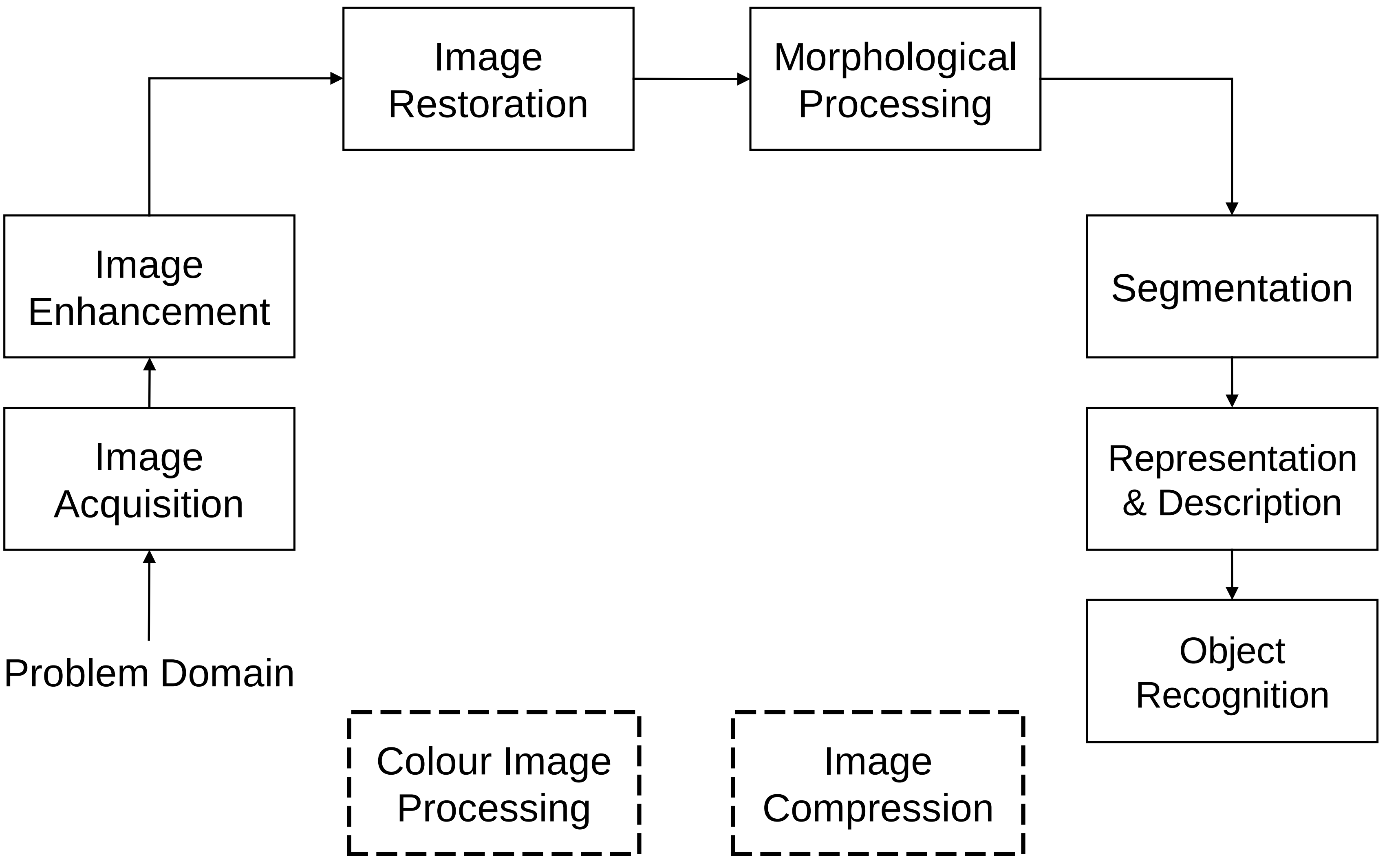
# Image Segmentation-I

## Pixel and Region based Approach

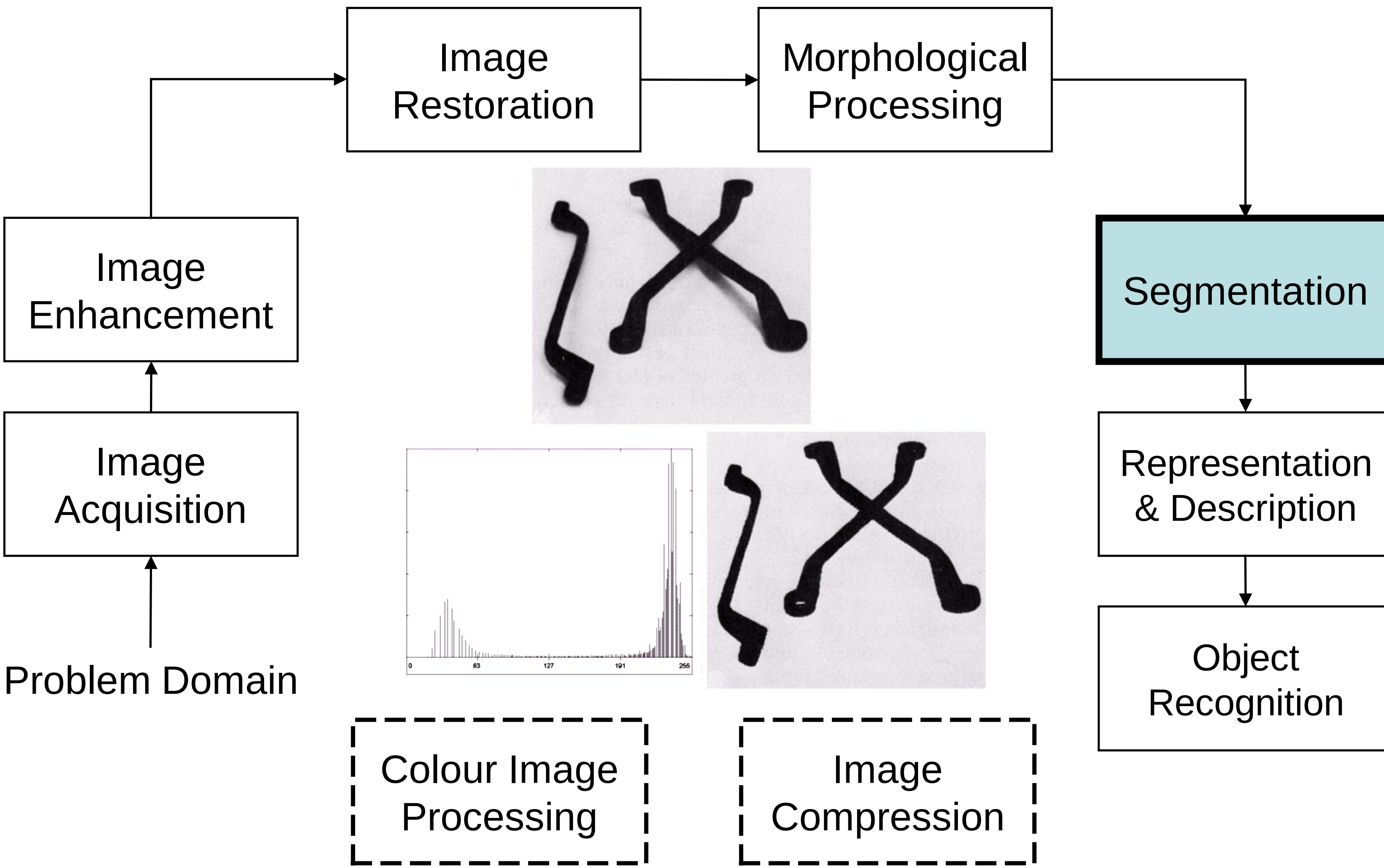
*Partially Adopted from* Brian Mac Namee

# Contents

# Phases of Digital Image Processing



# Phases of Digital Image Processing : Segmentation




Images taken from Gonzalez & Woods, Digital Image Processing (2002)

# Image Segmentation

We have been considered image processing techniques used to transform images for human interpretation.

We will begin looking at automated image analysis by examining the challenging issues of image segmentation:

- The segmentation problem
- Different segmentation approaches

# Image Segmentation

Image processing techniques - enhancement and restoration - take a digital image as input and gives out another image that is of improved quality.

On the other hand, in image analysis technique, the same input gives out somewhat detail description of the scene whose image is being considered.

# Image Segmentation

Most of the image analysis algorithms perform segmentation as a first step towards producing the description.

Here input and output are still images, but output is an abstract representation of the input.

Segmentation technique basically divides the spatial domain, on which the image is defined, into meaningful parts of region.

# Image Segmentation

The segmentation algorithms try to make systematic use of some physically measured image features, but its performance is measured based on the meaning associated with the extracted regions.

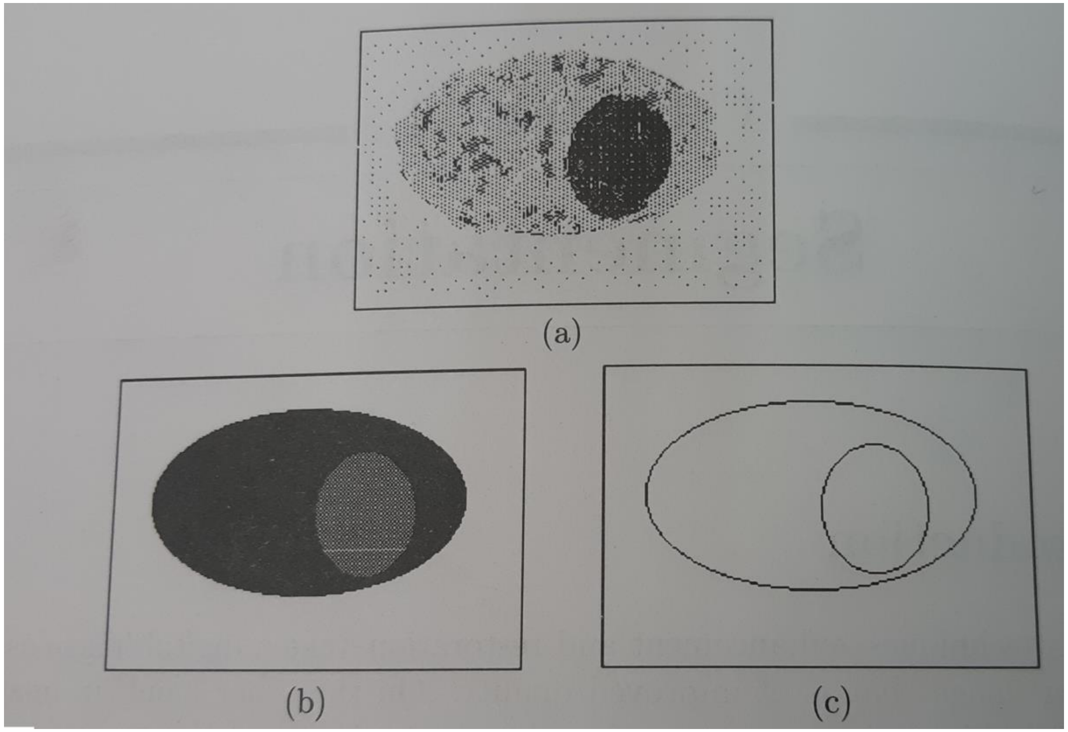
So, we may consider segmentation as a psycho-physical problem.

