



# Automatic Thresholding

---

# Basic Idea of Image Segmentation

- Segmentation is often considered to be the first step in image analysis.
  - The purpose is to subdivide an image into meaningful non-overlapping regions, which would be used for further analysis.
  - It is hoped that the regions obtained correspond to the physical parts or objects of a scene (3-D) represented by the image (2-D).
- In general, autonomous segmentation is one of the most difficult tasks in digital image processing.

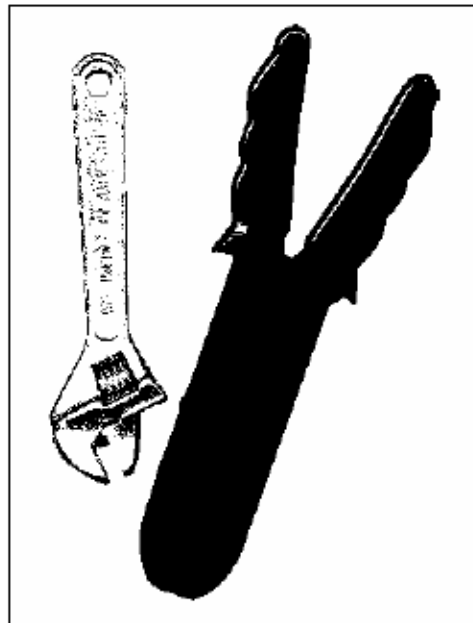
Example



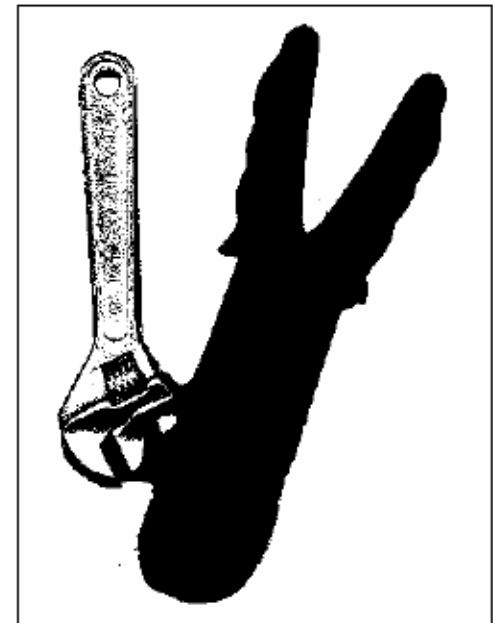
# Basic Idea of Image Segmentation

- All the image segmentation methods assume that:
  1. the intensity values are different in different regions, and,
  2. within each region, which represents the corresponding object in a scene, the intensity values are similar.

Original



Threshold = 50



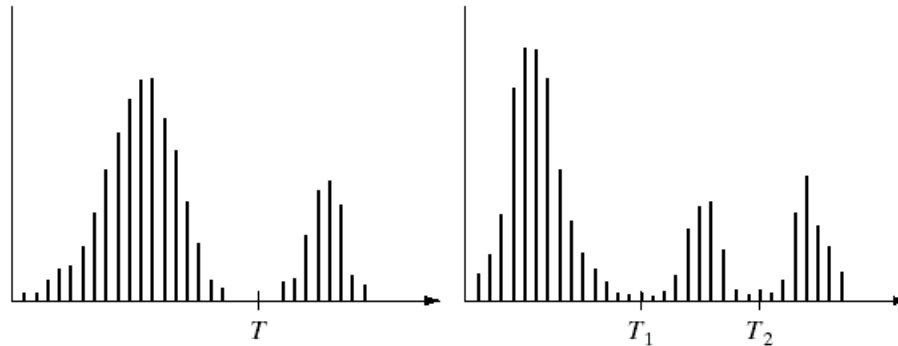
Threshold = 75

# Thresholding

- Thresholding is used to extract an object from its background by assigning an intensity value  $T$  (threshold) for **each pixel** such that each pixel is either classified as an object point or a background point
- In general  $T = T[x, y, p(x, y), f(x, y)]$
- If  $T$  is a function of  $f(x, y)$  only
  - Global thresholding
- If  $T$  is a function of both  $f(x, y)$  and local properties  $p(x, y)$ 
  - Local thresholding
- If  $T$  depends on the coordinates  $(x, y)$ 
  - Dynamic/adaptive thresholding

