BBM 413 Fundamentals of Image Processing

Erkut Erdem
Dept. of Computer Engineering
Hacettepe University

Point Operations
Histogram Processing

Today's topics

- Point operations
- Histogram processing

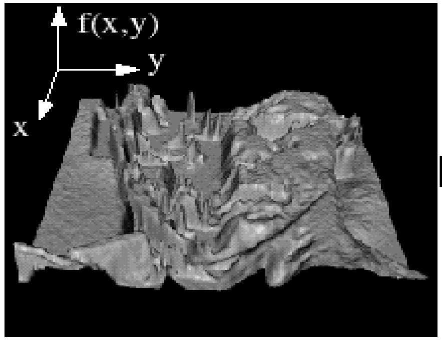
Today's topics

- Point operations
- Histogram processing

Digital images

- Sample the 2D space on a regular grid
- Quantize each sample (round to nearest integer)

- Image thus represented as a matrix of integer values.

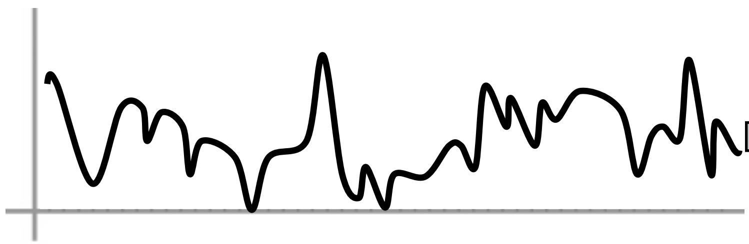


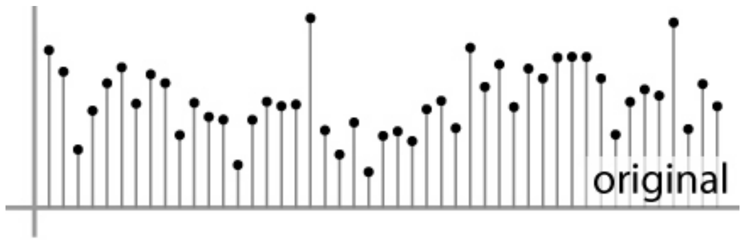
j

i

62	79	23	119	120	105	4	0
10	10	9	62	12	78	34	0
10	58	197	46	46	0	0	48
176	135	5	188	191	68	0	49
2	1	1	29	26	37	0	77
0	89	144	147	187	102	62	208
255	252	0	166	123	62	0	31
166	63	127	17	1	0	99	30

2D





1D

Slide credit: K. Grauman, S. Seitz

Image Transformations

- $g(x,y)=T[f(x,y)]$

$g(x,y)$: output image $f(x,y)$: input image   M: transformation function

1. Point operations: operations on single pixels
2. Spatial filtering: operations considering pixel neighborhoods
3. Global methods: operations considering whole image

Image Transformations

- g(x,y)=*T*[*f*(x,y)]

g(x,y): output image   *f*(x,y): input image   *M*: transformation function

1. Point operations: operations on single pixels
2. Spatial filtering: operations considering pixel neighborhoods
3. Global methods: operations considering whole image

$$g(x,y) = M(f(x,y))$$

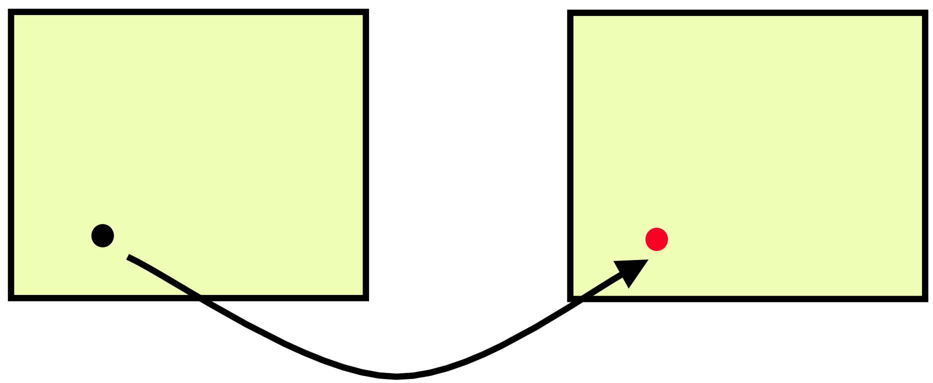


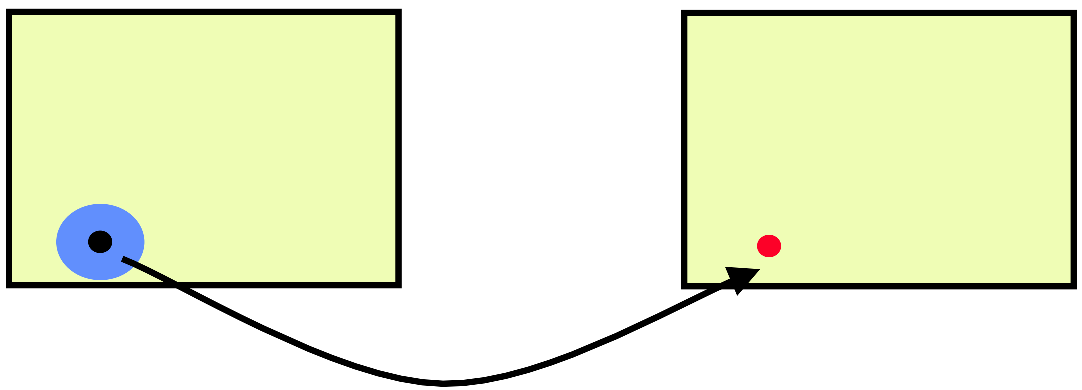
Image Transformations

- $g(x,y)=M[f(x,y)]$

$g(x,y)$: output image $f(x,y)$: input image   M: transformation function

1. Point operations: operations on single pixels
2. Spatial filtering: operations considering pixel neighborhoods
3. Global methods: operations considering whole image

$$g(x,y)=M(\{f(i,j)|(i,j)\in N(x,y)\})$$



Point operations

- Smallest possible neighborhood is of size 1x1
- Process each point independently of the others
- Output image *g* depends only on the value of *f* at a single point (x,y)
- Map each pixel's value to a new value
- Transformation function *T* remaps the sample's value:

$$s = T(r)$$

where

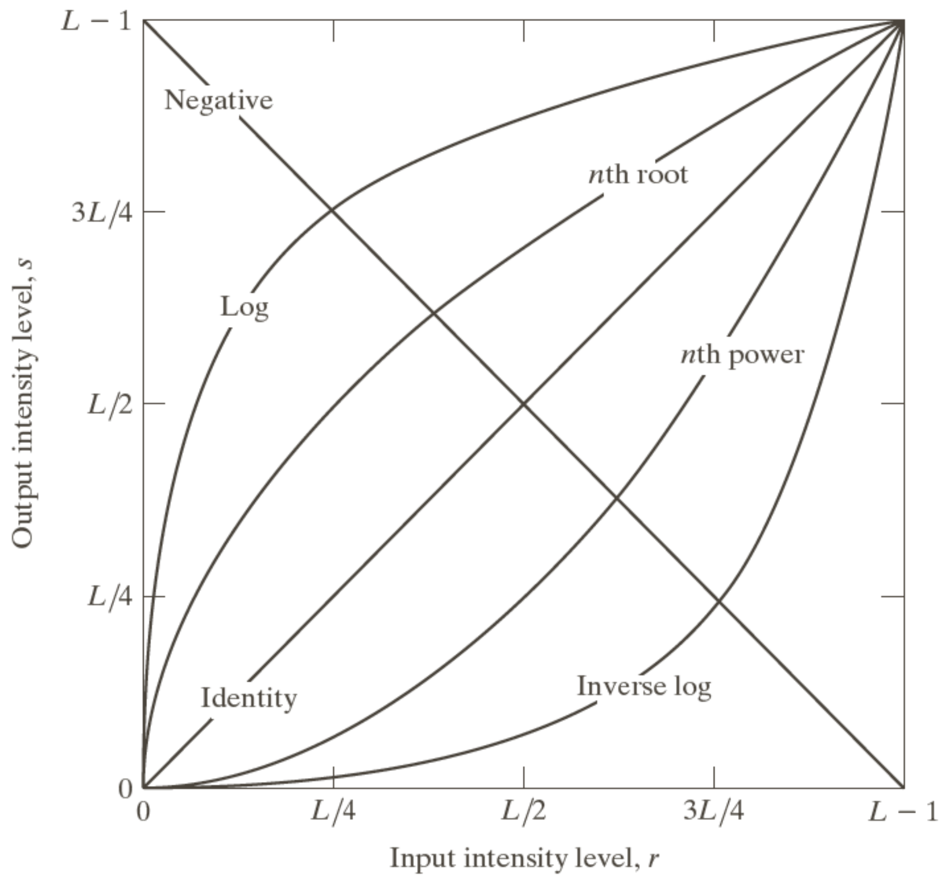
- r is the value at the point in question
- s is the new value in the processed result
- T is a *intensity transformation* function

Point operations

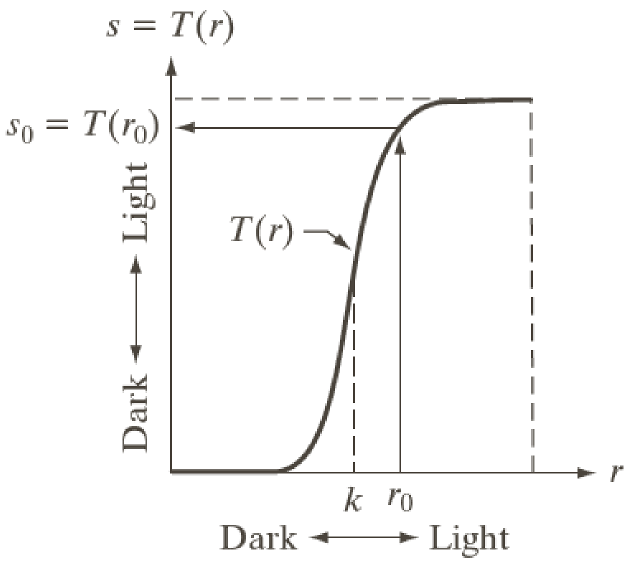
- Is mapping one color space to another (e.g. RGB2HSV) a point operation?
- Is image arithmetic a point operation?
- Is performing geometric transformations a point operation?
  - Rotation
  - Translation
  - Scale change
  - etc.

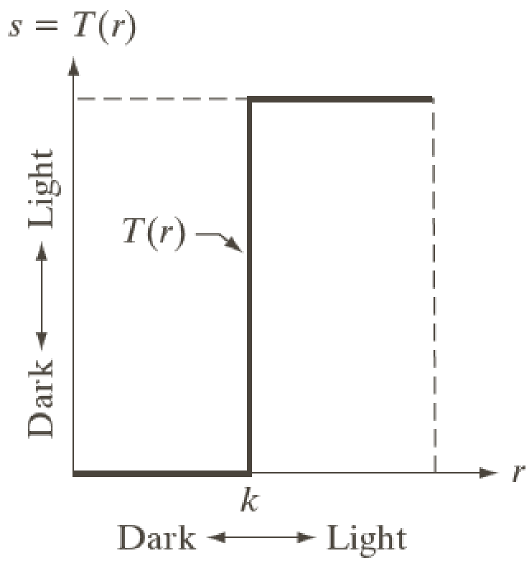
Sample intensity transformation functions

- Image negatives
- Log transformations
  - Compresses the dynamic range of images
- Power-law transformations
  - Gamma correction



Point Processing Examples



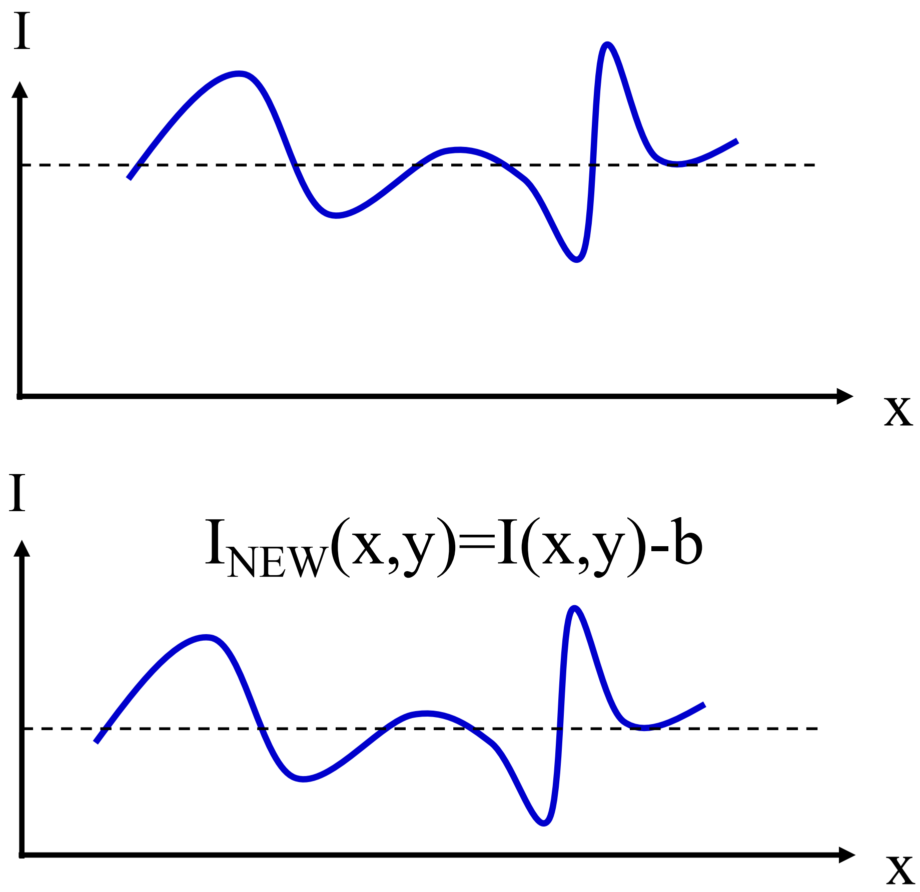


produces an image of higher contrast than the original by darkening the intensity levels below k and brightening intensities above k

produces a binary (two-intensity level) image

Image Mean

$$I_{av} = \frac{\sum_i \sum_j I(i,j)}{\sum_i \sum_j 1}$$



$I_{NEW}(x,y)=I(x,y)-b$

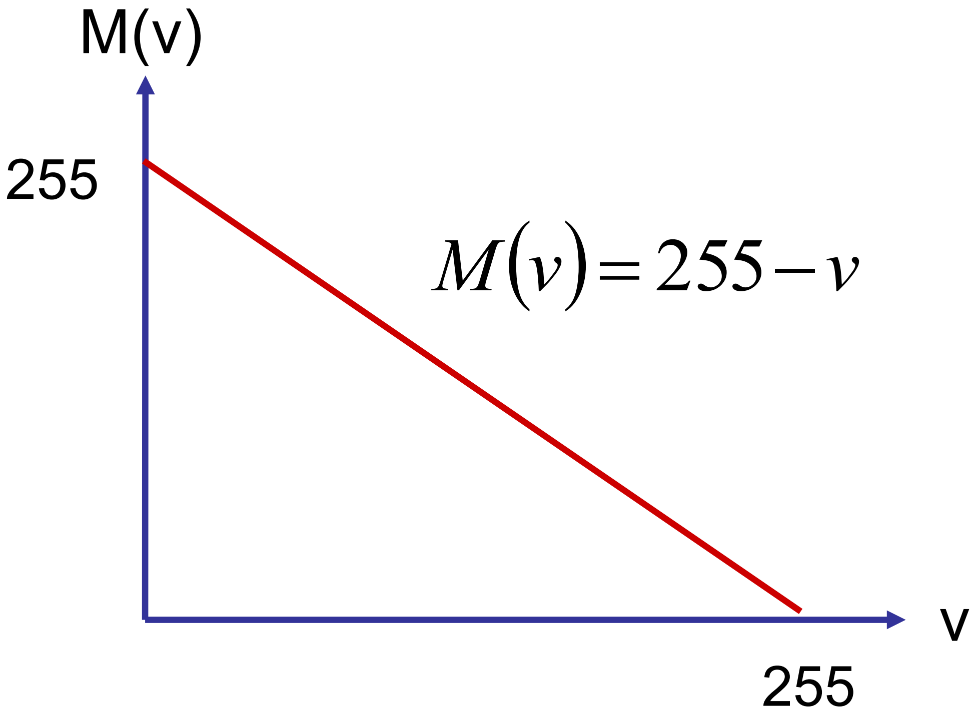
Slide credit: Y. Hel-Or

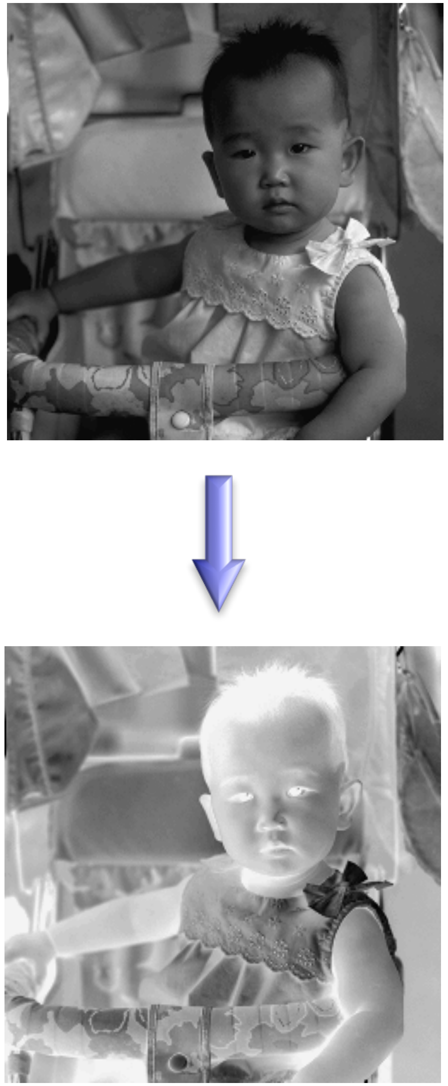
Image Mean

Changing the image mean

Slide credit: Y. Hel-Or

Image Negative

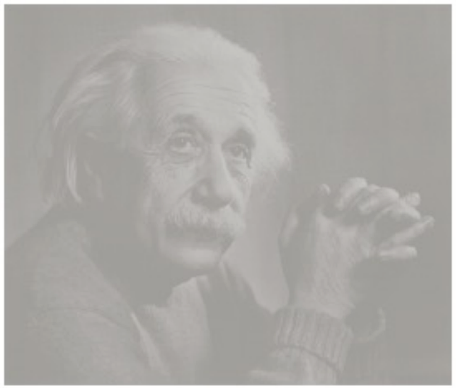




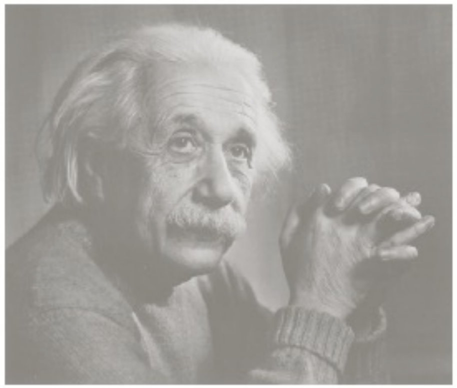
Slide credit: Y. Hel-Or

Dynamic range

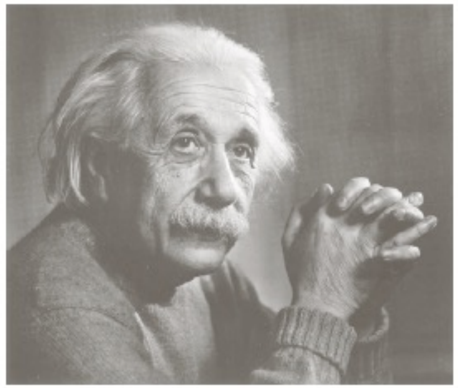
- Dynamic range $R_d = I_{\max} / I_{\min}$, or $(I_{\max} + k) / (I_{\min} + k)$
  - determines the degree of image contrast that can be achieved
  - a major factor in image quality
- Ballpark values
  - Desktop display in typical conditions: 20:1
  - Photographic print: 30:1
  - High dynamic range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



low contrast



medium contrast

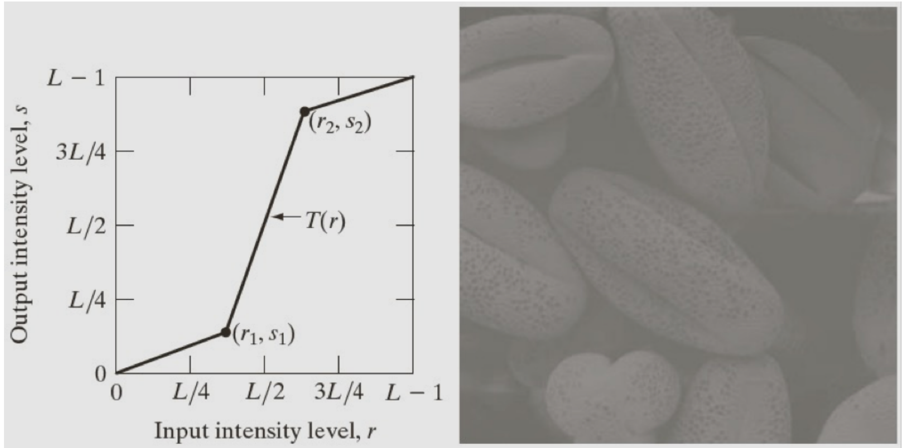


high contrast

Slide credit: S. Marschner

Point Operations: Contrast stretching and Thresholding

- <u>Contrast stretching:</u> produces an image of higher contrast than the original



