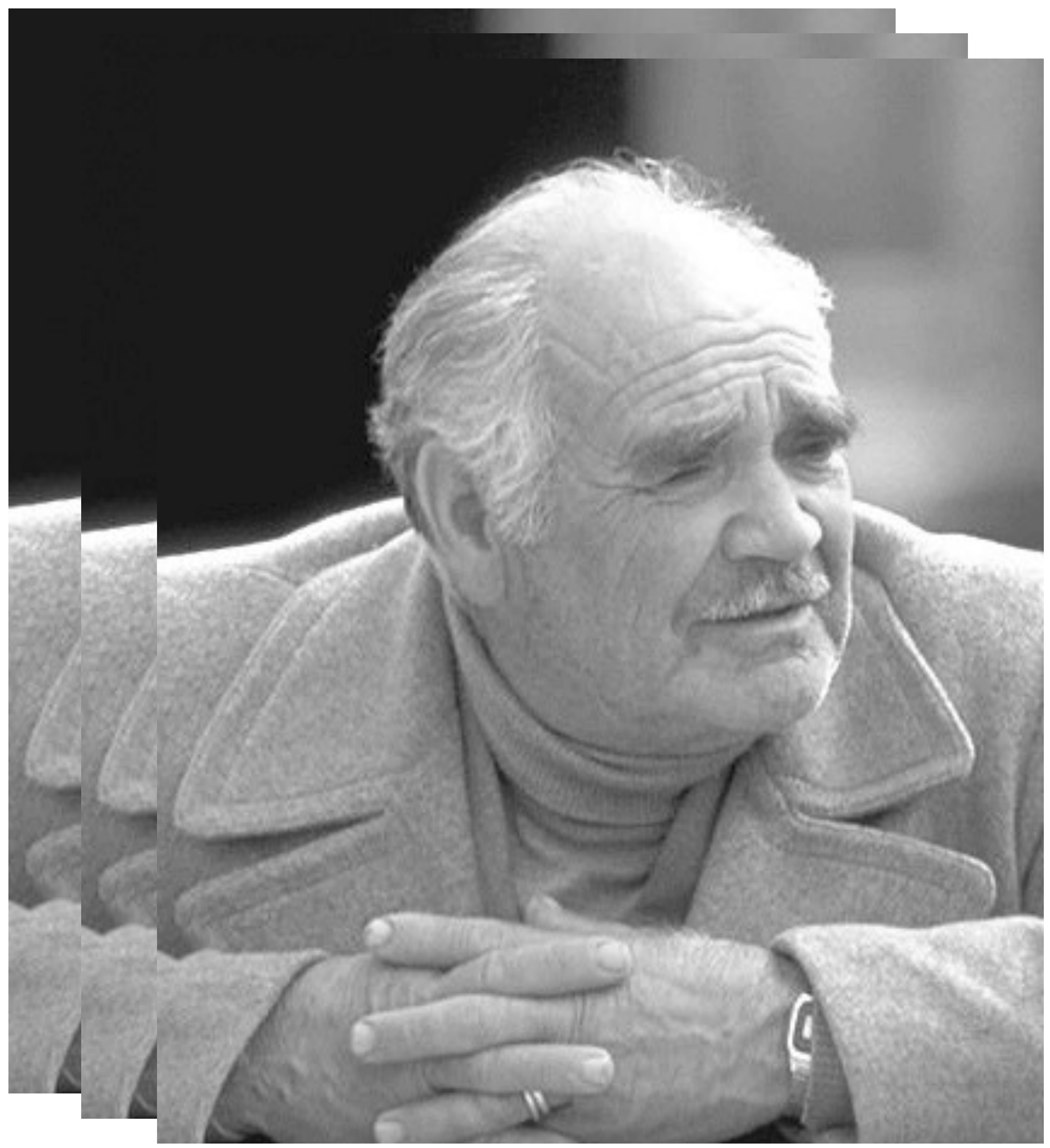


7 bpp (128 grays)

[photo: Philip Greenspun]

Converting pixel precision

- Up is easy; down loses information—be careful

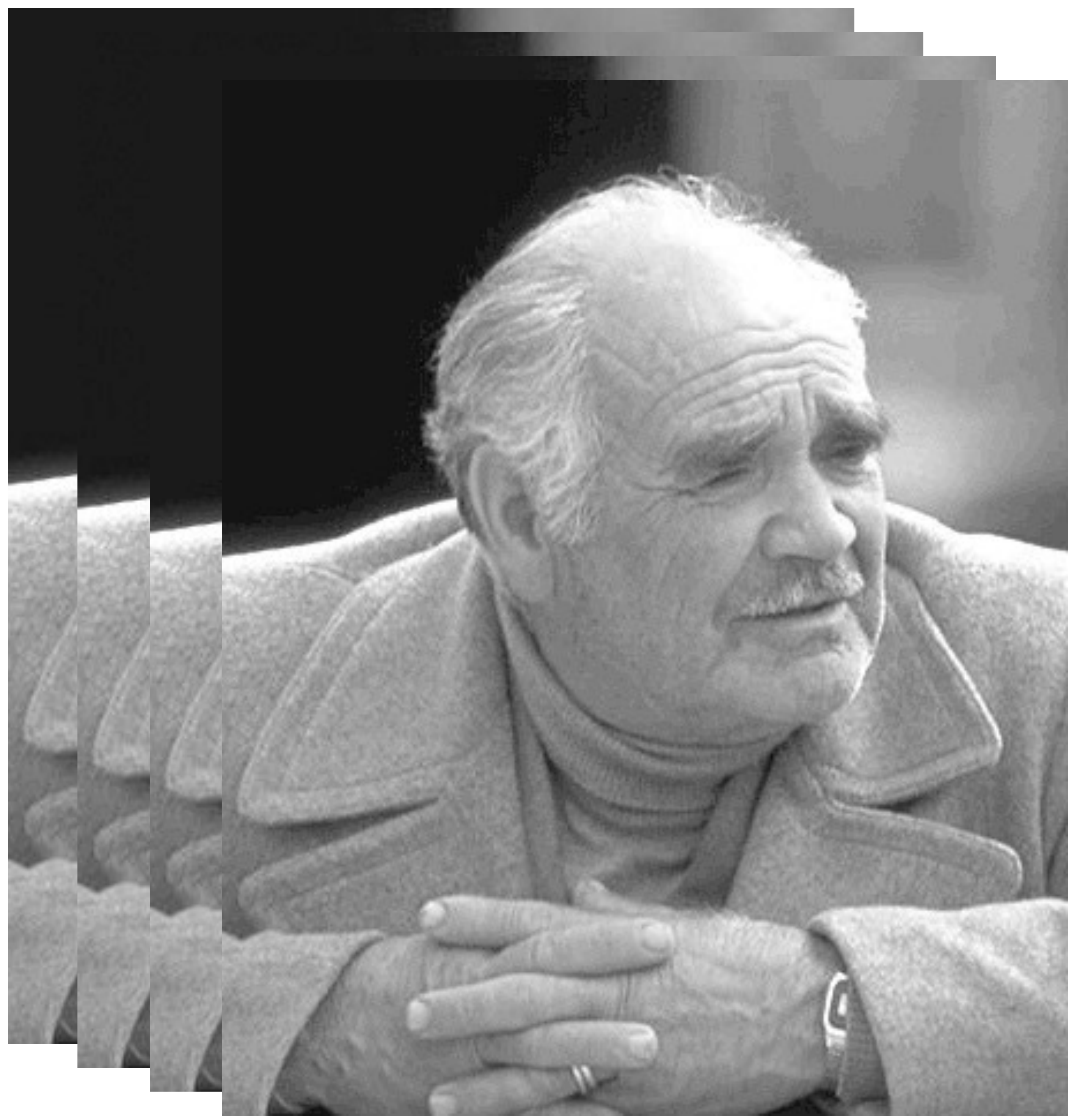


6 bpp (64 grays)

[photo: Philip Greenspun]

Converting pixel precision

- Up is easy; down loses information—be careful

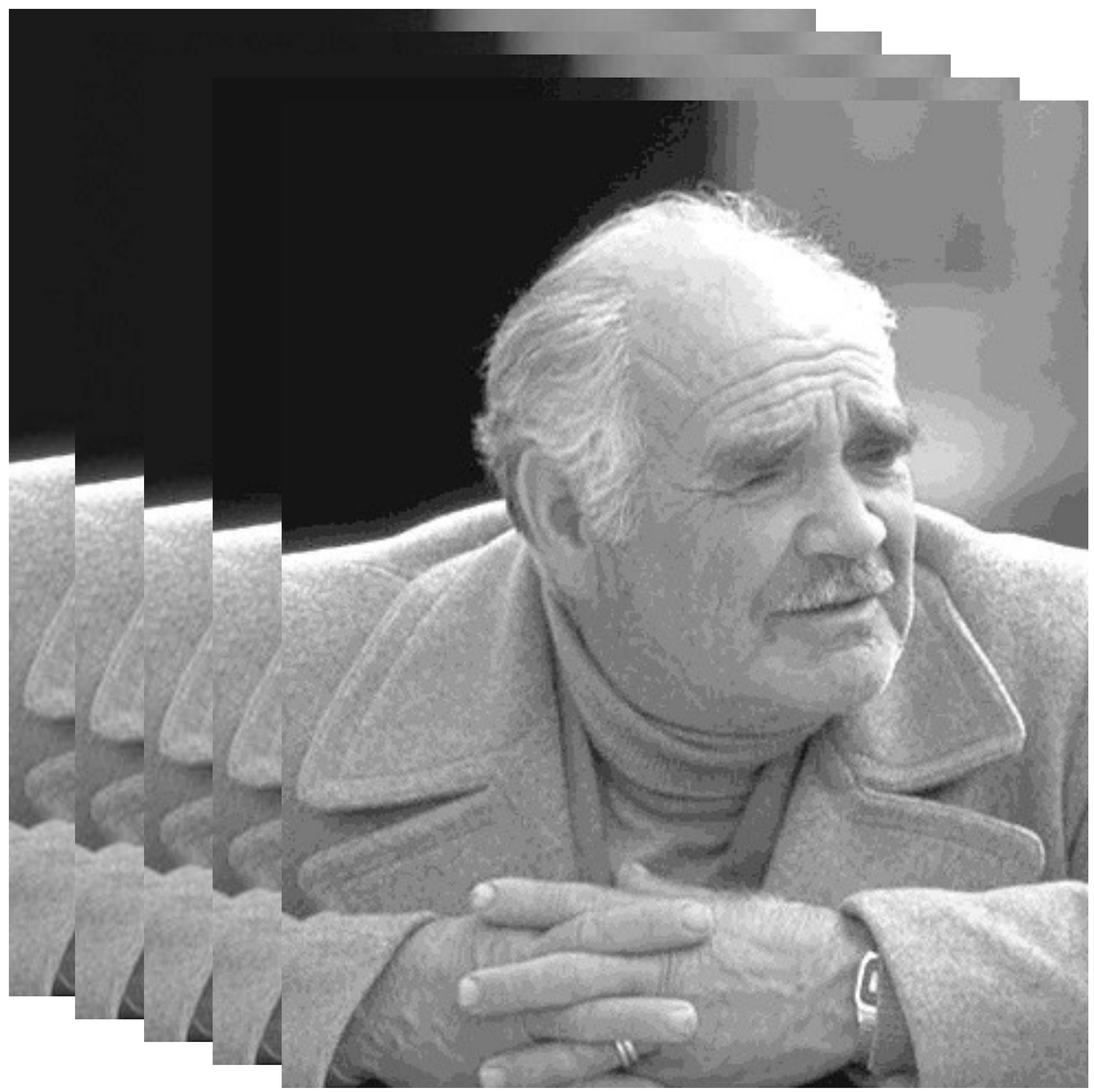


5 bpp (32 grays)

[photo: Philip Greenspun]

Converting pixel precision

- Up is easy; down loses information—be careful

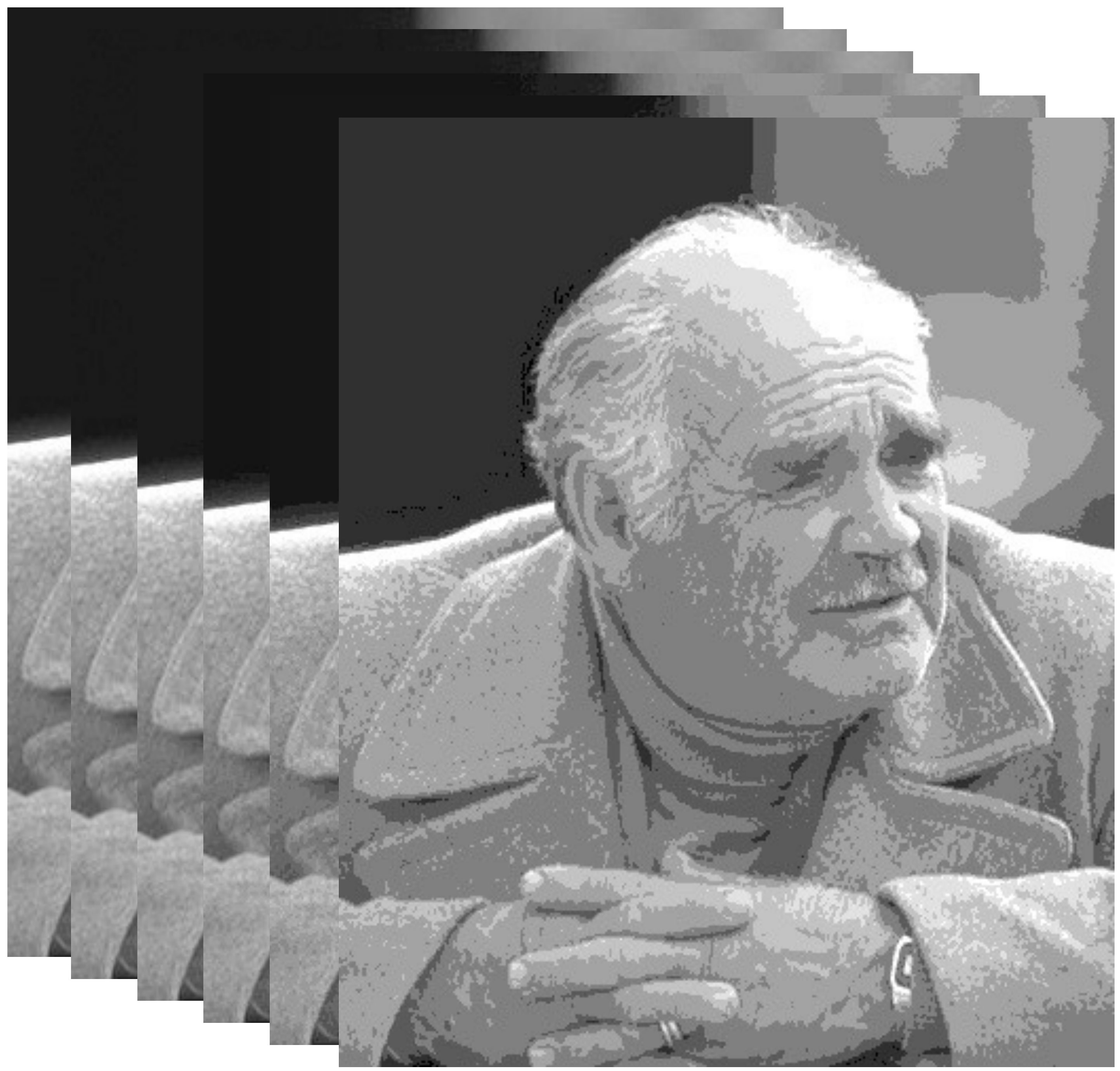


4 bpp (16 grays)

[photo: Philip Greenspun]

Converting pixel precision

- Up is easy; down loses information—be careful

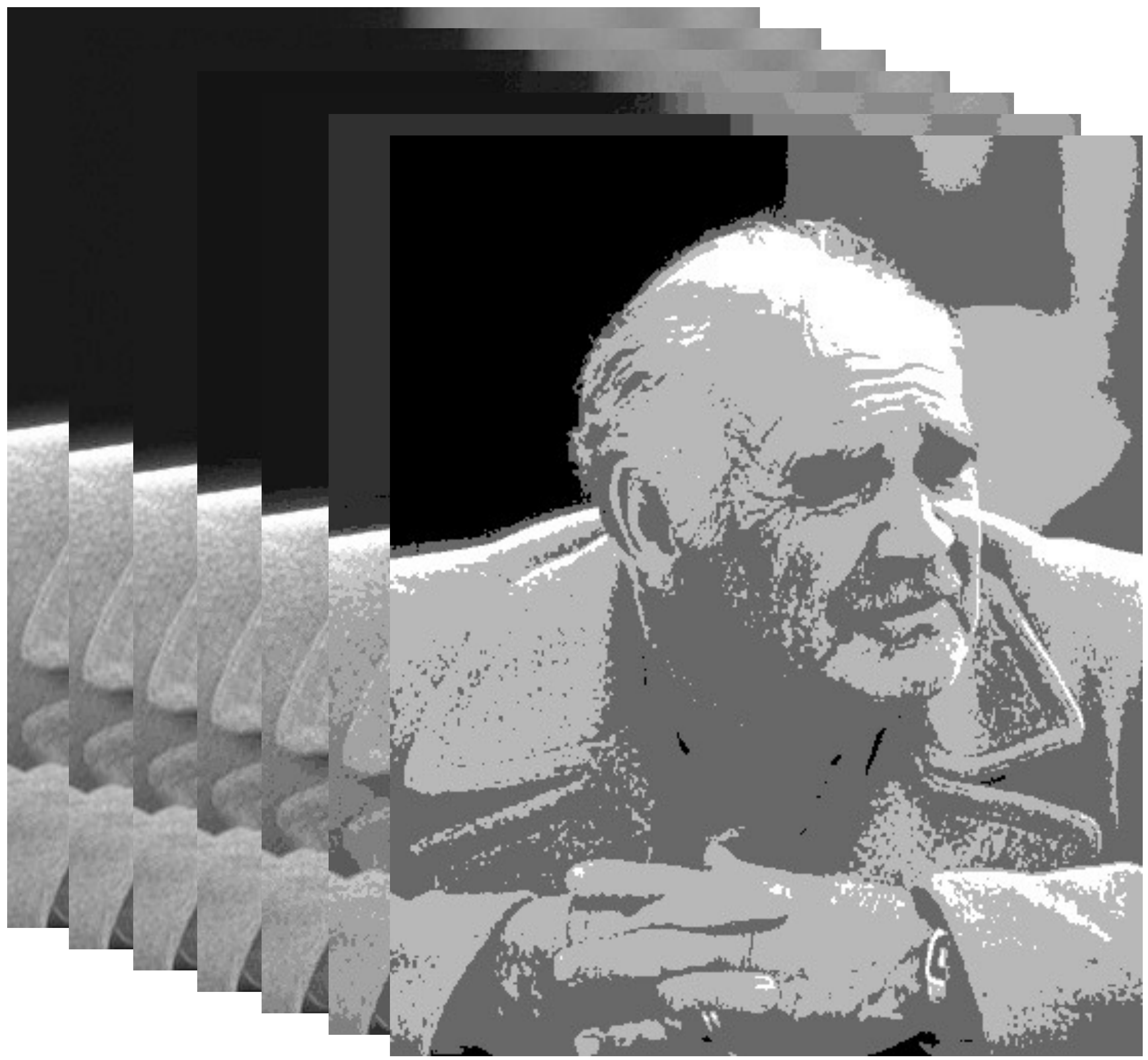


3 bpp (8 grays)

