CPSC 4040/6040 Computer Graphics Images

Joshua Levine
levinej@clemson.edu

Lecture 11
Histograms and Filtering

Sept. 24, 2015

Slide Credits:
Ross T. Whitaker

Agenda

- Q02 Assigned
- PA03 Questions?

Last Time

Taxonomy

- Images can be represented in two domains
  - **Spatial Domain**: Represents light intensity at locations in space
  - **Frequency Domain**: Represents frequency amplitudes across a spectrum
- Operations can be classified as either per
  - **Point**: a single input sample is processed to produce an output
  - **Regional**: the output is dependent upon a region of samples

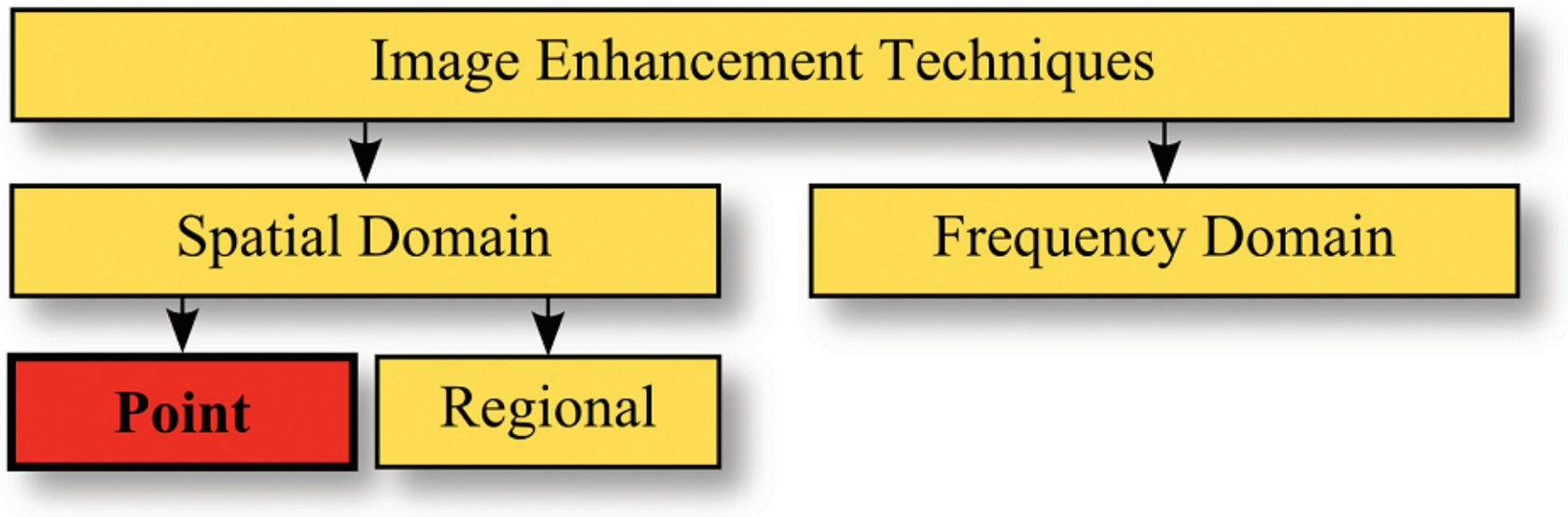
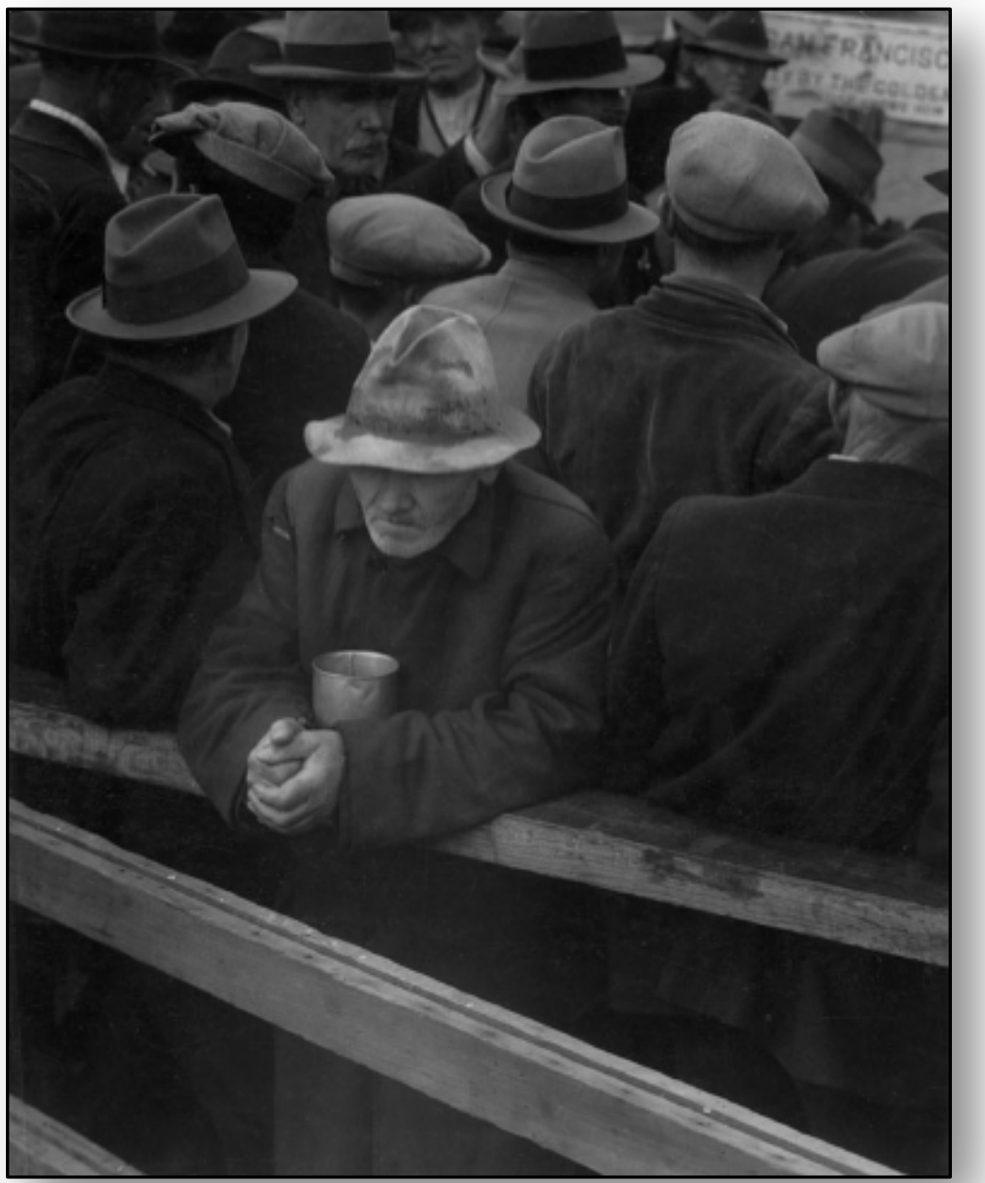


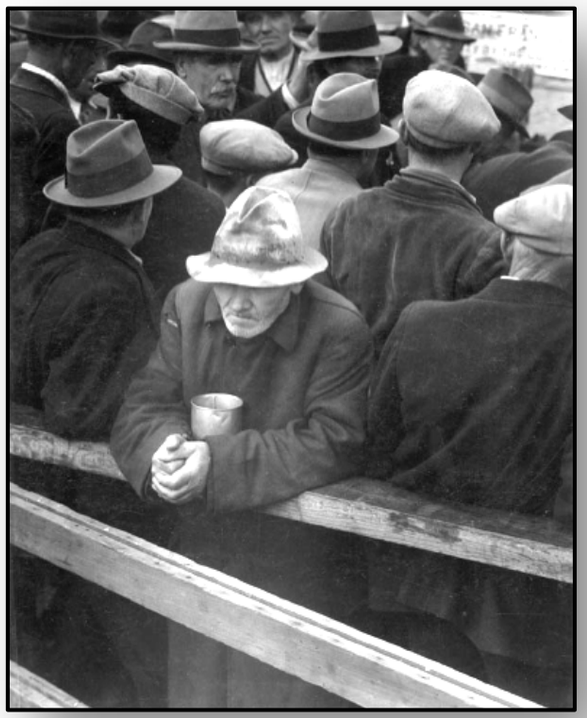
Figure 5.1. Taxonomy of image processing techniques.

Examples



gain = 1, bias = 55

gain = 1, bias = -55

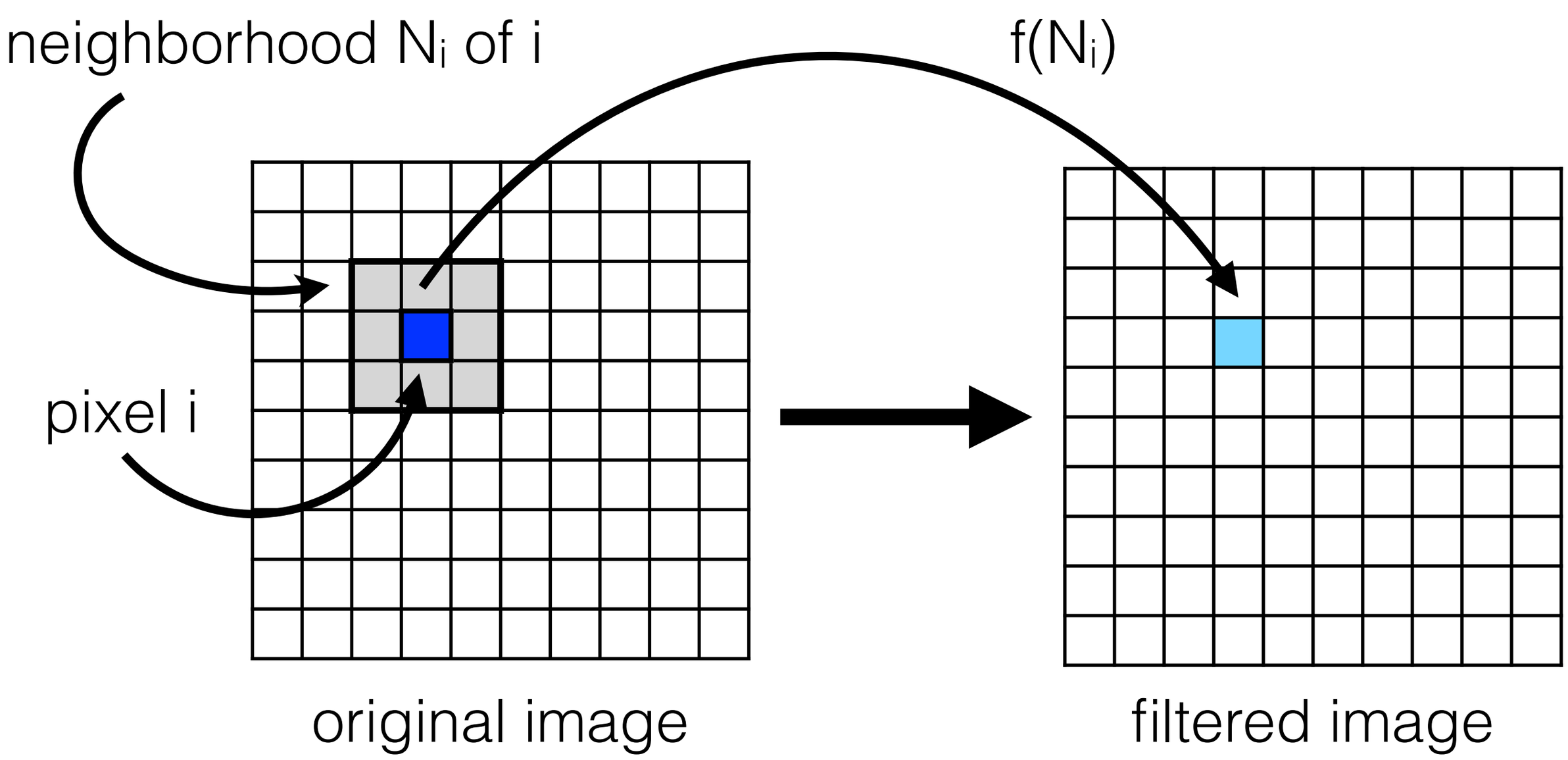


gain = 2, bias=0

gain = .5, bias=0

Filtering (Schematic)

$$C_{out} = f(N_{in})$$



Filtering (Algorithmic)

```
//given input: image

//produces output image: output


for (row = 0, row < H; row++) {
   for (col = 0; col < W; col++) {
      N = compute_neighborhood(image, row, col);


      new_color = filter(N);


      output[row][col] = new_color;
   }
}
```

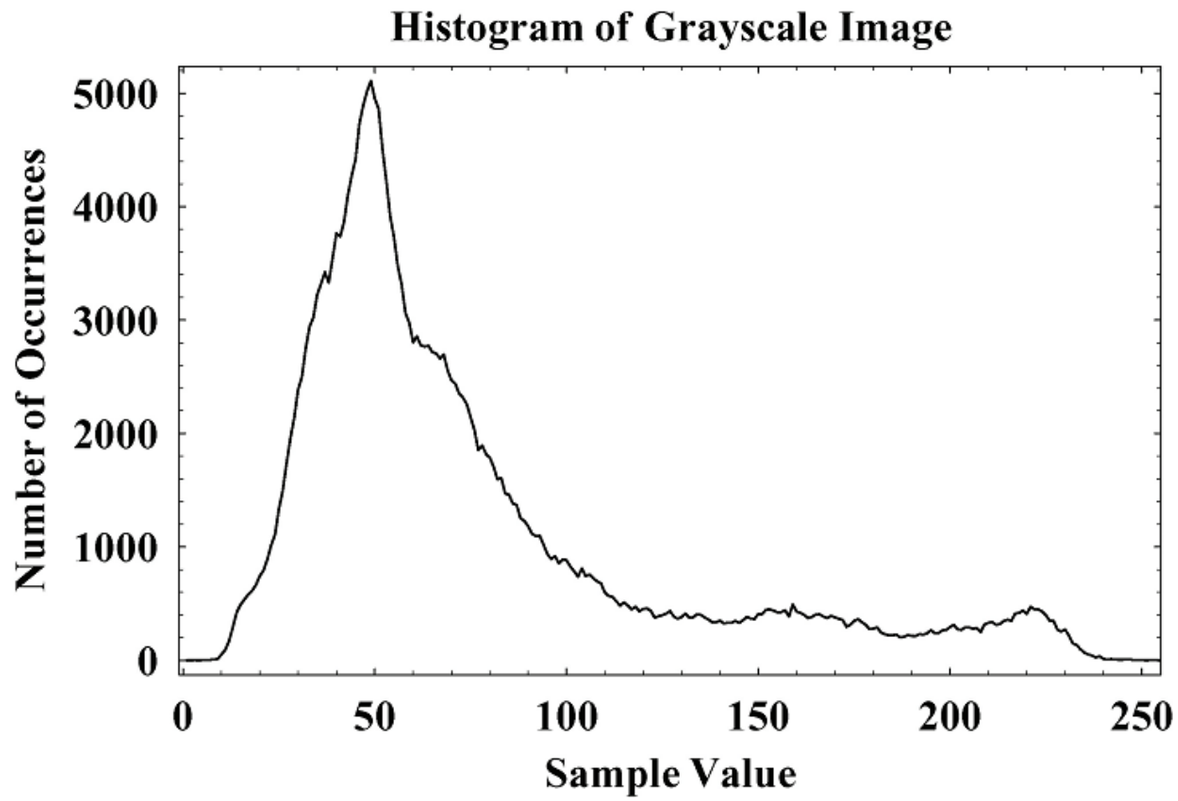
Histograms and Segmentation

Histograms

- A **histogram** is a table that simply counts the number of times a value appears in some data set.
  - In image processing, a histogram counts the image samples by color channel values (often, luminance).
  - For an 8-bit channel there will be 256 possible values a sample can be, so the histogram will store how many times each sample occurs.
  - In other words, the histogram gives the frequency distribution of sample values within the image.

Histogram Example

(a)



(b)

Figure 5.8. An example histogram: (a) an 8-bit grayscale image and (b) its histogram.

Histograms

- A histogram is typically **plotted** as a bar chart where the horizontal axis corresponds to the dynamic range of the image and the height of each bar corresponds to the sample count or the probability.

- The overall shape of a histogram doesn't always convey useful information, but there are several key insights that can be gained.
    - Narrow histogram distributions are representative of low contrast images
    - Wide distributions are representative of higher contrast images.
- We saw last time:
    - The histogram of an underexposed image will have a relatively narrow distribution with a peak that is significantly shifted to the left
    - The histogram of an overexposed image will have a relatively narrow distribution with a peak that is significantly shifted to the right.