- <u>Thresholding:</u> produces a binary (two-intensity level) image

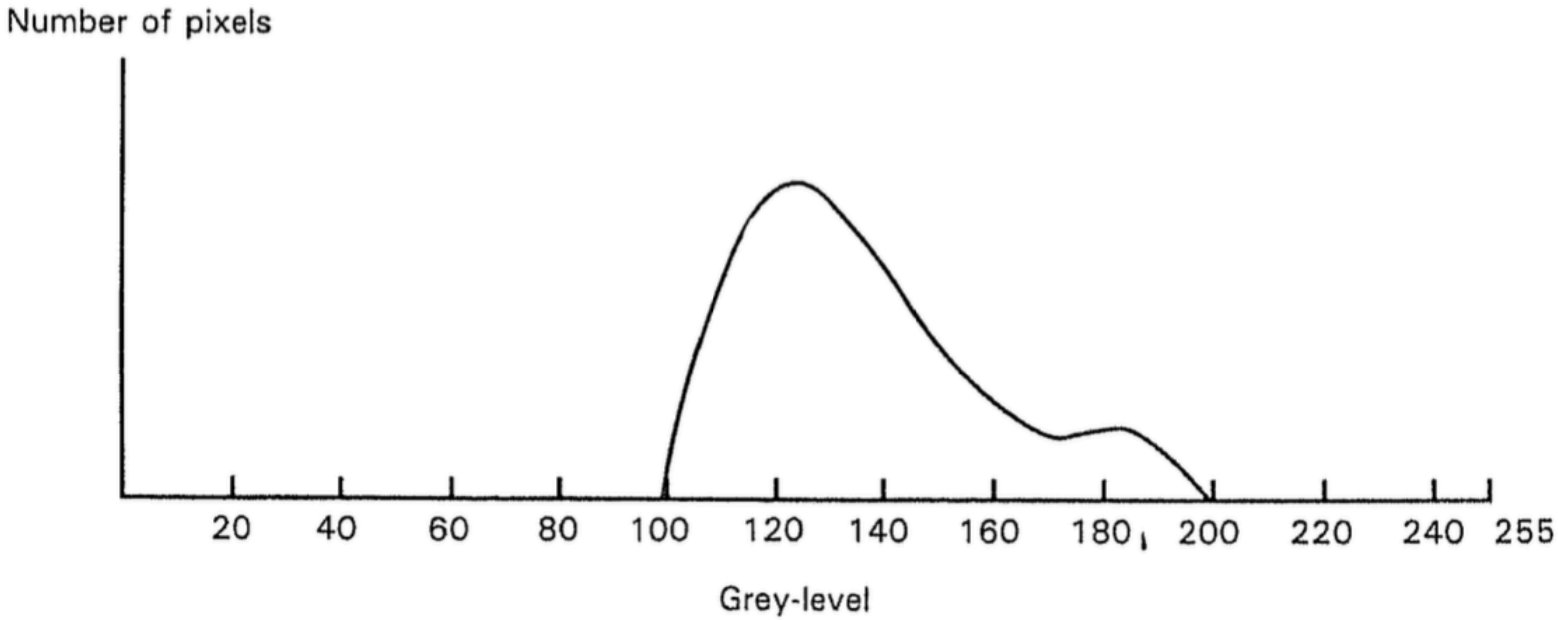
Point Operations: Contrast stretching and Thresholding

- <u>Contrast stretching:</u> produces an image of higher contrast than the original

- <u>Thresholding:</u> produces a binary (two-intensity level) image

Point Operations

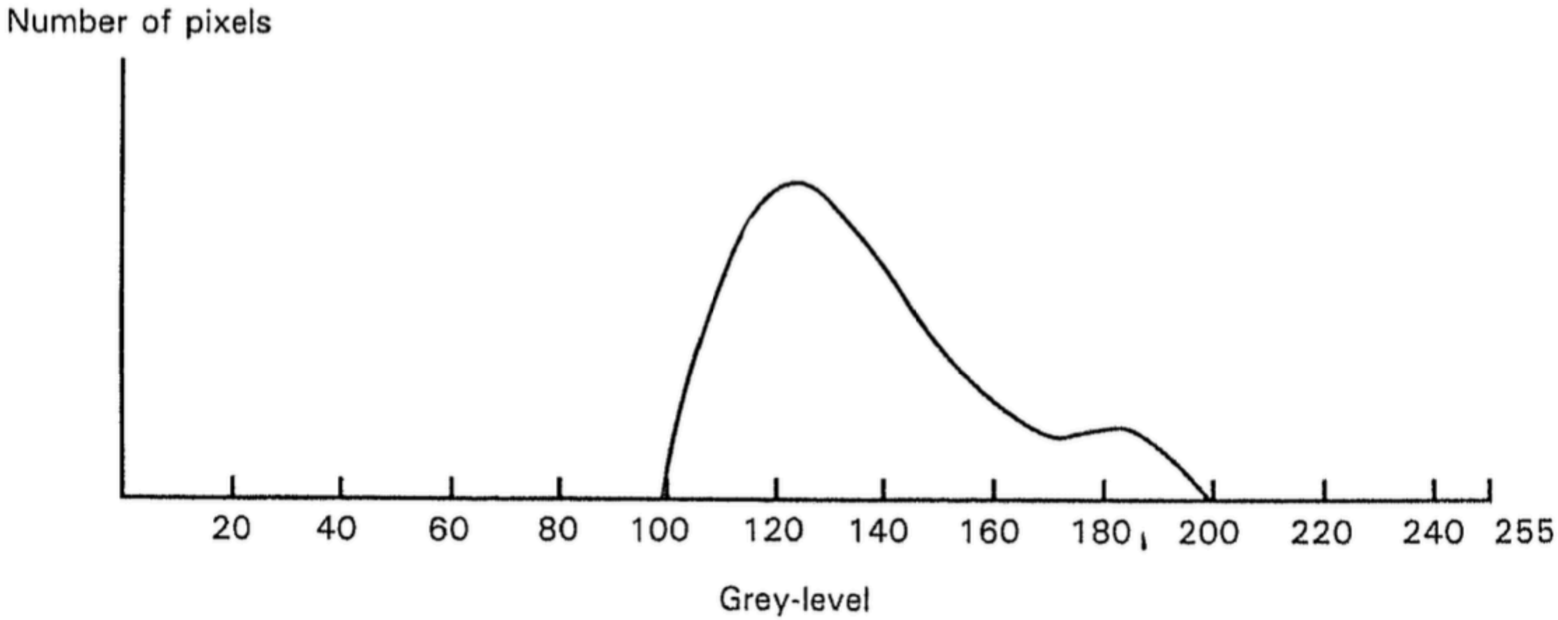
- What can you say about the image having the following histogram?



- A low contrast image
- How we can process the image so that it has a better visual quality?

Point Operations

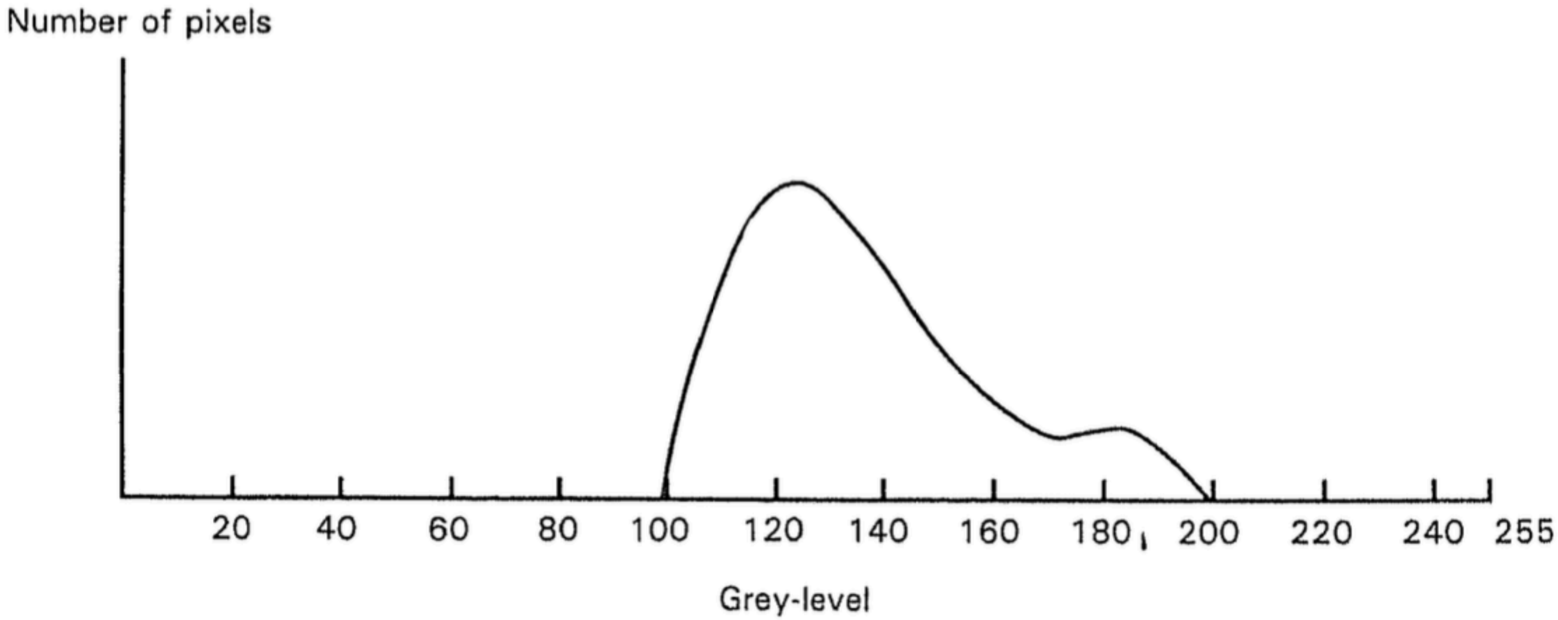
- How we can process the image so that it has a better visual quality?



- Answer is contrast stretching!

Point Operations

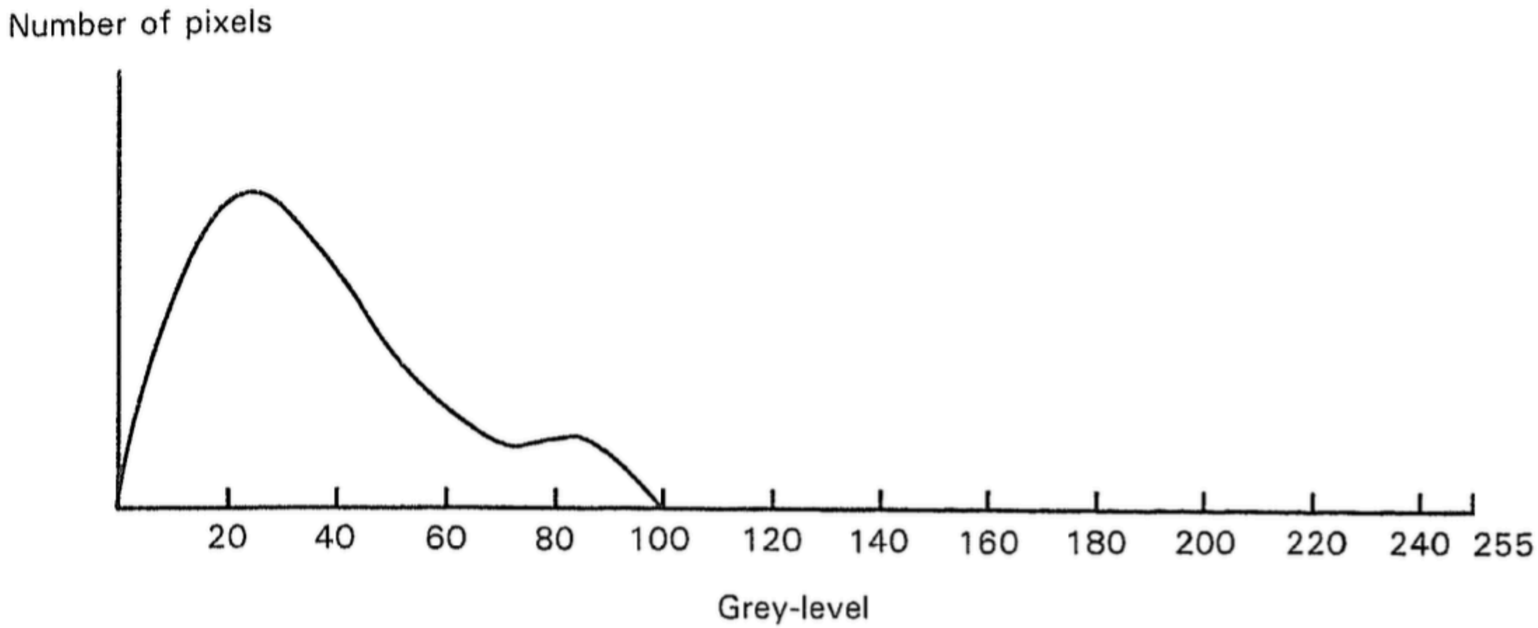
- Let us devise an appropriate point operation.



- Shift all values so that the observable pixel range starts at 0.

Point Operations

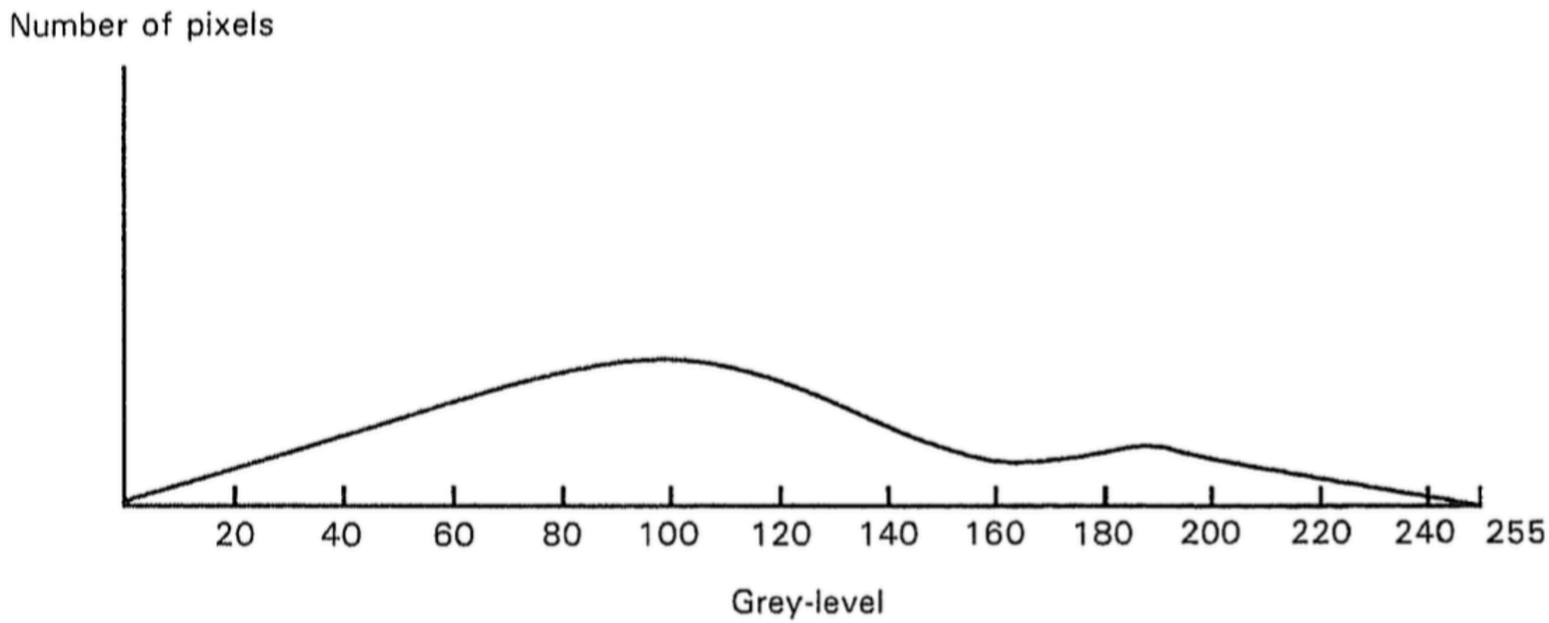
- Let us devise an appropriate point operation.



- Now, scale everything in the range 0-100 to 0-255.

Point Operations

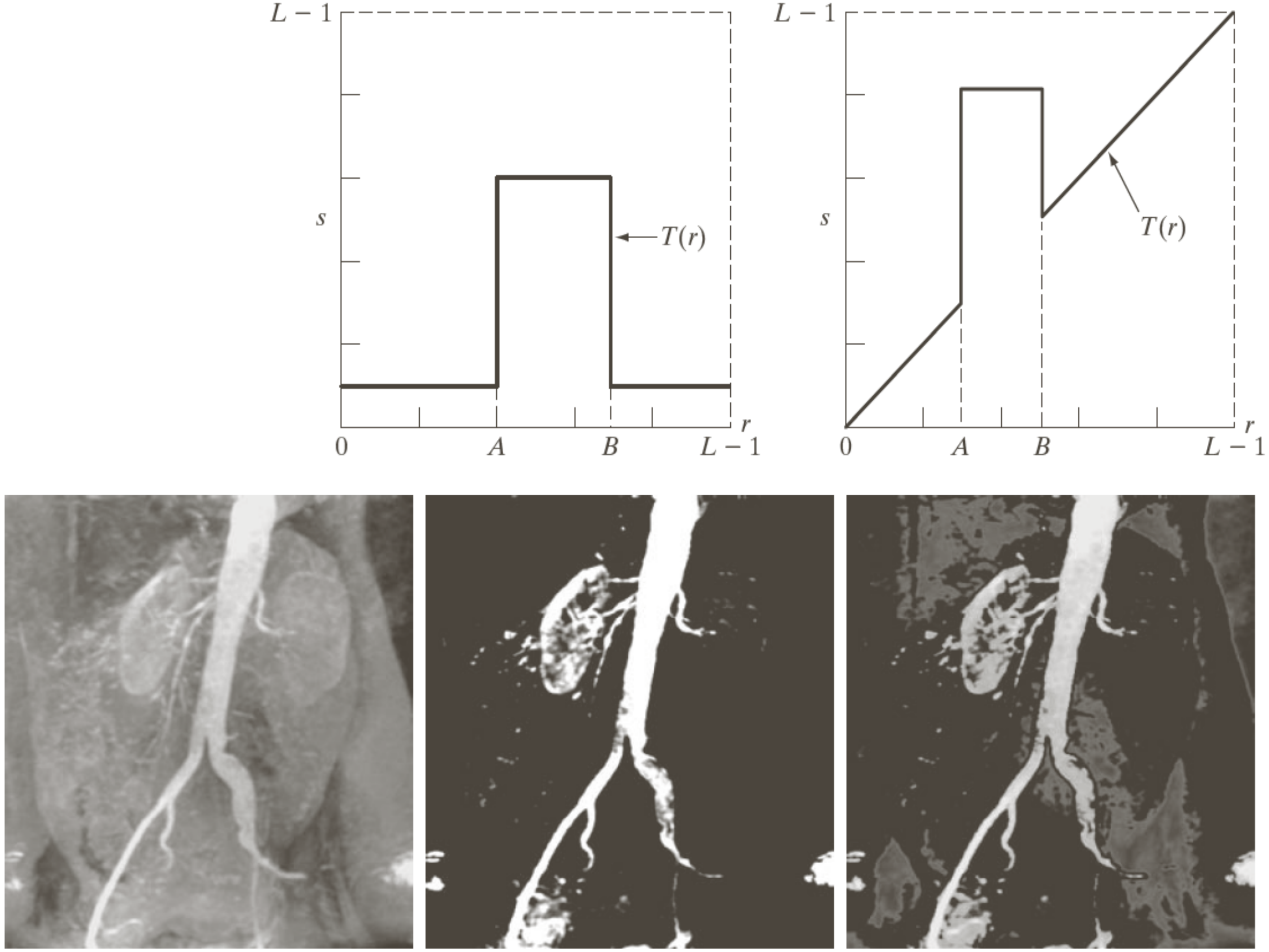
- Let us devise an appropriate point operation.



