THE UNIVERSITY of TEXAS
HEALTH SCIENCE CENTER AT HOUSTON
SCHOOL of HEALTH INFORMATION SCIENCES

Image Operations I

For students of HI 5323
“Image Processing”

Willy Wriggers, Ph.D.
School of Health Information Sciences

http://biomachina.org/courses/processing/03.html

1. Introduction

- There are variety of ways to classify and characterize image operations

  Reasons for doing so:

  - understand what type of results we might expect to achieve with a given type of operation
  - what might be the computational burden associated with a given operation

Types of Image Operations

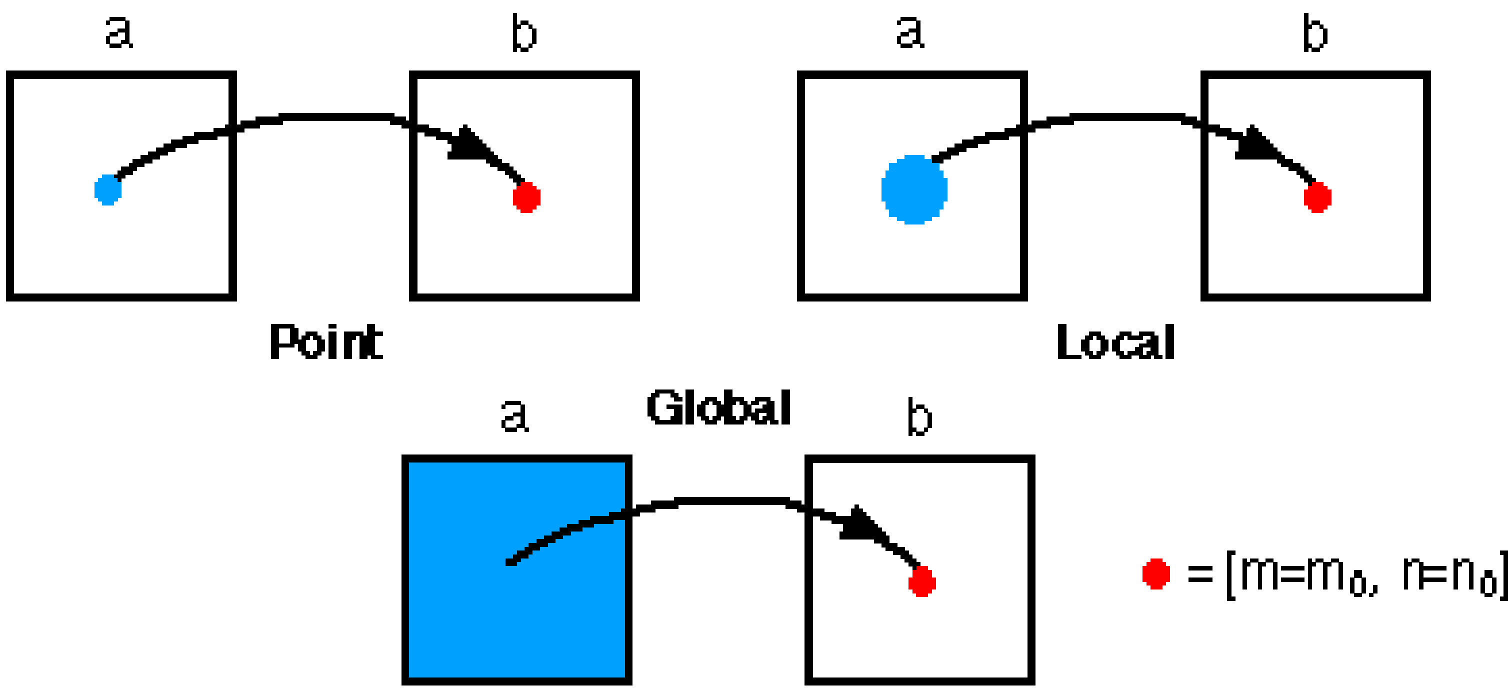
- The types of operations that can be applied to digital images to transform an input image *a*[*m*,*n*] into an output image *b*[*m*,*n*] (or another representation) can be classified into three categories