# Image Segmentation

The segmentation algorithms are basically based on one of the two approaches as:

- Satisfying homogeneity property in image feature(s) over a large region.
- Detecting abrupt changes in image feature(s) within a small neighbourhood.

# Image Segmentation

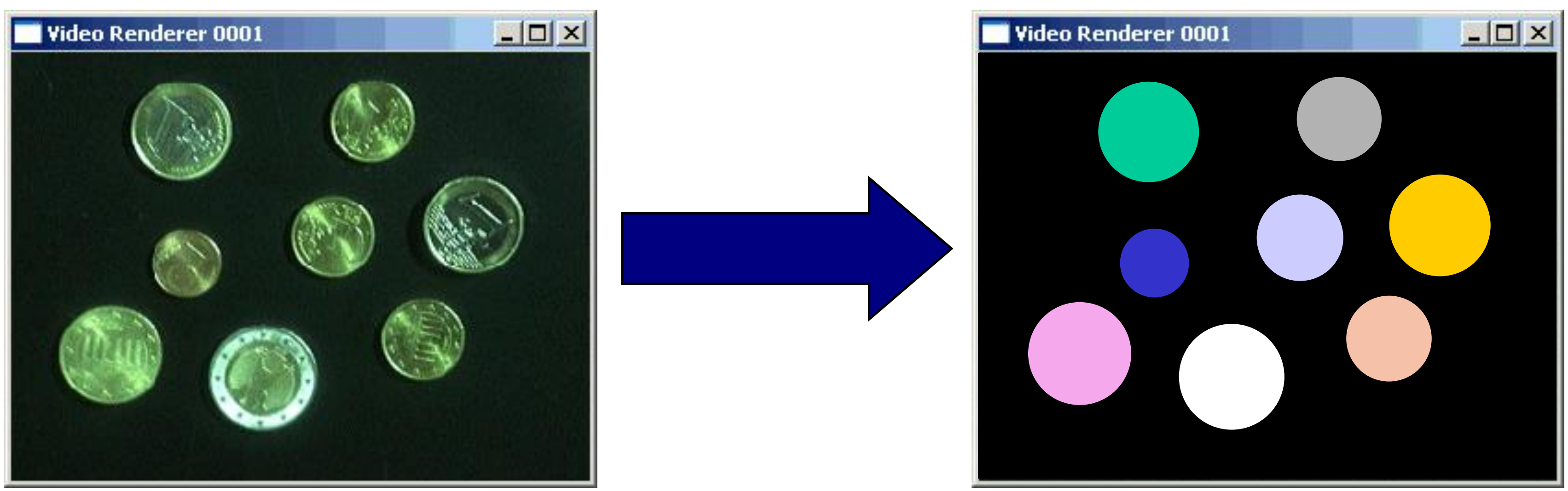



Output of various segmentation techniques: (a) input image, (b) expected result of the region-extraction techniques and (c) expected result of edge detection technique.

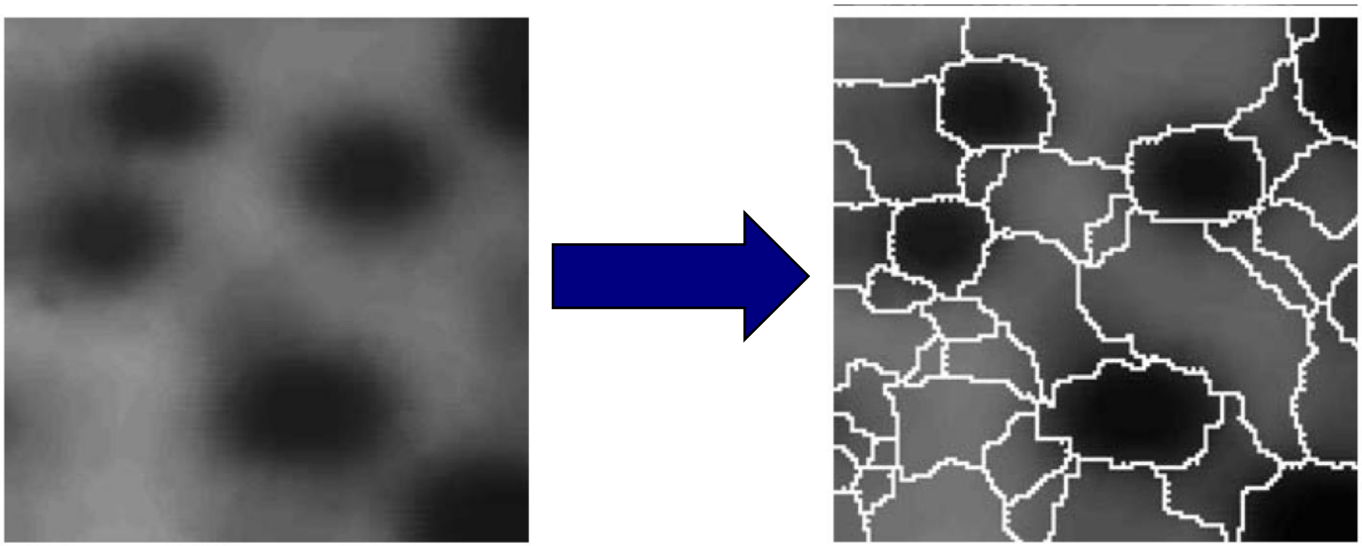
# The Segmentation Problem

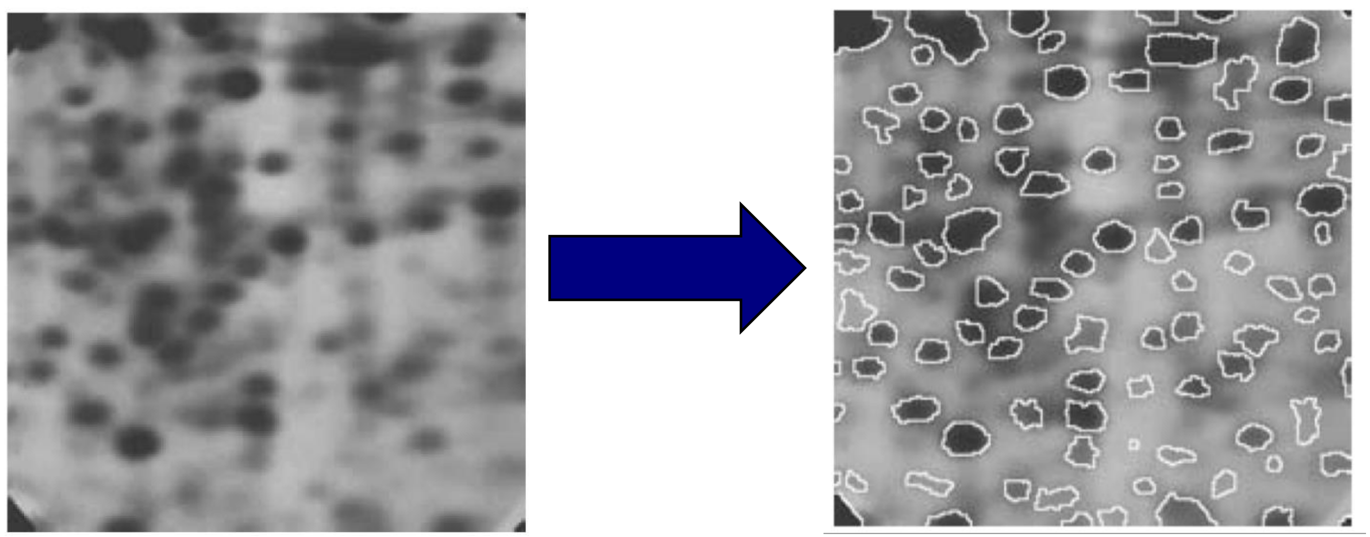
Segmentation attempts to partition the pixels of an image into groups that strongly correlate with the objects in an image.

Typically the first step in any automated Computer Vision Application.



# Segmentation Examples





Images taken from Gonzalez & Woods, Digital Image Processing (2002)

# Basic Formulation

Let R represent the entire image region. Segmentation partitions R into n subregions, $R_1$, $R_2$, ..., $R_n$, such that:

a) $\bigcup_{i=1}^{n} R_i = R$

b) $R_i$ is a connected region, $i = 1, 2, \ldots, n$.

c) $R_i \cap R_j = \phi$ for all $i$ and $j$, $i \neq j$

d) $P(R_i) = TRUE$ for $i = 1, 2, \ldots, n$.

e) $P(R_i \cup R_j) = FALSE$ for $i \neq j$.

a) Every pixel must be in a region.

b) Points in a region must be connected.

c) Regions must be disjoint.

d) All pixels in a region satisfy specific properties.

e) Different regions have different properties.

# Pixel-based Segmentation

# Thresholding

Thresholding is usually the first step in any segmentation approach.

