Lecture 23
Segmentation: Basic Operations

MP574: Applications

Sean B. Fain, PhD (sfain@wisc.edu)
Diego Hernando, PhD (dhernando@wisc.edu)
ITK/VTK Applications: Andrew Hahn, PhD (adhahn@wisc.edu)

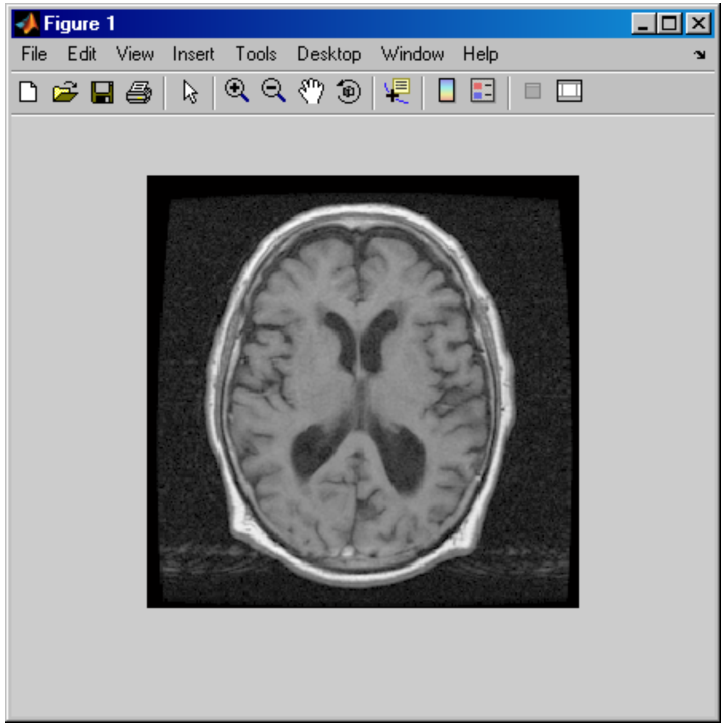
Learning Objectives

- Image Thresholding and Segmentation
  - Equalization
  - Adaptive Equalization (local kernels)
- Morphological (Structural) Operators
  - Open/Close
  - Erosion/Dilation
- Connectivity and Basic Region Growing

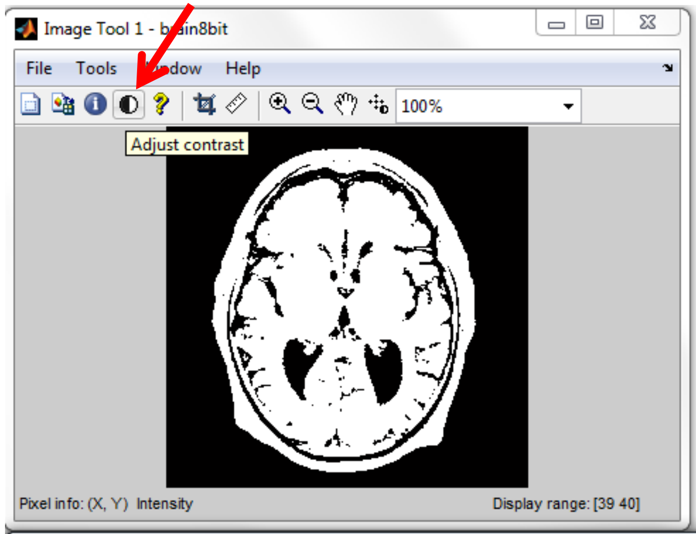
Thresholding

- Global
  - Single value applied to whole image
  - MRI of the brain

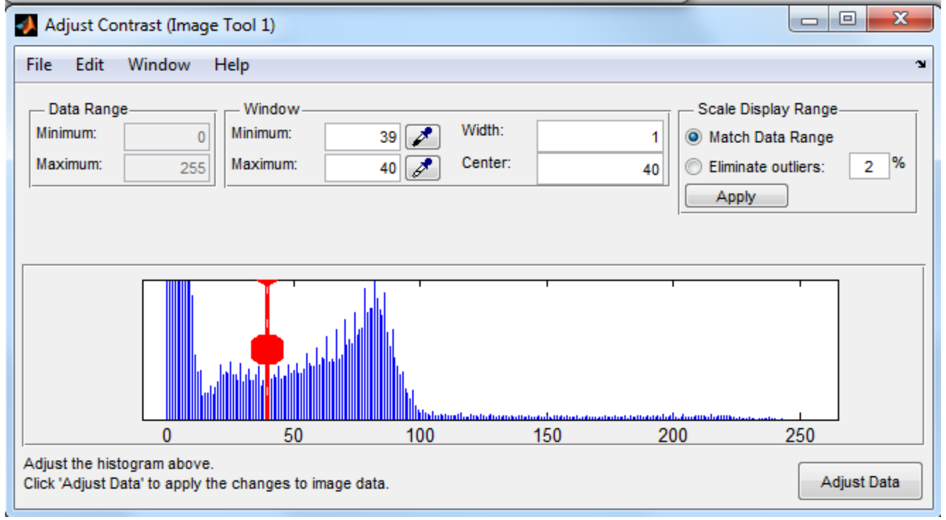
```
brain = dicomread('brain__5.dcm');
brain8bit =
      uint8(double(brain)./max(max(double(brain))).*255);
imtool(brain8bit,[0 255])
```



Bimodal Image Histogram



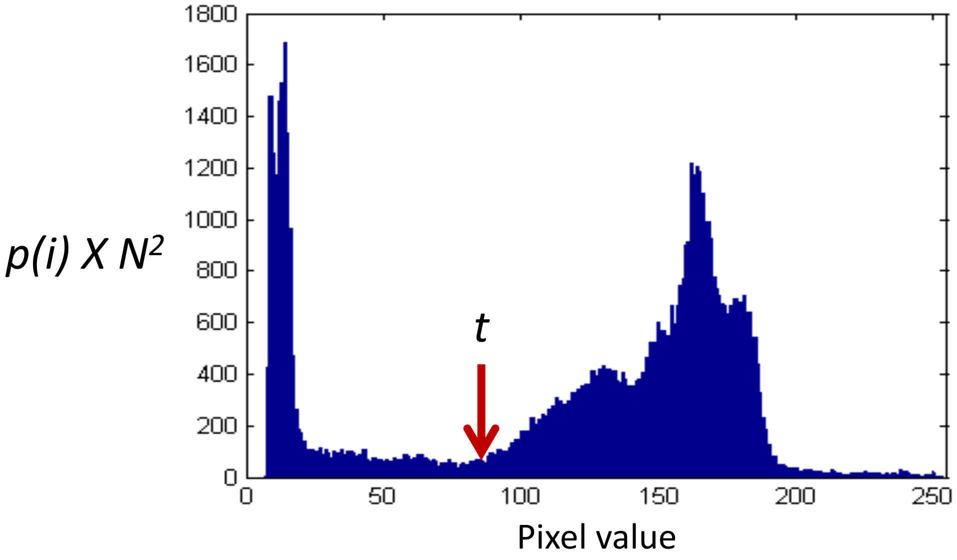
imtool

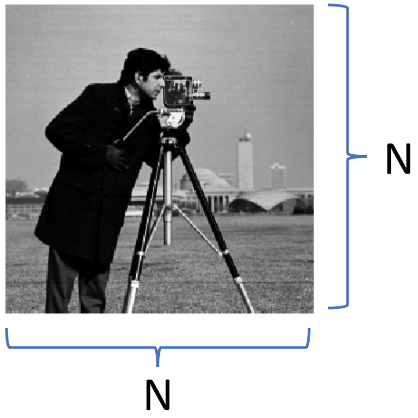


imcontrast

Otsu’s Method

- Clustering-based thresholding that converts a gray-level image to a binary image
- Search for the threshold that minimizes the intra-class variance