Operations	characterizations	Generic Complexity/Pixel
Point	the output value at a specific coordinate is dependent only on the input value at that same coordinate	constants
Local	the output value at a specific coordinate is dependent on the input values in the *neighborhood* of that same coordinate	P^2
Global	the output value at a specific coordinate is dependent on all the values in the input image	N^2

Types of image operations. Image size = N x N; neighborhood size = P x P

Types of Image Operations

- An input image *a*[*m*,*n*] and an output image *b*[*m*,*n*] (or another representation)



Type of Neighborhoods

- Many of the things we'll do involve using "neighboring" samples
  - "Who is my neighbor"
- Common approaches:
  - 4-connected (N, S, E, W)
  - 8-connected (add NE, SE, SW, NW)

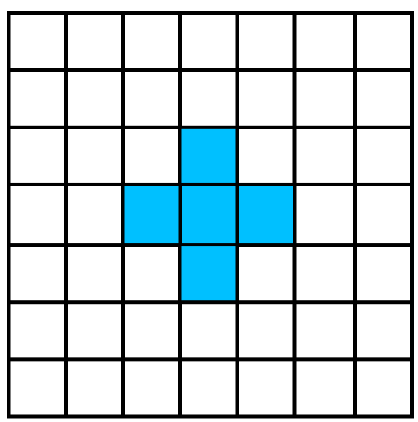
Neighborhoods

- Neighborhood operations play a key role in modern digital image processing
- It is therefore important to understand how images can be sampled and how that relates to the various neighborhoods that can be used to process an image
  - Rectangular sampling - In most cases, images are sampled by laying a rectangular grid over an image
  - Hexagonal sampling - An alternative sampling scheme

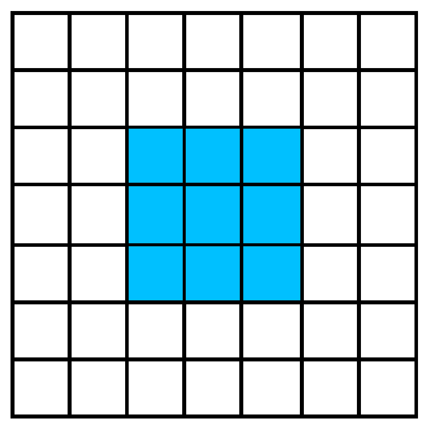
Both sampling schemes have been studied extensively and both represent a possible periodic tiling of the continuous image space

Types of Neighborhoods

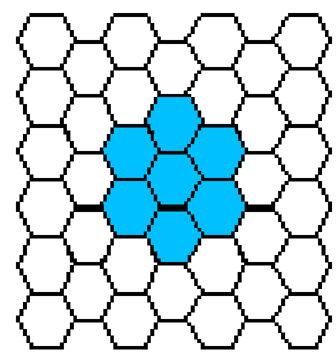
- Local operations produce an output pixel value $b[m=mo,n=no]$ based upon the pixel values in the *neighborhood* of $a[m=mo,n=no]$
- Some of the most common neighborhoods are the 4-connected neighborhood and the 8-connected neighborhood in the case of rectangular sampling and the 6-connected neighborhood in the case of hexagonal sampling



4-connected neighborhood



8-connected neighborhood



6-connected neighborhood

2. Point-Based Image Arithmetic

Image-Image Operations:

$$C[x, y] = f(A[x, y], B[x, y])$$

- Operates on each corresponding point from two (or more) images
- (Usually) requires that both images have the same dimensions: both in their domain and range

Image Addition

Used to create double-exposures

$$C[x, y] = A[x, y] + B[x, y]$$

+

=

Image Averaging

- Average multiple images (frames) of the same scene together
- Useful for removing noise

