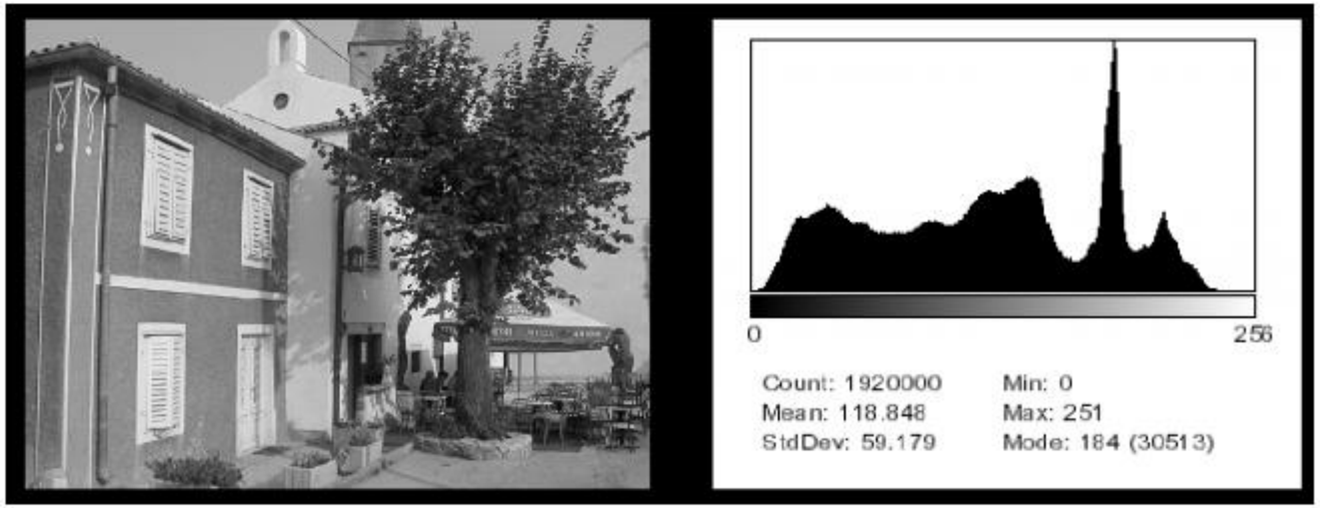
BSB663
Image Processing

Pinar Duygulu

Slides are adapted from
Gonzales & Woods,
Emmanuel Agu
Suleyman Tosun

Histograms

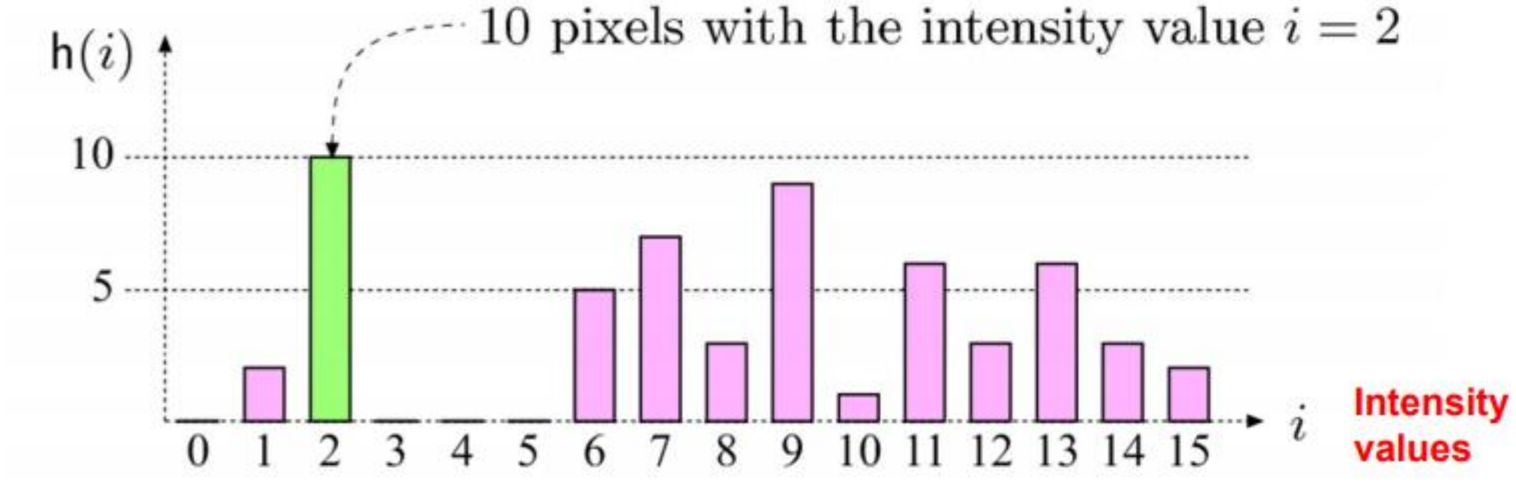
- Histograms plots how many times (frequency) each intensity value in image occurs
- Example:
  - Image (left) has 256 distinct gray levels (8 bits)
  - Histogram (right) shows frequency (how many times) each gray level occurs



Histograms

- Many cameras display real time histograms of scene
- Helps avoid taking over-exposed pictures
- Also easier to detect types of processing previously applied to image

Histograms

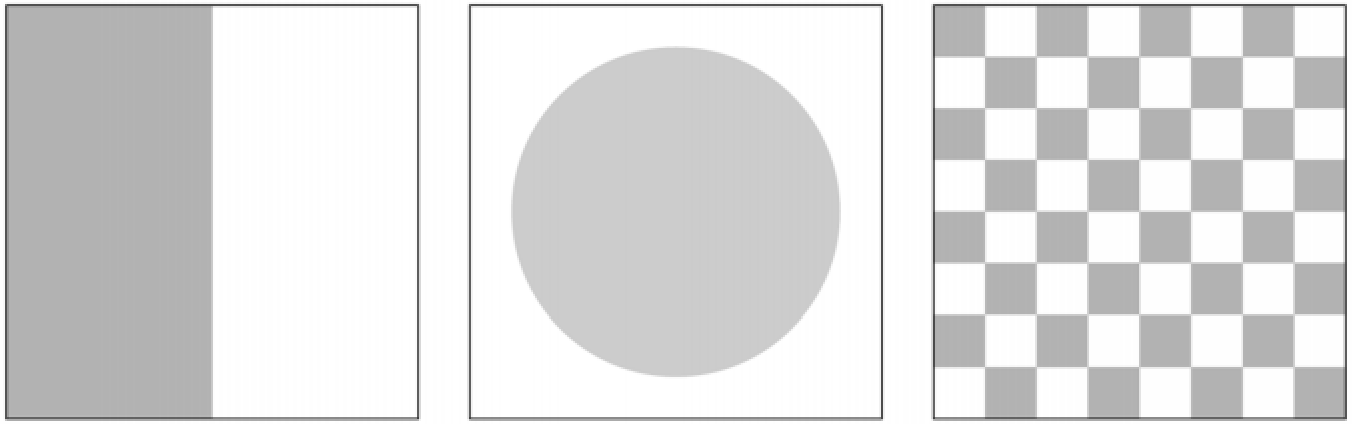


h(i)	0	2	10	0	0	0	5	7	3	9	1	6	3	6	3	2
i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

- E.g. K = 16, 10 pixels have intensity value = 2
- Histograms: only statistical information
- No indication of **location** of pixels

Histograms

- Different images can have **same** histogram
- 3 images below have same histogram



- Half of pixels are gray, half are white
  - Same histogram = same statistics
  - Distribution of intensities could be different
- Can we reconstruct image from histogram? No!

Histograms

- So, a histogram for a grayscale image with intensity values in range

$$I(u,v) \in [0, K-1]$$

would contain exactly *K* entries

- E.g. 8-bit grayscale image, $K = 2^8 = 256$
- Each histogram entry is defined as:

  h(i) = number of pixels with intensity *I* for all 0 < *i* < *K*.

- E.g: h(255) = number of pixels with intensity = 255
- Formal definition $$\mathrm{h}(i) = \mathrm{card}\{(u,v) \mid I(u,v) = i\}$$

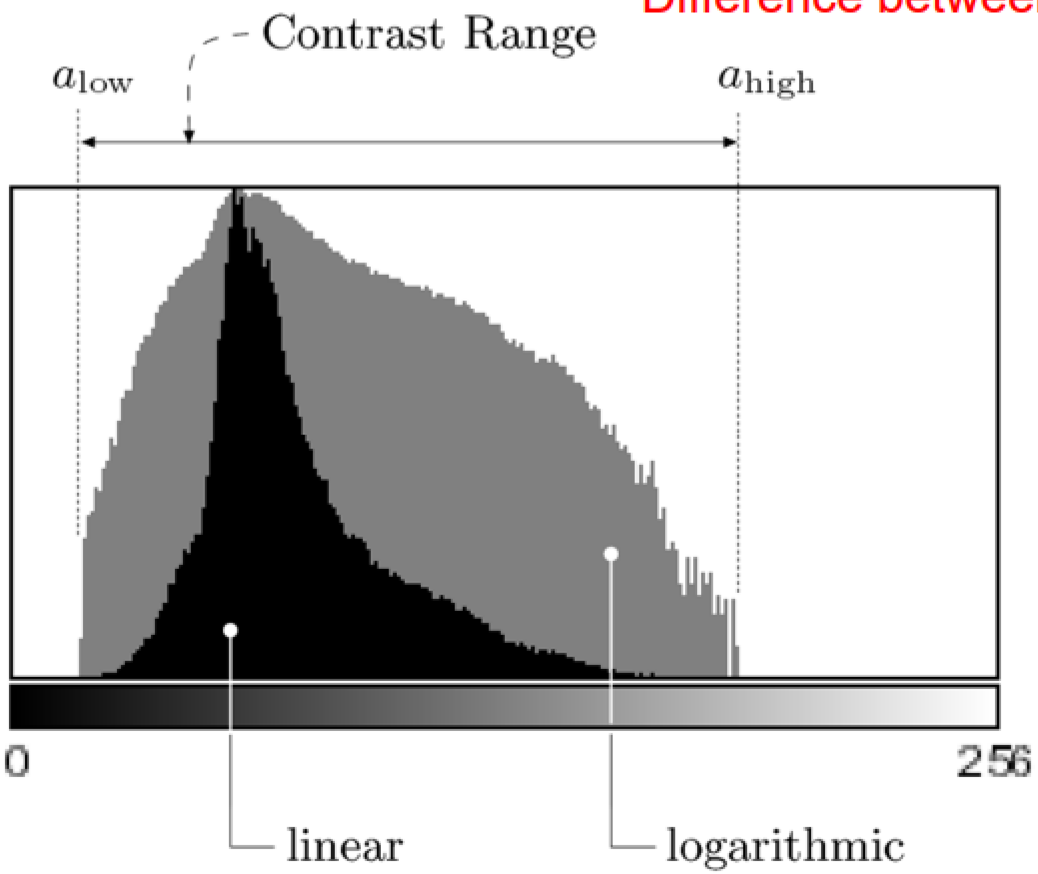
Number (size of set) of pixels

such that

Interpreting histograms

- Log scale makes low values more visible

Difference between darkest and lightest



Histograms

- Histograms help detect image acquisition issues
- Problems with image can be identified on histogram
  - Over and under exposure
  - Brightness
  - Contrast
  - Dynamic Range
- Point operations can be used to alter histogram. E.g
  - Addition
  - Multiplication
  - Exp and Log
  - Intensity Windowing (Contrast Modification)

Image Brightness

- Brightness of a grayscale image is the **average intensity** of all pixels in image

$$B(I) = \frac{1}{wh} \sum_{v=1}^{h} \sum_{u=1}^{w} I(u, v)$$

2. Divide by total number of pixels

1. Sum up all pixel intensities

Detecting Bad Exposure using Histograms

Exposure? Are intensity values spread (good) out or bunched up (bad)

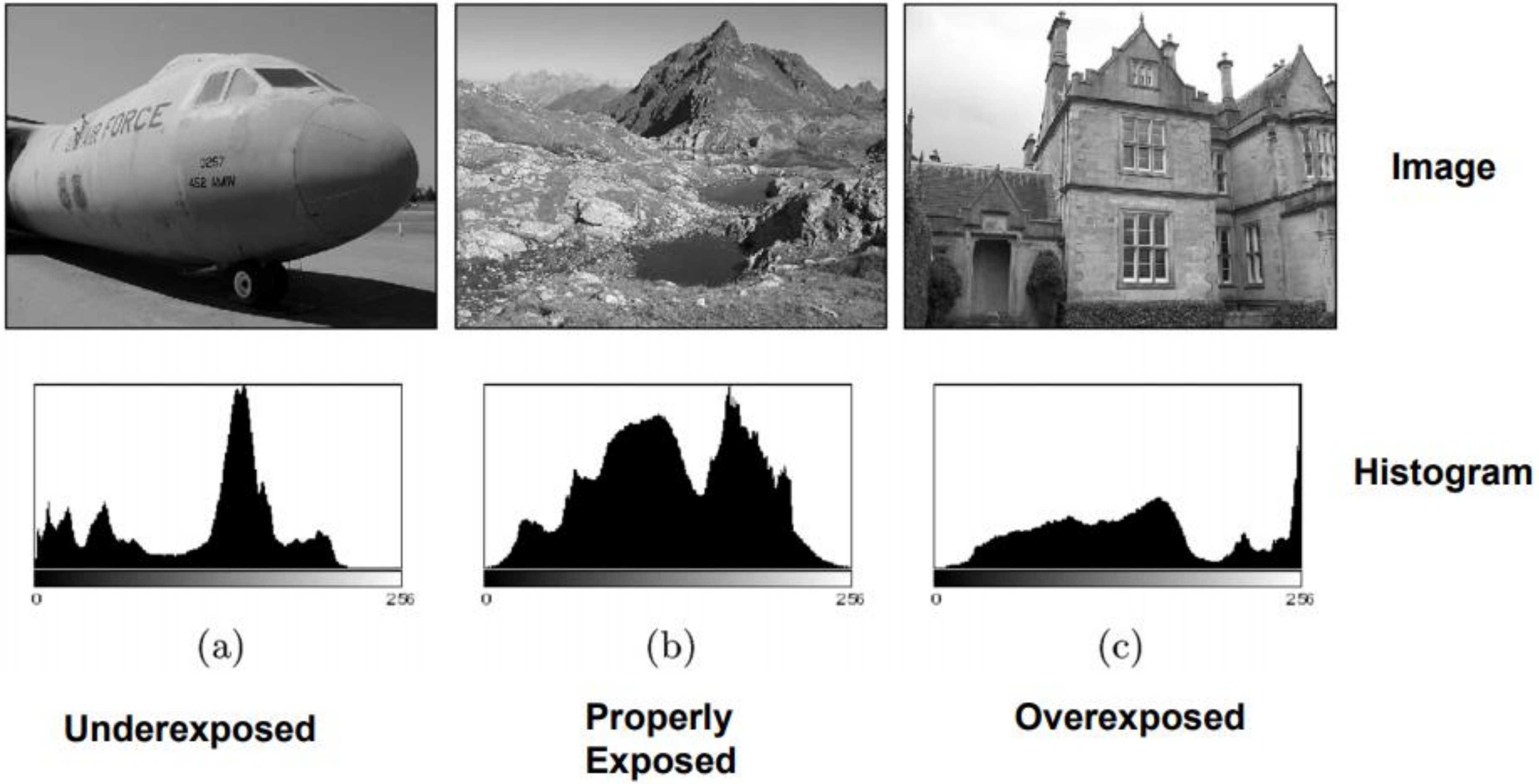
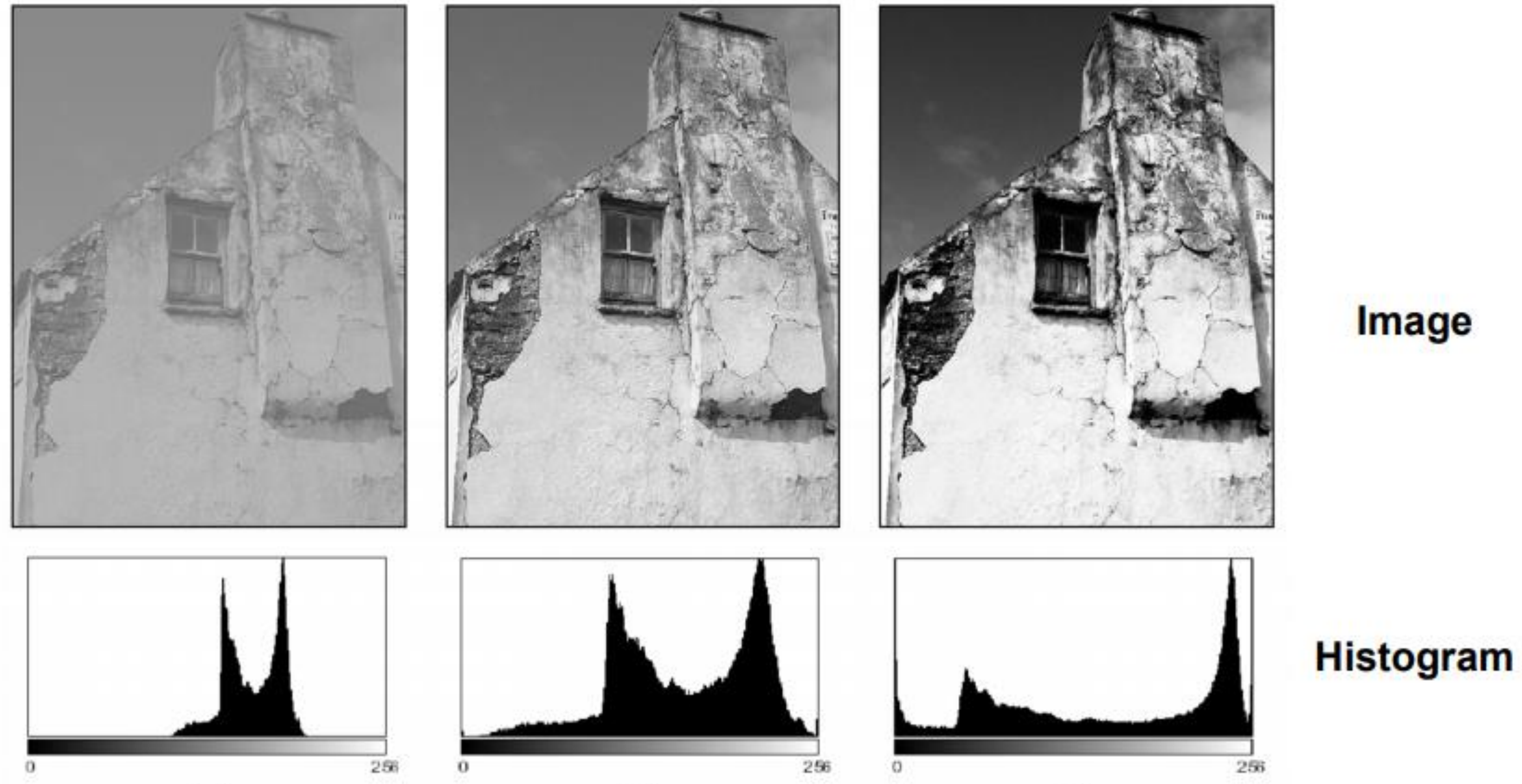


Image Contrast

- The contrast of a grayscale image indicates how easily objects in the image can be distinguished
- **High contrast image:** many distinct intensity values
- **Low contrast:** image uses few intensity values

Histograms and contrast

Good Contrast? Widely spread intensity values
+ large difference between min and max intensity values



(a) **Low contrast** (b) **Normal contrast** (c) **High contrast**

Contrast equation

- Many different equations for contrast exist
- Examples:

$$\text{Contrast} = \frac{\text{Change in Luminance}}{\text{Average Luminance}}$$

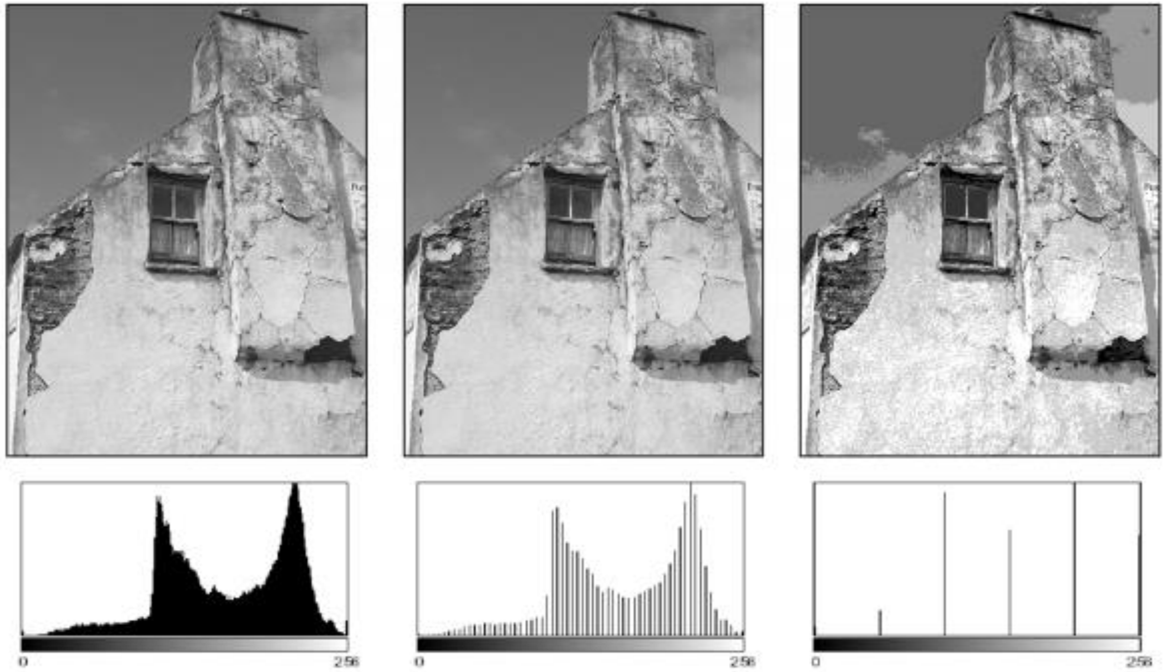
- Michalson's equation for contrast

$$C_M(I) = \frac{\max(I) - \min(I)}{\max(I) + \min(I)}$$

- These equations work well for simple images with 2 luminances (i.e. uniform foreground and background)
- Does not work well for complex scenes with many luminances or if min and max intensities are small

Histograms and dynamic range

- **Dynamic Range:** Number of distinct pixels in image



(a) **High Dynamic Range**

(b) **Low Dynamic Range (64 intensities)**

(c) **Extremely low Dynamic Range (6 intensity values)**

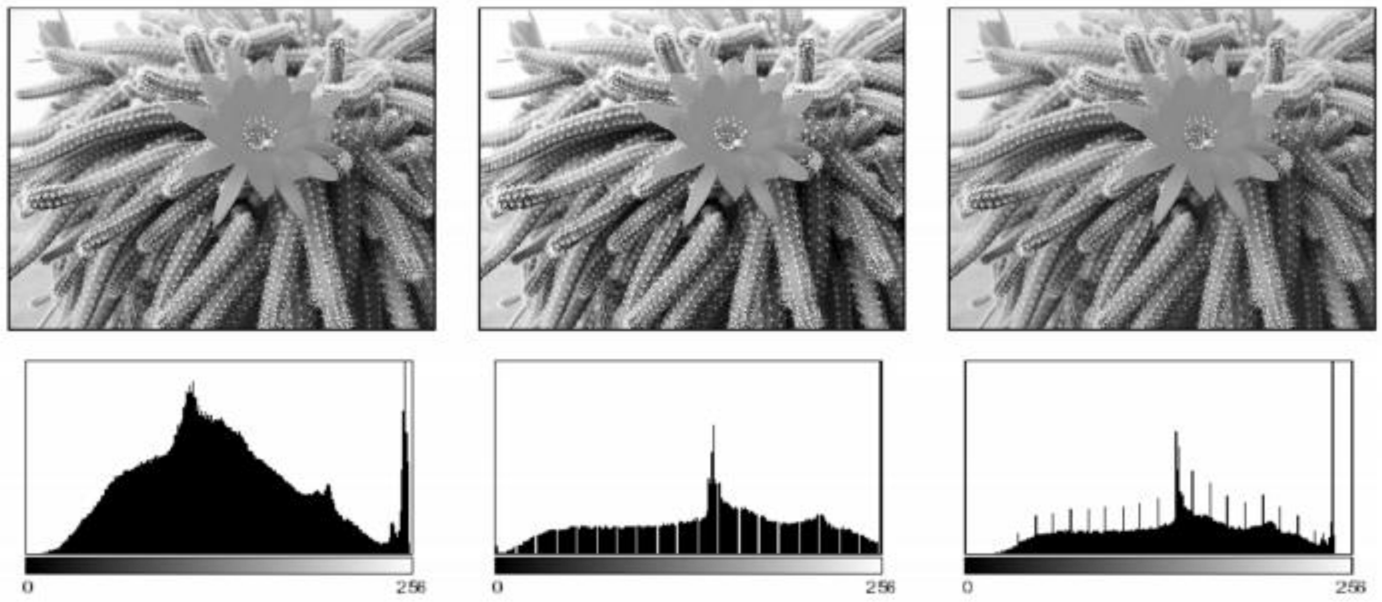
- Difficult to increase image dynamic range (e.g. interpolation)
- HDR (12-14 bits) capture typical, then down-sample

High Dynamic Range (HDR) Imaging

- **High dynamic range** means very bright and very dark parts in a single image (many distinct values)
- Dynamic range in photographed scene may exceed number of available bits to represent pixels
- Solution:
  - Capture multiple images at different exposures
  - Combine them using image processing

Detecting Image Defects using Histograms

- No "best" histogram shape, depends on application
- Image defects
  - **Saturation:** scene illumination values outside the sensor's range are set to its min or max values => results in spike at ends of histogram
  - **Spikes and Gaps in manipulated images** (not original). Why?