# Global Thresholding

- When the modes of histogram can be clearly distinguished



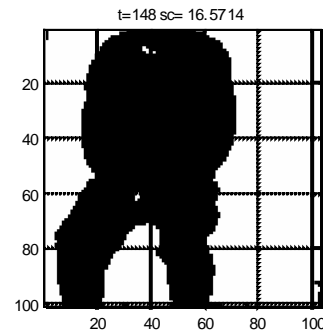
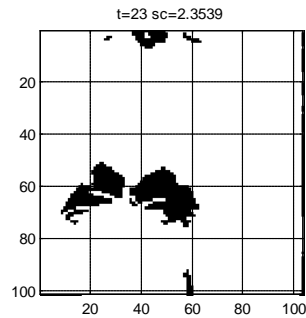
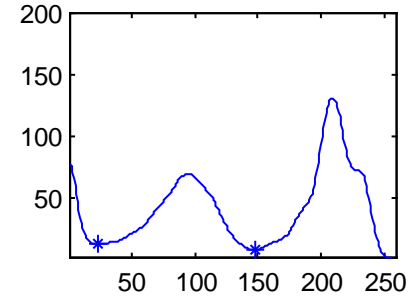
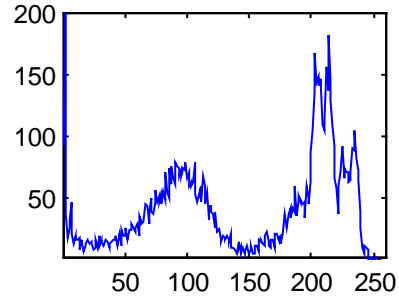
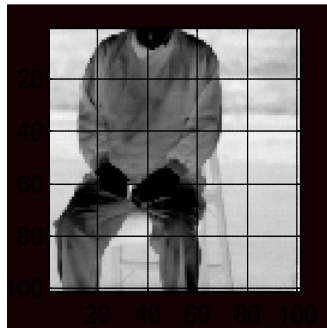
a b

**FIGURE 10.26** (a) Gray-level histograms that can be partitioned by (a) a single threshold, and (b) multiple thresholds.

- Each mode represents either the background or an object

# Global Thresholding

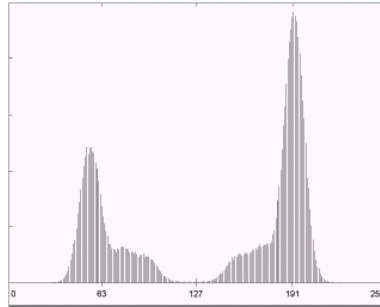
- T is usually located at the valley/ one of the valleys



# Global Thresholding

An automated procedure for bimodal histograms

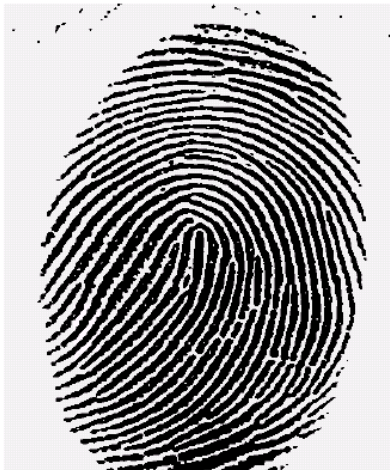
1. Select an initial estimate of  $T$
  2. Compute the means of the two regions determined by  $T$
  3. Set the new  $T$  as the average of the two means
  4. Repeat steps 2,3 until the difference in  $T$  in successive iterations is smaller than a predefined parameter
- Small Project: Implement this algorithm and experiment on images with a) two major modes b) three major modes and observe the results using different initial estimates of  $T$ . Discuss your findings. (10 points, due Sept. 27)



a b  
c

**FIGURE 10.29**

(a) Original image. (b) Image histogram. (c) Result of segmentation with the threshold estimated by iteration. (Original courtesy of the National Institute of Standards and Technology.)