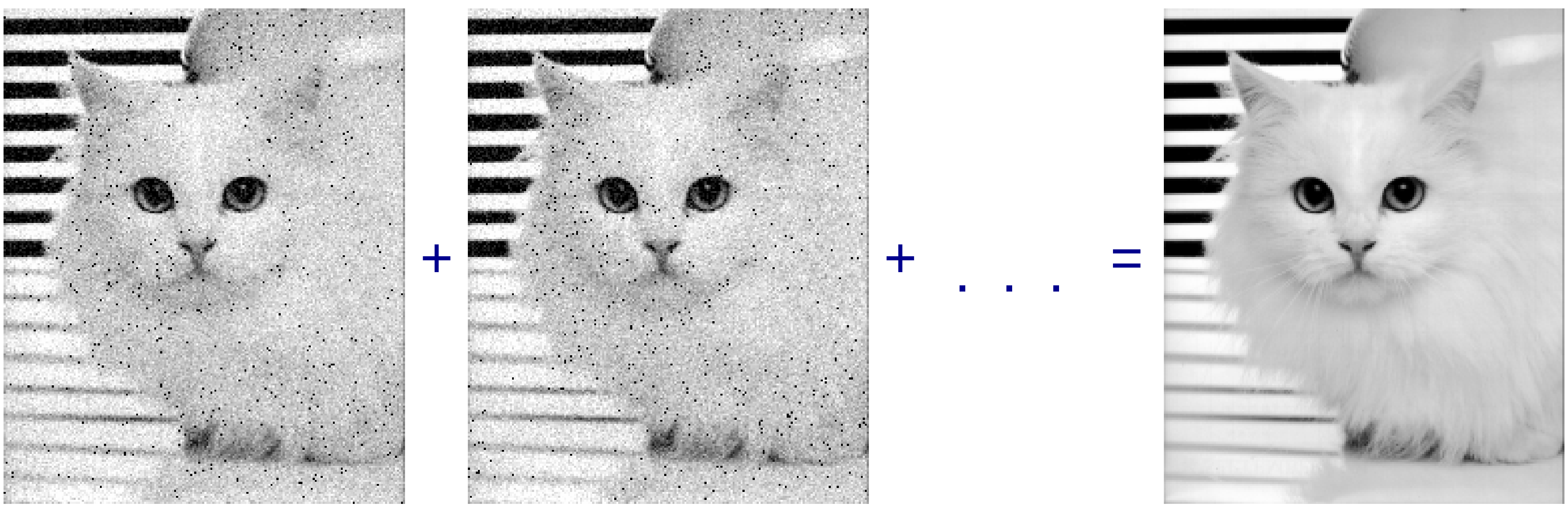


Image Subtraction

- Useful for finding changes between two images of (basically) the same scene.

$$C[x, y] = A[x, y] - B[x, y]$$

- More useful to use absolute difference

$$C[x, y] = |A[x, y] - B[x, y]|$$

Background Subtraction

"What's changed?"

(absolute difference)

Motion

- Use differencing to identify motion in an otherwise unchanging scene
- I.e.: object motion, not camera motion