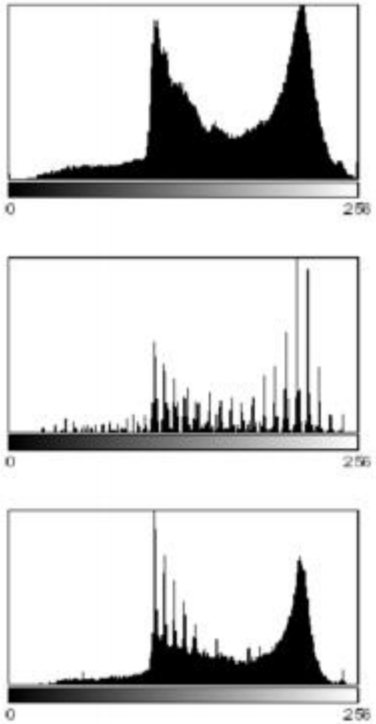


(a) (b) (c)

Image Defects: Effect of Image Compression

- Histograms show impact of image compression
- Example: in GIF compression, dynamic range is reduced to only few intensities (quantization)

Original Image



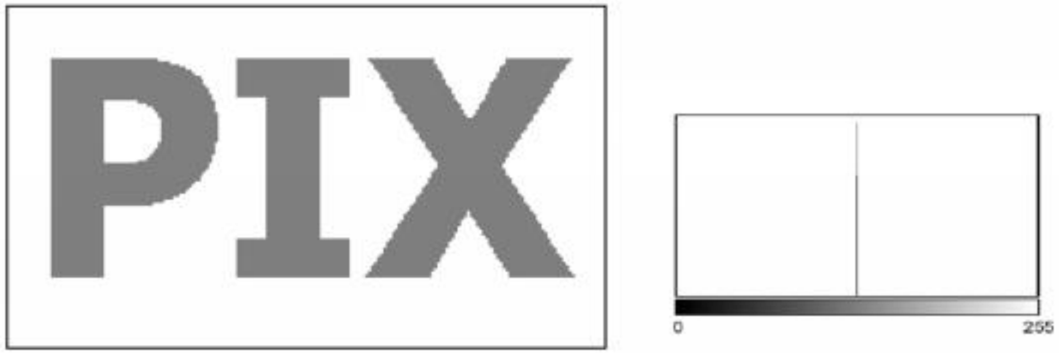
(a) **Original Histogram**

(b) **Histogram after GIF conversion**

(c) **Fix? Scaling image by 50% and Interpolating values recreates some lost colors**
But GIF artifacts still visible

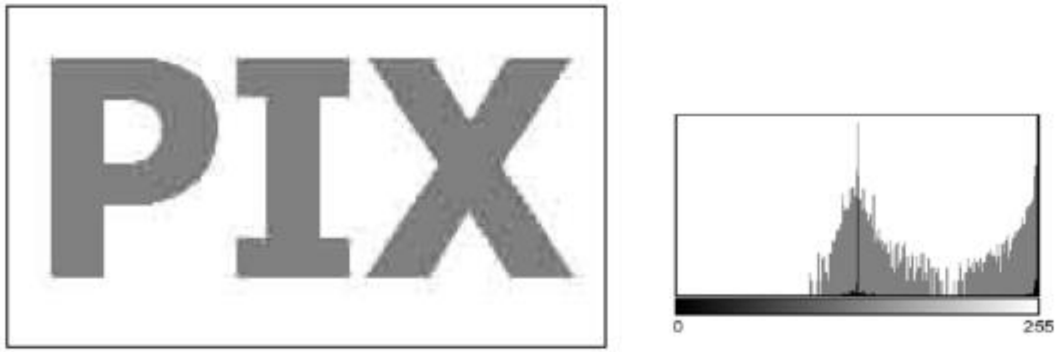
Effect of Image Compression

- Example: Effect of JPEG compression on line graphics
- JPEG compression designed for color images



(a) (b)

Original histogram has only 2 intensities (gray and white)



(c) (d)

JPEG image appears dirty, fuzzy and blurred

Its Histogram contains gray values not in original

Computing Histograms

```
Hist = zeros(256);
[w, h] = size(I);
for (int v = 0; v<h; v++)
    for (int u=0; u<w; u++)
        i= I(u,v);
        Hist[i] = Hist[i] + 1;
```

Large Histograms: Binning

- High resolution image can yield very large histogram
- Example: 32-bit image = 2^{32} = 4,294,967,296 columns
- Such a large histogram impractical to display
- Solution? Binning!
  - Combine **ranges of intensity values** into histogram columns

So, given the image $I : \Omega \rightarrow [0, K-1]$, the binned histogram for I is the function

$$h(i) = \text{card}\{(u,v) \mid a_i \leq I(u,v) < a_{i+1}\},$$

where $0 = a_0 < a_1 < \ldots < a_B = K$.

Number (size of set) of pixels

such that

Pixel's intensity is between a_i and a_{i+1}

Calculating Bin Size

- Typically use equal sized bins
- Bin size? $$\frac{\textit{Number of distinct values in image}}{\textit{Number of bins}}$$

- Example: To create 256 bins from 14-bit image

$$Bin\ size = \frac{2^{14}}{256} = 64$$

$$\begin{array}{lcrcl}
\mathsf{h}(0) & \leftarrow & 0 \leq I(u,v) < & 64 \\
\mathsf{h}(1) & \leftarrow & 64 \leq I(u,v) < & 128 \\
\mathsf{h}(2) & \leftarrow & 128 \leq I(u,v) < & 192 \\
\vdots & & \vdots \quad \vdots & \vdots \\
\mathsf{h}(j) & \leftarrow & a_j \leq I(u,v) < & a_{j+1} \\
\vdots & & \vdots \quad \vdots & \vdots \\
\mathsf{h}(255) & \leftarrow & 16320 \leq I(u,v) < & 16384
\end{array}$$

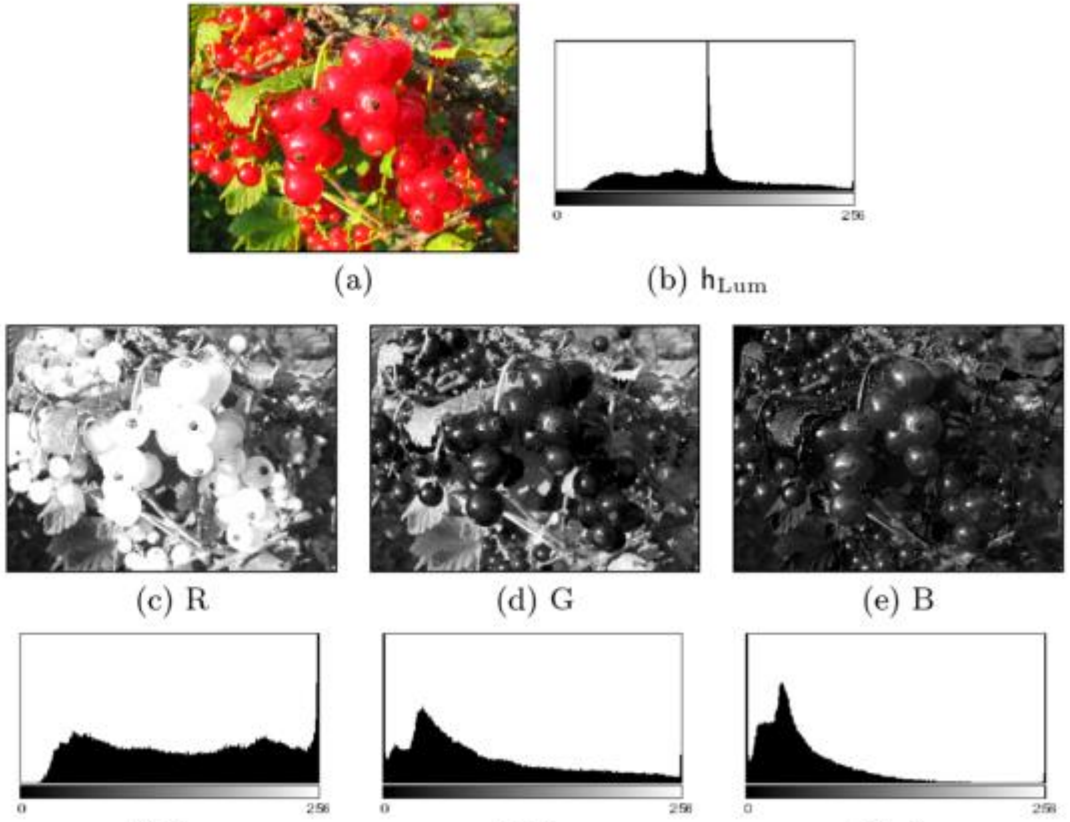
Binned histogram

```
K = 256;
B = 32;
Hist = zeros(B);
[w, h] = size(I);
for (int v = 0; v<h; v++)
    for (int u=0; u<w; u++)
        a= I(u,v);
        i = a* B / K
        Hist[i] = Hist[i] + 1;
```

Color Image Histograms

Two types:

1. **Intensity histogram:**
   - Convert color image to gray scale
   - Display histogram of gray scale
2. **Individual Color Channel Histograms:** 3 histograms (R,G,B)



(a) (b) h_{Lum}
(c) R (d) G (e) B
(f) h_R (g) h_G (h) h_B

Color Image Histograms

- Both types of histograms provide useful information about lighting, contrast, dynamic range and saturation effects
- No information about the actual color distribution!
- Images with totally different RGB colors can have same R, G and B histograms
- Solution to this ambiguity is the **Combined Color Histogram**.
  - More on this later

Cumulative Histograms

- Useful for certain operations (e.g. histogram equalization) later
- Analogous to the **Cumulative Density Function (CDF)**
- Definition:

$$\mathsf{H}(i) = \sum_{j=0}^{i} \mathsf{h}(j) \quad \text{for} \quad 0 \leq i < K$$

- Recursive definition

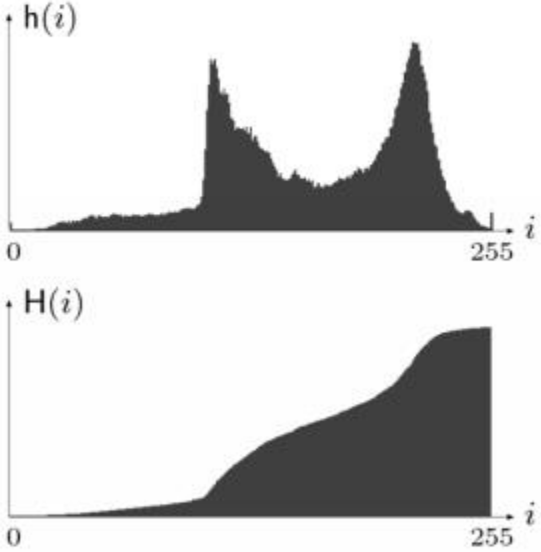
$$\mathsf{H}(i) = \begin{cases} \mathsf{h}(0) & \text{for} \quad i = 0 \\ \mathsf{H}(i-1) + \mathsf{h}(i) & \text{for} \quad 0 < i < K \end{cases}$$

- Monotonically increasing

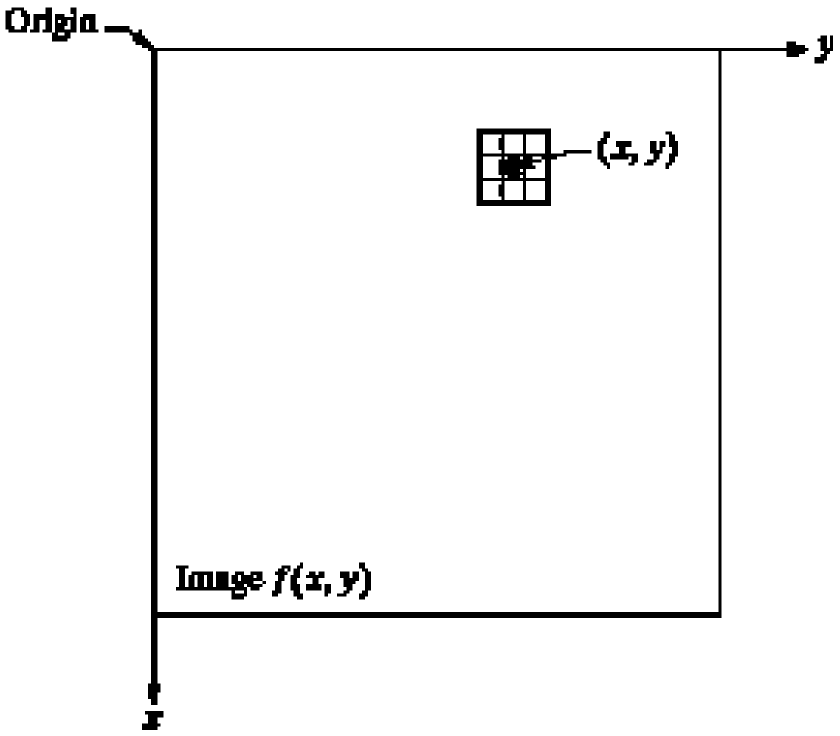
$$\mathsf{H}(K-1) = \sum_{j=0}^{K-1} \mathsf{h}(j) = M \cdot N$$

Last entry of Cum. histogram

Total number of pixels in image



Point Operations



Procedures that operate directly on the pixels composing an image.

$$I'(x,y) = f[I(x,y)]$$

where

- $I(x,y)$ is the input image
- $I'(x,y)$ is the processed image
- f is an operator on I

Point Operations

- Point operations changes a pixel's intensity value according to some function (don't care about pixel's neighbor)

$$a' \leftarrow f(a)$$
$$I'(u,v) \leftarrow f\big(I(u,v)\big)$$

- Also called a **homogeneous operation**
- New pixel intensity **depends on**
  - Pixel's previous intensity *I(u,v)*
  - Mapping function *f*( )
- **Does not depend on**
  - Pixel's location *(u,v)*
  - Intensities of neighboring pixels

Some homogeneous point operations

- Addition (Changes brightness)

  $f(p) = p + k$ E.g. $f_{\text{bright}}(p) = p + 10$

- Multiplication (Stretches/shrinks image contrast range)

  $f(p) = k \times p$ E.g. $f_{\text{contrast}}(p) = p \text{ x } 1.5$

- Real-valued functions

  $\exp(x), \log(x), (1/x), x^k$, etc.

- Quantizing pixel values
- Global thresholding
- Gamma correction

Pseudocode

- Input: Image with pixel intensities I(u,v) defined on [1 .. w] x [1 .. H]
- Output: Image with pixel intensities I’(u,v)

```
for v = 1 .. h
    for u = 1 .. w
        I’(u, v) = f (I(u,v))
```

Non-homogeneous point operations

- New pixel value depends on:
  - Old value + **pixel's location *(u,v)***

$$a' \leftarrow g(a, u, v)$$
$$I'(u, v) \leftarrow g\left(I(u, v), u, v\right)$$

Clamping

- Deals with pixel values outside displayable range
  - If (a > 255)   a = 255;
  - If (a < 0) a = 0;
- Function below will **clamp** (force) all values to fall within range [*a*,*b*]

$$f(p) = \begin{cases} a & \text{if } p < a \\ p & \text{if } a \leq p \leq b \\ b & \text{if } p > b \end{cases}$$

Example: Modify Intensity and Clamp

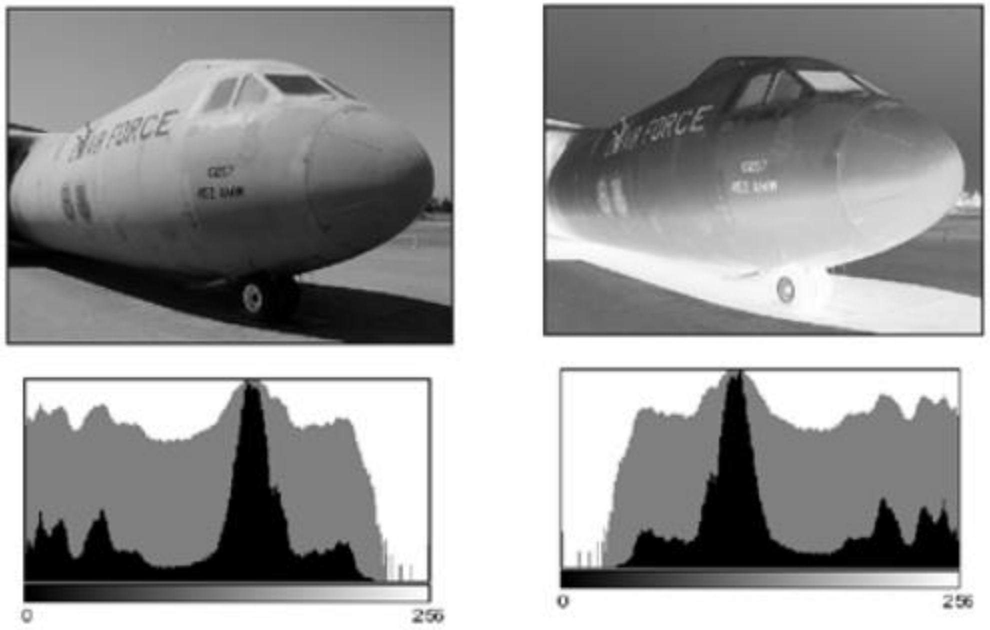
Point operation: increase image contrast by 50%
then clamp values above 255

```
[w, h] = size(I);
for (int v = 0; v<h; v++)
     for (int u=0; u<w; u++)
          a= I(u,v) * 1.5 + 0.5;
          if (a> 255) a=255
      I'(u,v) = a;
```

Inverting images

$$f_{\text{invert}}(a) = -a + a_{\max} = a_{\max} - a$$

- 2 steps
  1. Multiple intensity by -1
  2. Add constant (e.g. a_{max}) to put result in range $[0,a_{max}]$
- Implemented as ImageJ method `invert( )`



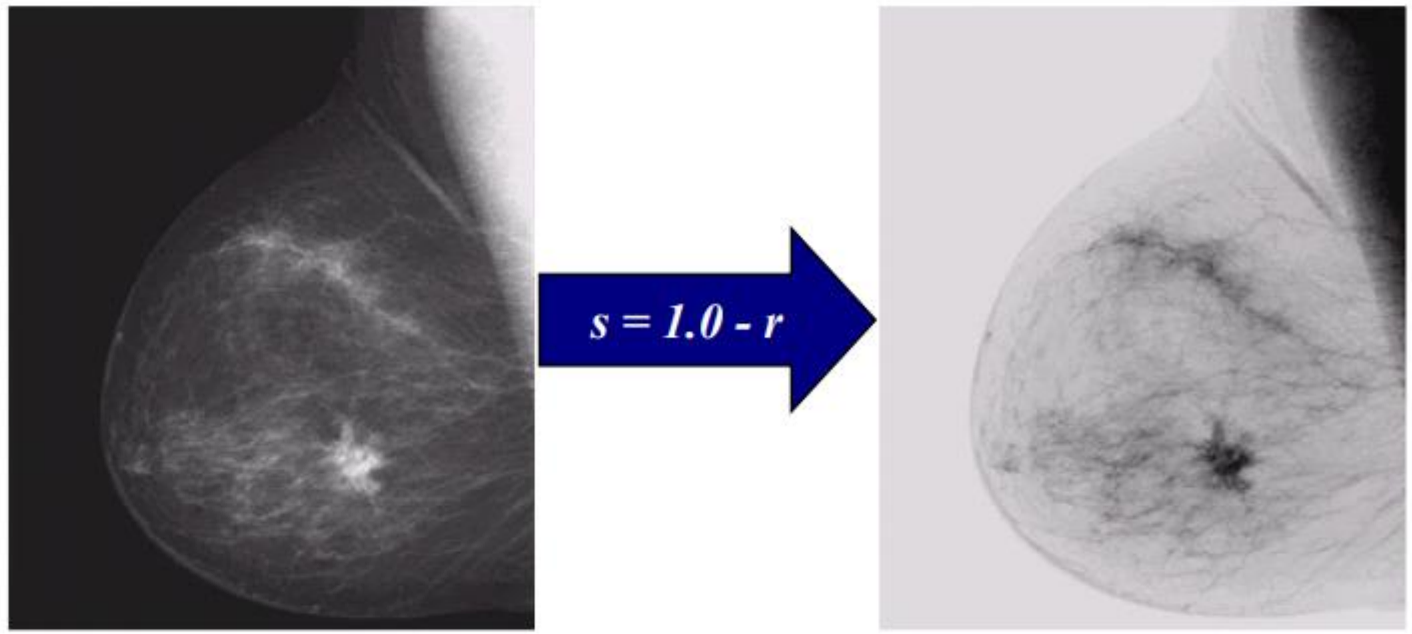
(a) (c)

Original Inverted Image

Image Negatives

- Image negatives useful for enhancing white or grey detail embedded in dark regions of an image
  - Note how much clearer the tissue is in the negative image of the mammogram below

Original Image



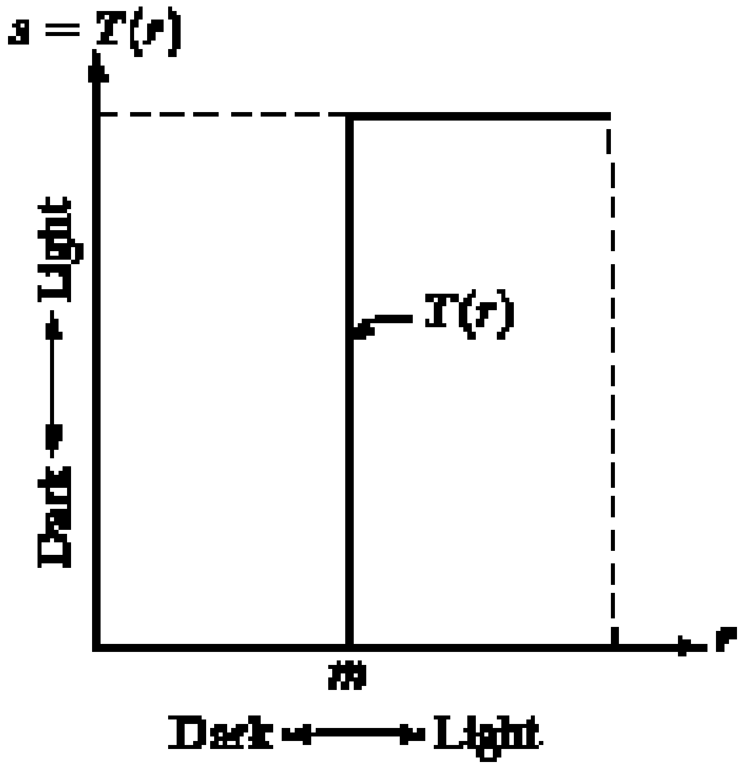
Negative Image

Thresholding

- Input values below **threshold** a_{th} set to a_0
- Input values above **threshold** a_{th} set to a_1

$$f_{\text{threshold}}(a) = \begin{cases} a_0 & \text{for } a < a_{\text{th}} \\ a_1 & \text{for } a \geq a_{\text{th}} \end{cases}$$

- Converts grayscale image to binary image (binarization) if
  - $a_0 = 0$
  - $a_1 = 1$
- Implemented as imageJ method `threshold( )`