- Initial threshold is the average of the gray values
- Clear bimodal histogram
- Download the image (a) and choose initial thresholds from 30 ~ 220
- Plot the values of final thresholds vs. initial thresholds
- Repeat the experiment using an image with three major modes

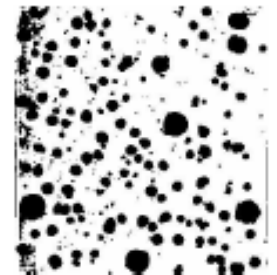
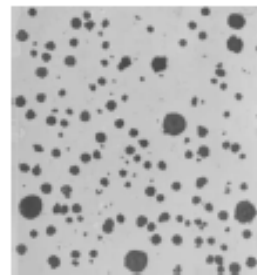


# Choosing a threshold: Review

- Image thresholding classifies pixels into two categories:
- Those to which some property measured from the image falls below a threshold, and those at which the property equals or exceeds a threshold.
  - Thresholding creates a binary image → binarization  
e.g. perform cell counts in histological images

**Choosing a threshold is something of a “black art”:**

```
n=imread('nodules1.jpg');  
figure(1); imshow(n);  
n1=im2bw(n,0.35);  
n2=im2bw(n,0.75);  
figure(2), imshow(n1);  
figure(3), imshow(n2);
```



n: Original image    n1: Threshold too low    n2: Threshold too high



# Fixed Thresholding

---

- In **fixed** (or global) thresholding, the threshold value is held constant throughout the image:
  - Determine a single threshold value by treating each pixel independently of its neighborhood.
- Fixed thresholding is of the form: 
$$g(x,y) = \begin{cases} 0 & f(x,y) < T \\ 1 & f(x,y) \geq T \end{cases}$$
- Assumes high-intensity pixels are of interest, and low-intensity pixels are not.



# Threshold Values & Histograms

---

- Thresholding usually involves *analyzing the histogram*
  - Different features give rise to distinct features in a histogram
  - In general the histogram peaks corresponding to two features will overlap. The degree of overlap depends on peak separation and peak width.
- An example of a threshold value is the *mean intensity* value



# Automatic Thresholding Algorithm: Iterative threshold selection

---

- 1 Select an initial estimate of the threshold  $T$ . A good initial value is the average intensity of the image.
3. Calculate the mean grey values  $\mu_1$  and  $\mu_2$  of the partitions,  $R1, R2$ .
2. Partition the image into two groups,  $R1, R2$ , using the threshold  $T$ .
4. Select a new threshold:

$$T = \frac{1}{2}(\mu_1 + \mu_2)$$

5. Repeat steps 2-4 until the mean values  $\mu_1$  and  $\mu_2$  in successive iterations do not change.

# Matlab Implementation

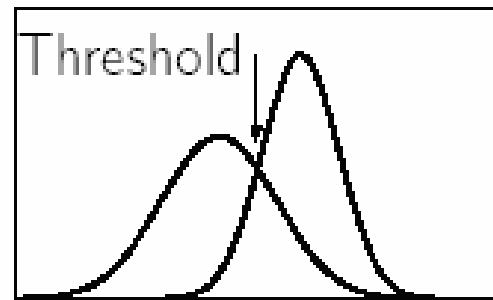
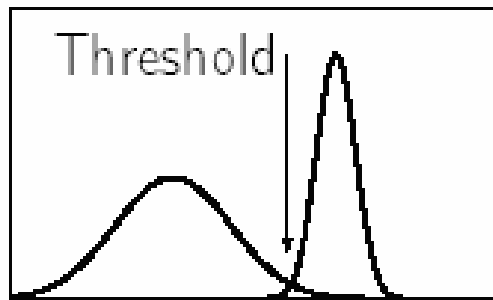
Matlab function *opthr*

```
addpath C:\Zhang\scm3511\week3\matlab
im = imread('boy.jpg');
I = rgb2gray(im);
I=double(I);
T=opthr(I);
bim=(I>T);
subplot(1,2,1), imshow(I, gray(256));
subplot(1,2,2), imshow(bim)
```



# Optimal Thresholding

- Histogram shape can be useful in locating the threshold.
  - However it is not reliable for threshold selection when peaks are not clearly resolved.
  - A “flat” object with no discernable surface texture, and no colour variation will give rise to a relatively narrow histogram peak.
- Choosing a threshold in the valley between two overlapping peaks, and inevitably some pixels will be incorrectly classified by the thresholding.