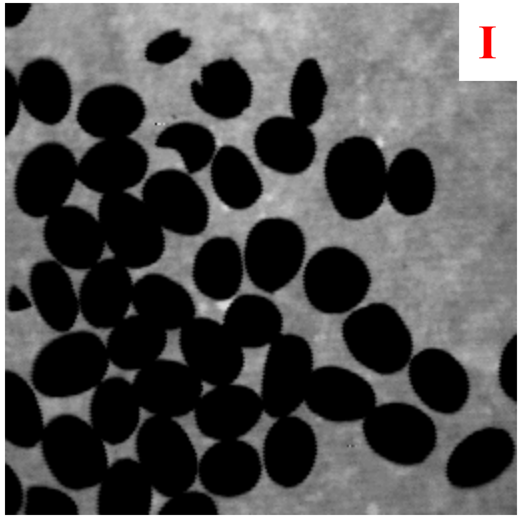
$$\sigma_w^2(t) = \omega_1(t)\sigma_1^2(t) + \omega_2(t)\sigma_2^2(t)$$

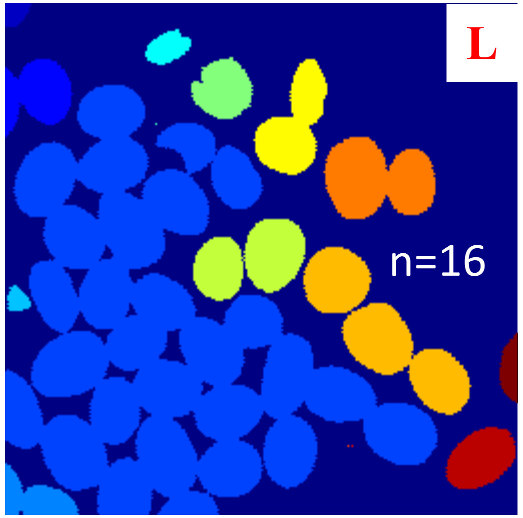
$$\omega_1(t) = \Sigma_0^t p(i) \quad \omega_2(t) = \sum_{t+1}^{L-1} p(i)$$

Basic Connectivity and Region Growing

Coffee Bean Counting

The matlab function graythresh() uses Otsu's method
Im2bw() converts to a binary image.






```
>> level = graythresh(I);
>> M = im2bw(I,level);
>> [L,n] = bwlabel(~M);
```

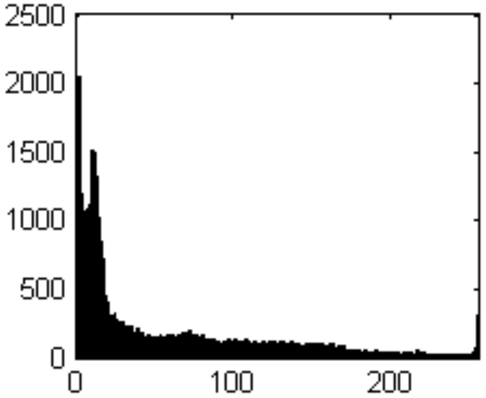
Histogram Equalization

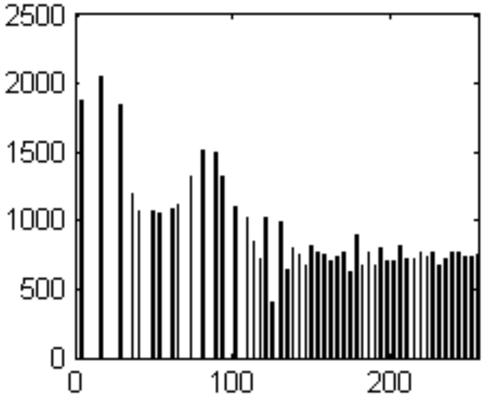
- Low contrast images: desirable to stretch the dynamic range

- Histogram equalization uses the normalized cumulative distribution function (cdf), *cdf(x)*, where x is pixel intensity:

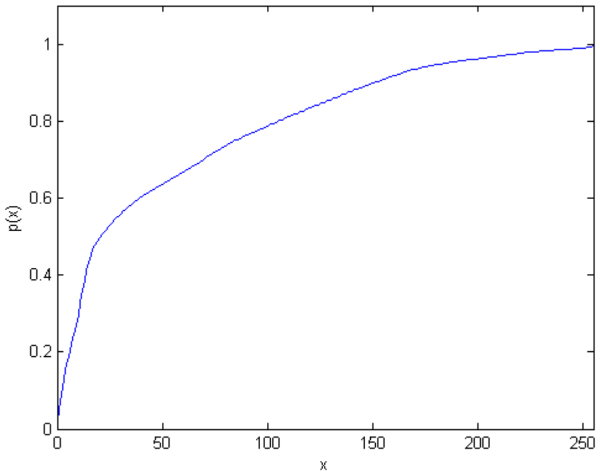
$$q = T(x) = \frac{q_{max} - q_{min}}{N^2} \sum_{i=0}^{x} H(i) + q_{min}$$

Histogram Equalization





Normalized cumulative distribution function

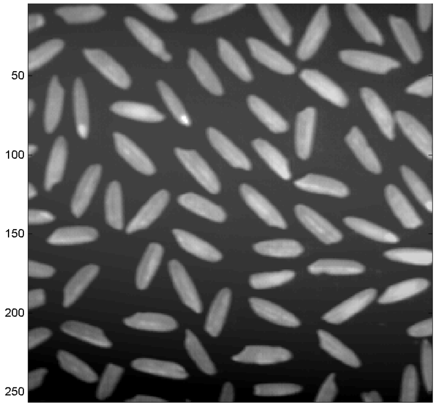


histeq()

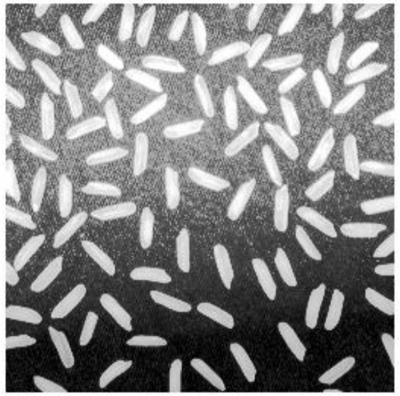
Adaptive Histogram Equalization: Correcting Low Frequency Intensity Modulations

- Break image up into N × N pixel region
- Calculate gray level histogram for each section; apply histogram equalization to each N × N region