- What is the corresponding transformation function?
- T(r) = 2.55*(r-100)

Point Operations: Intensity-level Slicing

- highlights a certain range of intensities

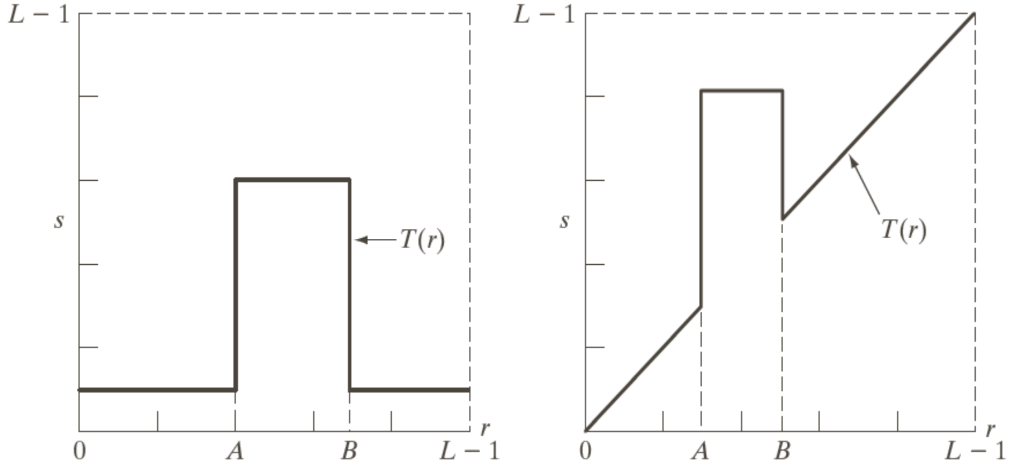


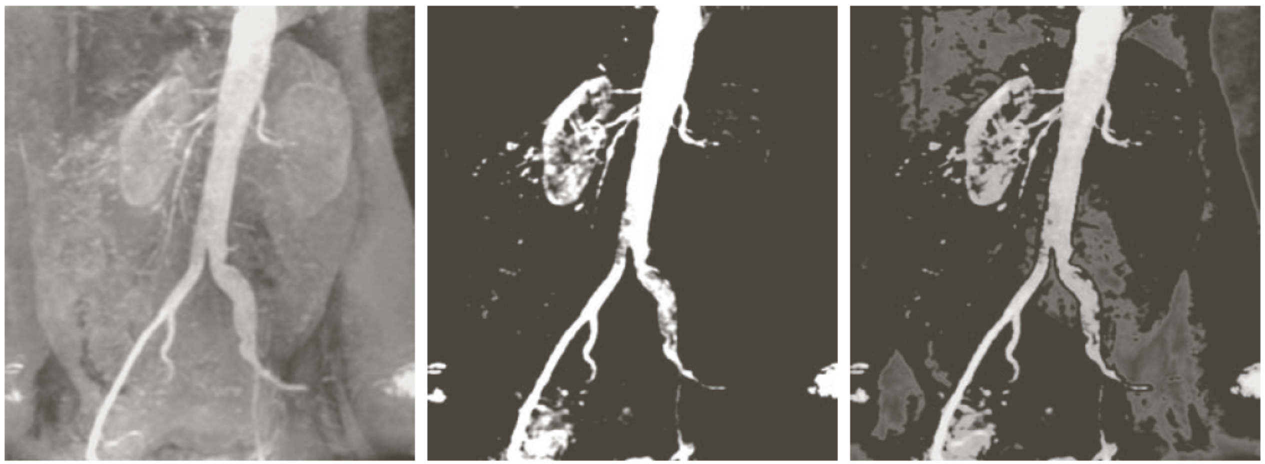
Point Operations: Intensity-level Slicing

- highlights a certain range of intensities

a b

FIGURE 3.11 (a) This transformation highlights intensity range $[A, B]$ and reduces all other intensities to a lower level. (b) This transformation highlights range $[A, B]$ and preserves all other intensity levels.





Intensity encoding in images

- Recall that the pixel values determine how bright that pixel is.
- Bigger numbers are (usually) brighter
- *Transfer function*: function that maps input pixel value to luminance of displayed image

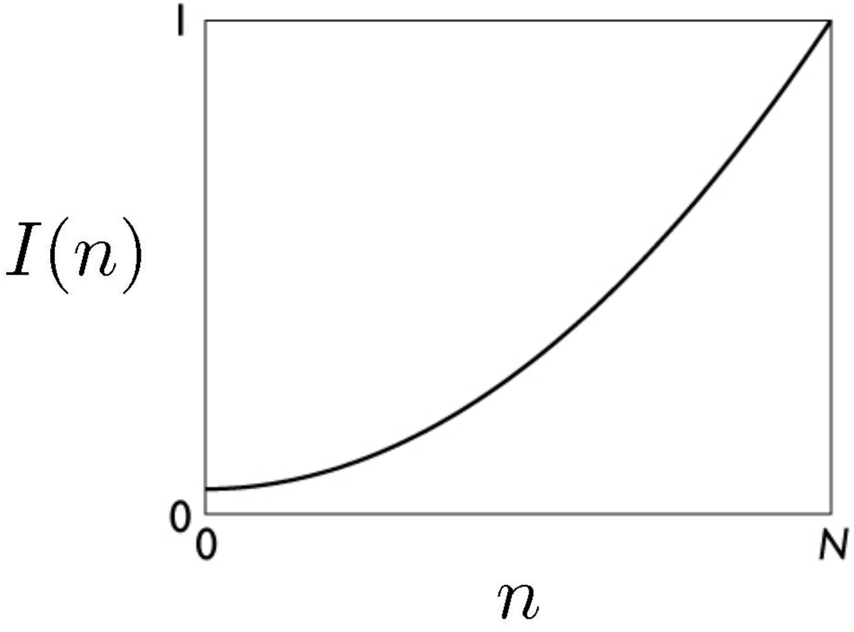
$$I = f(n) \quad f : [0, N] \rightarrow [I_{\min}, I_{\max}]$$

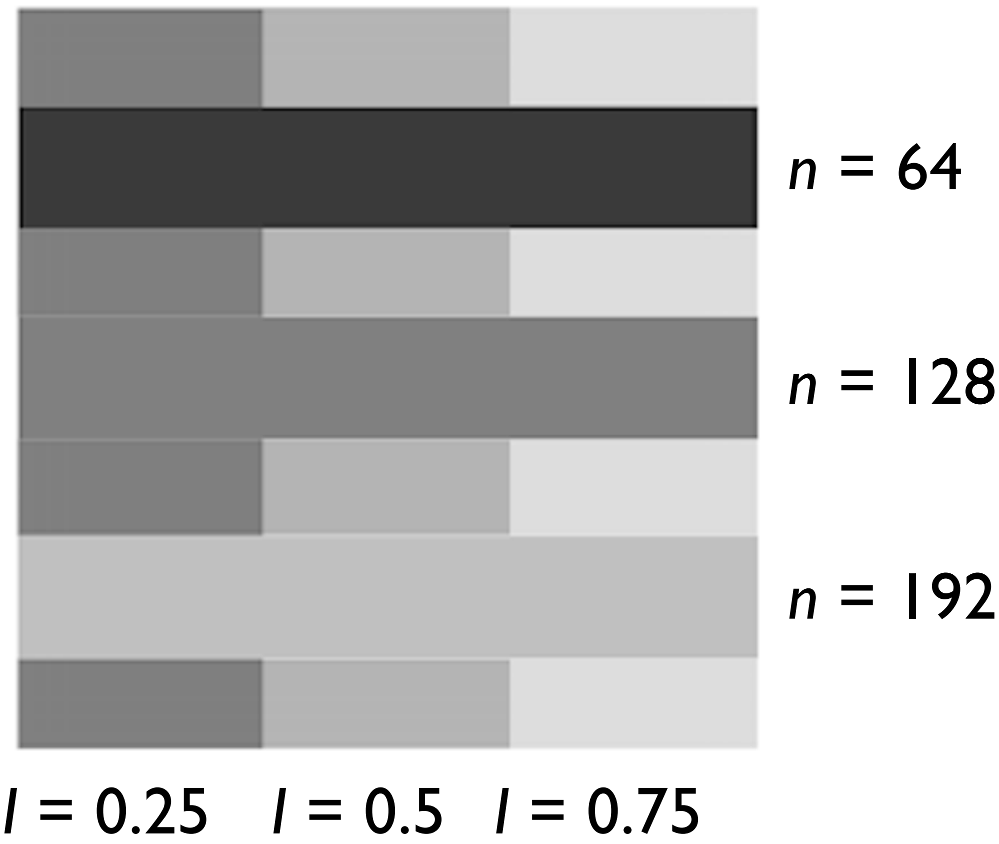
- What determines this function?
  - physical constraints of device or medium
  - desired visual characteristics

adapted from: S. Marschner

What this projector does?

- Something like this:





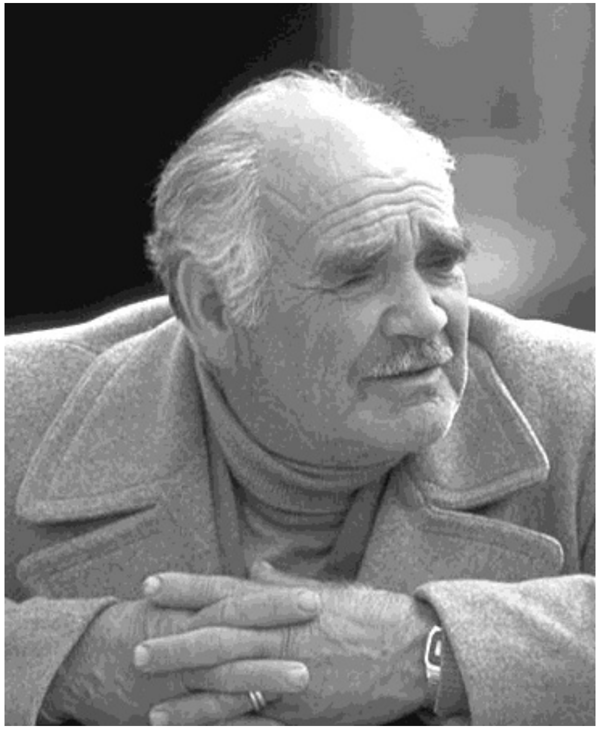
adapted from: S. Marschner

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

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]

Slide credit: S. Marschner

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 $I_{\min} \cdot R_d$ 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

Slide credit: S. Marschner

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

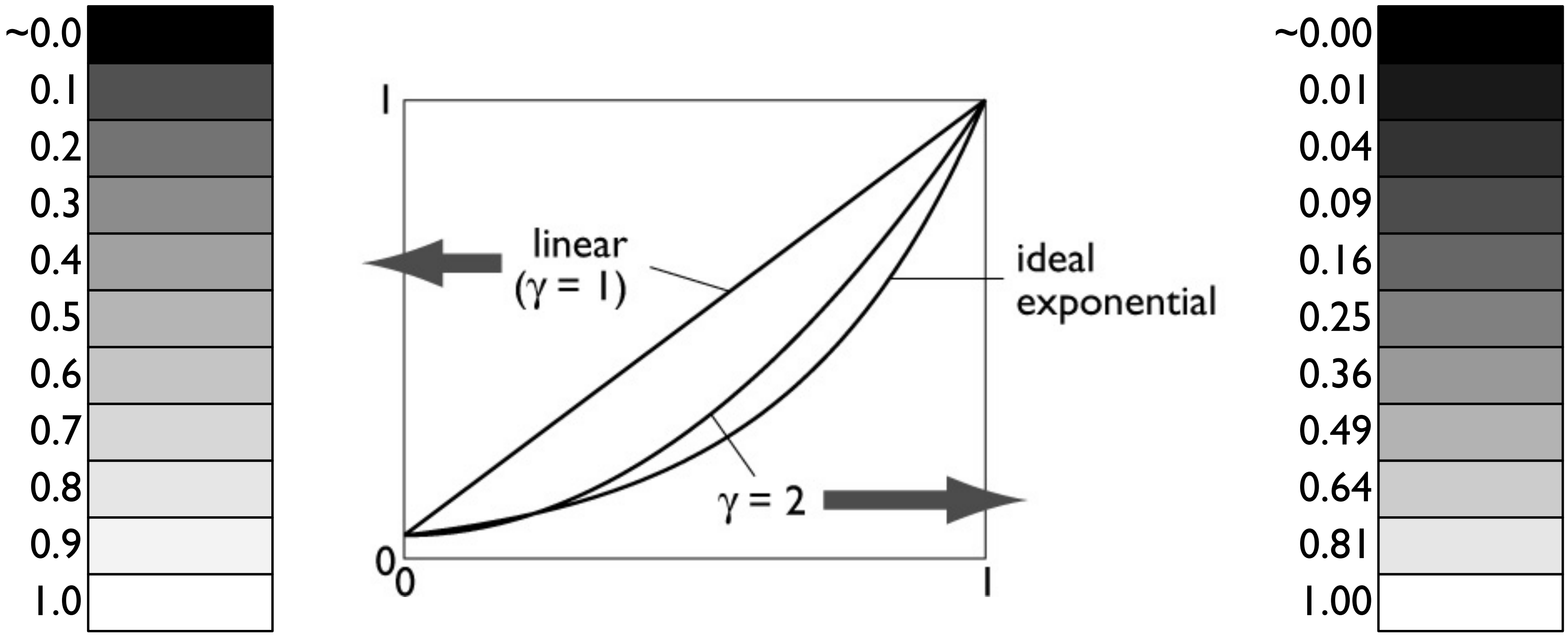
Slide credit: S. Marschner

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

Slide credit: S. Marschner

Gamma quantization



- Close enough to ideal perceptually uniform exponential

Slide credit: S. Marschner

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,

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

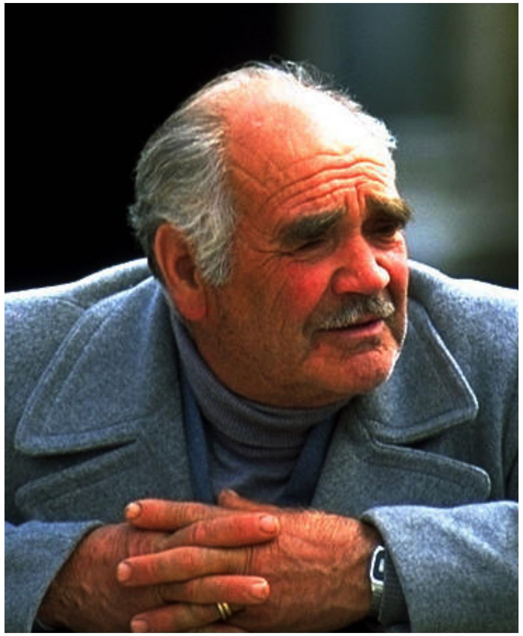
(where $N = 2^n - 1$ in *n* bits). Solving for *n*:

$$n = N a^{\frac{1}{\gamma}}$$

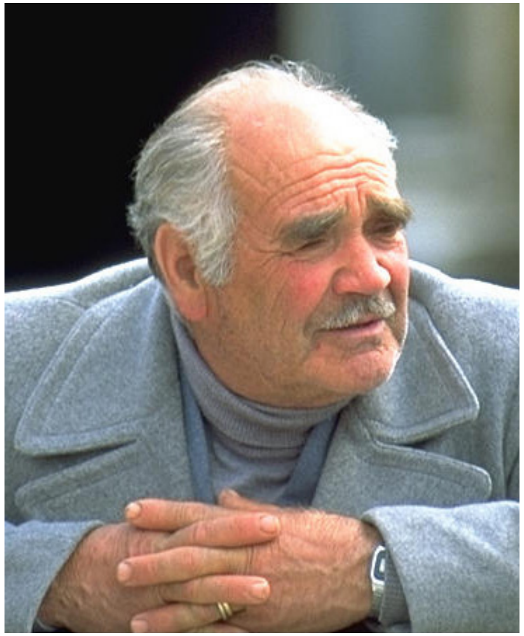
- This is the "gamma correction" recipe that has to be applied when computed values are converted to 8 bits for output
  - failing to do this (implicitly assuming gamma = 1) results in dark, oversaturated images

Slide credit: S. Marschner

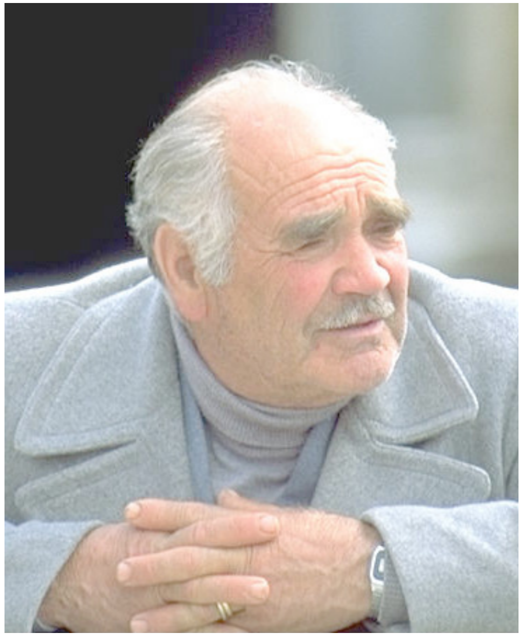
Gamma correction



corrected for
γ lower than
display



OK



corrected for
γ higher than
display

[Philip Greenspun]

Slide credit: S. Marschner

Instagram Filters

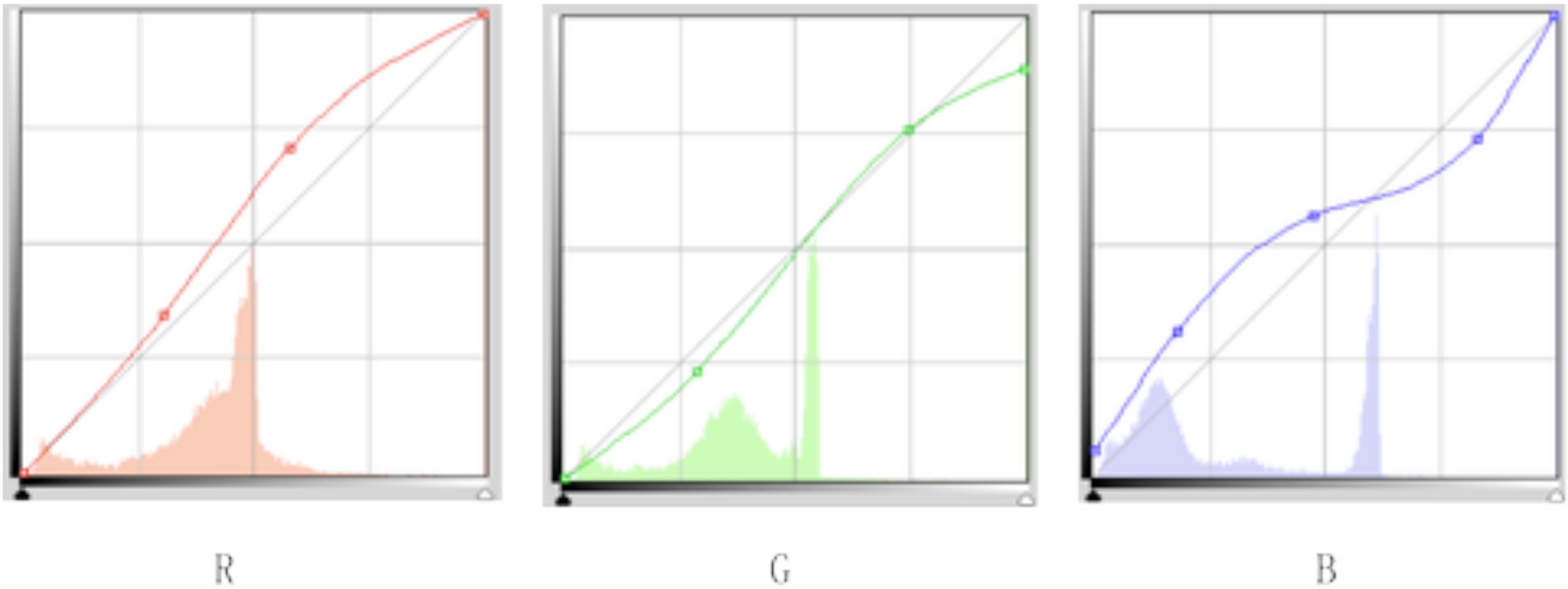
- How do they make those Instagram filters?