The determination of peaks and valleys is a non-trivial problem

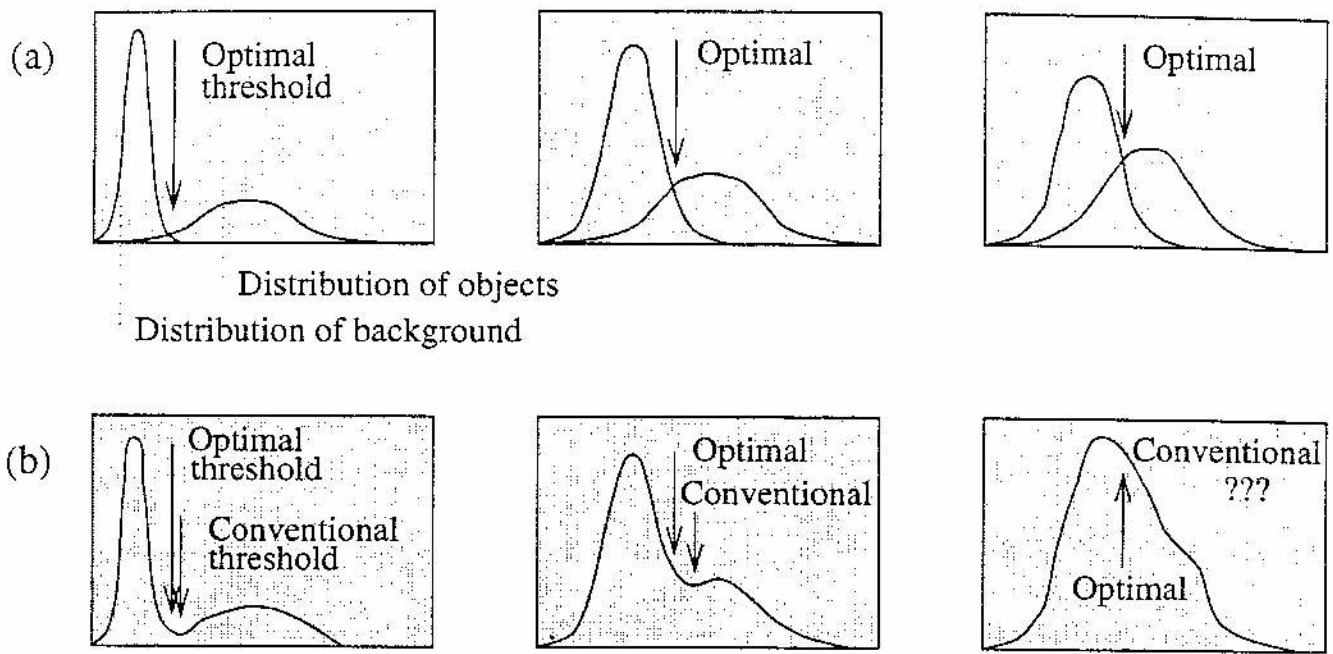


Figure 5.4: *Gray-level histograms approximated by two normal distributions—the threshold is set to give minimum probability of segmentation error: (a) probability distributions of background and objects; (b) corresponding histograms and optimal threshold.*

The following algorithm represents a simpler version that shows a rationale for this approach [Ridler and Calvard 78] and works well even if the image histogram is not bi-modal. This method assumes that regions of two main gray-levels are present in the image, thresholding of printed text being an example. The algorithm is iterative, four to ten iterations usually being sufficient.

**Algorithm 5.2: Iterative (optimal) threshold selection**

1. Assuming no knowledge about the exact location of objects, consider as a first approximation that the four corners of the image contain background pixels only and the remainder contains object pixels.
2. At step  $t$ , compute  $\mu_B^t$  and  $\mu_O^t$  as the mean background and object gray-level, respectively, where segmentation into background and objects at step  $t$  is defined by the

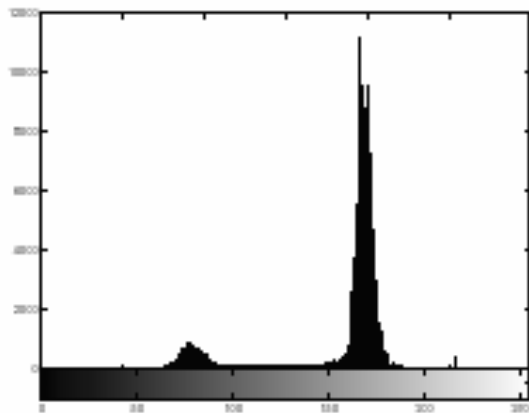
$$\mu_B^t = \frac{\sum_{(i,j) \in \text{background}} f(i,j)}{\#\text{background\_pixels}} \quad \mu_O^t = \frac{\sum_{(i,j) \in \text{objects}} f(i,j)}{\#\text{object\_pixels}} \quad (5.8)$$