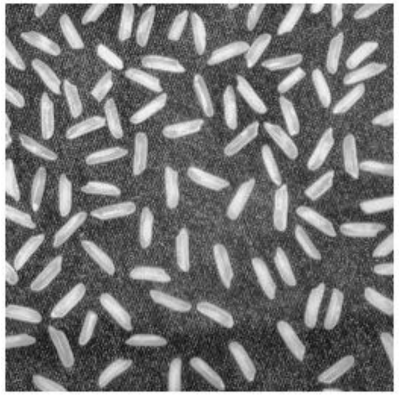
Original



Histogram Equalization

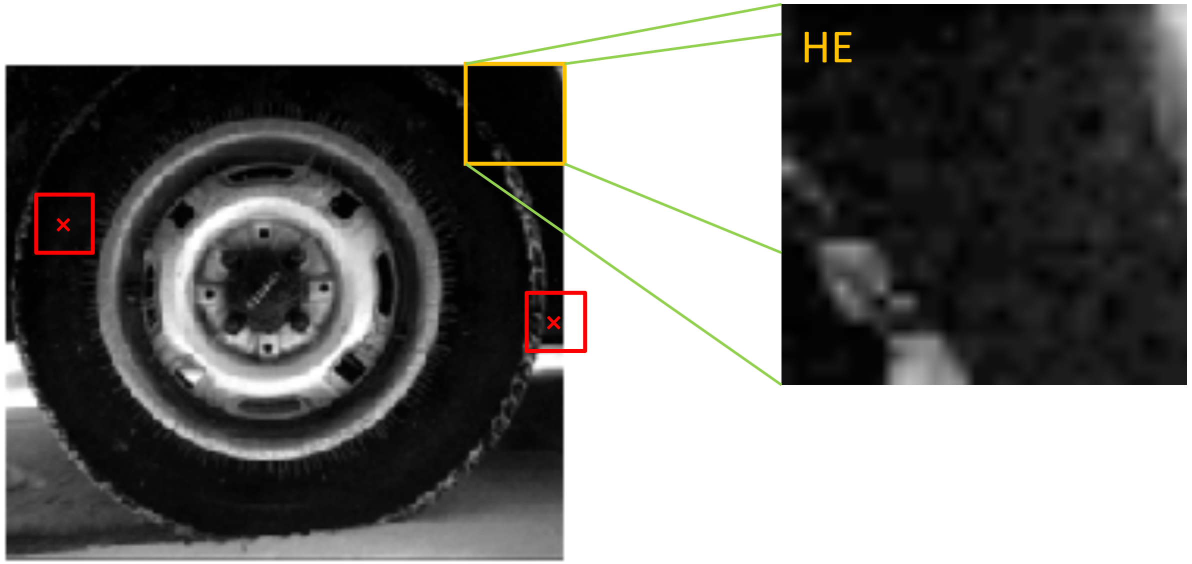


Adaptive Equalization



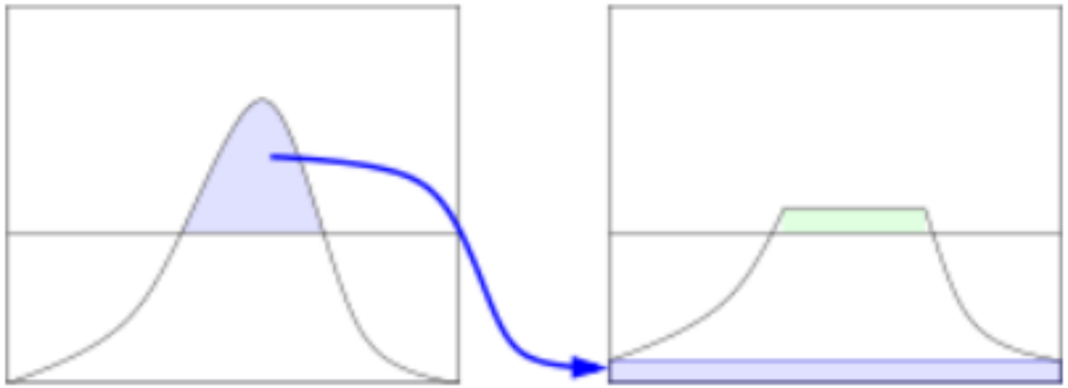
Adaptive Histogram Equalization (AHE)

- Ordinary histogram equalization works well when the distribution of pixel values is similar throughout the image.
- Adaptive histogram equalization is operated on small regions rather than the entire image.



Contrast-limited AHE (CLAHE)

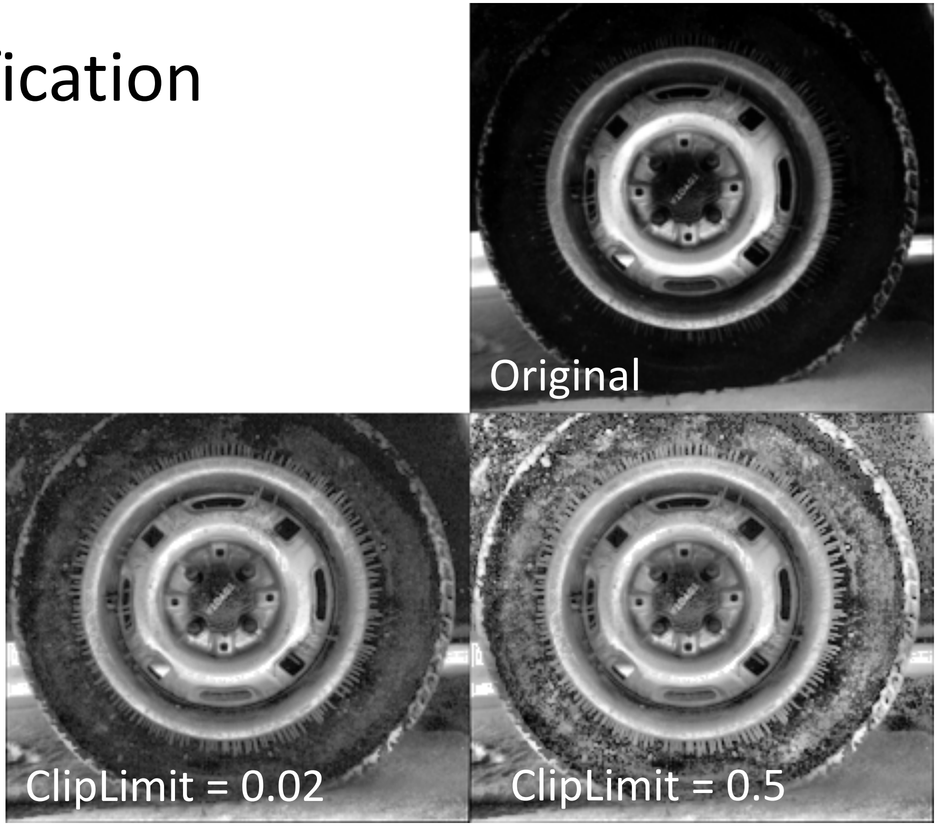
- CLAHE limits the amplification by clipping the histogram at a predefined value before computing the CDF.
- J = adapthisteq(I,'NumTiles‘,[8 8], 'ClipLimit‘,0.5)



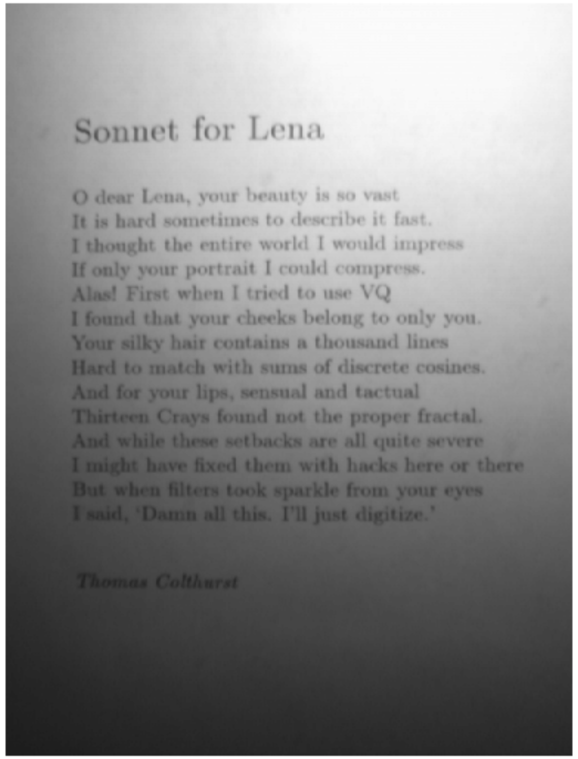
Adaptive Histogram Equalization

- Performs histogram equalization in small regions rather than the entire image
- Contrast limit can be used to limit amplification
- Matlab: adapthisteq()

```
I = imread('tire.tif');
A = adapthisteq(I,'clipLimit',0.02, …
        'Distribution','rayleigh');
A2 = adapthisteq(I,'clipLimit',0.5, …
        'Distribution','rayleigh');
figure, imshow([I, A, A2]);
```