Digital Subtraction Angiography

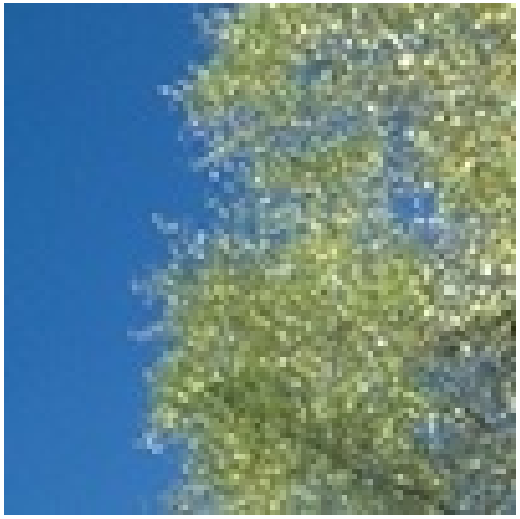
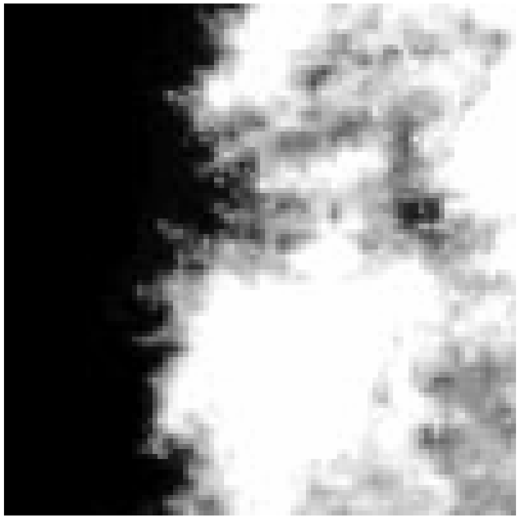
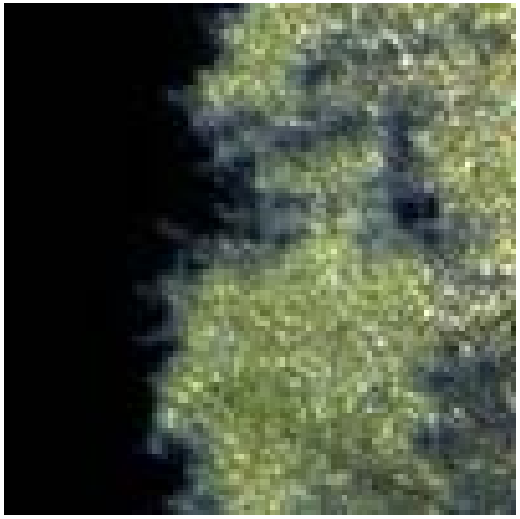
Medical imaging technique used to see blood vessels:

- Take one X-ray
- Inject a contrast agent
- Take another X-ray (and hope the patient hasn't moved, or even breathed too much)
- Subtract the first (background) from the second (background + vessels)

Multiplication

Useful for masking and alpha blending:

$$C[x, y] = A[x, y] \times B[x, y]$$


×

=


Alpha Blending

- Addition of two images, each with fractional (0..1) masking weights
- Useful for transparency, compositing, etc.
- Color images are often stored as RGBα (or RGBA)

3. Single Image Point Operations

- Simplest kind of image enhancement
- Also called *level* operations
- Process each point *independently* of the others
- Remaps each sample value:

$$g' = f(g)$$

where

- g is the input value (graylevel)
- g' is the new (processed) result
- f is a level operation

Adding a Constant

Simplest level operation:

$$f(g) = g + b$$

for some constant (bias) b

- $b > 0$ Brighter Image
- $b < 0$ Darker Image

Amplification (Gain)

Another simple level operation is amplification (multiplication):

$$f(g) = ag$$

for some constant gain (amplification) a

- $a > 1$ Amplifies signal (louder, more contrast)
- $a < 1$ Diminishes signal (softer, less contrast)

Linear Level Operators

Linear operator combine gain (multiplication) and offset (addition):

$$f(g) = ag + b$$

where

- a is the gain
- b is the bias

Negative

- Computing the "negative" of the signal/image:

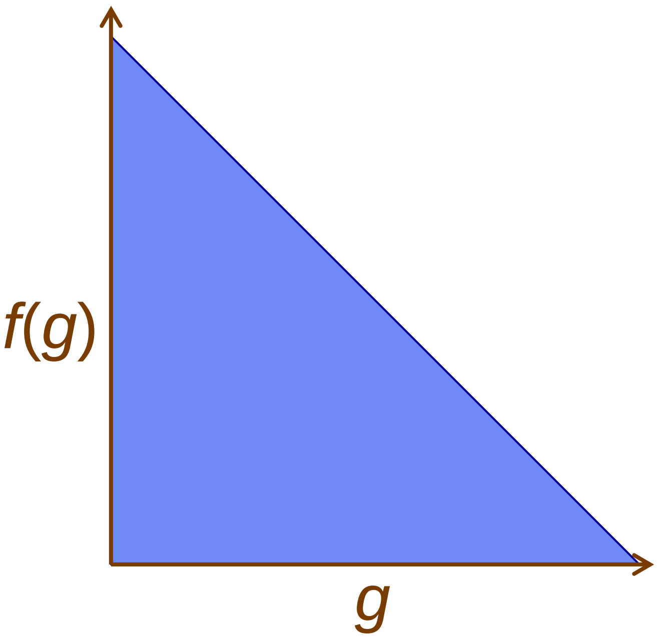
$$f(g) = -g$$

- Or, to keep the range positive:

