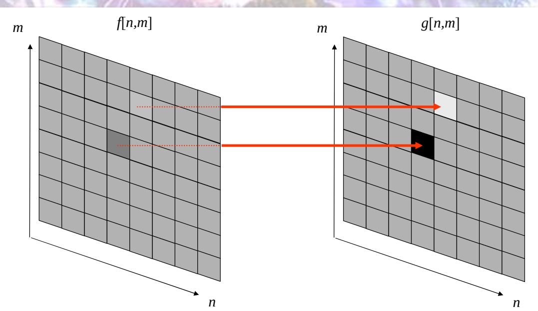
Types of DIP Operations

1. Pixel Operations
   - Output value depends on value of same pixel in input
2. Neighborhood Operations
   - Output value depends on values of pixels in neighborhood
3. Others are used, too
   - "Global" operations: output depends on values of ALL pixels
   - "Shape" operations: output depends on object shape

Pixel (or "Point") Operations



gray value g of output pixel $[n_0,m_0]$ depends **ONLY** on gray value f of the **SAME** input pixel $[n_0,m_0]$

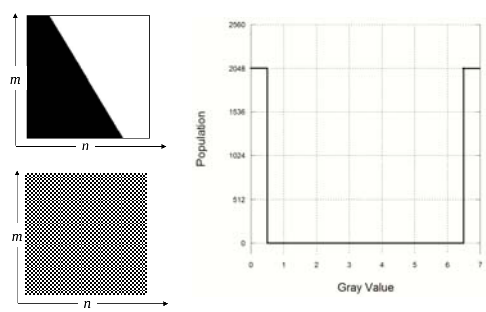
Point Operations

- Gray value g of output pixel depends only on gray value f of the corresponding input pixel
  - If one pixel with input value "10" becomes output value "100", then all inputs at "10" → "100"
- Useful for Contrast Enhancement and Image Segmentation
  - "Segmentation" ⇒ Classify pixels

Image "Histogram"

- Useful statistic of an image
- Graph of the "pixel population" for each gray value
  - "frequency of occurrence" of gray values
- Histogram of "dark" (low-contrast) image is concentrated at small grays

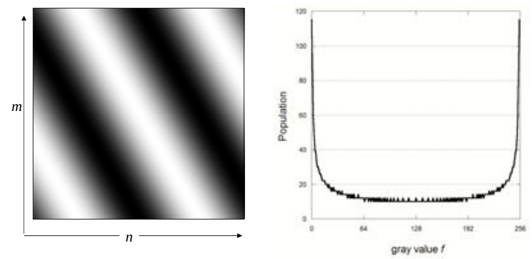
Images with Same Histogram



Lesson:

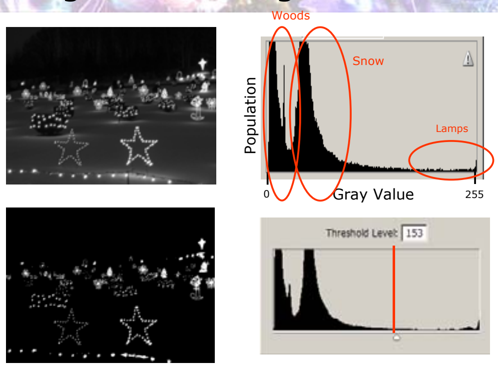
- Histograms are "statistics"
  - More than one image has the same histogram
- Histograms of many images are "bimodal"
  - two arithmetic "modes"
  - "mode" = value occuring with largest frequency = most populated level
  - Modes for "foreground" and "background" objects

Examples of Images and Histograms



Imaging Science Fundamentals

Histograms for Segmentation



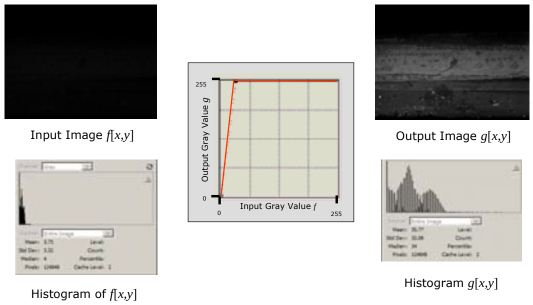
Imaging Science Fundamentals

Use of Histograms for Contrast Enhancement

- Use Histogram to find “mapping” of gray values to enhance contrast
  1. Thresholding: produces “binary” (“black-and-white” image) from input gray-scale image
  2. Stretching: “steepens” mapping
  3. Equalization: maximizes differences between image gray values