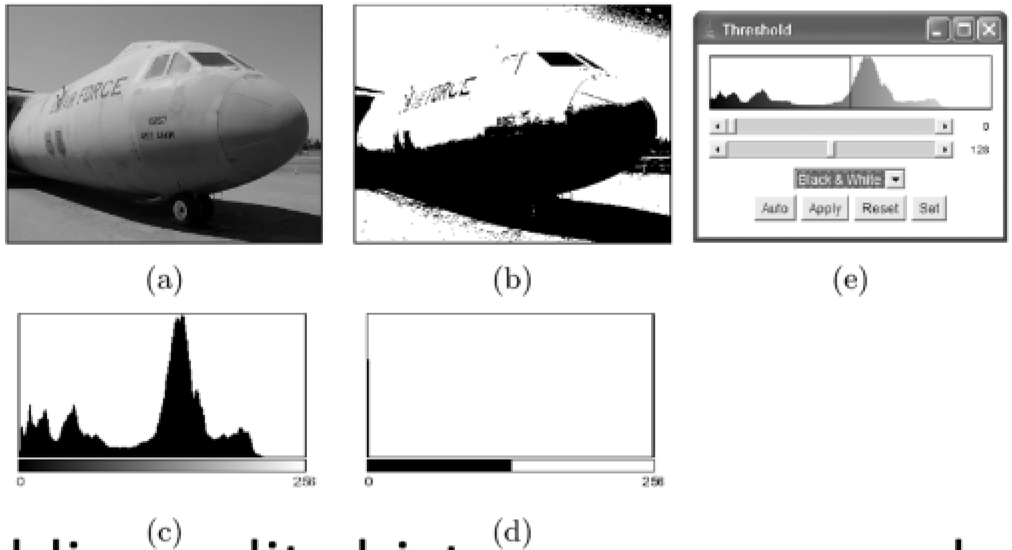


Original Image

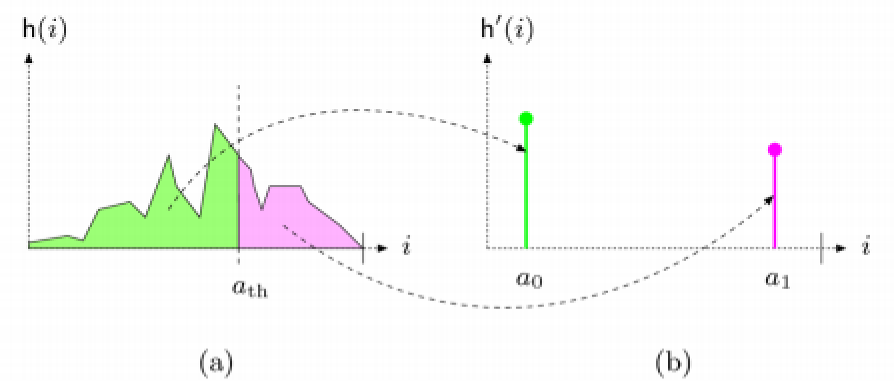
Thresholded Image

Thresholding and histograms

- Example with a_{th} = 128



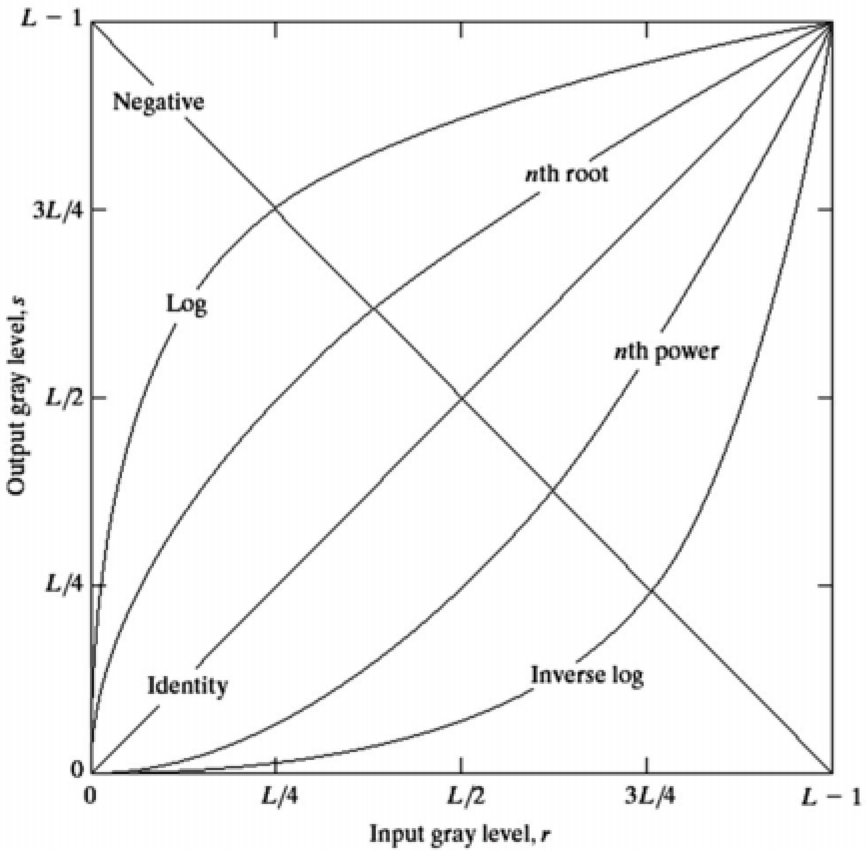
- Thresholding splits histogram, merges halves into a_0 a_1



Basic grey-level transformations

Images taken from Gonzalez & Woods, Digital Image Processing (2002)

- 3 most common gray level transformation:
  - Linear
    - Negative/Identity
  - Logarithmic
    - Log/Inverse log
  - Power law
    - n^{th} power/n^{th} root

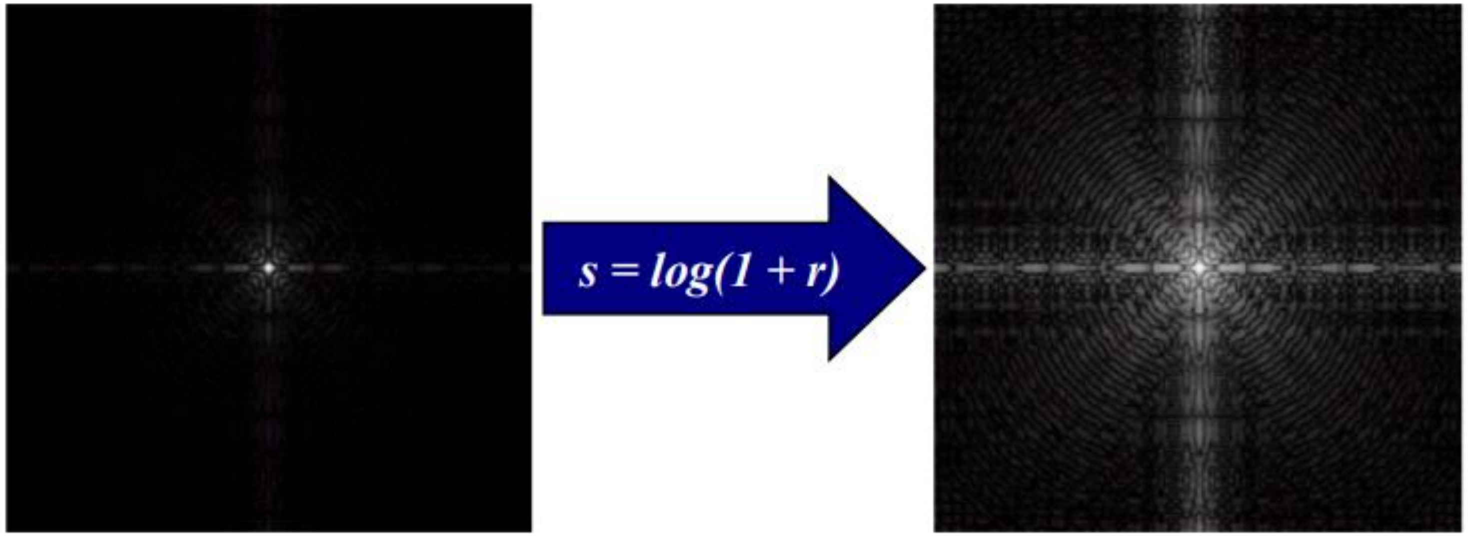


Logarithmic transformations

- Maps narrow range of input levels => wider range of output values
- Inverse log transformation does opposite transformation
- The general form of the log transformation is

New pixel value → $$s = c * \log(1 + r)$$ ← Old pixel value

- Log transformation of Fourier transform shows more detail



Power Law transformations

Images taken from Gonzalez & Woods, Digital Image Processing (2002)

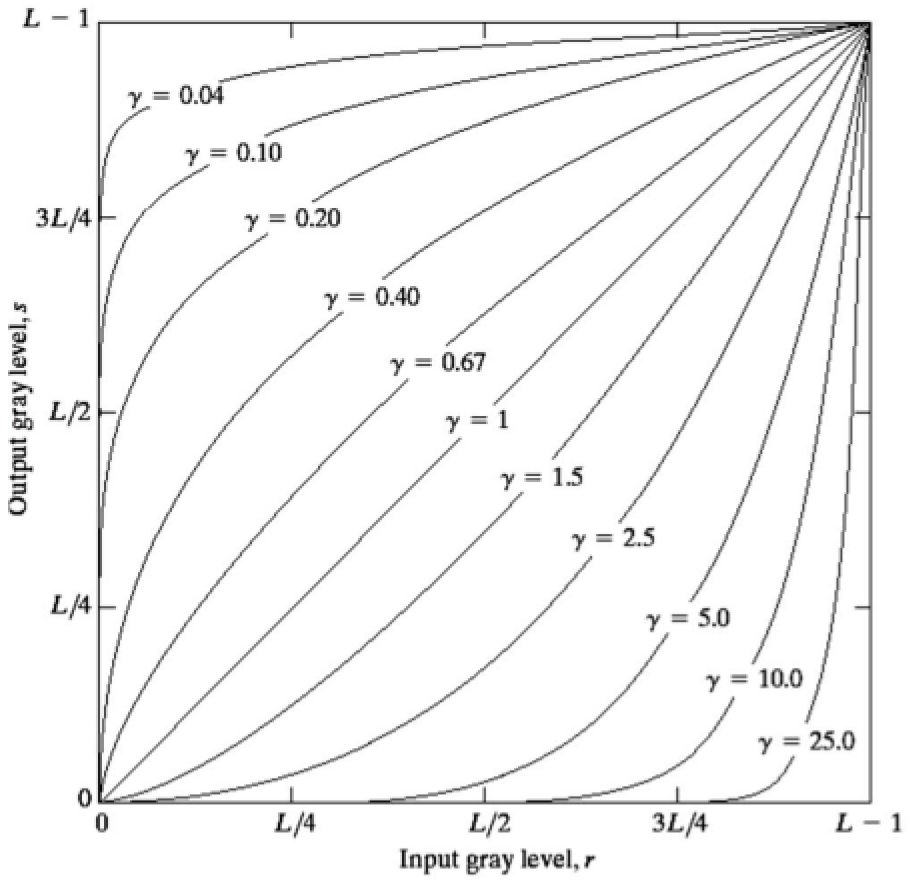
- Power law transformations have the form

$$s = c * r^{\gamma}$$

New pixel value (s), Constant (c), Old pixel value (r), Power (γ)

- Map narrow range of dark input values into wider range of output values or vice versa

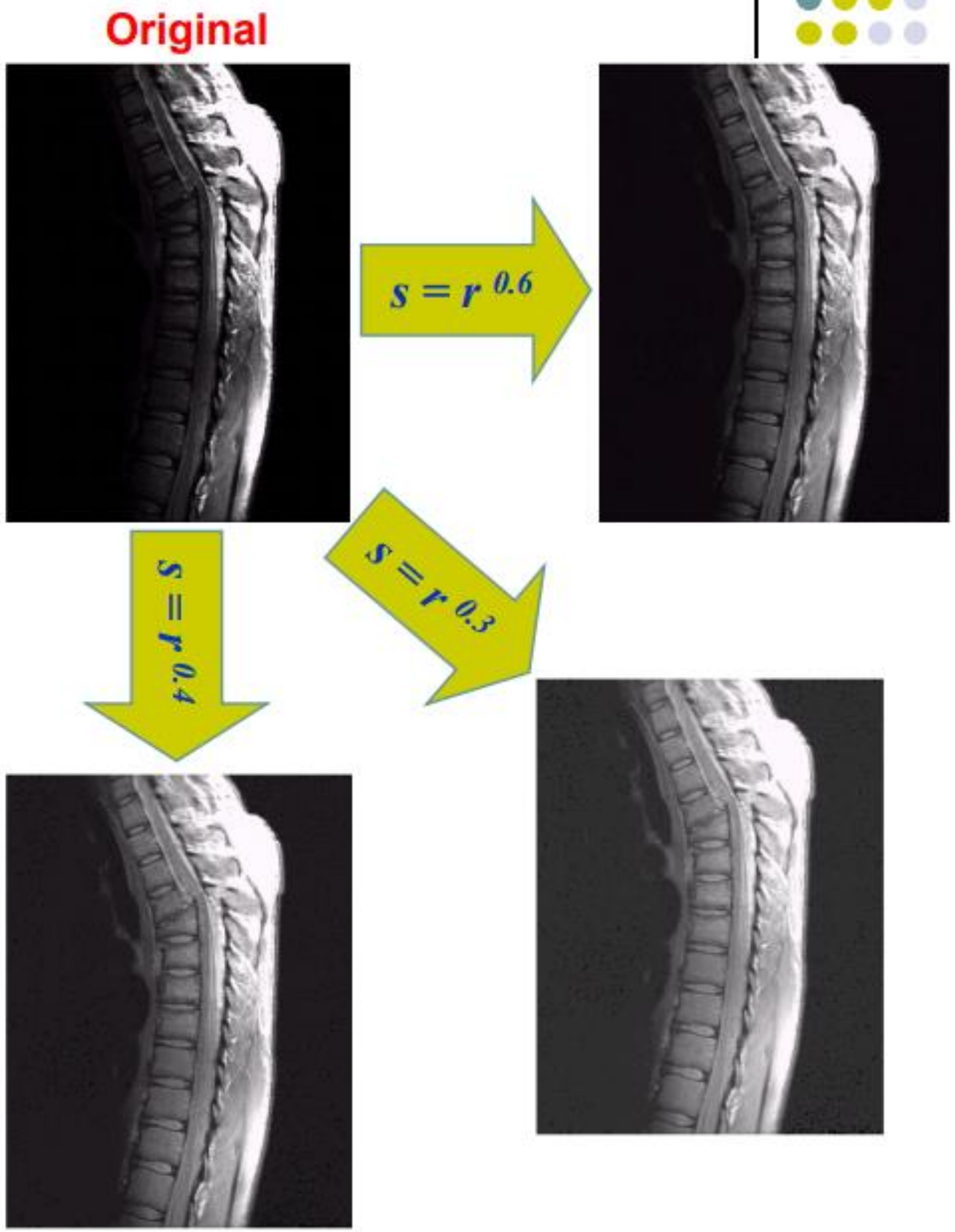
- Varying γ gives a whole family of curves



Images taken from Gonzalez & Woods, Digital Image Processing (2002)

- Magnetic Resonance (MR) image of fractured human spine

- Different power values highlight different details



Effect of decreasing gamma

- When the γ is reduced too much, the image begins to reduce contrast to the point where the image may start to have slight "washed-out" look, especially in the background

(a) image has a washed-out appearance, it needs a compression of lighter gray levels ⇨ needs $\gamma > 1$

(b) result after power-law

transformation with $\gamma = 3.0$ (suitable)

(c) transformation with $\gamma = 4.0$

(suitable)

(d) transformation with $\gamma = 5.0$

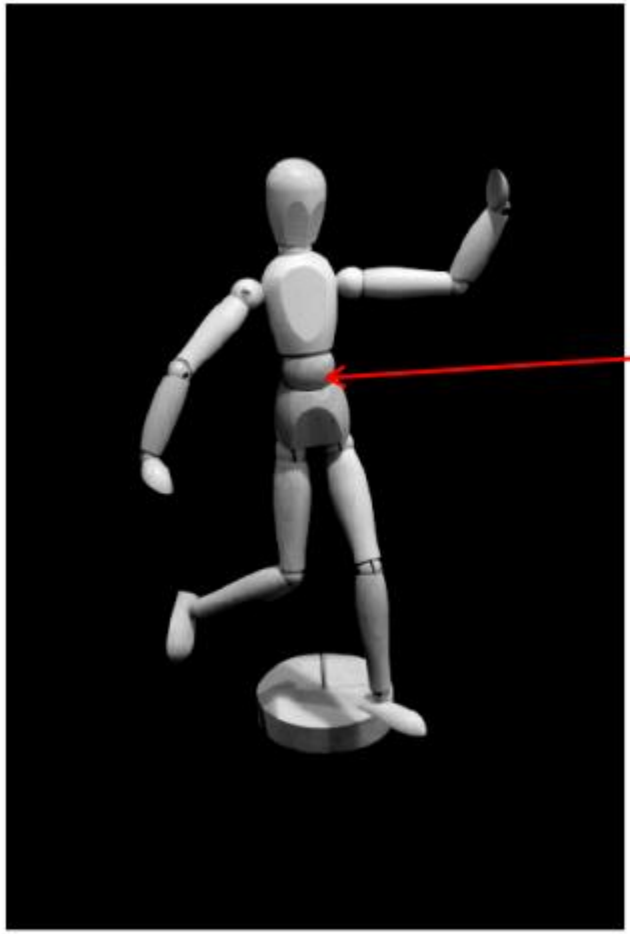
(high contrast, the image has areas that are too dark, some detail is lost)

Intensity windowing

- A clamp operation, then linearly stretching image intensities to fill possible range
- To window an image in [*a*,*b*] with max intensity ***M***

$$f(p) = \begin{cases} 0 & \text{if } p < a \\ M \times \frac{p-a}{b-a} & \text{if } a \leq p \leq b \\ M & \text{if } p > b \end{cases}$$

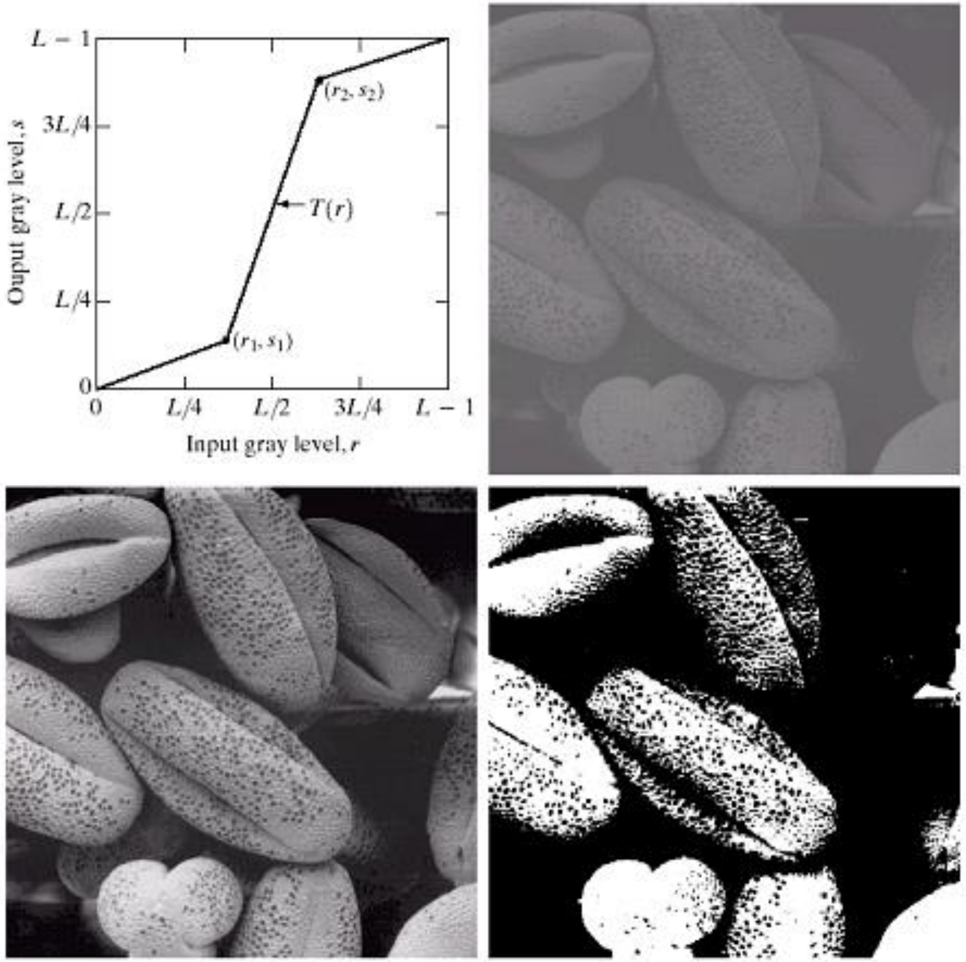
Original Image



Windowed Image

Contrasts easier to see

Contrast Stretching



- increase the dynamic range of the gray levels in the image
- (b) a low-contrast image : result from poor illumination, lack of dynamic range in the imaging sensor, or even wrong setting of a lens aperture of image acquisition
- (c) result of contrast stretching: $(r_1,s_1) = (r_{min},0)$ and $(r_2,s_2) = (r_{max},L-1)$
- (d) result of thresholding ($r_1=r_2=m$, binary image). m, mean grey level in the image.

Contrast Stretching

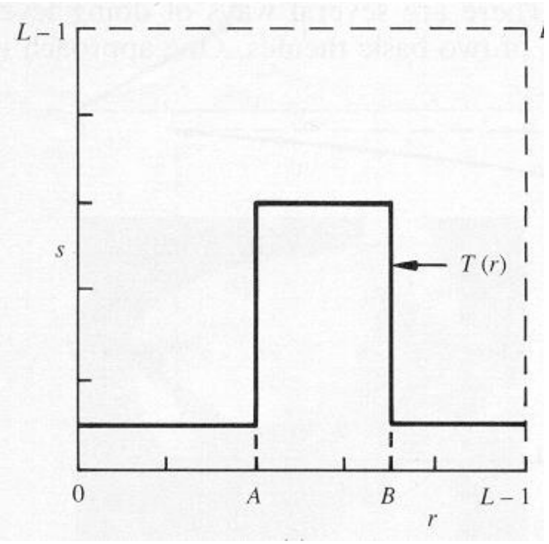
- The locations of (r_1,s_1) and (r_2,s_2) control the shape of the transformation function.
  - If $r_1= s_1$ and $r_2= s_2$ the transformation is a linear function and produces no changes.
  - If $r_1=r_2$, $s_1=0$ and $s_2=L-1$, the transformation becomes a thresholding function that creates a binary image.
  - Intermediate values of (r_1,s_1) and (r_2,s_2) produce various degrees of spread in the gray levels of the output image, thus affecting its contrast.


© all rights reserved: Ravadits Kornél 2001 MAX R3


© all rights reserved: Ravadits Kornél 2001 MAX R3

Gray-Level Slicing

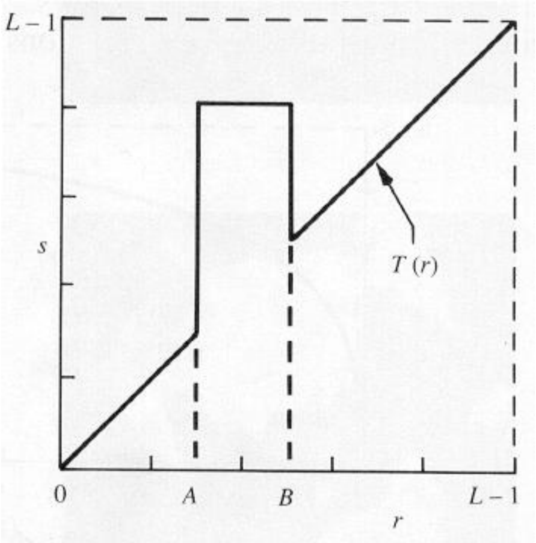
- To highlight a specific range of gray levels in an image (e.g. to enhance certain features).



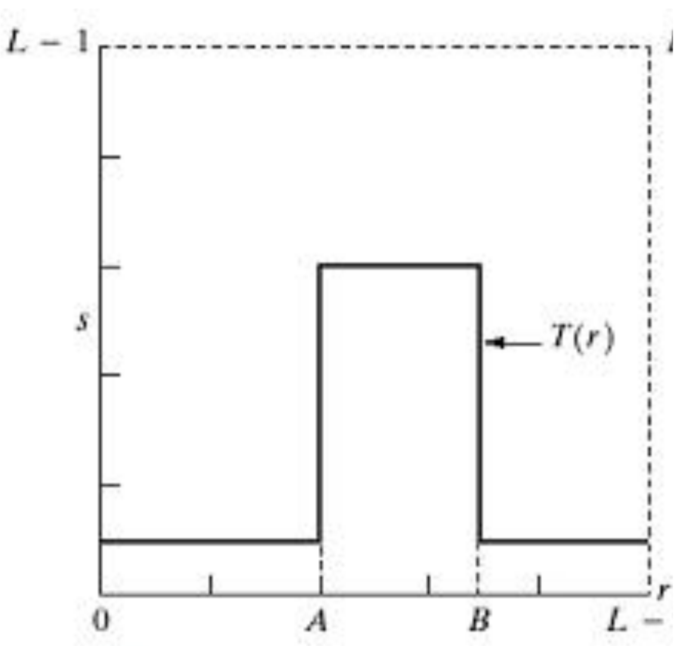
One way is to display a high value for all gray levels in the range of interest and a low value for all other gray levels (binary image).