"It's really a combination of a bunch of different methods. In some cases we draw on top of images, in others we do pixel math. It really depends on the effect we're going for." --- Kevin Systrom, co-founder of Instagram

Source: C. Dyer

Example Instagram Steps

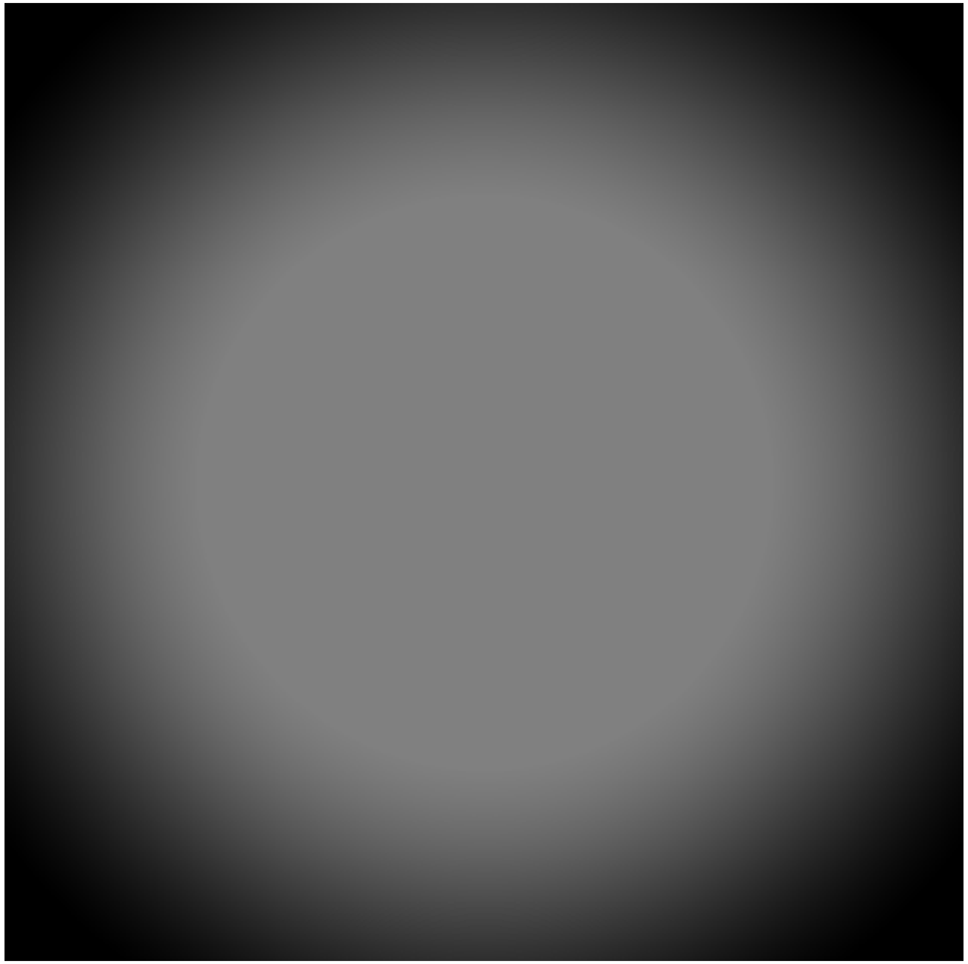
1. Perform an independent RGB color point transformation on the original image to increase contrast or make a color cast



Source: C. Dyer

Example Instagram Steps

2. Overlay a circle background image to create a vignette effect



Source: C. Dyer

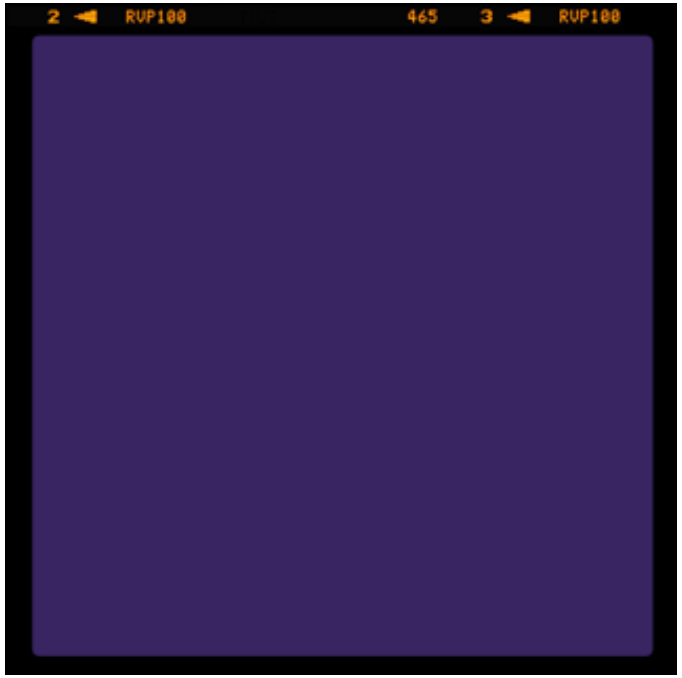
Example Instagram Steps

3. Overlay a background image as decorative grain

Source: C. Dyer

Example Instagram Steps

4. Add a border or frame



Source: C. Dyer

Result

Javascript library for creating Instagram-like effects, see:
http://alexmic.net/filtrr/

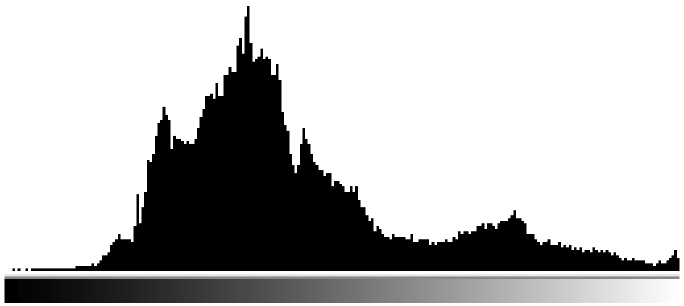
Source: C. Dyer

Today's topics

- Point operations
- Histogram processing

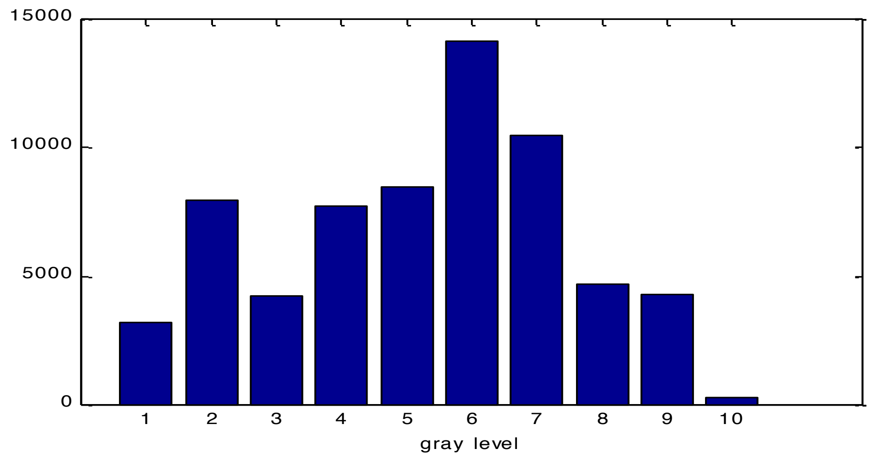
Histogram

- Histogram: a discrete function *h(r)* which counts the number of pixels in the image having intensity *r*
- If *h(r)* is normalized, it measures the probability of occurrence of intensity level *r* in an image

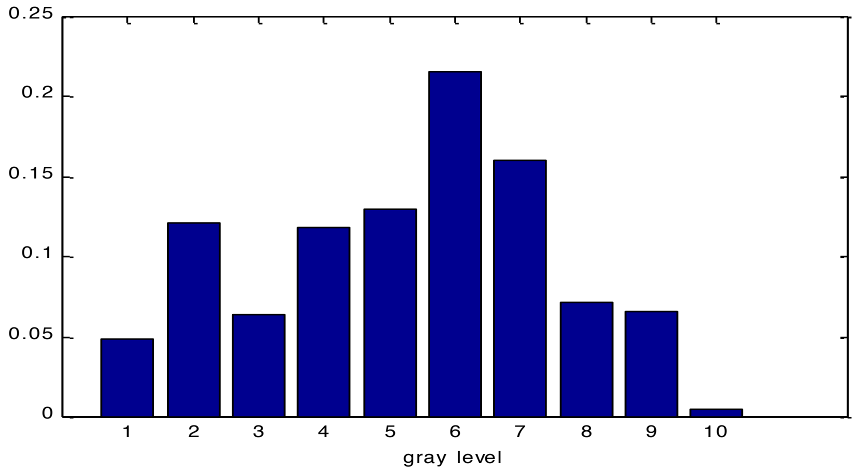


- What histograms say about images?
- What they don't?
  - No spatial information

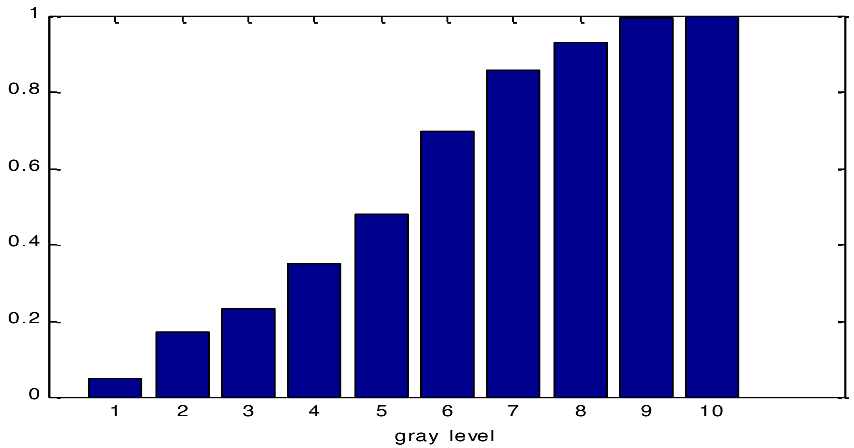
A descriptor for visual information



Histogram



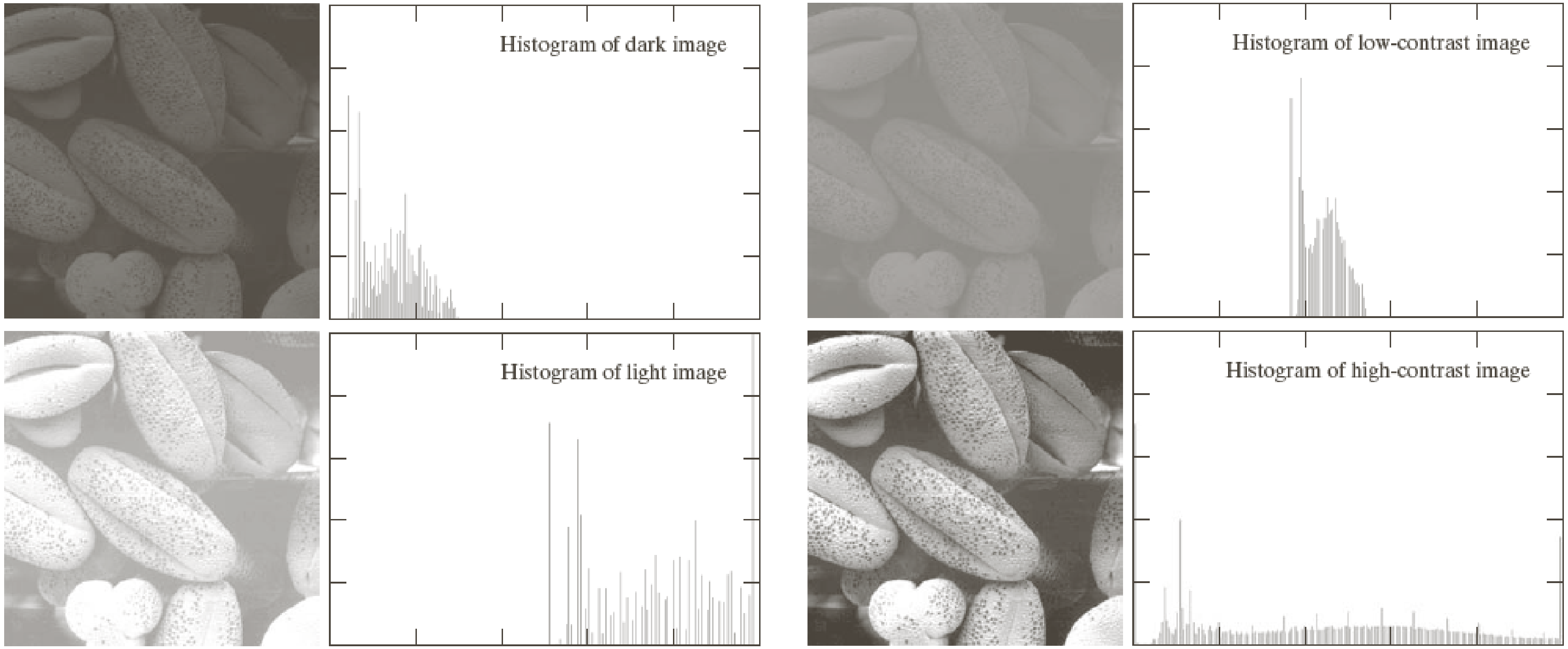
Normalized Histogram



Cumulative Histogram

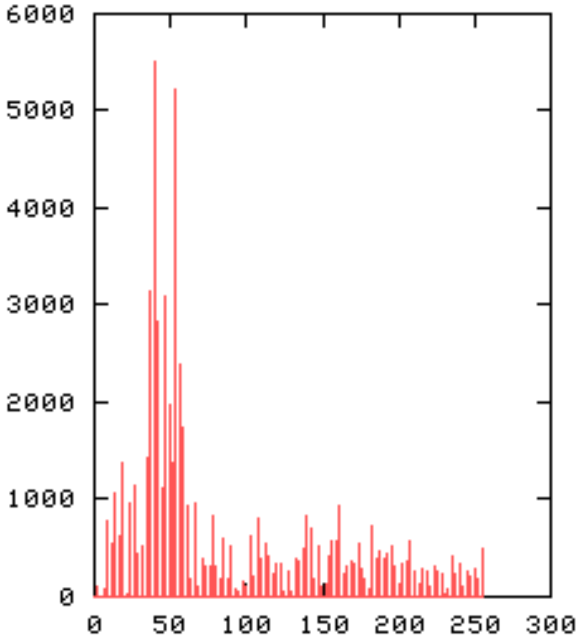
Slide credit: Y. Hel-Or

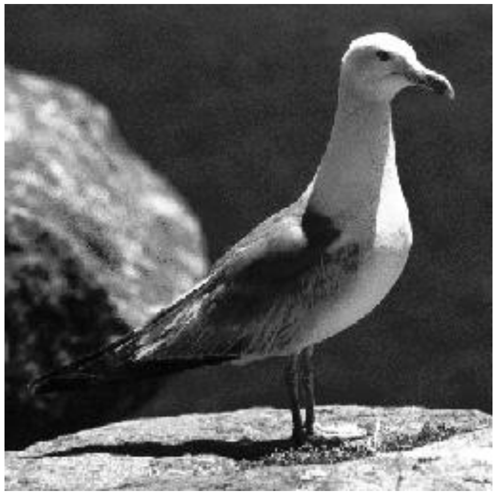
Images and histograms



- How do histograms change when
  - we adjust brightnesss? shifts the histogram horizontally
  - we adjust constrast? stretches or shrinks the histogram horizontally

Image Representations: Histograms





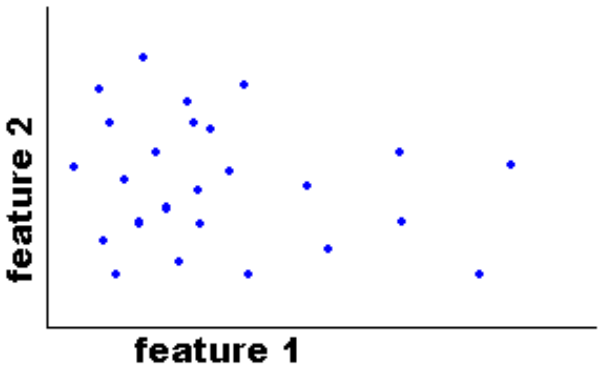
Global histogram

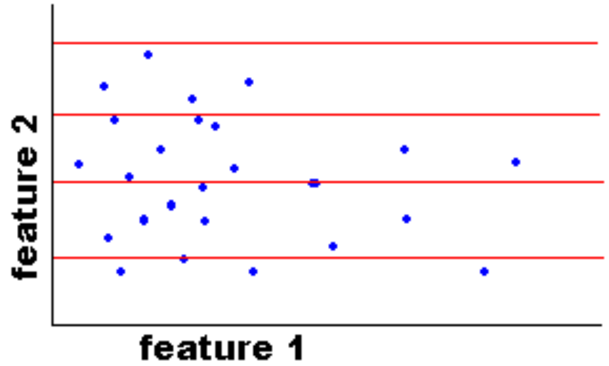
- Represent distribution of features
  - – Color, texture, depth, …

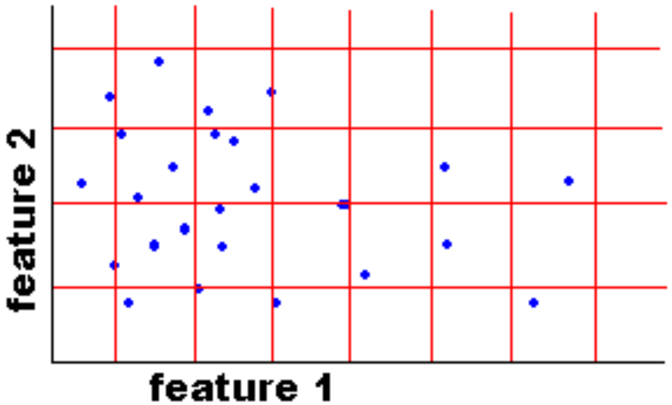
Image credit: D. Kauchak

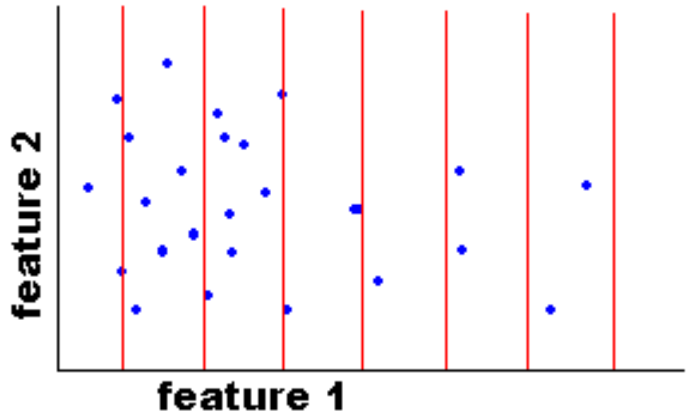
Image Representations: Histograms

Histogram: Probability or count of data in each bin









Joint histogram

- Requires lots of data
- Loss of resolution to avoid empty bins

Marginal histogram

- Requires independent features
- More data/bin than joint histogram

Image credit: D. Kauchak

Histograms: Implementation issues

- Quantization
  - Grids: fast but applicable only with few dimensions

Few Bins
Need less data
Coarser representation

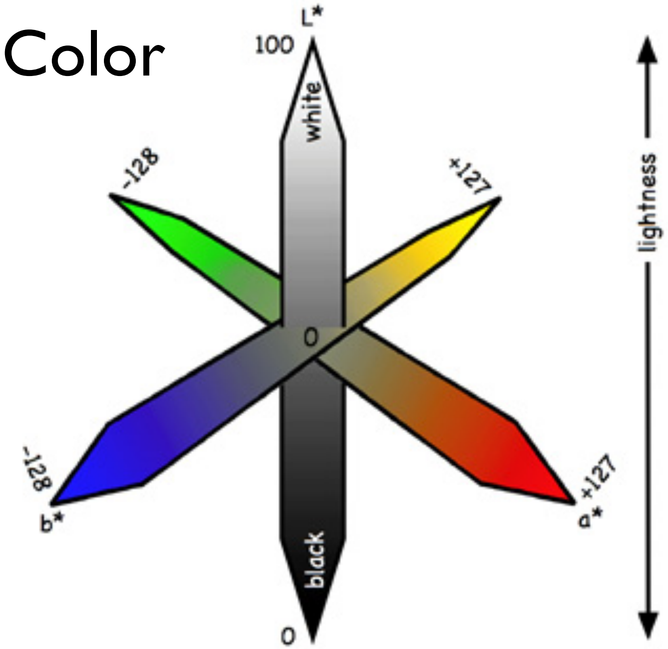
Many Bins
Need more data
Finer representation

- Matching
  - Histogram intersection or Euclidean may be faster
  - Chi-squared often works better
  - Earth mover's distance is good for when nearby bins represent similar values

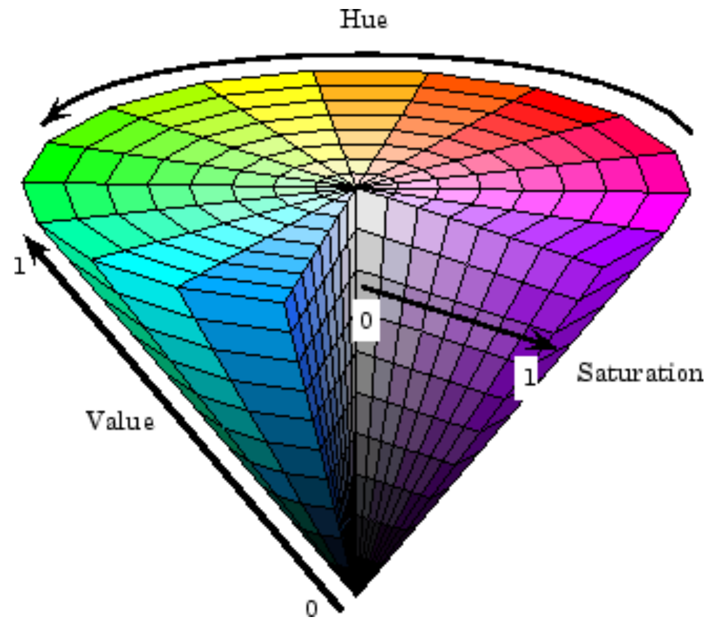
Slide credit: J. Hays

What kind of things do we compute histograms of?

- Color



L*a*b* color space



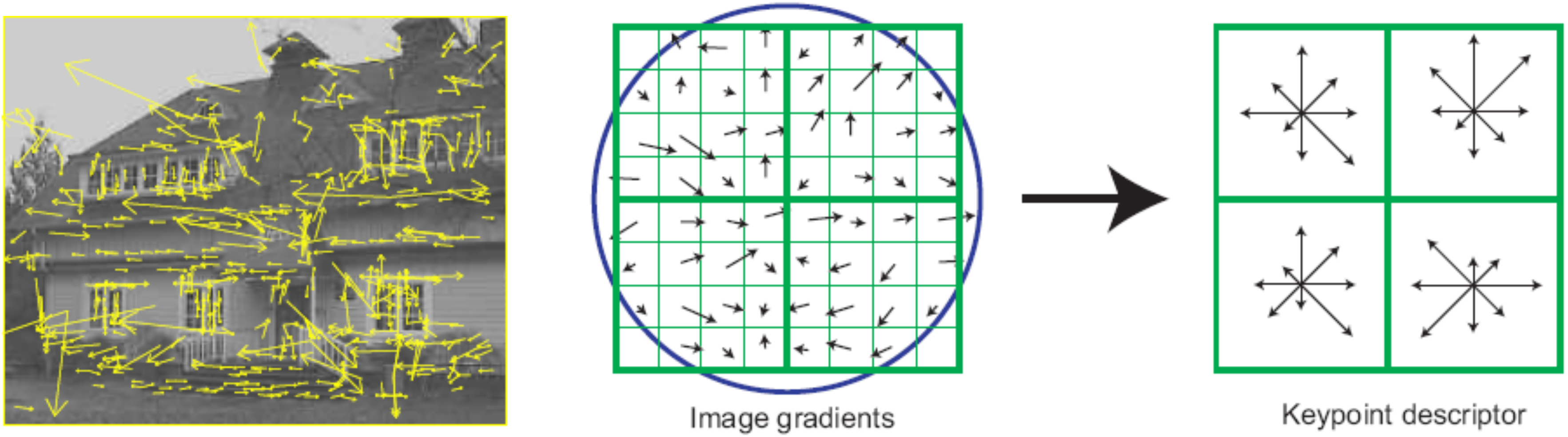
HSV color space

- Texture (filter banks over regions – later on)

Slide credit: J. Hays

What kind of things do we compute histograms of?