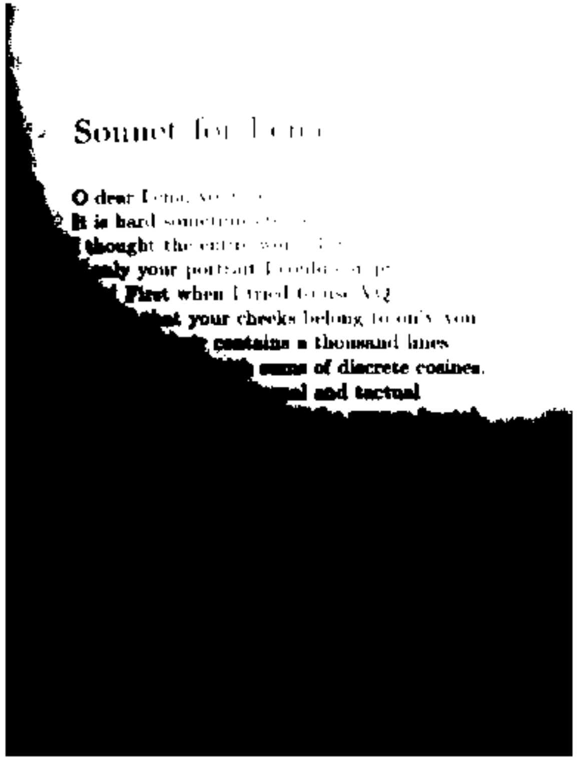


Example: Illumination Gradient


Sonnet for Lena

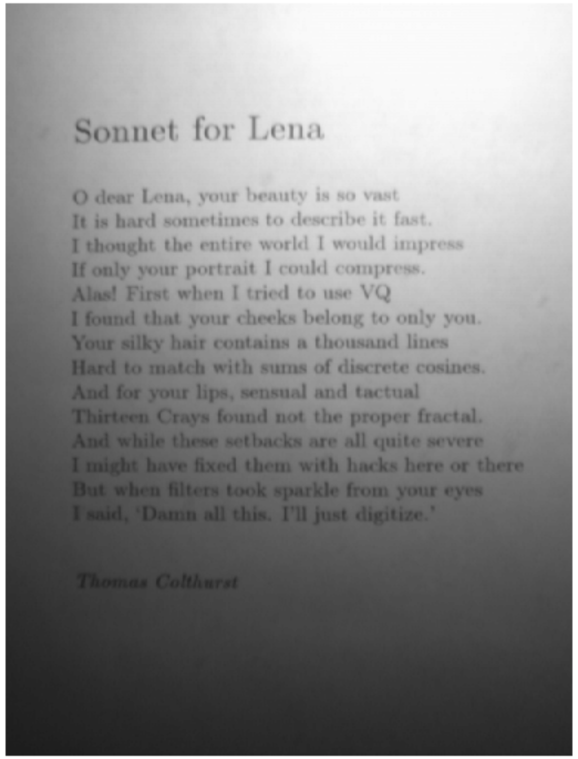
O dear Lena, your beauty is so vast
It is hard sometimes to describe it fast.
I thought the entire world I would impress
If only your portrait I could compress.
Alas! First when I tried to use VQ
I found that your cheeks belong to only you.
Your silky hair contains a thousand lines
Hard to match with sums of discrete cosines.
And for your lips, sensual and tactual
Thirteen Crays found not the proper fractal.
And while these setbacks are all quite severe
I might have fixed them with hacks here or there
But when filters took sparkle from your eyes
I said, 'Damn all this. I'll just digitize.'

Thomas Colthurst



level = graythresh(I); %Otsu's method

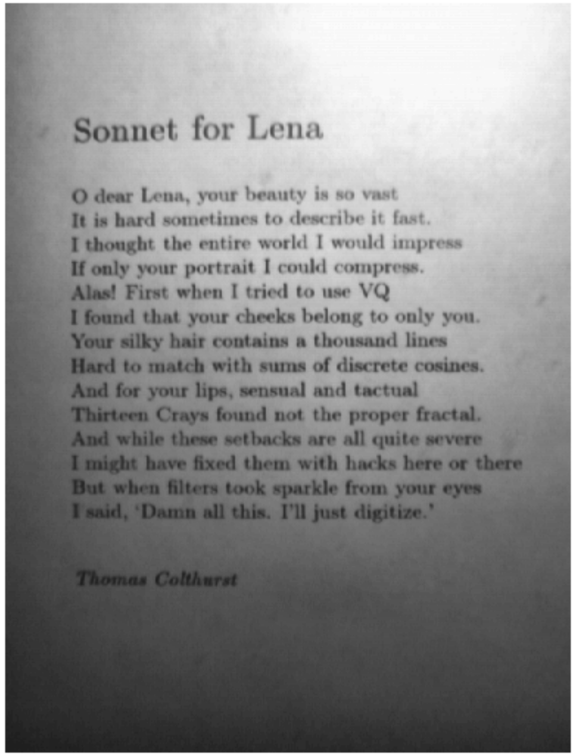
Adaptive Histogram Equalization



Sonnet for Lena

O dear Lena, your beauty is so vast
It is hard sometimes to describe it fast.
I thought the entire world I would impress
If only your portrait I could compress.
Alas! First when I tried to use VQ
I found that your cheeks belong to only you.
Your silky hair contains a thousand lines
Hard to match with sums of discrete cosines.
And for your lips, sensual and tactual
Thirteen Crays found not the proper fractal.
And while these setbacks are all quite severe
I might have fixed them with hacks here or there
But when filters took sparkle from your eyes
I said, 'Damn all this. I'll just digitize.'

Thomas Colthurst



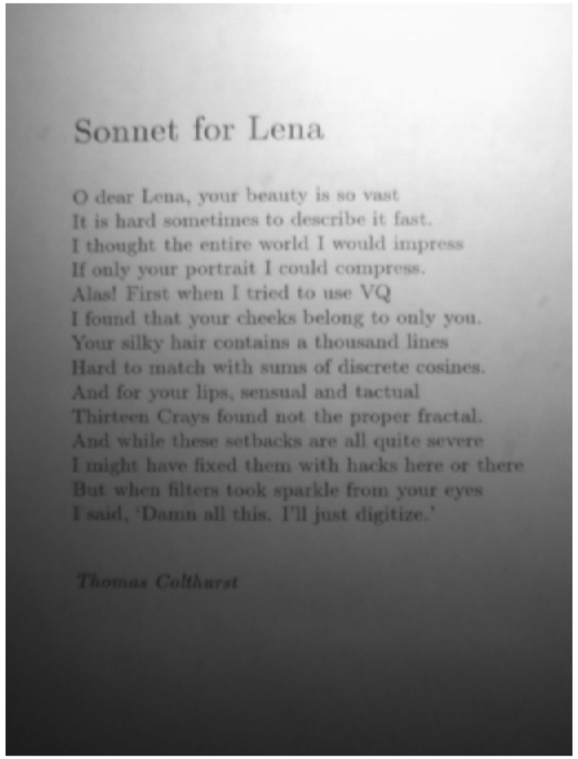
Sonnet for Lena

O dear Lena, your beauty is so vast
It is hard sometimes to describe it fast.
I thought the entire world I would impress
If only your portrait I could compress.
Alas! First when I tried to use VQ
I found that your cheeks belong to only you.
Your silky hair contains a thousand lines
Hard to match with sums of discrete cosines.
And for your lips, sensual and tactual
Thirteen Crays found not the proper fractal.
And while these setbacks are all quite severe
I might have fixed them with hacks here or there
But when filters took sparkle from your eyes
I said, 'Damn all this. I'll just digitize.'