Gray-Level Slicing

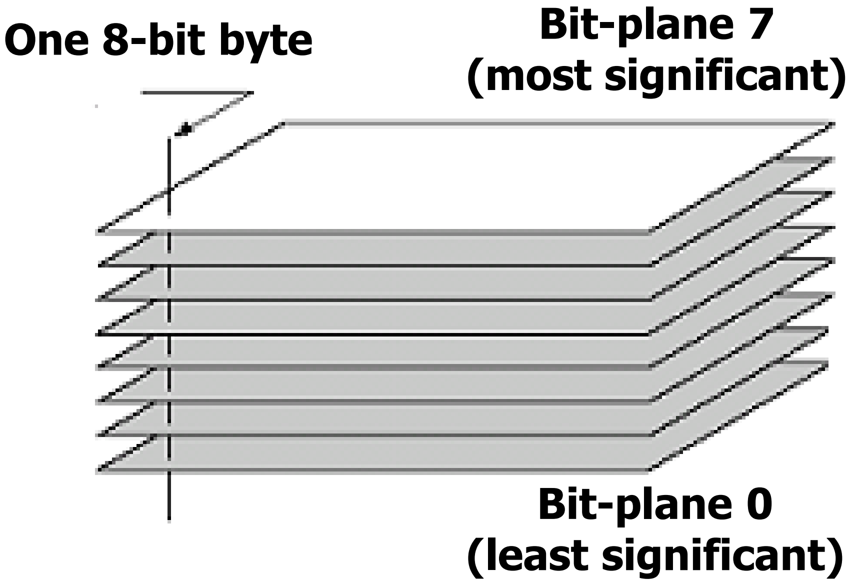
- The second approach is to brighten the desired range of gray levels but preserve the background and gray-level tonalities in the image:



Gray-level slicing



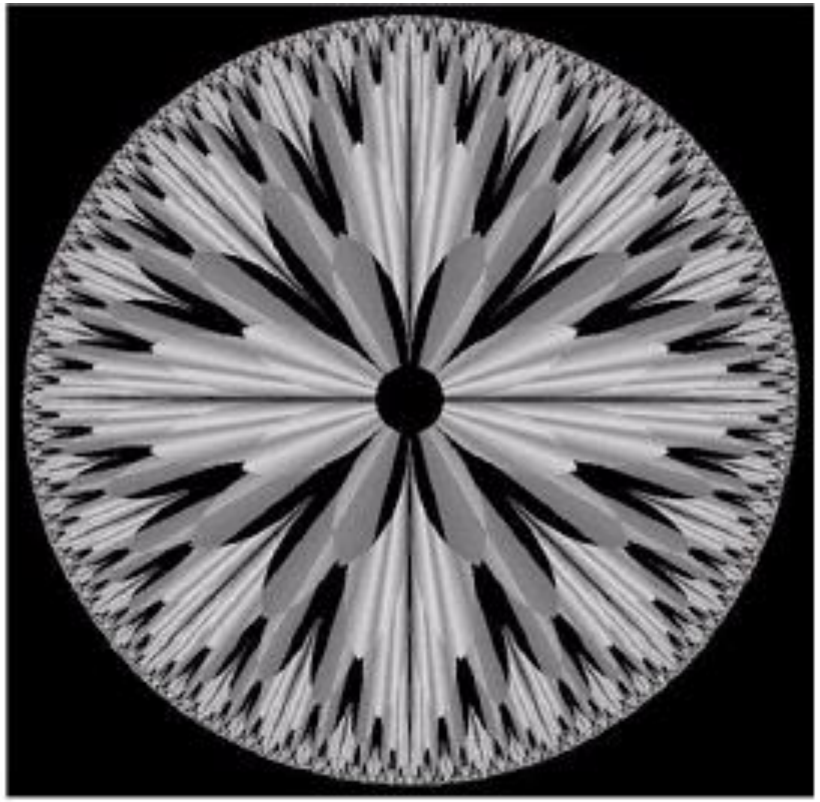
Bit-plane slicing



- Highlighting the contribution made to total image appearance by specific bits
- Suppose each pixel is represented by 8 bits
- Higher-order bits contain the majority of the visually significant data
- Useful for analyzing the relative importance played by each bit of the image

- bit planes:
  - Only the higher order bits (top four) contain visually significant data. The other bit planes contribute the more subtle details.
  - Plane 7 corresponds exactly with an image thresholded at gray level 128.
  - Plane 6 corresponds to grey levels in the ranges [64,127) and [192, 255)

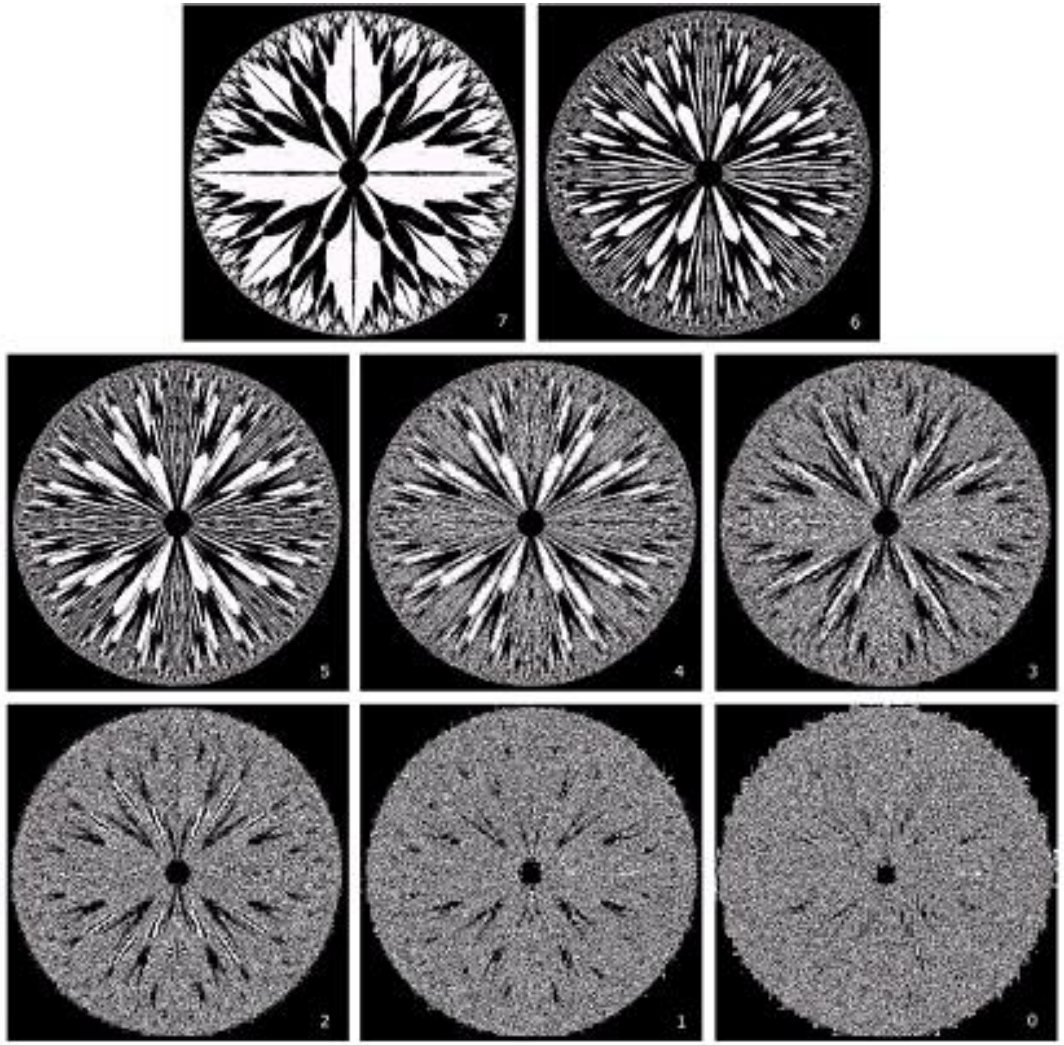
Example



An 8-bit fractal image

- The (binary) image for bit-plane 7 can be obtained by processing the input image with a thresholding gray-level transformation.
  - Map all levels between 0 and 127 to 0
  - Map all levels between 129 and 255 to 255

8 bit planes



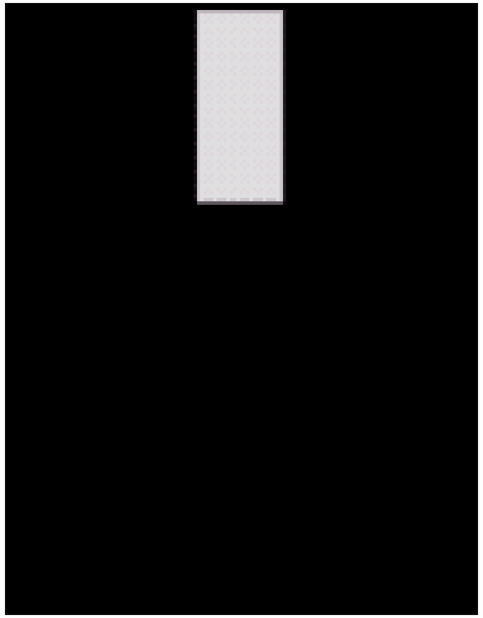
Bit-plane 7		Bit-plane 6	
Bit-plane 5	Bit-plane 4	Bit-plane 3	
Bit-plane 2	Bit-plane 1	Bit-plane 0	

Logic Operations

- Logic operation performs on gray-level images, the pixel values are processed as binary numbers
- light represents a binary 1, and dark represents a binary 0
- NOT operation = negative transformation

Example of AND Operation

original image

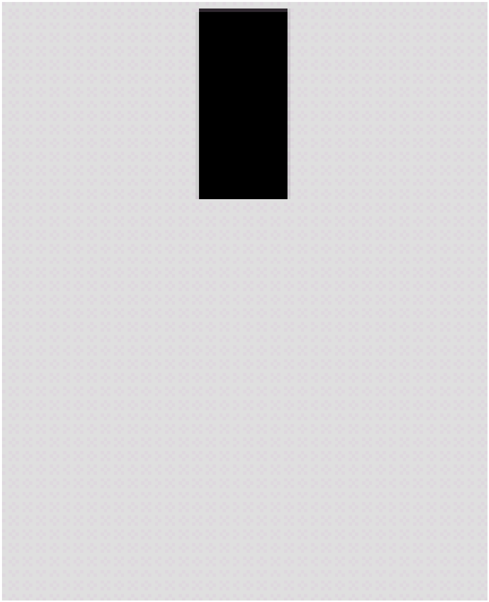


AND image mask

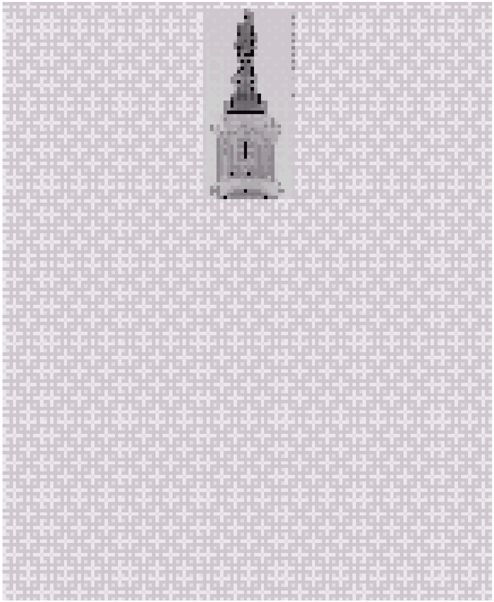
result of AND operation

Example of OR Operation

original image



OR image mask



result of OR operation

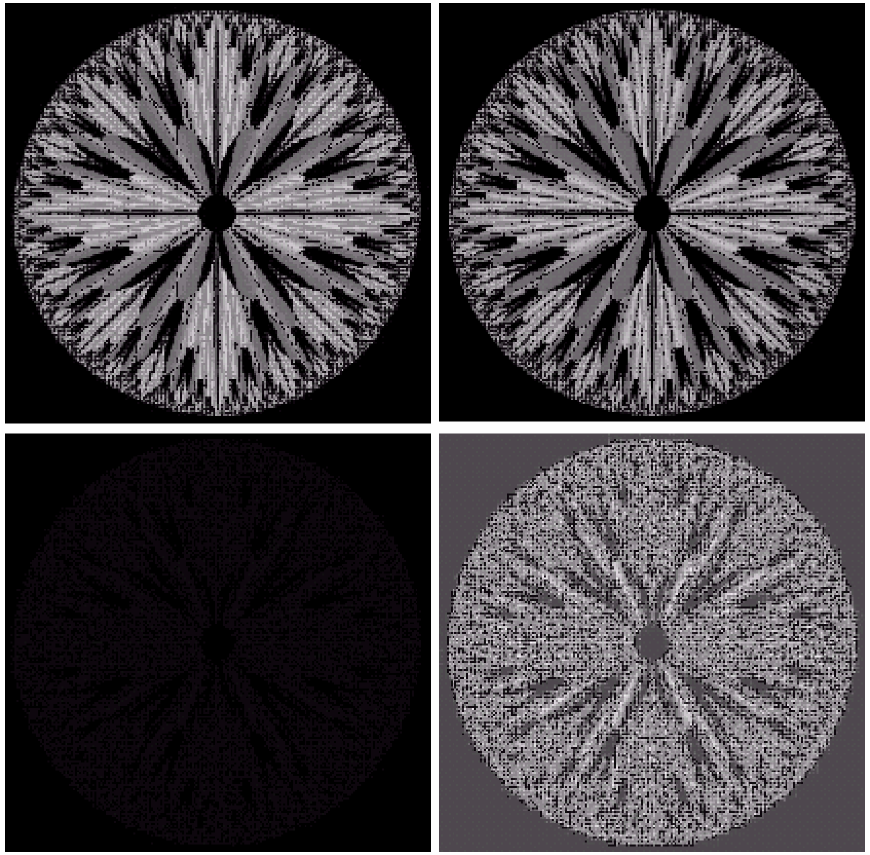
Image Subtraction

$$g(x,y) = f(x,y) - h(x,y)$$

- enhancement of the differences between images

a	b
c	d

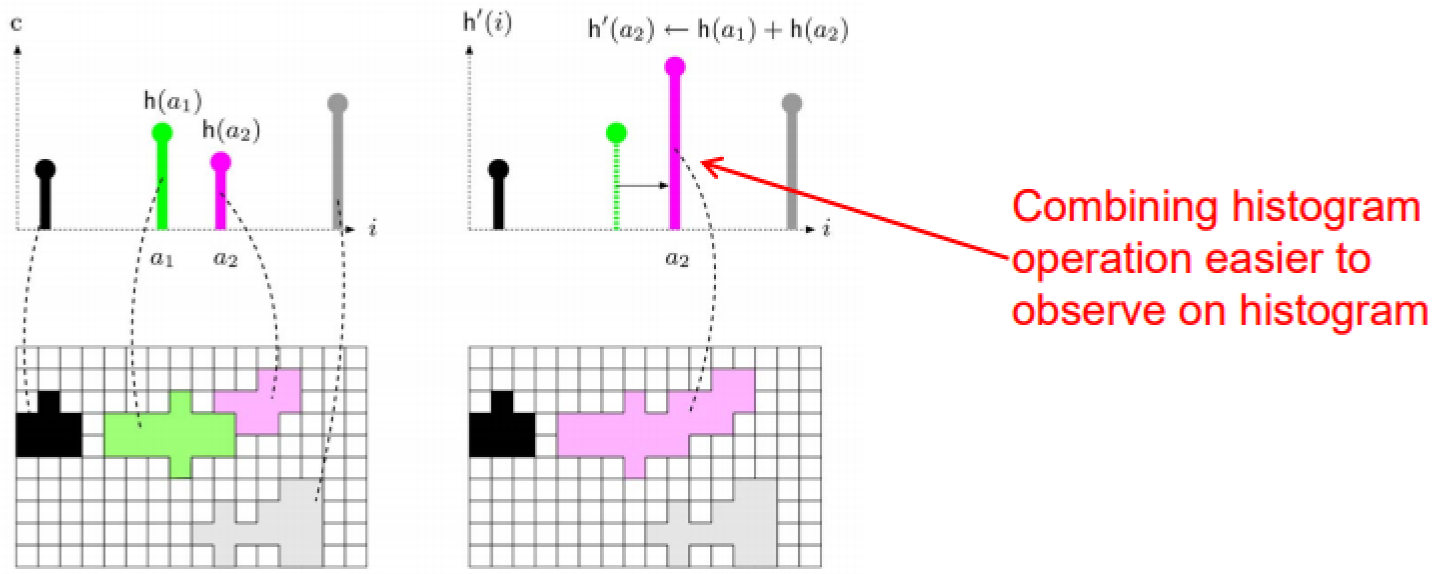
Image Subtraction



- a). original fractal image
- b). result of setting the four lower-order bit planes to zero
  - refer to the bit-plane slicing
  - the higher planes contribute significant detail
  - the lower planes contribute more to fine detail
  - image b). is nearly identical visually to image a), with a very slightly drop in overall contrast due to less variability of the gray-level values in the image.
- c). difference between a). and b). (nearly black)
- d). histogram equalization of c). (perform contrast stretching transformation)

Point Operations and Histograms

- Effect of some point operations easier to observe on histograms
  - Increasing brightness
  - Raising contrast
  - Inverting image
- Point operations only shift, merge histogram entries
- Operations that merge histogram bins are irreversible



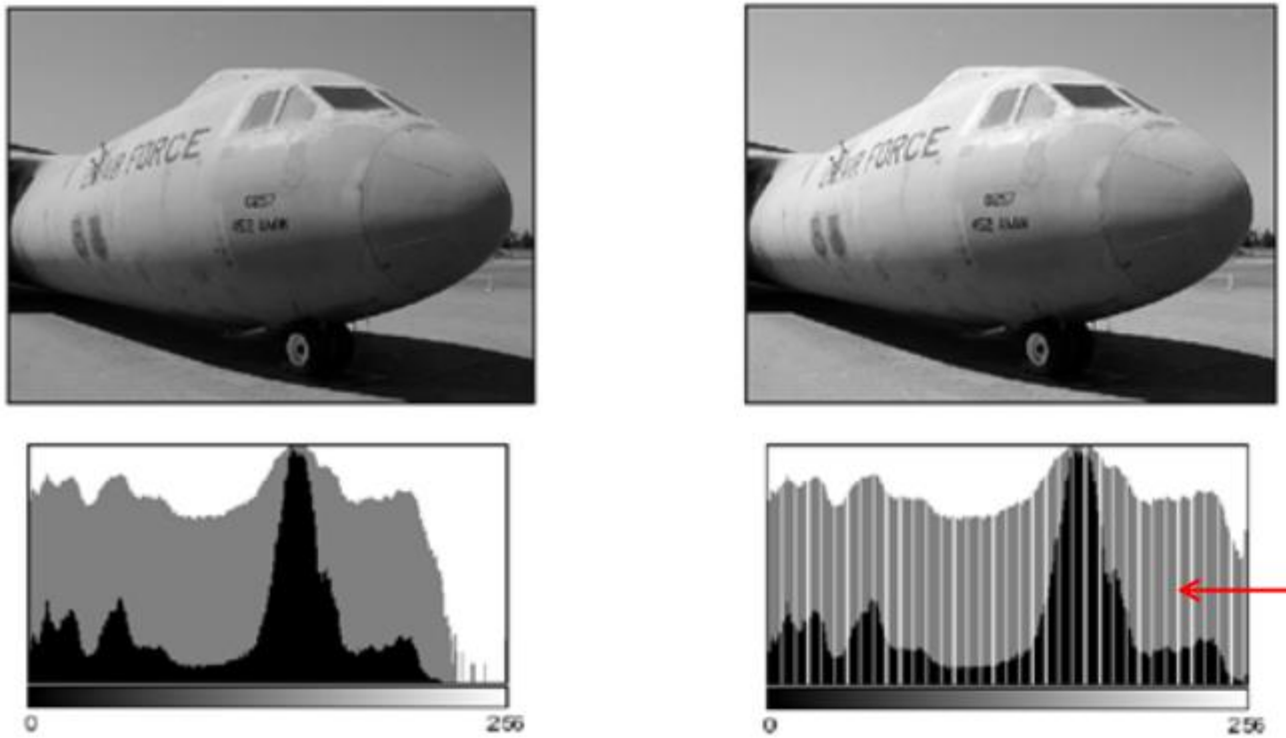
Automatic Contrast Adjustment

- Point operation that modifies pixel intensities such that available range of values is fully covered
- Algorithm:
  - Find high and lowest pixel intensities $a_{\text{low}}, a_{\text{high}}$
  - Linear stretching of intensity range

$$f_{\text{ac}}(a) = a_{\text{min}} + (a - a_{\text{low}}) \cdot \frac{a_{\text{max}} - a_{\text{min}}}{a_{\text{high}} - a_{\text{low}}}$$

If $a_{min} = 0$ and $a_{max} = 255$

$$f_{\text{ac}}(a) = (a - a_{\text{low}}) \cdot \frac{255}{a_{\text{high}} - a_{\text{low}}}$$



(a) (b)

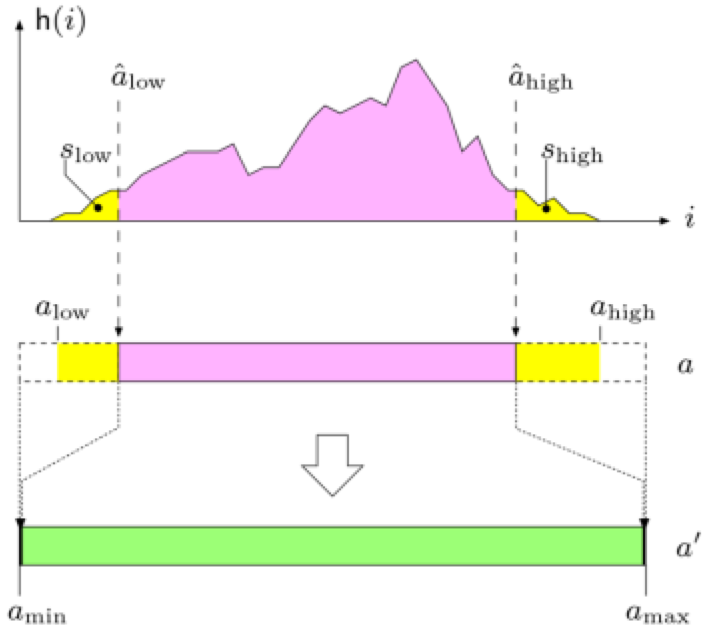
Original

Result of automatic Contrast Adjustment

Linearly stretching range causes gaps in histogram

Modified Contrast Adjustment

- Better to map only certain range of values
- Get rid of tails (usually noise) based on predefined percentiles ($s_{\text{low}}, s_{\text{high}}$)