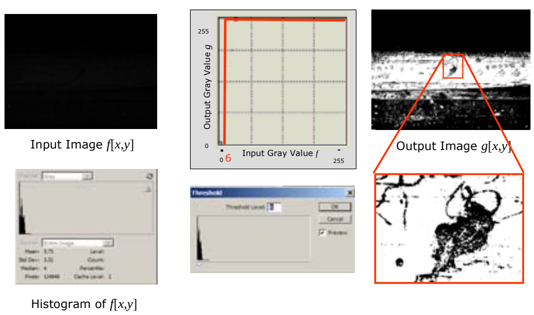
Imaging Science Fundamentals

Histogram Operations



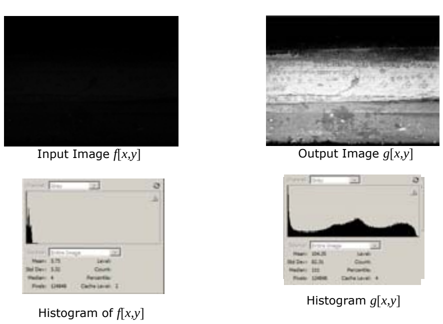
Imaging Science Fundamentals

Histogram Thresholding



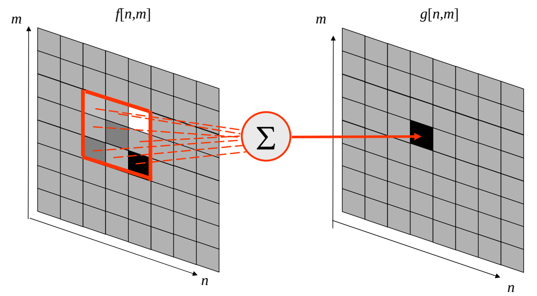
Imaging Science Fundamentals

Histogram Equalization



Imaging Science Fundamentals

Neighborhood Operations



gray value g of output pixel depends on gray values of f in the neighborhood of corresponding input pixel

Neighborhood Operations

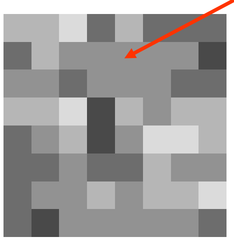
- Gray value g of one output pixel depends on the gray values of pixels in the neighborhood of the corresponding input pixel
- Useful to (1) attenuate or (2) enhance differences in gray values
  1. "noise reduction"
  2. "edge enhancement"

Noise Reduction/Removal

- "Noise" = Distribution of gray values due to statistical fluctuations in "something"
  - variation in sensitivity of pixels
  - random fluctuations

3-Bit Image with Noise

Average Value = 4
“Variation” = 1



5	5	6	3	5	3	3	3
3	5	4	4	4	4	4	2
4	4	3	4	4	4	3	3
5	5	6	2	5	4	5	5
3	4	5	2	4	6	6	5
3	3	4	3	3	5	4	4
3	4	4	5	4	5	5	6
3	2	4	4	4	4	4	3

Reduce Noise by “Local Average”

- Replace each pixel by average of 9 neighbors centered on that pixel

5	5	6	3	5	3	3	3
3	5	4	4	4	4	4	2
4	4	3	4	4	4	3	3
5	5	6	2	5	4	5	5
3	4	5	2	4	6	6	5
3	3	4	3	3	5	4	4
3	4	4	5	4	5	5	6
3	2	4	4	4	4	4	3

Replace “2” in red box by average of values of pixels in blue box

$$\frac{3+4+4+6+2+5+5+2+4}{9}=\frac{35}{9}=3\frac{8}{9}\cong 3.89$$

Round average to nearest integer

$$3\frac{8}{9}\rightarrow 4$$

Side Comment About Local Averagers

- Summation of gray values across the image
- Related to the “Integral” in Calculus, which is a summation of amplitudes across a function