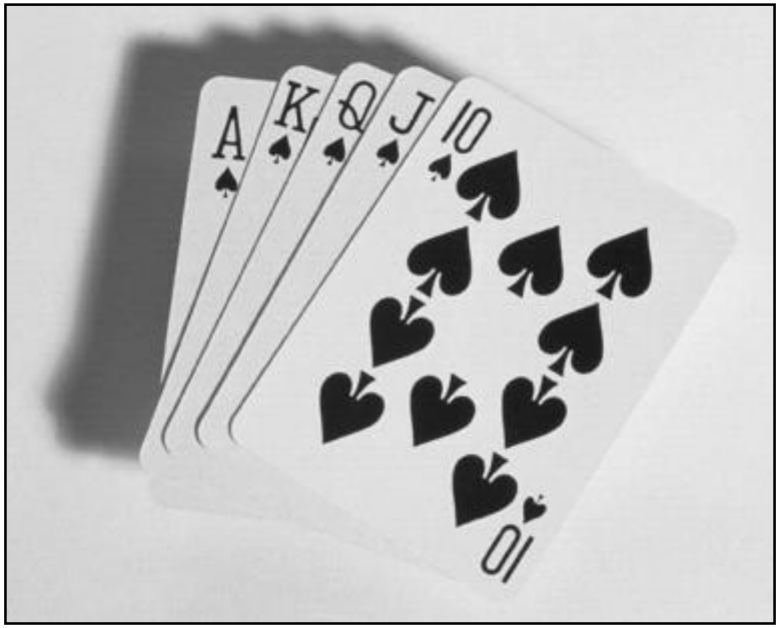
We have talked about simple single value thresholding already.

Single value thresholding can be given mathematically as follows:

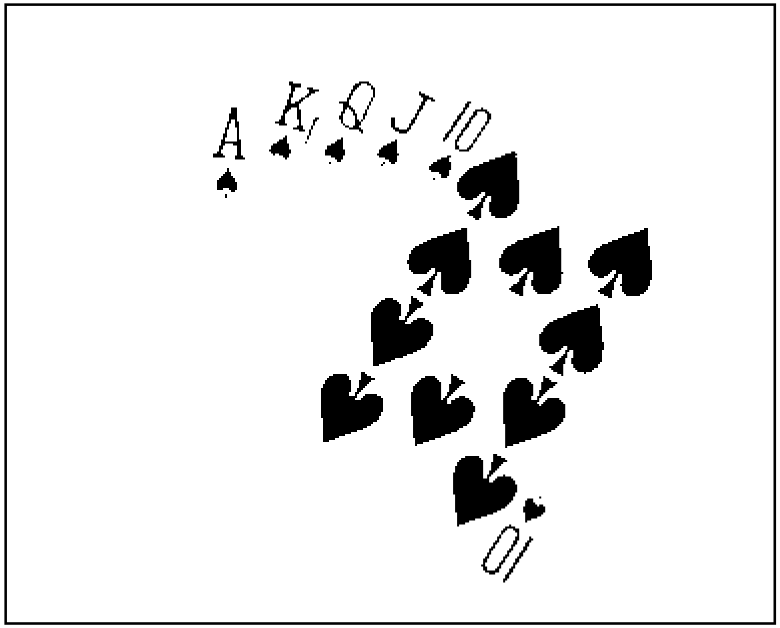
$$g(x,y)=\begin{cases} 1 & if\ \ f(x,y)>T \\ 0 & if\ \ f(x,y)\leq T \end{cases}$$

# Thresholding Example

Imagine a poker playing robot that needs to visually interpret the cards in its hand.



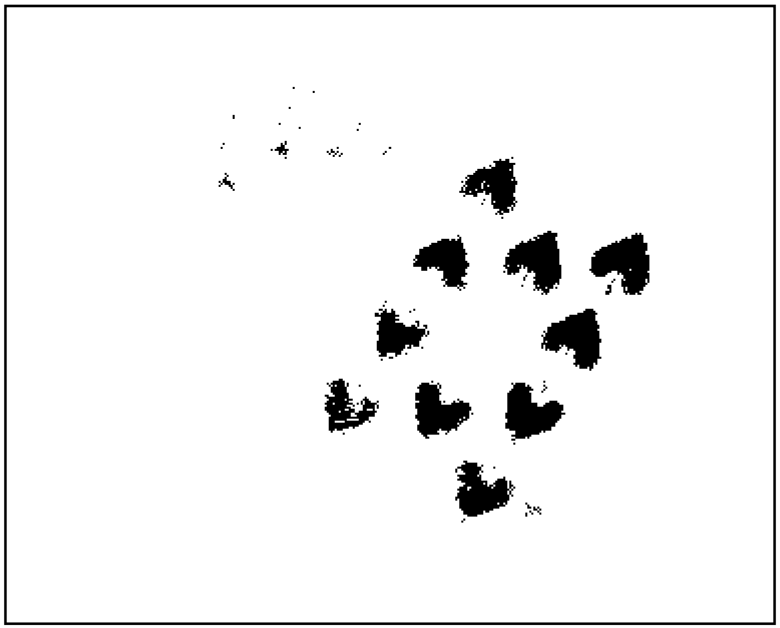
Original Image



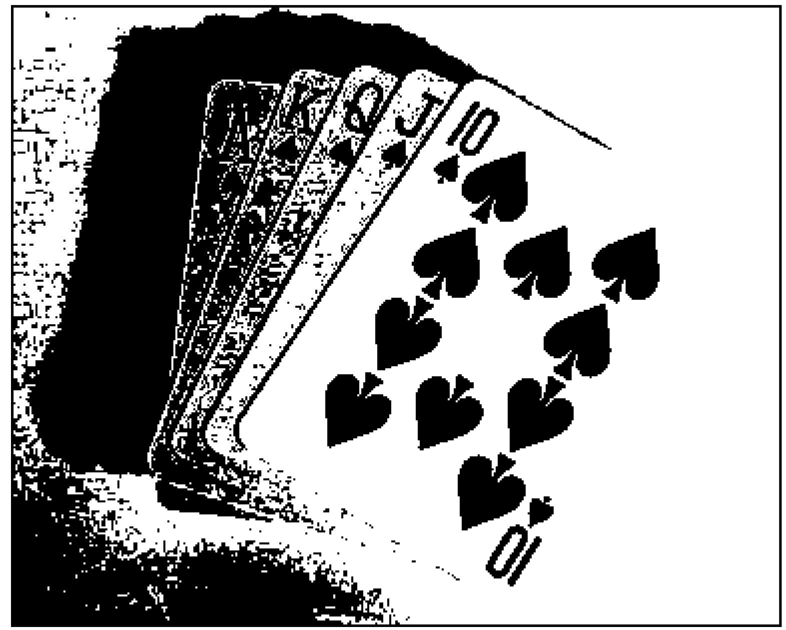
Thresholded Image

# Drawbacks of Thresholding

If you get the threshold wrong the results can be disastrous.



Threshold Too Low



Threshold Too High

# Basic Global Thresholding

Based on the histogram of an image partition the image histogram using a single global threshold.

The success of this technique very strongly depends on how well the histogram can be partitioned.

# Basic Global Thresholding Algorithm

The basic global threshold, T, is calculated as follows:

1. Select an initial estimate for T (typically the average grey level in the image).
2. Segment the image using T to produce two groups of pixels: $G_1$ consisting of pixels with grey levels >T and $G_2$ consisting pixels with grey levels $\leq$ T.
3. Compute the average grey levels of pixels in $G_1$ to give $\mu_1$ and $G_2$ to give $\mu_2$.

# Basic Global Thresholding Algorithm

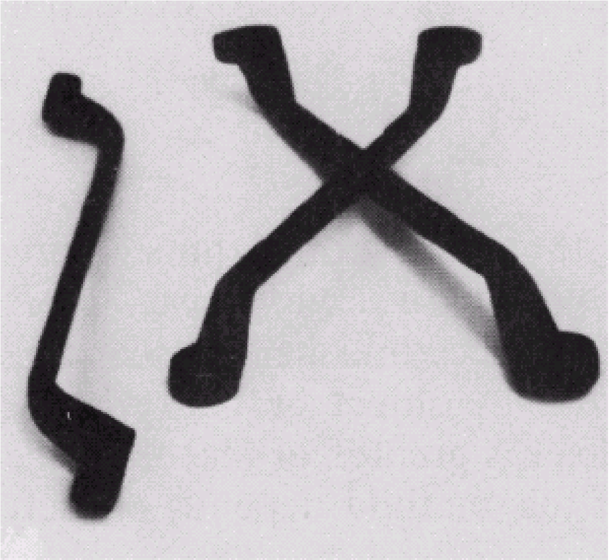
4. Compute a new threshold value:

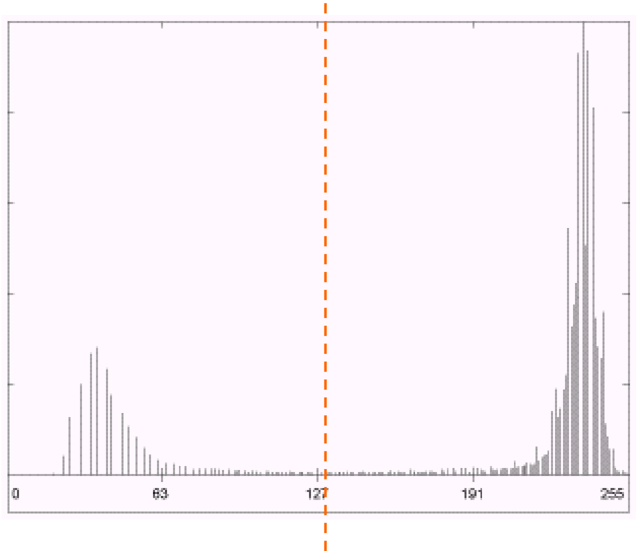
$$T = \frac{\mu_1 + \mu_2}{2}$$

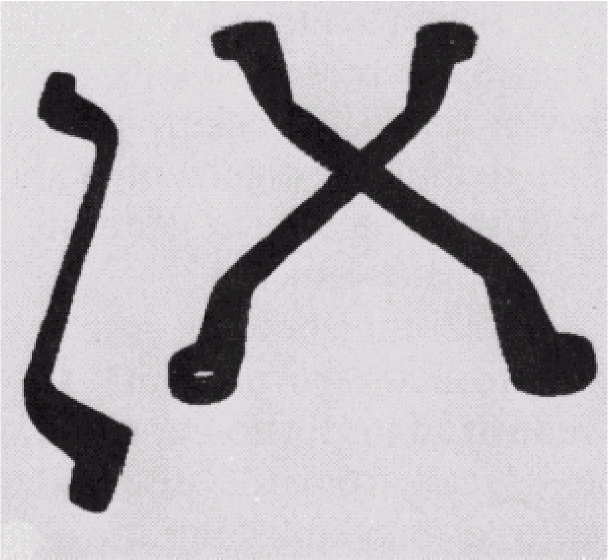
5. Repeat steps 2 – 4 until the difference in T in successive iterations is less than a predefined limit.