Thomas Colthurst

```
J = adapthisteq(I); %Correct the gradient first
level = graythresh(J);
```

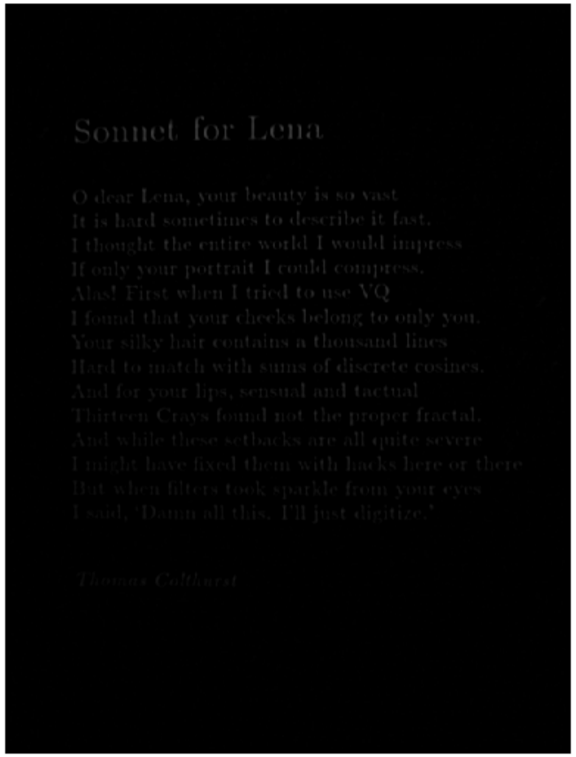
Adaptive Thresholding



Sonnet for Lena

O dear Lena, your beauty is so vast
It is hard sometimes to describe it fast.
I thought the entire world I would impress
If only your portrait I could compress.
Alas! First when I tried to use VQ
I found that your cheeks belong to only you.
Your silky hair contains a thousand lines
Hard to match with sums of discrete cosines.
And for your lips, sensual and tactual
Thirteen Crays found not the proper fractal.
And while these setbacks are all quite severe
I might have fixed them with hacks here or there
But when filters took sparkle from your eyes
I said, 'Damn all this. I'll just digitize.'

Thomas Colthurst



Sonnet for Lena

O dear Lena, your beauty is so vast
It is hard sometimes to describe it fast.
I thought the entire world I would impress
If only your portrait I could compress.
Alas! First when I tried to use VQ
I found that your cheeks belong to only you.
Your silky hair contains a thousand lines
Hard to match with sums of discrete cosines.
And for your lips, sensual and tactual
Thirteen Crays found not the proper fractal.
And while these setbacks are all quite severe
I might have fixed them with hacks here or there
But when filters took sparkle from your eyes
I said, 'Damn all this. I'll just digitize.'

Thomas Colthurst

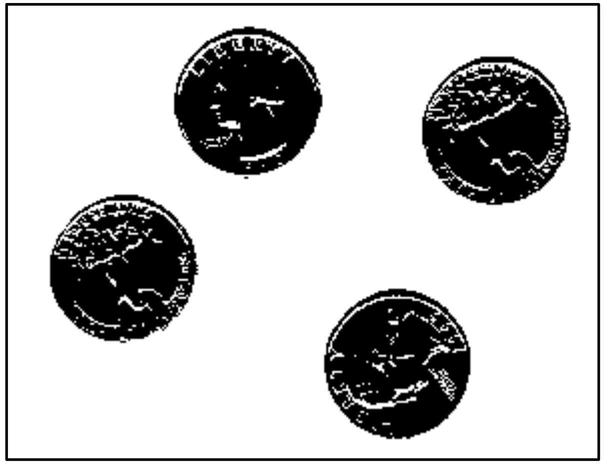
```
h = fspecial('average',8); %Creates a 2D low pass filter
F = imfilter(I,h); %Applies the filter to the image
S = F - I; %Remove the low spatial frequencies
```

Global Thresholding

- Divide image into two classes
  - Pixels with value > threshold: **object**
  - Pixels with value ≤ threshold: **background**

  Can also be reversed
- Upper and lower threshold to define a range
- Matlab: imbw() or use relational operators (>, <, …)

```
I = imread('eight.tif');
% All pixels above 128 are white
bw1 = I > 128;
% im2bw normalizes I before thresholding
bw2 = im2bw(I, 0.5);
figure, imshow([bw1 bw2]);
```



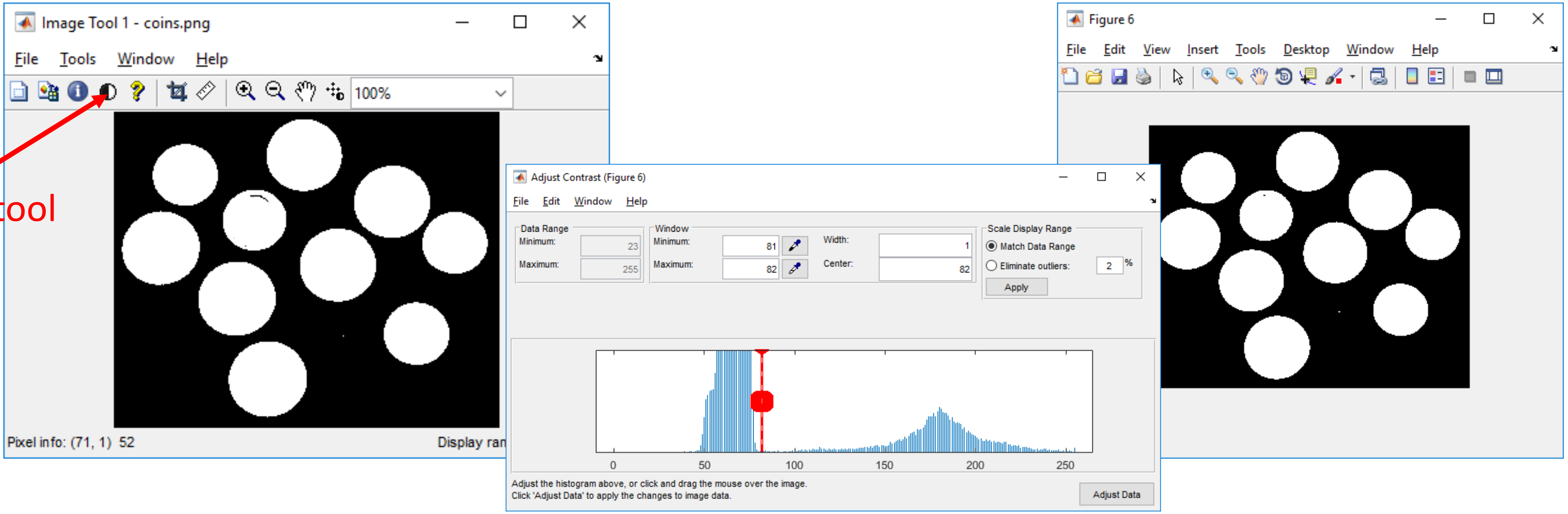
Choosing a threshold value

- Manual threshold selection: imcontrast(), imtool()

```
imtool('coins.png');
```

```
figure, imshow('coins.png');
imcontrast;
```

Contrast tool



Automatic threshold calculation (Otsu)

- Otsu's threshold determines threshold by
  - Minimizing intra-class variance
  - Maximizing inter-class variance

  equivalent
  - $\sigma_w^2(t) = \omega_0(t)\sigma_0^2(t) + \omega_1(t)\sigma_1^2(t)$
  - ω_0, ω_1 probabilities of class 1 and 2 respectively
- Matlab: *graythresh()*

```
I = mat2gray(imread('coins.png'));
tau = graythresh(I);
figure, imshow(I > tau);
```