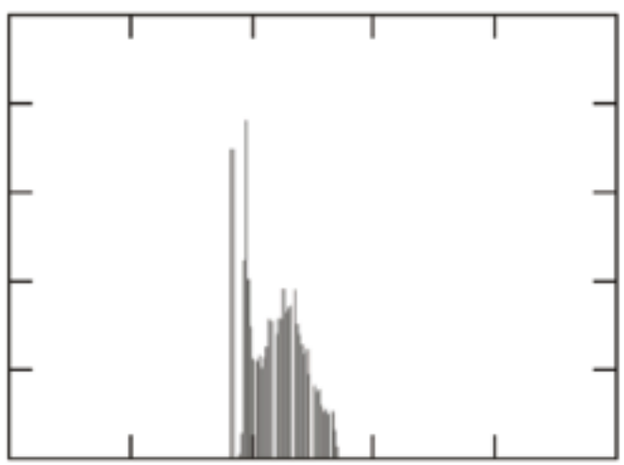
Match the Histogram to the Image

A

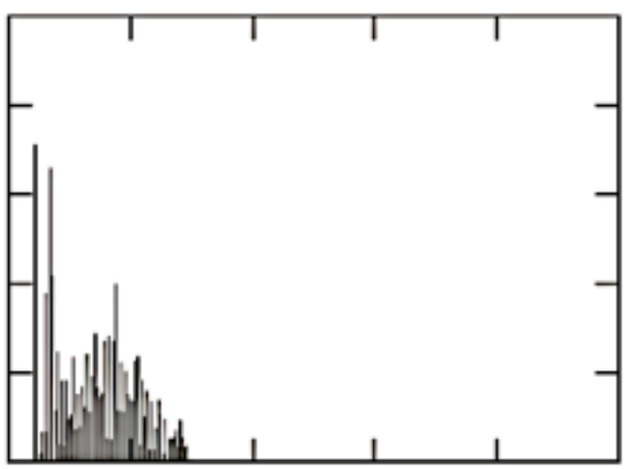
B

C

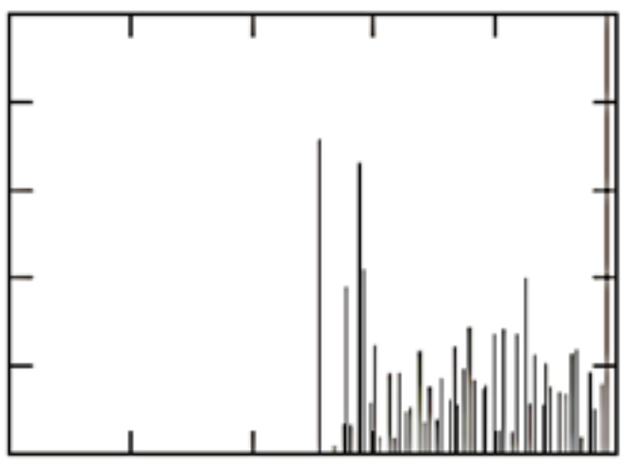
D



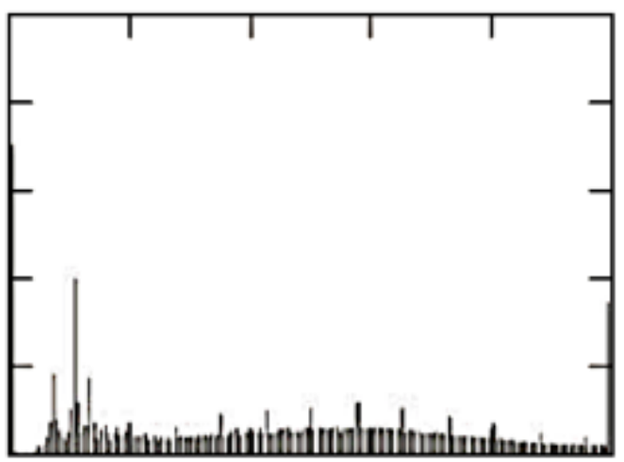
1



2

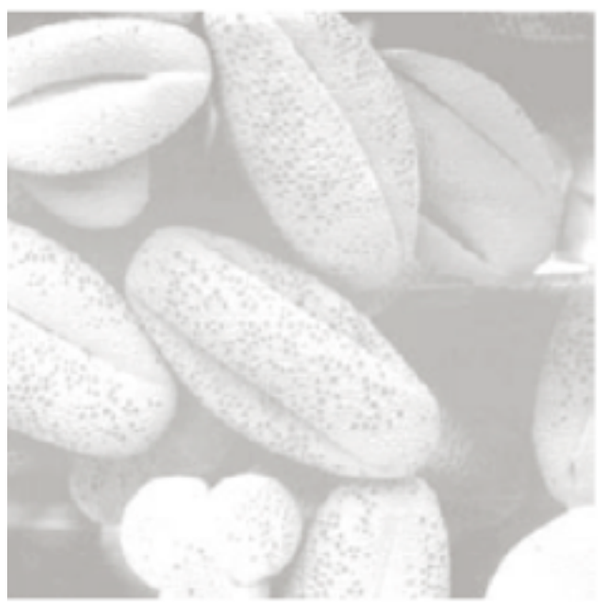


3



4

Match the Histogram to the Image

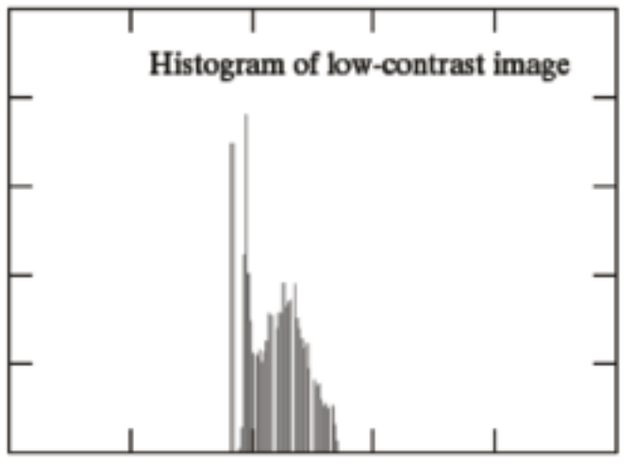


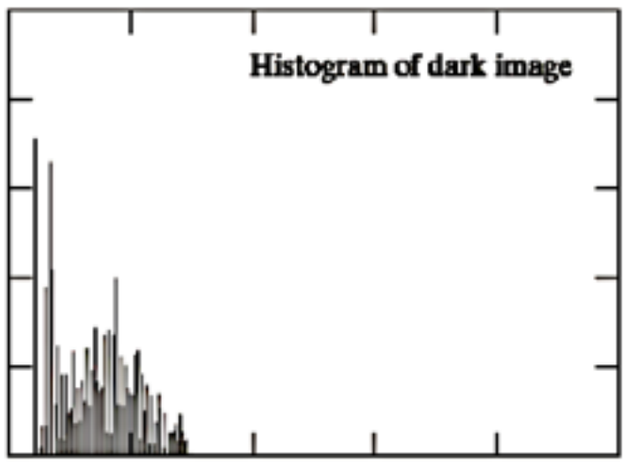
A

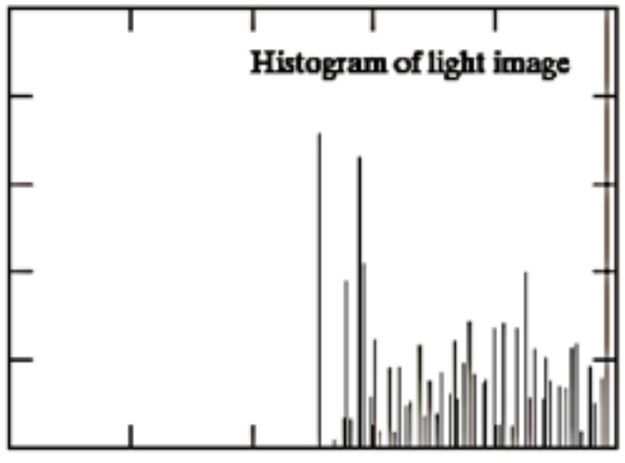
B

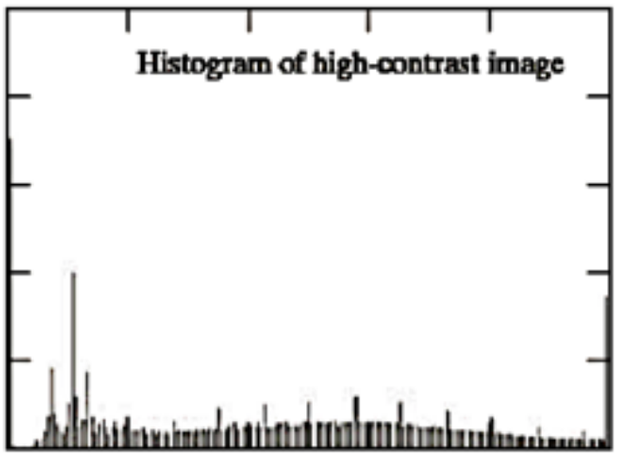
C

D









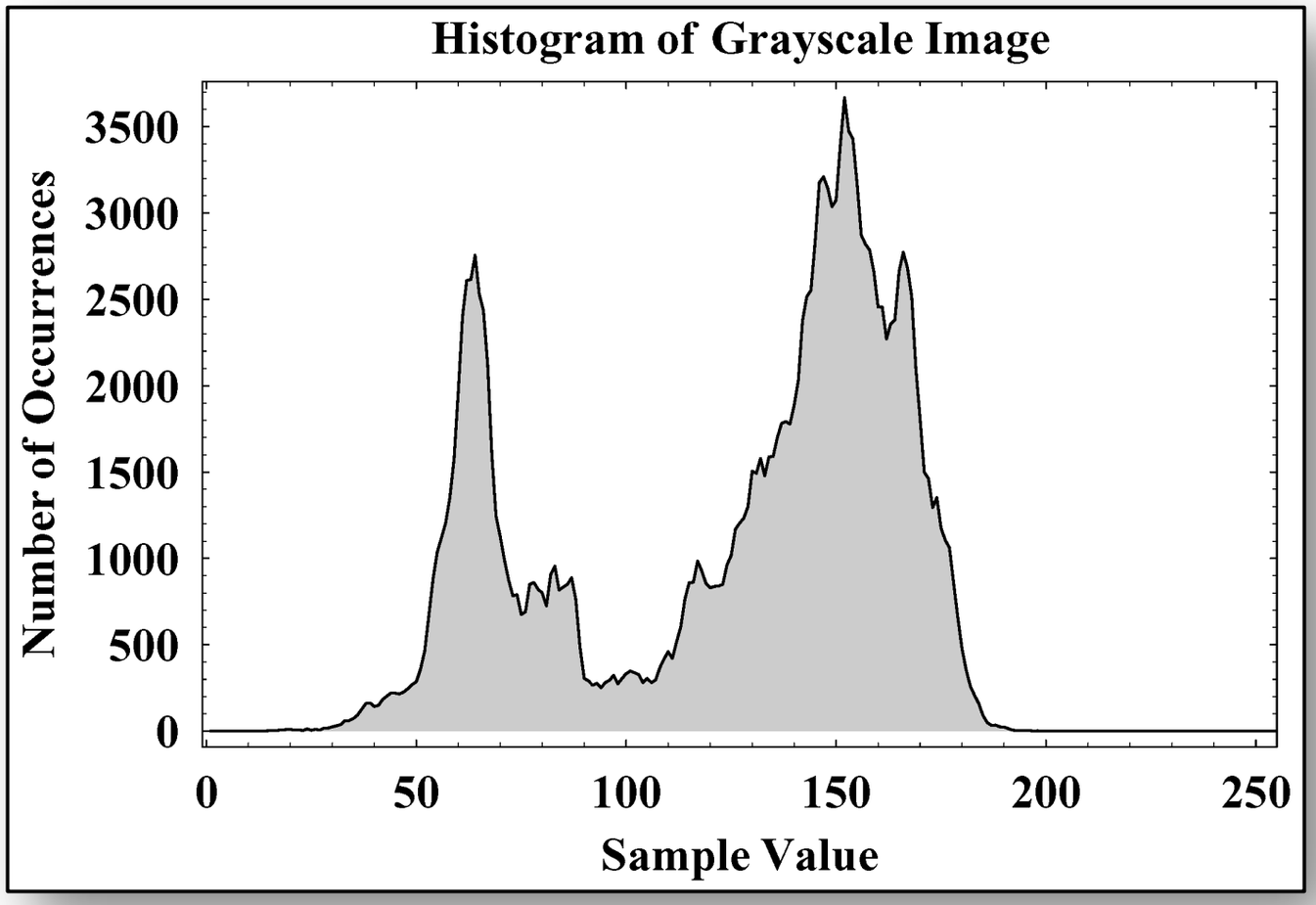
1

2

3

4

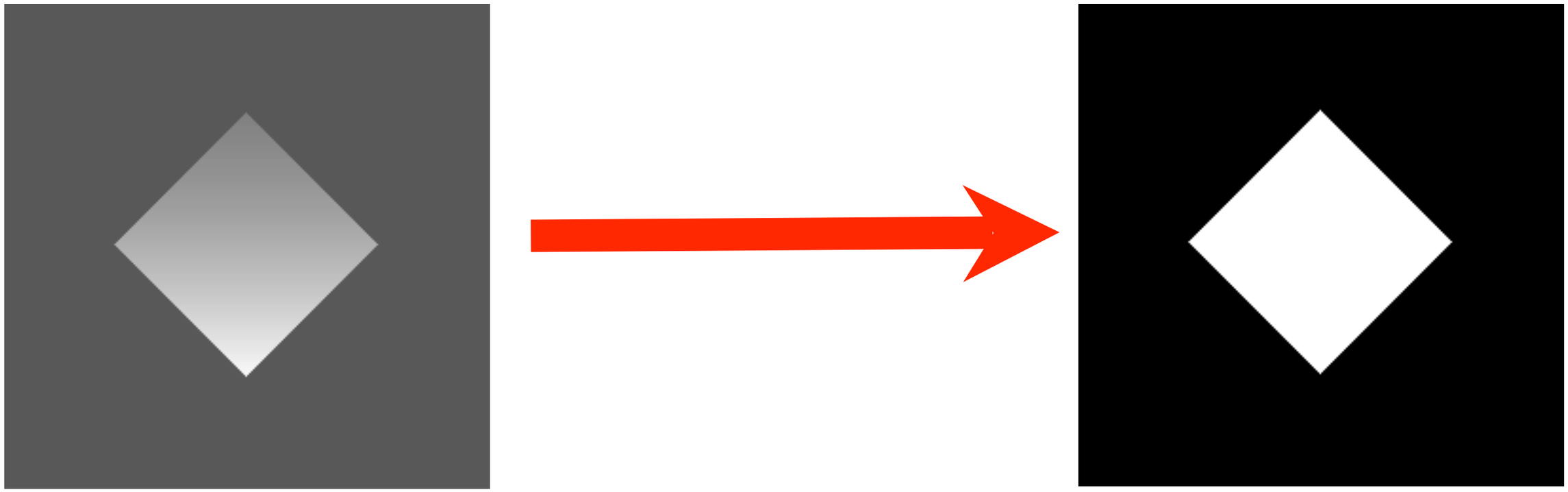
Histogram Example 2



- The image is characterized by low contrast. This is reflected in the histogram which shows that most samples are in a narrow region of the dynamic range (little information below 40 or above 190).

What is image segmentation?

- The process of subdividing an image into its constituent regions or objects.