Round Average Values for Display

Before Rounding

4.6	4.7	4.4	4.2	3.9	3.8	3.2	3.4
4.3	4.3	4.2	4.1	3.9	3.8	3.2	3.1
4.4	4.3	4.1	4.0	3.9	4.1	3.8	3.4
4.2	4.3	3.9	3.9	3.9	4.6	4.6	4.3
3.9	4.2	3.8	3.8	3.8	4.7	4.9	4.6
3.6	3.7	3.8	3.8	4.1	4.7	5.1	4.9
3.1	3.3	3.7	3.9	4.1	4.2	4.4	4.1
3.3	3.7	4.1	4.2	4.1	4.0	4.2	4.0

Mean Value = 4.04
Variation = 0.44

After Rounding

5	5	4	4	4	4	3	3
4	4	4	4	4	4	3	3
4	4	4	4	4	4	4	3
4	4	4	4	4	5	5	4
4	4	4	4	4	5	5	5
4	4	4	4	4	5	5	5
3	3	4	4	4	4	4	4
3	4	4	4	4	4	4	4

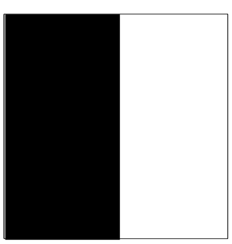
Mean Value = 4.03
Variation = 0.53

3-bit Noise Image After Averaging

5	5	4	4	4	4	3	3
4	4	4	4	4	4	3	3
4	4	4	4	4	4	4	3
4	4	4	4	4	5	5	4
4	4	4	4	4	5	5	5
4	4	4	4	4	5	5	5
3	3	4	4	4	4	4	4
3	4	4	4	4	4	4	4

Mean Value = 4
Variation = 1/3

Section of 3-bit Image of Edge



0	0	0	0	7	7	7	7
0	0	0	0	7	7	7	7
0	0	0	0	7	7	7	7
0	0	0	0	7	7	7	7
0	0	0	0	7	7	7	7
0	0	0	0	7	7	7	7
0	0	0	0	7	7	7	7
0	0	0	0	7	7	7	7

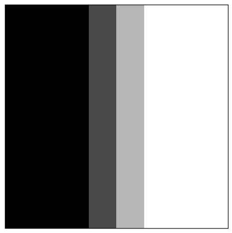
Section of 3-bit Image of Edge

Result of Average

0.0	0.0	0.0	2.3	4.7	7.0	7.0	7.0
0.0	0.0	0.0	2.3	4.7	7.0	7.0	7.0
0.0	0.0	0.0	2.3	4.7	7.0	7.0	7.0
0.0	0.0	0.0	2.3	4.7	7.0	7.0	7.0
0.0	0.0	0.0	2.3	4.7	7.0	7.0	7.0
0.0	0.0	0.0	2.3	4.7	7.0	7.0	7.0
0.0	0.0	0.0	2.3	4.7	7.0	7.0	7.0
0.0	0.0	0.0	2.3	4.7	7.0	7.0	7.0

After Rounding

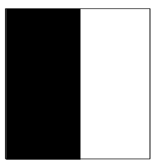
0	0	0	2	5	7	7	7
0	0	0	2	5	7	7	7
0	0	0	2	5	7	7	7
0	0	0	2	5	7	7	7
0	0	0	2	5	7	7	7
0	0	0	2	5	7	7	7
0	0	0	2	5	7	7	7
0	0	0	2	5	7	7	7



Effect of Local Average on Edge

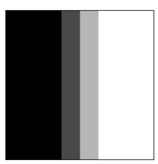
Input image has "Sharp" Edge

0	0	0	0	7	7	7	7
0	0	0	0	7	7	7	7
0	0	0	0	7	7	7	7
0	0	0	0	7	7	7	7
0	0	0	0	7	7	7	7
0	0	0	0	7	7	7	7
0	0	0	0	7	7	7	7
0	0	0	0	7	7	7	7



Output has "Fuzzy" Edge

0	0	0	2	5	7	7	7
0	0	0	2	5	7	7	7
0	0	0	2	5	7	7	7
0	0	0	2	5	7	7	7
0	0	0	2	5	7	7	7
0	0	0	2	5	7	7	7
0	0	0	2	5	7	7	7
0	0	0	2	5	7	7	7