This algorithm works very well for finding thresholds when the histogram is suitable.

# Thresholding Example 1



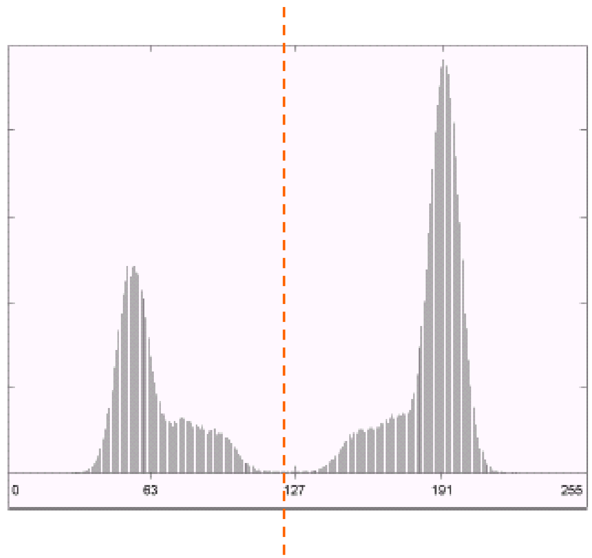


Images taken from Gonzalez & Woods, Digital Image Processing (2002)

# Thresholding Example 2






# Threshold Selection Methods

# Mode Method

It is one of the most popular method for threshold selection.

It utilizes the information contained in the histogram of the image.

This method selects the threshold corresponding to the bottom valley between the two peaks of the histogram.

# Mode Method

This method needs the histogram to be bimodal.

Since the image contains two distinct types of regions, $R_1$ and $R_2$, the gray level histogram $n_i$ contains two distinct peaks or modes at $z = k$ and $z = l$ corresponding to graylevels of pixels belonging to those regions.

# Mode Method

Ideally, the graylevels between $k$ and $l$ should not occur frequently in the image.

Therefore, we expect a deep valley at, say, $z = m$ between the peaks giving the histogram a bimodal shape.

# Mode Method

Suppose $s_1$ and $s_2$ are magnitude of slopes of lines joining bottom of valley and each of the peaks, i.e.

$$s_1 = \left|\frac{n_k - n_m}{k - m}\right|$$

$$s_2 = \left|\frac{n_l - n_m}{l - m}\right|$$

# Mode Method

Therefore, the histogram is said to

1. be bimodal if $s_1 > t_{sl}$ and $s_2 > t_{sl}$, or
2. have peak and shoulder if $s_1 > t_{sl}$ and $s_2 \leq t_{sl}$, or vice versa, or
3. be flat if $s_1 \leq t_{sl}$ and $s_2 \leq t_{sl}$.

The values of $s_1$ and $s_2$ depend on the number of pixels in the image.

# Mode Method

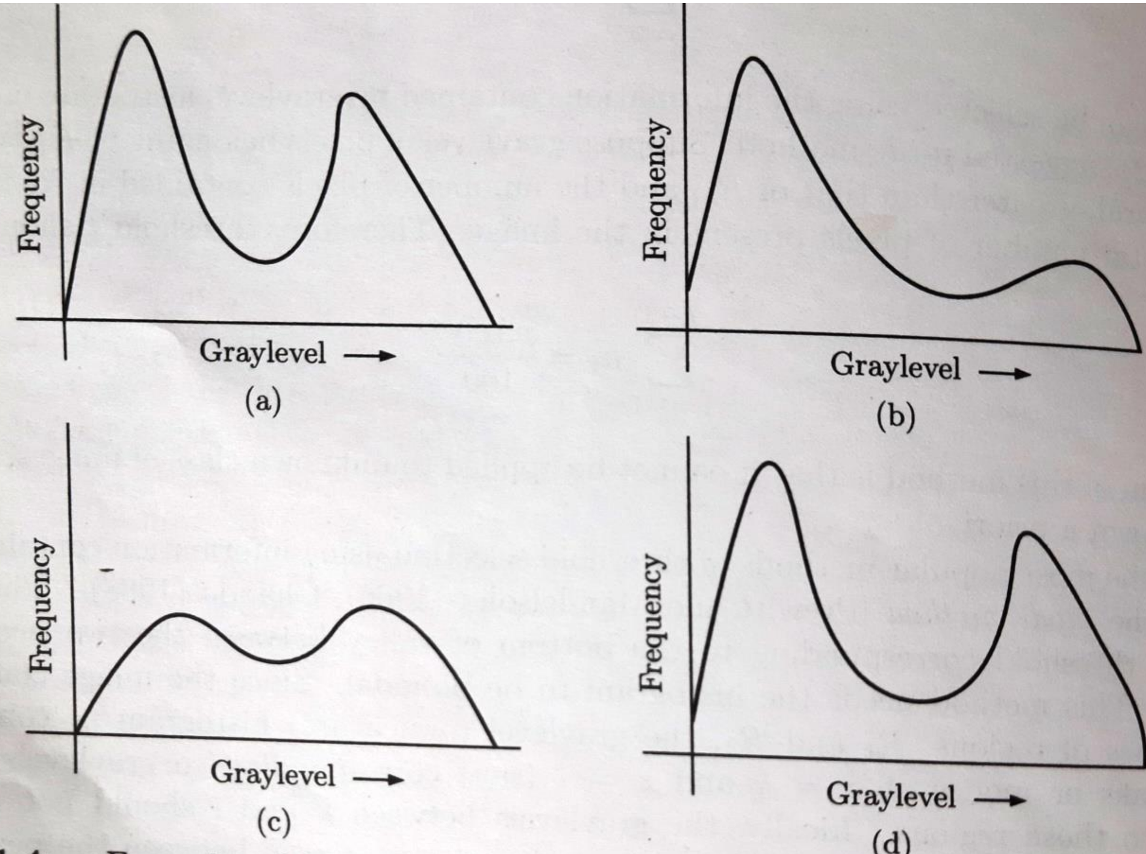



**Figure 11.4:** Examples of histogram: (a) strongly bimodal, (b) peak and shoulder, (c) flat and (d) weakly bimodal. Envelope of bar-heights are drawn for simplicity in representation.

# Mode Method

Before computing $s_1$ and $s_2$, $n_i$'s may be normalized with respect to $MN$. The strength of bimodality can be measured as

$$s_b = s_1 s_2$$

If the histogram is strongly bimodal, graylevel corresponding to bottom of this valley can be taken as near optimum threshold.

# Mode Method

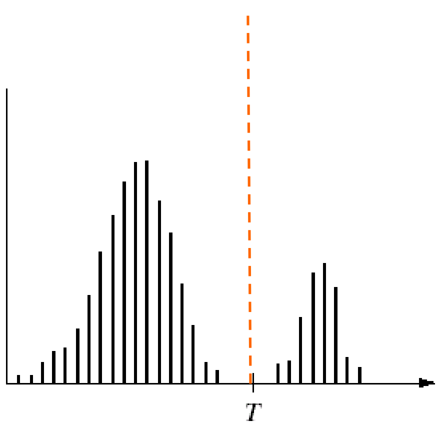
In real images, because of noise and blurring, variance of graylevels of pixels belonging to a region becomes large.

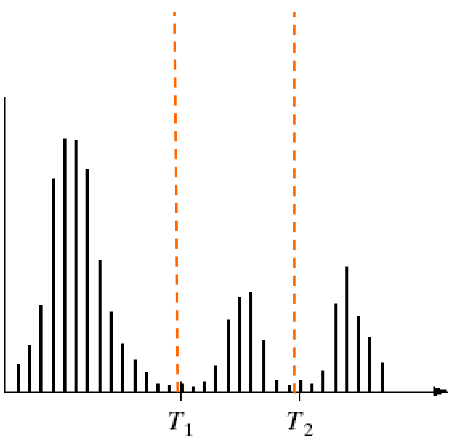
As a result, graylevel distributions corresponding to different region overlap and histogram may not have the bimodal shape.

# Problems With Single Value Thresholding

Single value thresholding only works for bimodal histograms.