3. Set

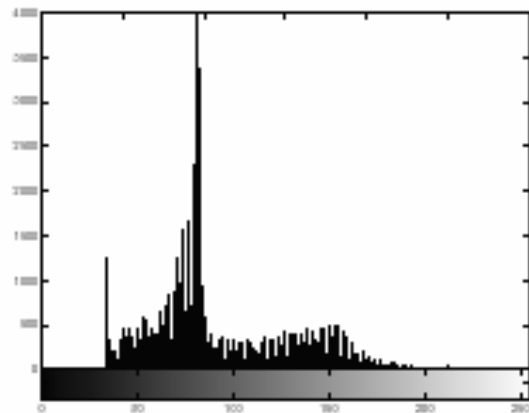
$$T^{(t+1)} = \frac{\mu_B^t + \mu_O^t}{2} \quad (5.9)$$

$T^{(t+1)}$  now provides an updated background—object distinction.

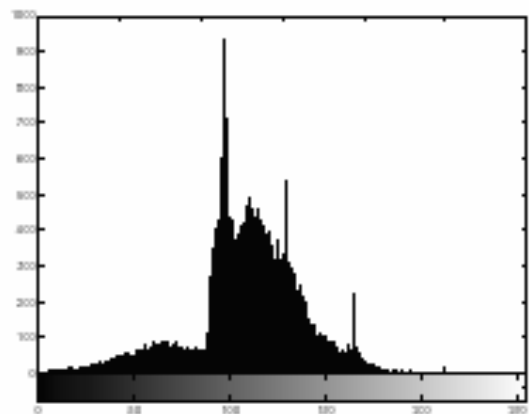
4. If  $T^{(t+1)} = T^{(t)}$ , halt; otherwise return to step 2.



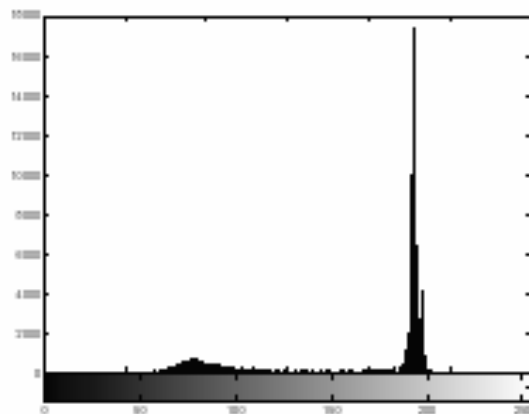
Nodules image: nodules.tif



Rice image: rice.tif



Bacteria image: bacteria.tif



Coins image: coins.tif





# Optimal Thresholding

---

- In **optimal thresholding**, a **criterion function** is devised that yields some measure of separation between regions.
  - A criterion function is calculated for each intensity and that which maximizes this function is chosen as the threshold.
- **Otsu's thresholding** chooses the threshold to minimize the intraclass variance of the thresholded black and white pixels.
  - Formulated as *discriminant analysis*: a particular criterion function is used as a measure of statistical separation.

# Otsu's Thresholding Method <sup>(1979)</sup>

- Based on a very simple idea: Find the threshold that *minimizes the weighted within-class variance*.
- This turns out to be the same as *maximizing the between-class variance*.
- Operates directly on the gray level histogram [*e.g.* 256 numbers,  $P(i)$ ], so it's fast (once the histogram is computed).
- I've used it with considerable success in “murky” situations.

# Otsu: Assumptions

- Histogram (and the image) are *bimodal*.
- No use of *spatial coherence*, nor any other notion of object structure.
- Assumes stationary statistics, but can be modified to be locally adaptive. (exercises)
- Assumes uniform illumination (implicitly), so the bimodal brightness behavior arises from object appearance differences only.

The *weighted within-class variance* is:

$$\sigma_w^2(t) = q_1(t)\sigma_1^2(t) + q_2(t)\sigma_2^2(t)$$

Where the class probabilities are estimated as:

$$q_1(t) = \sum_{i=1}^t P(i) \quad q_2(t) = \sum_{i=t+1}^I P(i)$$

And the class means are given by:

$$\mu_1(t) = \sum_{i=1}^t \frac{iP(i)}{q_1(t)} \quad \mu_2(t) = \sum_{i=t+1}^I \frac{iP(i)}{q_2(t)}$$

Finally, the individual class variances are:

$$\sigma_1^2(t) = \sum_{i=1}^t [i - \mu_1(t)]^2 \frac{P(i)}{q_1(t)}$$

$$\sigma_2^2(t) = \sum_{i=t+1}^I [i - \mu_2(t)]^2 \frac{P(i)}{q_2(t)}$$

Now, we could actually stop here. All we need to do is just run through the full range of  $t$  values [1,256] and pick the value that minimizes  $\sigma_w^2(t)$ .