Lesson of Local Averaging

- Local Averaging Reduces the Variation in Gray Value
  - "Pushes Values Towards the Mean Value"
- Effective for Decreasing Visibility of Noise
- Also "Blurs" Edges of Image Features

Edge Detector

- Observation: Local Averager Makes Edge "Blurry" and More Difficult to See
- How to "Detect" an Edge?
  - Can't use Local "Average"
  - Try a Local "Difference"

Local Difference

- Compute difference of gray values of adjacent pixels

"Local Difference" at Edge

0	0	0	0	7	7	7	7
0	0	0	0	7	7	7	7
0	0	0	0	7	7	7	7
0	0	0	0	7	7	7	7
0	0	0	0	7	7	7	7
0	0	0	0	7	7	7	7
0	0	0	0	7	7	7	7
0	0	0	0	7	7	7	7

Replace "0" in red box by difference of gray value in blue – red

7 – 0 = 7

Replace "7" in yellow box by of gray value in yellow – green

7 – 7 = 0

Result of “Local Difference” at Edge

Input Image $f[n,m]$

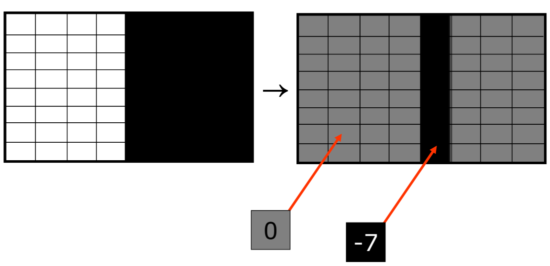
0	0	0	0	7	7	7	7
0	0	0	0	7	7	7	7
0	0	0	0	7	7	7	7
0	0	0	0	7	7	7	7
0	0	0	0	7	7	7	7
0	0	0	0	7	7	7	7
0	0	0	0	7	7	7	7
0	0	0	0	7	7	7	7

Output Image $g[n,m]$

0	0	0	0	7	0	0	0
0	0	0	0	7	0	0	0
0	0	0	0	7	0	0	0
0	0	0	0	7	0	0	0
0	0	0	0	7	0	0	0
0	0	0	0	7	0	0	0
0	0	0	0	7	0	0	0
0	0	0	0	7	0	0	0

Reversed Edge ⇒ Negative Values

7 − 0 = +7
0 − 7 = −7



Local Difference ⇒ More Grays

- Input Values: $0 \leq f \leq +7$
- Possible Output Values: $-7 \leq g \leq +7$
  - 8 levels in input, 15 levels in output
- To Display on Monitor
  - Add a constant value – the “bias” to make all values positive
  - Divide by Two to fit in original gray scale

Side Comment About Local Differencers

- Difference of adjacent gray values
- Related to "Derivative" in Calculus
  - Derivative = difference of adjacent amplitudes (values) across a function
  - Derivative is "large" where the gray values change, as at edges

Image "Sharpener"

- Combination of "averaging" and "differencing"
- Replace each pixel by average of 9 neighbors centered on that pixel

40	40	40	40	50	50	50	50
40	40	40	40	50	50	50	50
40	40	40	40	50	50	50	50
40	40	40	40	50	50	50	50
40	40	40	40	50	50	50	50
40	40	40	40	50	50	50	50
40	40	40	40	50	50	50	50
40	40	40	40	50	50	50	50

Replace "0" in red box by difference between that value and average of values of surrounding pixels in blue box

Round average to nearest integer

$$g = 40 - \frac{40+40+50+40+40+50+40+40+50}{9} = -3.333$$

$$g = 50 - \frac{40+40+50+40+40+50+40+40+50}{9} = +6.666$$

Combine Average and Difference

- One Example of an Image "Sharpener"

-1	-1	-1
-1	+9	-1
-1	-1	-1

  - multiply center pixel by 9
  - subtract sum of neighbors

Results of Sharpening

Original Image

After Averaging to “Blur”

After Sharpening

Image “Sharpener”

- Creates “steeper” edges with “overshoot” to give appearance of sharper image