[photo: Philip Greenspun]

Converting pixel precision

- Up is easy; down loses information—be careful

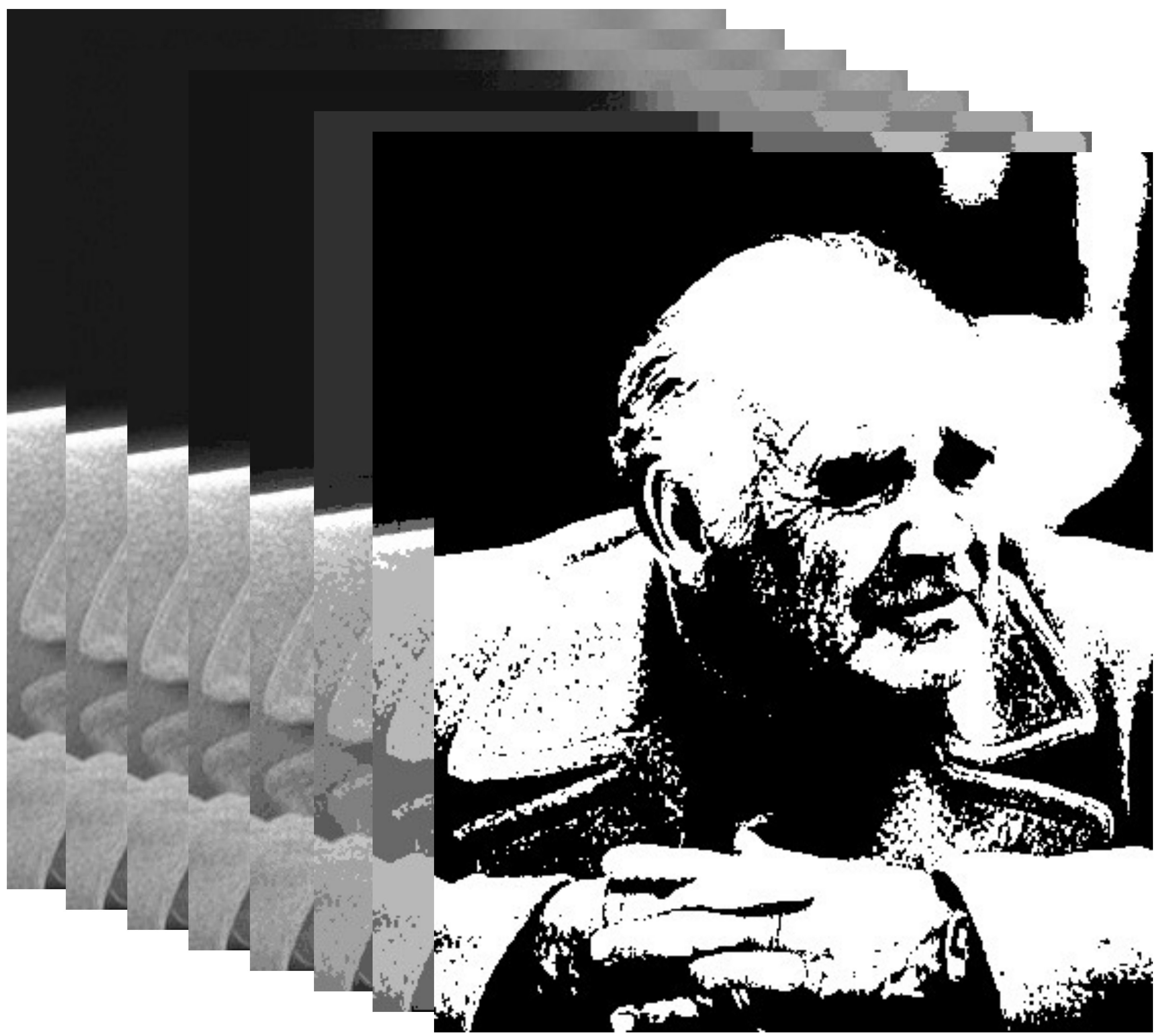


2 bpp (4 grays)

[photo: Philip Greenspun]

Converting pixel precision

- Up is easy; down loses information—be careful



1 bpp (2 grays)

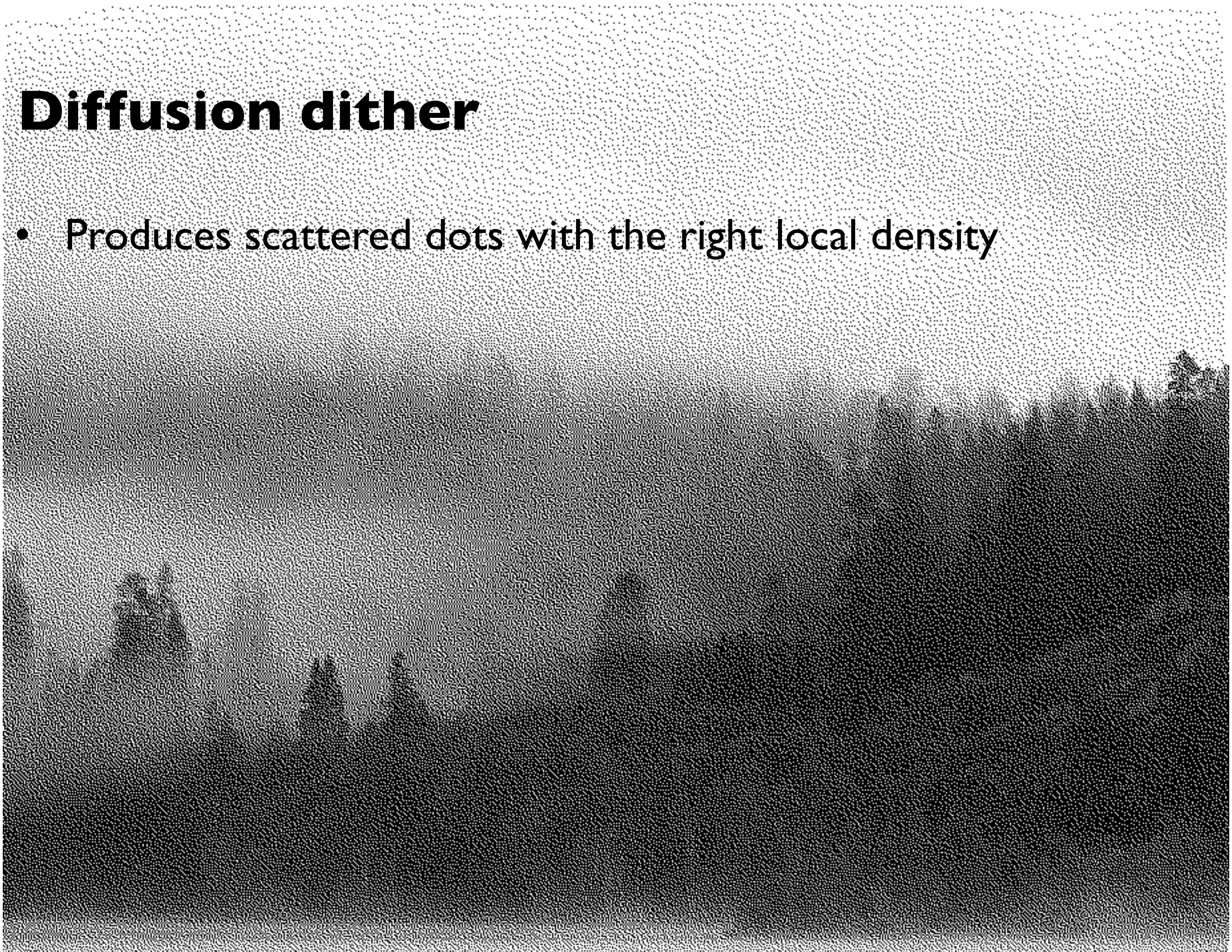
[photo: Philip Greenspun]

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
  - usually want random-looking, non-distracting patterns

Diffusion dither

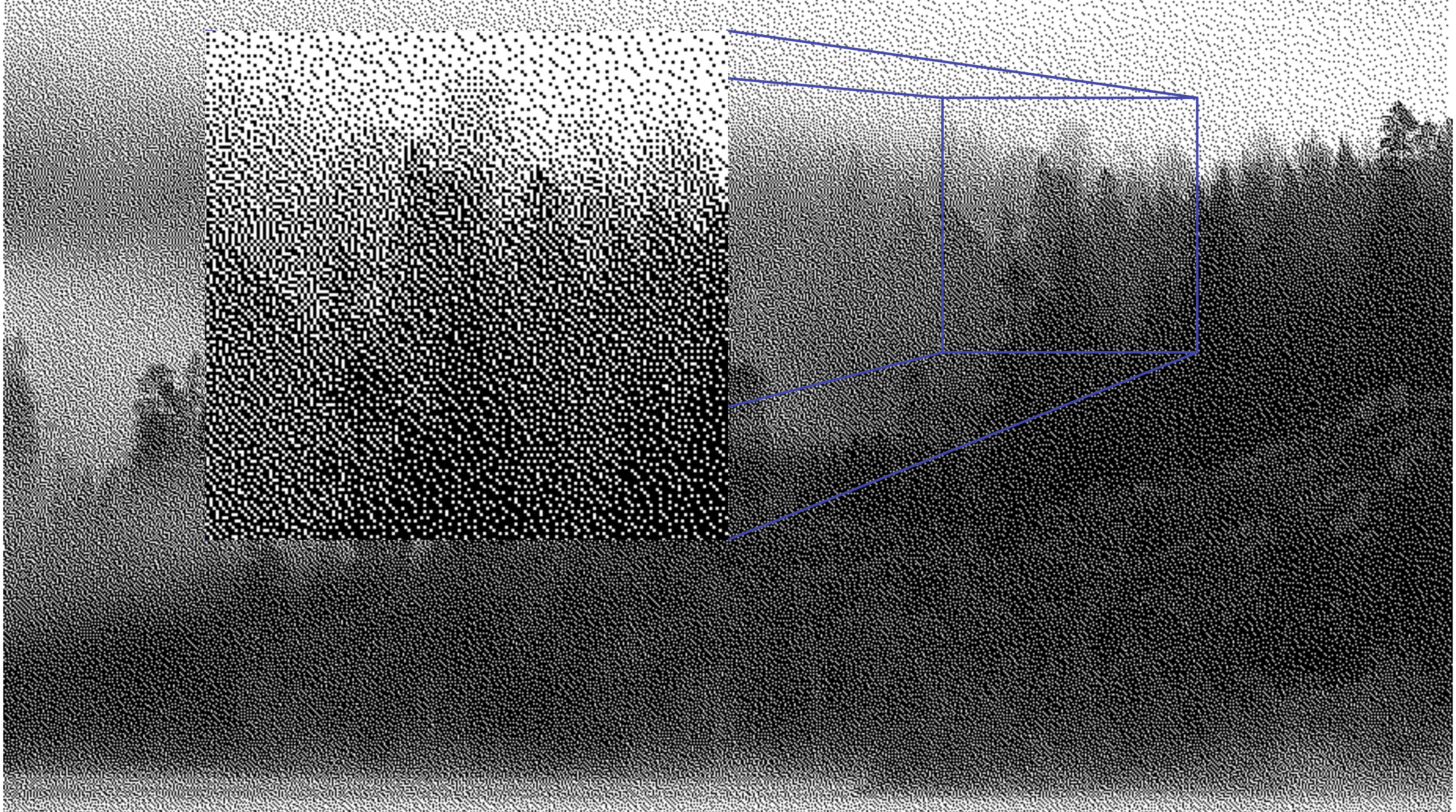
- Produces scattered dots with the right local density



[photo: Philip Greenspun]

Diffusion dither

- Produces scattered dots with the right local density



[photo: Philip Greenspun]

Images and Displays

3. What pixel values mean

CS4620

Fall 2020

Intensity encoding in images

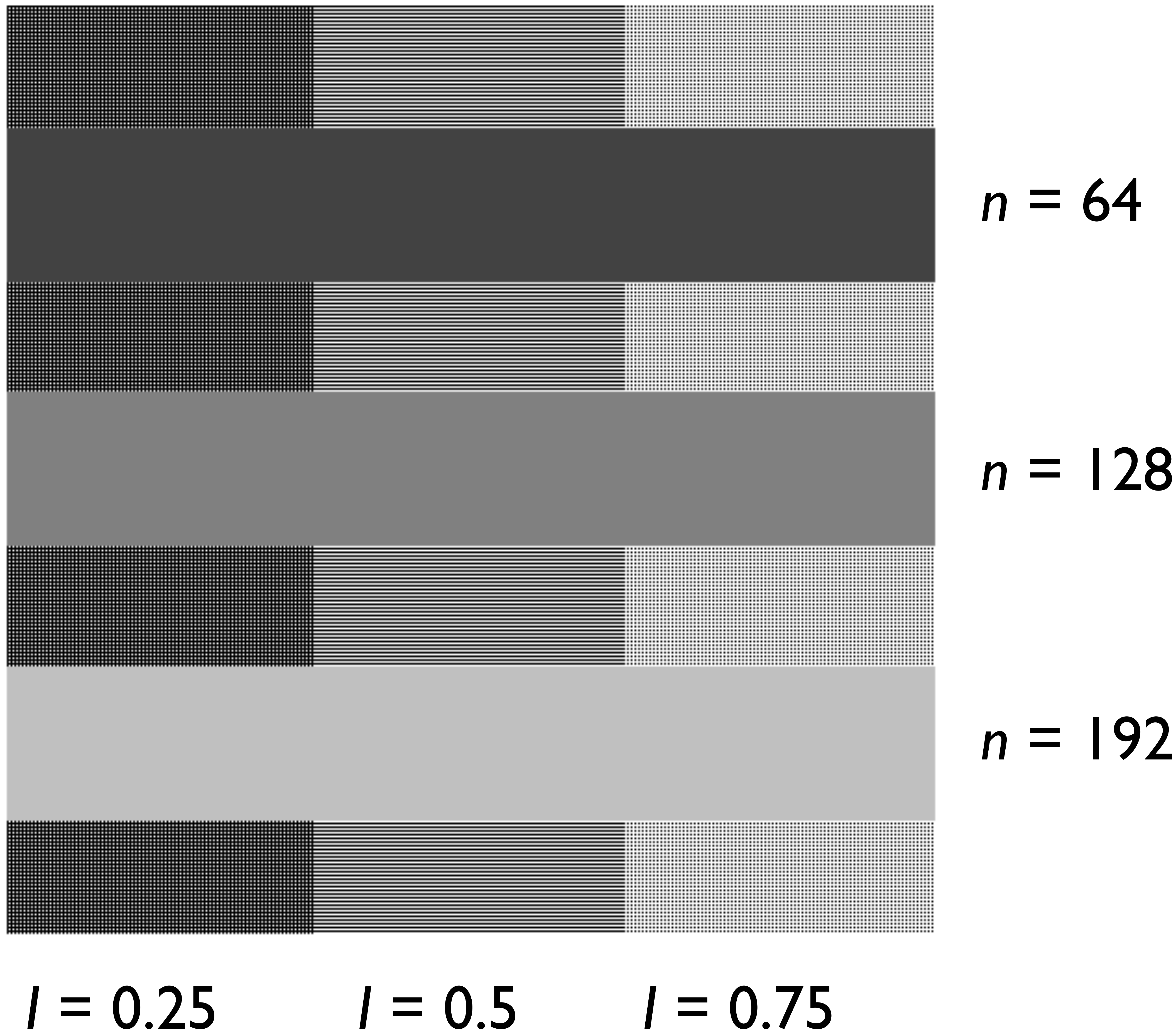
- What do the numbers in images (pixel values) mean?
  - they determine how bright that pixel is
  - bigger numbers are (usually) brighter
  - for floating point pixels, they directly give the intensity (in some units) — they are *linearly* related to the intensity
  - for pixels encoded in integers,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

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

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

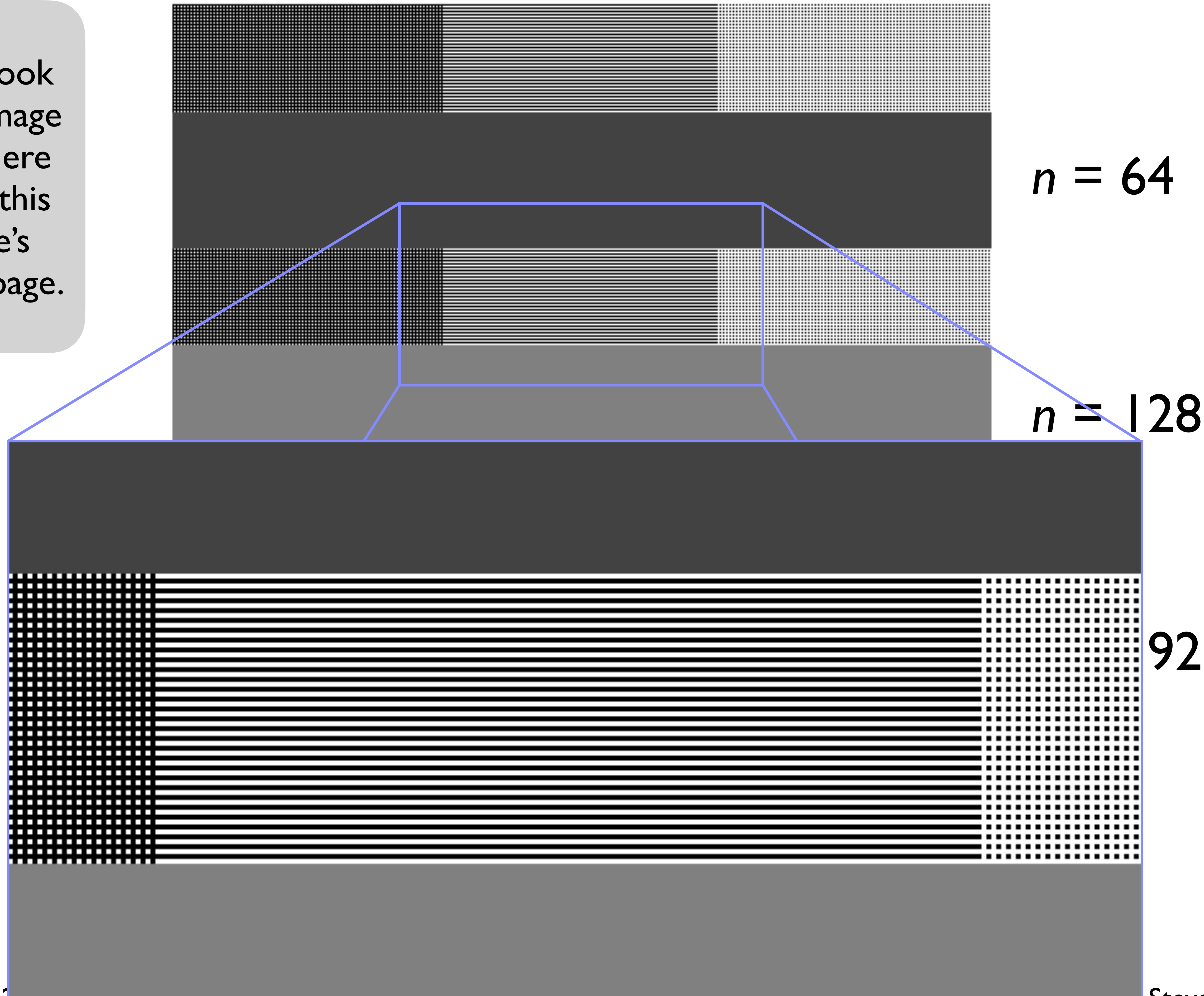
What this screen does

Take a look at this image linked here and on this lecture's Canvas page.