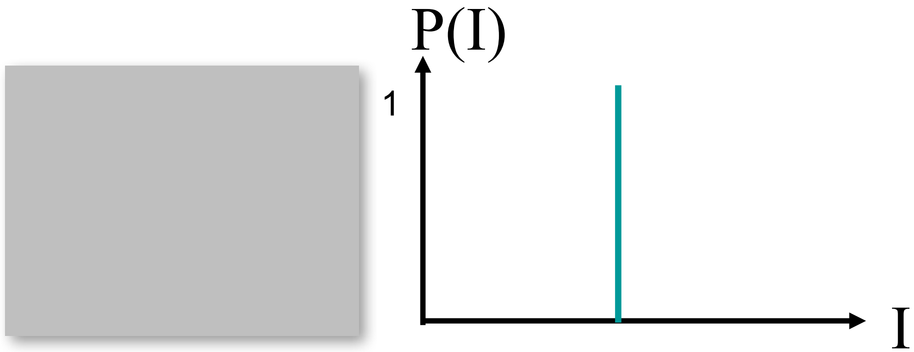
- Histograms of oriented gradients (later on)

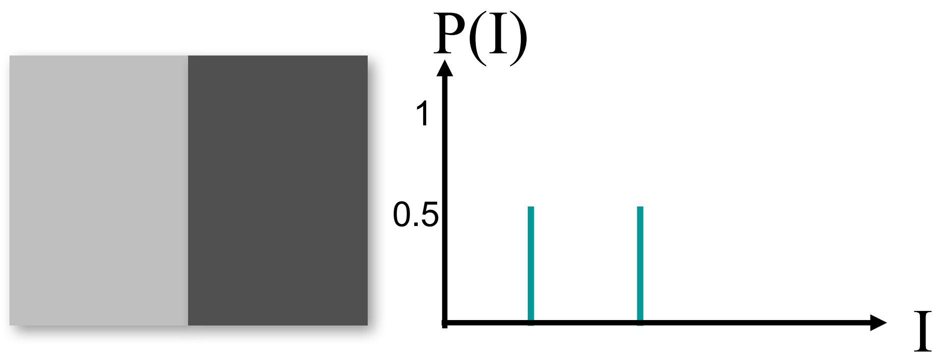


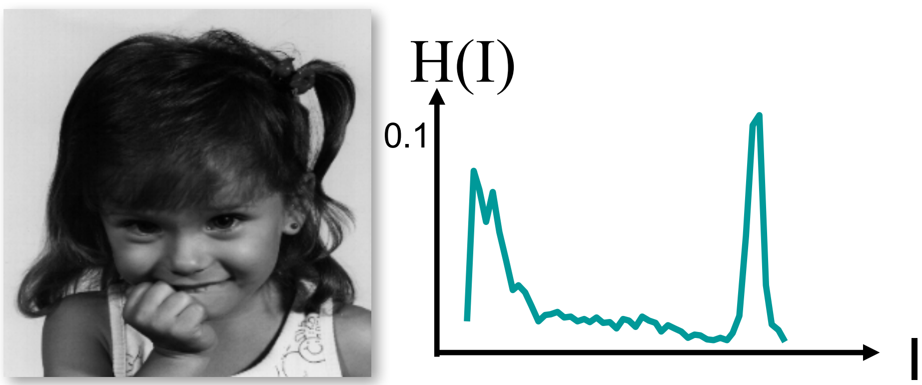
SIFT – Lowe IJCV 2004

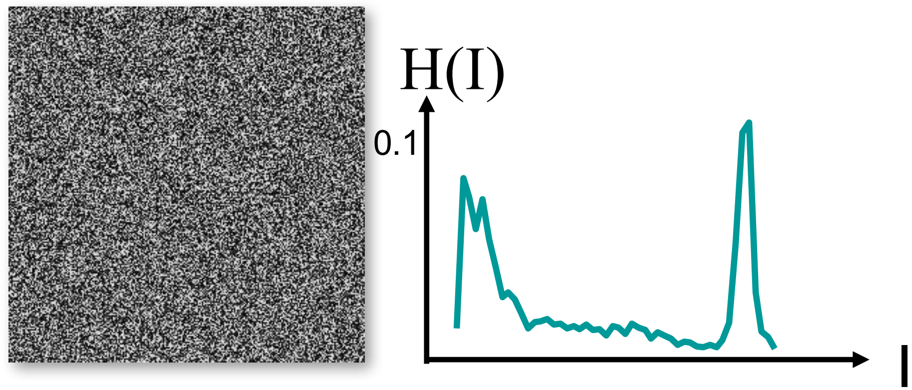
Slide credit: J. Hays

Examples









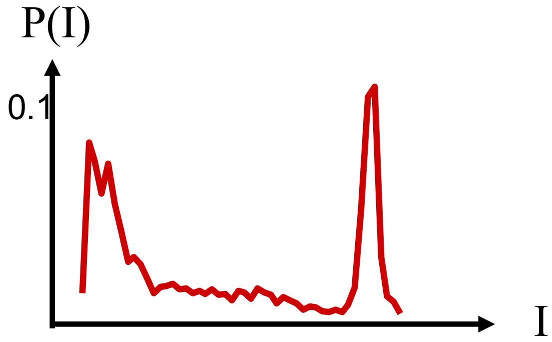
Pixel permutation of the left image

- The image histogram does not fully represent the image

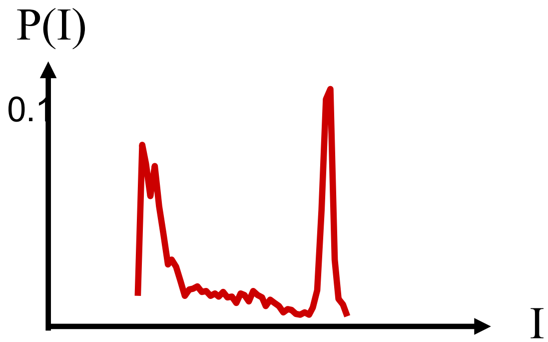
Slide credit: Y. Hel-Or

Examples

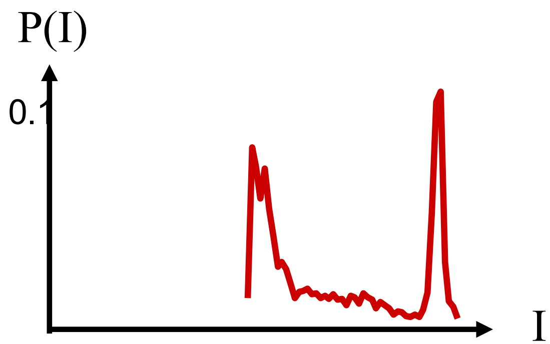
Original image



Decreasing contrast



Increasing average



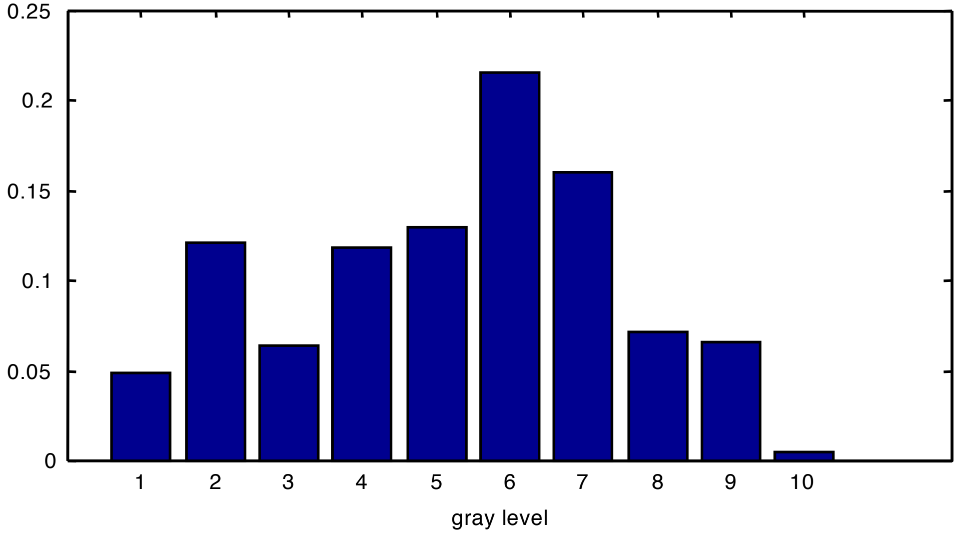
Slide credit: Y. Hel-Or

Image Statistics

- The image mean: $E\{I\} = \frac{1}{N}\sum_{i,j} I(i,j) = \frac{1}{N}\sum_{k} k\,H(k) = \sum_{k} k\,P(k)$

- Generally: $E\{g(k)\} = \sum_{k} g(k)\,P(k)$

- The image s.t.d. : $\sigma(I) = \sqrt{E\{(I - E\{I\})^2\}} = \sqrt{E(I^2) - E^2(I)}$



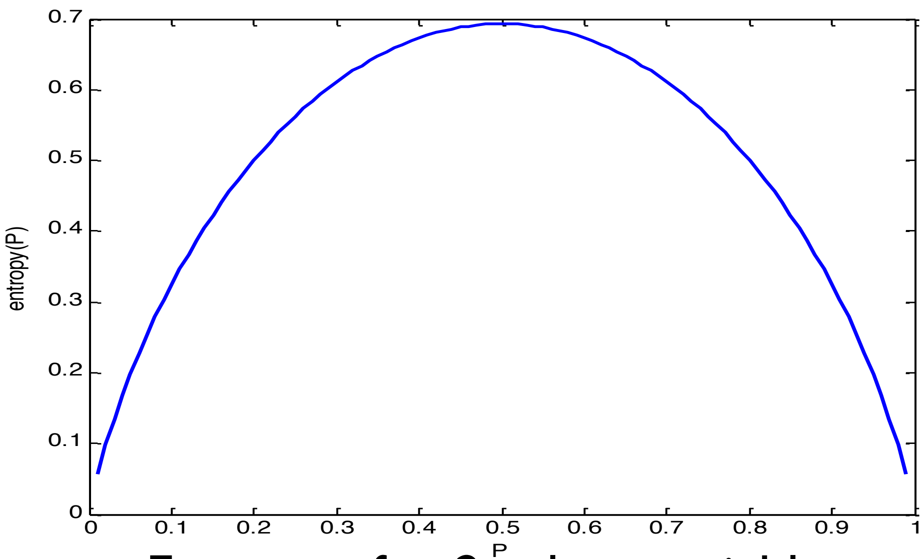
where $E\{I^2\} = \sum_{k} k^2 P(k)$

Slide credit: Y. Hel-Or

Image Entropy

$$Entropy(I) = -\sum_{k} P(k) \log P(k)$$

- The image entropy specifies the uncertainty in the image values.
- Measures the averaged amount of information required to encode the image values.

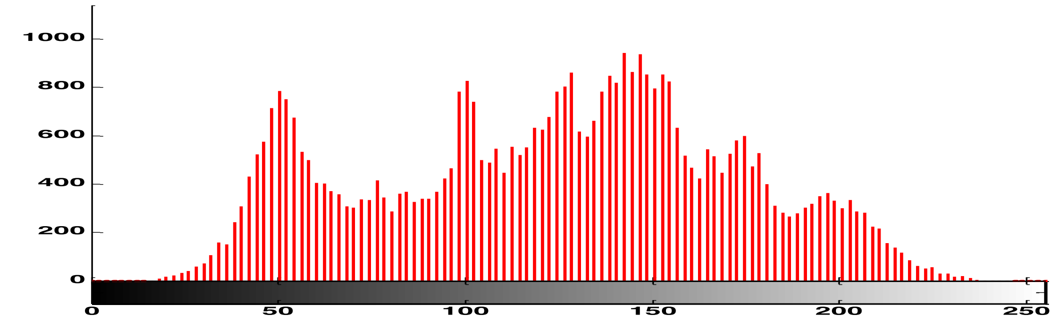


Entropy of a 2 values variable

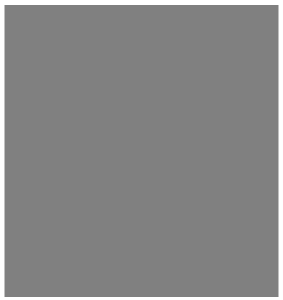
Slide credit: Y. Hel-Or

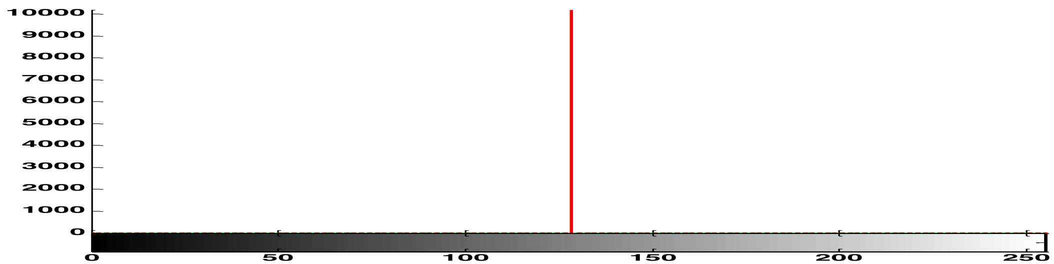
Image Entropy

- An infrequent event provides more information than a frequent event
- Entropy is a measure of histogram dispersion



entropy=7.4635





entropy=0

Slide credit: Y. Hel-Or

Adaptive Histogram

- In many cases histograms are needed for local areas in an image
- Examples:
  - Pattern detection
  - adaptive enhancement
  - adaptive thresholding
  - tracking

Slide credit: Y. Hel-Or

Histogram Usage

- Digitizing parameters
- Measuring image properties:
  - Average
  - Variance
  - Entropy
  - Contrast
  - Area (for a given gray-level range)
- Threshold selection
- Image distance
- Image Enhancement
  - Histogram equalization
  - Histogram stretching
  - Histogram matching

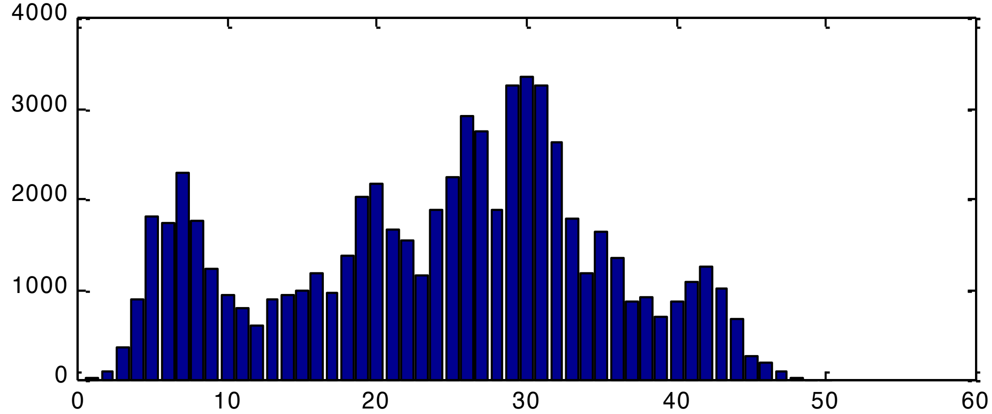
Slide credit: Y. Hel-Or

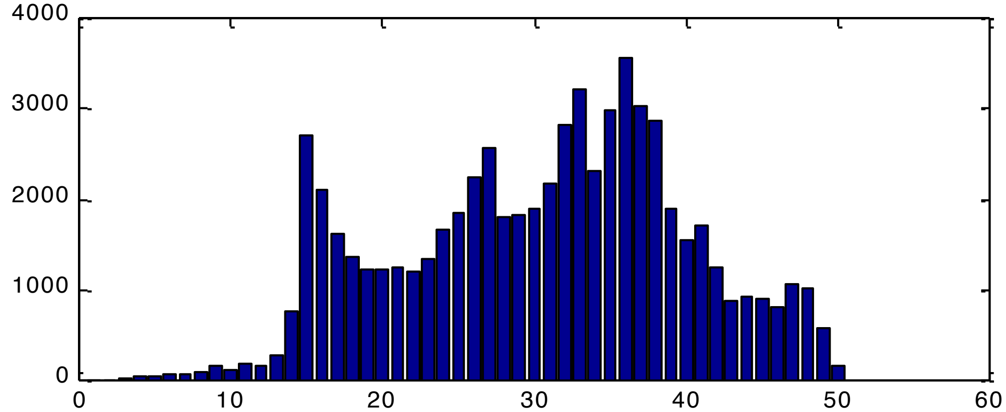
Example: Auto-Focus

- In some optical equipment (e.g. slide projectors) inappropriate lens position creates a blurred ("out-of-focus") image
- We would like to automatically adjust the lens
- How can we measure the amount of blurring?

Slide credit: Y. Hel-Or

Example: Auto-Focus

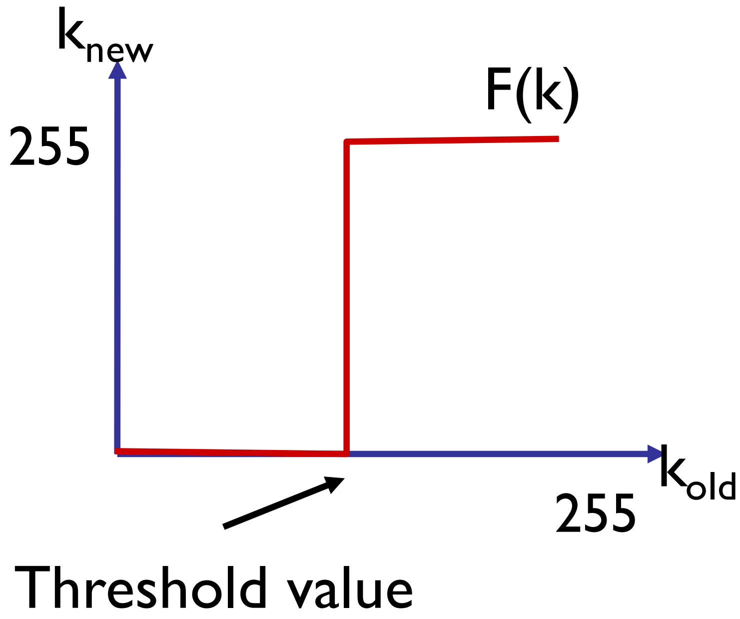


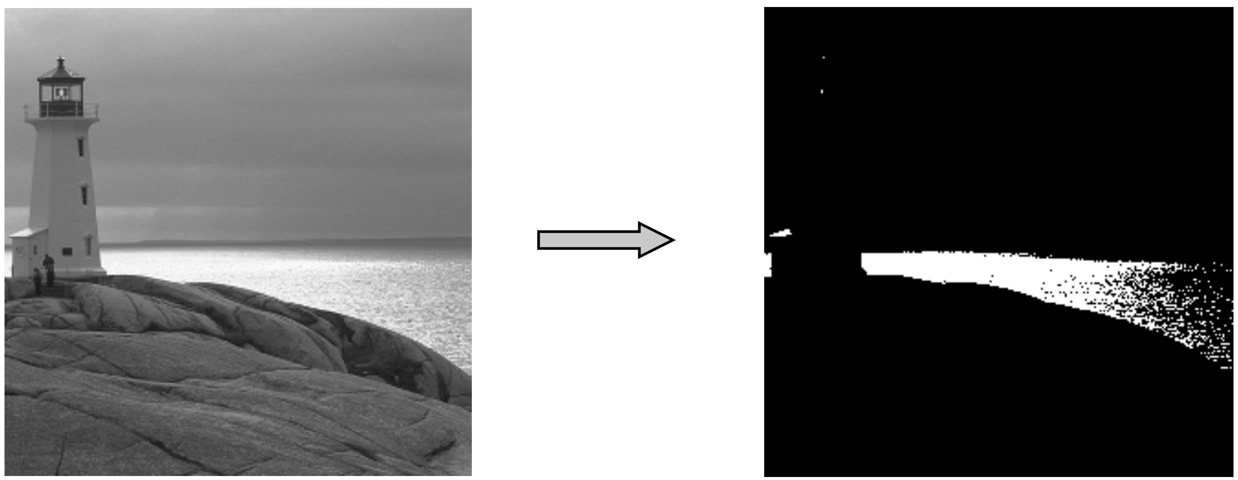


- Image mean is not affected by blurring
- Image s.t.d. (entropy) is decreased by blurring
- <u>Algorithm</u>: Adjust lens according the changes in the histogram s.t.d.