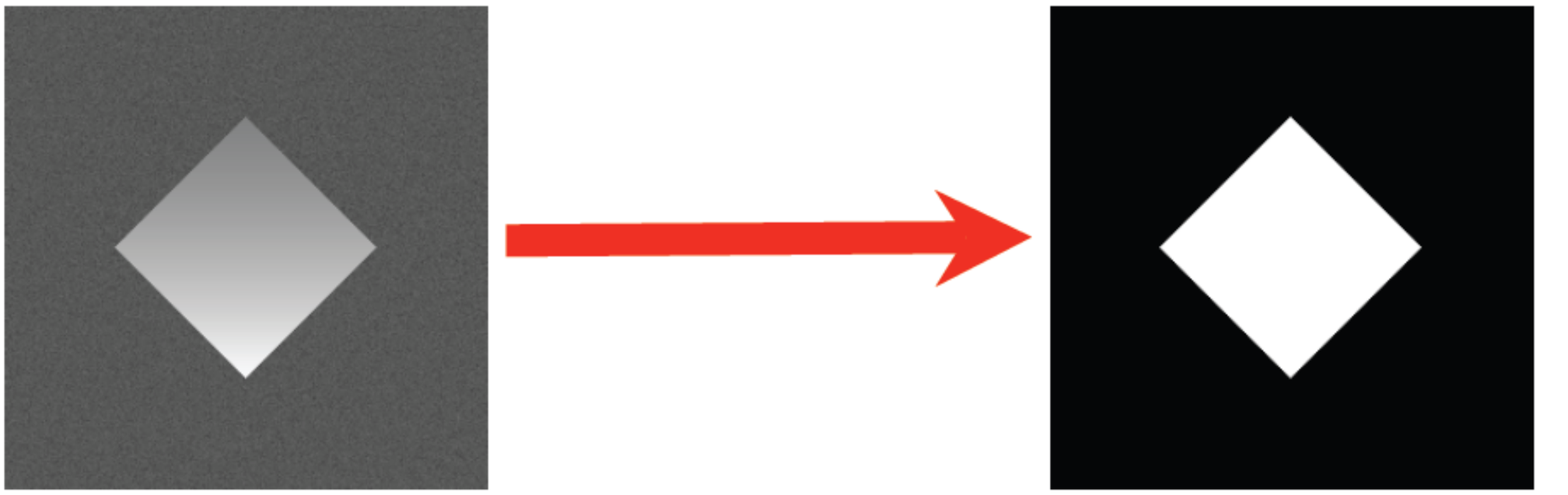


Input image
intensities 0-255

Segmentation output
0 (background)
1 (foreground)

Thresholding Based on Histogram

$$g(x,y) = \begin{cases} 1 & if \quad f(x,y) > T \\ 0 & if \quad f(x,y) \leq T \end{cases}$$



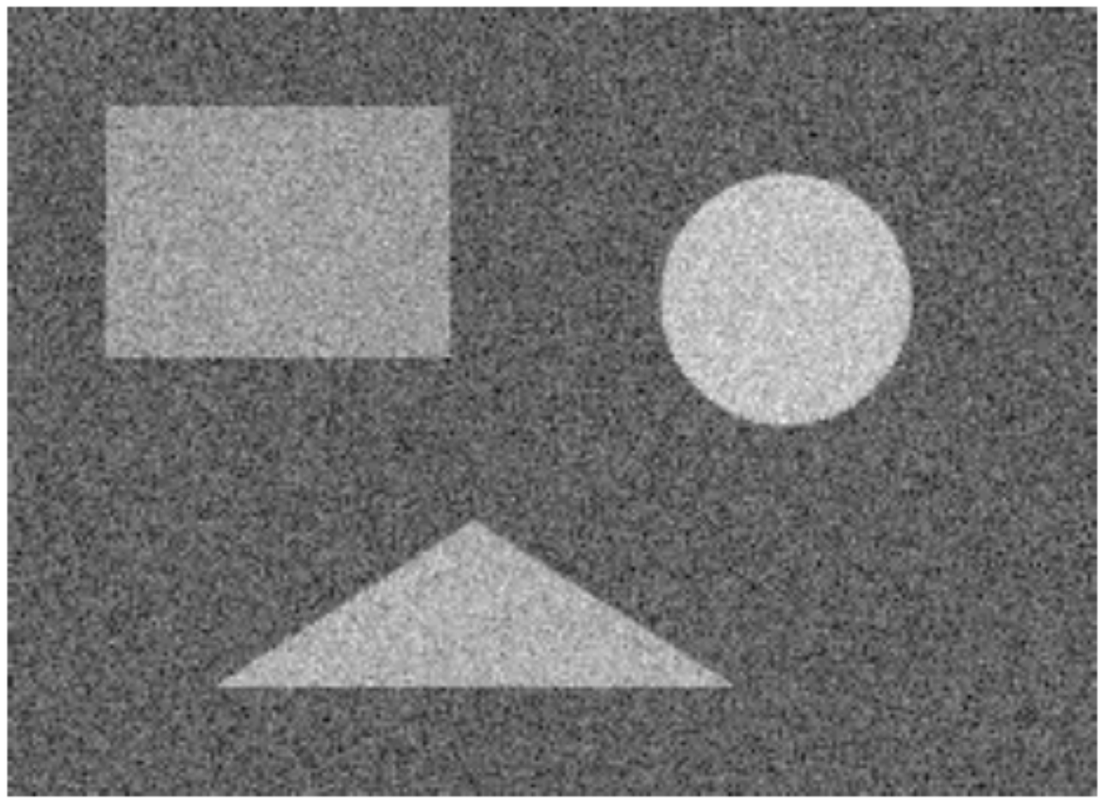
Input image f(x,y)
intensities 0-255

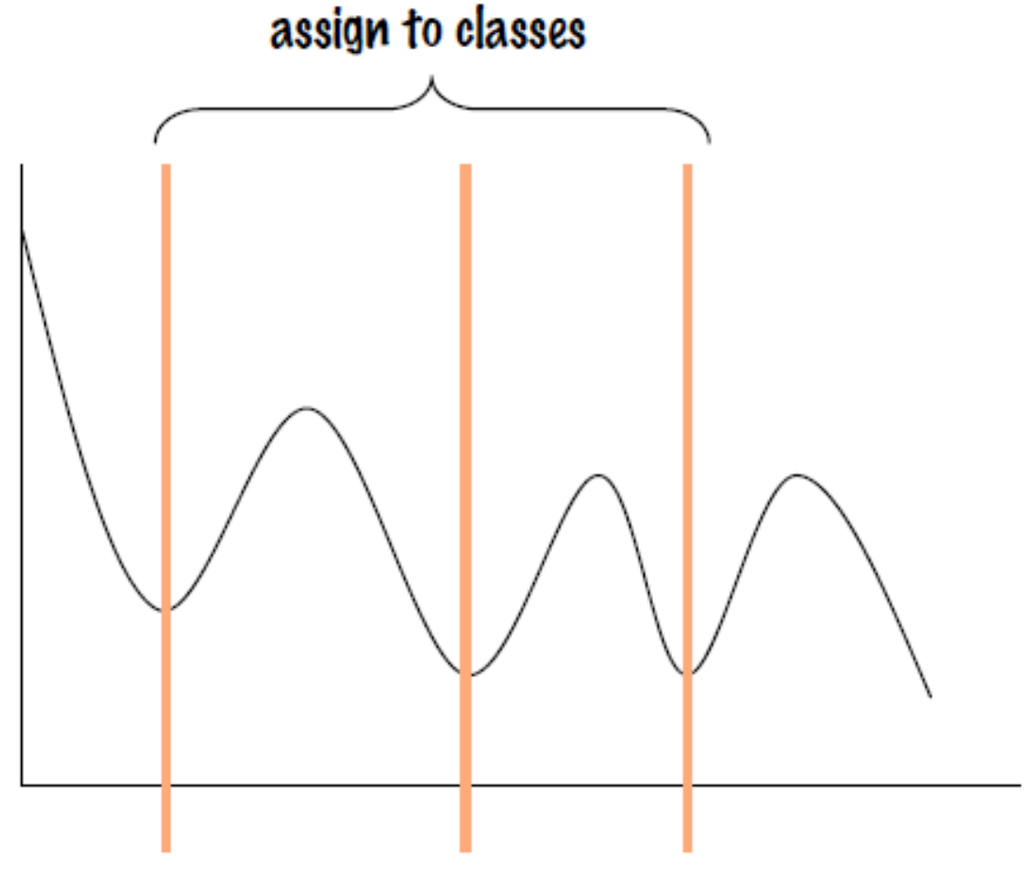
Segmentation output g(x,y)
0 (background)
1 (foreground)

- How to choose T is the key question:
  - Could use the color information (like in Lab03!), Trial and Error, etc.
  - Or you could use the histogram

Histograms and Noise

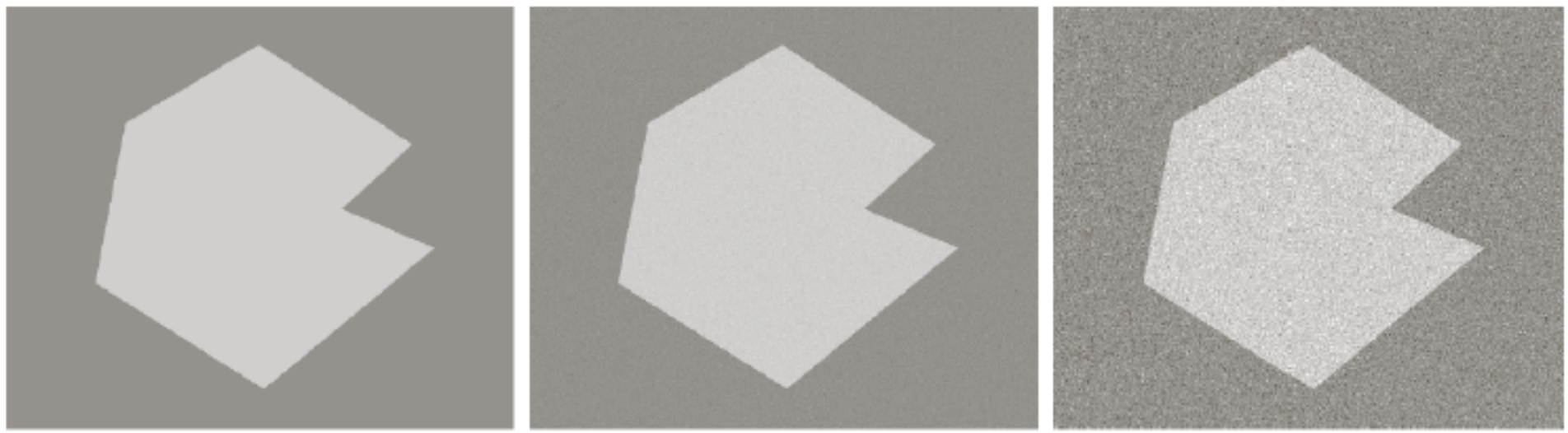
- What happens to the histogram if we add noise?
  - $g(x, y) = f(x, y) + n(x, y)$



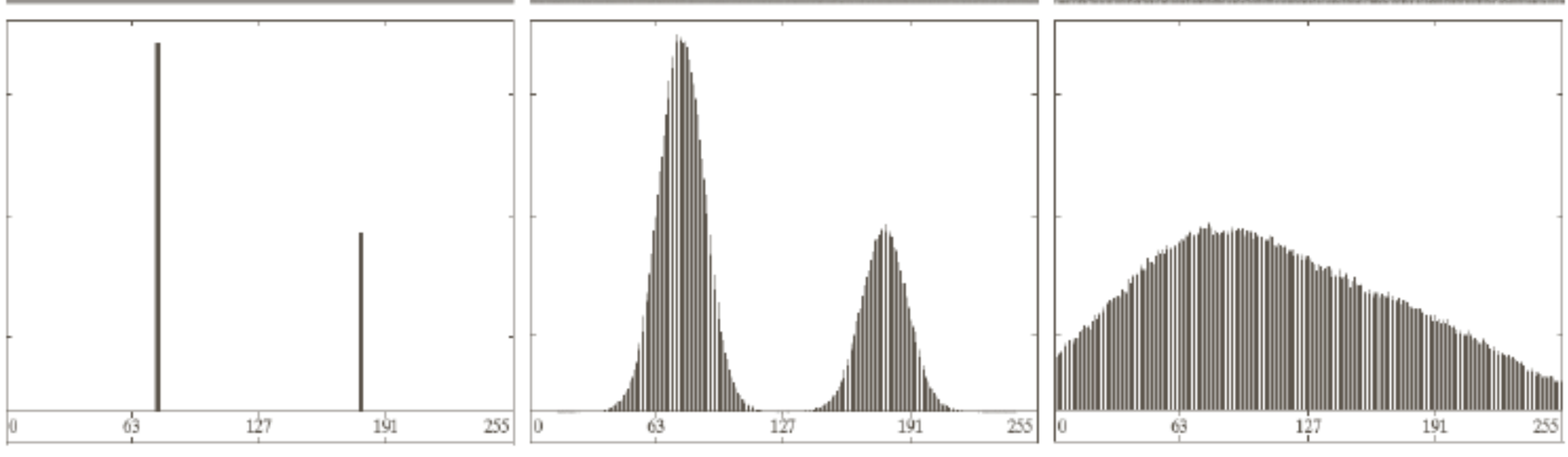


Effects of Noise on Histograms

Images



Histograms



No noise With noise More noise

Histogram Equalization

Histogram Equalization

- Given an image histogram, one question we could ask is how best to optimize the contrast of an image at all points.

- **Histogram equalization** seeks to improve image contrast by flattening, or equalizing, the histogram of an image.

- Can improve the local contrast of an image without altering the global contrast to a significant degree.

- Especially useful in images having large regions of similar tone such as an image with a very light background and dark foreground.

- Histogram equalization can expose hidden details in an image by stretching out the contrast of local regions, making the differences in the region more visible.

Filtering with Histograms

- Goal: redistribute luminances so that they used more equally
- Result: higher contrast image

$$\hat{Y}_i = 1/T \sum_{j=0}^{\bar{Y}_i} H[j]$$

Luma Update

Total # of pixels Histogram Counts

$$S_i = \hat{Y}_i / Y_i$$

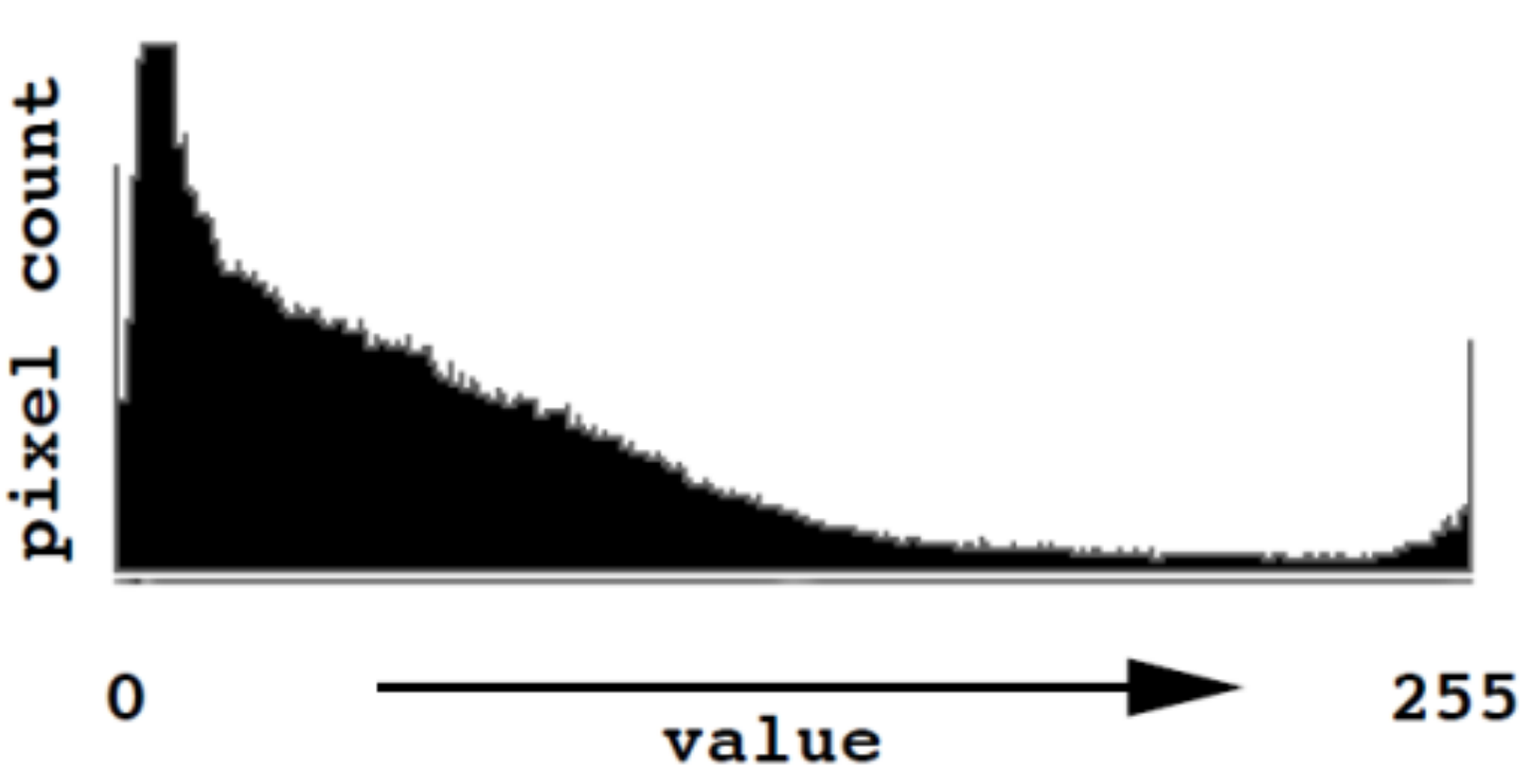
Scale Factor (Multiply each color channel by S)

$C_{new} = S * C$

then Normalize!!!