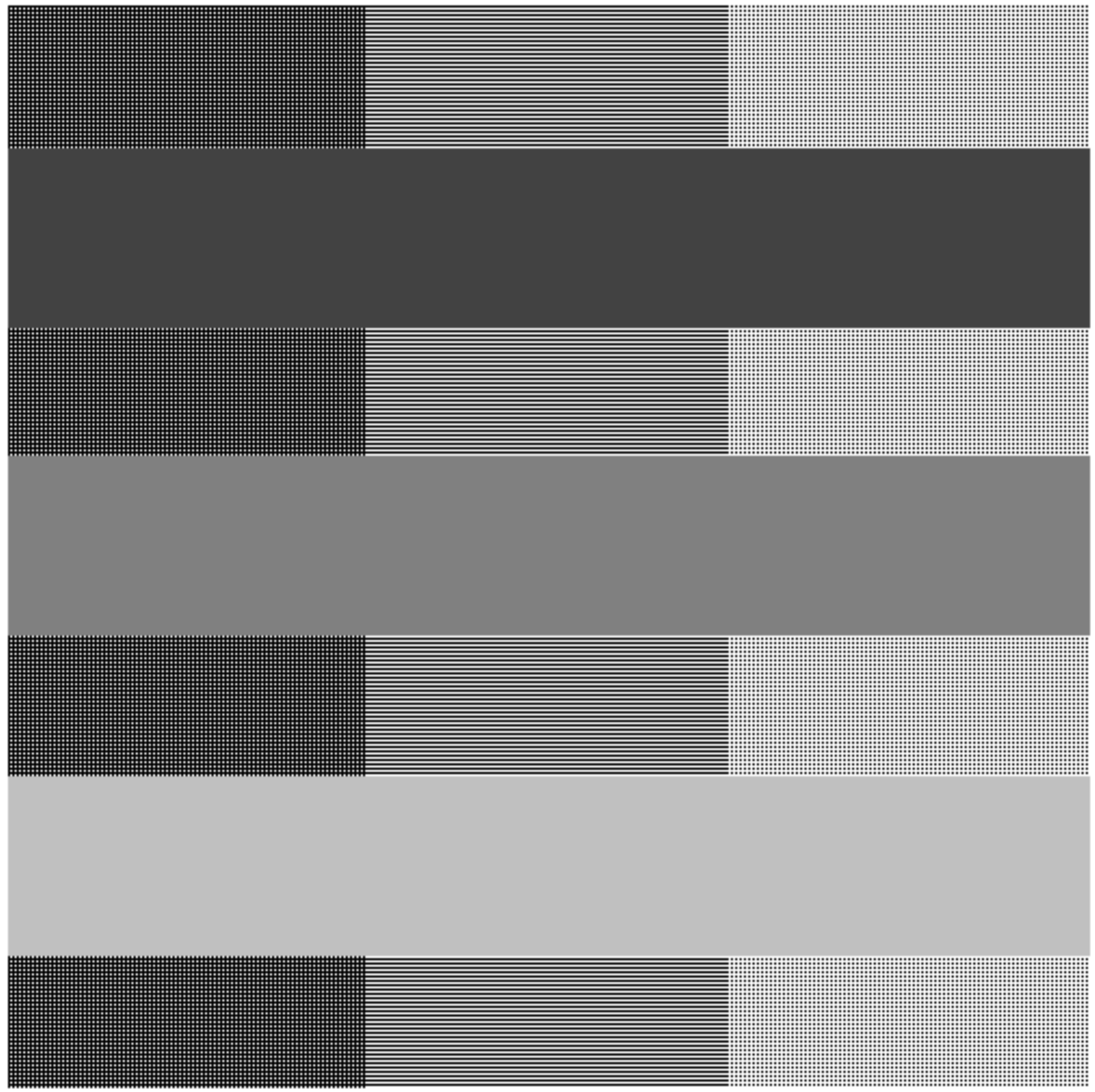
What this screen does

Take a look at this image linked here and on this lecture's Canvas page.



What this screen does

Take a look at this image linked here and on this lecture's Canvas page.



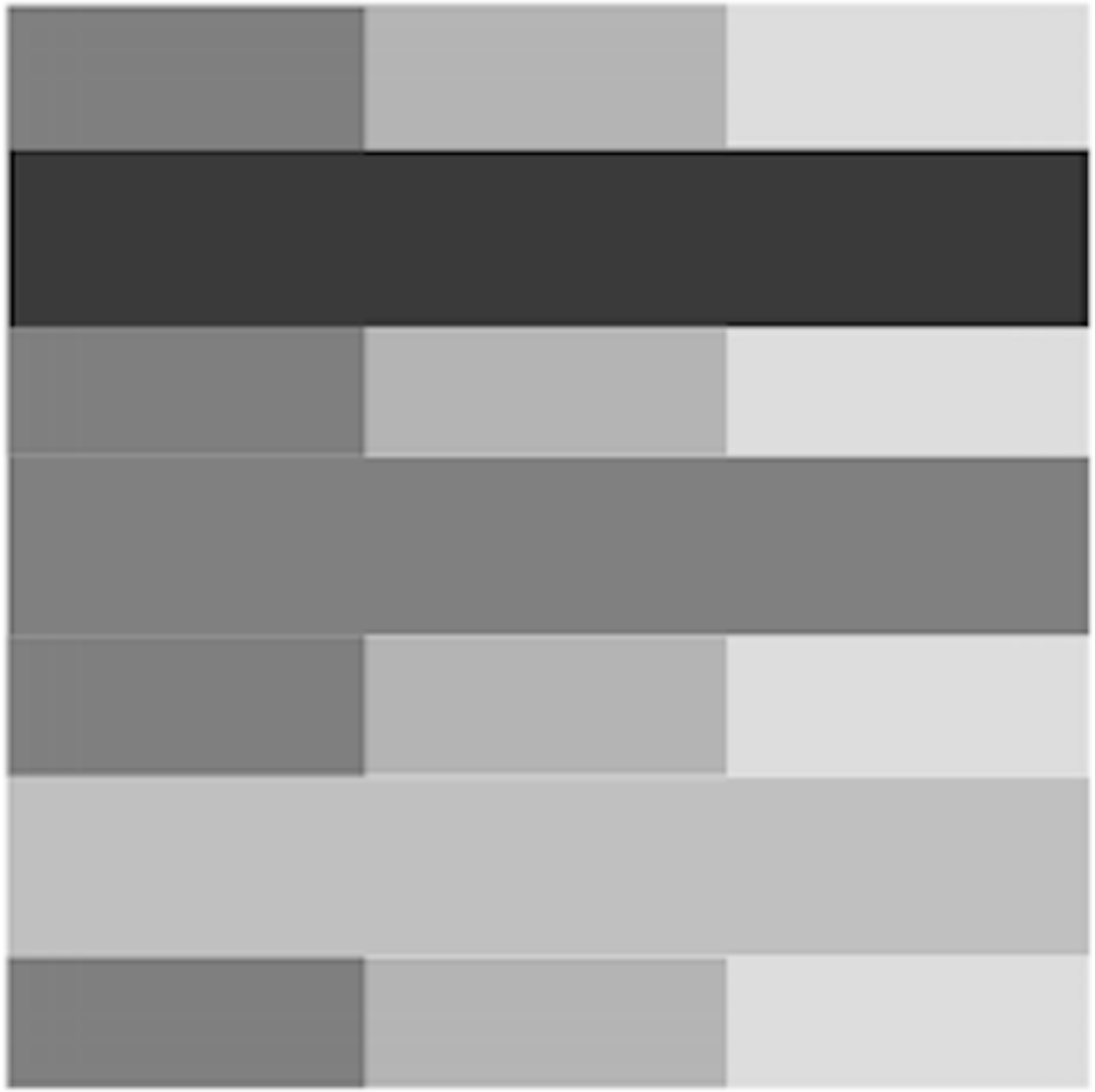
$n = 64$

$n = 128$

$n = 192$

$I = 0.25$ $I = 0.5$ $I = 0.75$

What this screen might do (simulated)



$n = 64$

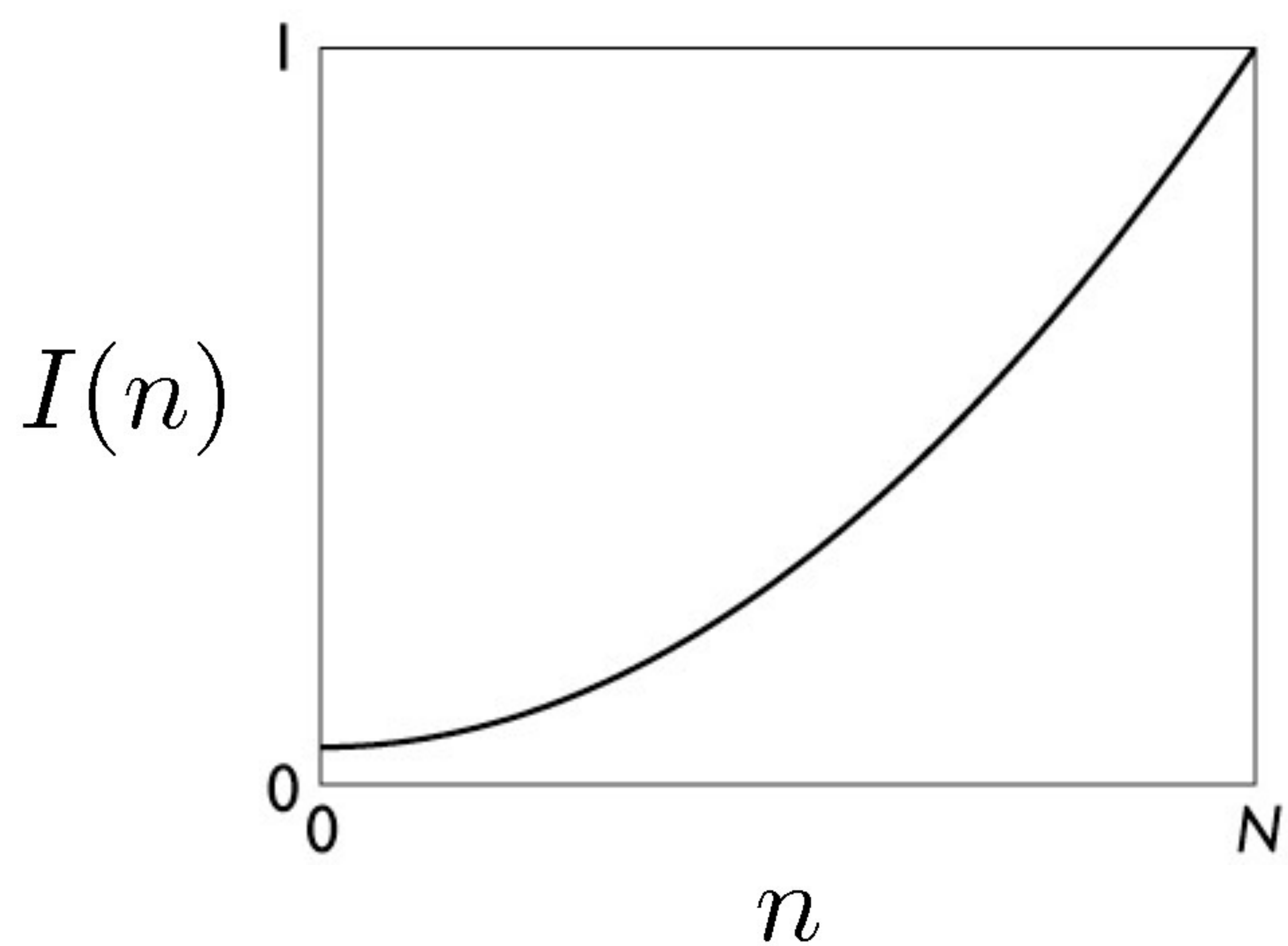
$n = 128$

$n = 192$

$I = 0.25$ $I = 0.5$ $I = 0.75$

What this screen does

- Something like this:



Constraints on transfer function

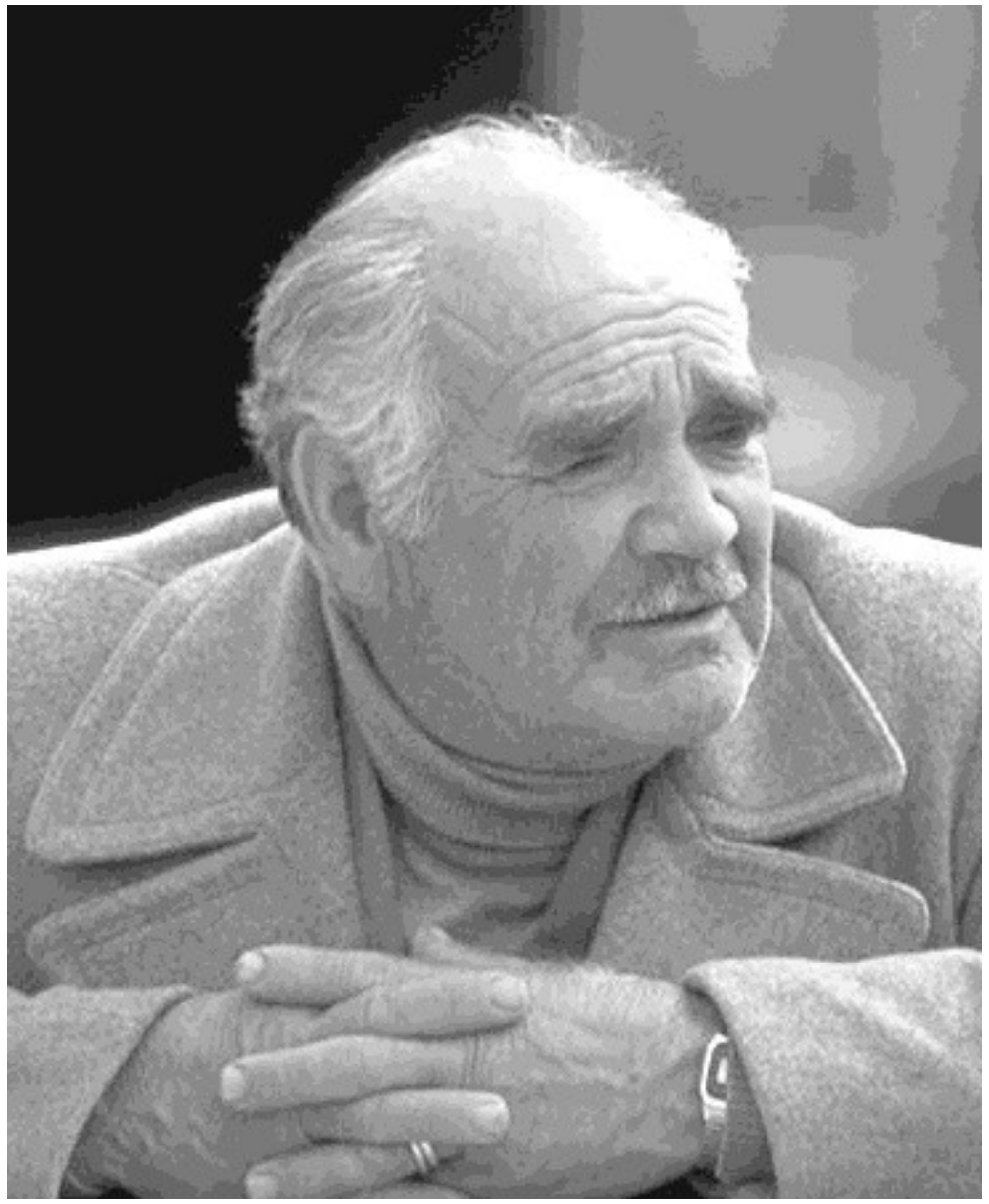
- Maximum displayable intensity, I_{max}
  - how much power can be channeled into a pixel?
    - LCD: backlight intensity, transmission efficiency (<10%)
    - projector: lamp power, efficiency of imager and optics
- Minimum displayable intensity, I_{min}
  - light emitted by the display in its "off" 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



[Philip Greenspun]

an image with severe *banding*

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 show at the cinema
    - 480 to drive an 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 $I_{\min} \cdot R_d$ 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