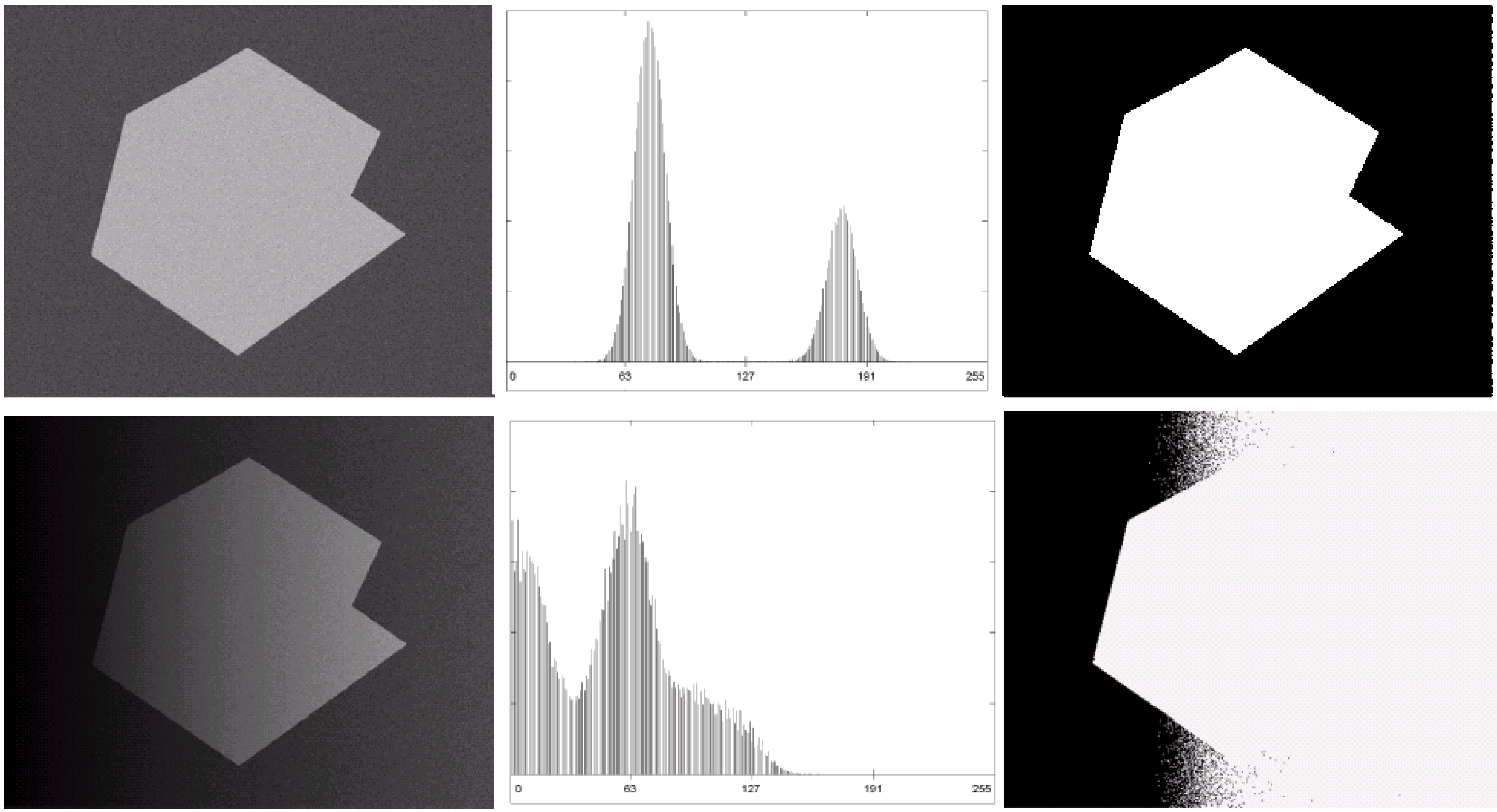
Images with other kinds of histograms need more than a single threshold.




**Multi-level thresholding**

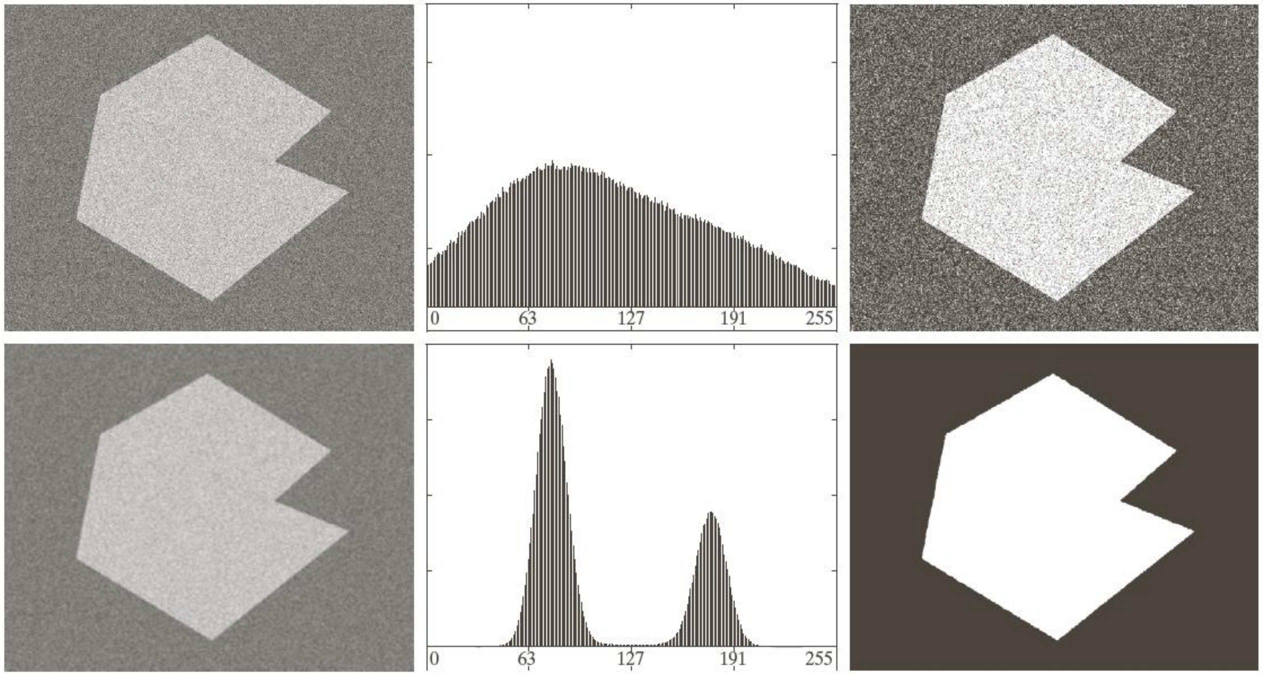
# Single Value Thresholding and Illumination




Uneven illumination can really upset a single valued thresholding scheme.

Images taken from Gonzalez & Woods, Digital Image Processing (2002)

# Noise Cleaning by Filtering




a b c
d e f

**FIGURE 10.40** (a) Noisy image from Fig. 10.36 and (b) its histogram. (c) Result obtained using Otsu's method. (d) Noisy image smoothed using a $5 \times 5$ averaging mask and (e) its histogram. (f) Result of thresholding using Otsu's method.