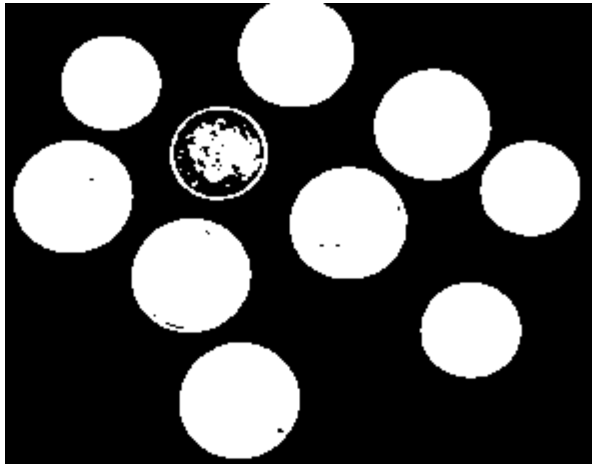
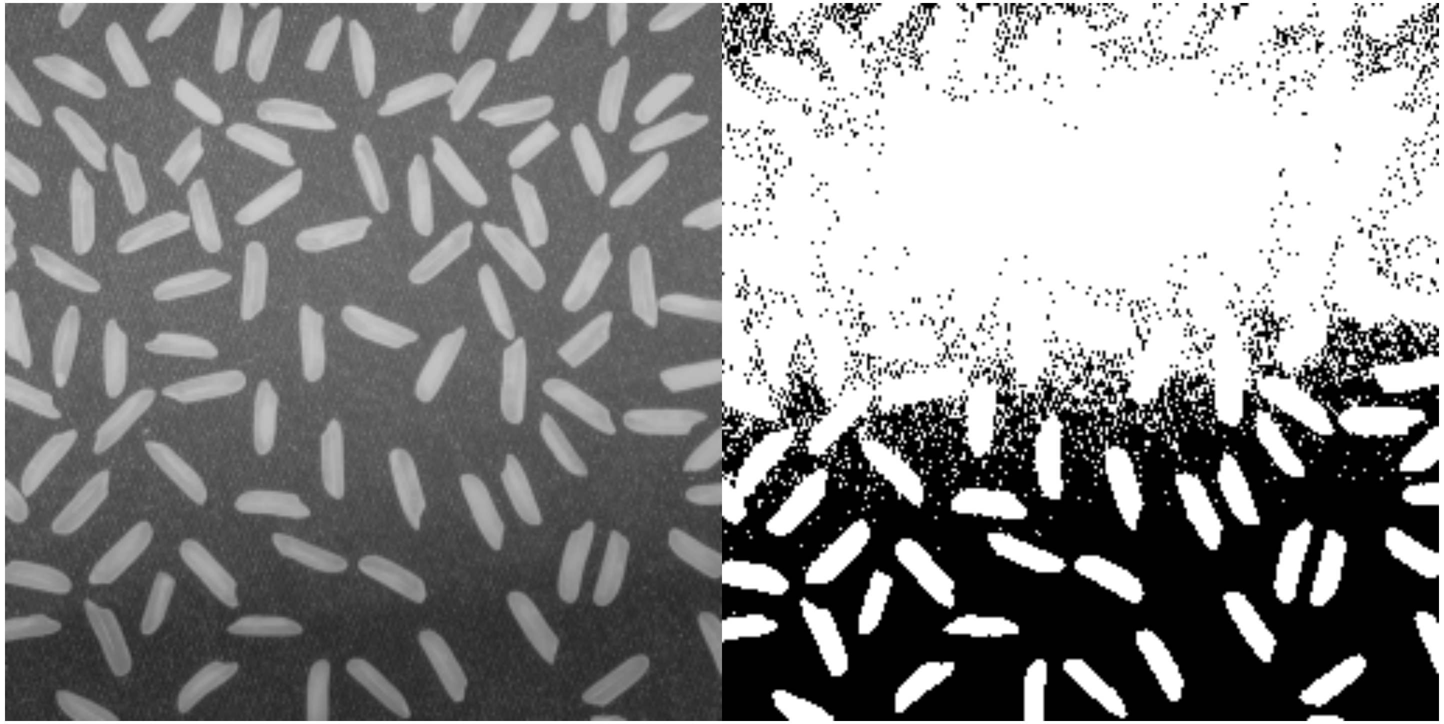


Image Inhomogeneities

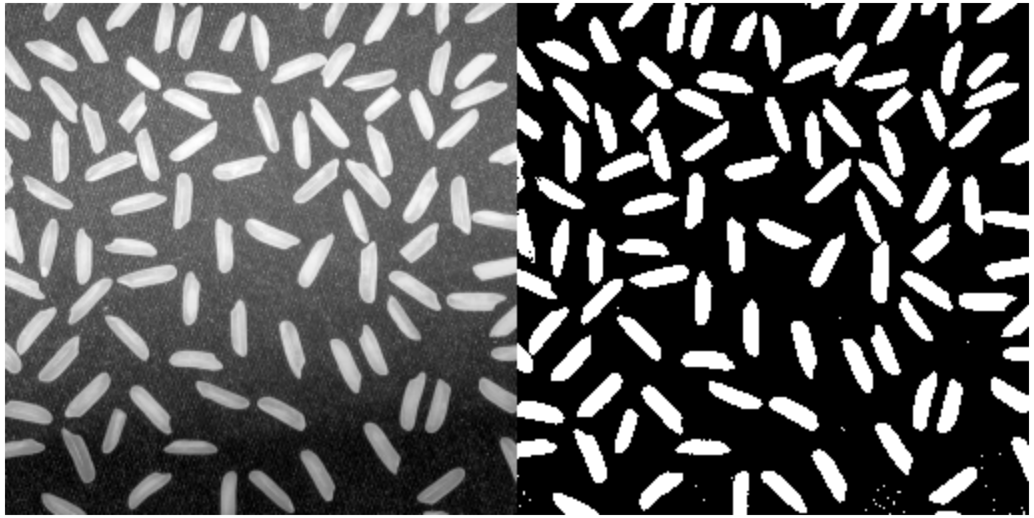
```
I = mat2gray(imread('rice.png'));
figure, imshow(I > 97);
```



Local Adaptive Thresholding

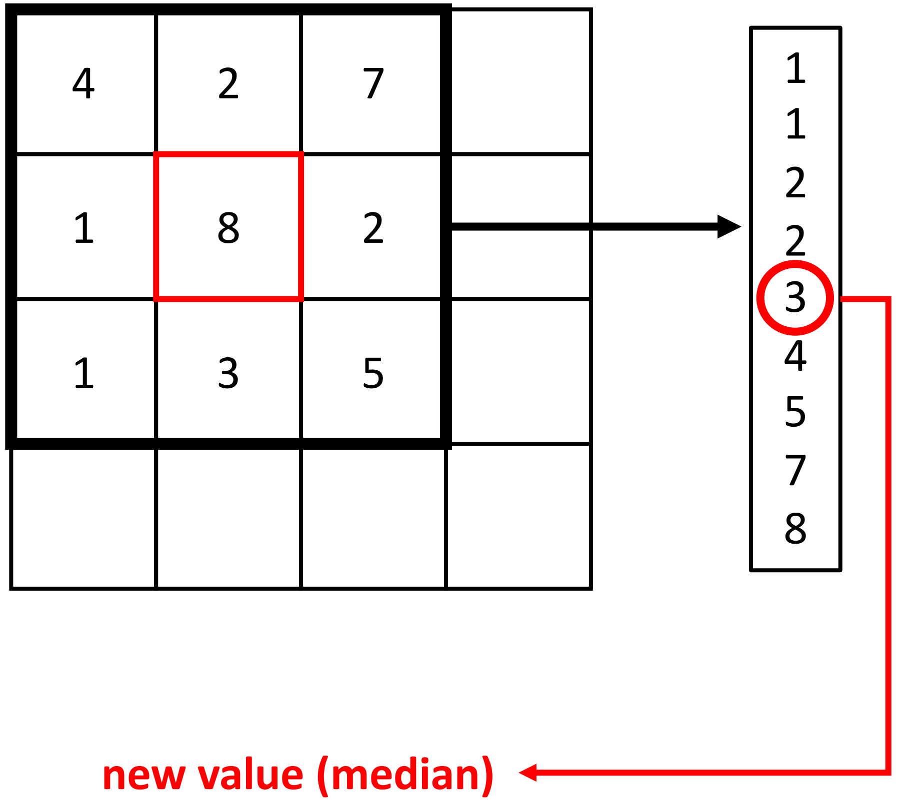
- Determines threshold for small regions in the image using local mean

```
I = imread('rice.png');
T = adaptthresh(I, 0.4);
BW = imbinarize(I,T);
figure, imshowpair(I, BW, 'montage');
```