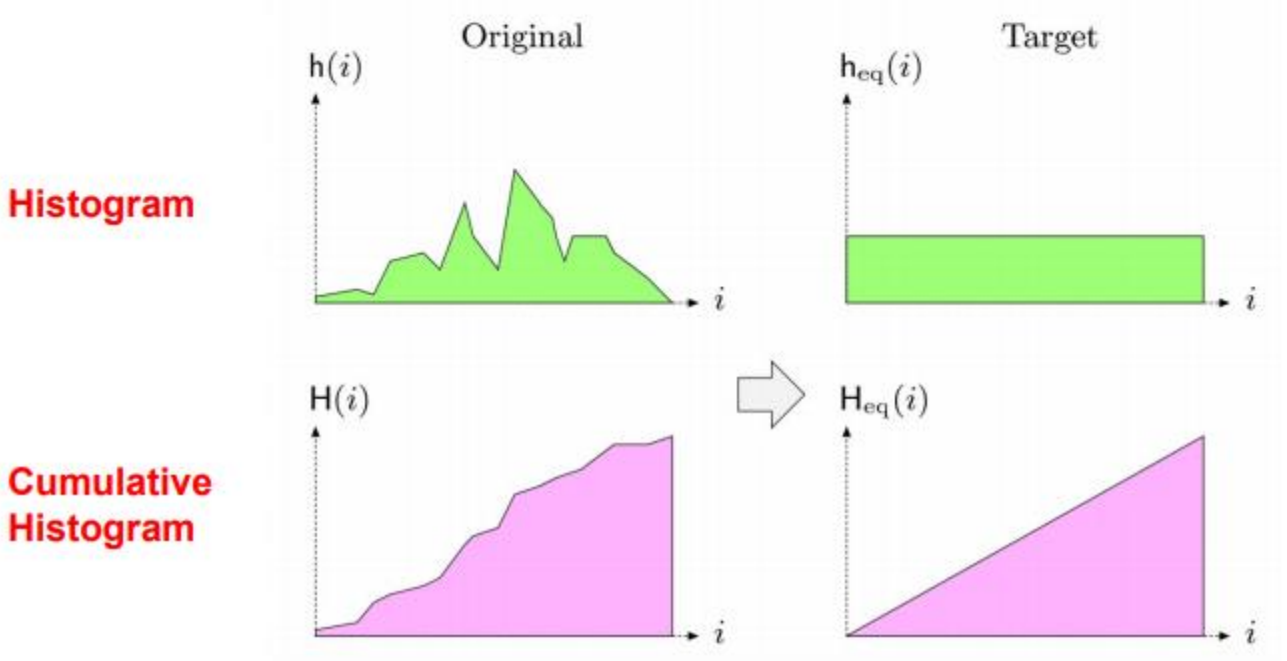
$$\hat{a}_{\text{low}} = \min\{\, i \mid \mathsf{H}(i) \geq M \cdot N \cdot s_{\text{low}} \}$$

$$\hat{a}_{\text{high}} = \max\{\, i \mid \mathsf{H}(i) \leq M \cdot N \cdot (1 - s_{\text{high}}) \}$$

$$f_{\text{mac}}(a) = \begin{cases} a_{\min} & \text{for } a \leq \hat{a}_{\text{low}} \\ a_{\min} + (a - \hat{a}_{\text{low}}) \cdot \dfrac{a_{\max} - a_{\min}}{\hat{a}_{\text{high}} - \hat{a}_{\text{low}}} & \text{for } \hat{a}_{\text{low}} < a < \hat{a}_{\text{high}} \\ a_{\max} & \text{for } a \geq \hat{a}_{\text{high}} \end{cases}$$

Histogram Equalization

- Adjust 2 different images to make their histograms (intensity distributions) similar
- Apply a point operation that changes histogram of modified image into **uniform distribution**



Histogram Equalization

Spreading out the frequencies in an image (or equalizing the image) is a simple way to improve dark or washed out images

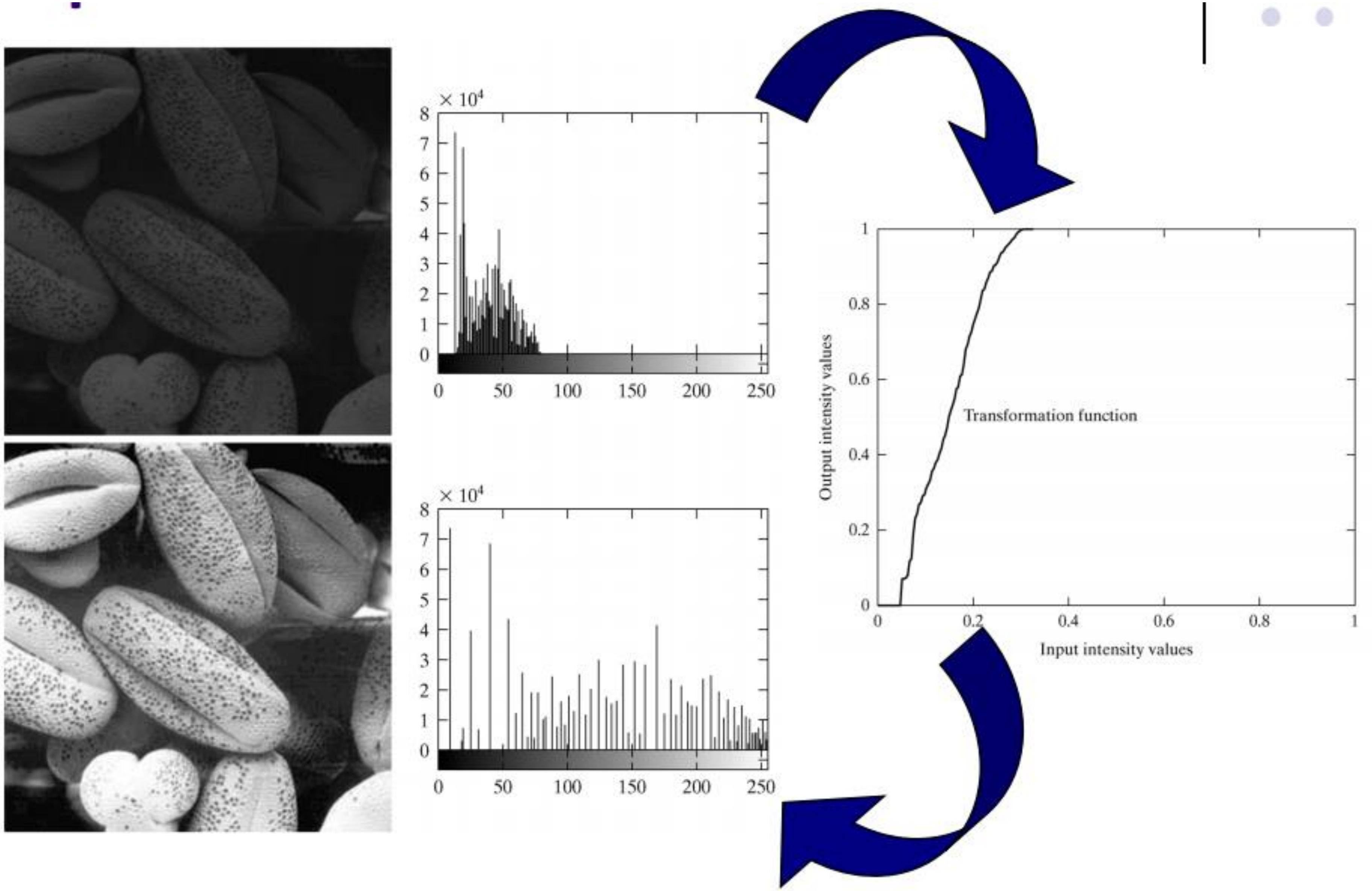
Can be expressed as a transformation of histogram

- r_k: input intensity
- s_k: processed intensity
- k: the intensity range (e.g 0.0 – 1.0)

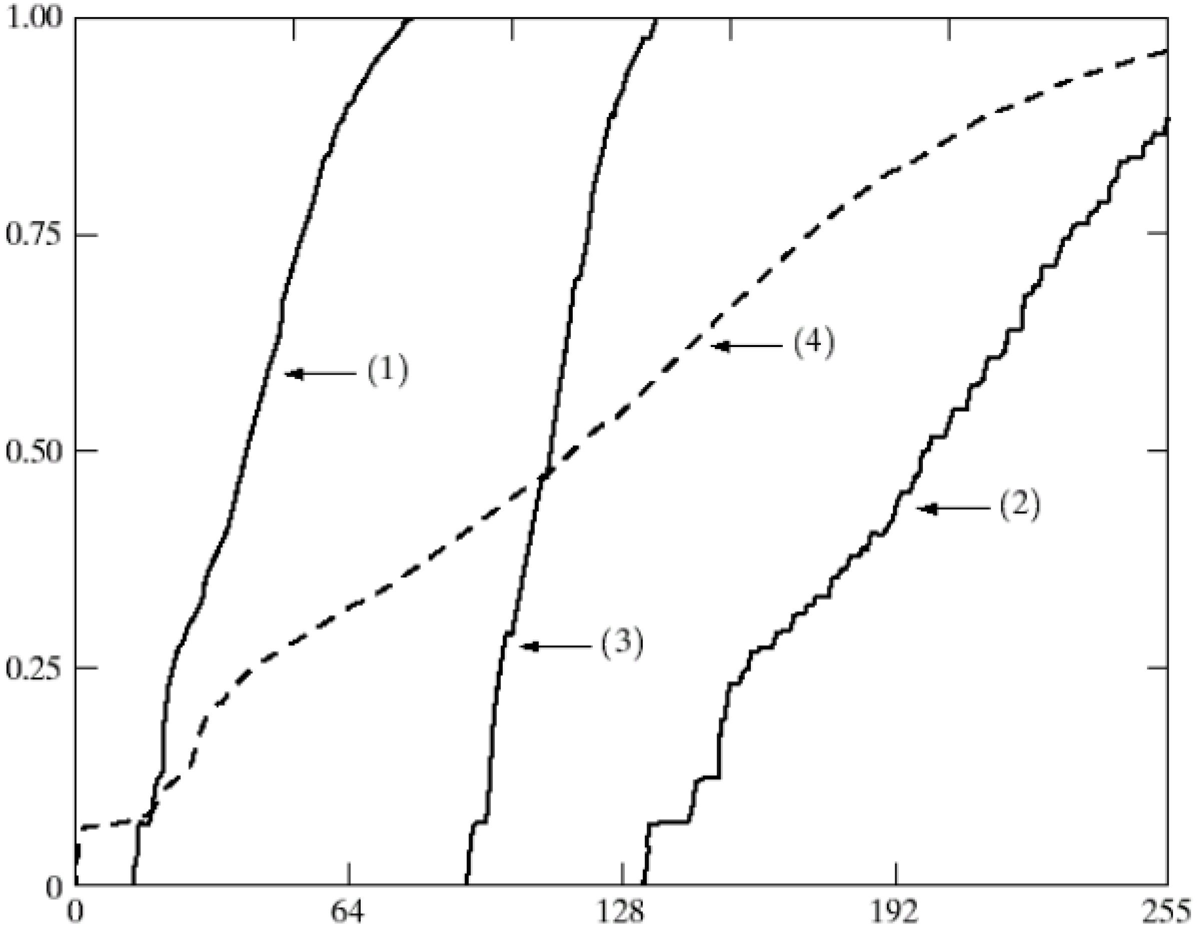
processed intensity → $s_k = T(r_k)$ ← input intensity

↑ Intensity range (e.g 0 – 255)

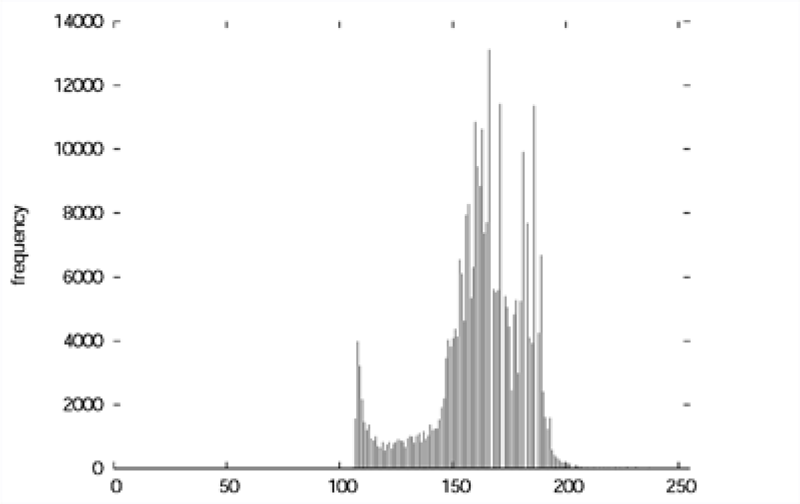
Images taken from Gonzalez & Woods, Digital Image Processing (2002)

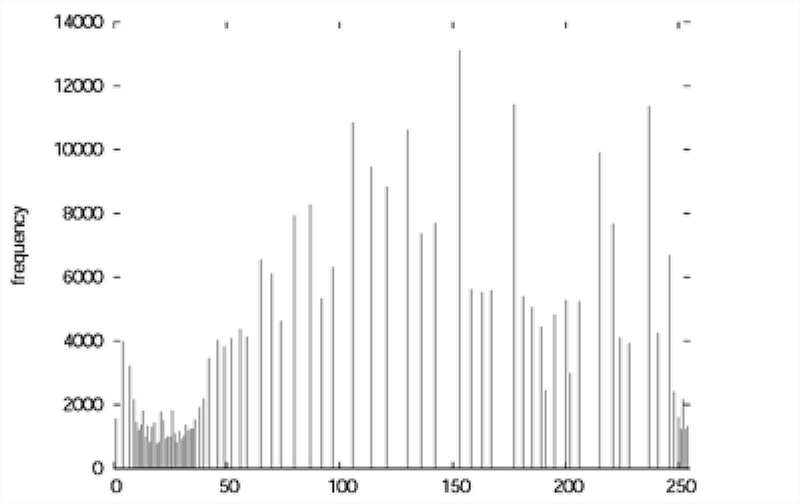


Different equalization function (1-4) may be used



Histogram Equalisation: an informal illustration



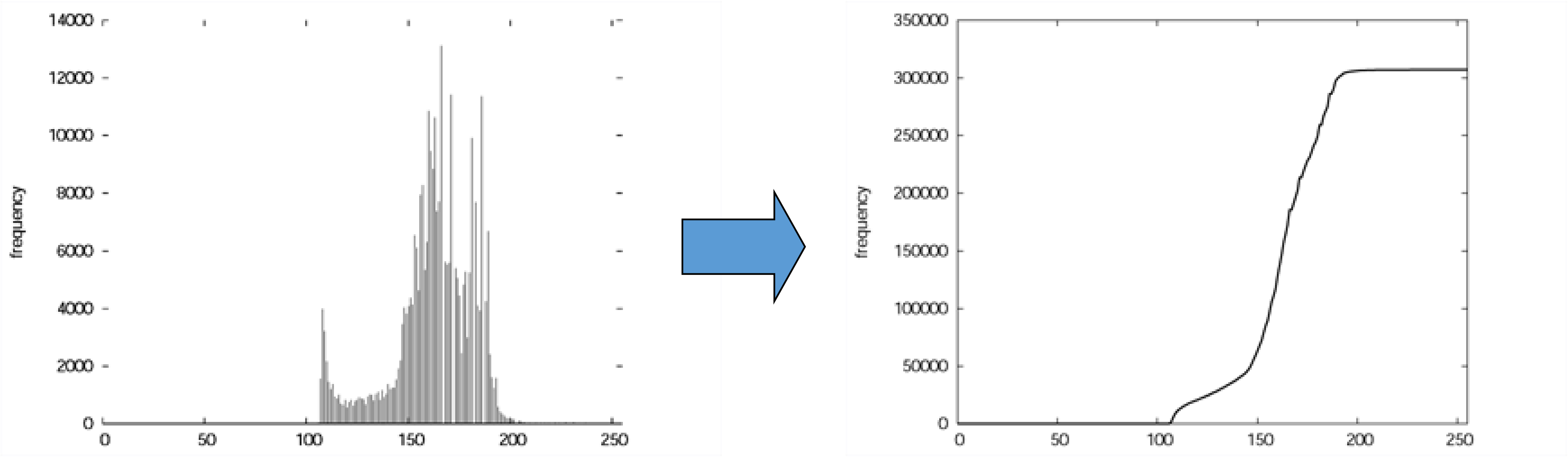


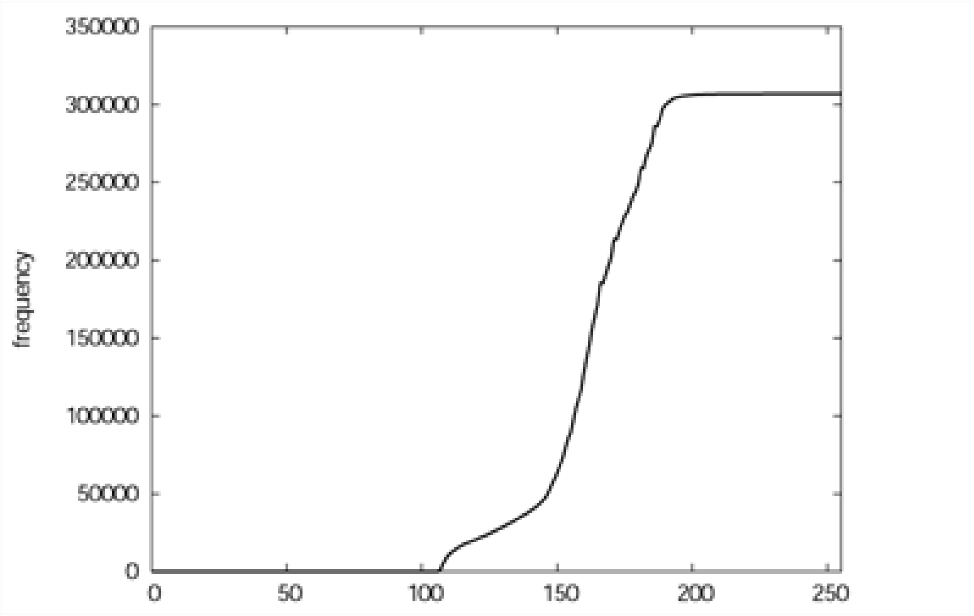
The goal in histogram equalisation is to expand the range of grey level values within the image to the entire 0-255 range

To do this we first calculate the *cumulative frequencies* for grey levels within the image

The cumulative frequency for grey level *g* is defined as the sum of the histogram data values from 0 to *g*.

We can graph the cumulative frequencies for our image:



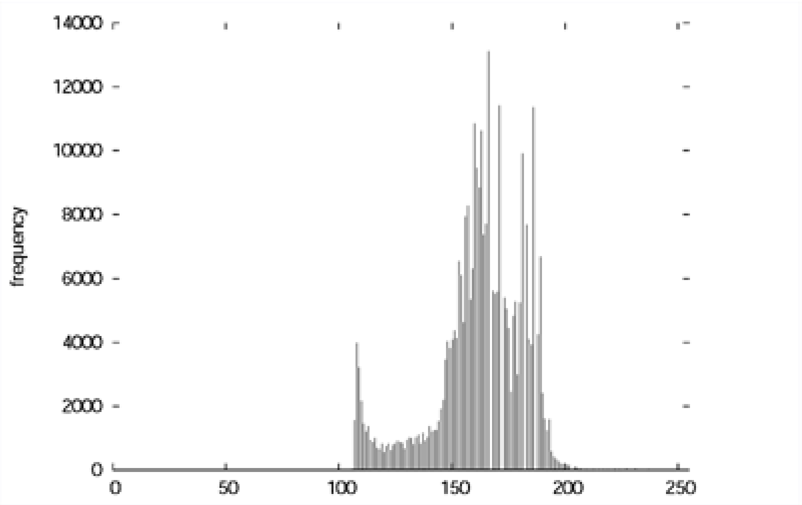


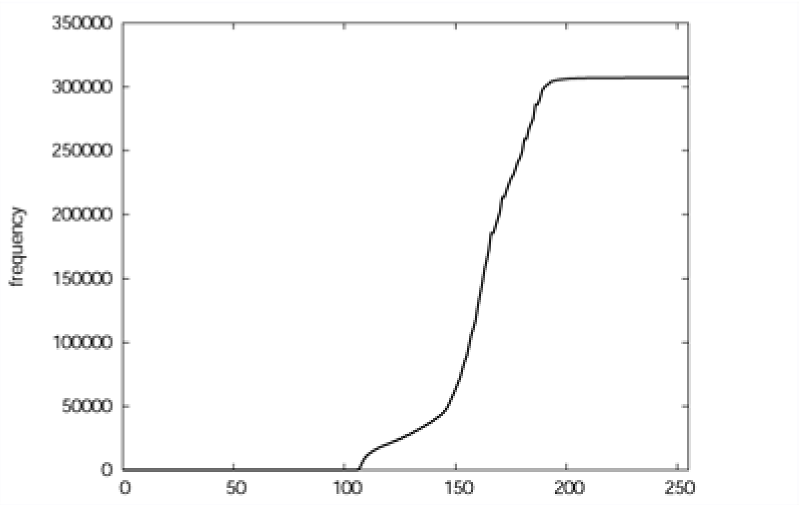
Use this information to redistribute the grey levels across the entire range.
The maximum of the cumulative frequency graph will always be equal to the number of pixels in the image (*numPixels*)

Original frequencies

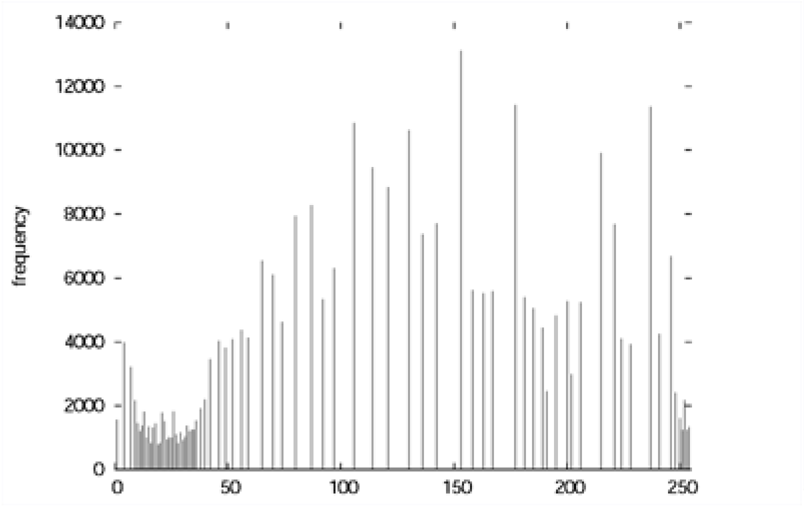
Cumulative frequencies

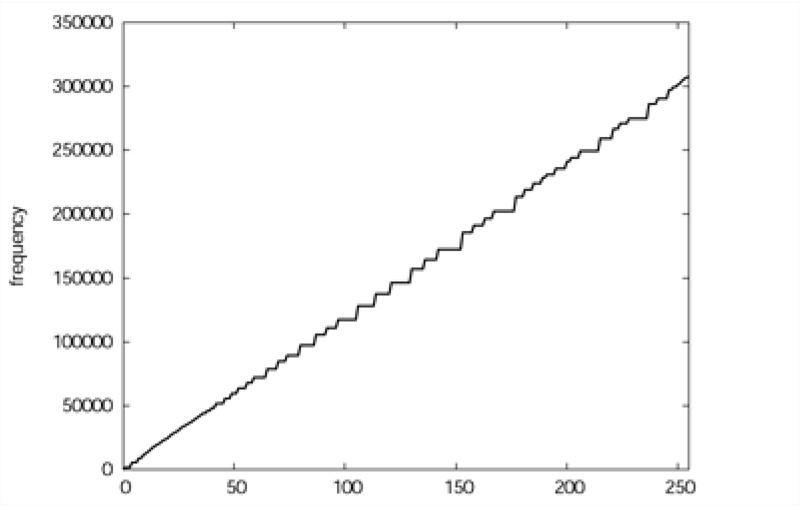
Original image





Equalised image





The same process can be applied to colour images by performing the process on the red, green and blue channels separately, as this image shows:

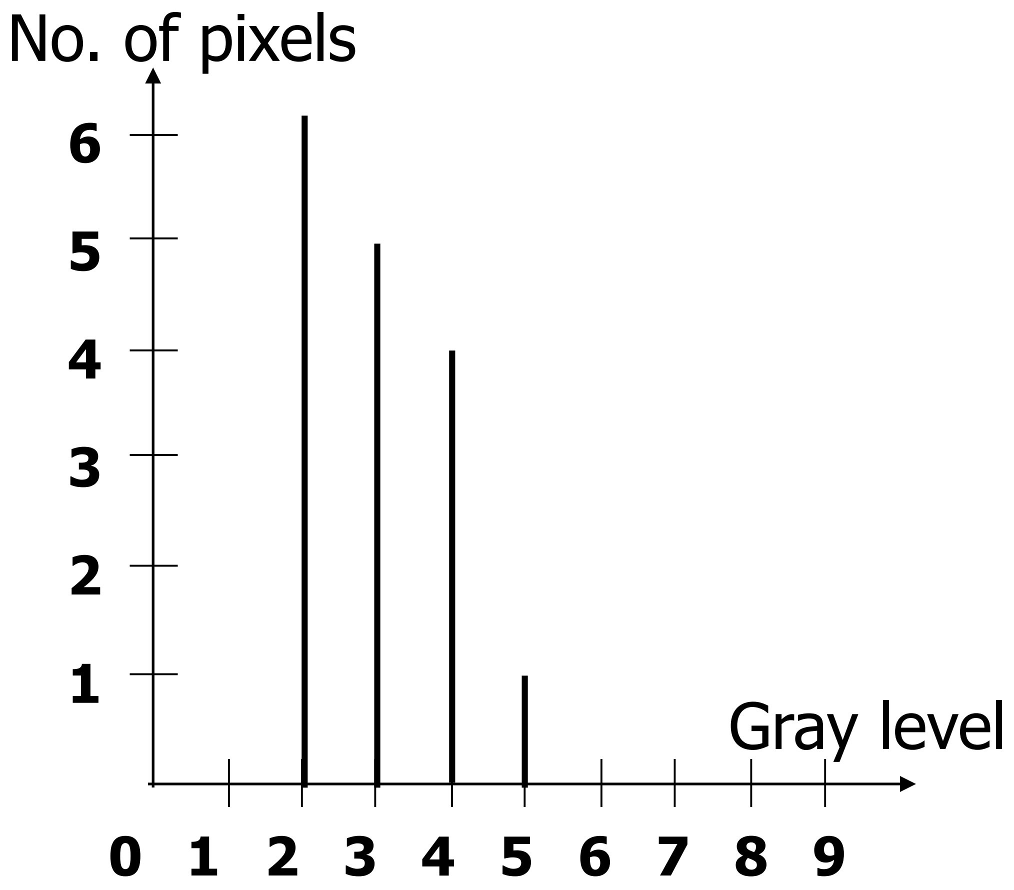
BUT this is a crude approach which can hugely alter the image colours !!   Better methods should be used.