Histogram Equalization

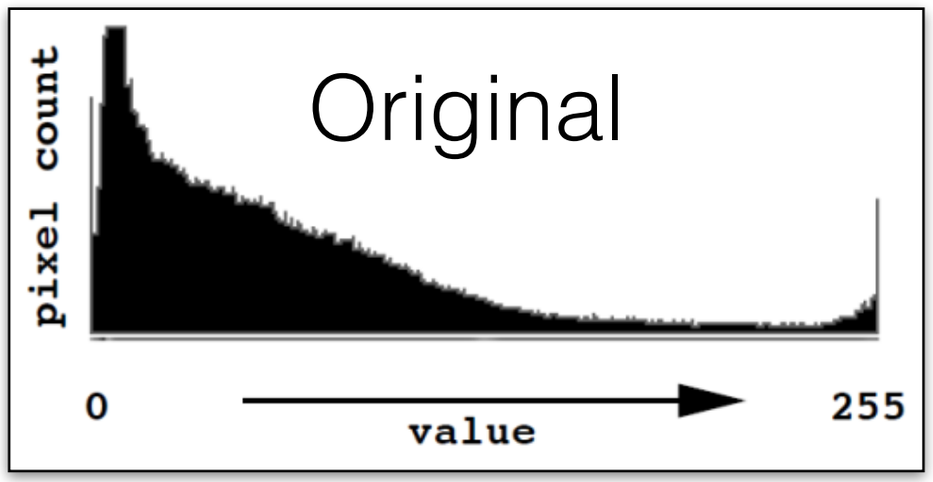
a) original image

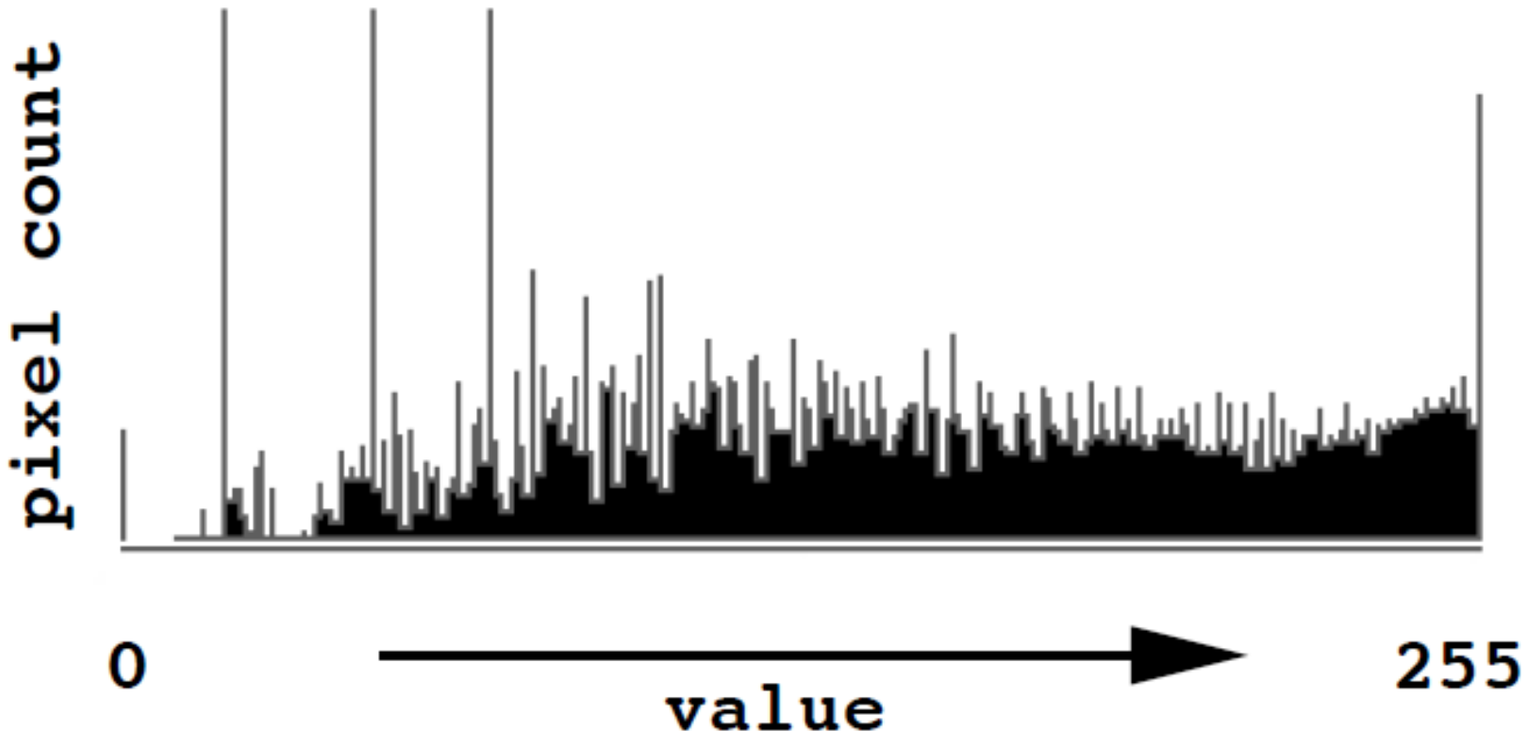


b) histogram

Histogram Equalization

a) equalized image





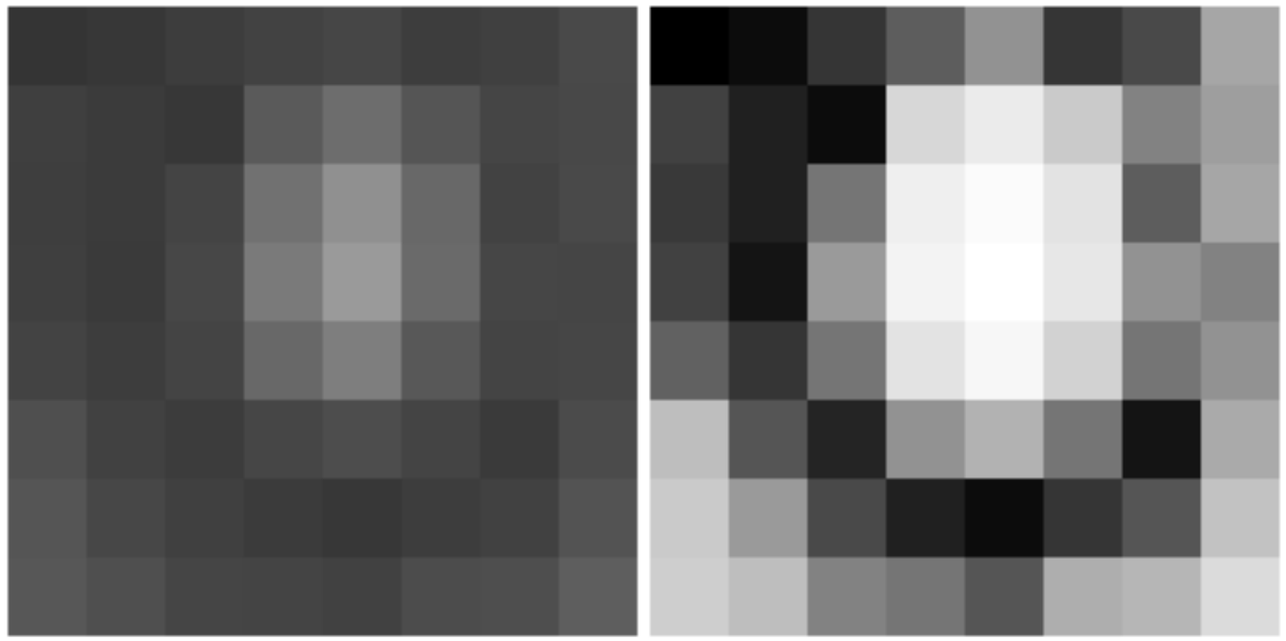
b) histogram

A Smaller Example

$$\begin{bmatrix} 52 & 55 & 61 & 66 & 70 & 61 & 64 & 73 \\ 63 & 59 & 55 & 90 & 109 & 85 & 69 & 72 \\ 62 & 59 & 68 & 113 & 144 & 104 & 66 & 73 \\ 63 & 58 & 71 & 122 & 154 & 106 & 70 & 69 \\ 67 & 61 & 68 & 104 & 126 & 88 & 68 & 70 \\ 79 & 65 & 60 & 70 & 77 & 68 & 58 & 75 \\ 85 & 71 & 64 & 59 & 55 & 61 & 65 & 83 \\ 87 & 79 & 69 & 68 & 65 & 76 & 78 & 94 \end{bmatrix}$$

$$\downarrow$$

$$\begin{bmatrix} 0 & 12 & 53 & 93 & 146 & 53 & 73 & 166 \\ 65 & 32 & 12 & 215 & 235 & 202 & 130 & 158 \\ 57 & 32 & 117 & 239 & 251 & 227 & 93 & 166 \\ 65 & 20 & 154 & 243 & 255 & 231 & 146 & 130 \\ 97 & 53 & 117 & 227 & 247 & 210 & 117 & 146 \\ 190 & 85 & 36 & 146 & 178 & 117 & 20 & 170 \\ 202 & 154 & 73 & 32 & 12 & 53 & 85 & 194 \\ 206 & 190 & 130 & 117 & 85 & 174 & 182 & 219 \end{bmatrix}$$



Original Equalized

Value	Count	Value	Count	Value	Count	Value	Count	Value	Count
52	1	64	2	72	1	85	2	113	1
55	3	65	3	73	2	87	1	122	1
58	2	66	2	75	1	88	1	126	1
59	3	67	1	76	1	90	1	144	1
60	1	68	5	77	1	94	1	154	1
61	4	69	3	78	1	104	2		
62	1	70	4	79	2	106	1		
63	2	71	2	83	1	109	1		

http://en.wikipedia.org/wiki/Histogram_equalization

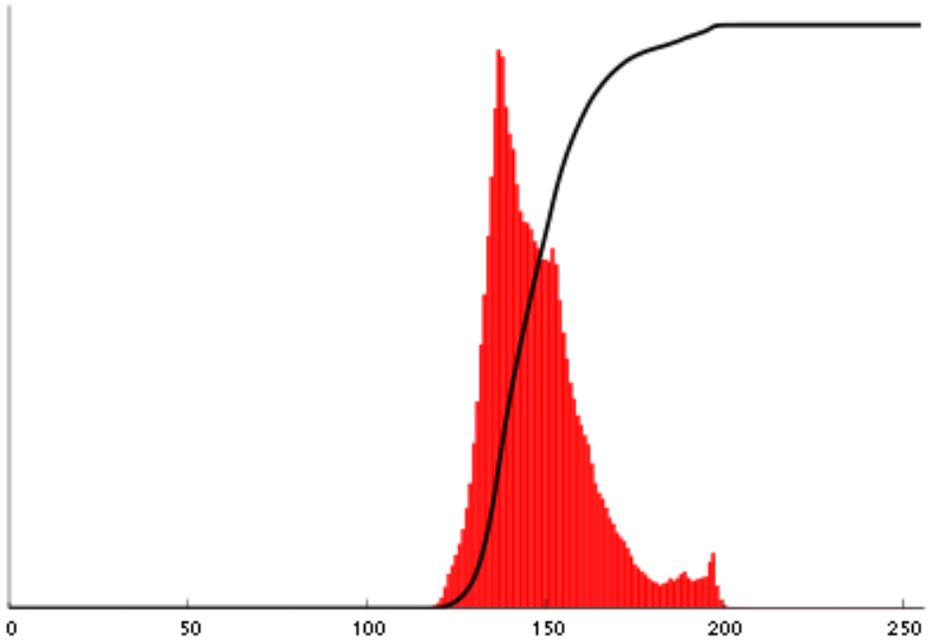
Histogram Equalization Linearizes the Cumulative Density Function

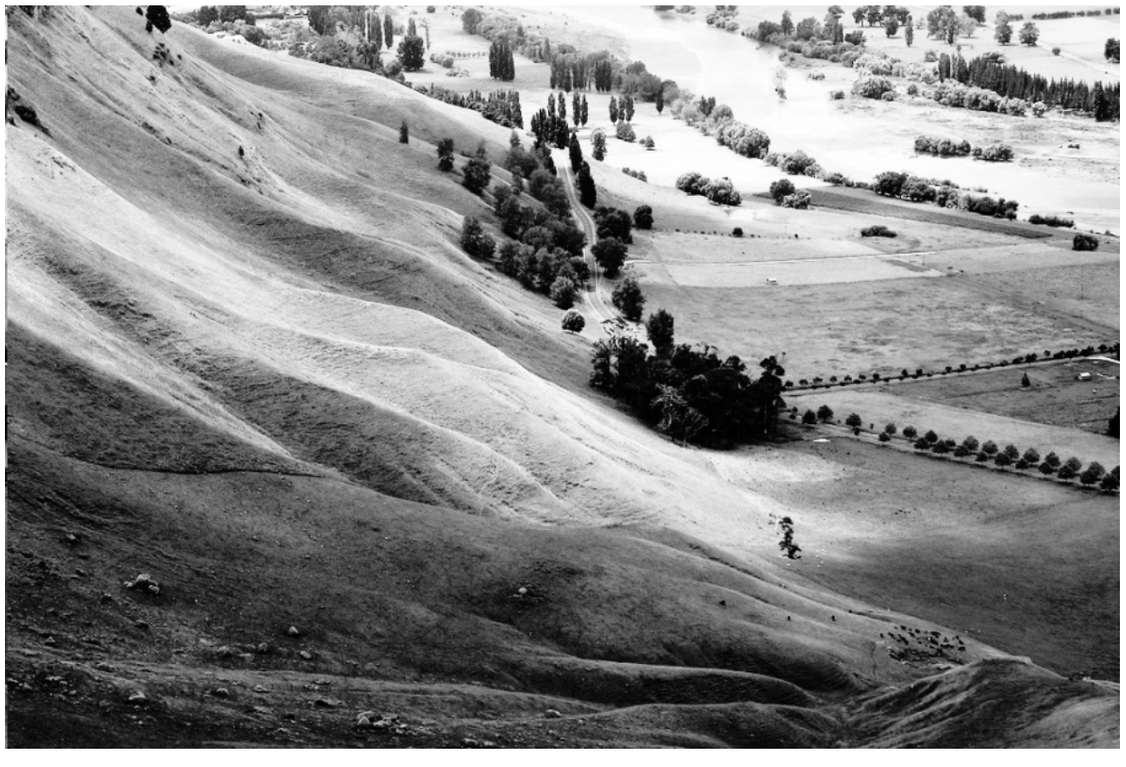
- Uses the cumulative distribution function (CDF) as the lookup table.
- Given an N-bit image having histogram h, the normalized CDF is given by:

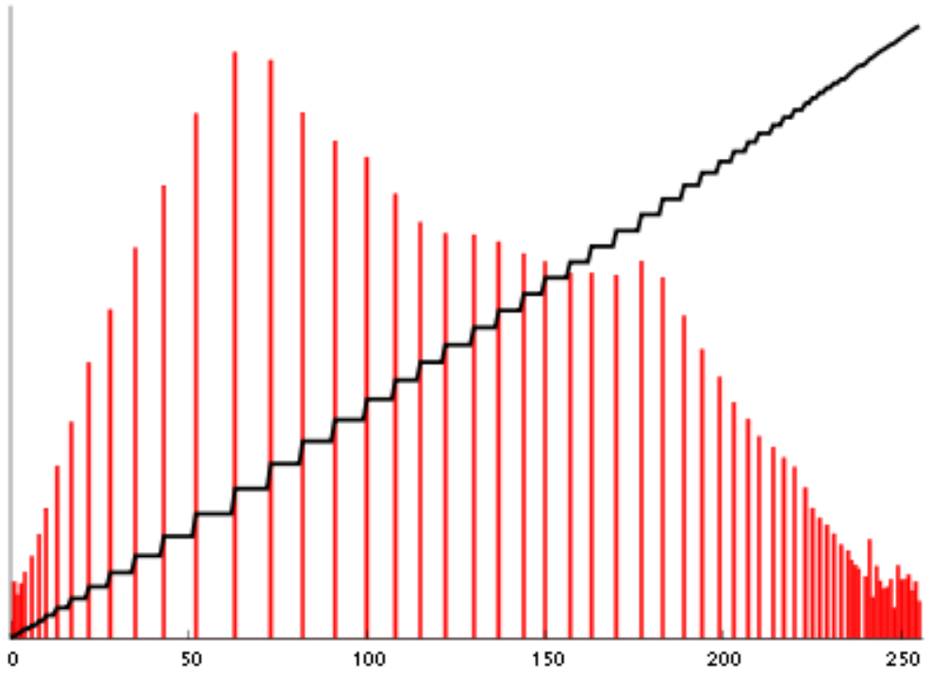
$$\hat{C}_j = \sum_{i=0}^{j} \hat{h}_i, \qquad j \in \{0, 1, \ldots, 255\}\,. \tag{5.12}$$

- The CDF essentially answer the question “what percentage of the samples in an image are equal-to-or-less-than value j?”
- The normalized CDF must be rescaled to [0,255] and is then used as the lookup table.