$$f(g) = g_{max} - g$$

where $g \in [0, g_{max}]$

  - This is simply a line with slope = –1

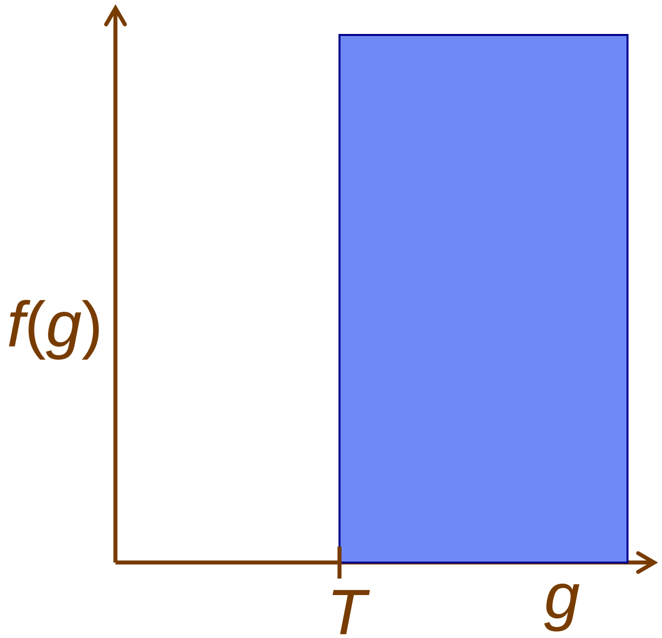


Thresholding

Thresholding a signal:

$$f(g) = \begin{cases} 0 & \text{if } g < T \\ 1 & \text{otherwise} \end{cases}$$

for some intensity threshold T



Quantization

- Quantization is choosing different finite values to represent each value of a (possibly analog) input signal
- Quantization is usually monotonic:

$$g_1 \leq g_2 \text{ implies } f(g_1) \leq f(g_2)$$

- Can be thought of as multi-level thresholding:

$$f(g) = \begin{cases} q_1 & \text{if } g_{min} \leq g < T_1 \\ q_2 & \text{if } T_1 \leq g < T_2 \\ q_3 & \text{if } T_2 \leq g < T_3 \\ \vdots & \\ q_n & \text{if } T_{n-1} \leq g < g_{max} \end{cases}$$

Logarithm

- Used to consider relative changes g_1/g_2 instead of absolute ones $g_1 - g_2$:

$$f(g) = \log(g)$$

- Useful when the dynamic range is large
- Examples:
  - Apparent brightness
  - Richter scale
  - Human vision

Exponential

Can be used to “undo” logarithmic processing:

$$f(g) = e^g$$

Contrast Enhancement

- Any time we use level operations to make one level more distinguishable from another we call it contrast enhancement
- If number of levels stays fixed, contrast enhancement trades off decreased contrast in one part of our signal range for increased contrast in a range we're interest in
- If we plot our level operation as a function:
  - All sections where the slope is greater than one increase the contrast in that intensity range
  - All sections where the slope is less than one dimish the contrast in that intensity range

Windowing

- Windowing is contrast enhancement of one part of the signal range
- Example: mapping [0, 4095] input to [0, 255] display
- The simplest mapping is:

$$f(g) = \frac{256}{4096} g = g/64$$

Windowing (cont.)

- Suppose we're interest mainly in the range [500,1200]

- Better mapping:

$$f(g) = \begin{cases} 0 & \text{if } g < 500 \\ 255(g - 500)/(1200 - 500) & \text{if } 500 \le g \le 1200 \\ 255 & \text{if } g > 1200 \end{cases}$$

Windowing is often a continuous piecewise-linear mapping

See also histogram equalization (nonlinear mapping)!