Example

2	3	3	2
4	2	4	3
3	2	3	5
2	4	2	4

4x4 image

Gray scale = [0,9]



histogram

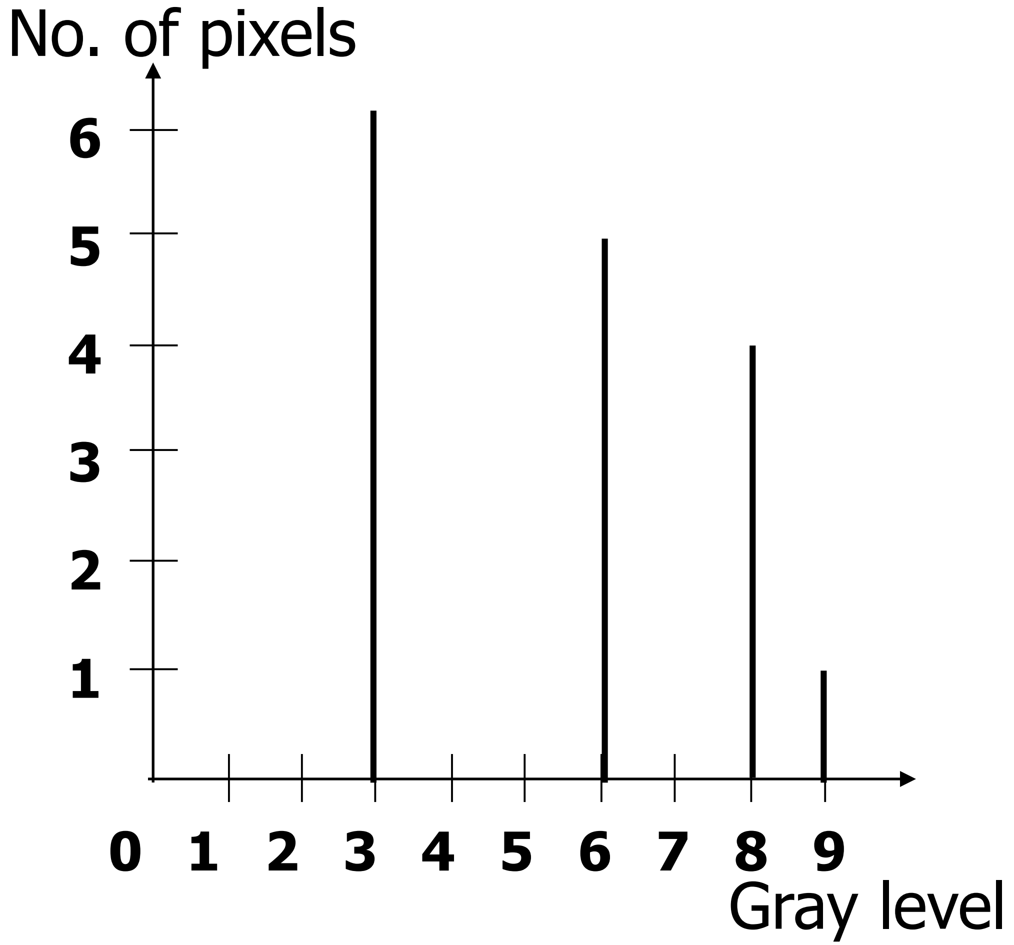
Gray Level(j)	0	1	2	3	4	5	6	7	8	9
No. of pixels	0	0	6	5	4	1	0	0	0	0
$\sum_{j=0}^{k} n_j$	0	0	6	11	15	16	16	16	16	16
$\sum_{j=0}^{k} \frac{n_j}{n}$	0	0	6 / 16	11 / 16	15 / 16	16 / 16	16 / 16	16 / 16	16 / 16	16 / 16
s* x *(L-1)	0	0	3.3 ≈3	6.1 ≈6	8.4 ≈8	9	9	9	9	9

Example

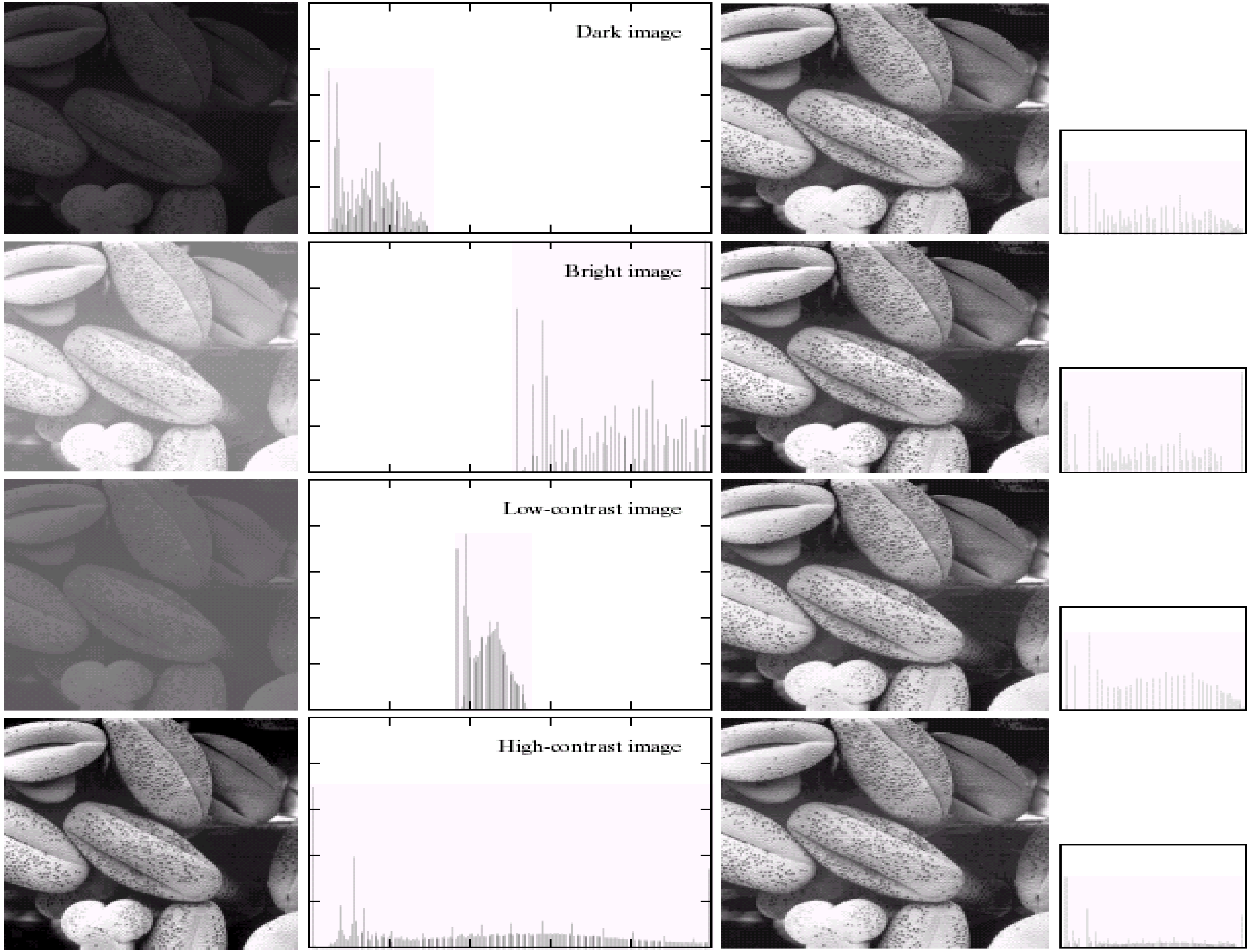
3	6	6	3
8	3	8	6
6	3	6	9
3	8	3	8

Output image

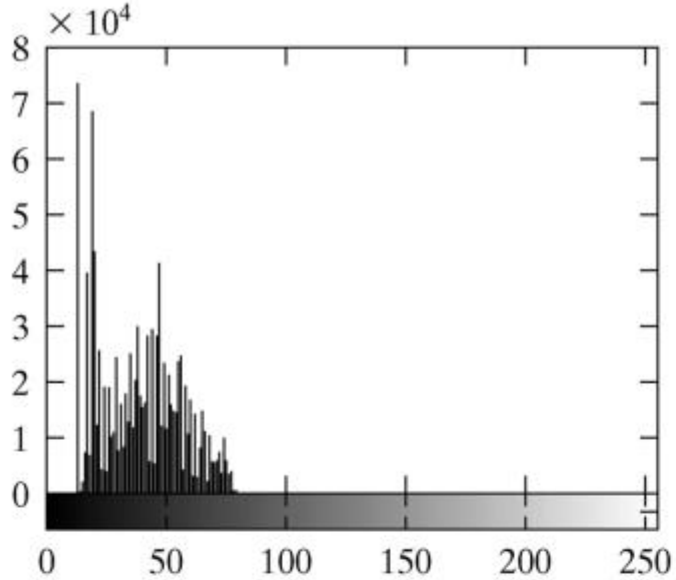
Gray scale = [0,9]

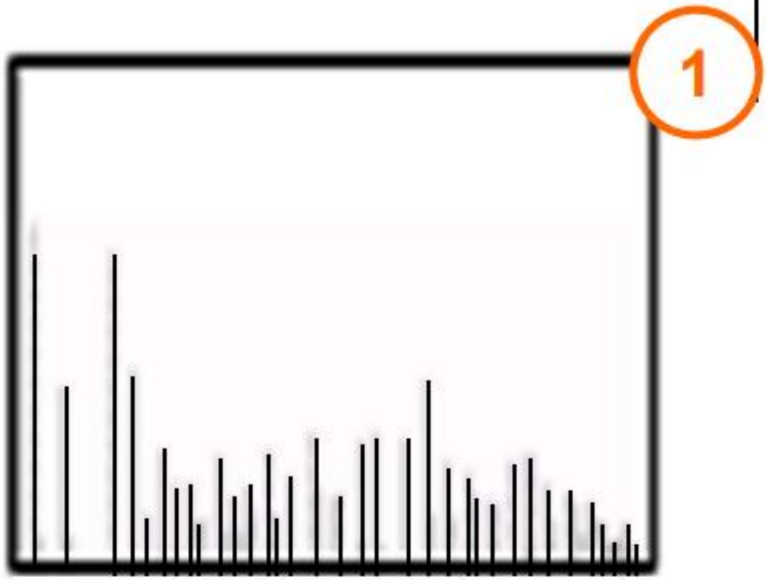


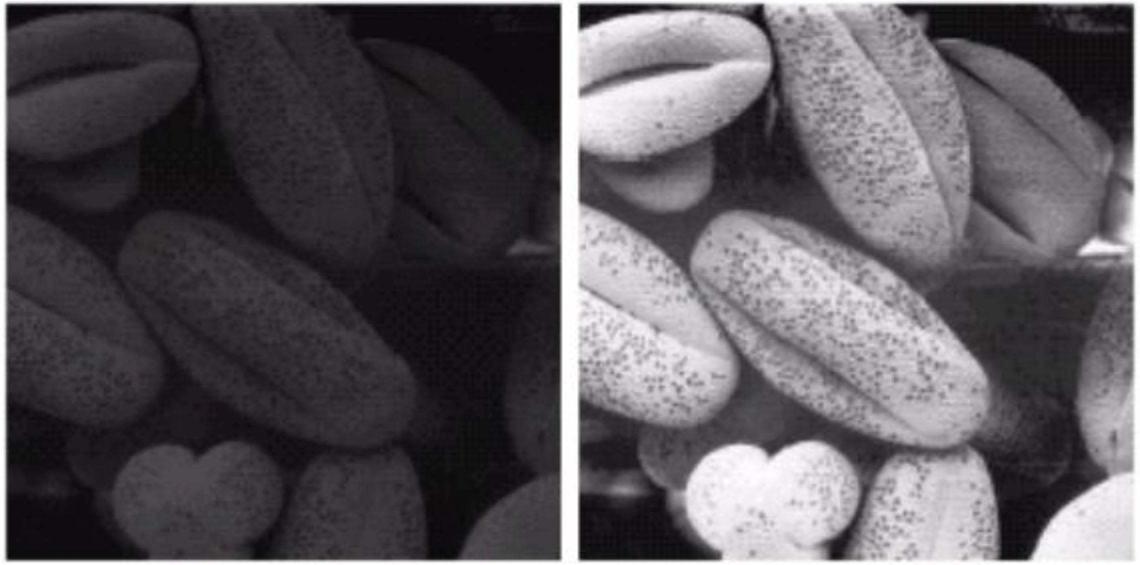
Histogram equalization

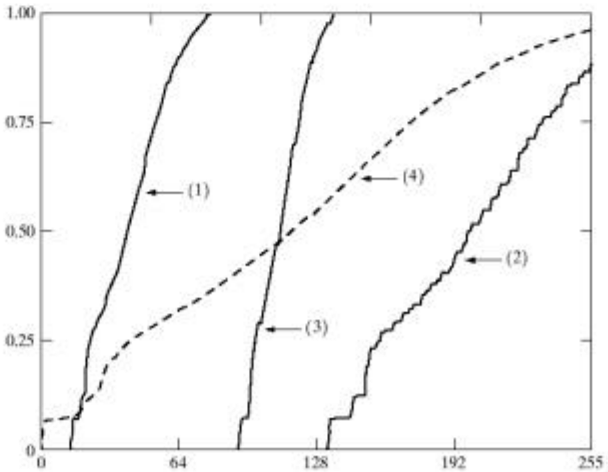

Dark image
Bright image
Low-contrast image
High-contrast image

Equalization examples



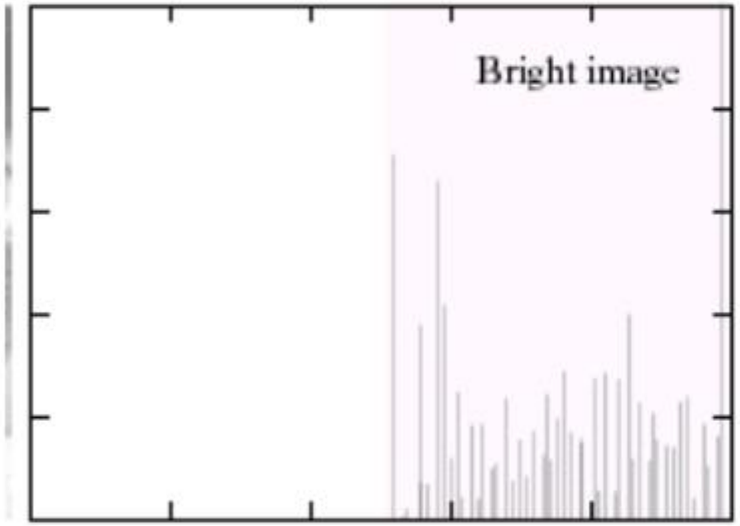


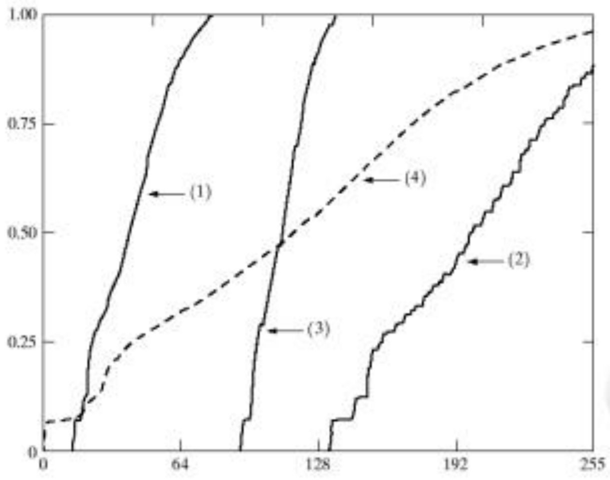




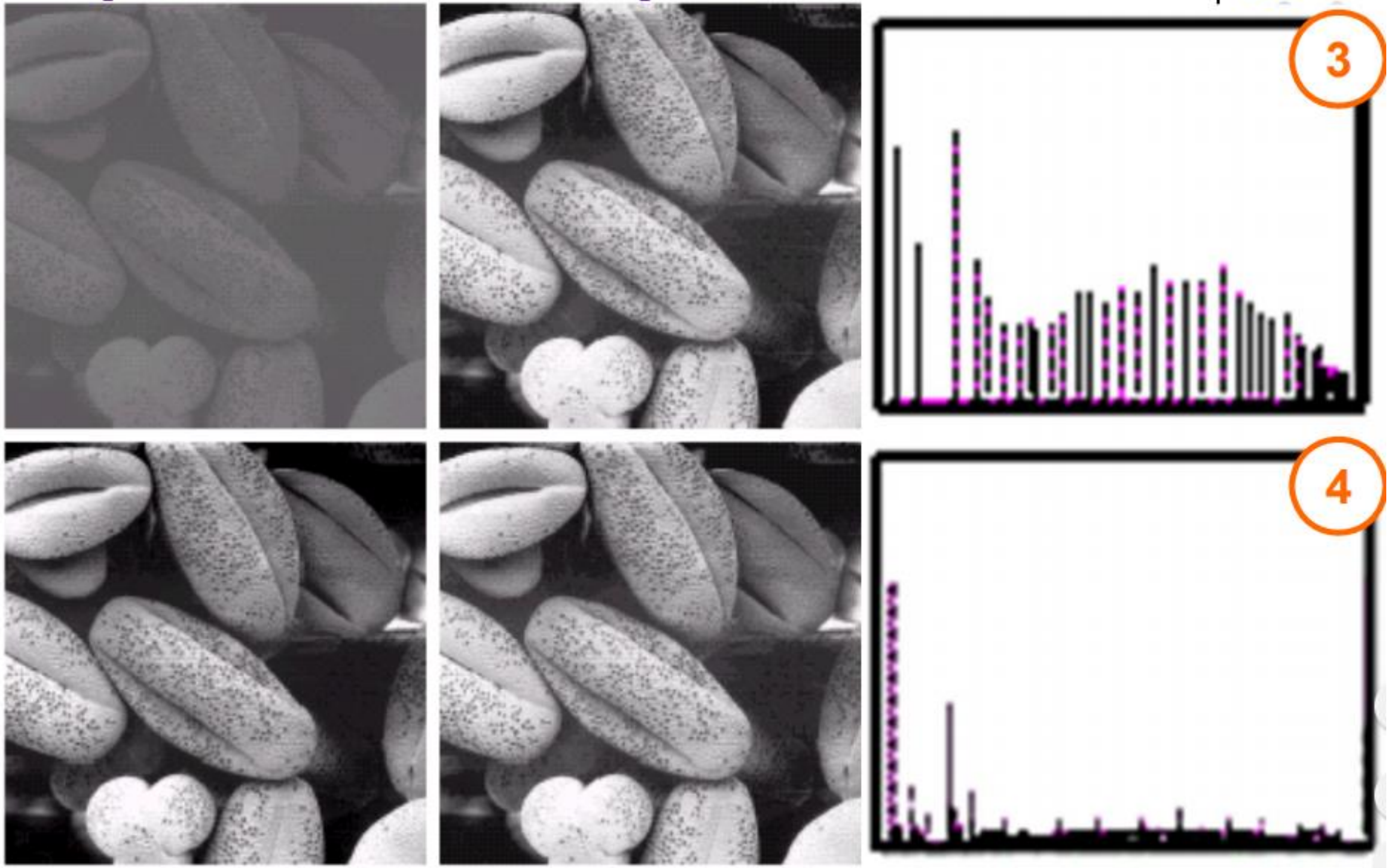
Images taken from Gonzalez & Woods, Digital Image Processing (2002)

2





Images taken from Gonzalez & Woods, Digital Image Processing (2002)



Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Lin

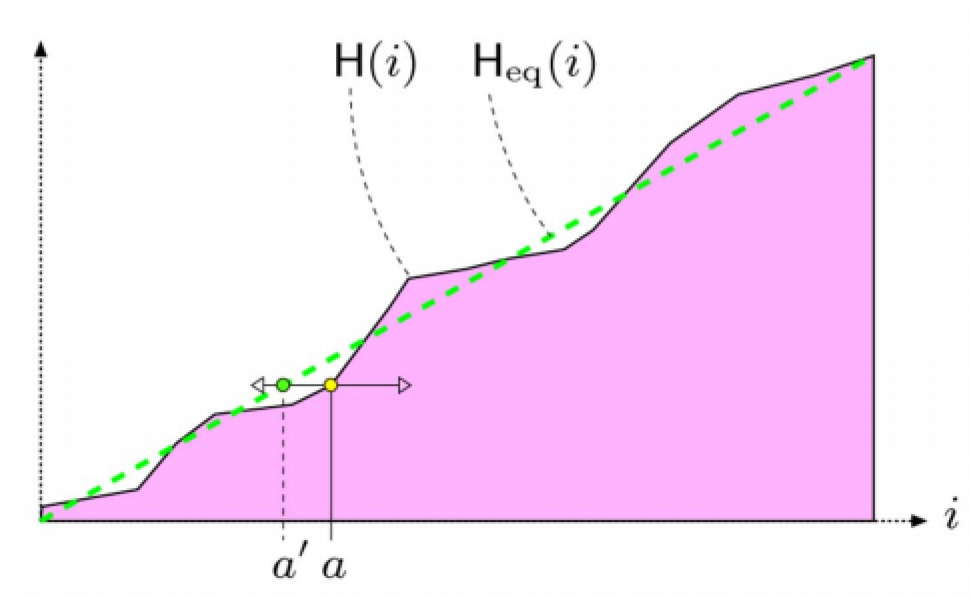
- Histogram cannot be made exactly flat – peaks cannot be increased or decreased by point operations.
- Following point operation makes histogram as flat as possible:

(assuming *M x N* image and pixels in range [0, *K* - 1])

Point operation that returns Linear equalized value of *a*

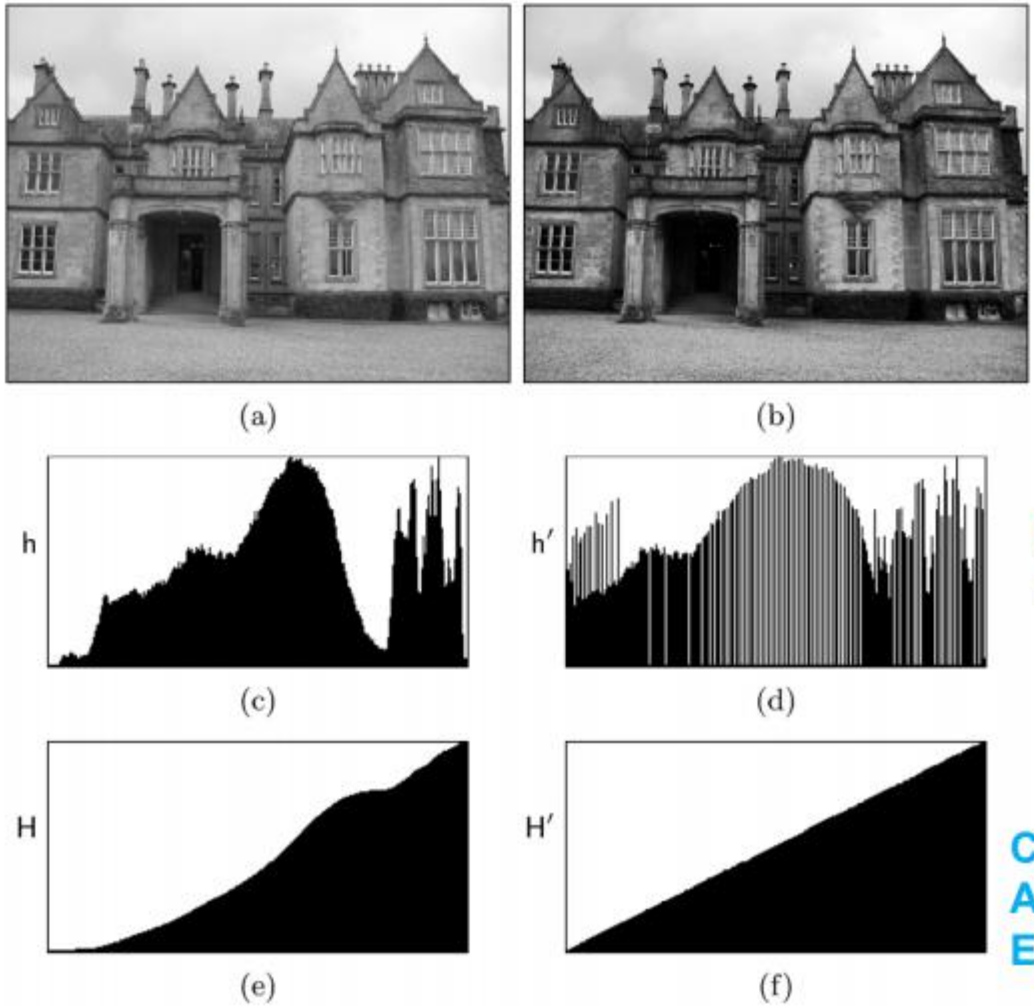
$$f_{eq}(a) = \left\lfloor \mathsf{H}(a) \cdot \frac{K-1}{MN} \right\rfloor$$

Cumulative Histogram: Σ how many times intensity a occurs



Original Image *I*

Image *I'* after Linear Equalization



Original histogram

Histogram after Linear Equalization

Cumulative Histogram

Cumulative Histogram After Linear Equalization

Histogram Specification

- Real images never show uniform distribution (unnatural)
- Most real images, distribution of pixel intensities is gaussian
- **Histogram specification**
  - modifies an image's histogram into an arbitrary intensity distribution (may not be uniform)
- Image 1's histogram can also be used as target for image 2
  - Why? Makes images taken by 2 different cameras to appear as if taken by same camera

Histograms and Probability

Histograms can be interpreted as probabilities.