Morphological Operators

- Non linear operations
- Sliding neighborhood blocks
- Sort pixel values within each block
- Select value based on sort order
  - Maximum (Dilation)
  - Minimum (Erosion)
  - Middle (Median)



Dilation

$$\begin{matrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{matrix}$$

S

$$\begin{matrix}
1 & 1 & 1 \\
1 & 1 & 1 \\
1 & 1 & 1
\end{matrix}$$

origin

$$\begin{matrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\
0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\
0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\
0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\
0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\
0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{matrix}$$

Erosion

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	1	1	1	1	1	1	0	0
0	0	1	1	1	1	1	1	0	0
0	0	1	1	1	1	1	1	0	0
0	0	1	1	1	1	1	1	0	0
0	0	1	1	1	1	1	1	0	0
0	0	1	1	1	1	1	1	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

S

0	1	0
1	1	1
0	1	0

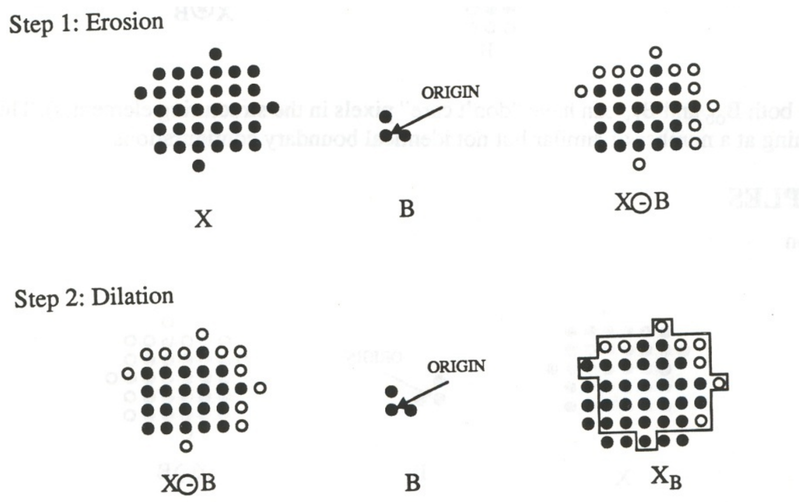
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	1	0	0	0
0	0	0	1	1	1	1	0	0	0
0	0	0	1	1	1	1	0	0	0
0	0	0	1	1	1	1	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

“Open” and “Close” Operations

- Opening: $B \circ S = (B \otimes S) \oplus S$
  - **Erosion followed by Dilation**

- Closing: $B \bullet S = (B \oplus S) \otimes S$
  - **Dilation followed by Erosion**

Example: Opening

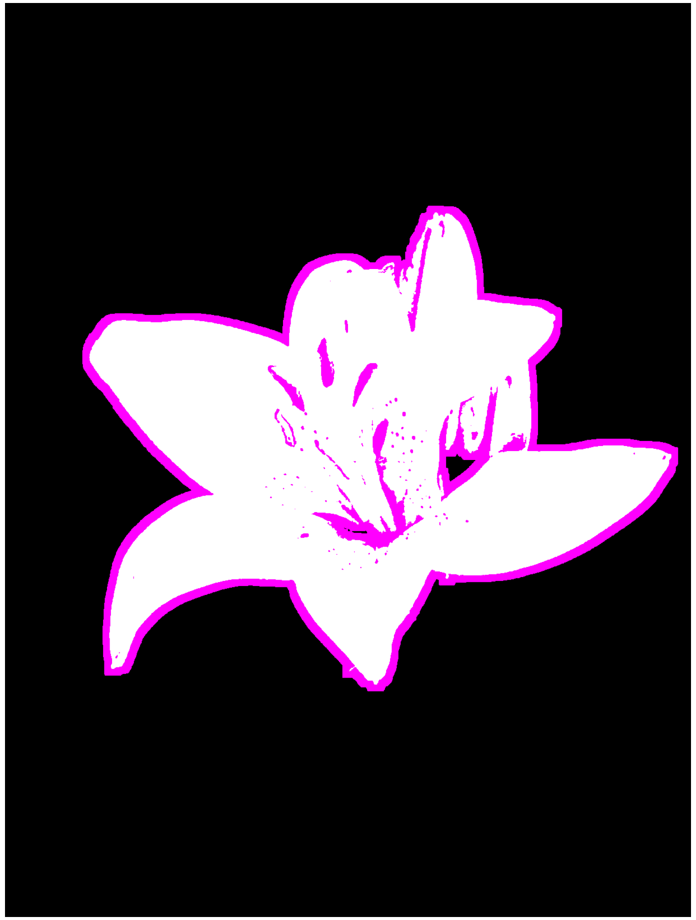


structure element (se)

Opening breaks connections, while closing mends connections

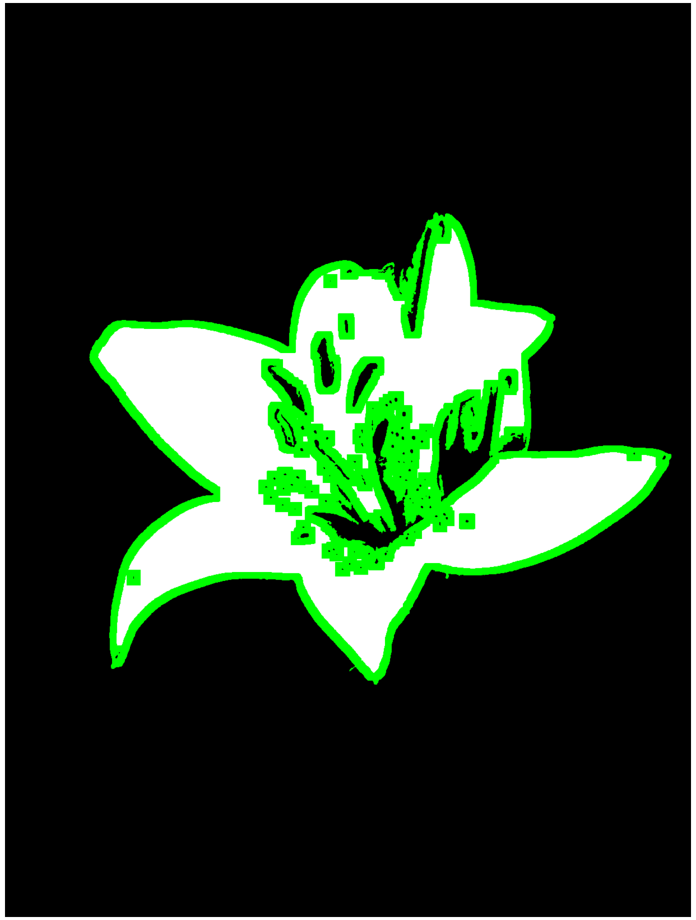
Dilation

- Choose **maximum** value in each neighborhood
- Binary images
    - **Expands** object regions
- Neighborhoods can have different shapes (e.g. diamond, disk, rectangle)
- Matlab: *imdilate()*