4. Segmentation

• A technique that is used to find the objects of interest.

• Segmentation can divide the image into regions – segment foreground from background.

Why?

- Industrial inspection
- Character recognition
- Tracking objects
- Geographical applications
- Medical analyses

What?

• Histogram-based - Thresholding
• Edge-based – Edge detection
• Region-base - Edge growing

• There is no universally applicable segmentation technique that will work for all images
• No segmentation technique is perfect

Thresholding

Thresholding is used to segment an image by setting all pixels whose intensity values are above a threshold to a foreground value and all the remaining pixels to a background value

If we we are interested in light objects on a dark background

if $a[m,n] \geq T$ $\quad a[m,n] = object = 1$
else $\quad a[m,n] = background = 0$

If we we are interested in dark objects on a light background

if $a[m,n] < T$ $\quad a[m,n] = object = 0$
else $\quad a[m,n] = background = 1$

Where a[m,n] is a image, T is called gray level threshold

Thresholding is simplest and most widely used method to improve the result of segmentation

Thresholding Value

- Thresholding can be viewed as an operation that involves tests against a function T of the form:

  $$T = T(x,y, p(x,y), f(x,y))$$

  Where thresholding value can be a function of position (x,y), local neighborhood p(x,y), and pixel value f(x,y)

  - Global thresholding $T = T(f(x,y))$
  - Adaptive (dynamic) thresholding $T = T(x,y, p(x,y), f(x,y))$

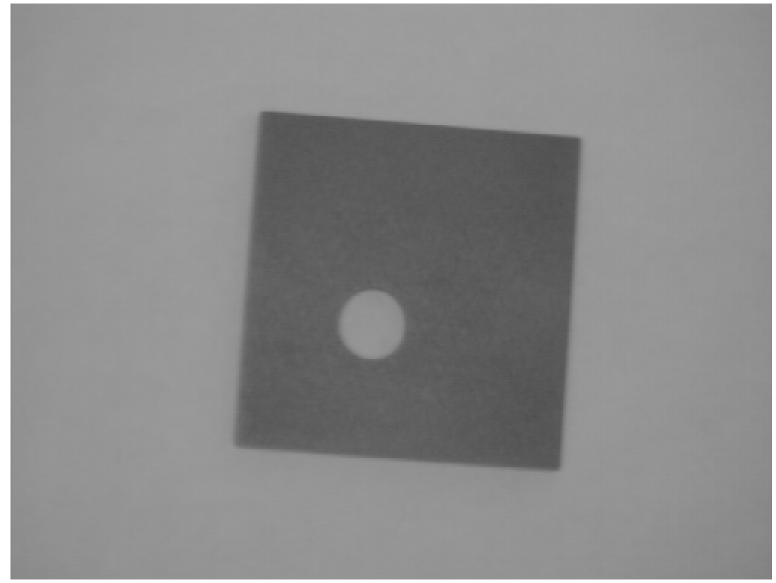
- Intensity histograms are a tool which simplify the selection of thresholds