Question: If I pick a pixel from an image at random, what is the probability that the pixel has intensity i?

Answer: $P(I(u,v) = i) = \frac{\text{\# pixels with value } i}{\text{total \# pixels}}$

Or, in terms of the histogram h:

$$P(I(u,v) = i) = h(i)/wh$$

Histogram Specification

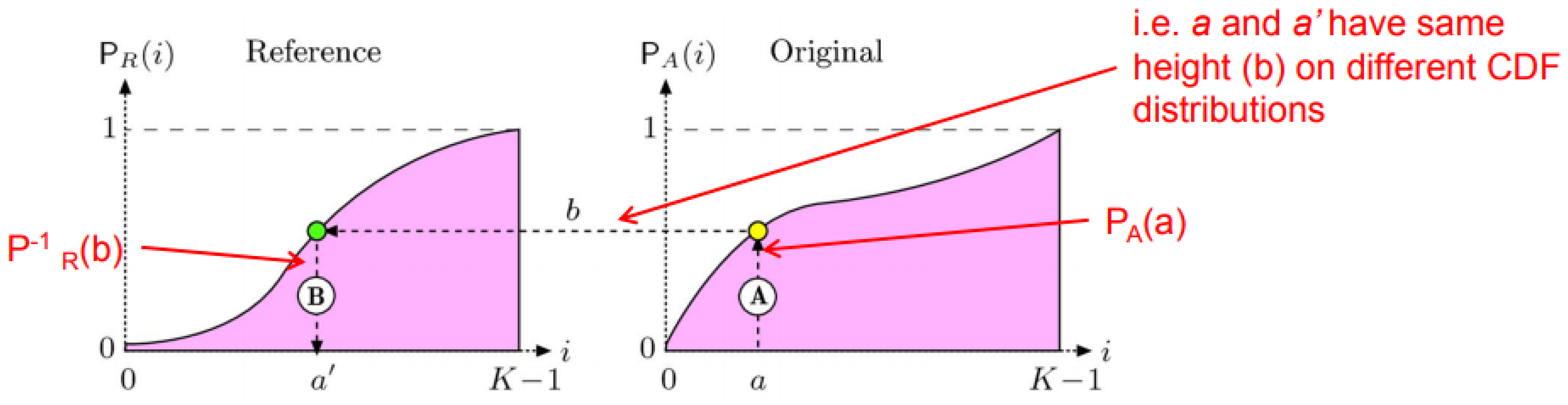
- Find a mapping such that distribution of *a* matches some reference distribution.i.e

$$a' = f_{\text{hs}}(a)$$

Mapping function: maps distribution on right to equivalent point (same height) On distribution on left

to convert original image I_A *into* $I_{A'}$ such that

$$\mathsf{P}_{A'}(i) \approx \mathsf{P}_R(i) \qquad \text{for } 0 \le i < K$$

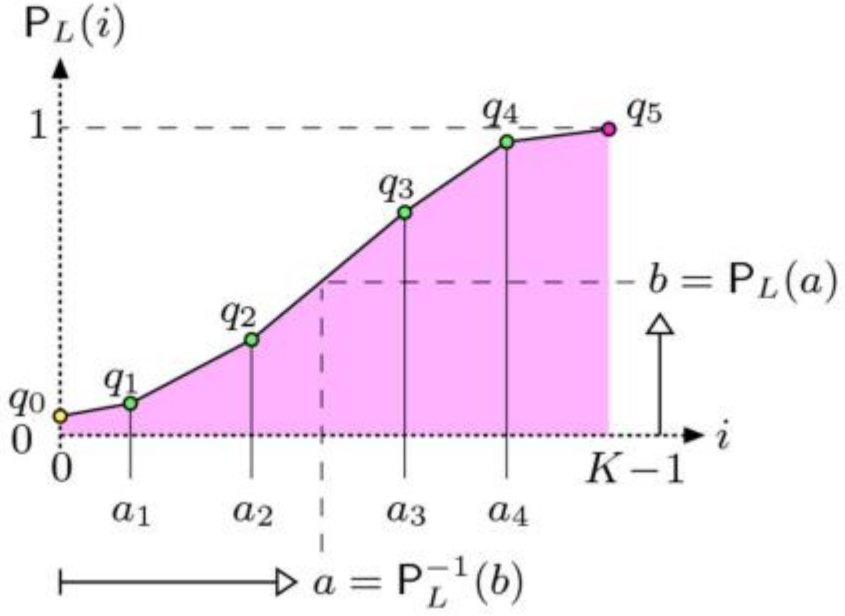


$$f_{\text{hs}}(a) = a' = \mathsf{P}_R^{-1}(\mathsf{P}_A(a))$$

Adjusting Linear Distribution Piecewise

- In practice, reference distribution may be specified as a *piecewise linear* function

$$\mathcal{L} = [\langle a_0, q_0 \rangle, \langle a_1, q_1 \rangle, \ldots \langle a_k, q_k \rangle, \ldots \langle a_N, q_N \rangle]$$

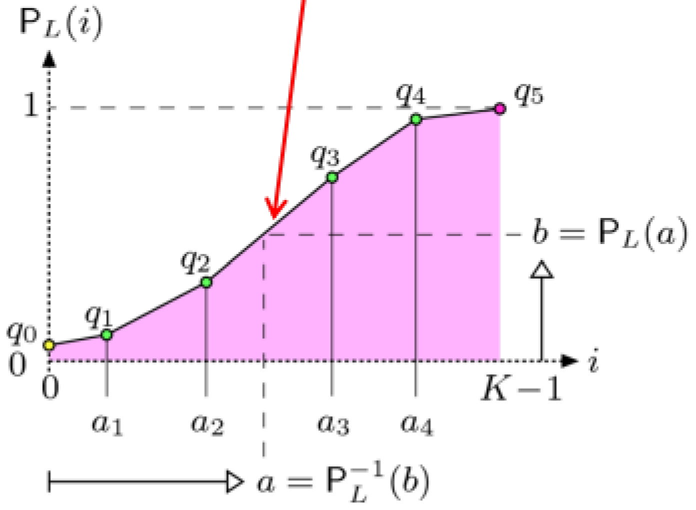


- 2 endpoints are fixed $\langle 0, q_0 \rangle$ and $\langle K-1, 1 \rangle$

Adjusting Linear Distribution Piecewise

For each segment, linearly
Interpolate to find any value

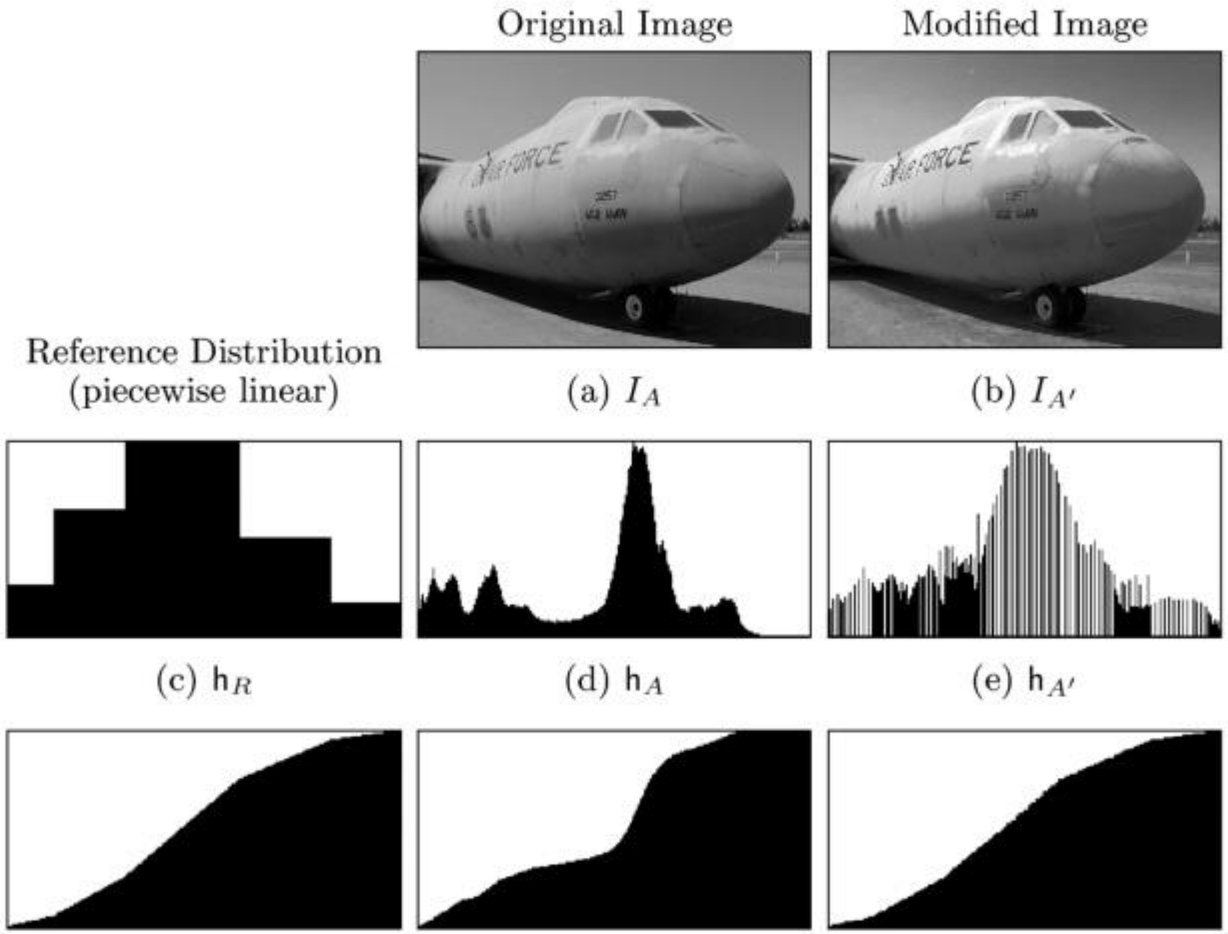
$$\mathsf{P}_L(i) = \begin{cases} q_m + (i-a_m)\cdot\dfrac{(q_{m+1}-q_m)}{(a_{m+1}-a_m)} & \text{for } 0 \le i < K-1 \\ 1 & \text{for } i = K-1 \end{cases}$$



$$\mathsf{P}_L^{-1}(b) = \begin{cases} 0 & \text{for } 0 \le b < \mathsf{P}_L(0) \\ a_n + (b-q_n)\cdot\dfrac{(a_{n+1}-a_n)}{(q_{n+1}-q_n)} & \text{for } \mathsf{P}_L(0) \le b < 1 \\ K-1 & \text{for } b \ge 1 \end{cases}$$

We also need the inverse mapping

$$n = \max\{j \in \{0,\ldots N-1\} \mid q_j \le b\}$$



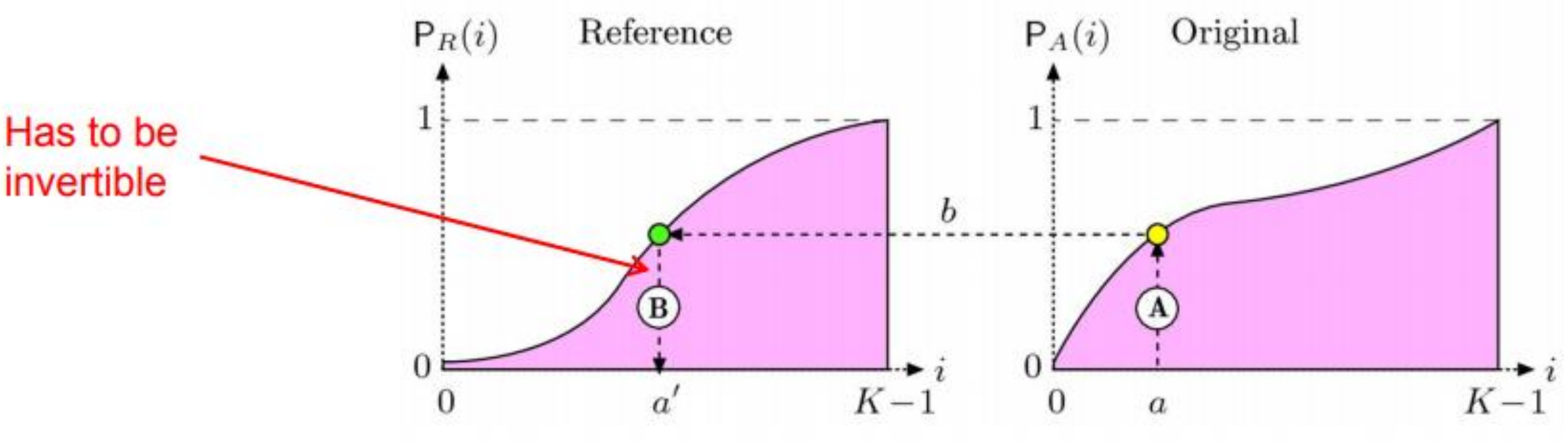
(a) I_A (b) $I_{A'}$

(c) h_R (d) h_A (e) $\mathsf{h}_{A'}$

(f) P_R (g) P_A (h) $\mathsf{P}_{A'}$

Histogram Matching

- Prior method needed reference distribution to be invertible



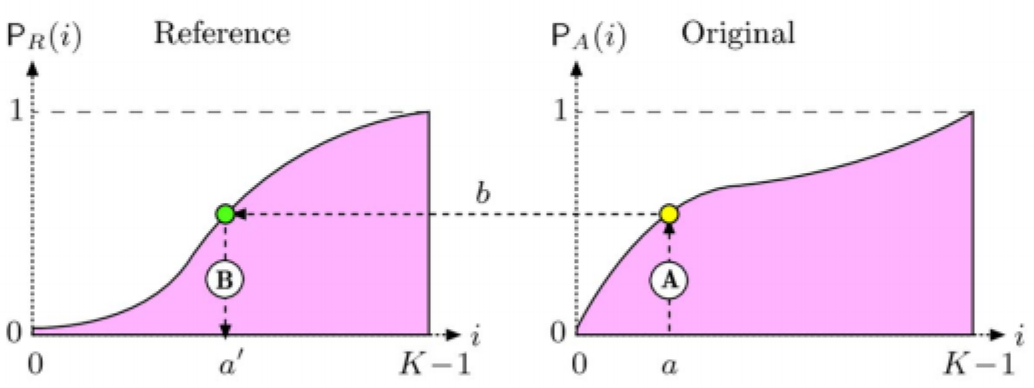
$$f_{\mathrm{hs}}(a) = a' = \mathsf{P}_R^{-1}(\mathsf{P}_A(a))$$

- What if reference histogram is not invertible?
- For example not invertible if histogram has some intensities that occur with probability 0? i.e. p(k) = 0
- Use different method called **histogram matching**

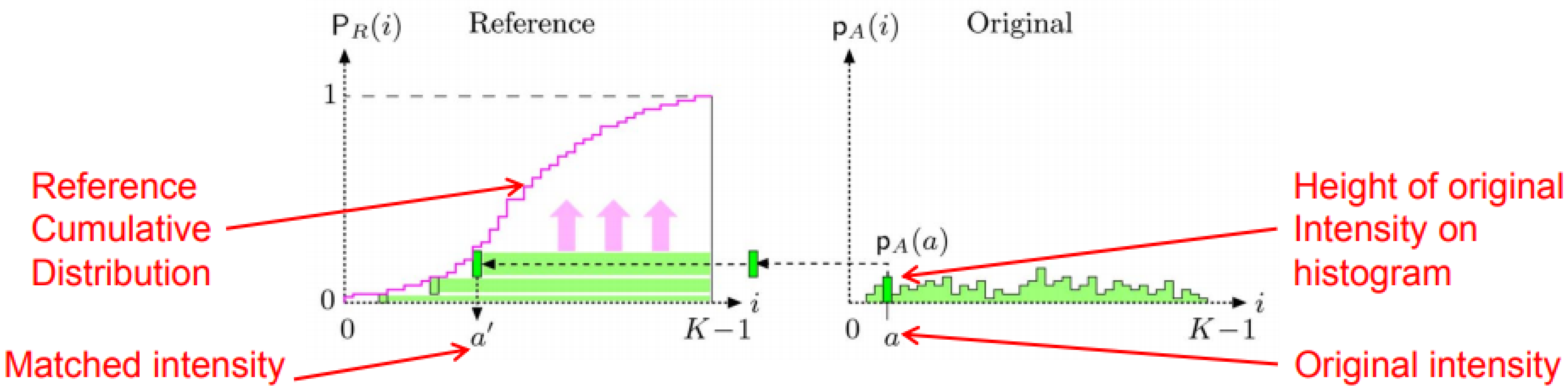
Histogram Matching

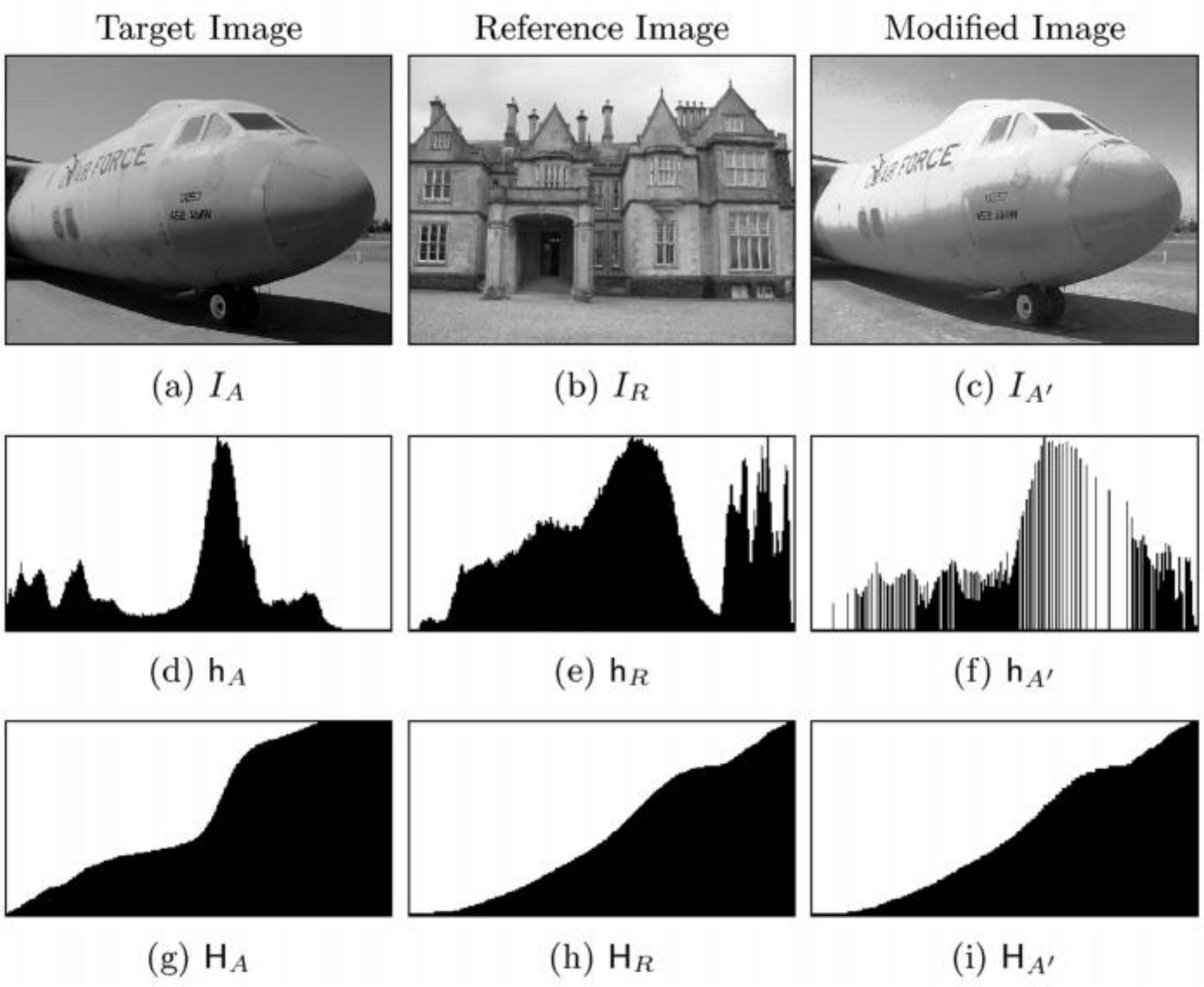
- Given two images I_A and I_B, we want to make their intensity profiles look as similar as possible.
- How?
  - "Match" their cumulative histograms H_A and H_B
- Works well for images with similar content.
- Looks bad for images with different content.