But the relationship between the within-class and between-class variances can be exploited to generate a recursion relation that permits a much faster calculation.

# Between/Within/Total Variance

- The book gives the details, but the basic idea is that the total variance does not depend on threshold (obviously).
- For any given threshold, the total variance is the sum of the within-class variances (weighted) and the *between class variance*, which is the sum of weighted squared distances between the class means and the grand mean.

After some algebra, we can express the total variance as...

$$\sigma^2 = \underbrace{\sigma_w^2(t)}_{\substack{\text{Within-class,} \\ \text{from before}}} + \underbrace{q_1(t)[1 - q_1(t)][\mu_1(t) - \mu_2(t)]^2}_{\text{Between-class, } \sigma_B^2(t)}$$

Since the total is constant and independent of  $t$ , the effect of changing the threshold is merely to move the contributions of the two terms back and forth.

*So, minimizing the within-class variance is the same as maximizing the between-class variance.*

The nice thing about this is that we can compute the quantities in  $\sigma_B^2(t)$  *recursively* as we run through the range of  $t$  values.

# Finally...

*Initialization...*  $q_1(1) = P(1); \mu_1(0) = 0$

*Recursion...*

$$q_1(t+1) = q_1(t) + P(t+1)$$

$$\mu_1(t+1) = \frac{q_1(t)\mu_1(t) + (t+1)P(t+1)}{q_1(t+1)}$$

$$\mu_2(t+1) = \frac{\mu - q_1(t+1)\mu_1(t+1)}{1 - q_1(t+1)}$$



# Matlab function for Otsu's method

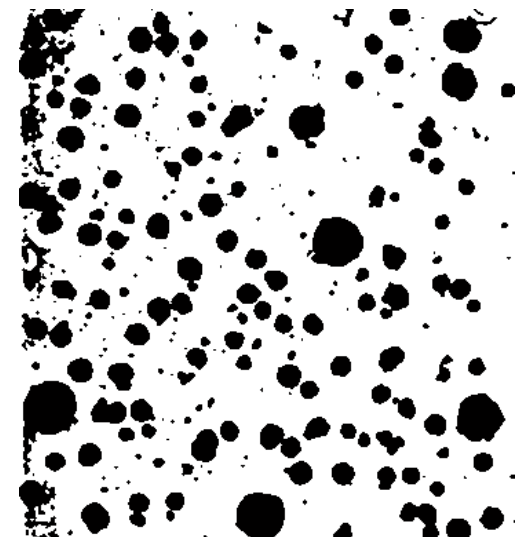
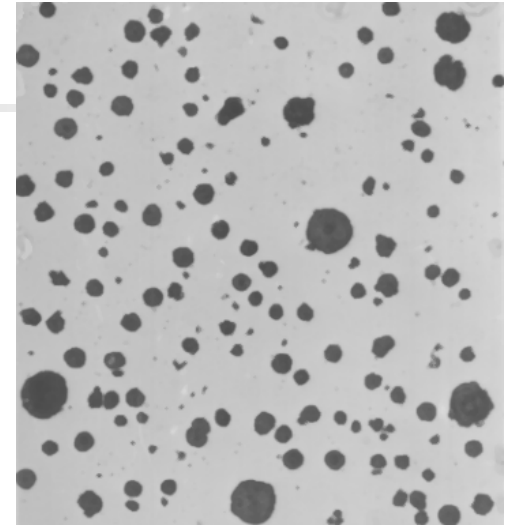
function level = graythresh(I)  
GRAYTHRESH Compute global threshold using Otsu's method. Level is a normalized intensity value that lies in the range [0, 1].