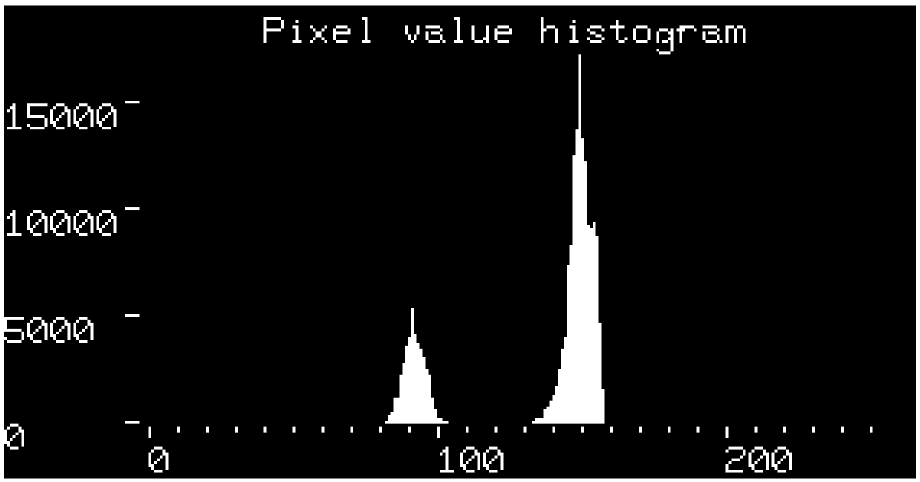
Global Thresholding

- Global threshold (constant):
  - To partition the image histogram by using a single threshold T
  - Apply on a image whose intensity histogram has distinctive peaks
- How to estimate threshold
  - Automatic: thresholding value is calculated from the histogram (assume a bimodal histogram)
  - Manual: fixed thresholding value. E.g. T = 128
- For image of simple object on a contrasting background, placing the threshold at the dip of the bimodal histogram minimizes the sensitivity of the measured area to threshold variations

Global Thresholding Example

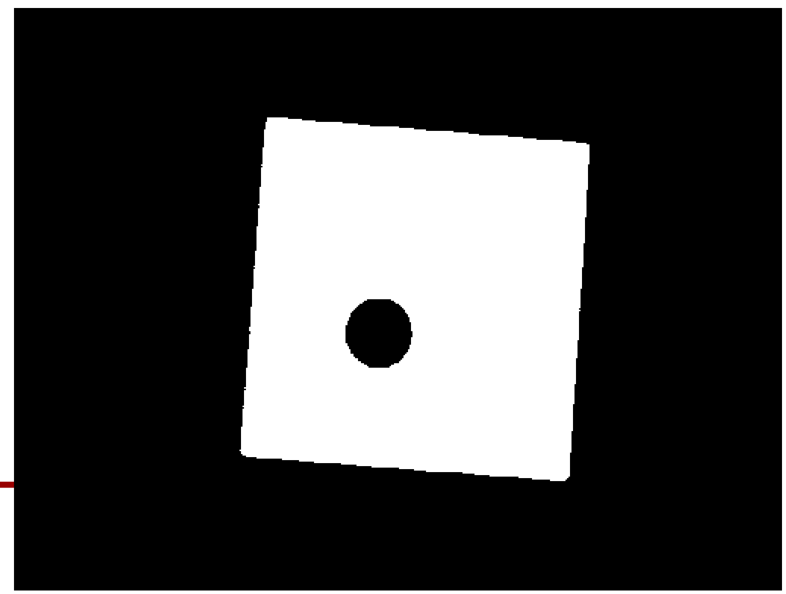
input →





T = 120

output ←



Adaptive Thresholding

- Change the threshold dynamically over the image
- Two ways to find the threshold value
  - The region-oriented thresholding
  - The local thresholding

The assumption behind both methods is that smaller image regions are more likely to have approximately uniform illumination, thus being more suitable for thresholding

Region-Oriented Thresholding

- Region-oriented thresholding
  - Divide an image into an array of overlapping subimages and then find the optimum threshold for each subimage by investigating its histogram
  - The threshold for each single pixel is found by interpolating the results of the subimages

Tradeoff - it is computational expensive

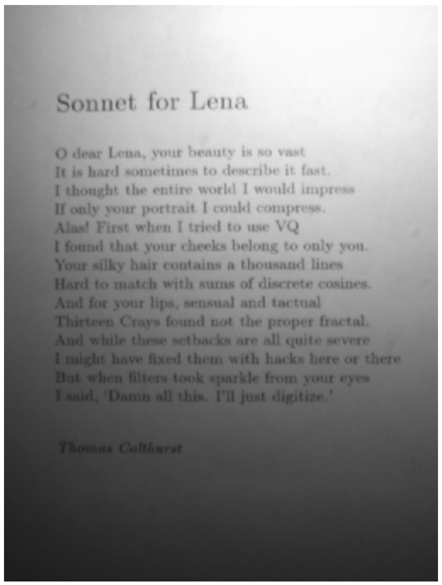
Local Thresholding

- The local thresholding
  - Statistically examine the intensity values of the local neighborhood of each pixel
  - The functions include:
    - the *mean* of the *local* intensity distribution T = mean
    - The *median* value T = median
    - The mean of the minimum and maximum values