Adjusting to a given histogram



$$f_{\text{hs}}(a) = a' = \min\{\, j \mid (0 \le j < K) \wedge \big(P_A(a) \le P_R(j)\big)\}$$

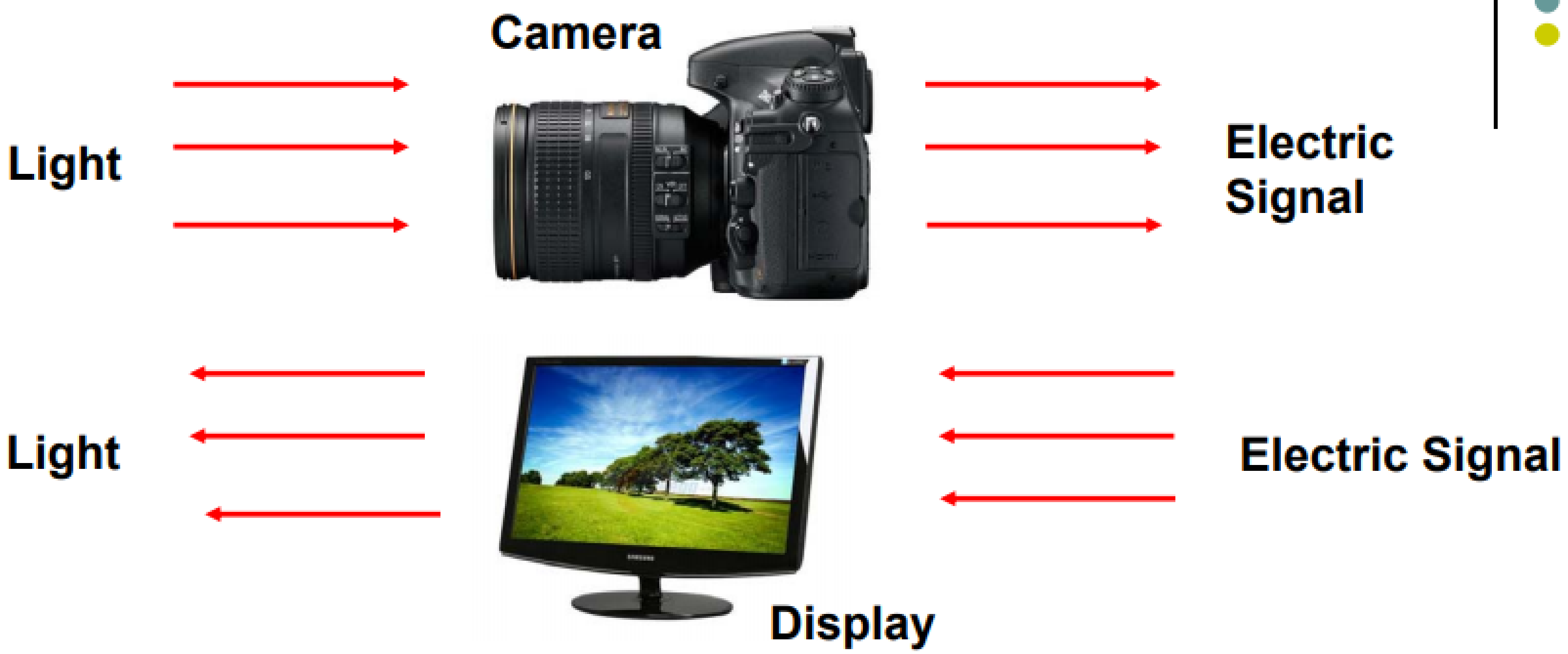



Target Image
Reference Image
Modified Image
(a) I_A
(b) I_R
(c) $I_{A'}$
(d) h_A
(e) h_R
(f) $h_{A'}$
(g) H_A
(h) H_R
(i) $H_{A'}$

ram

ıal
ter

Gamma Correction



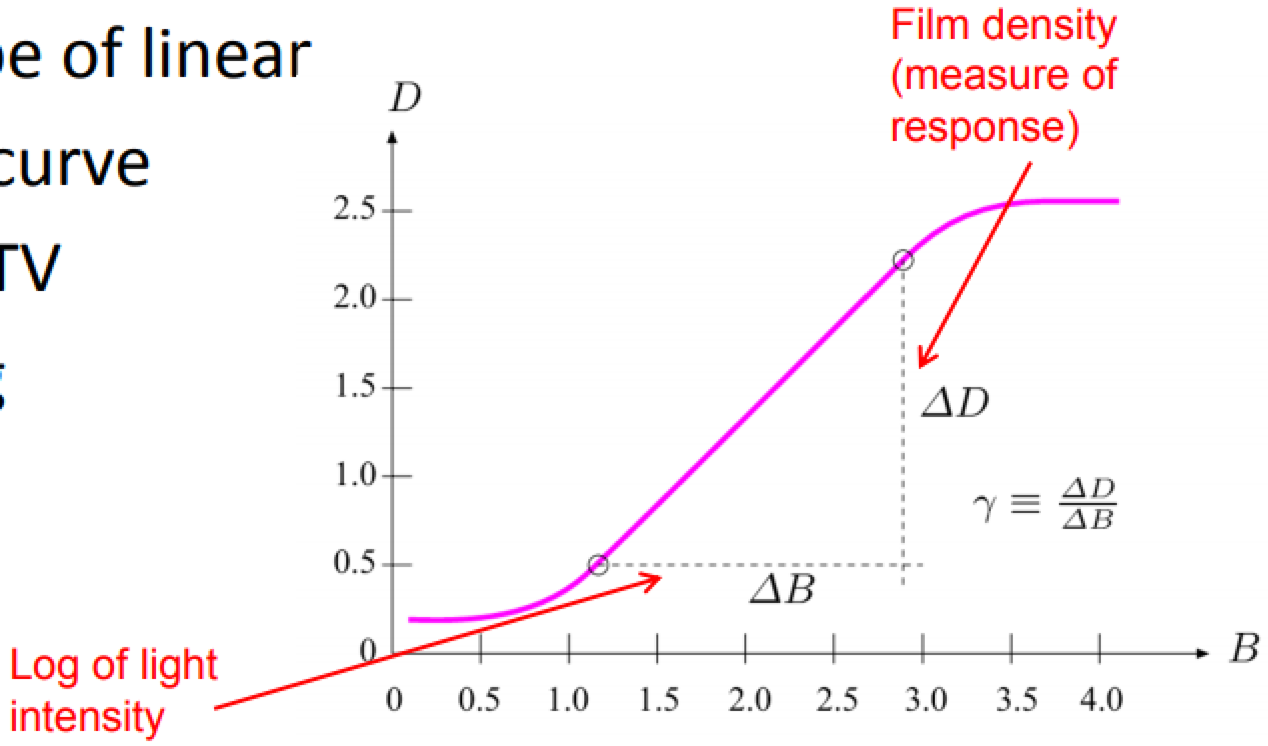
- Different camera sensors
  - Have different responses to light intensity
  - Produce different electrical signals for same input
- How do we ensure there is consistency in:
  a) Images recorded by different cameras for given light input
  b) Light emitted by different display devices for same image?

Gamma Correction

- What is the relation between:
  - **Camera:** Light on sensor vs. **"intensity"** of corresponding pixel
  - **Display:** Pixel intensity vs. light from that pixel
- Relation between pixel value and corresponding physical quantity is usually complex, nonlinear
- An approximation ?

What is Gamma?

- Originates from analog photography
- **Exposure function:** relationship between:
  - logarithmic light intensity vs. resulting film density.
- **Gamma:** slope of linear range of the curve
- The same in TV broadcasting



What is Gamma?

- **Gamma function:** a good approximation of exposure curve
- Inverse of a Gamma function is another gamma function with

$$\bar{\gamma} = 1/\gamma$$

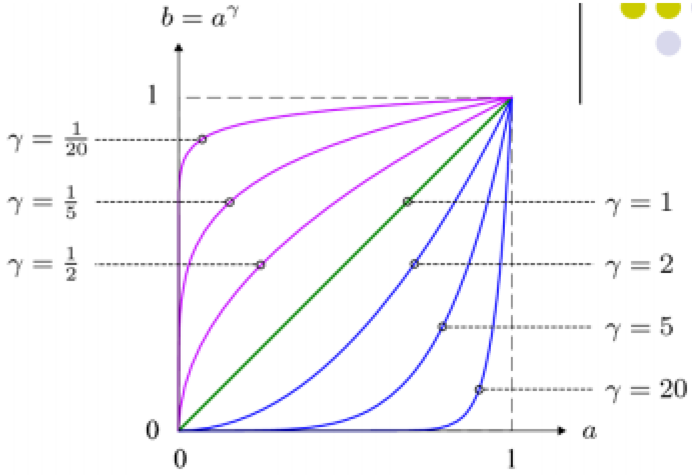
- Gamma of CRT and LCD monitors:
- 1.8-2.8 (typically 2.4)

$$s = B^{\gamma_c}$$

Output signal
Raised by gamma

$$b = s^{1/\gamma_c} = (B^{\gamma_c})^{1/\gamma_c} = B^{(\gamma_c \frac{1}{\gamma_c})} = B^1$$

Correct output signal
By dividing by 1/ gamma
(called Gamma correction)



$$b = f_{\gamma}(a) = a^{\gamma} \quad \text{for } a \in \mathbb{R}, \gamma > 0$$

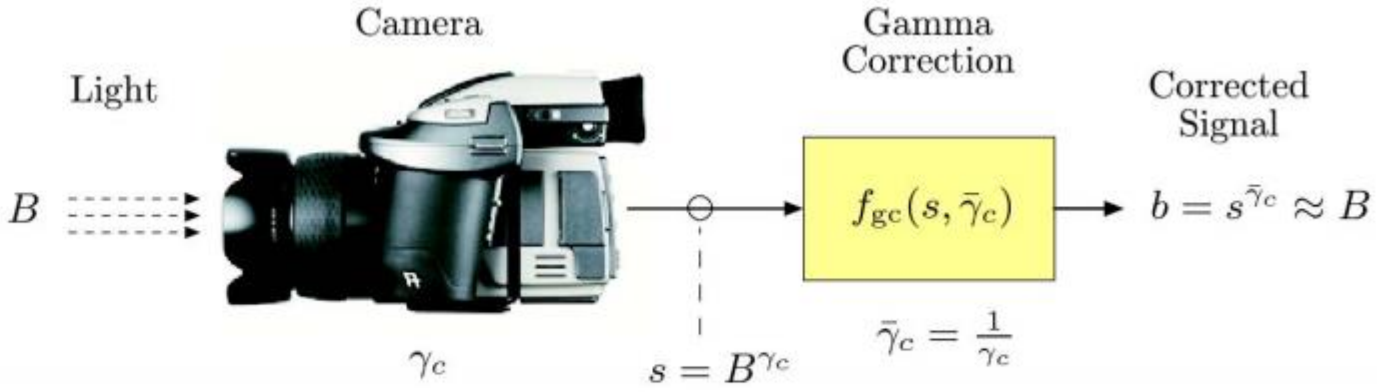
$$a = f_{\gamma}^{-1}(b) = b^{1/\gamma}$$

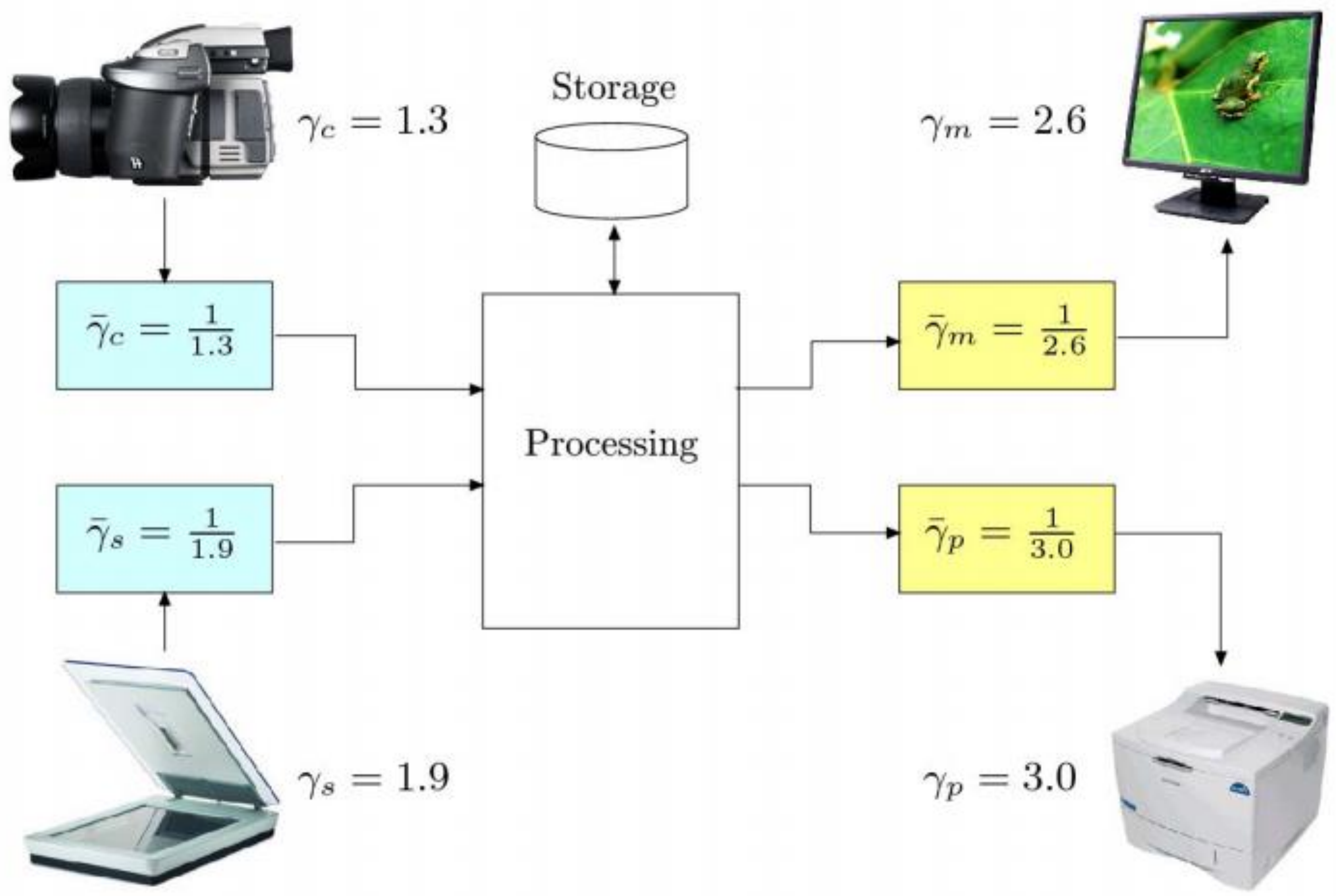
$$f_{\gamma}^{-1}(b) = f_{\bar{\gamma}}(b)$$

$$\bar{\gamma} = 1/\gamma$$

Gamma Correction

- Obtain a measurement **b** proportional to original light intensity **B** by applying inverse gamma function
- Gamma correction is important to achieve a device independent representation





$\gamma_c = 1.3$
Storage
$\gamma_m = 2.6$
$\bar{\gamma}_c = \frac{1}{1.3}$
$\bar{\gamma}_m = \frac{1}{2.6}$
Processing
$\bar{\gamma}_s = \frac{1}{1.9}$
$\bar{\gamma}_p = \frac{1}{3.0}$
$\gamma_s = 1.9$
$\gamma_p = 3.0$

Example: Alpha Blending

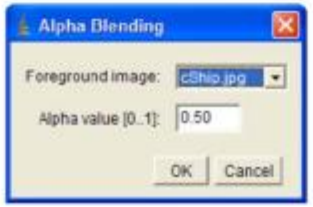
$$I'(u,v) \leftarrow \alpha \cdot I_{\mathrm{BG}}(u,v) + (1-\alpha) \cdot I_{\mathrm{FG}}(u,v)$$

I_{BG}

$\alpha = 0.25$

I_{FG}

$\alpha = 0.50$



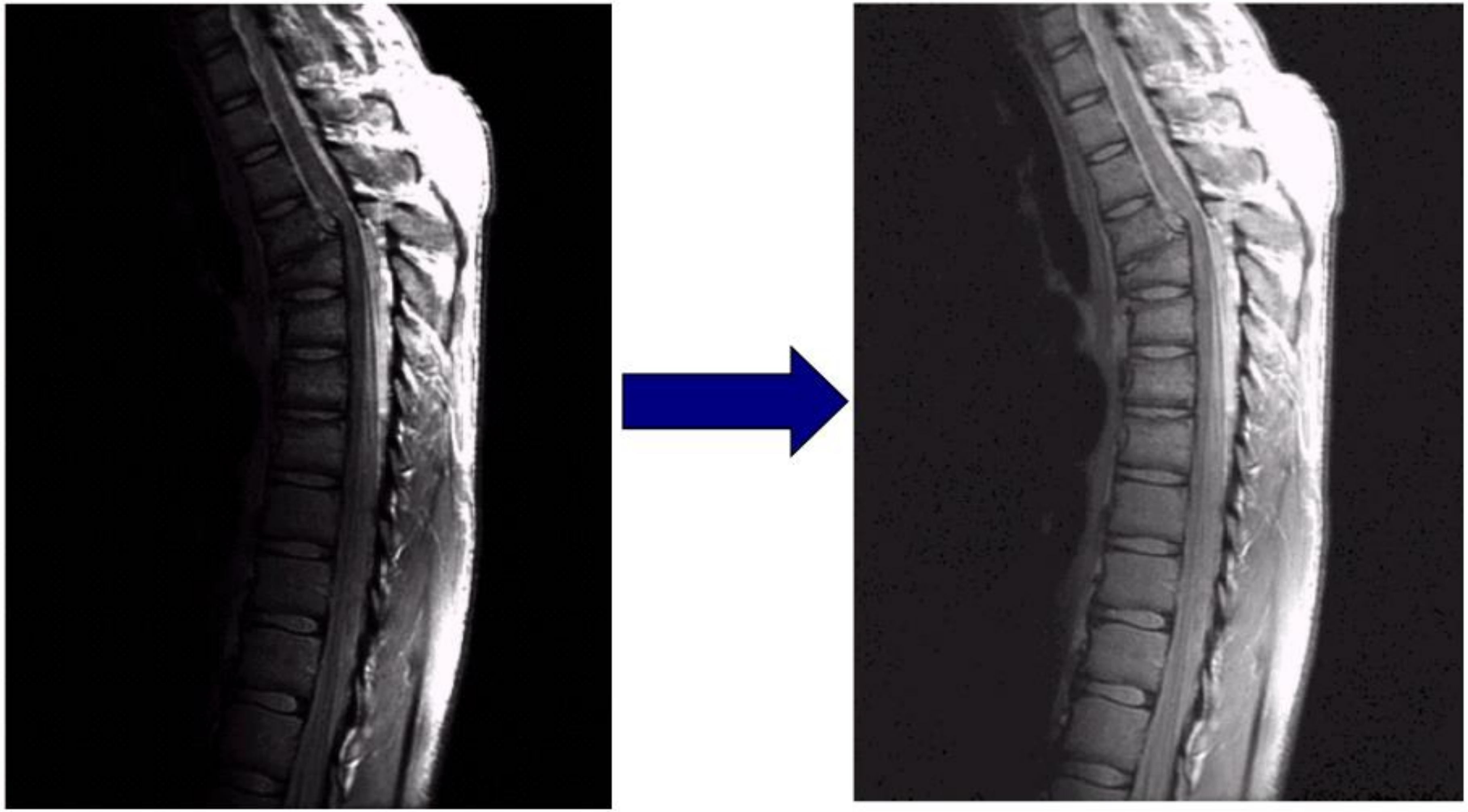
GenericDialog

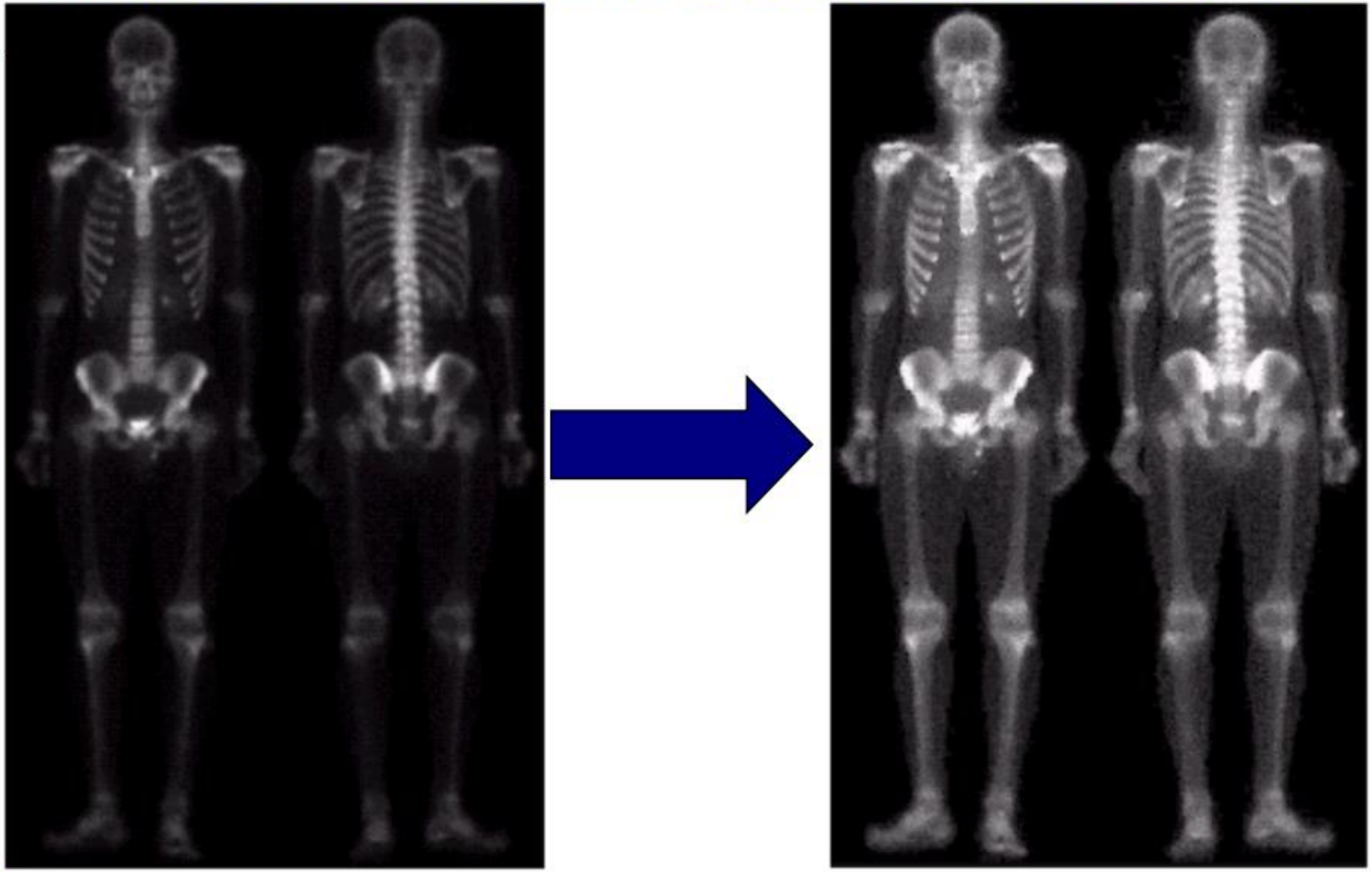
What is Image Enhancement

Image enhancement makes images more useful by:

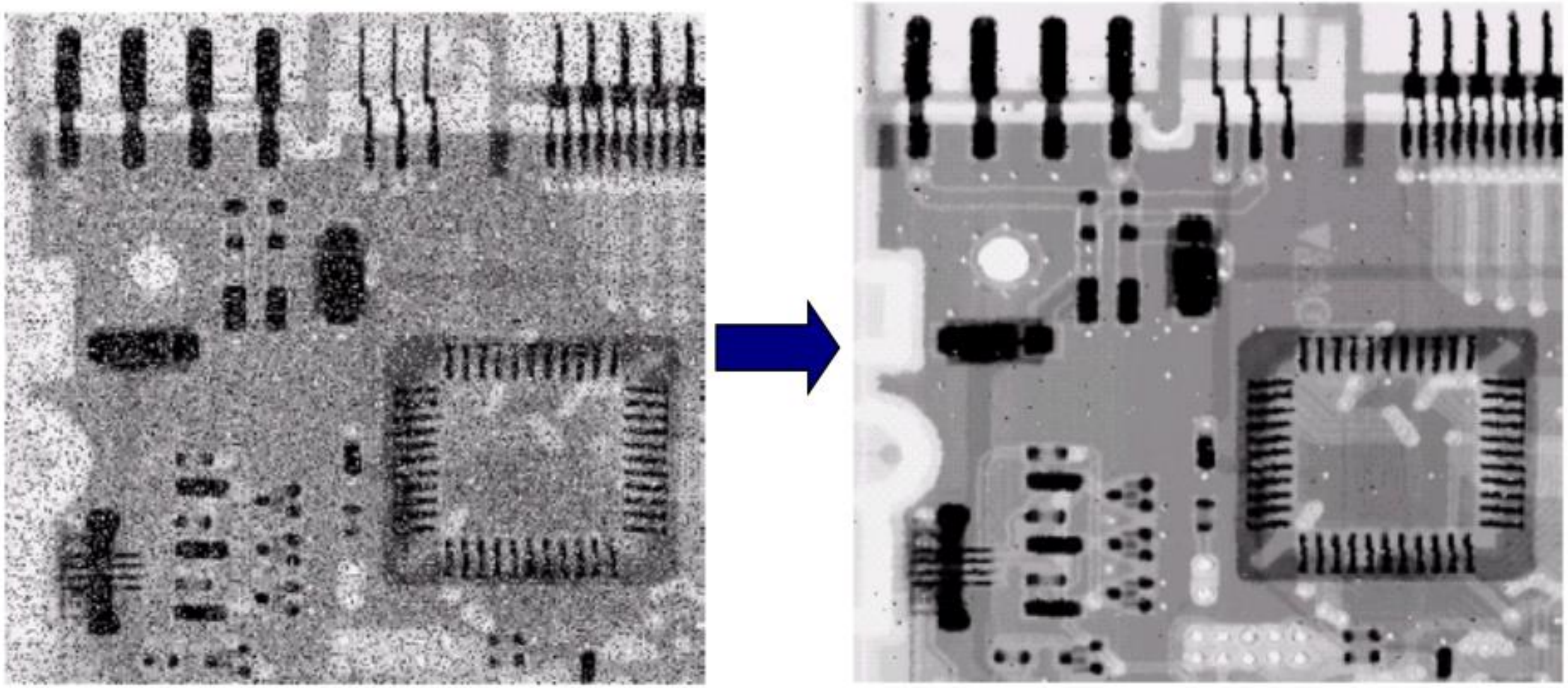
- Highlighting interesting detail in images
- Removing noise from images
- Making images more visually appealing

Images taken from Gonzalez & Woods, Digital Image Processing (2002)





Images taken from Gonzalez & Woods, Digital Image Processing (2002)



Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Images taken from Gonzalez & Woods, Digital Image Processing (2002)

What point operations can't do?

- Example: sharpening

→

What point operations can't do?

- Other cool artistic patterns by combining pixels