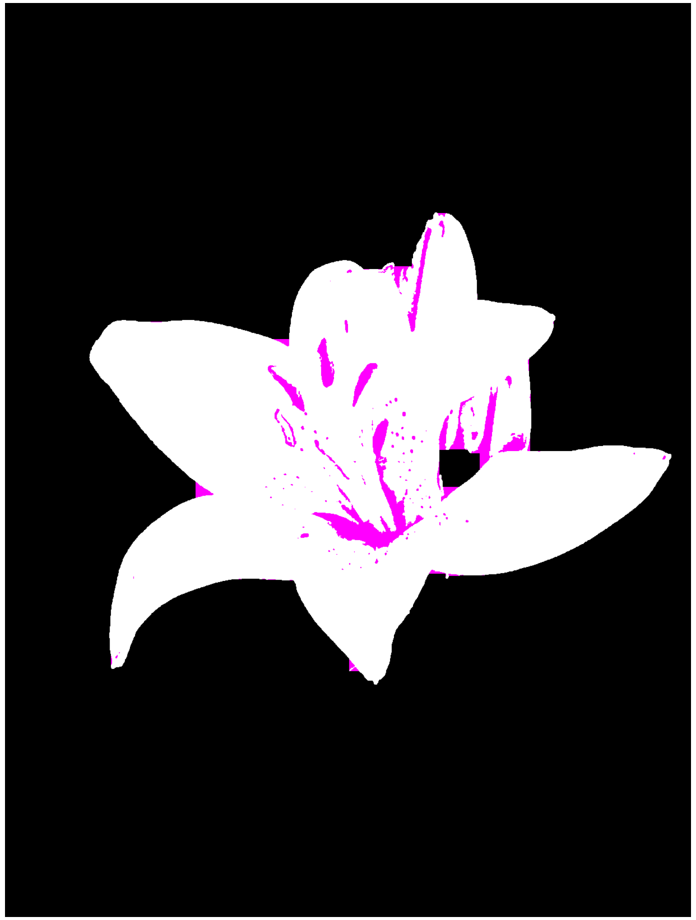
Erosion

- Choose **minimum** value in each neighborhood
- Binary images
  - **Shrinks** object regions
- Neighborhoods can have different shapes (e.g. diamond, disk, rectangle)
- Matlab: *imerode()*



Close

- Combinations of morphological filters can be used to clean up noisy segmentations
- Fill holes in objects:
    1. Apply erosion
    2. Apply dilation
- Size of neighborhood determines size of holes that can be filled



Open

- Opposite order of fill holes operations
- Fill holes in objects:
  1. Apply dilation
  2. Apply erosion
- Size of neighborhood determines size of noise regions that can be remove

Image "Open" for the Brain Image

- An opening can eliminate undesirable connections due to structural noise in segmented images:

% make a binary mask from the **thresholded** brain image

>> mask = brain4;

>> mask(find(mask))=1;

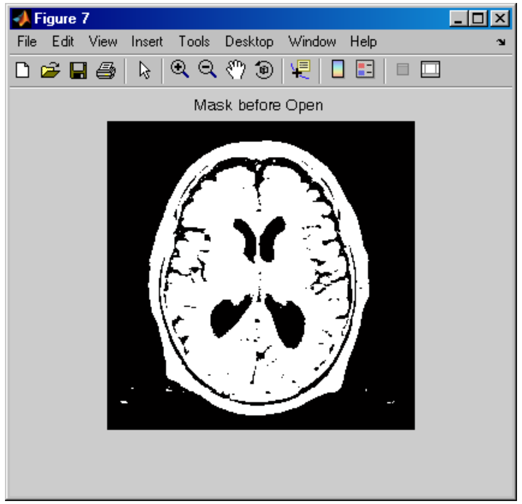
% apply an **open operation** using a 3 x 3 kernel, se.

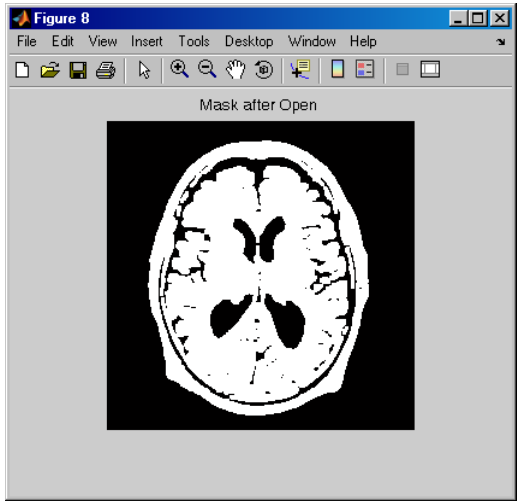
>> se = ones(3);

>> brain8 = imopen(mask, se);

% segment by applying (**multiplying by**) mask after open operation

>> brain9 = brain8.*brain4;





Global Threshold + Open

- Segment the skin, fat and skull region from the brain

```
brainmask = brain8bit;
brainmask(find(brain8bit>72))=1;
brainmask(find(brain8bit<72))=0;
% convert to binary image
brainmask = logical(brainmask);

se_3by3=strel('square',3);
brainseg = imopen(brainmask,se_3by3);
```

Region Growing + Masking

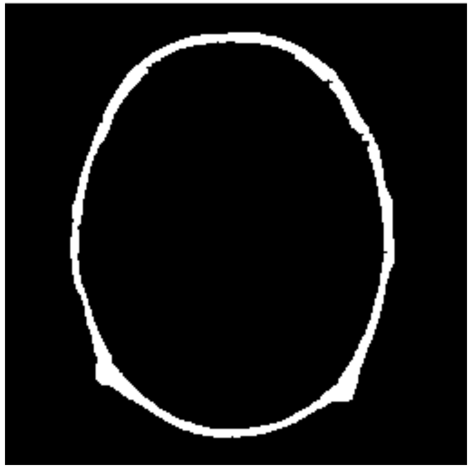
% manually select objects using mouse in a binary image using **region growing**

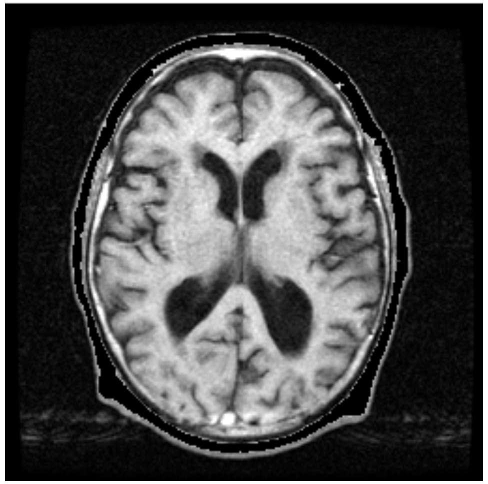
brainskull = **bwselect(brainseg)**

% Apply mask to original image:

brain = double(brain8bit).*(1-double(brainskull));

Fat = double(brain8bit).*(double(brainskull));





Summary

- Basic operations for segmentation:
  - Thresholding (global and adaptive)
  - Morphological operators (open and close)
- Region growing, filter design and connectivity concepts can be used in concert to achieve simple segmentation tasks
  - Matlab Implementation Examples