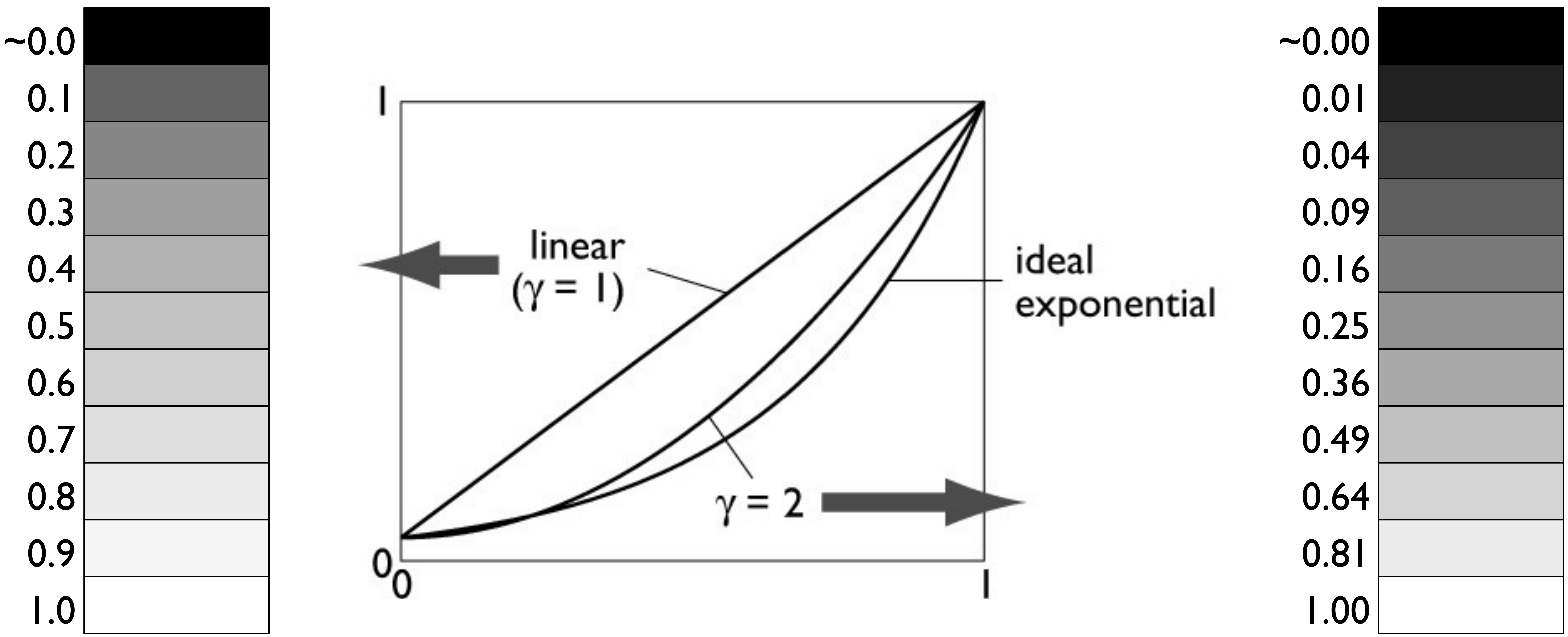
Intensity quantization in practice

- 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 3: 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 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,

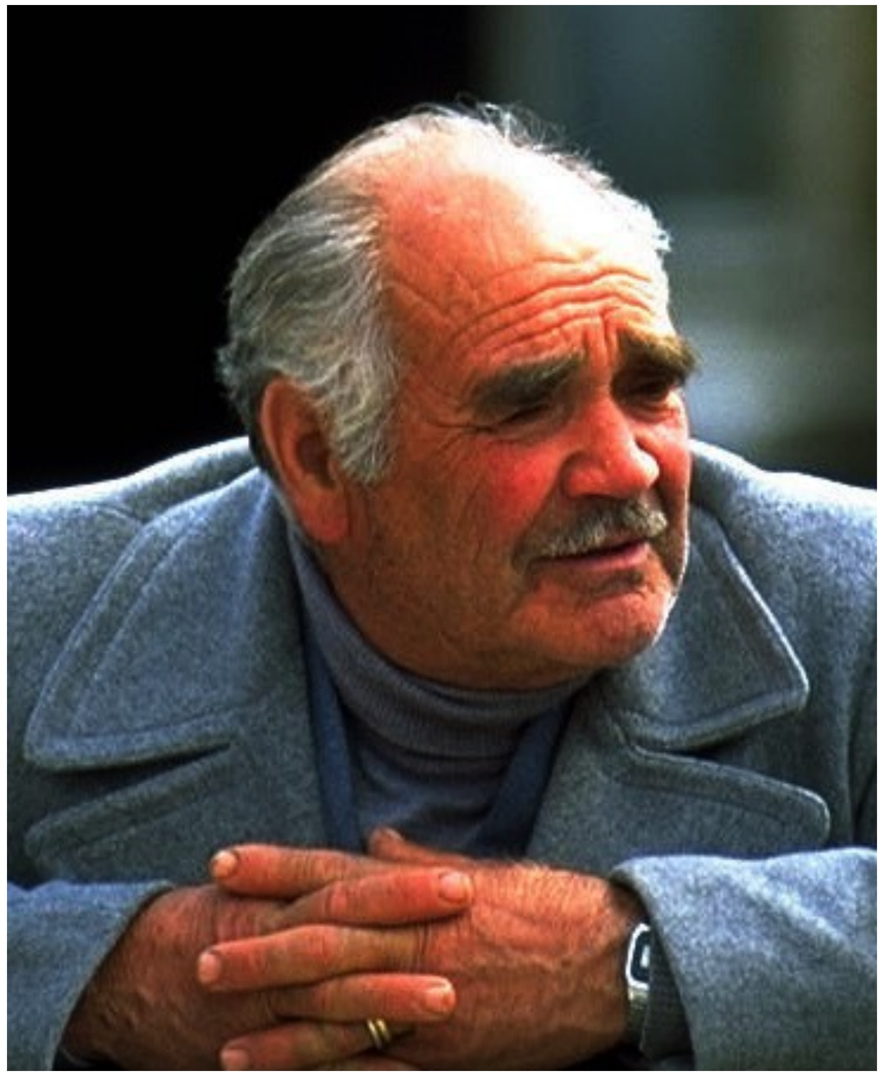
$$I(n) = (n/N)^\gamma$$

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

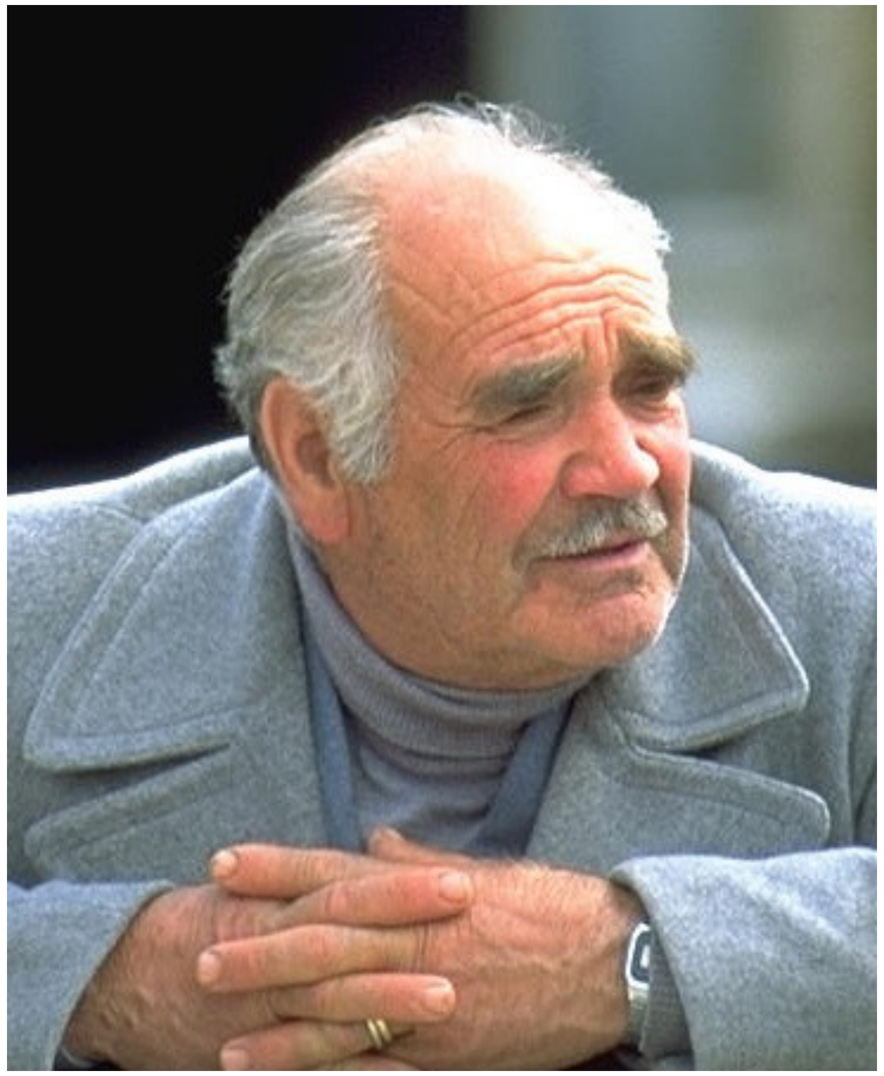
$$n(I) = N I^{\frac{1}{\gamma}}$$

- 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

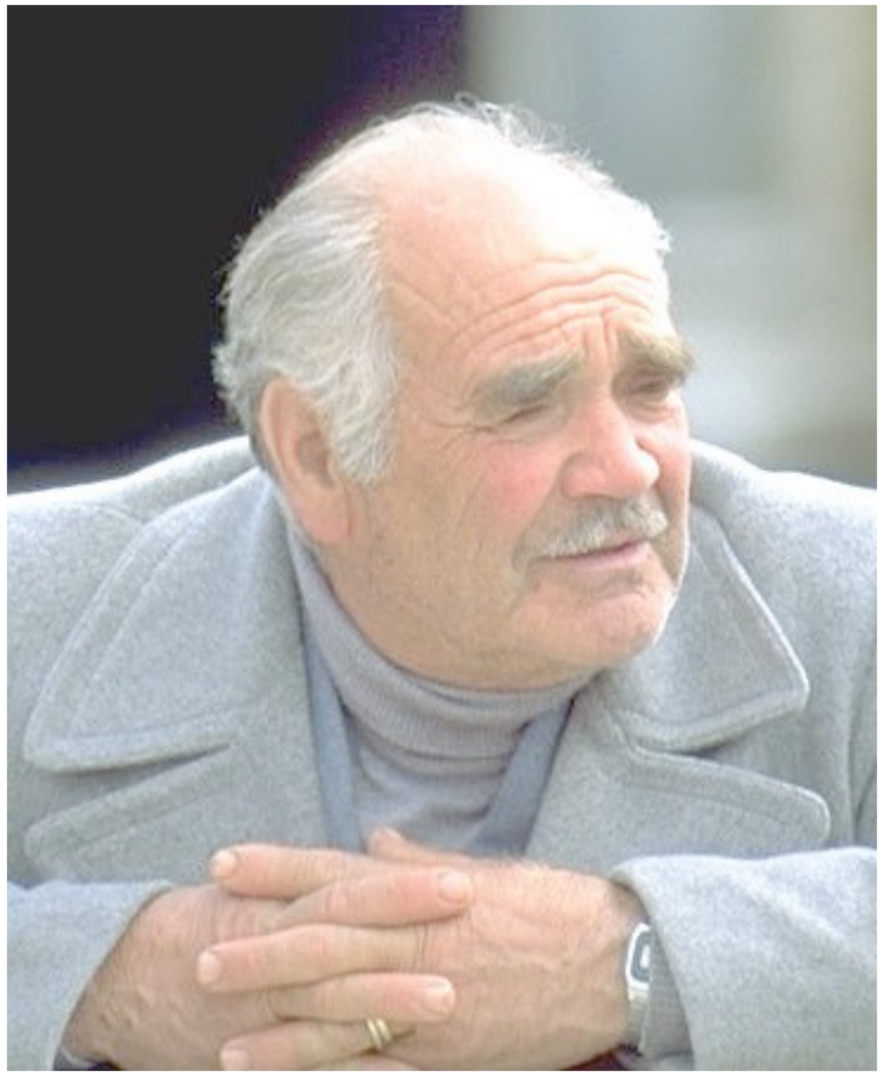
Gamma correction



corrected for
γ lower than
display



OK



corrected for
γ higher than
display

[Philip Greenspun]

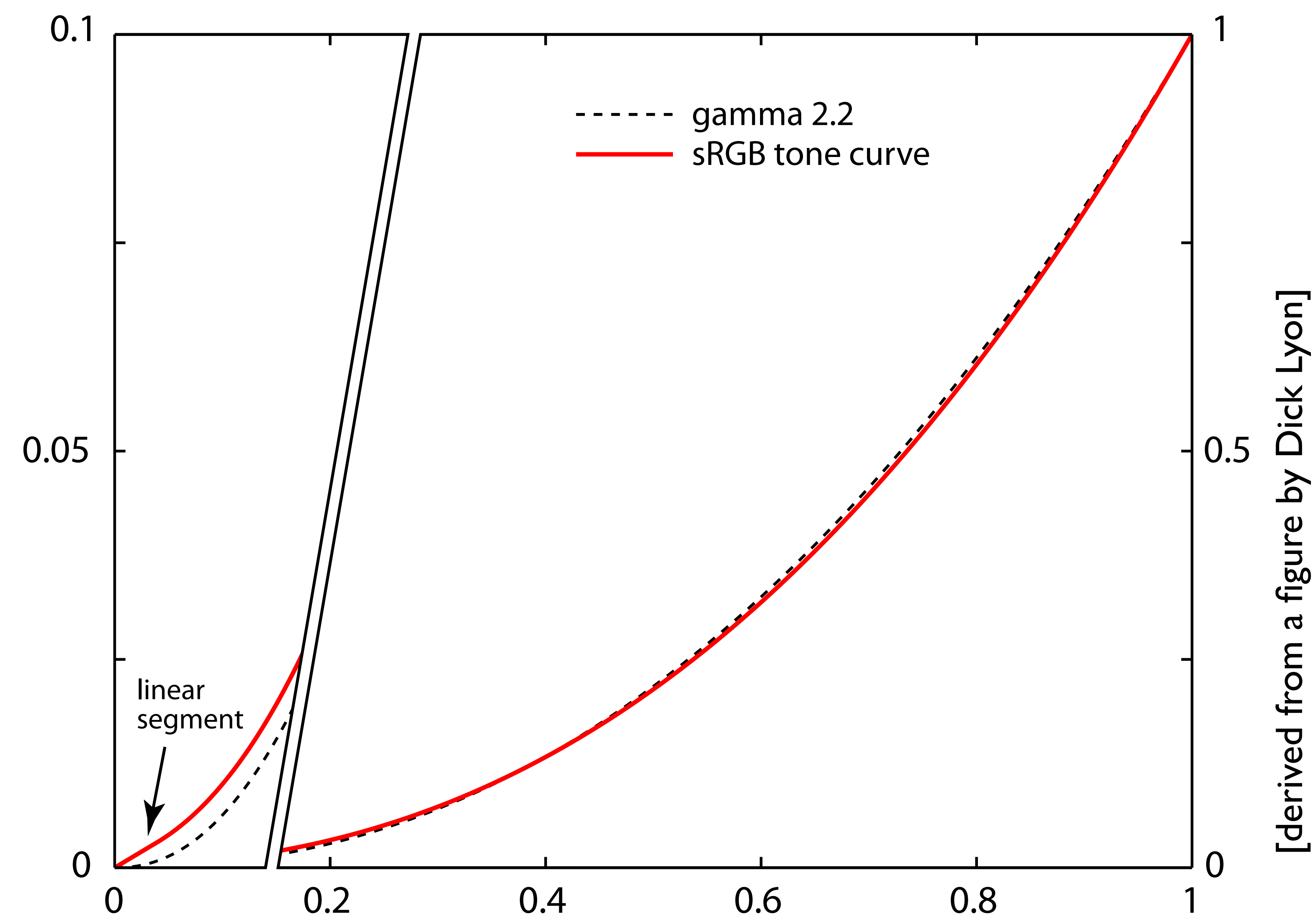
sRGB quantization curve

- The predominant standard for "casual color" in computer displays
  - consistent with older typical practice
  - designed to work well under imperfect conditions
  - monitors are calibrated to sRGB by default
  - sRGB is the standard interpretation of images on the web
  - in practice, sRGB usually defines what your pixel values mean

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

$$C = n/N$$

$$a = 0.055$$



High dynamic range images

demo

- We've been brought up to think of pixel values in a bounded range like [0, 255].
- But this is only because output devices work this way.
- Measured images (e.g. from cameras) or rendered images can easily contain a wide range of intensities

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

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
- Modern methods are content-dependent and spatially varying

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

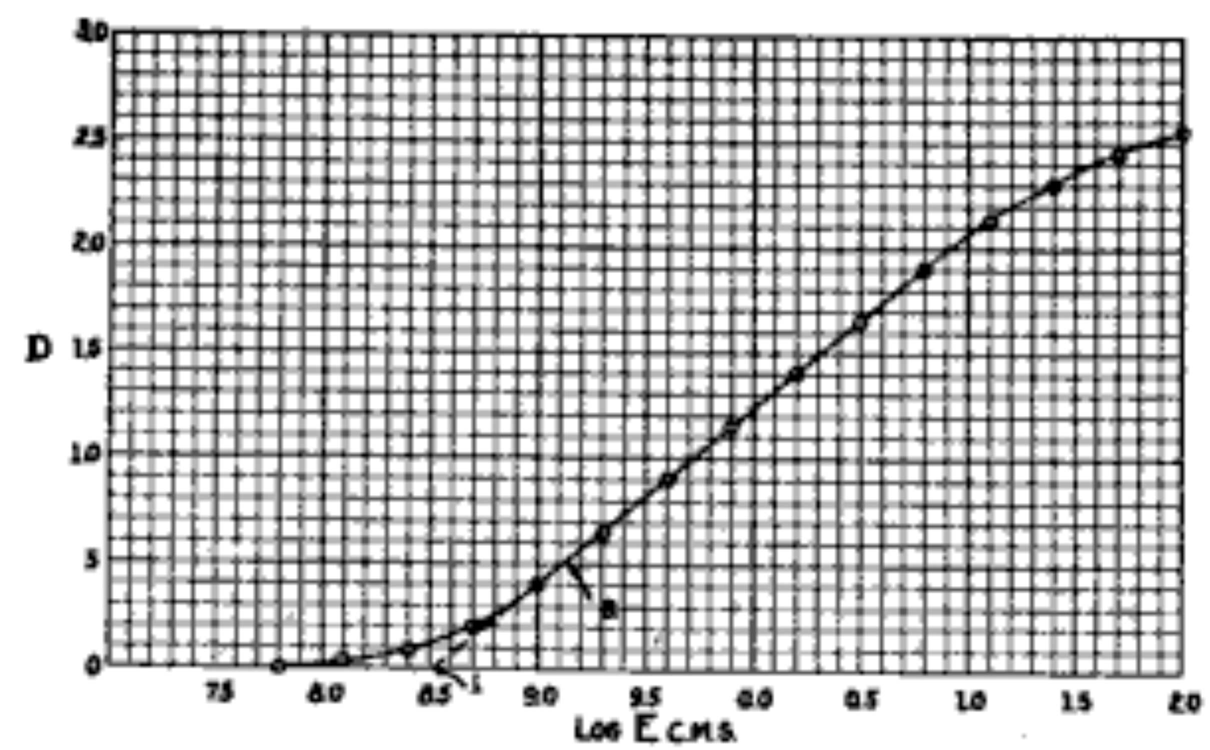


FIG. 6—*Density-exposure curve showing toe, straightline, and shoulder*

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)