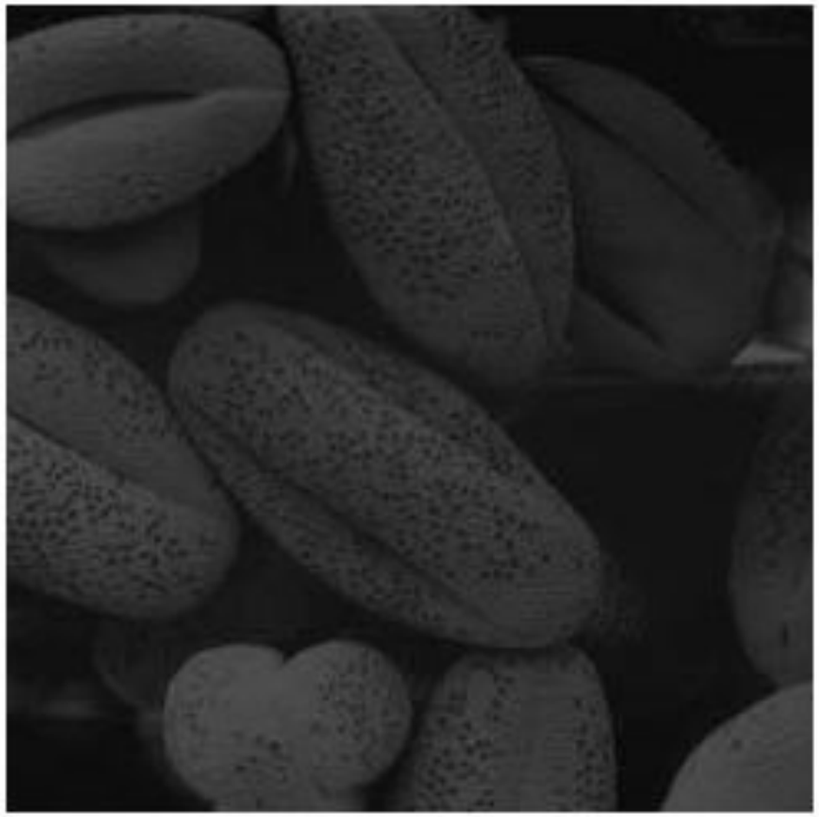
The Black Line is the CDF

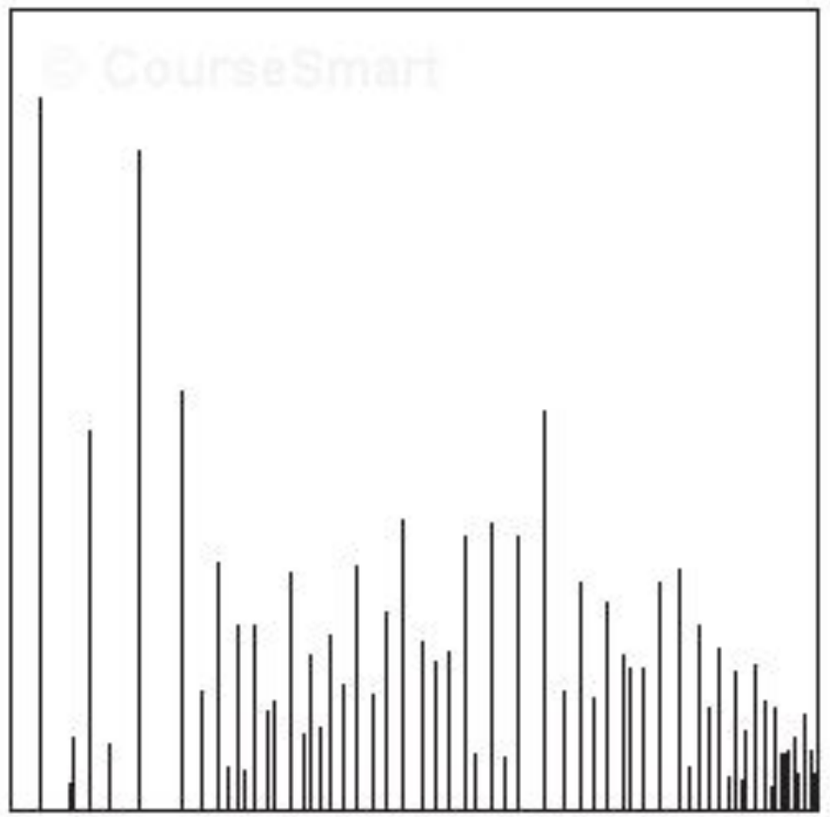


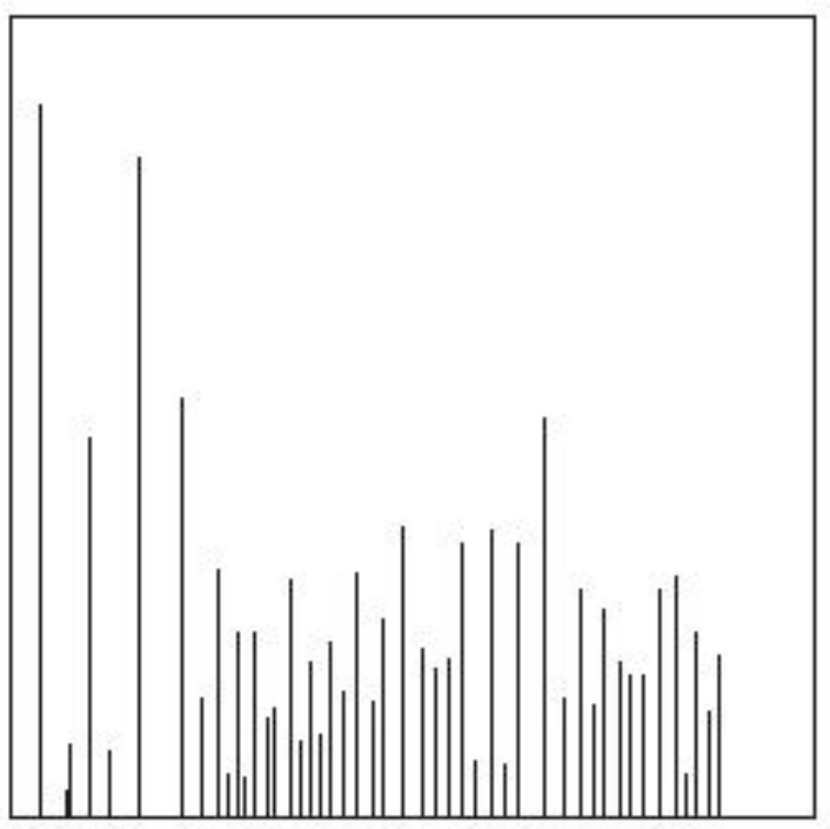




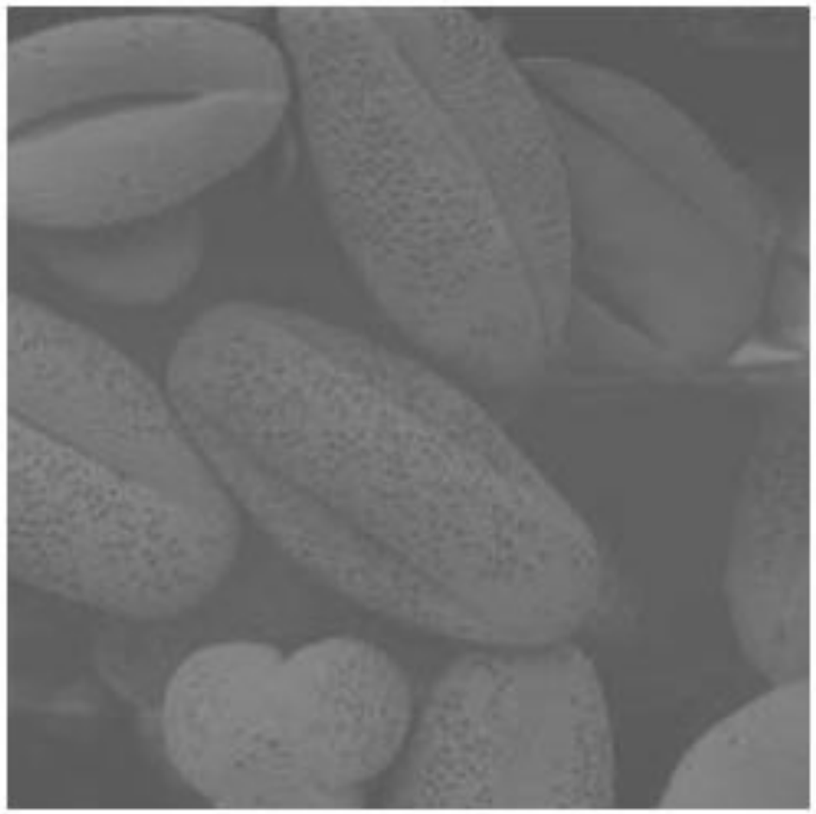
Histogram After Equalization

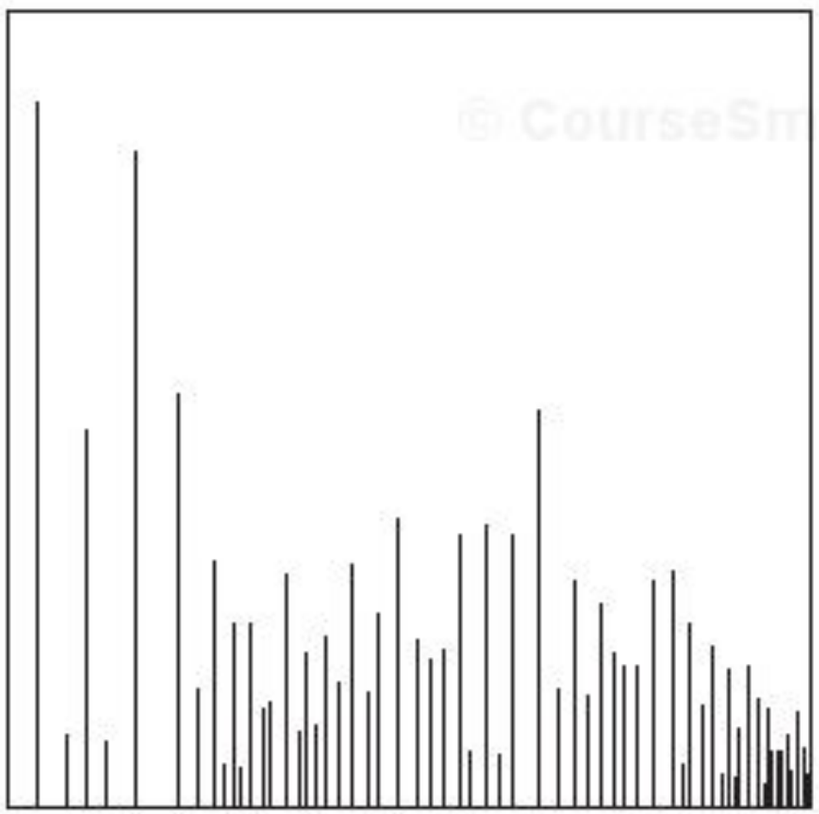


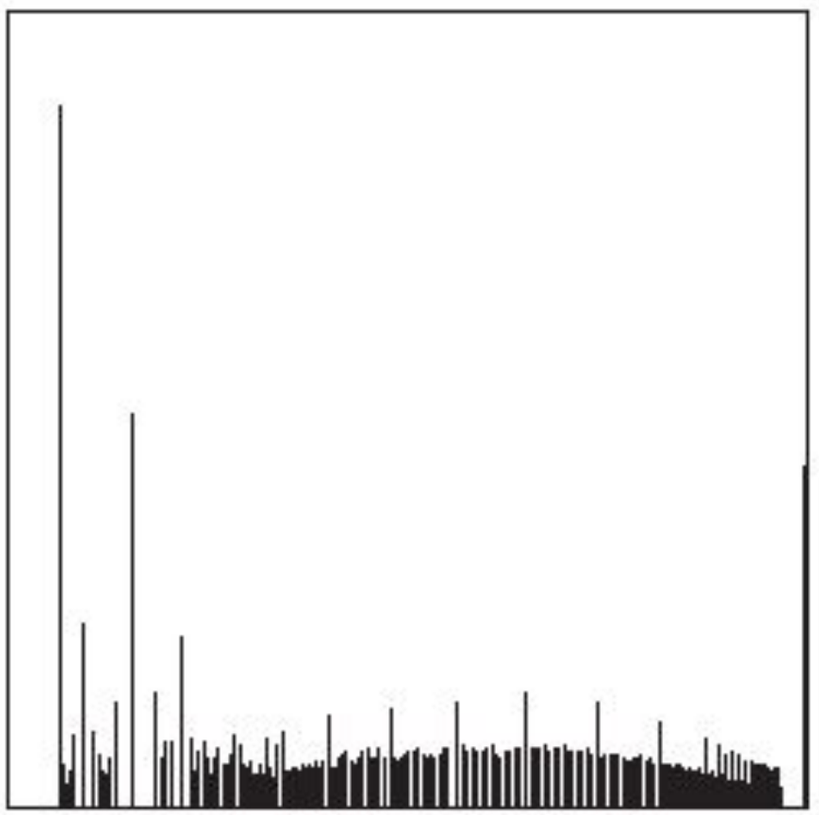


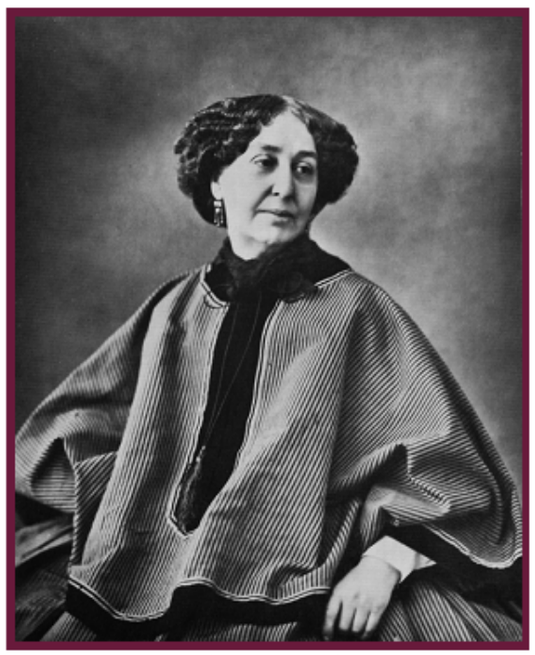


Histogram After Equalization

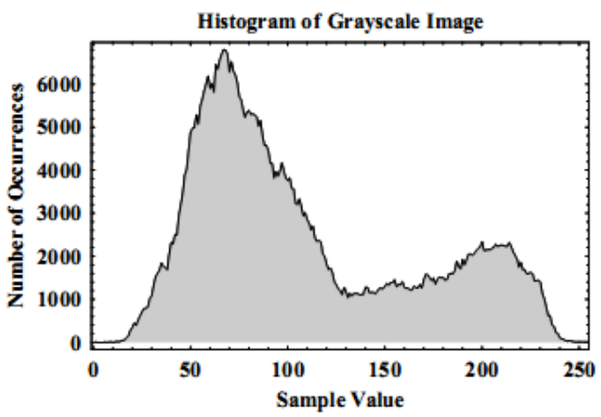




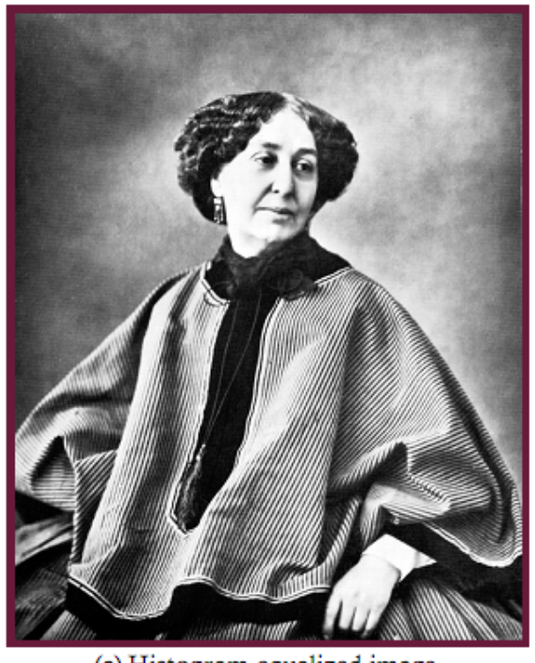




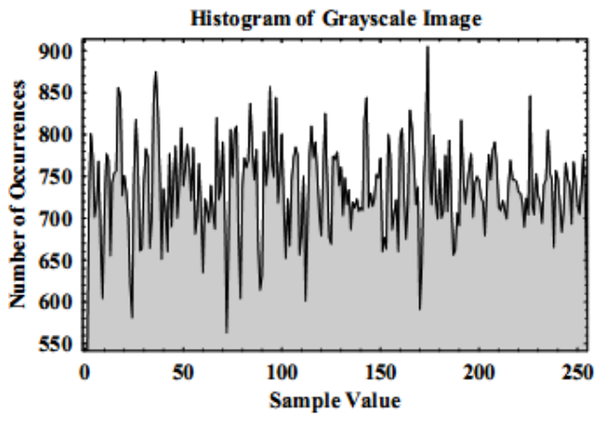
(a) Underexposed image.