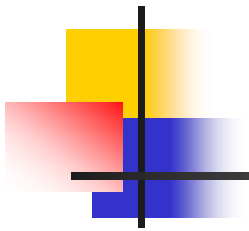
```
>>n=imread('nodules1.tif');
```

```
>> tn=graythresh(n)
```

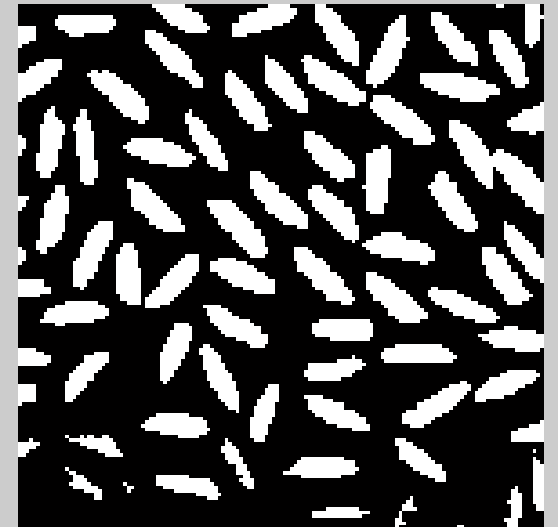
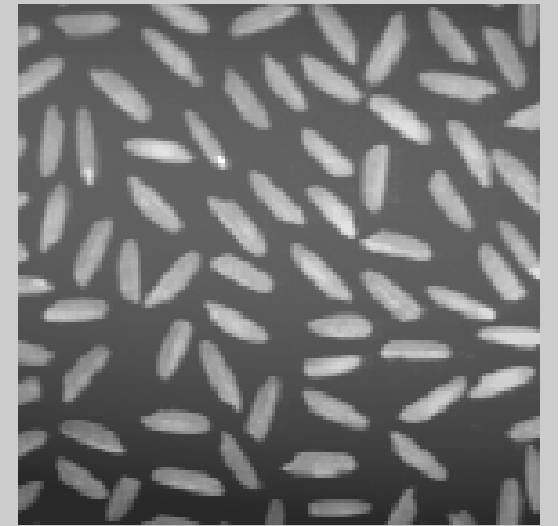
```
tn = 0.5804
```

```
>> imshow(im2bw(n,tn))
```





```
r=imread('rice.tif');  
graythresh;  
subplot(211);  
imshow;  
subplot(212);  
imshow(im2bw(r,tr))
```



- Otsu's method is implemented in MATLAB as "graythresh"
  - Use "help graythresh" to learn the syntax of this function

ponents or broken connection paths. There is no point past the level of detail required to identify those components.

Segmentation of nontrivial images is one of the most difficult tasks in image processing. Segmentation accuracy determines the effectiveness of computerized analysis procedures. For this reason, care should be taken to improve the probability of rugged segmentation, such as industrial inspection applications, at least some of the time. The experienced designer invariably pays considerable attention to such

ponents or broken connection paths. There is no point past the level of detail required to identify those components.

Segmentation of nontrivial images is one of the most difficult tasks in image processing. Segmentation accuracy determines the effectiveness of computerized analysis procedures. For this reason, care should be taken to improve the probability of rugged segmentation, such as industrial inspection applications, at least some of the time. The experienced designer invariably pays considerable attention to such

# graythresh

Global image threshold using Otsu's method

## Syntax

```
level = graythresh(I)  
[level EM] = graythresh(I)
```

## Description

`level = graythresh(I)` computes a global threshold (`level`) that can be used to convert an intensity image to a binary image with `im2bw`. `level` is a normalized intensity value that lies in the range `[0, 1]`.

The `graythresh` function uses Otsu's method, which chooses the threshold to minimize the intraclass variance of the black and white pixels.

Multidimensional arrays are converted automatically to 2-D arrays using `reshape`. The `graythresh` function ignores any nonzero imaginary part of `I`.

`[level EM] = graythresh(I)` returns the effectiveness metric, `EM`, as the second output argument. The effectiveness metric is a value in the range `[0 1]` that indicates the effectiveness of the thresholding of the input image. The lower bound is attainable only by images having a single gray level, and the upper bound is attainable only by two-valued images.

## Class Support

The input image `I` can be of class `uint8`, `uint16`, `int16`, `single`, or `double` and it must be nonsparse. The return value `level` is a double scalar. The effectiveness metric `EM` is a double scalar.

## Examples

```
I = imread('coins.png');  
level = graythresh(I);  
BW = im2bw(I,level);  
imshow(BW)
```

## Reference

[1] Otsu, N., "A Threshold Selection Method from Gray-Level Histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 9, No. 1, 1979, pp. 62-66.