Images and Displays

4. Image I/O

CS4620

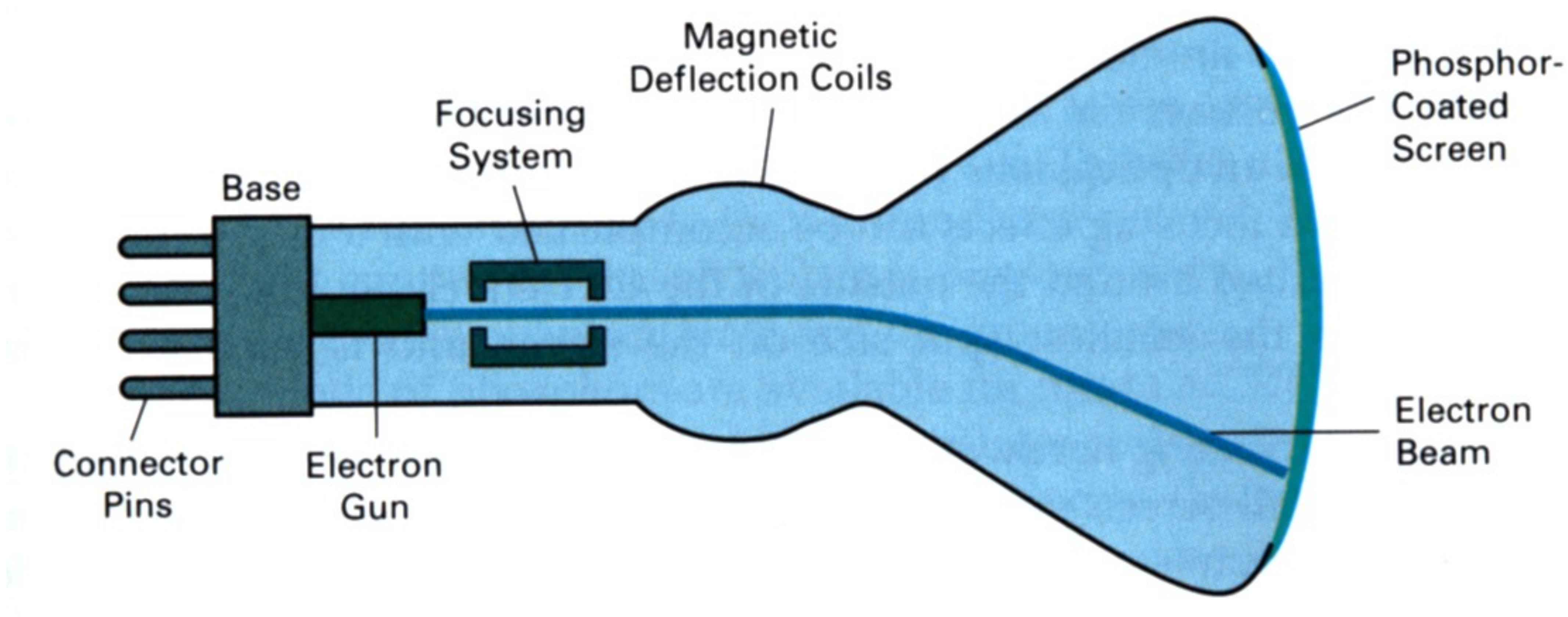
Fall 2020

Representative display technologies

- Direct-view displays
  - Raster CRT display
  - LCD display
  - LED display
- Projection displays
  - LCD projectors
  - DLP/DMD projectors
- Printers
  - Laser printer
  - Inkjet printer

Cathode ray tube

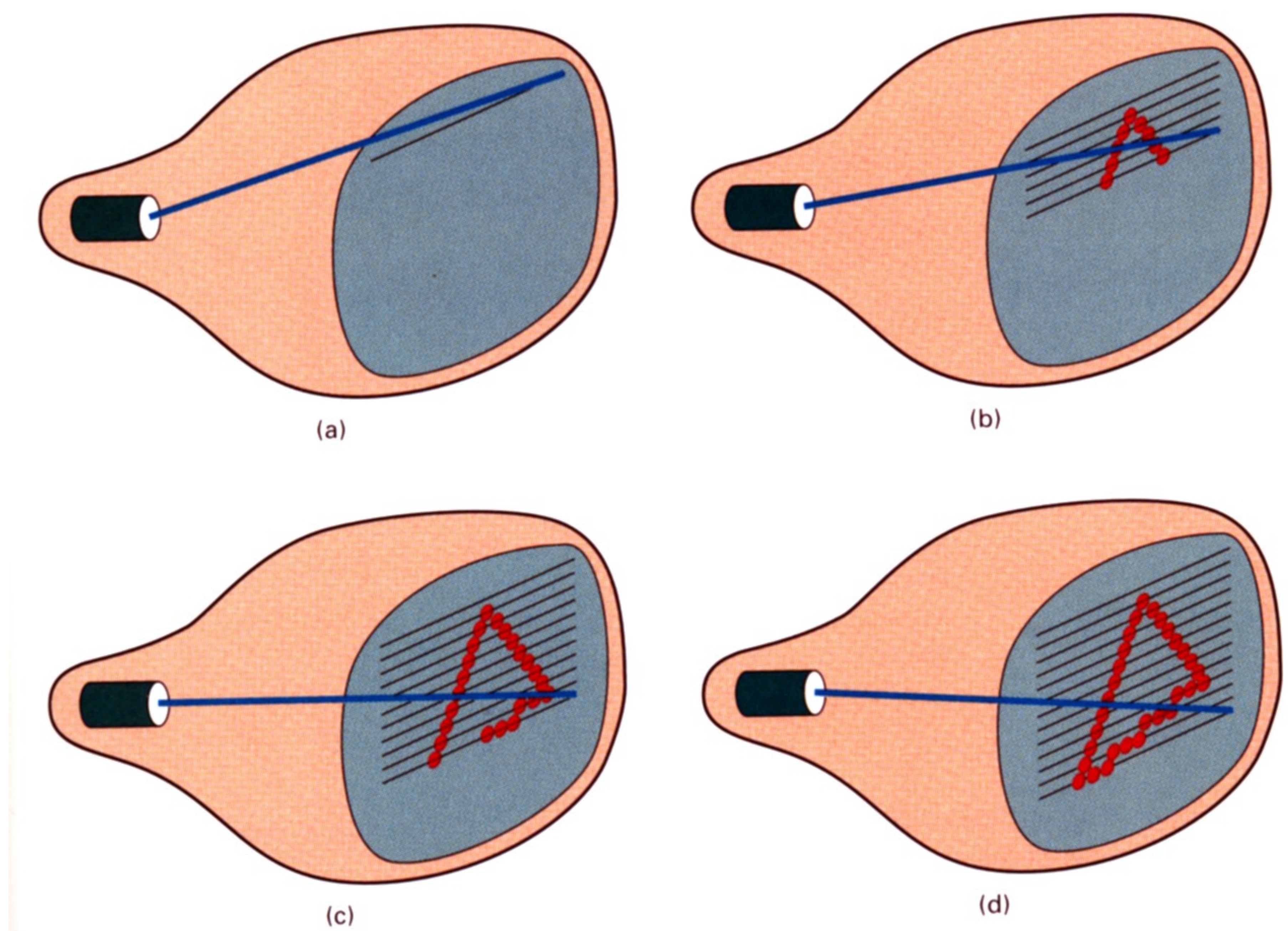
- First widely used electronic display
  - developed for TV in the 1920s–1930s



[H&B fig. 2-2]

Raster CRT display

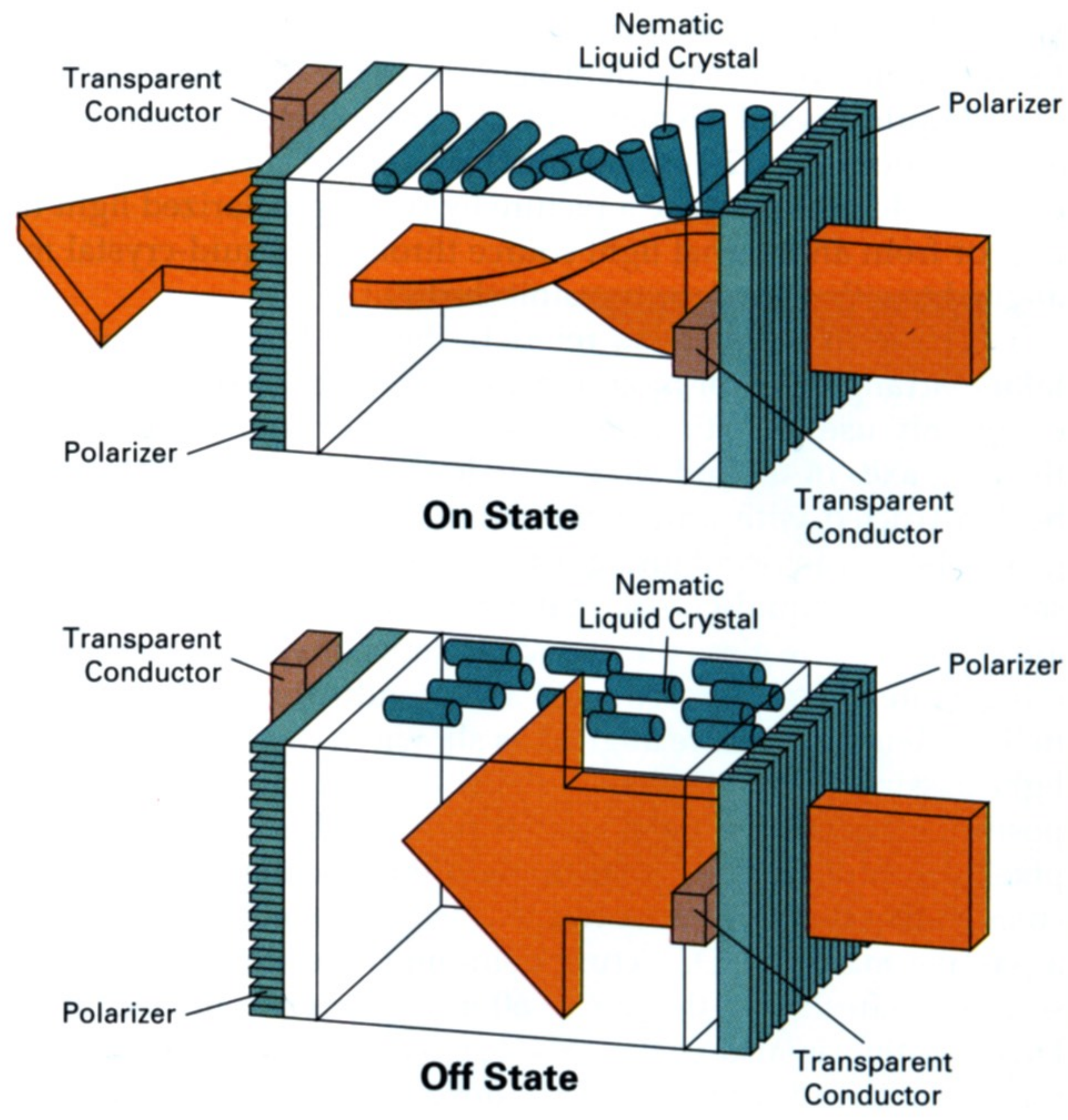
- Scan pattern fixed in display hardware
- Intensity modulated to produce image
- Originally for TV
  - (continuous analog signal)
- For computer, intensity determined by contents of framebuffer



[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]

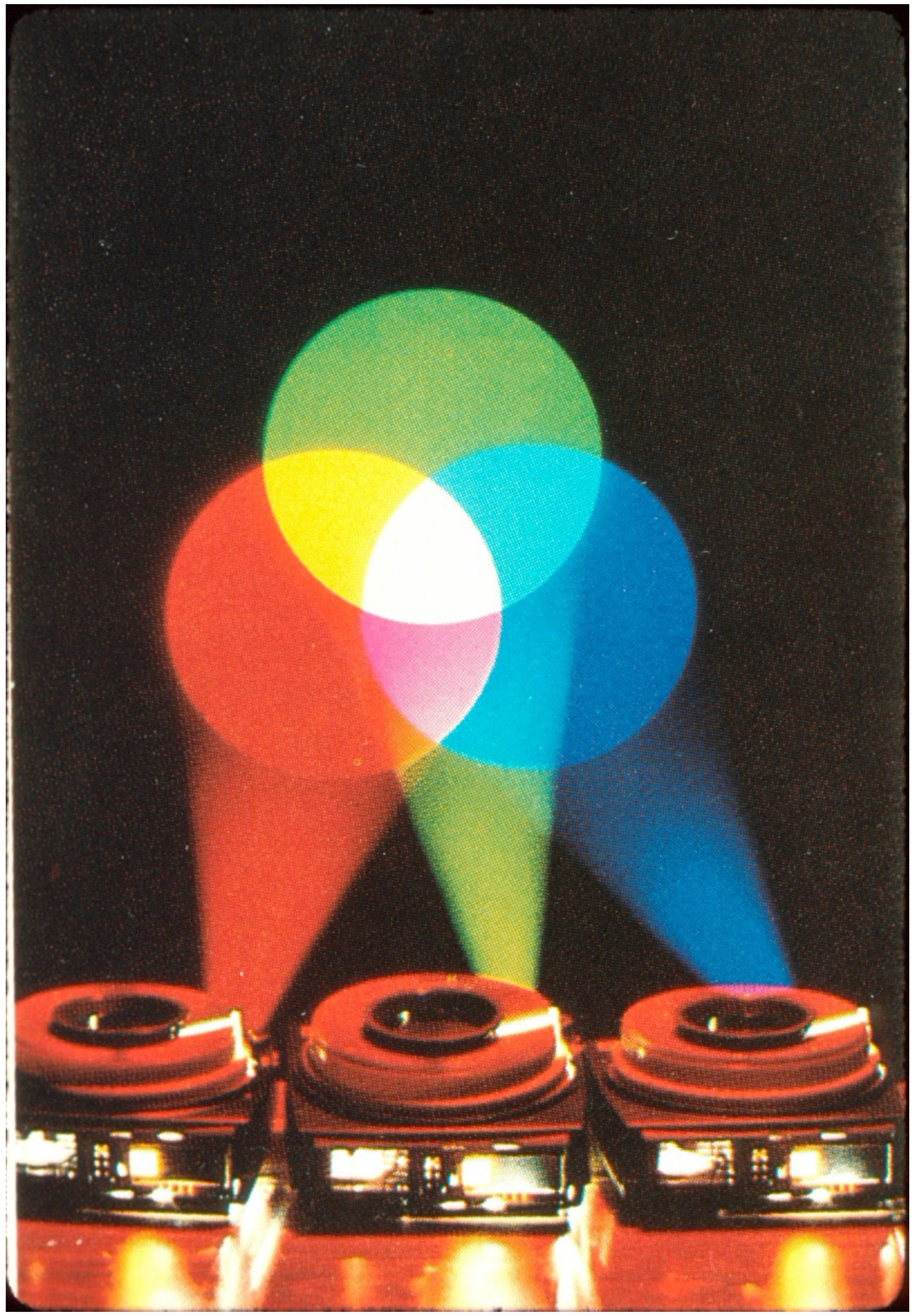
LED Displays

[Wikimedia Commons]

[Google—Nexus 4]

Color displays

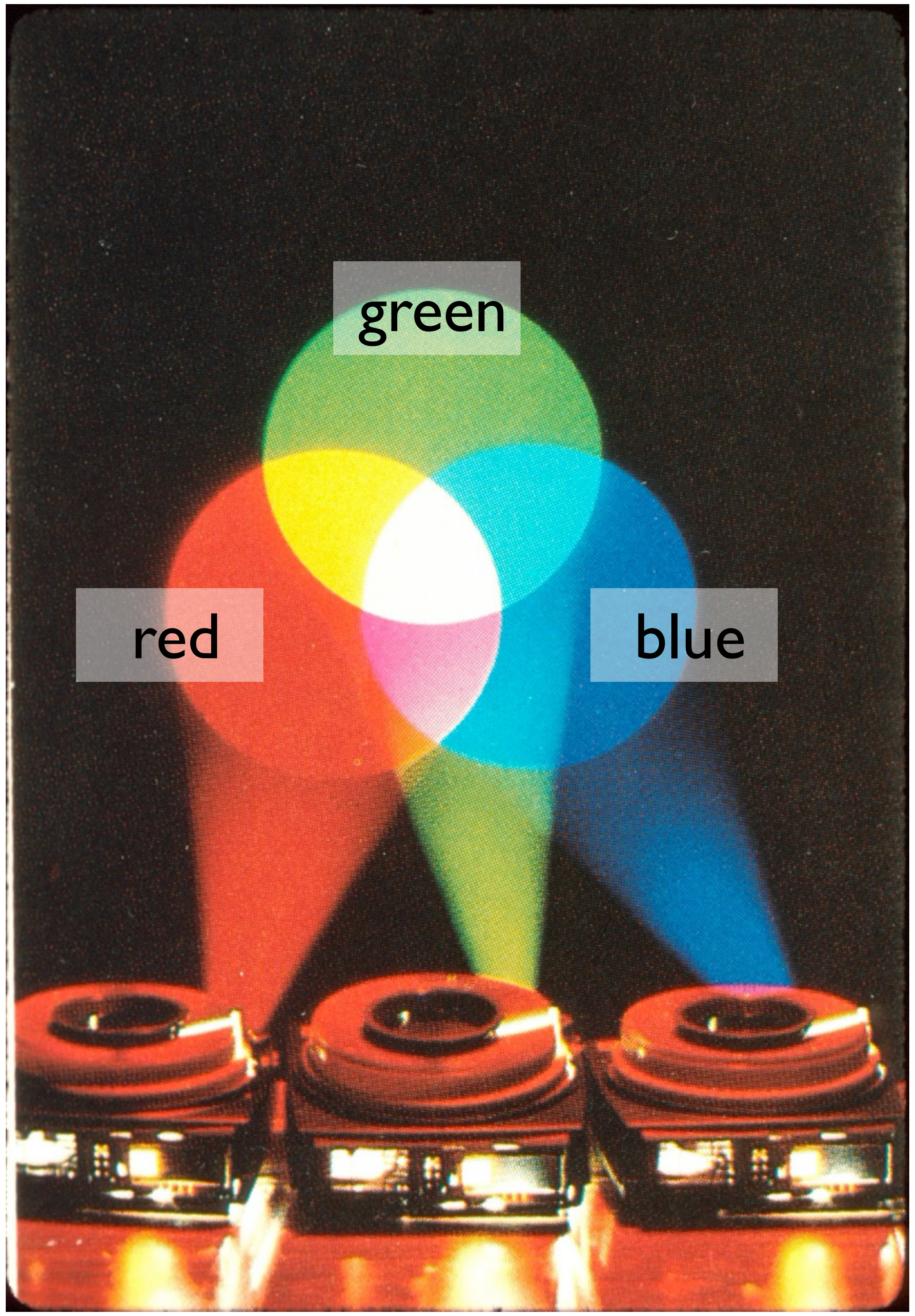
- Operating principle: humans are trichromatic
  - match any color with blend of three
  - therefore, problem reduces to producing 3 images and blending
- Additive color
  - blend images by sum
  - e.g. overlapping projection
  - e.g. unresolved dots
  - R, G, B make good primaries