(b) Histogram of the underexposed image.

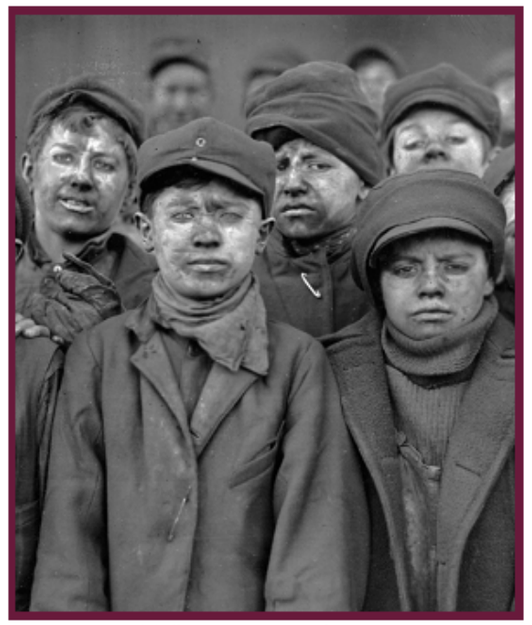


(c) Histogram-equalized image.

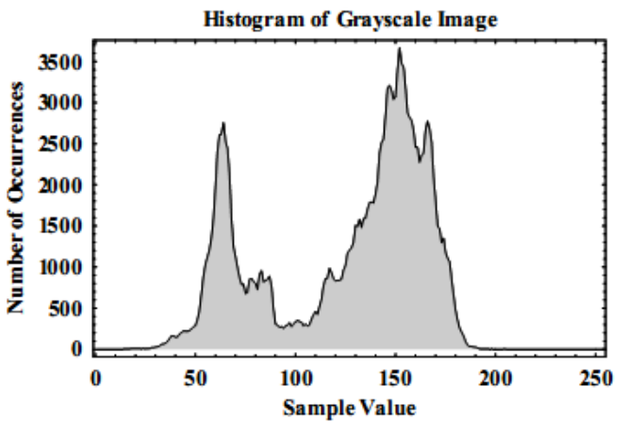


(d) Histogram of the equalized image.

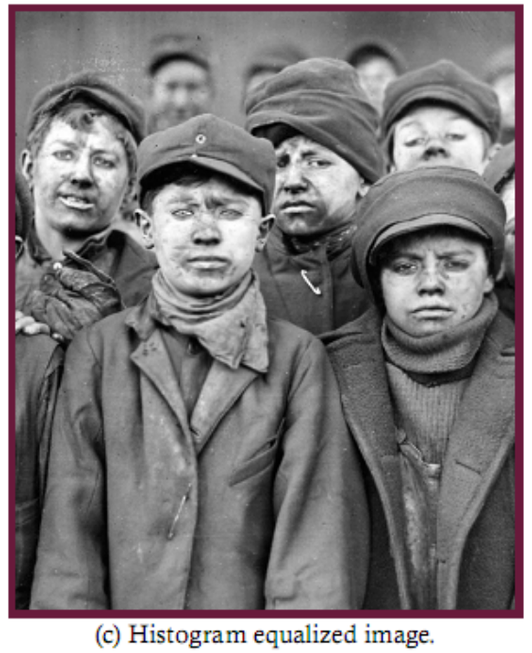
Figure 5.10. Histogram equalization of an underexposed image.



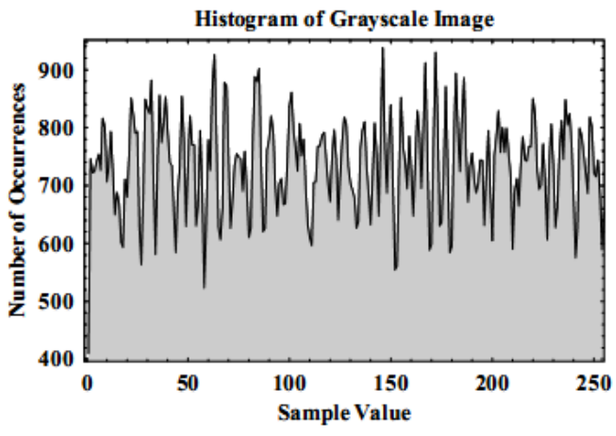
(a) Low-contrast image.



(b) Histogram of the low-contrast image.



(c) Histogram equalized image.

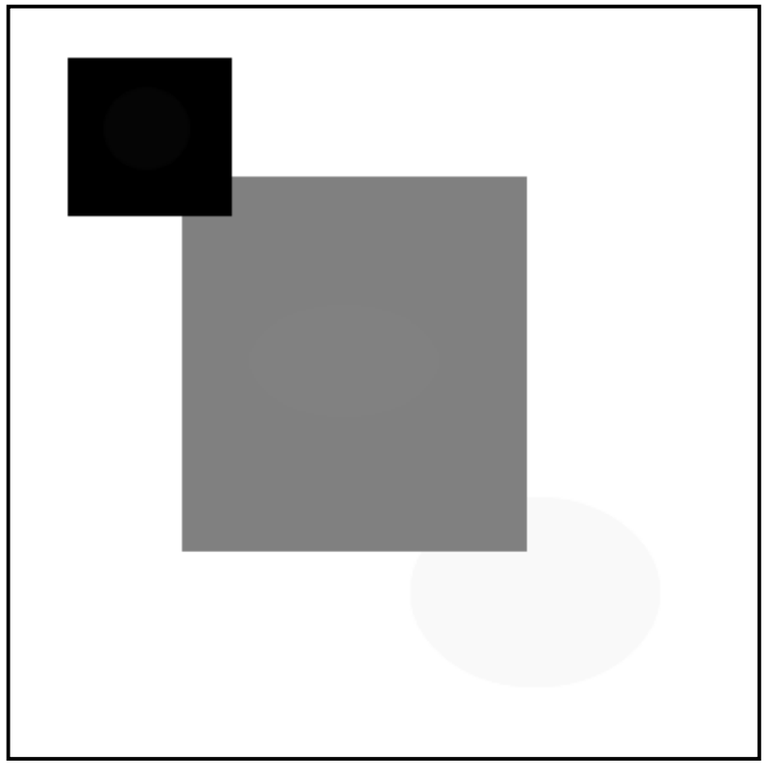


(d) Histogram of the equalized image.

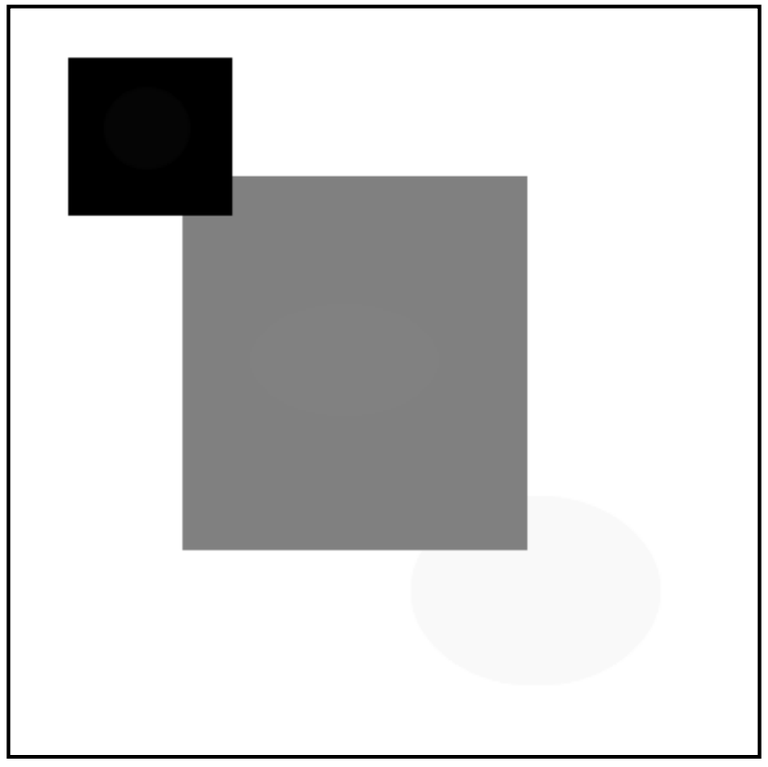
Figure 5.11. Histogram equalization of a low-contrast image.

Local Equalization

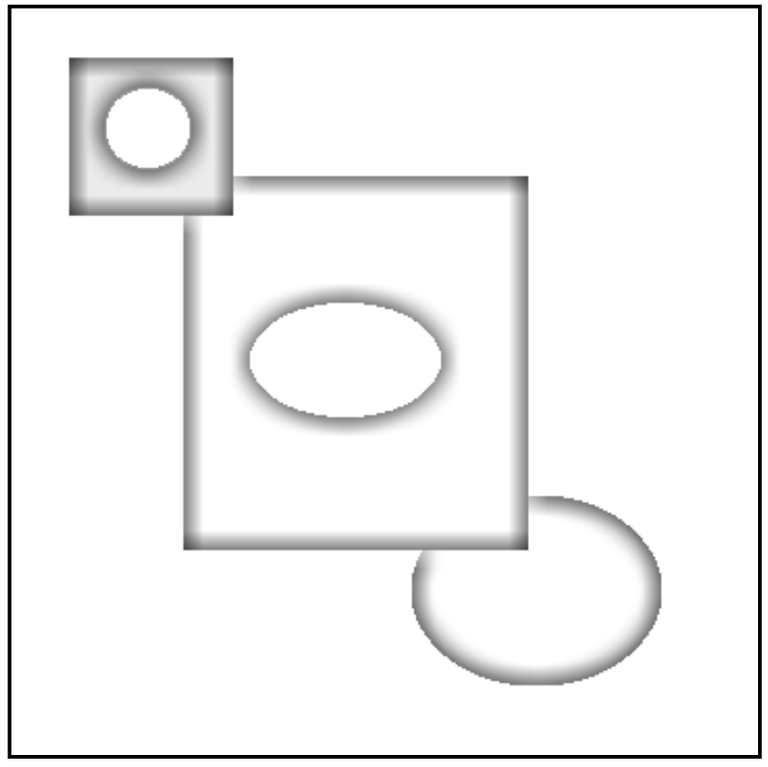
- Histogram equalization can be performed on a “local” level.
- Compute the histogram and CDF of a local region about each pixel and then use that CDF as a lookup table for **only that pixel**.
- Has the (possibly negative) effect of eliminating global contrast!



Original Image



Equalized Image



Locally Equalized Image

Local Histogram Equalization Example (bad)

Color Histograms

Histogram Equalization for Color Images

- Histogram equalization can also be done on color images by performing the grayscale technique on each separate channel of the image.

- The colors will likely be dramatically altered as a result.

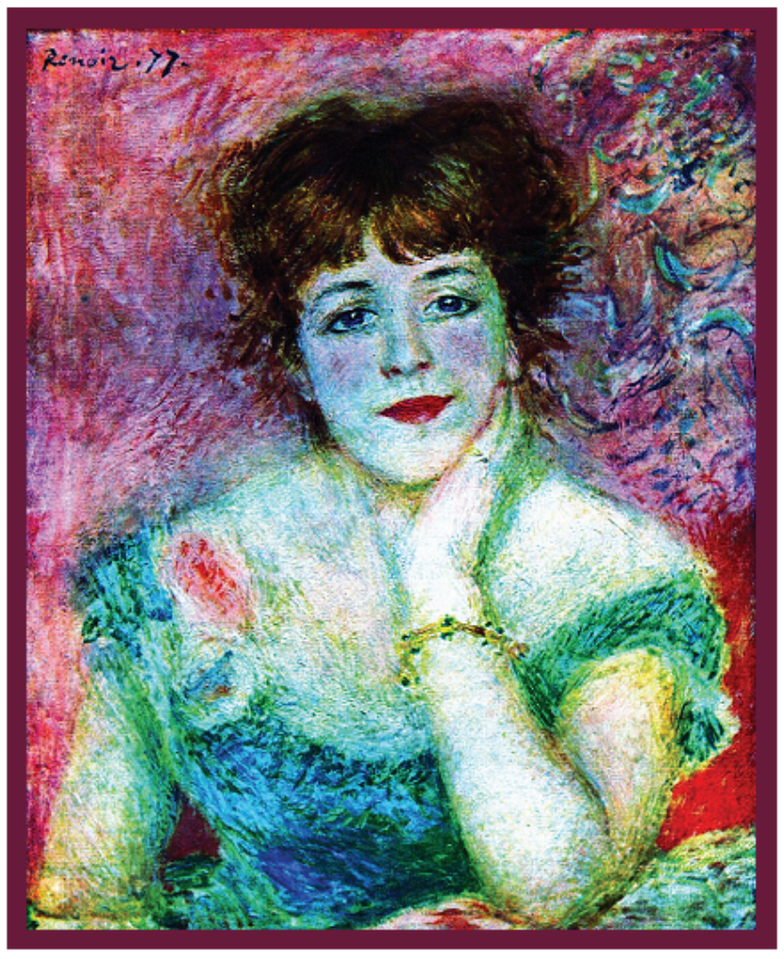
- If the tonal distributions are different among the red, green, and blue channels of an image, for example, the lookup tables for each channel will be vastly different and equalization will alter the color patterns present in the source.

- Histogram equalization of a color image is often best performed on the intensity channel only, e.g. using the B or Y channel of HSB or YUV color spaces.

Color Example

(b) each band, (c) only the luminance

(a) Source.



(b) Equalized RGB.

(c) Equalized intensity.

Figure 5.12. Equalizing a color image.

Color Histograms

- A **color histogram** is a 3D entity where each pixel of an image (rather than each sample) is placed into a bin.

- The color space is divided into volumetric bins each of which represent a range of colors.

- Each axis of the color space may be divided independently of the others. This allows the axes to have different resolutions.

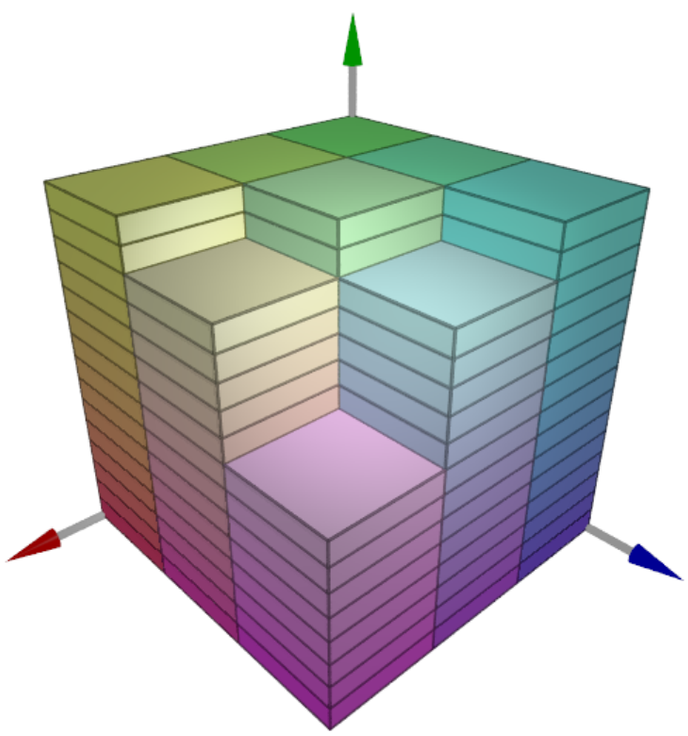
  - In HSV may want to allocate more resolution on V than H or S

  - In RGB may want to allocation more resolution in G than R or B (why?)