# Noise Cleaning by Filtering

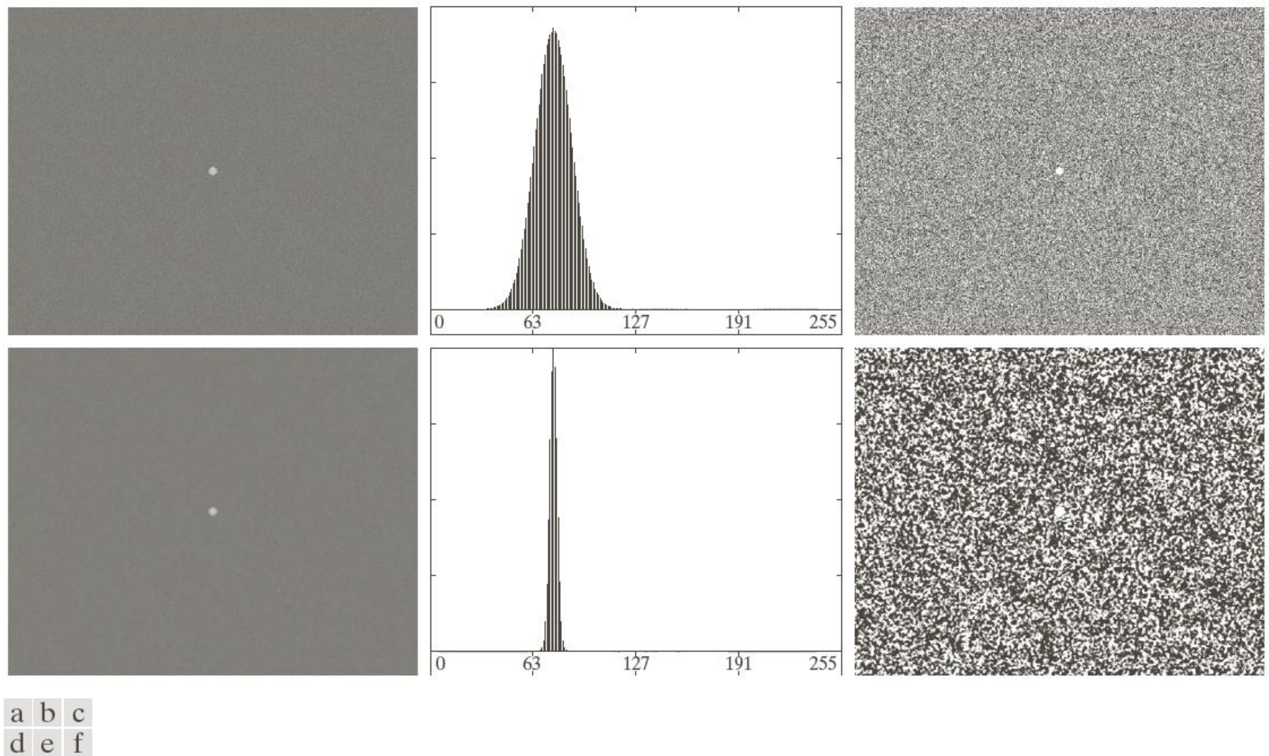


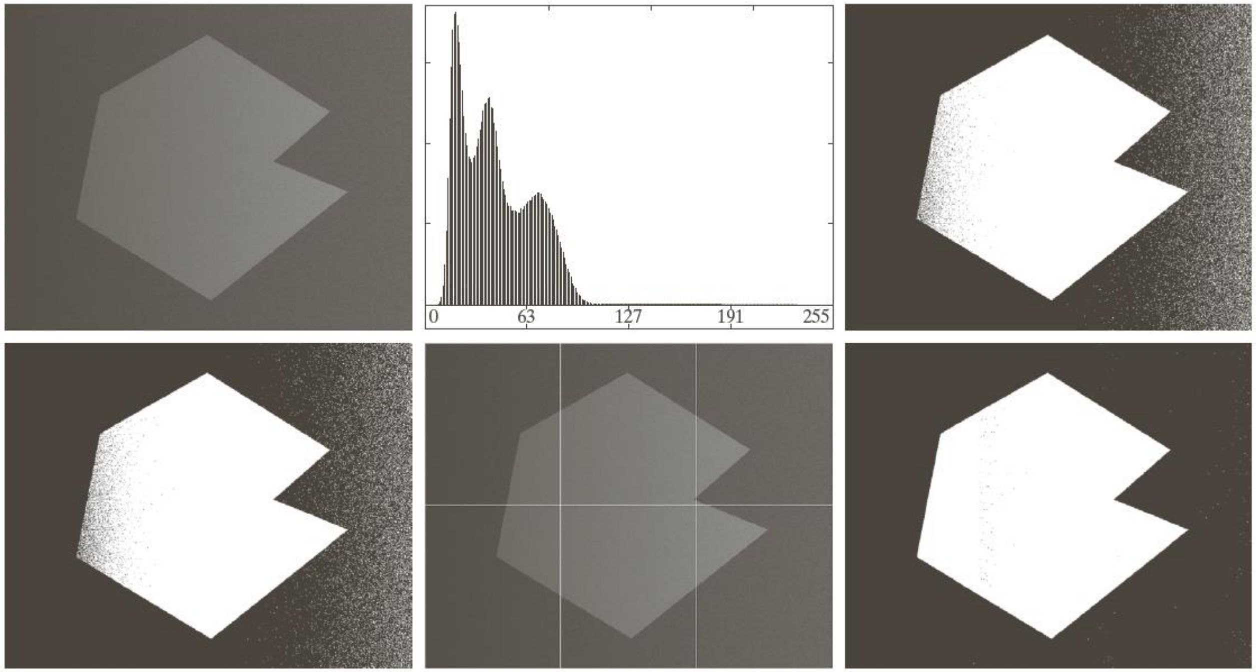

**FIGURE 10.41** (a) Noisy image and (b) its histogram. (c) Result obtained using Otsu's method. (d) Noisy image smoothed using a $5 \times 5$ averaging mask and (e) its histogram. (f) Result of thresholding using Otsu's method. Thresholding failed in both cases.

# Basic Adaptive Thresholding

An approach to handle situations in which single value thresholding will not work is to <u>divide an image into sub images and threshold these individually.</u>

Since the threshold for each pixel depends on its location within an image this technique is said to be adaptive.

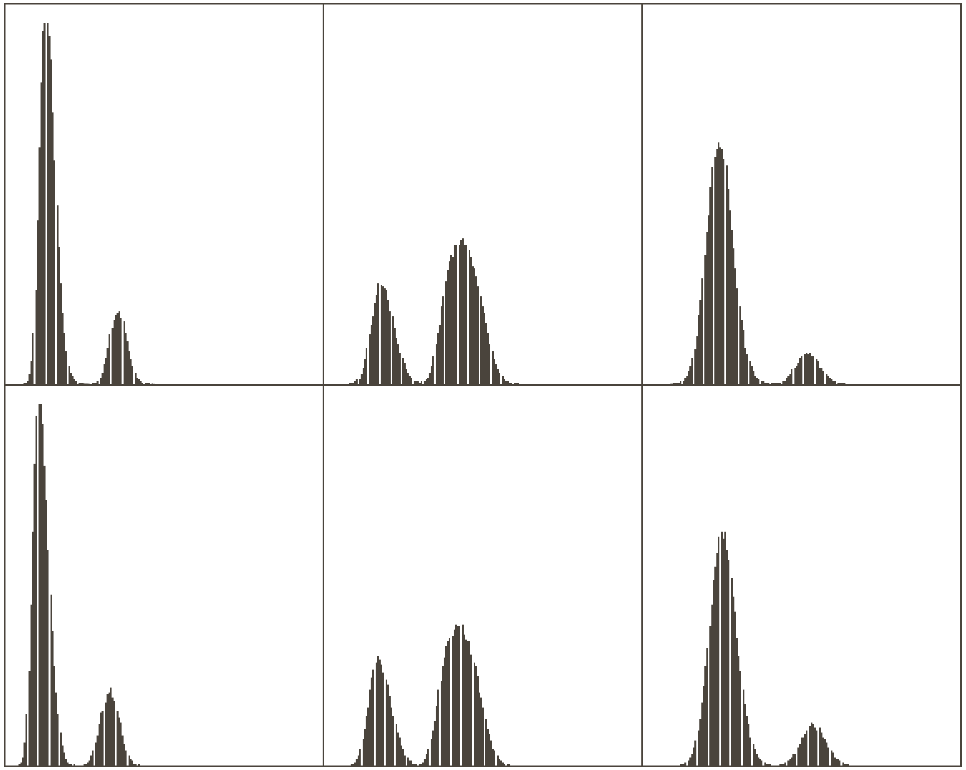
# Basic Adaptive Thresholding




a b c
d e f

**FIGURE 10.46** (a) Noisy, shaded image and (b) its histogram. (c) Segmentation of (a) using the iterative global algorithm from Section 10.3.2. (d) Result obtained using Otsu's method. (e) Image subdivided into six subimages. (f) Result of applying Otsu's method to each subimage individually.

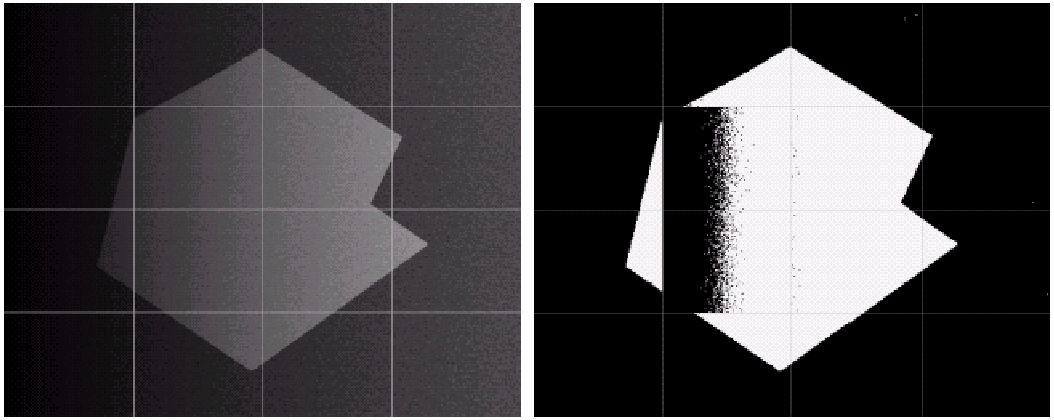
# Basic Adaptive Thresholding



Histogram of six subimages

# Basic Adaptive Thresholding Example

The image below shows an example of using adaptive thresholding with the image shown previously.

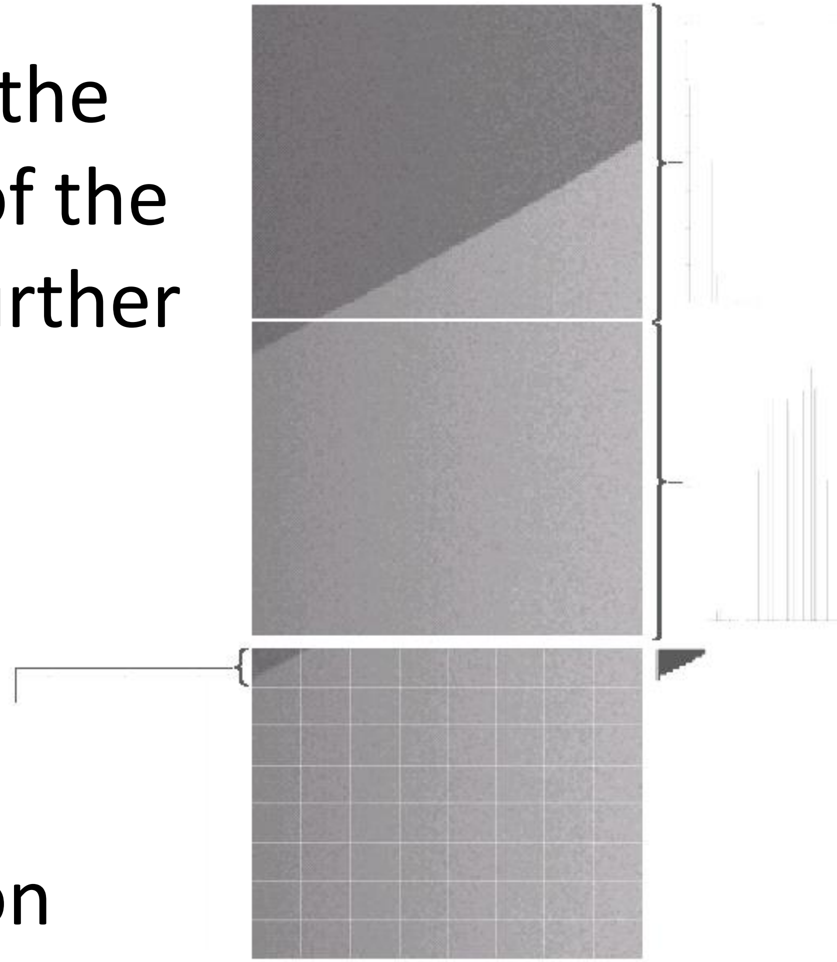


As can be seen success is mixed.

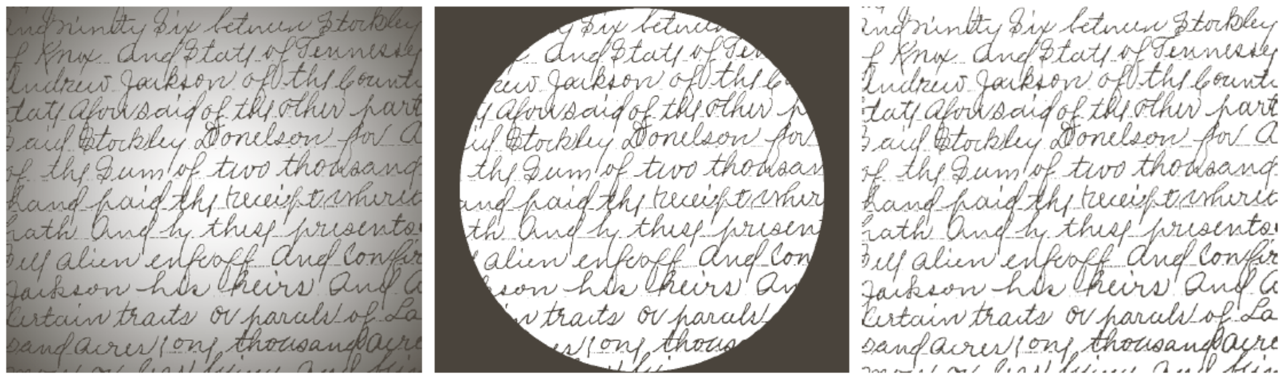
But, we can further subdivide the troublesome sub images for more success.