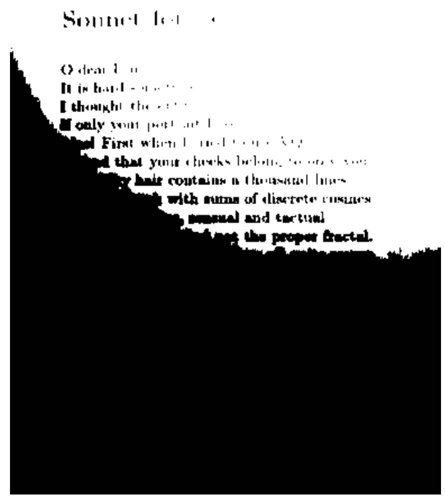
$$T = \frac{\text{minimum} + \text{maximum}}{2}$$

The size of the neighborhood has to be large enough to cover sufficient foreground and background pixels, but not too large

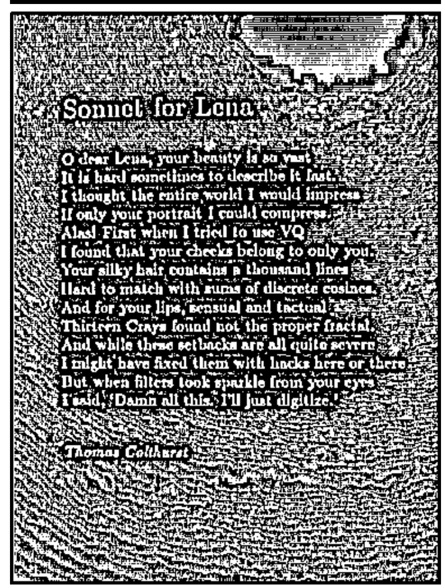
Local Thresholding Example



original



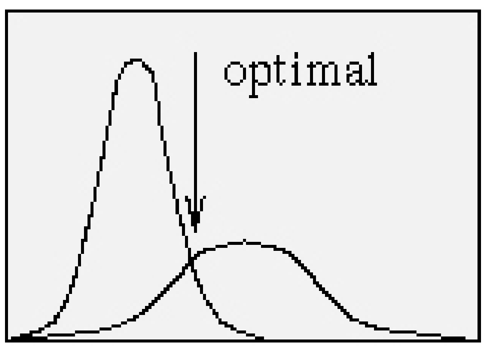
With global thresholding with T= 87

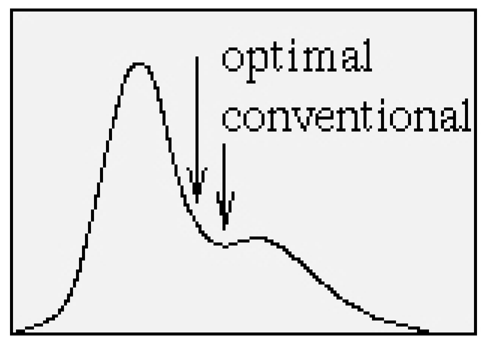


With local thresholding by mean of 7X7 neighborhood

Optimal Threshold Selection

- The method based on approximation of the histogram of an image using a weighted sum of two or more probability densities with normal distributions
- The threshold is set as the closest gray level corresponding to the probability between the maxima of two or more normal distributions, which results in minimum error segmentation





Threshold Selection

- The chances of selecting a good threshold are increased if the histogram peaks are
  - Tall
  - Narrow
  - Symmetric
  - Separated by deep valleys
- One way to improve the shape of histograms is to consider only those pixels that lie on or near the boundary between objects and the background

Resources

Textbooks:
Kenneth R. Castleman, Digital Image Processing, Chapter 6,7,18
John C. Russ, The Image Processing Handbook, Chapter 3,4,6,7

Reading Assignment

Textbooks:
Kenneth R. Castleman, Digital Image Processing, Chapter 8, 9
John C. Russ, The Image Processing Handbook, Chapter 5