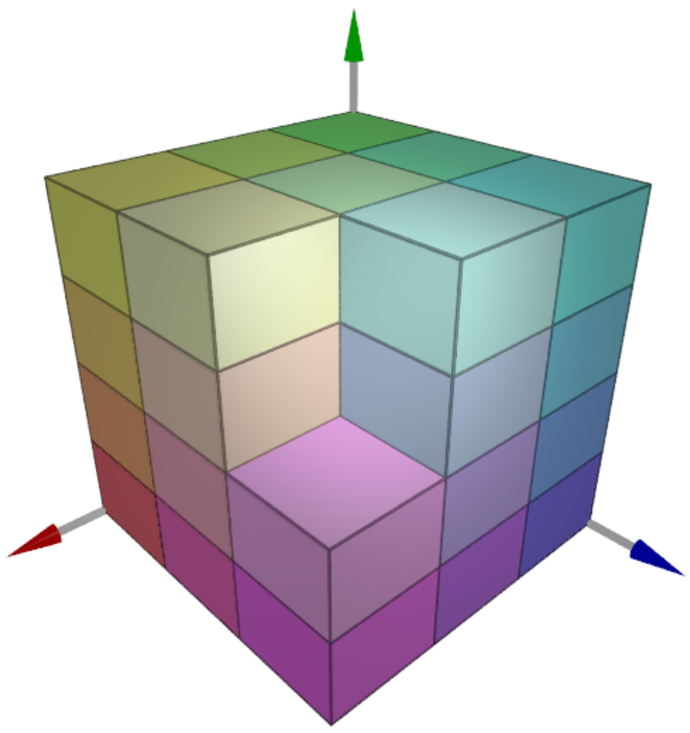
- Color histograms provide a concise but usually coarse characterization of an image

Binning Color Spaces

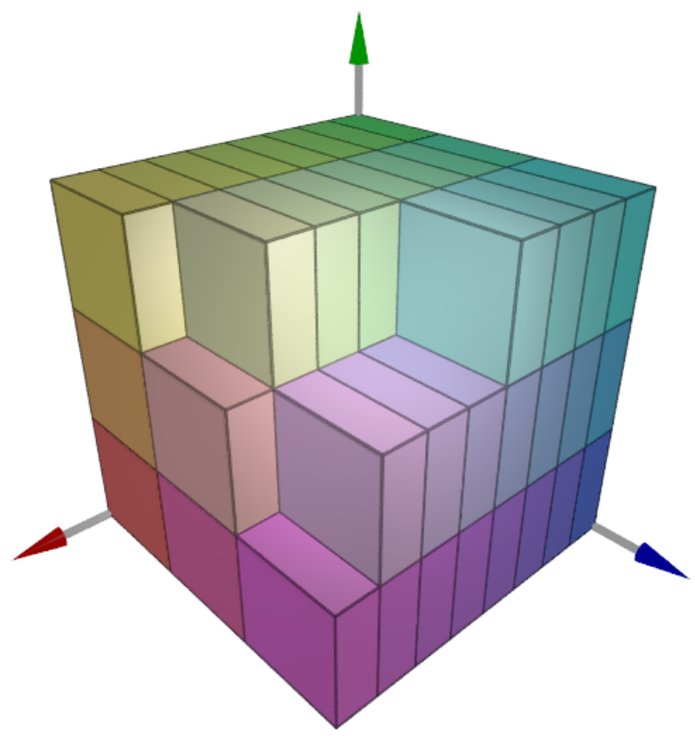
- The resolution of each axis may be set independently of the others.



3x15x3



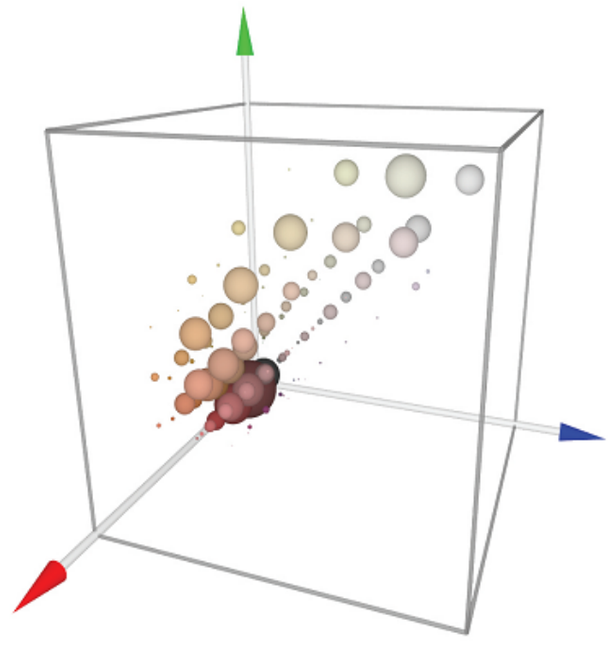
3x4x3



8x3x3

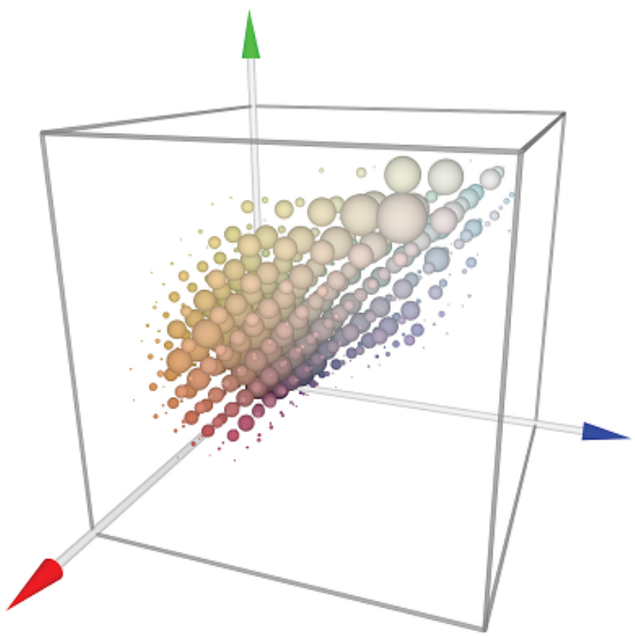
Color Histogram Example

(a) Source image.



(b) RGB color histogram.

(c) Source image.



(d) RGB color histogram.

Figure 5.13. The color distribution of two source images as given by their $12 \times 12 \times 12$ RGB color histograms.

Some Comments on Filter Neighborhoods

Local Filters

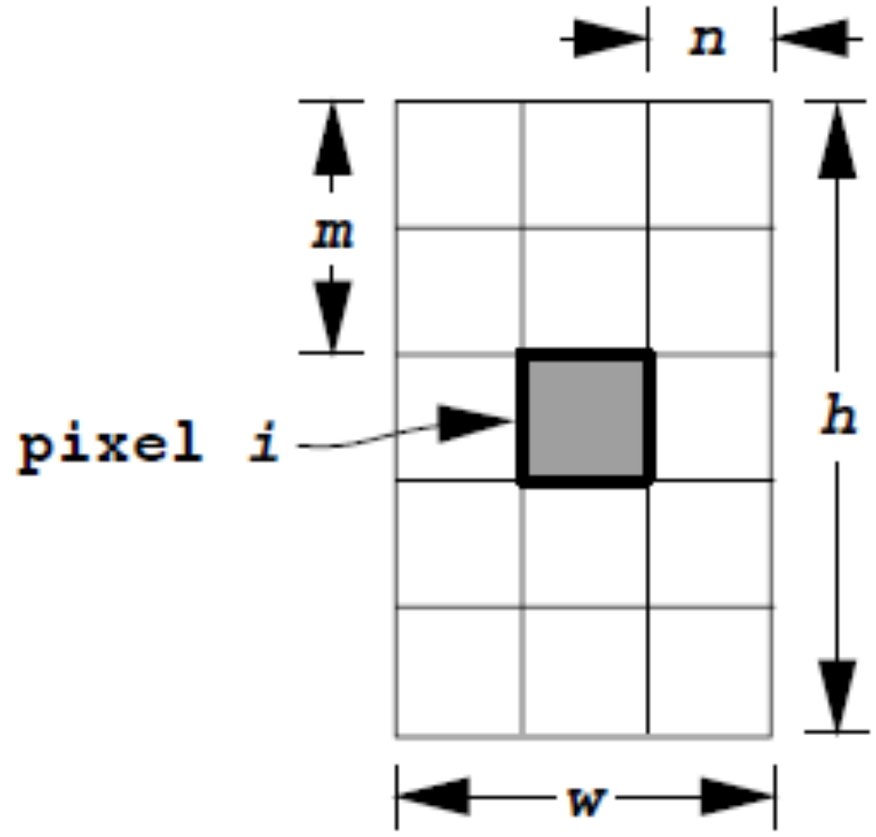
$$\hat{C}_i = f(N_i)$$

- What about if N_i is not the whole image?
- Many other powerful operations can be constructed this way
- Intuition: image information that is spatially close to each pixel is relevant, but further away is less meaningful
- These are commonly called **spatial filters** because they work in the spatial domain of the image

Defining a Neighborhood

- Typically, rectangular region of width w and height h centered at pixel

- w,h are typically odd numbers (why?)



$$w = 2n + 1$$

$$h = 2m + 1$$

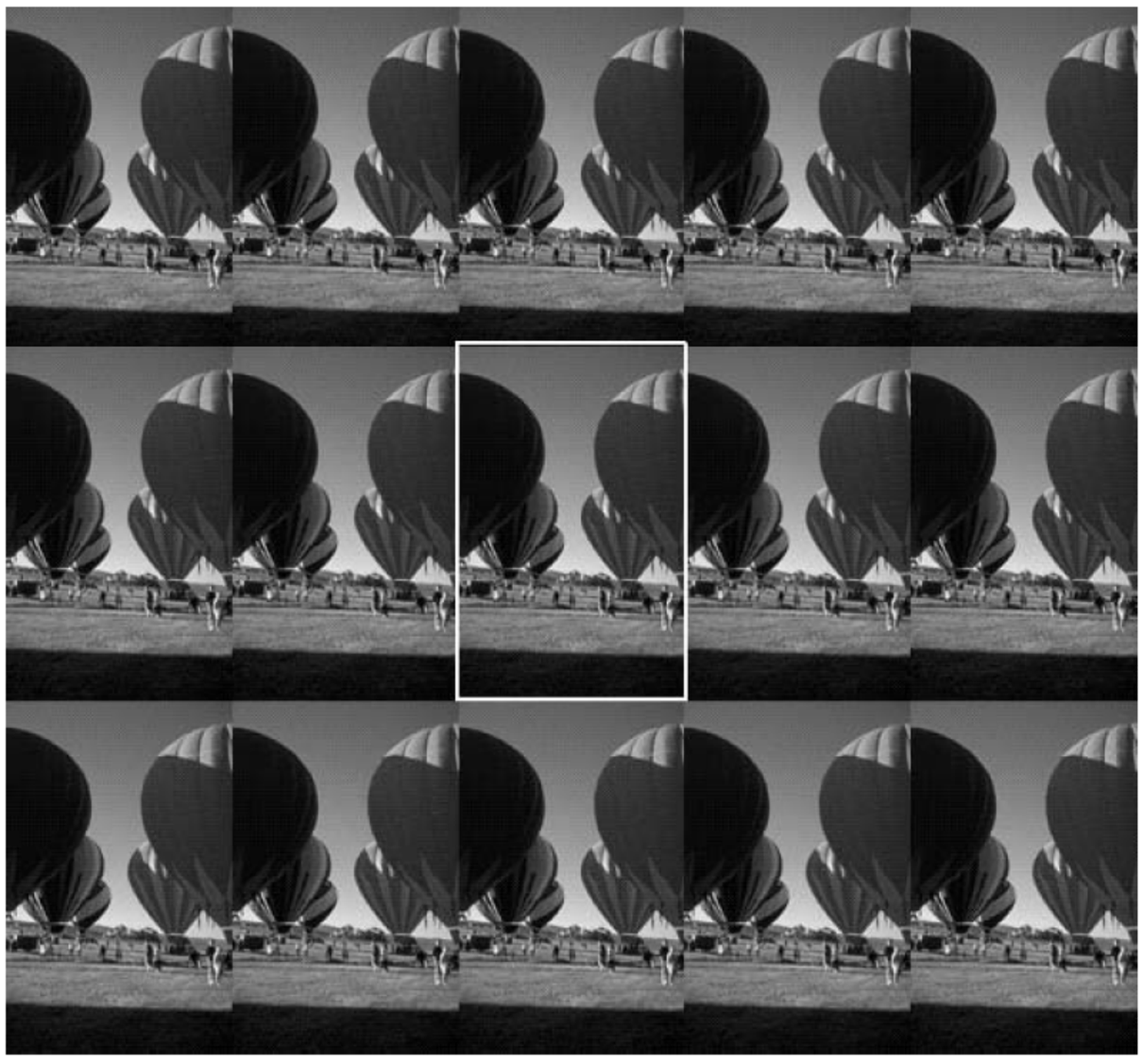
Handling Image Boundaries

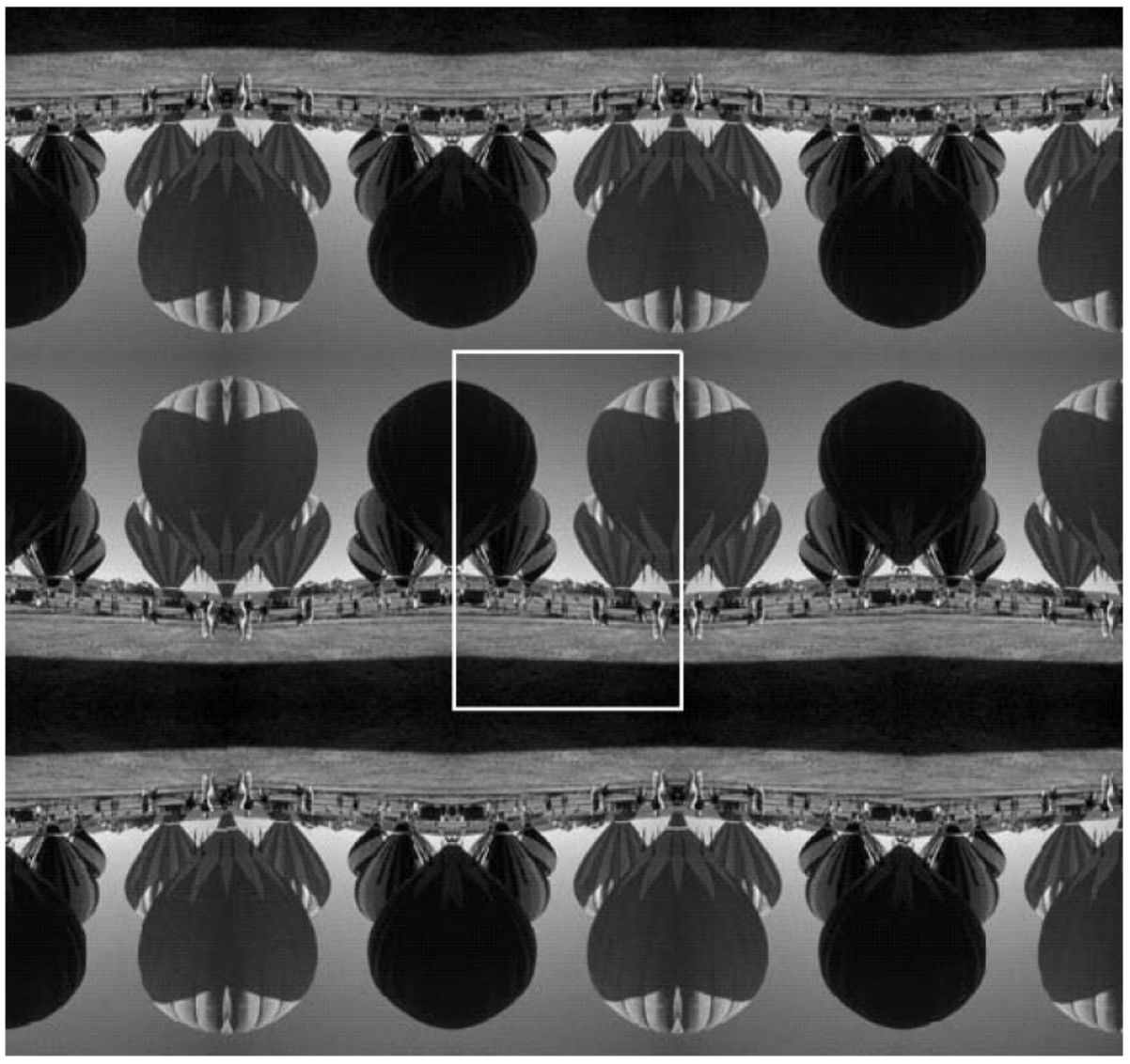
- When pixels are near the edge of the image, neighborhoods become tricky to define

- Two choices:
  - Shrink the output image (ignore pixels near the boundary)
  - Expanding the input image (create values near the boundary which are meaningful)
    - Padding, Tiling, or Mirroring

Boundary Expansion: Tiling vs. Mirroring

- Why prefer left (tiling) or right (mirroring)?





Rank Filtering

Rank Filtering

- **Rank filtering** is a nonlinear process that is based on a statistical analysis of neighborhood samples as an ordered list.
    - Frequently used to remove image noise
- The central idea is consider of all samples within the neighborhood sorted in ascending order. The term rank means “an ordering of sample values”. The rank filter examines the neighborhood and returns a value based on this list.
- The most common rank filters are given as:
    - Median: The median sample
    - Minimum: The lowest-ranked sample
    - Maximum: The highest-ranked sample

Median Filtering

- Idea: replace each pixel with the median color value in each neighborhood

- Why? Locally, pixel colors should not vary too much

- Useful for dealing with **shot** noise (aka impulse or **salt-and-pepper** noise), where individual pixels are corrupted
  - Could happen on a CCD device with defective sites

- Good at preserving straight edges, bad at preserving corners (why?)