[cs417 S02 slides]

Color displays

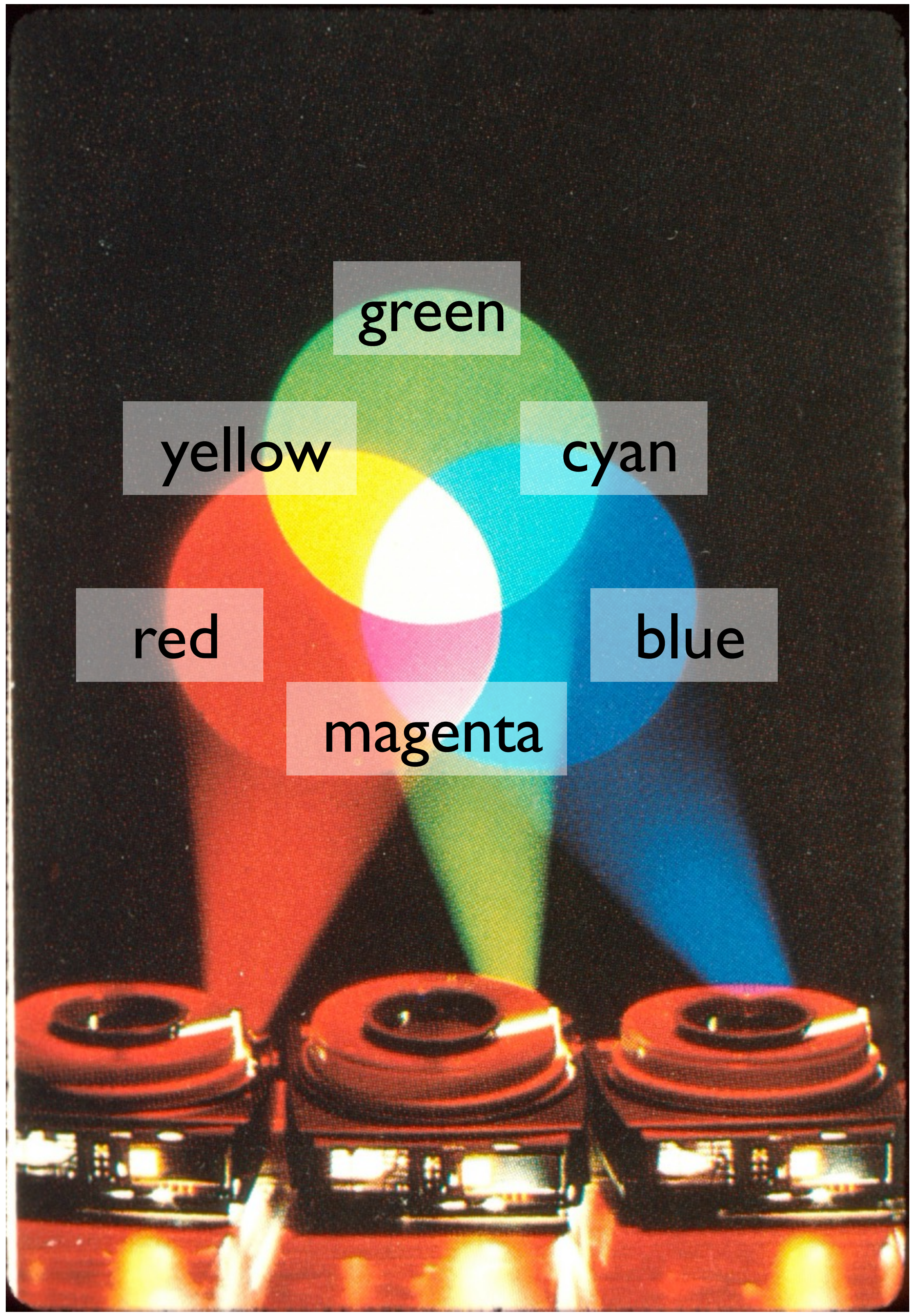
- Operating principle: humans are trichromatic
  - match any color with blend of three
  - therefore, problem reduces to producing 3 images and blending
- Additive color
  - blend images by sum
  - e.g. overlapping projection
  - e.g. unresolved dots
  - R, G, B make good primaries



[cs417 S02 slides]

Color displays

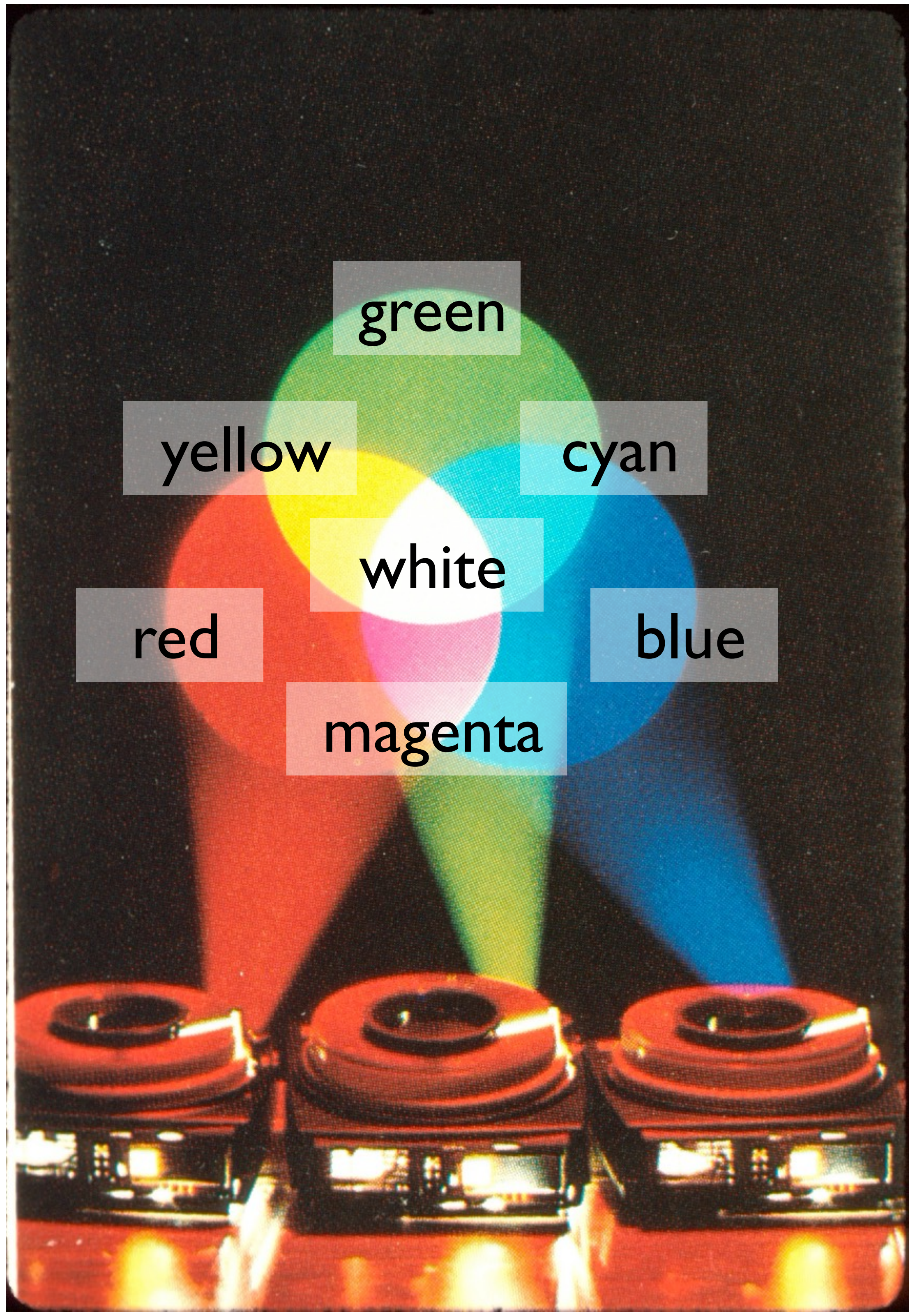
- Operating principle: humans are trichromatic
  - match any color with blend of three
  - therefore, problem reduces to producing 3 images and blending
- Additive color
  - blend images by sum
  - e.g. overlapping projection
  - e.g. unresolved dots
  - R, G, B make good primaries



[cs417 S02 slides]

Color displays

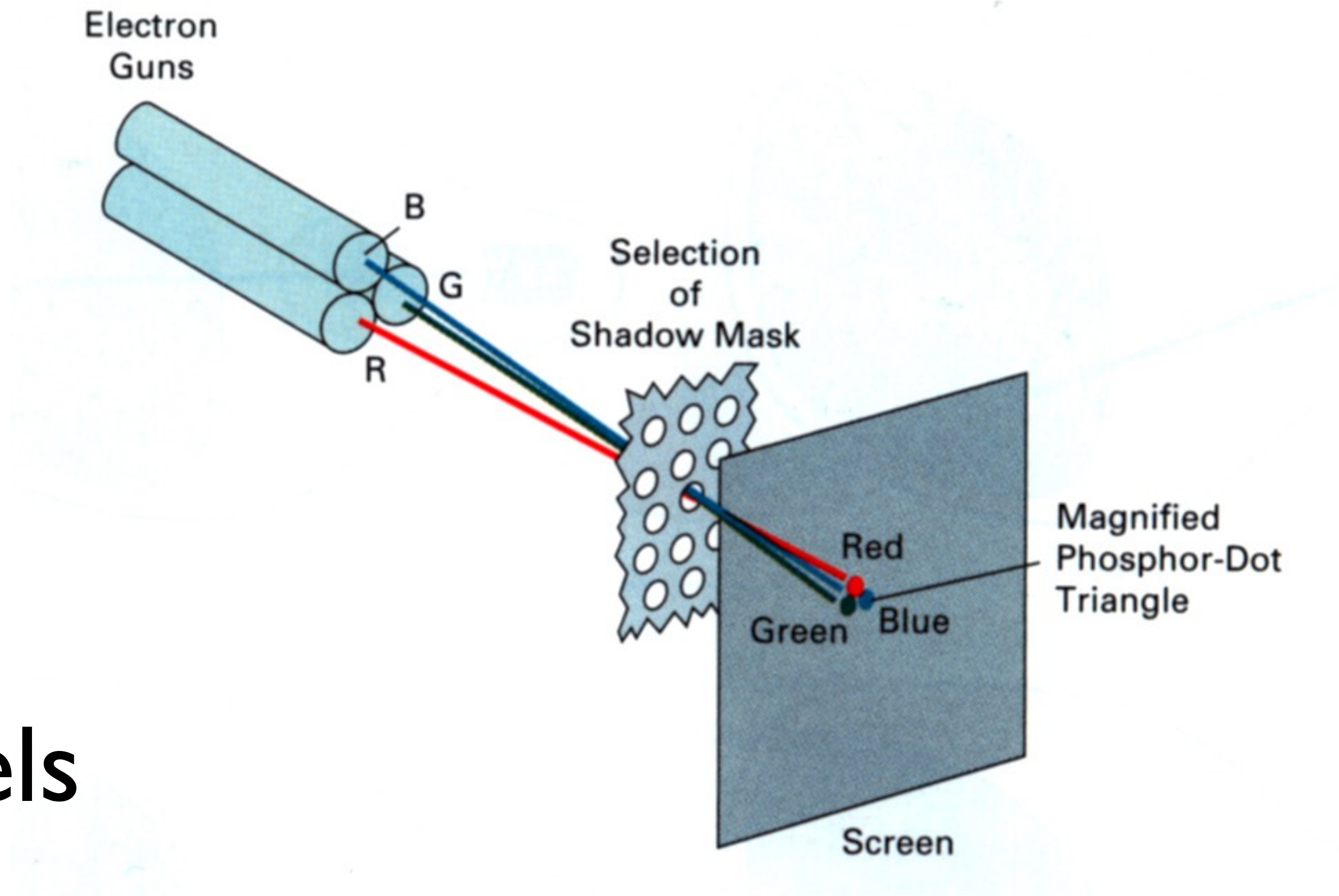
- Operating principle: humans are trichromatic
  - match any color with blend of three
  - therefore, problem reduces to producing 3 images and blending
- Additive color
  - blend images by sum
  - e.g. overlapping projection
  - e.g. unresolved dots
  - R, G, B make good primaries



[cs417 S02 slides]

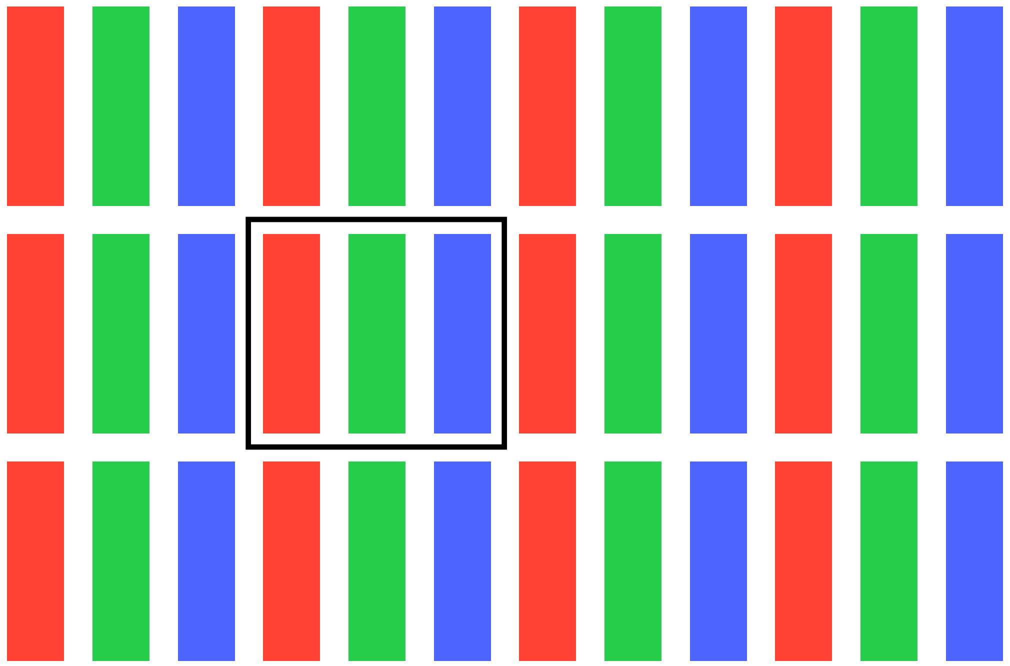
Color displays

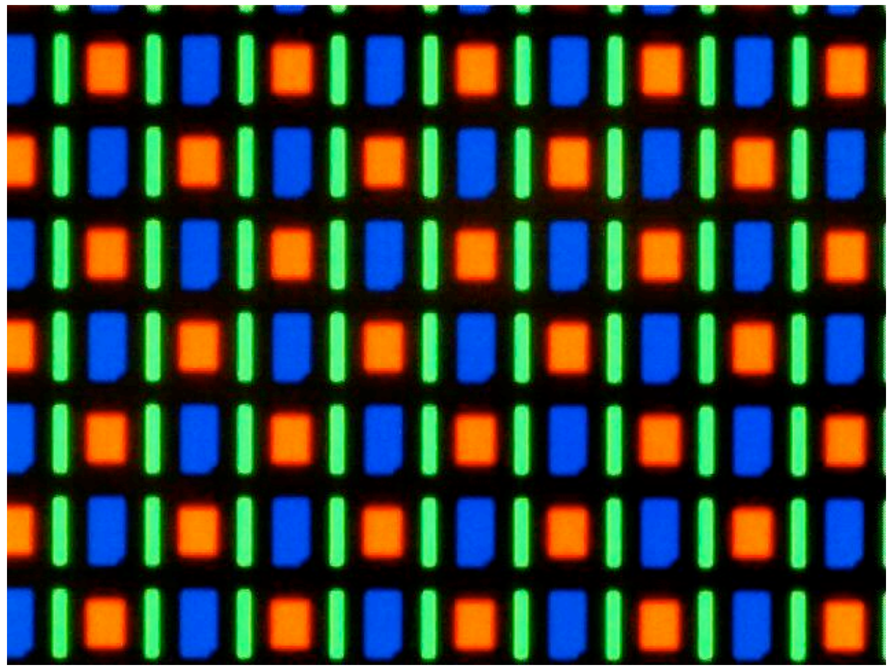
- CRT: phosphor dot pattern to produce finely interleaved color images



[H&B fig. 2-10]