# Basic Adaptive Thresholding Example

These images show the troublesome parts of the previous problem further subdivided.



After this sub division successful thresholding can be achieved.



a b c

**FIGURE 10.49** (a) Text image corrupted by spot shading. (b) Result of global thresholding using Otsu's method. (c) Result of local thresholding using moving averages.

a b c

**FIGURE 10.50** (a) Text image corrupted by sinusoidal shading. (b) Result of global thresholding using Otsu's method. (c) Result of local thresholding using moving averages.

# Thresholding Modifications

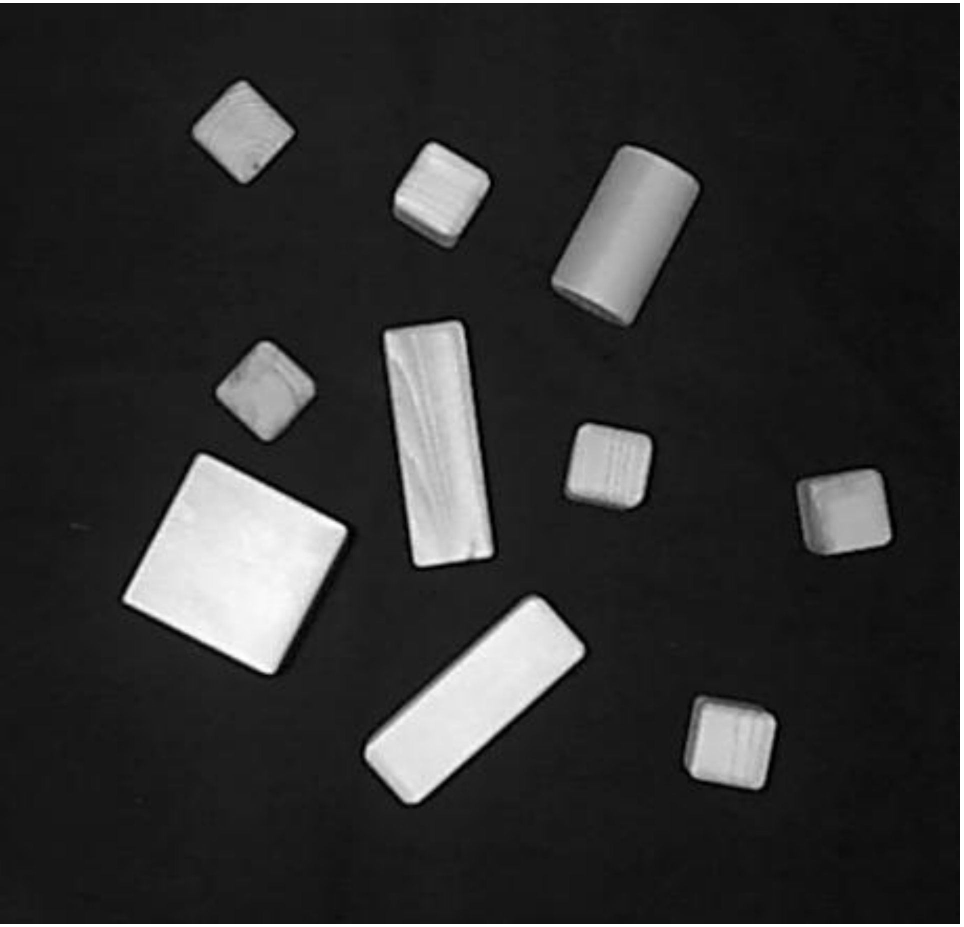
**Band-thresholding**

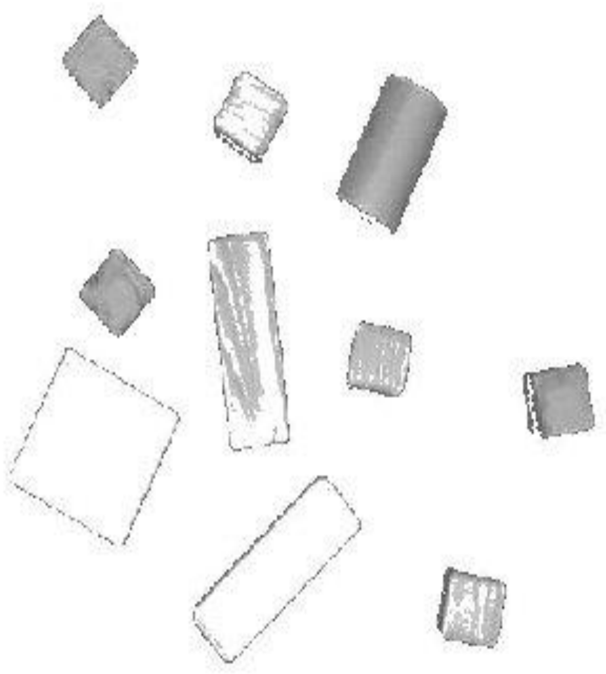
Segment an image into regions of pixels with gray levels from a set D and into background otherwise

$$
\begin{aligned}
g(i,j) &= 1 \quad \text{for } f(i,j) \in D \\
&= 0 \quad \text{otherwise}
\end{aligned}
$$

Can also serve as border detection

# Example





Intensity values between 90 and 200

# Other Thresholds

**Multithresholding**

Resulting image is no longer binary

$$
\begin{aligned}
g(i,j) &= 1 \quad \text{for } f(i,j) \in D_1 \\
&= 2 \quad \text{for } f(i,j) \in D_2 \\
&= 3 \quad \text{for } f(i,j) \in D_3 \\
&= 4 \quad \text{for } f(i,j) \in D_4 \\
&\ldots \\
&= n \quad \text{for } f(i,j) \in D_n \\
&= 0 \quad \text{otherwise}
\end{aligned}
$$

# Example

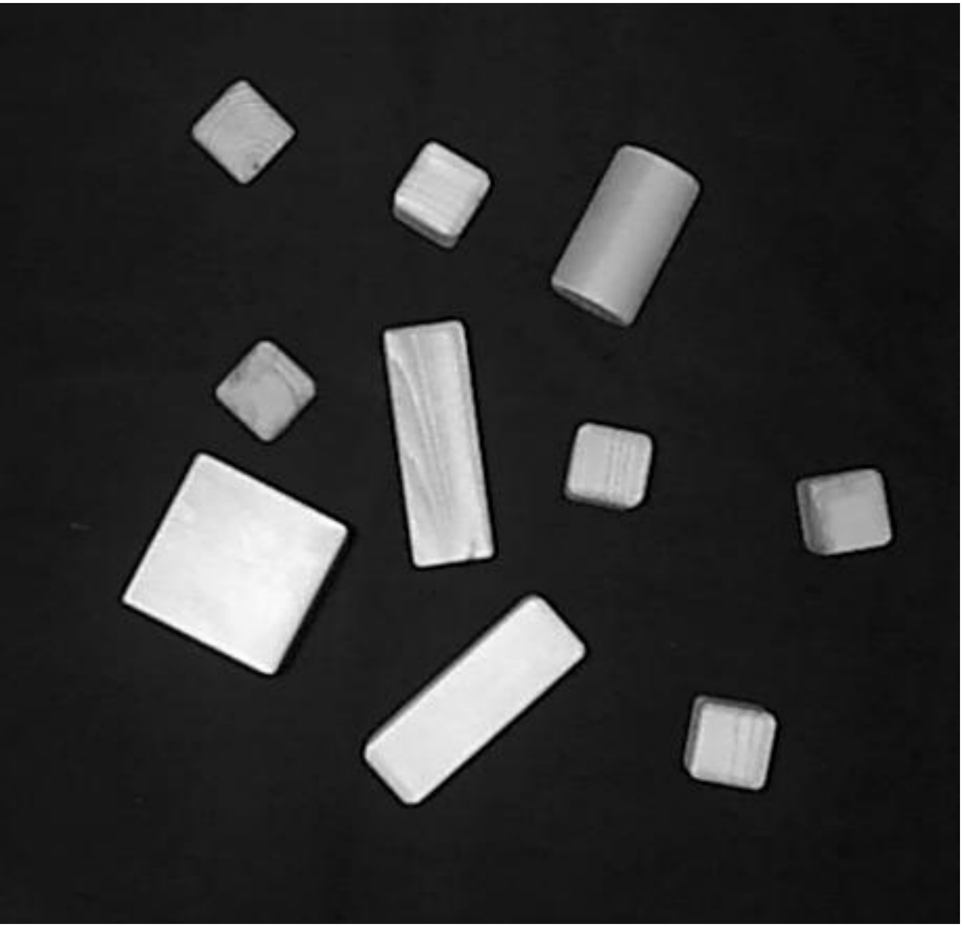

# Other Thresholds

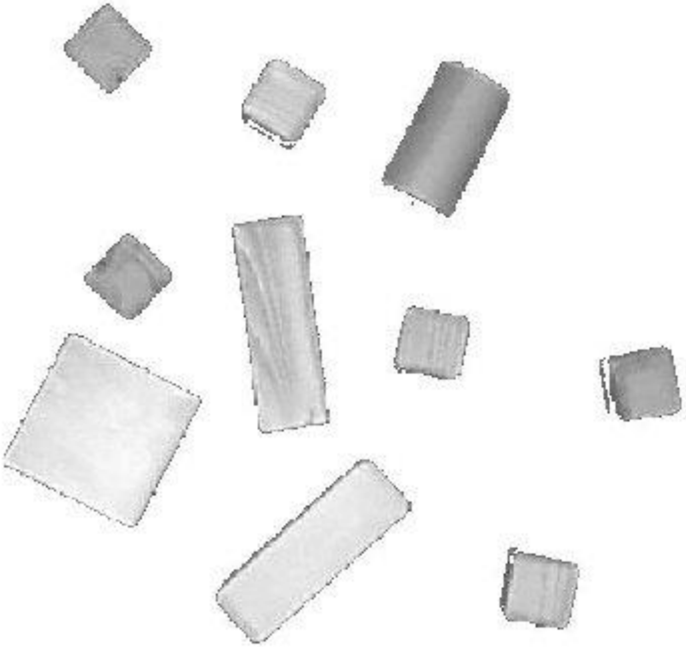
**Semi-thresholding**

Aims to mask out the image background leaving gray level information present in the objects

$$
\begin{aligned}
g(i,j) &= f(i,j) \quad && \text{for } f(i,j) \geq T \\
&= 0 && \text{for } f(i,j) < T
\end{aligned}
$$

# Example





# Summary

We have discussed pixel-based segmentation approaches.