Median Filtering (Algorithm)

- First, sort pixels in neighborhood by value
- Example:

0.3	0.2	1
0.4	0	0.2
0.2	0.3	0.1

Median Filtering (Algorithm)

- First, sort pixels in neighborhood by value
- Example:

0.3	0.2	1
0.4	0	0.2
0.2	0.3	0.1

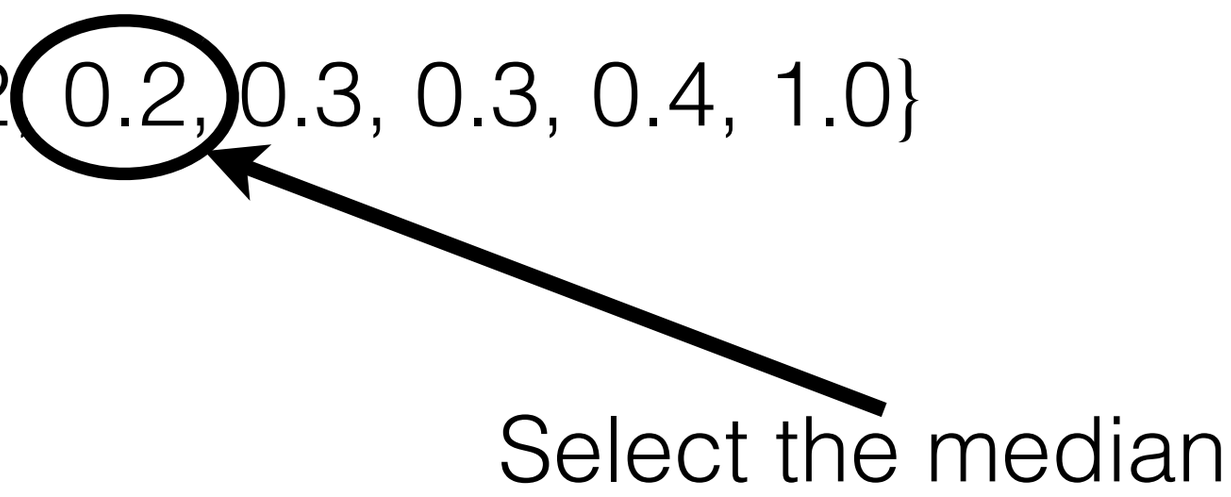
- Sorted: {0, 0.1, 0.2, 0.2, 0.2, 0.3, 0.3, 0.4, 1.0}

Median Filtering (Algorithm)

- First, sort pixels in neighborhood by value
- Example:

0.3	0.2	1
0.4	0	0.2
0.2	0.3	0.1

- Sorted: {0, 0.1, 0.2, 0.2, 0.2, 0.3, 0.3, 0.4, 1.0}

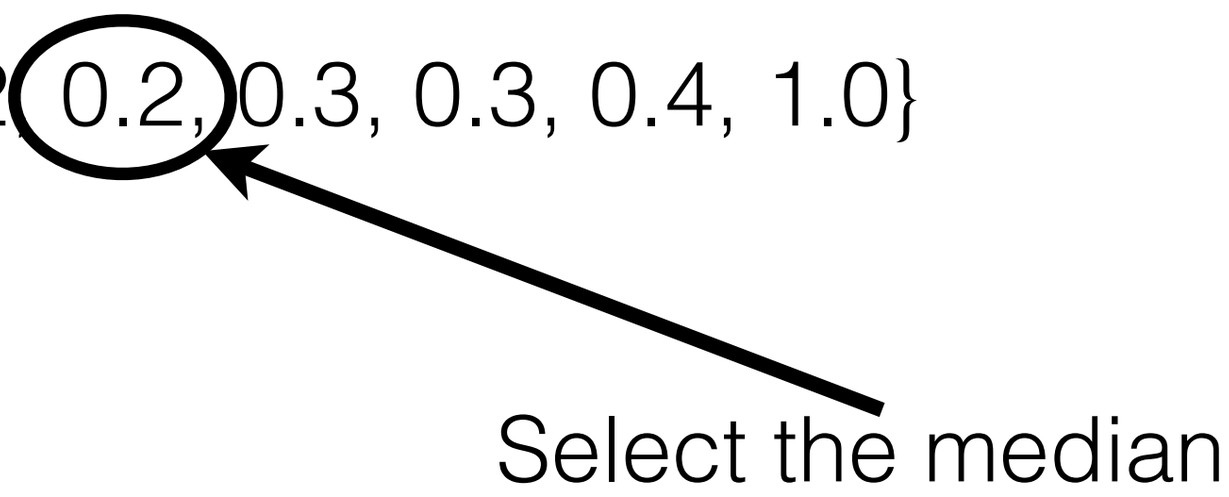


Median Filtering (Algorithm)

- First, sort pixels in neighborhood by value
- Example:

0.3	0.2	1
0.4	0	0.2
0.2	0.3	0.1

- Sorted: {0, 0.1, 0.2, 0.2, 0.2, 0.3, 0.3, 0.4, 1.0}
- Can do with colors too



Median Filtering Example

51	55	59	59	69
48	53	60	61	72
43	52	253	65	70
39	50	55	59	68
35	49	58	48	57

(a) Source image.

•	•	•	•	•
•	53	59	65	•
•	52	59	65	•
•	50	55	59	•
•	•	•	•	•

(b) Median filtered.

Figure 6.21. Numeric example of median filtering.

Noise Removal with Median Filtering

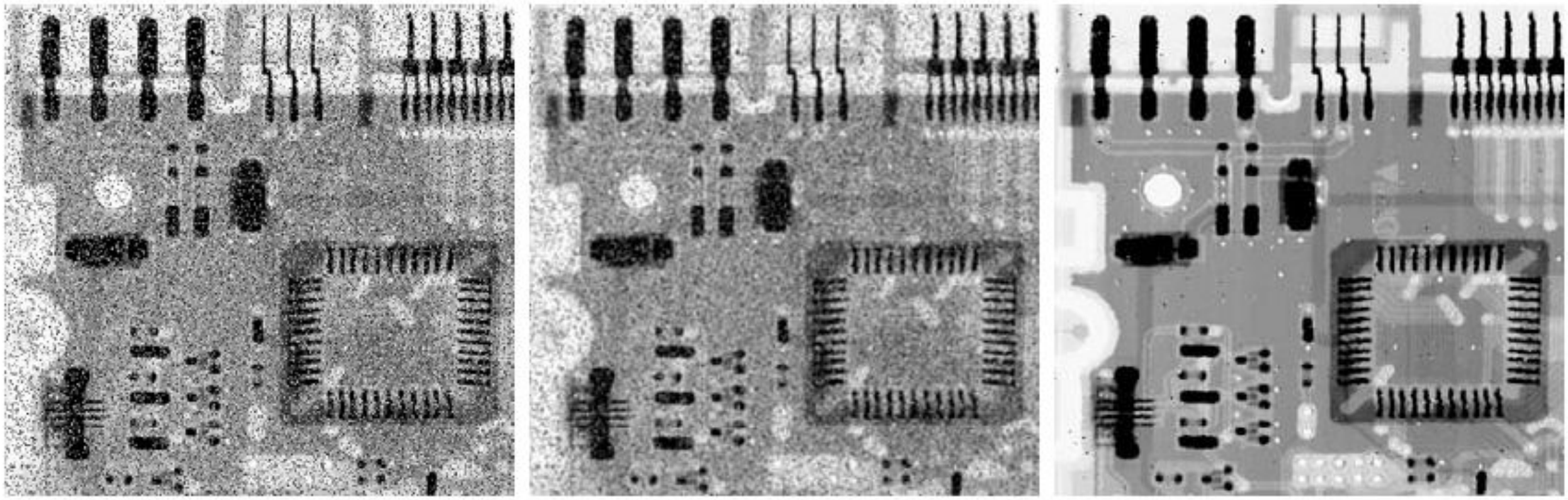
a) original image

b) with shot noise

c) after filtering

Median vs. Mean Filtering

Output = mean(N_i)



a b c

FIGURE 3.35 (a) X-ray image of circuit board corrupted by salt-and-pepper noise. (b) Noise reduction with a 3×3 averaging mask. (c) Noise reduction with a 3×3 median filter. (Original image courtesy of Mr. Joseph E. Pascente, Lixi, Inc.)

Noise Removal for Color Images

(a) Noisy source image.

(b) Median filtered.

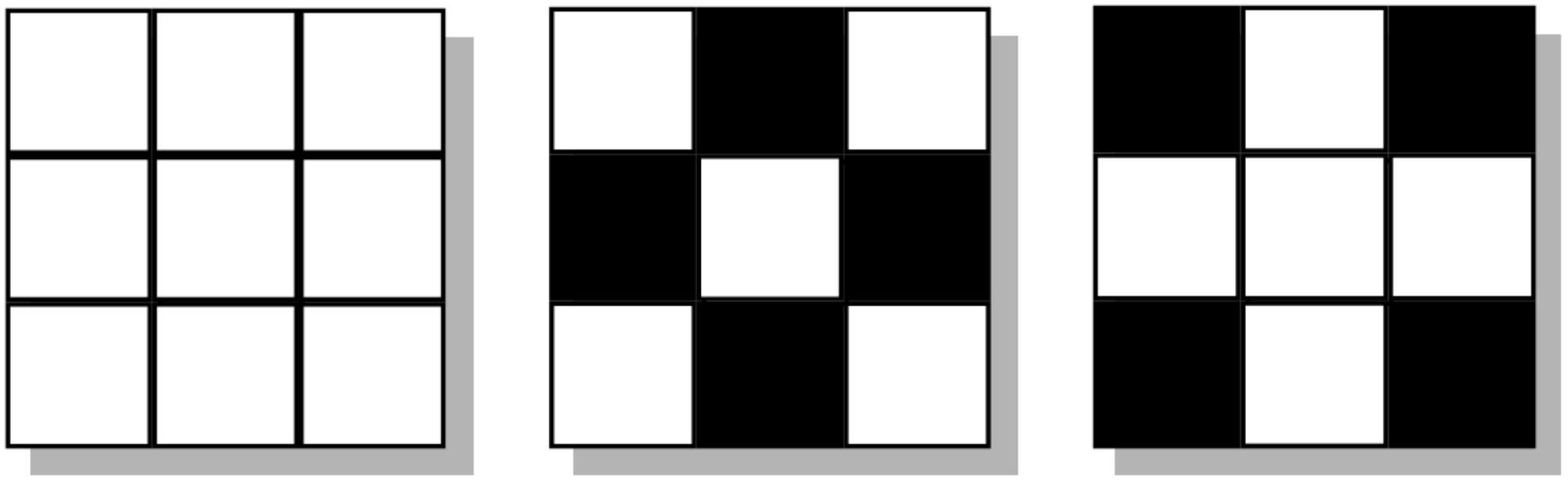
(c) Gaussian smoothed.

Figure 6.22. Median filtering.

Other Statistical Filtering

Rank Filtering w/ Masks

- Rank filtering can also use non-rectangular regions:
  - The most common of such regions are in the shape of either a plus (+) or a cross (x).
- A **mask** is used to specify non-rectangular regions where certain elements in the region are marked as either included (white) or excluded (black) from the region.



(a) Square. (b) An x mask. (c) A + mask.

Figure 6.23. Rank filtering masks for a 3×3 square, 3×3 x, and 3×3 +.

(a) Source image.

(b) 7 × 7 median.

(c) 7 × 7 maximum.

(d) 7 × 7 minimum.

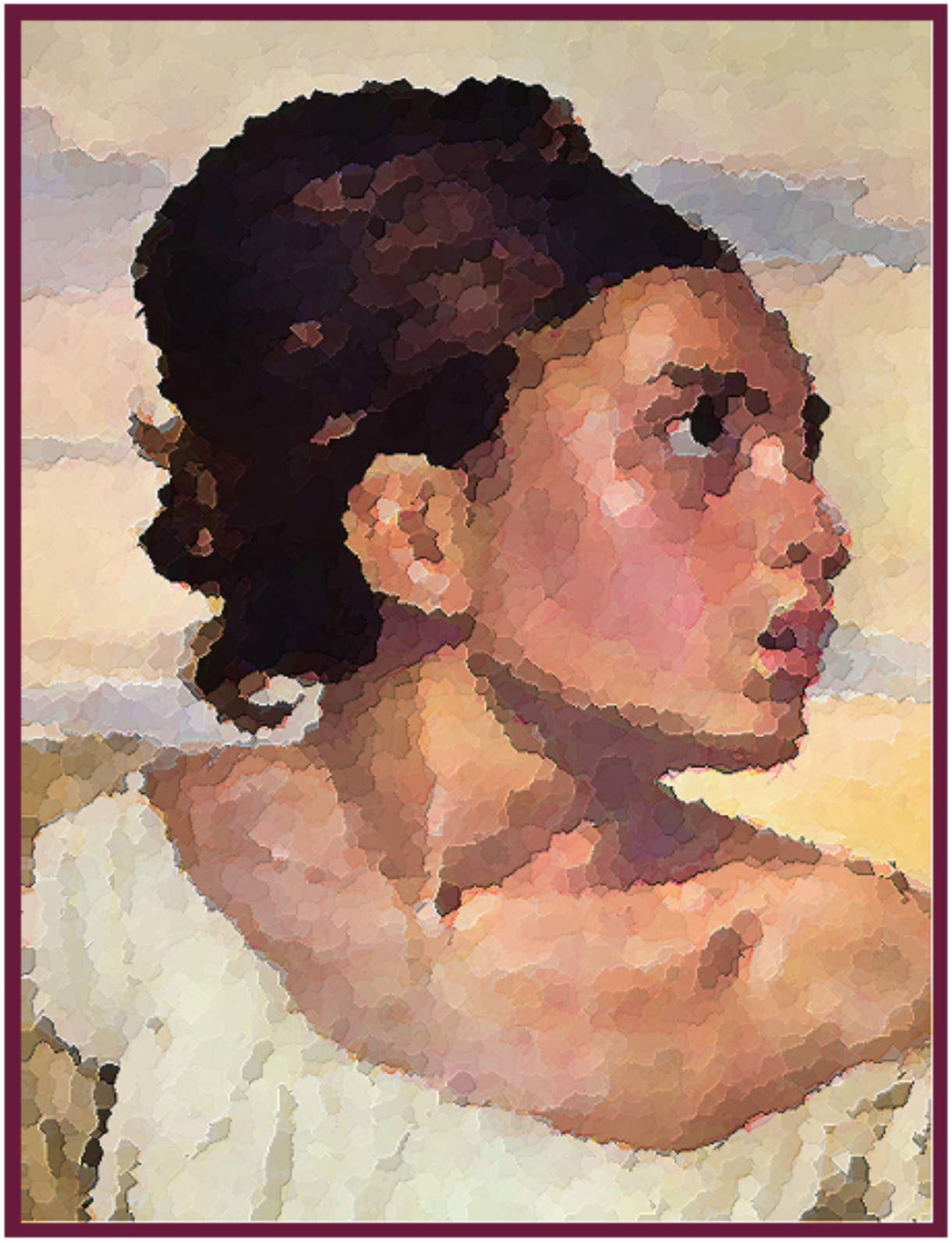
Figure 6.24. Rank filtering examples.

Min/Max Filtering

Mode and Range Filtering

- **Mode filtering** selects the most common element in the neighborhood
    - If two or more elements are most common pick the one nearest the average. Otherwise choose the darker value.
        - Using perceptual color spaces produces superior results as they introduce smaller color distortions.
- The resulting images will generally have the appearance of a mosaic or an oil painting with the size and shape of the mask determining the precise effect.

- **Range filtering** computes the difference between the maximum and minimum values of all samples in the neighborhood.
    - It generates large values where there are large differences (edges) in the source image
    - Generates small values where there is little variation in the source image.
- Could implement using image subtraction with the min and max filtered images.



(a) Most common sample.

(b) Range filtering.

Figure 6.26. Other statistically-based filters.

Approximating a Rank Filter Computation

- Computationally slow implementation. Assume an N x N rectangular region and a W x H source
  - Requires sorting N x N samples for each source sample
  - Assume that sorting N^2 samples takes time proportional to $N^2 \times \log(N^2)$
  - Filtering takes time proportional to $W \times H \times N^2 \times \log(N^2)$. Too slow!
- Can improve performance by noting that:
  - Minimum/Maximum filtering doesn't require sorting (N vs. N x log(N) time).
  - Can also improve performance by leveraging the fact that adjacent regions overlap. Use previous regions computational output to bootstrap the next region.
  - Can also use approximation based on successive N x 1 and 1 x N rank filtering.

Lec12 Required Reading

- House, 8.2.3, 8.2.4
- Hunt, 6.1-6.5