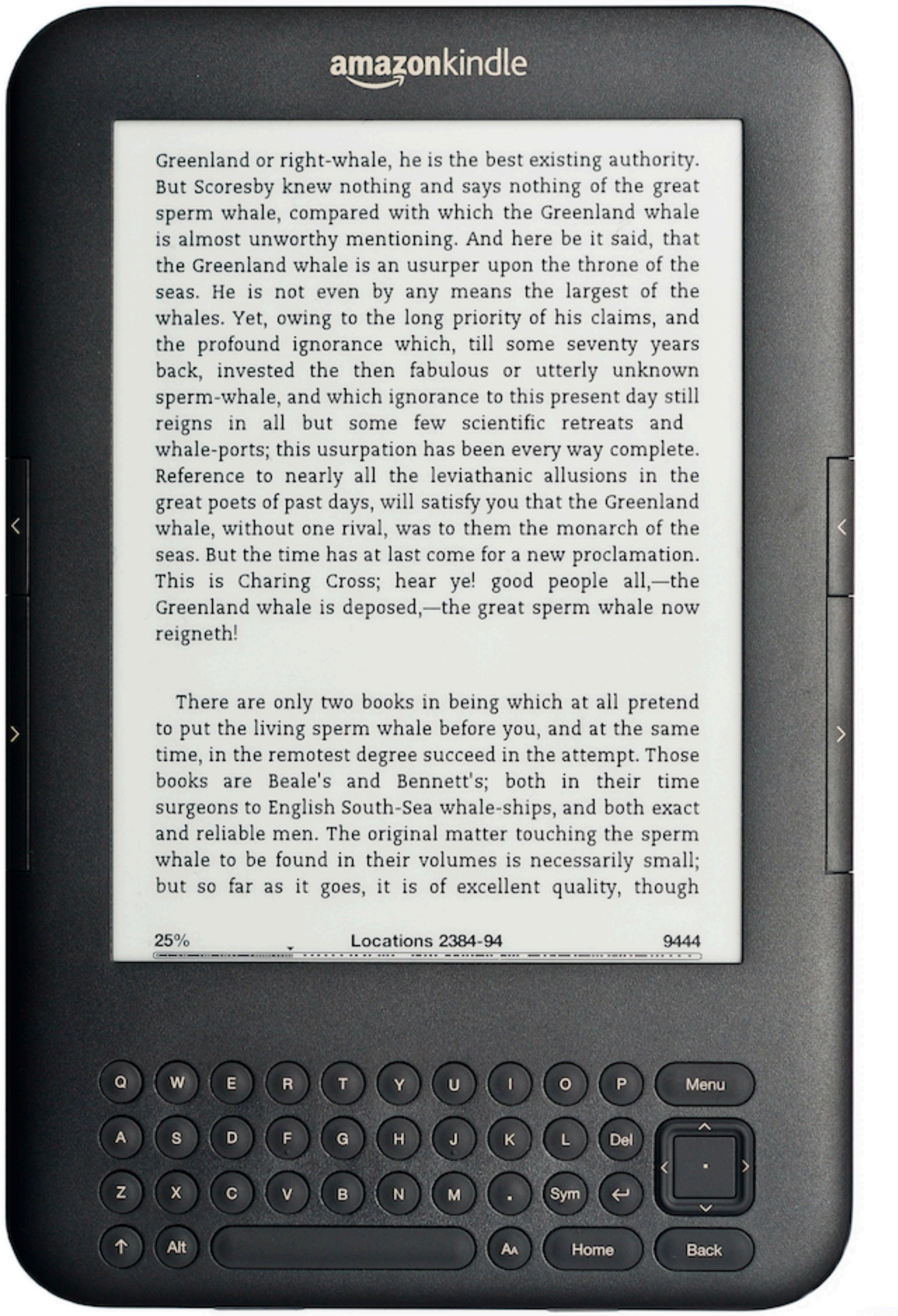
- LCD, LED: interleaved R,G,B pixels

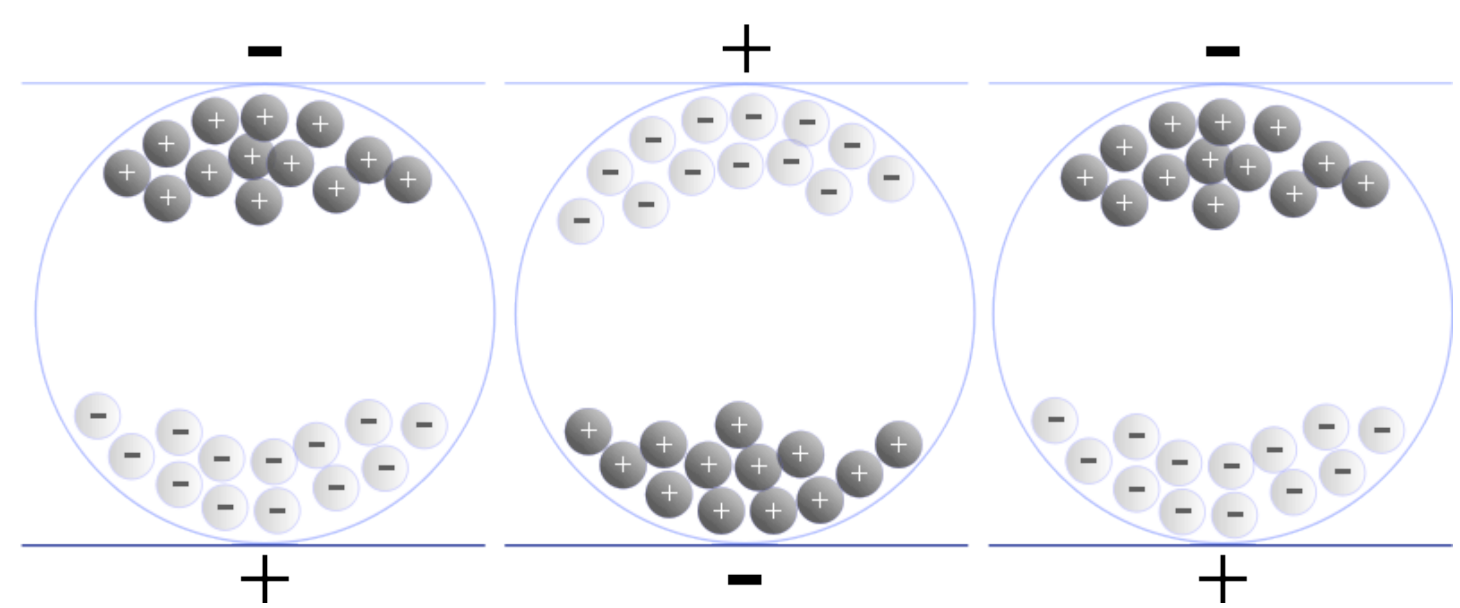




[Wikimedia Commons]

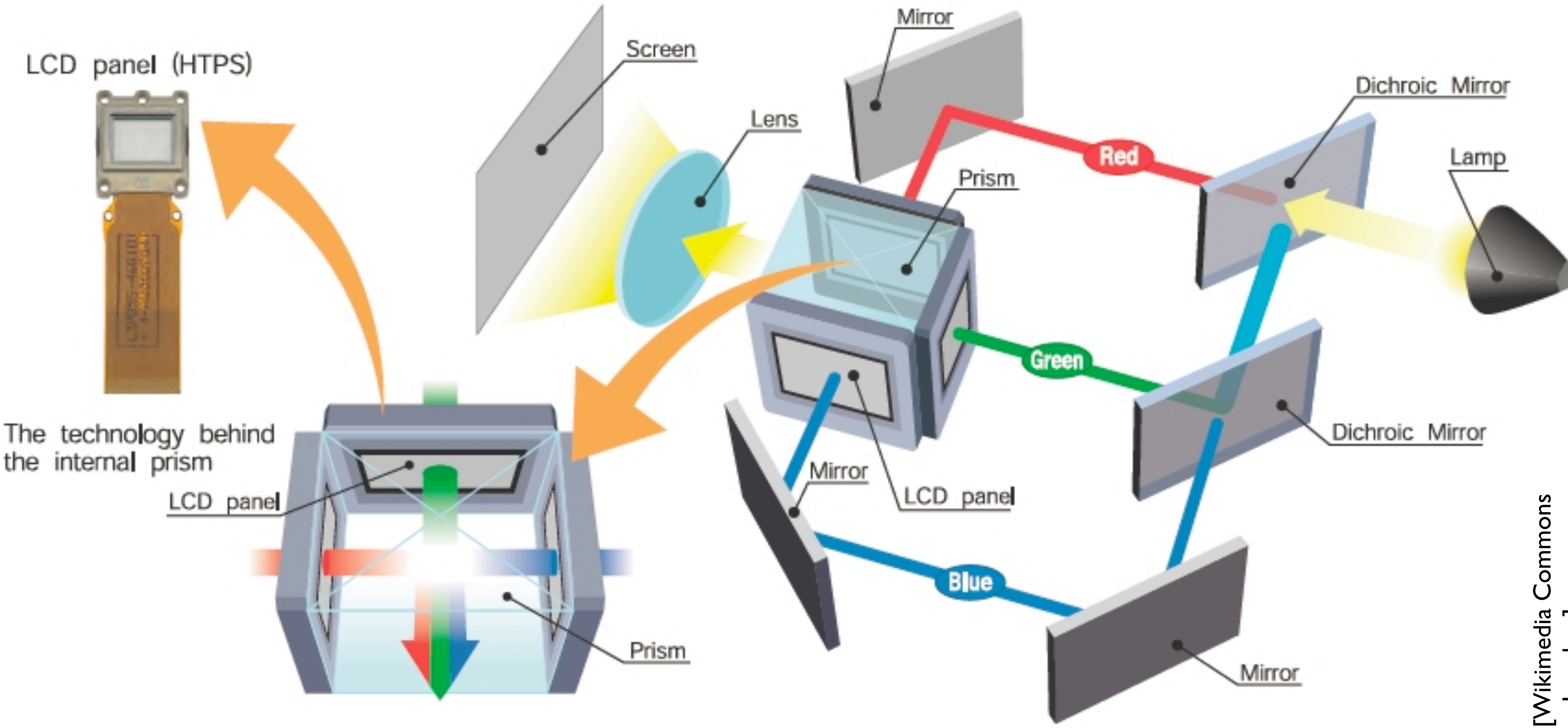
Electrophoretic (electronic ink)





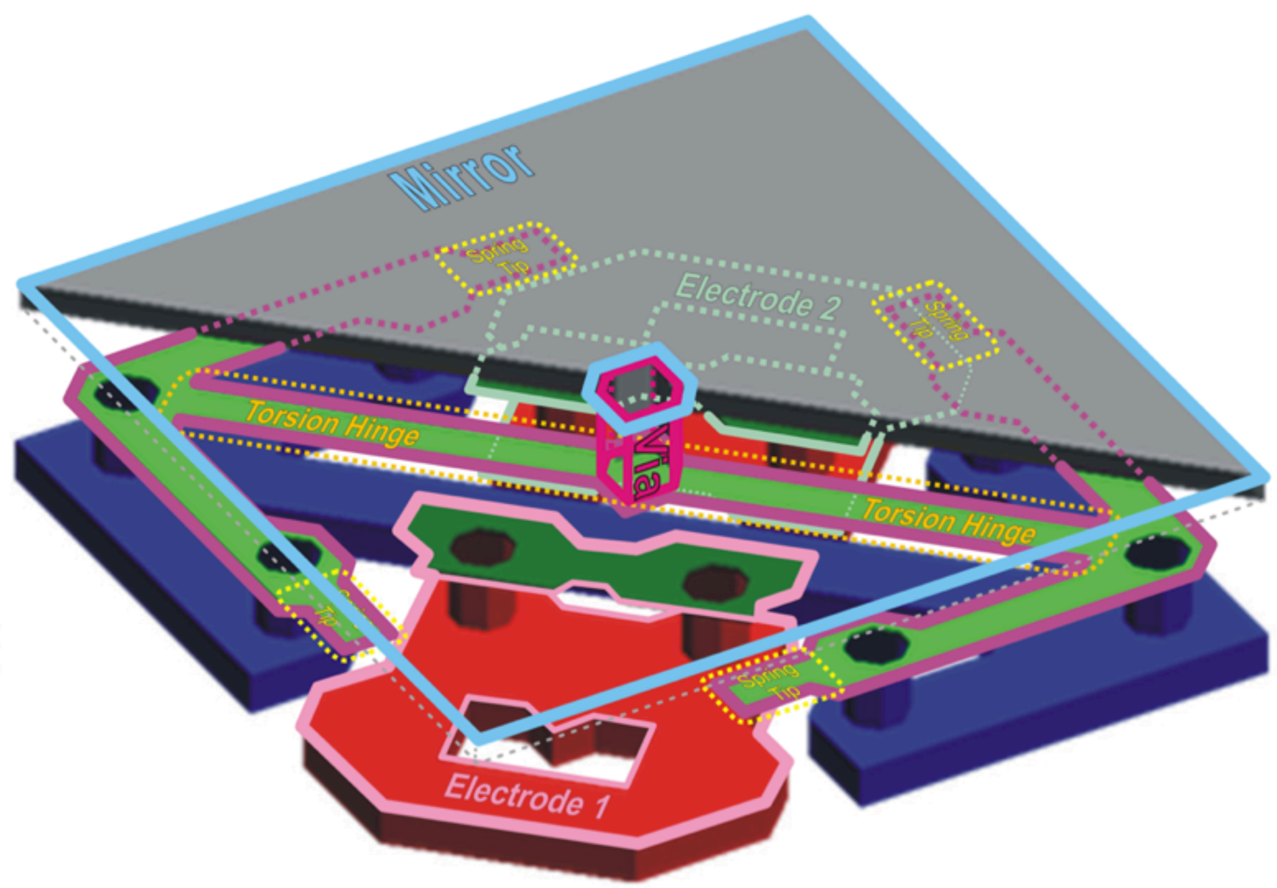
[Wikimedia Commons —Senarclens]

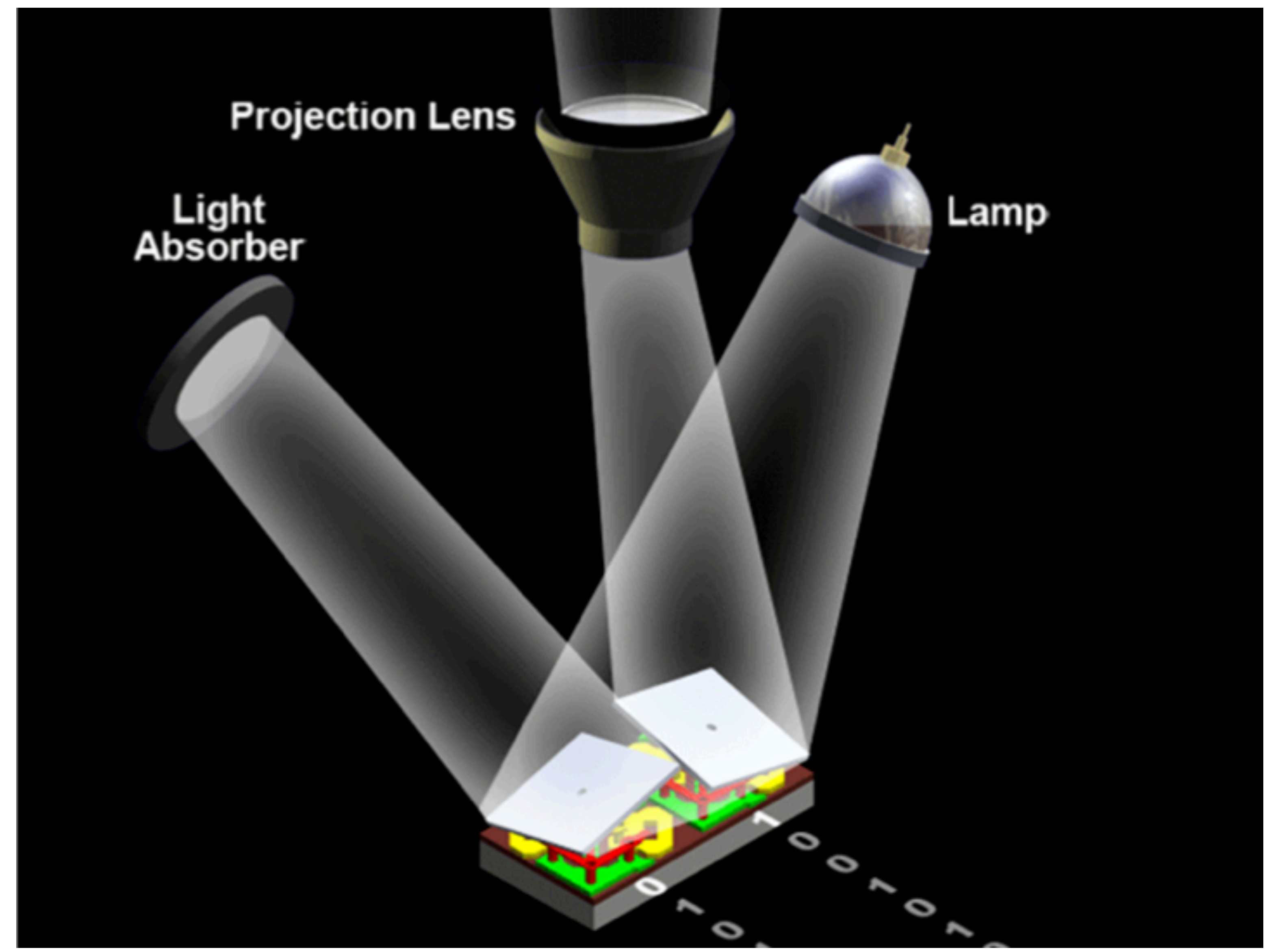
Projection displays: LCD



[Wikimedia Commons —Javachan]