Slide credit: Y. Hel-Or

Recall: Thresholding



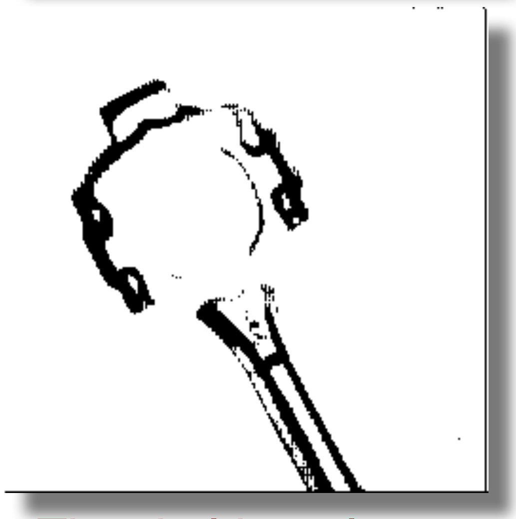


Slide credit: Y. Hel-Or

Threshold Selection

Original Image

Binary Image



Threshold too low

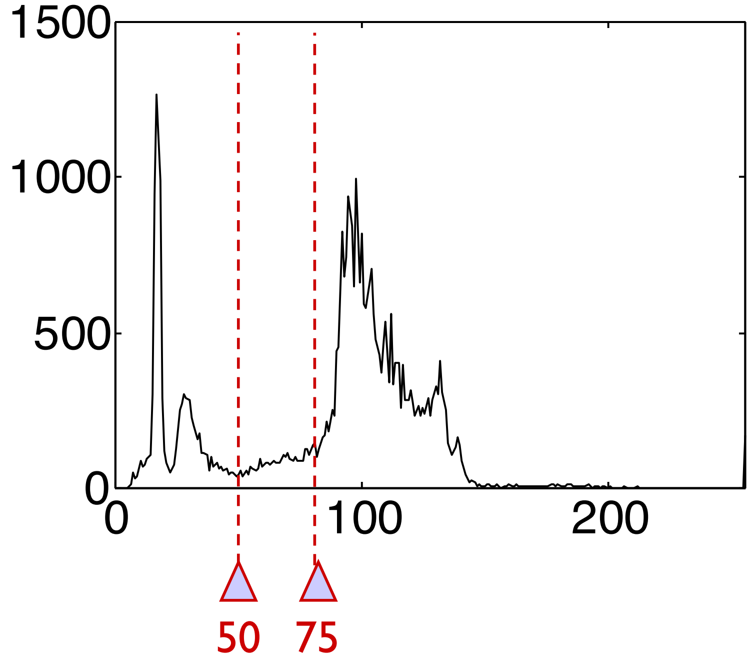
Threshold too high

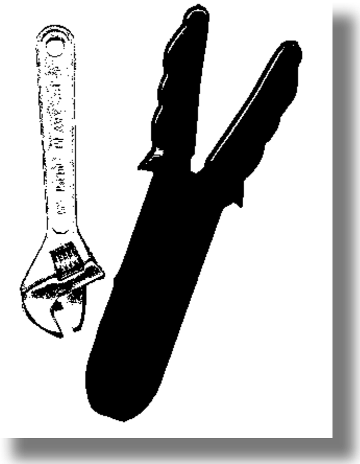
Slide credit: Y. Hel-Or

Segmentation using Thresholding

Original

Histogram





Threshold = 50

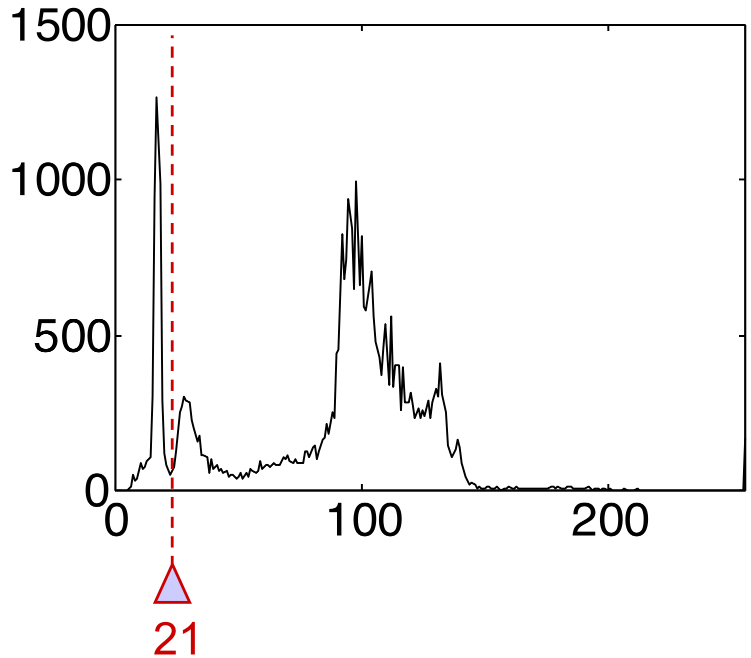
Threshold = 75

Slide credit: Y. Hel-Or

Segmentation using Thresholding

Original

Histogram

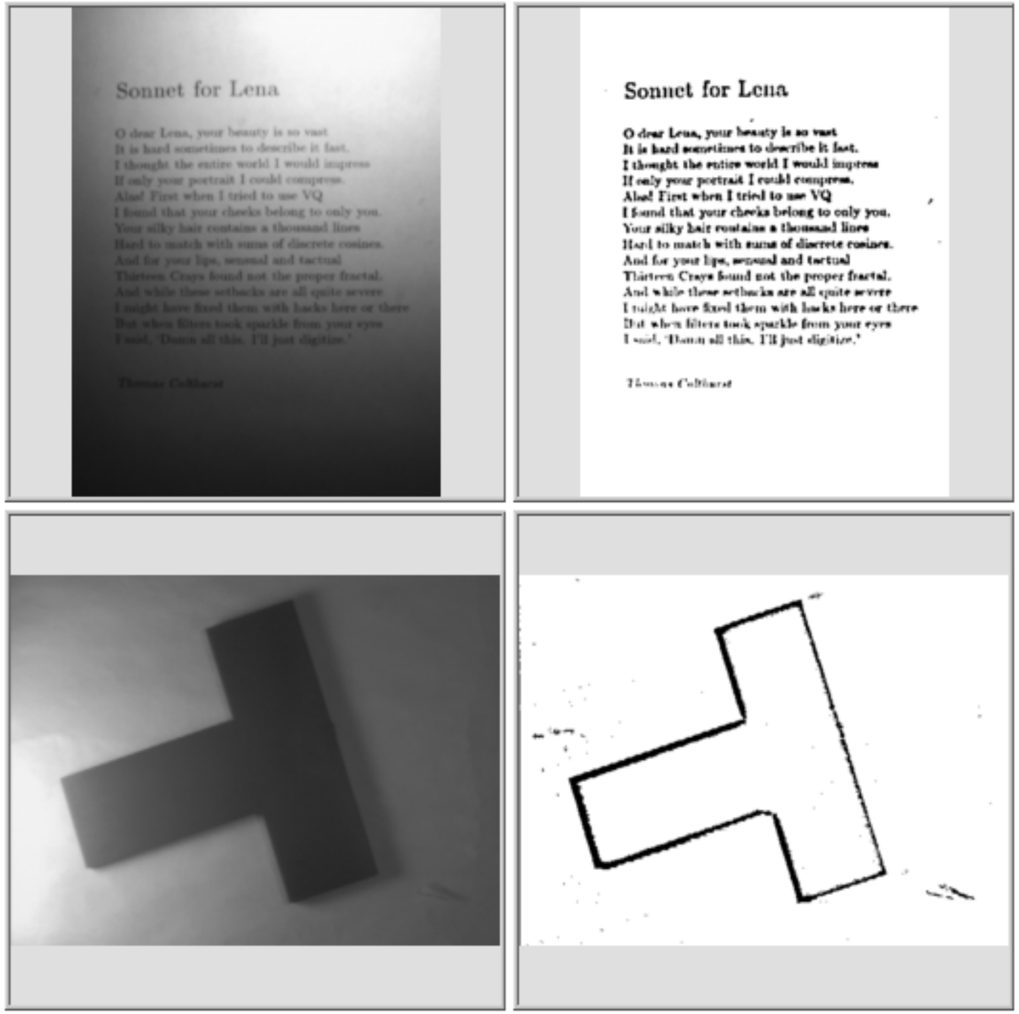


Threshold = 21

Slide credit: Y. Hel-Or

Adaptive Thresholding

- Thresholding is space variant.
- How can we choose the the local threshold values?



Slide credit: Y. Hel-Or

Histogram based image distance

- **Problem**: Given two images A and B whose (normalized) histogram are P_A and P_B define the distance D(A,B) between the images.

- Example Usage:
  – Tracking
  – Image retrieval
  – Registration
  – Detection
  – Many more …

input *target* *similarity*

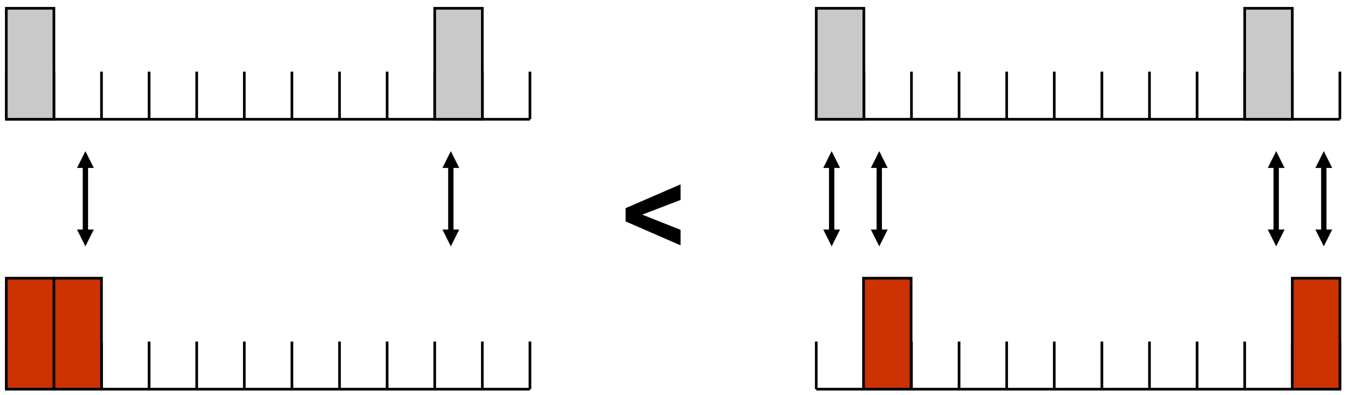
Porikli 05

Slide credit: Y. Hel-Or

Option 1: Minkowski Distance

$$D_p(A,B) = \left[\sum_k \left| P_A(k) - P_B(k) \right|^p \right]^{1/p}$$

- **Problem**: distance may not reflects the perceived dissimilarity:



Slide credit: Y. Hel-Or

Option 2: Kullback-Leibler (KL) Distance

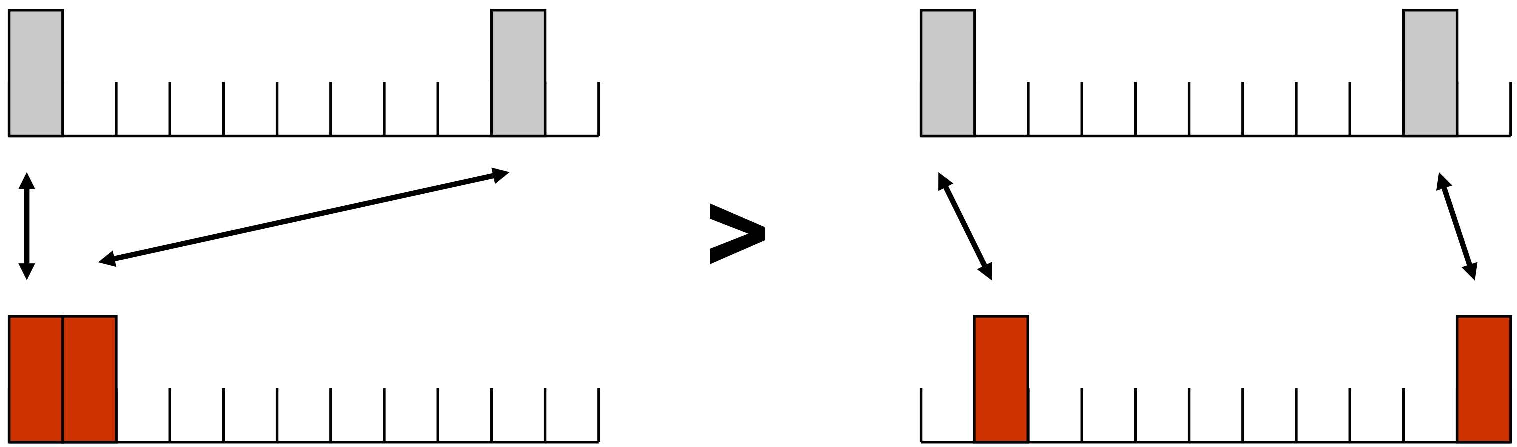
$$D_{KL}(A \parallel B) = -\sum_k P_A(k) \log \frac{P_A(k)}{P_B(k)}$$

- Measures the amount of added information needed to encode image A based on the histogram of image B.
- Non-symmetric: $D_{KL}(A,B) \neq D_{KL}(B,A)$
- Suffers from the same drawback of the Minkowski distance.

Slide credit: Y. Hel-Or

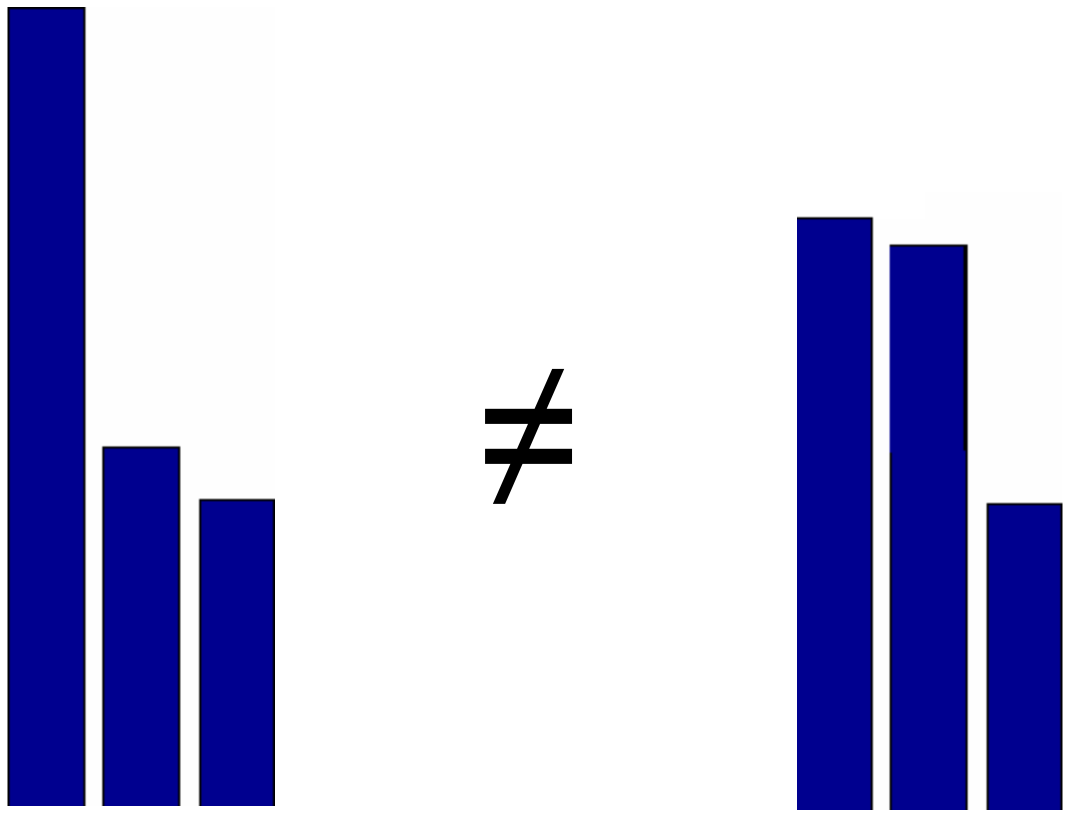
Option 3: The Earth Mover Distance (EMD)

- Suggested by Rubner & Tomasi 98
- Defines as the minimum amount of "work" needed to transform histogram H_A towards H_B
- The term d_{ij} defines the "ground distance" between gray-levels i and j.
- The term $F=\{f_{ij}\}$ is an admissible flow from $H_A(i)$ to $H_B(j)$



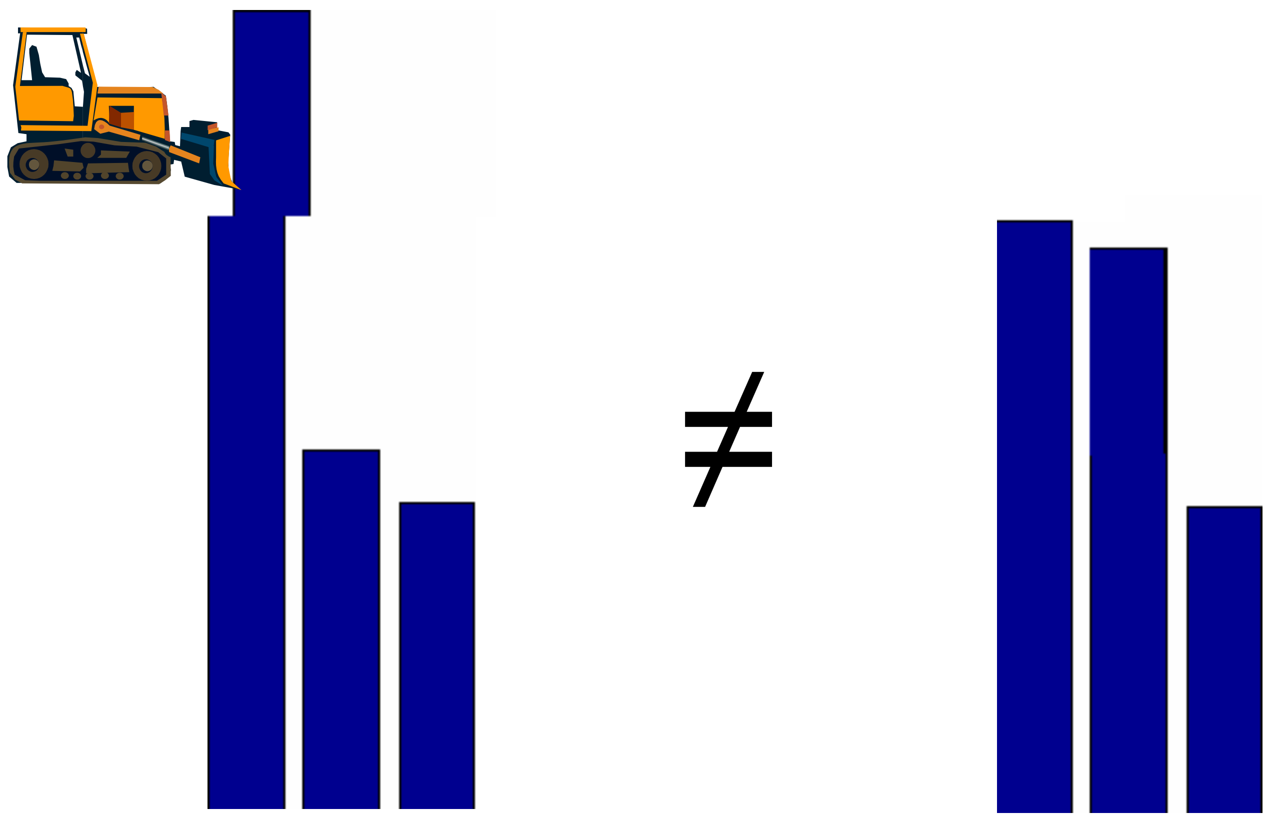
Slide credit: Y. Hel-Or

Option 3: The Earth Mover Distance (EMD)



Slide credit: P. Barnum

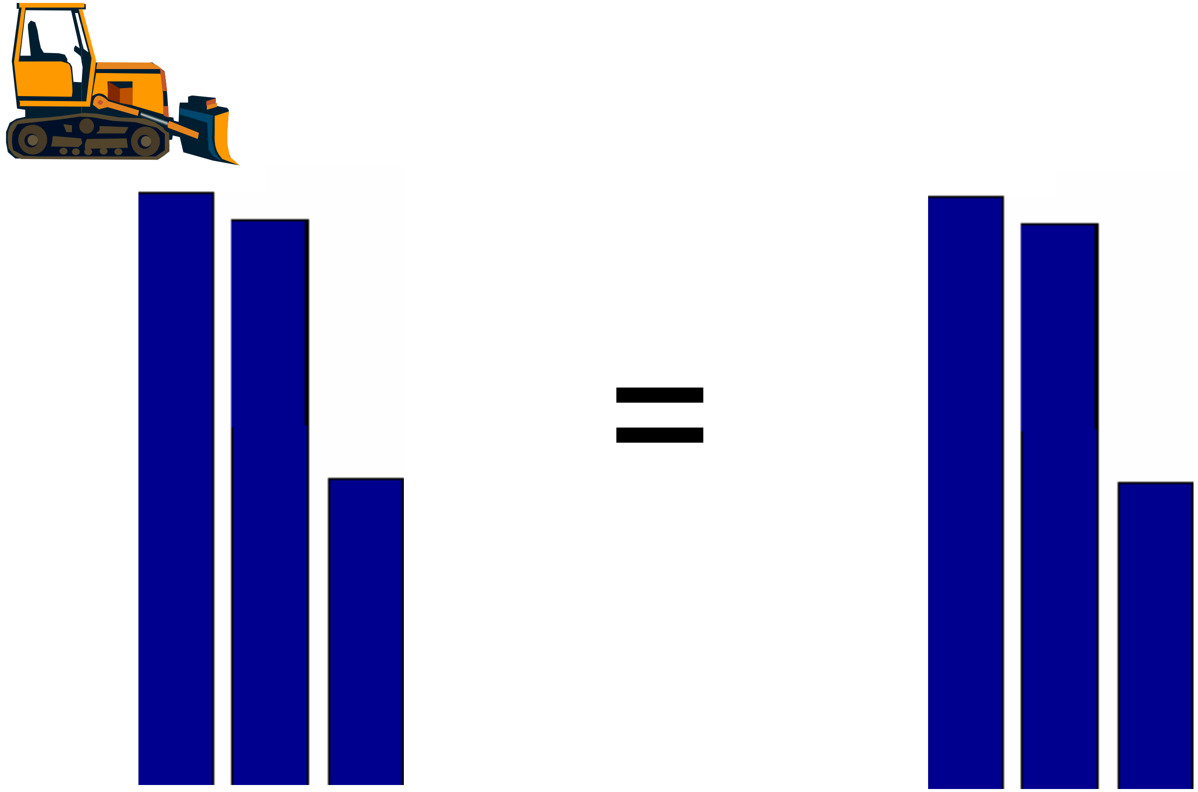
Option 3: The Earth Mover Distance (EMD)



From: Pete Barnum

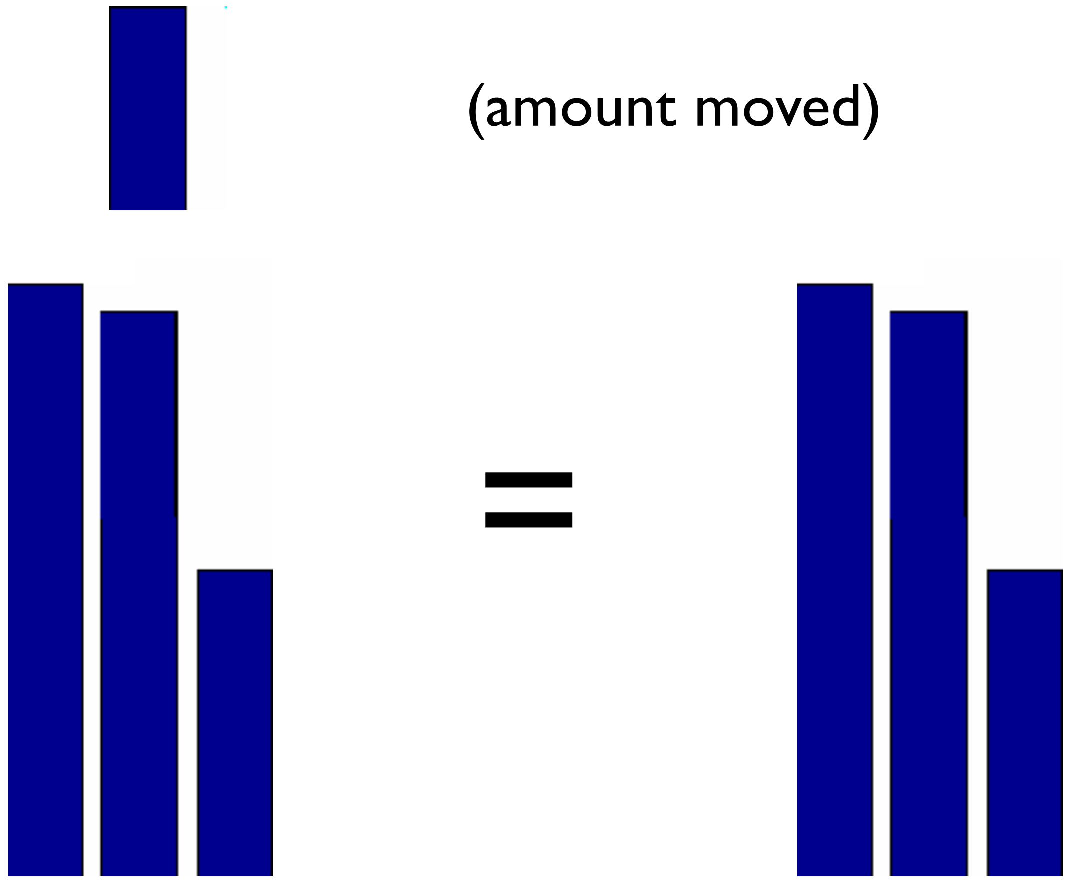
Slide credit: P. Barnum

Option 3: The Earth Mover Distance (EMD)



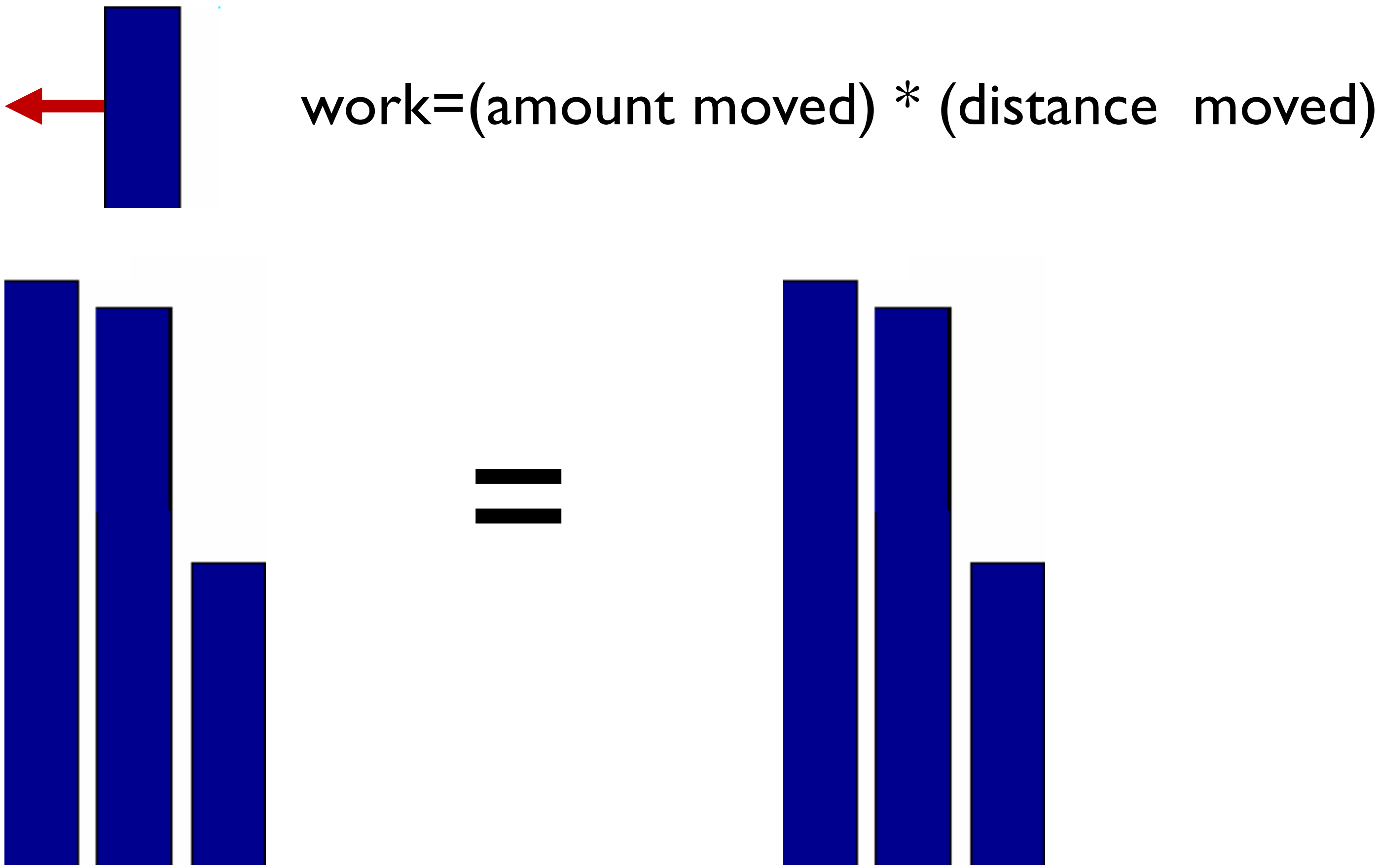
Slide credit: P. Barnum

Option 3: The Earth Mover Distance (EMD)



Slide credit: P. Barnum

Option 3: The Earth Mover Distance (EMD)



Slide credit: P. Barnum

Option 3: The Earth Mover Distance (EMD)

$$D_{EMD}(A,B) = \min_{F} \sum_{i} \sum_{j} f_{ij} \cdot d_{ij}$$

$$s.t. \quad f_{ij} \geq 0 \; ; \quad P_B(k) = \sum_{i} f_{ik} \quad ; \quad P_A(k) \geq \sum_{i} f_{ki}$$

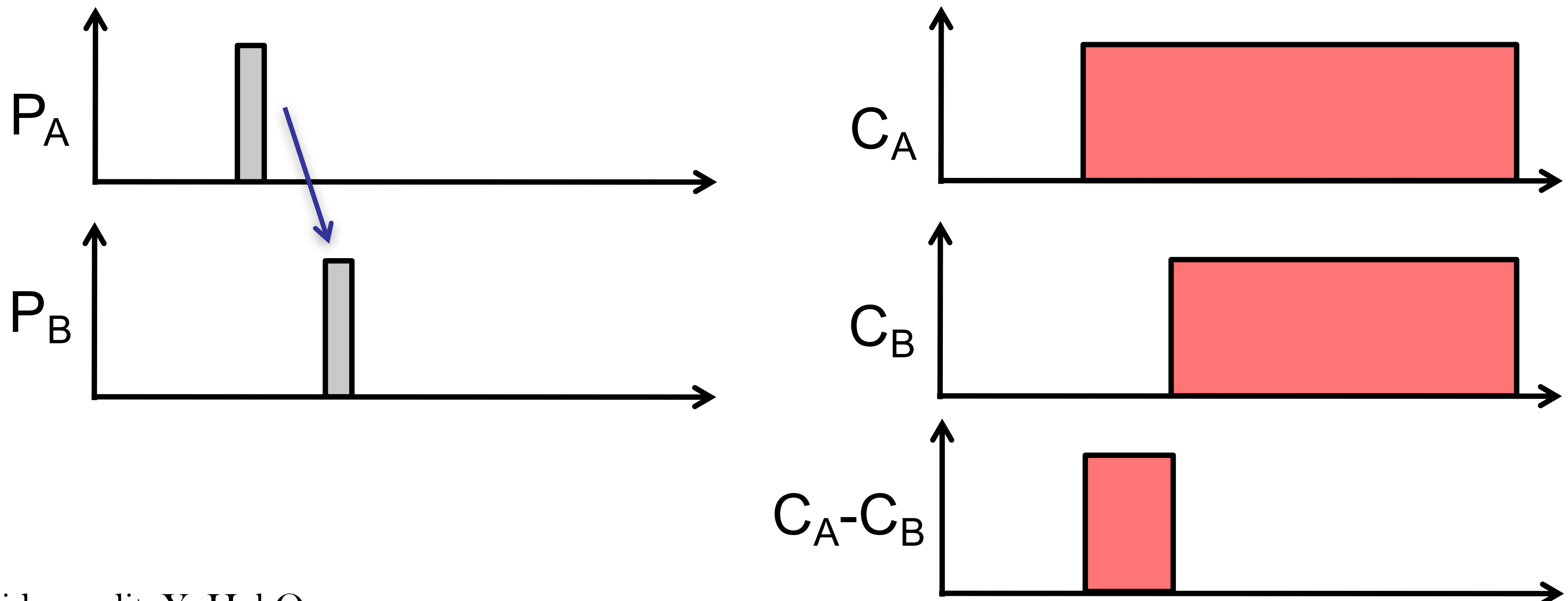
- Constraints:
  - Move earth only from A to B
  - After move P_A will be equal to P_B
  - Cannot send more "earth" than there is
- Can be solved using Linear Programming
- Can be applied in high dim. histograms (color).

Slide credit: Y. Hel-Or

Special case: EMD in 1D

- Define C_A and C_B as the cumulative histograms of image A and B respectively:

$$D_{EMD}(A,B) = \sum_k \left| C_A(k) - C_B(k) \right|$$



Slide credit: Y. Hel-Or

Histogram equalization

- A good quality image has a nearly uniform distribution of intensity levels. Why?
- Every intensity level is equally likely to occur in an image

- *Histogram equalization:* Transform an image so that it has a uniform distribution
  - create a lookup table defining the transformation

Histogram as a probability density function

- Recall that a normalized histogram measures the probability of occurrence of an intensity level *r* in an image
- We can normalize a histogram by dividing the intensity counts by the area

$$p(r) = \frac{h(r)}{Area}$$

Histogram equalization: Continuous domain

- Define a transformation function of the form

$$s = T(r) = (L-1) \underbrace{\int_0^r p(w)\,dw}_{\text{cumulative distribution function}}$$

where

- r is the input intensity level
- s is the output intensity level
- p is the normalized histogram of the input signal
- L is the desired number of intensity levels

(Continuous) output signal has a uniform distribution!

Histogram equalization: Discrete domain

- Define the following transformation function for an MxN image

$$s_k = T(r_k) = (L-1)\sum_{j=0}^{k}\frac{n_j}{MN} = \frac{(L-1)}{MN}\sum_{j=0}^{k} n_j$$

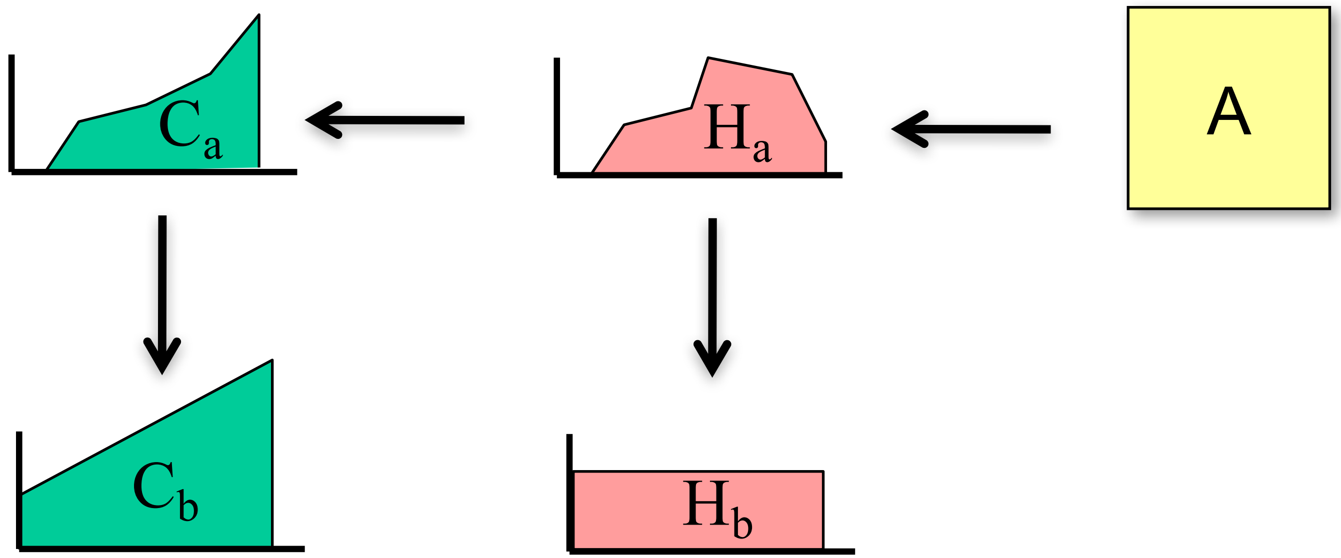
for $k = 0, \ldots, L-1$

where

- r_k is the input intensity level
- s_k is the output intensity level
- n_j is the number of pixels having intensity value j in the input image
- L is the number of intensity levels

(Discrete) output signal has a nearly uniform distribution!

Histogram equalization

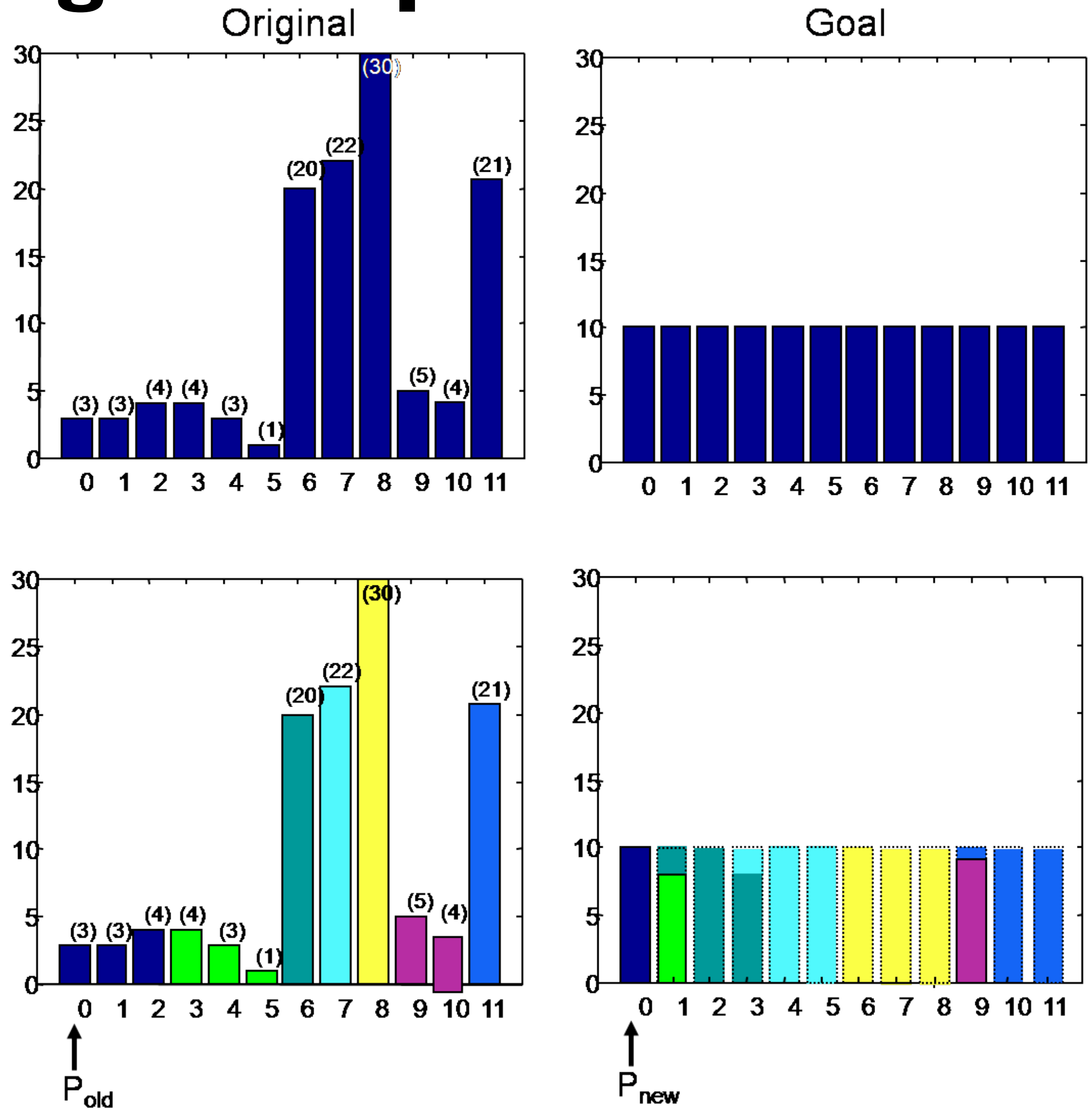


- Define: $C_b(v) = v * \frac{(\#\,pixels)}{\#\,grayValues}$

- Assign: $v_b = C_b^{-1}(C_a(v_a)) = M(v_a)$

Slide credit: Y. Hel-Or

Histogram equalization

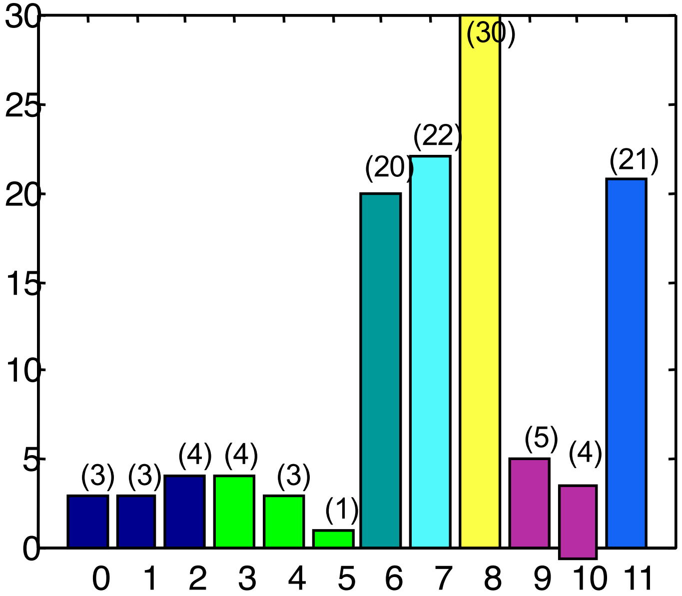


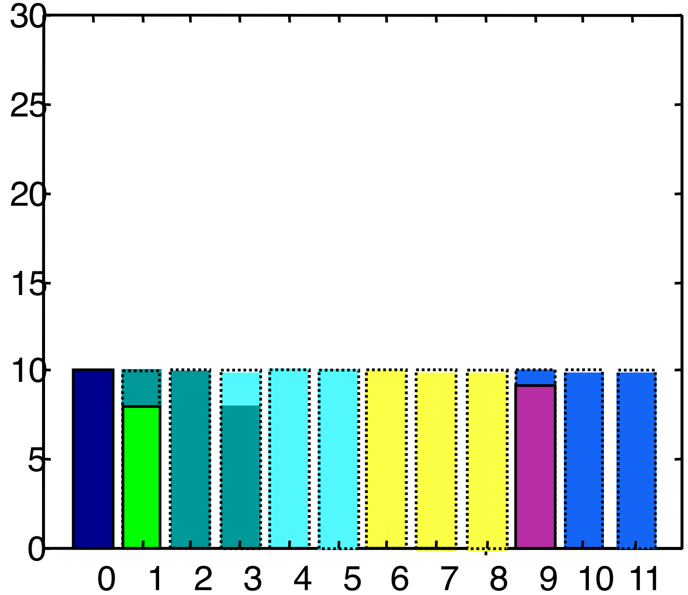
0	1	2	3	4	5	6	7	8	9	10	11	old
0	0	0	1	1	1	3	5	8	9	9	11	new