Projection displays: DLP

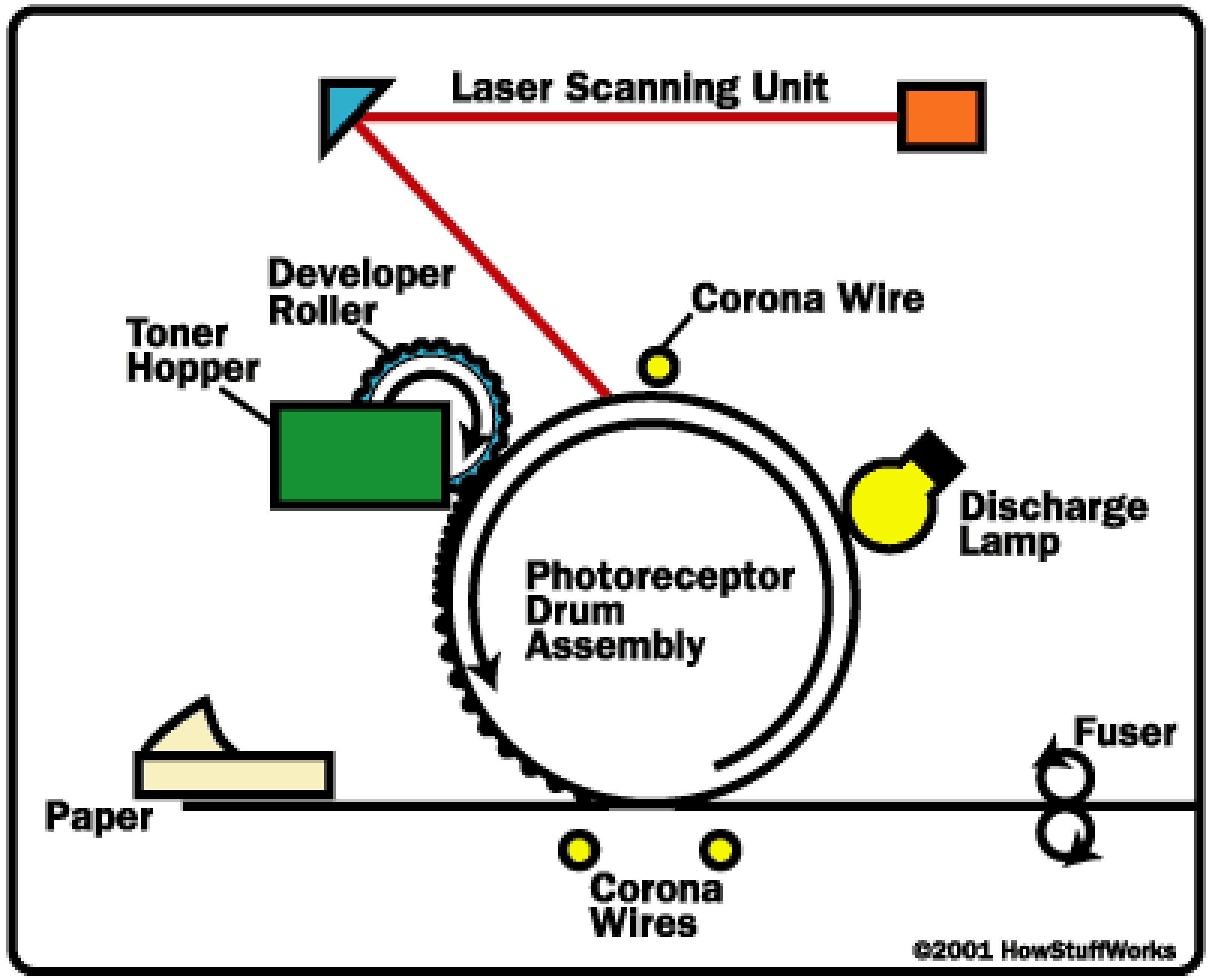




[Texas Instruments]

Laser printer

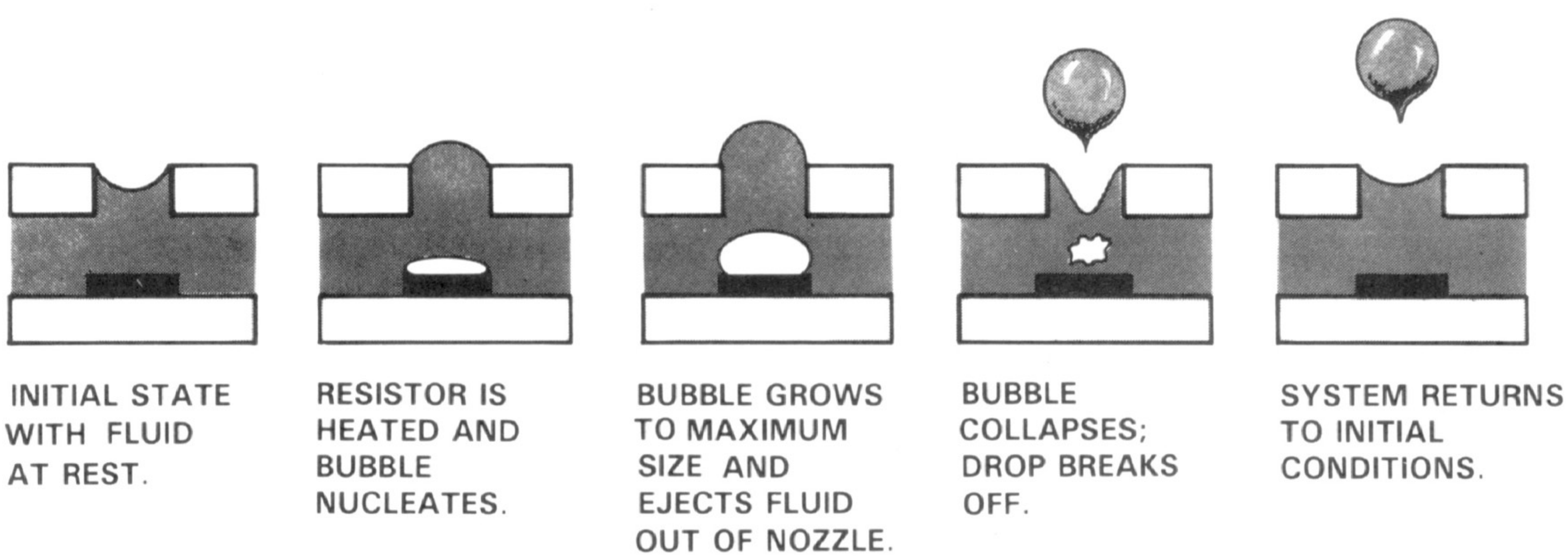
- Xerographic process
- Like a photocopier but with laser-scanned raster as source image
- Key characteristics
  - image is binary
  - resolution is high
  - limited minimum size of isolated dots



[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

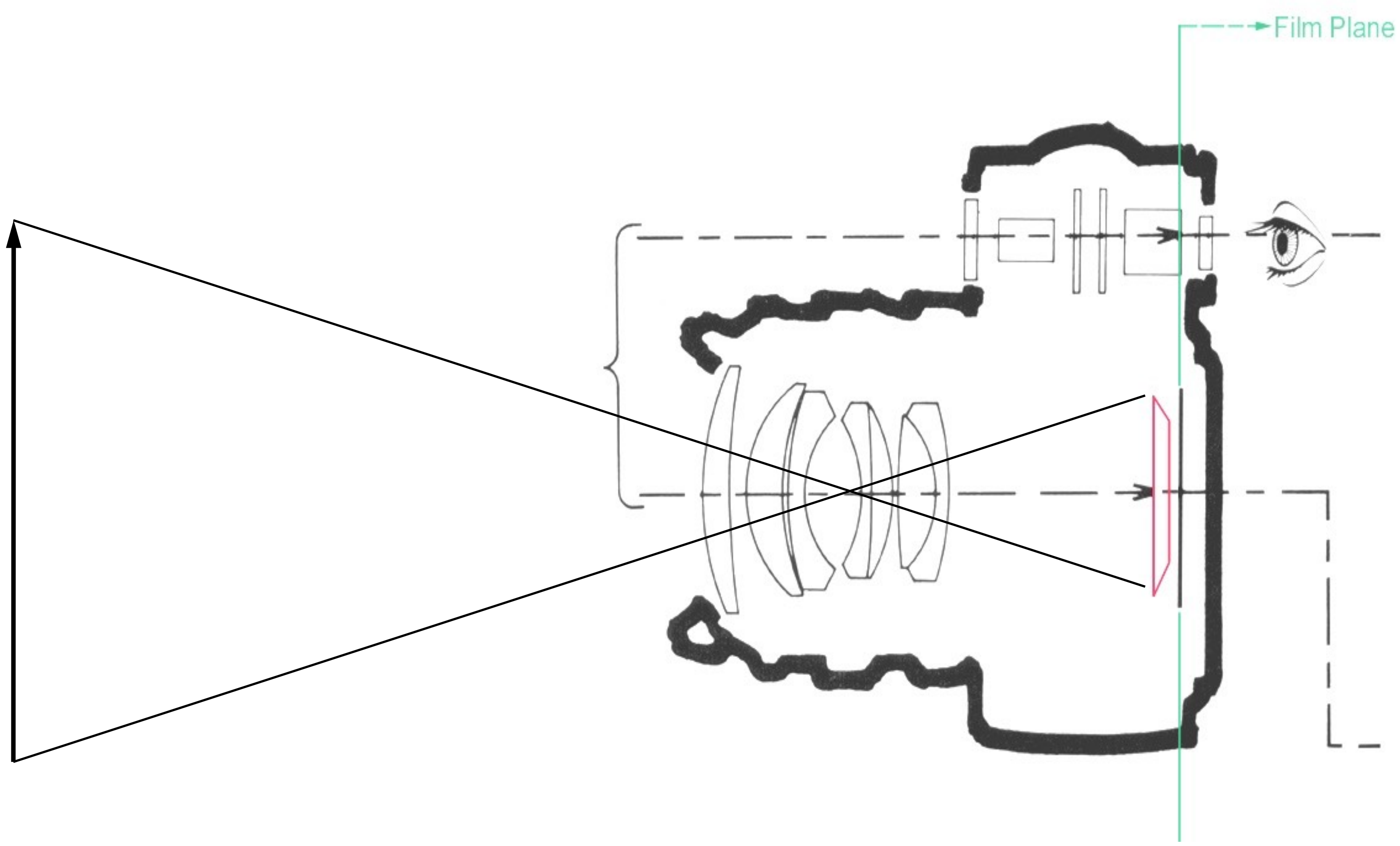


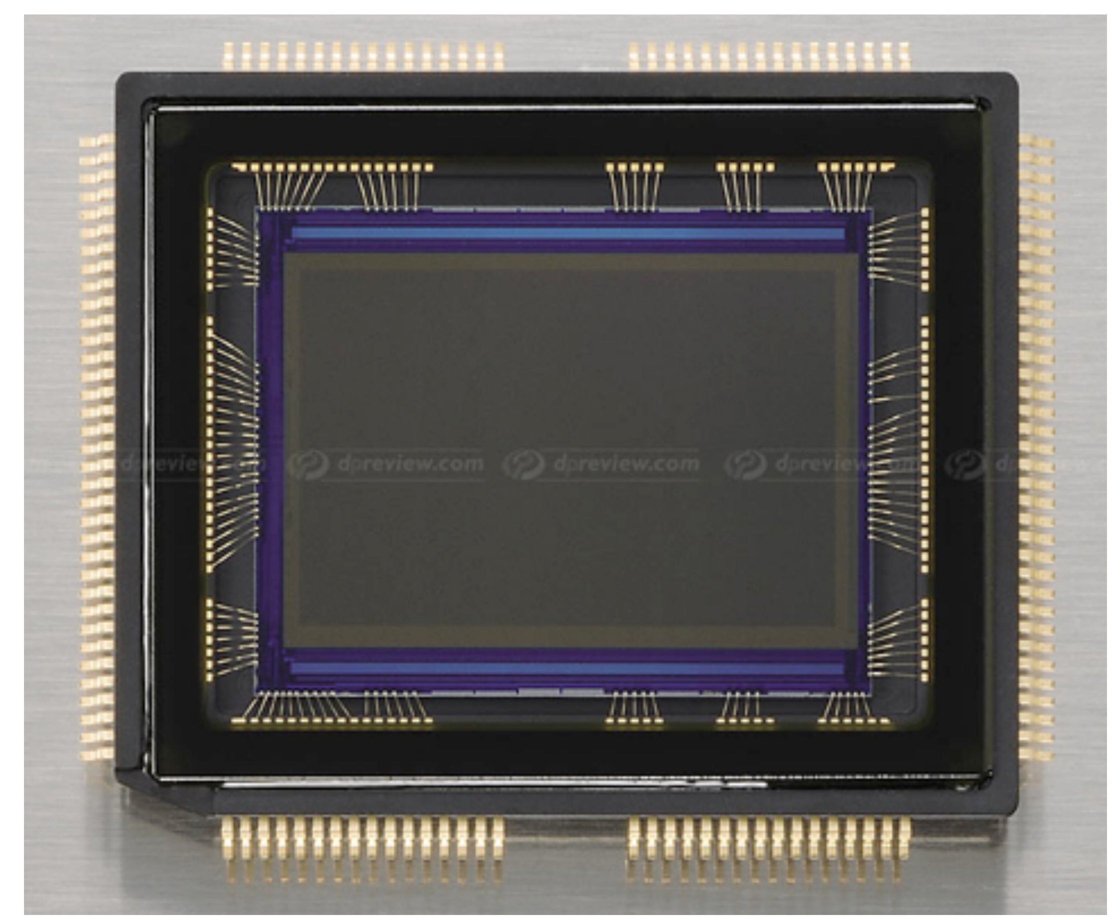
[cs417 S02 slides]

Digital camera

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

[CS 417 Spring 2002]

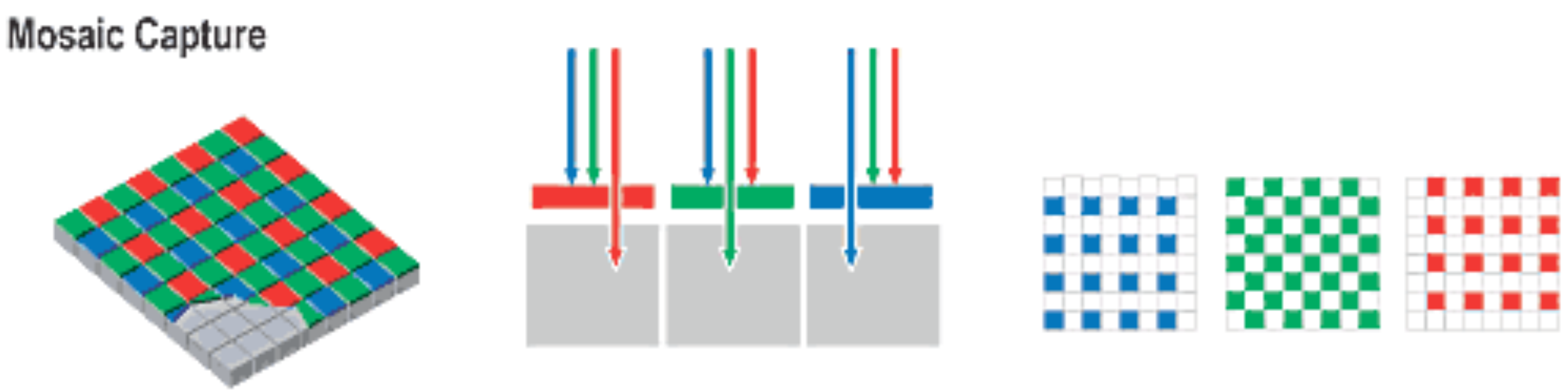




[dpreview.com]

Digital camera

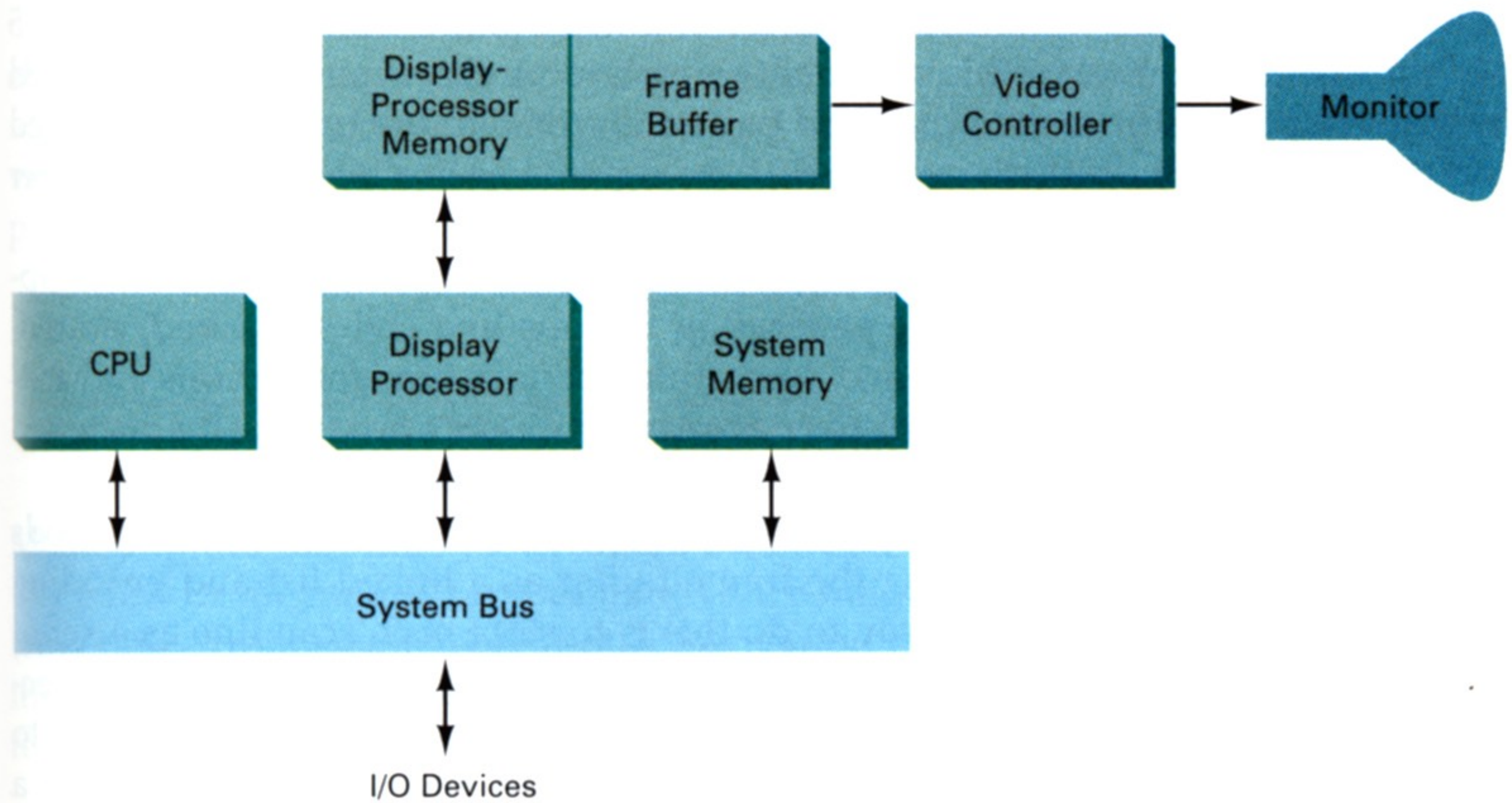
- Color typically captured using color mosaic



[Foveon]

Raster display system

- Screen image defined by a 2D array in RAM
- In older systems, and current high-end systems, it's in a separate memory from the normal CPU memory
- In low-end current systems it is just part of the main memory
- The *framebuffer* is the memory area that maps to the screen



[H&B fig. 2-29]