Slide credit: Y. Hel-Or

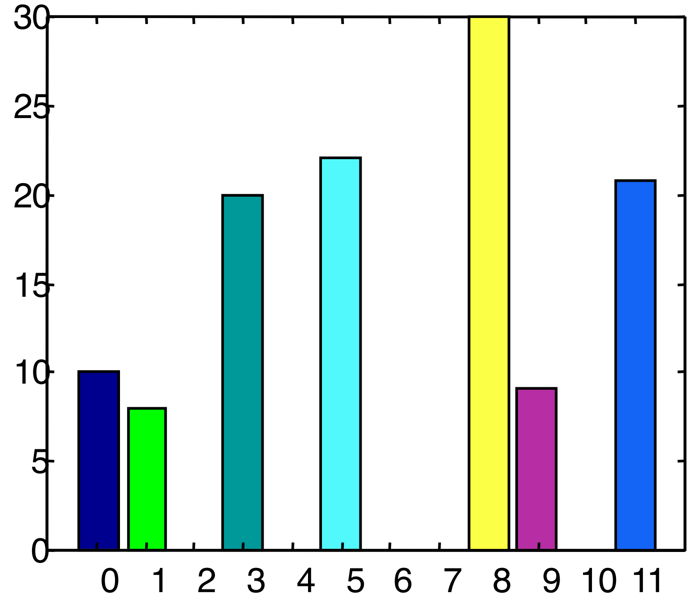
Histogram equalization

Original





Goal

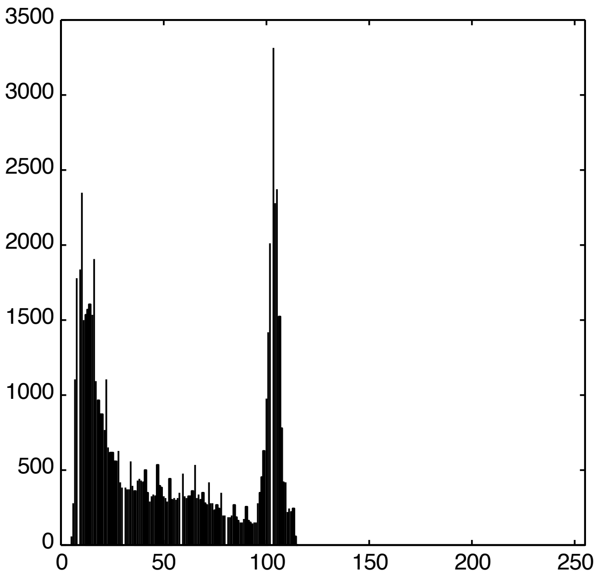


Result

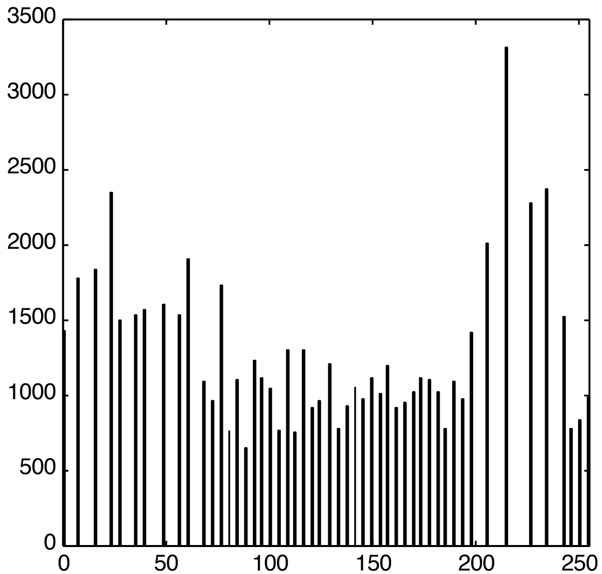
0	1	2	3	4	5	6	7	8	9	10	11	old
0	0	0	1	1	1	3	5	8	9	9	11	new

Slide credit: Y. Hel-Or

Histogram equalization examples



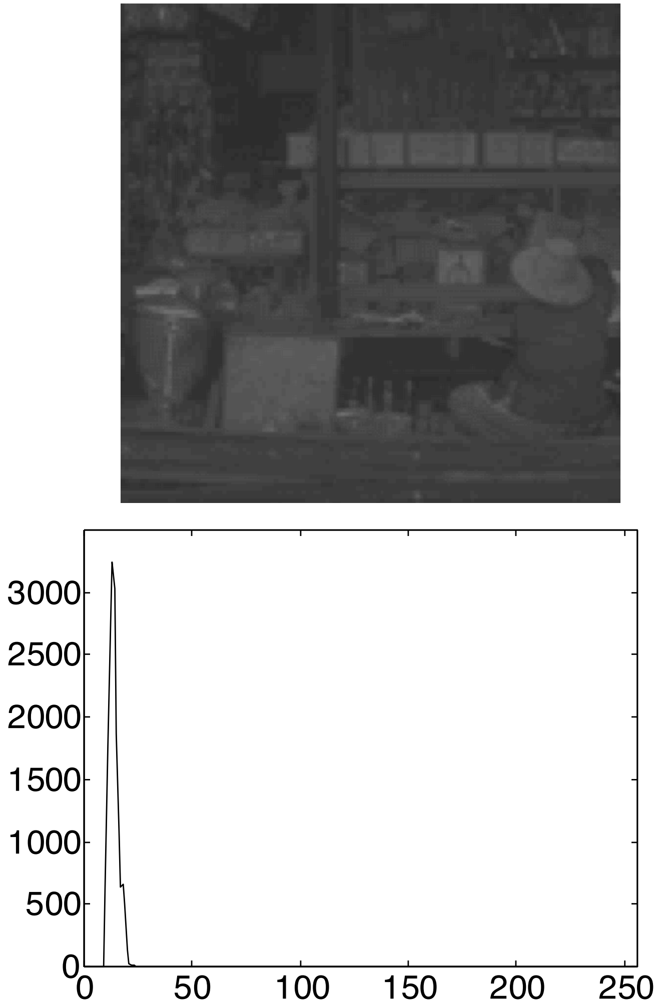
Original



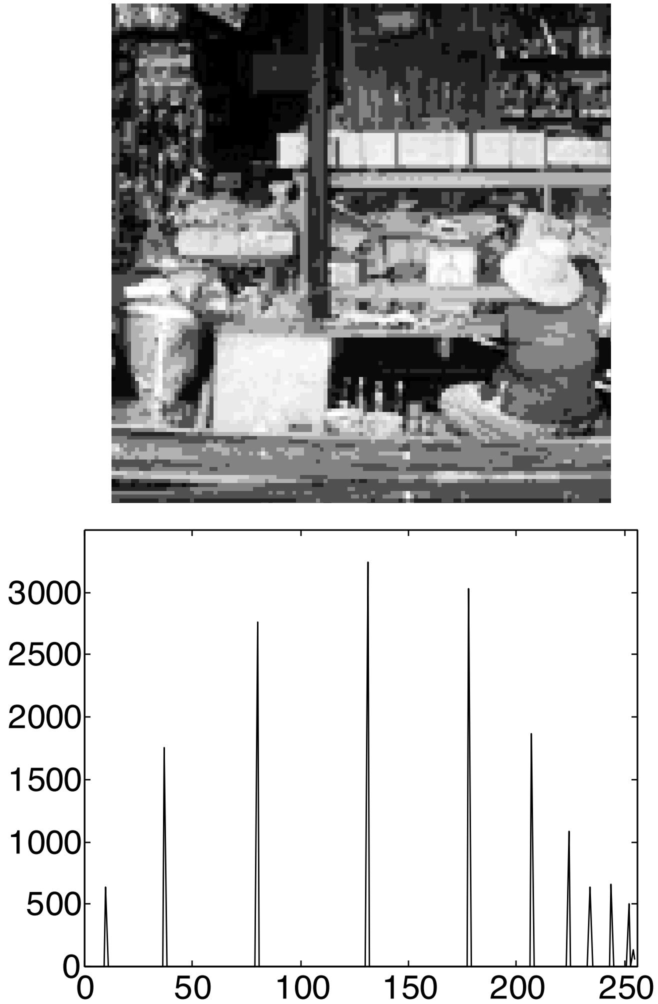
Equalized

Slide credit: Y. Hel-Or

Histogram equalization examples



Original



Equalized

Slide credit: Y. Hel-Or

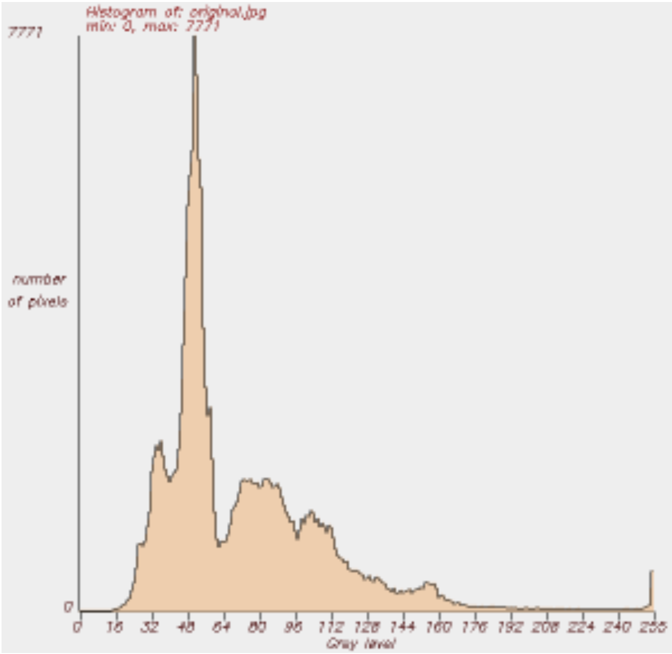
Histogram equalization examples

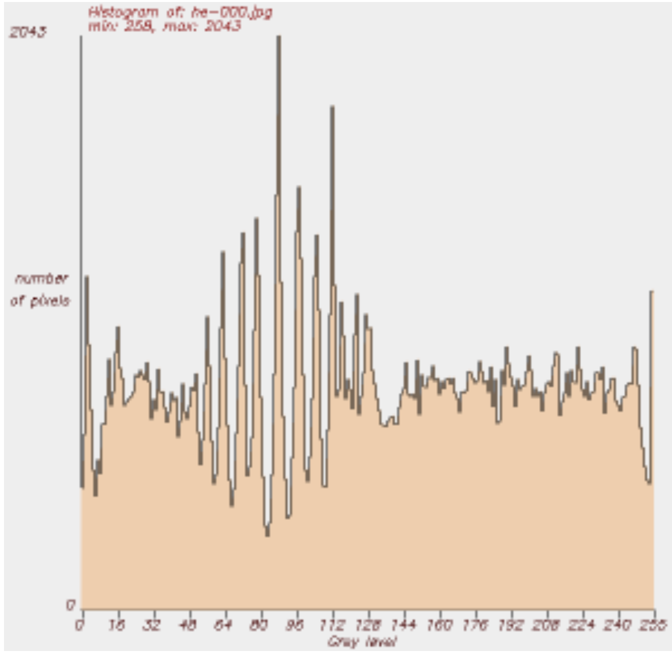
Original

Equalized

Slide credit: Y. Hel-Or

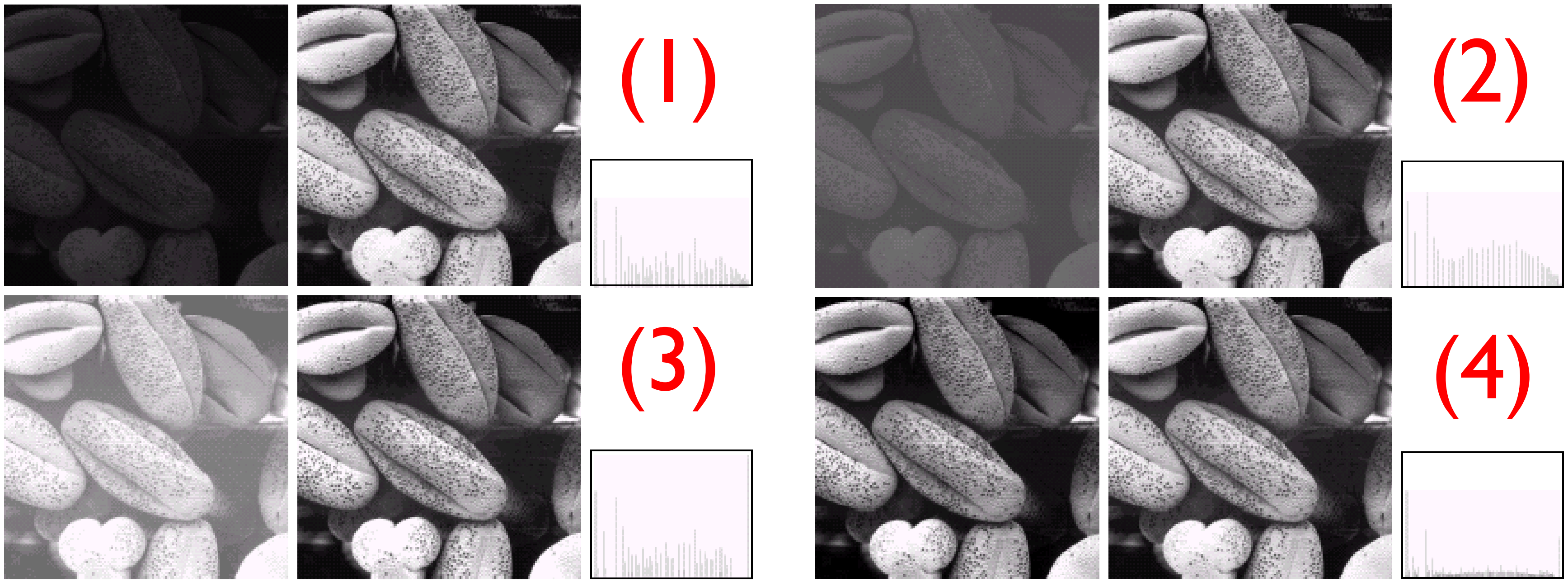
Histogram equalization examples

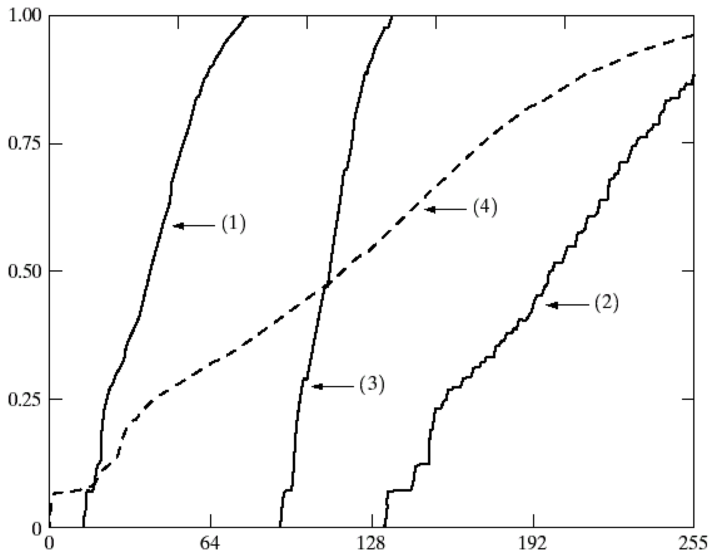




Slide credit: C. Dyer

Histogram equalization examples





Histogram Specification

- Given an input image *f* and a specific histogram $p_2(r)$, transform the image so that it has the specified histogram

source target mapped source

- How to perform histogram specification?
- Histogram equalization produces a (nearly) uniform output histogram
- Use histogram equalization as an intermediate step

Image credit: Y. Hel-Or

Histogram Specification

1. Equalize the histogram of the input image

$$T_1(r) = (L-1)\int_0^r p_1(w)\,dw$$

2. Histogram equalize the desired output histogram

$$T_2(r) = (L-1)\int_0^r p_2(w)\,dw$$

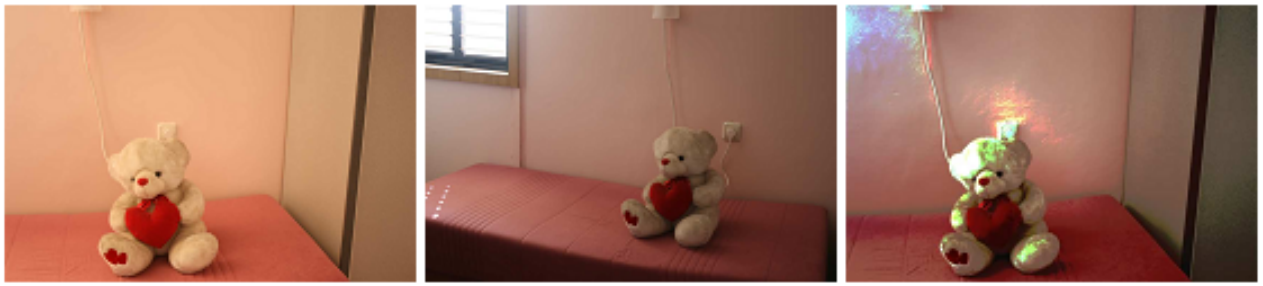
3. Histogram specification can be carried out by the following point operation:

$$s = T(r) = T_2^{-1}(T_1(r))$$

Histogram Specification

- In cases where corresponding colors between images are not “consistent”, this mapping may fail:

(a) (b) (c)



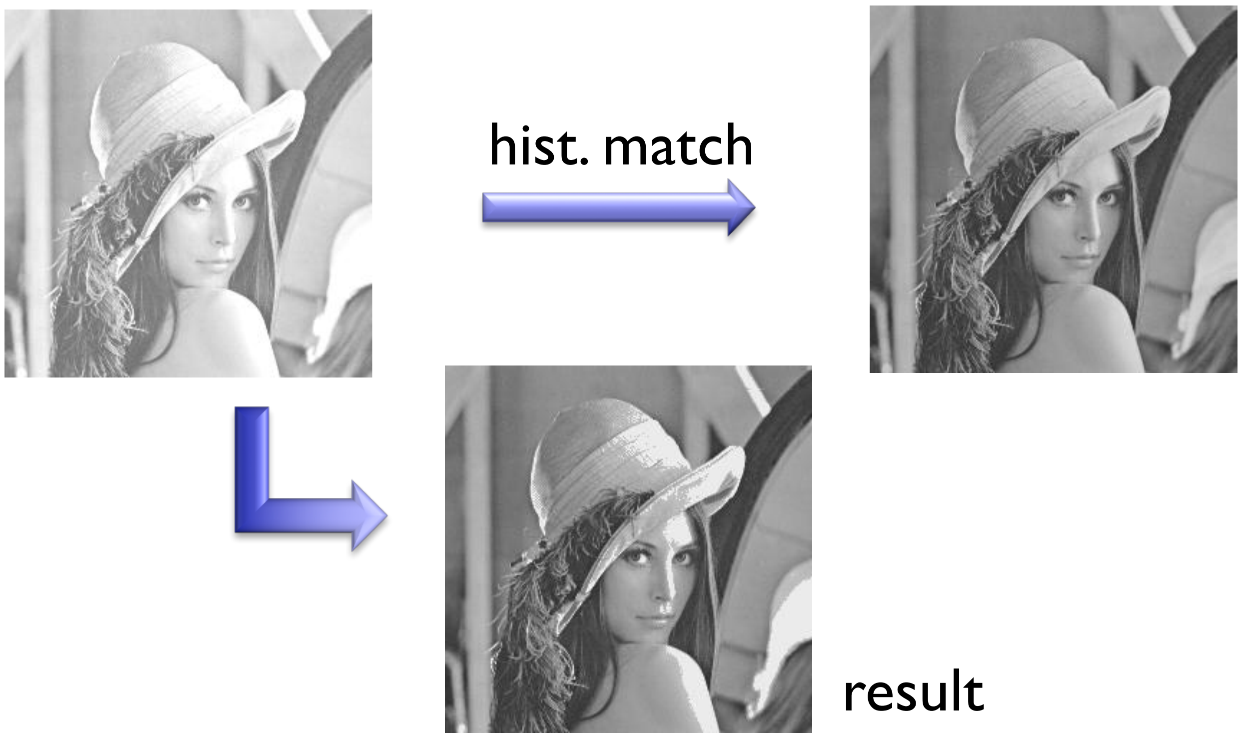
(a) (b) (c)

Images from: S. Kagarlitsky, M.Sc. thesis 2010.

Slide credit: Y. Hel-Or

Histogram Specification: Discussion

- Histogram matching produces the optimal **monotonic** mapping so that the resulting histogram will be as **close** as possible to the target histogram.
- This does not necessarily imply similar images.



Slide credit: Y. Hel-Or

Next week

- Spatial filtering