We have seen the basic global thresholding algorithm and its shortcomings.

We have also seen a simple way to overcome some of these limitations using adaptive thresholding.

# Region-based Segmentation

# Why Region-Based Segmentation?

Segmentation

- Thresholding is not always effective.

Homogenous regions

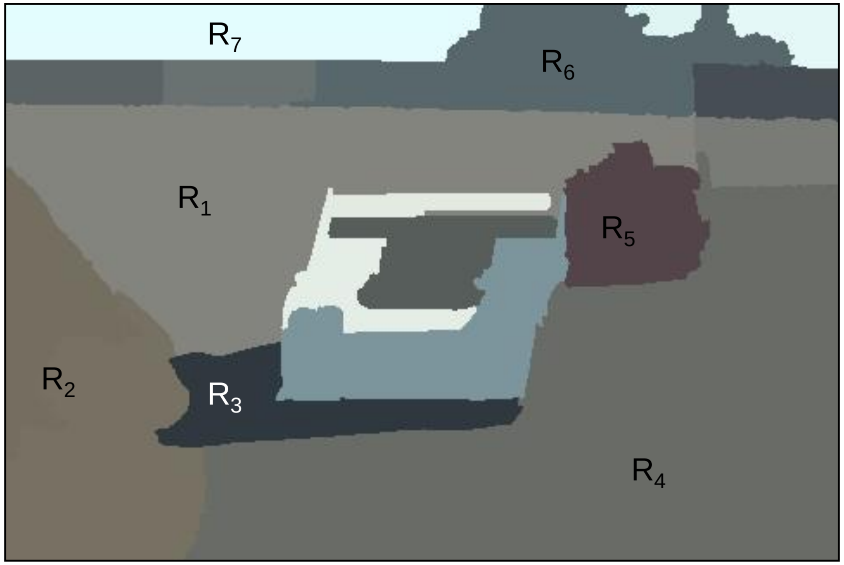
- *Region-based segmentation.*
- Effective in noisy images.


# Definitions

Based on *sets.*

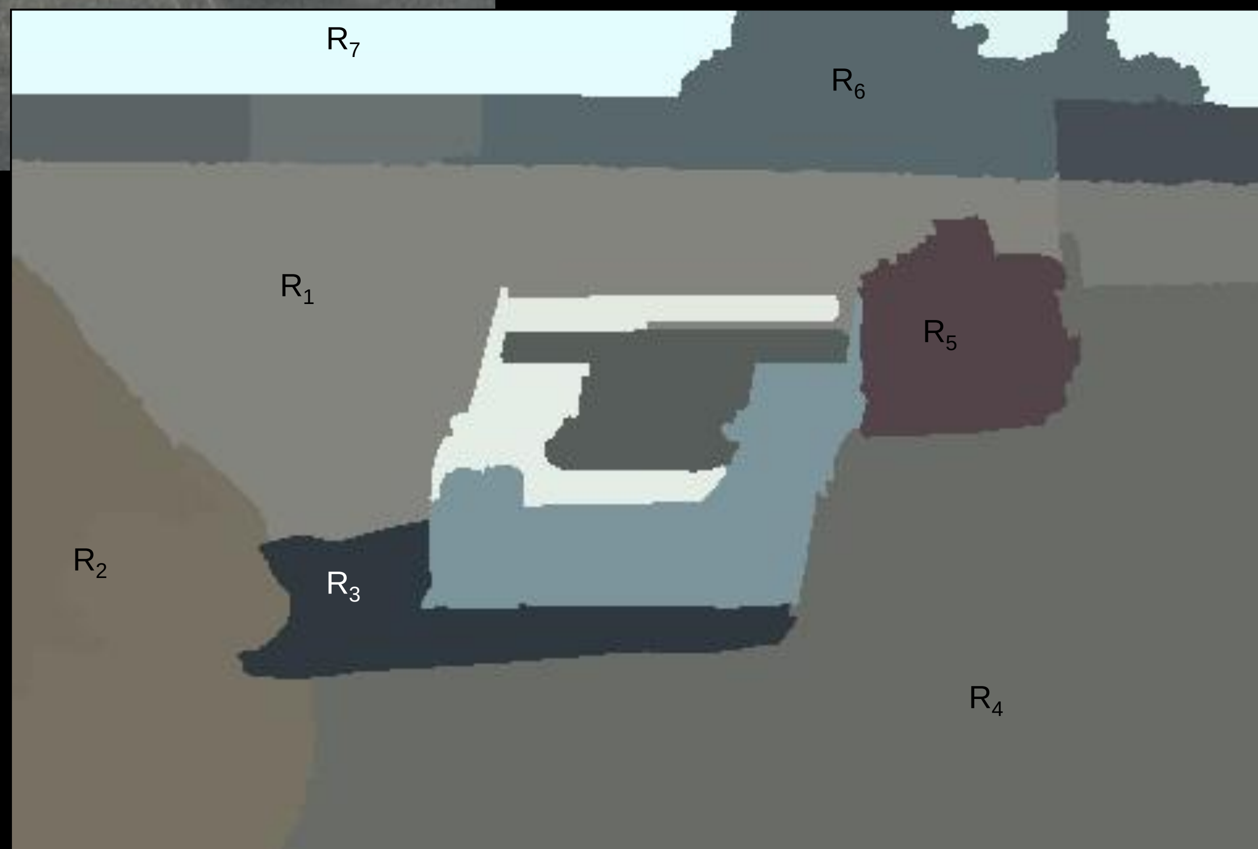
Each image R is a set of regions $R_i$

- Every pixel belongs to one region.
- One pixel can only belong to a single region.




$$R = \bigcup_{i=1}^{S} R_i \qquad R_i \bigcap R_j = \varnothing$$


CASTROL
Castrol
96
33


$R_7$
$R_6$
$R_1$
$R_5$
$R_2$
$R_3$
$R_4$

# Basic Formulation

Let R represent the entire image region. Segmentation partitions R into n subregions, $R_1$, $R_2$, ..., $R_n$, such that:

a) $\bigcup_{i=1}^{n} R_i = R$

b) $R_i$ is a connected region, $i = 1, 2, \ldots, n$.

c) $R_i \cap R_j = \phi$ for all $i$ and $j$, $i \neq j$

d) $P(R_i) = TRUE$ for $i = 1, 2, \ldots, n$.

e) $P(R_i \cup R_j) = FALSE$ for $i \neq j$.

a) Every pixel must be in a region.

b) Points in a region must be connected.

c) Regions must be disjoint.

d) All pixels in a region satisfy specific properties.

e) Different regions have different properties.

# How do we form regions?

Region Growing

Region Merging

Region Splitting

Split and Merge

…

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 0 | 3 | 2 | 5 | 4 | 7 | 6 | 9 | 8 |
| 3 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 2 | 1 | 0 | 3 | 2 | 5 | 4 | 7 | 6 |
| 5 | 2 | 3 | 0 | 1 | 2 | 3 | 4 | 5 |
| 4 | 3 | 2 | 1 | 0 | 3 | 2 | 5 | 4 |
| 7 | 4 | 5 | 2 | 3 | 0 | 1 | 2 | 3 |
| 6 | 5 | 4 | 3 | 2 | 1 | 0 | 3 | 2 |
| 9 | 6 | 7 | 4 | 5 | 2 | 3 | 0 | 1 |
| 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

Computer View

# Similarity Criteria

Homogeneity of regions is used as the main segmentation criterion in region growing.

- gray level
- color, texture
- shape
- model
- etc.

Choice of criteria affects segmentation results dramatically!

# Region Growing

Groups pixels into larger regions.

Starts with a **seed** region.

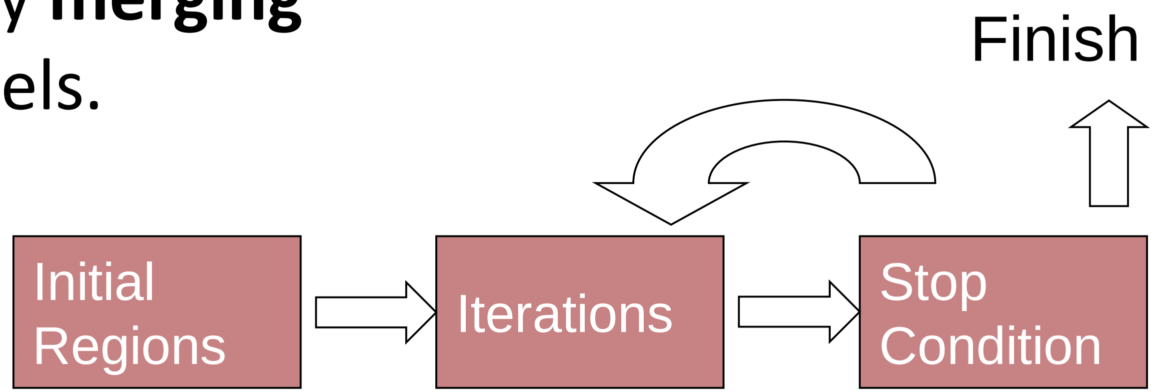
**Grows** region by **merging** neighboring pixels.

Iterative process

How to start?

How to iterate?

When to stop?



# Region Growing

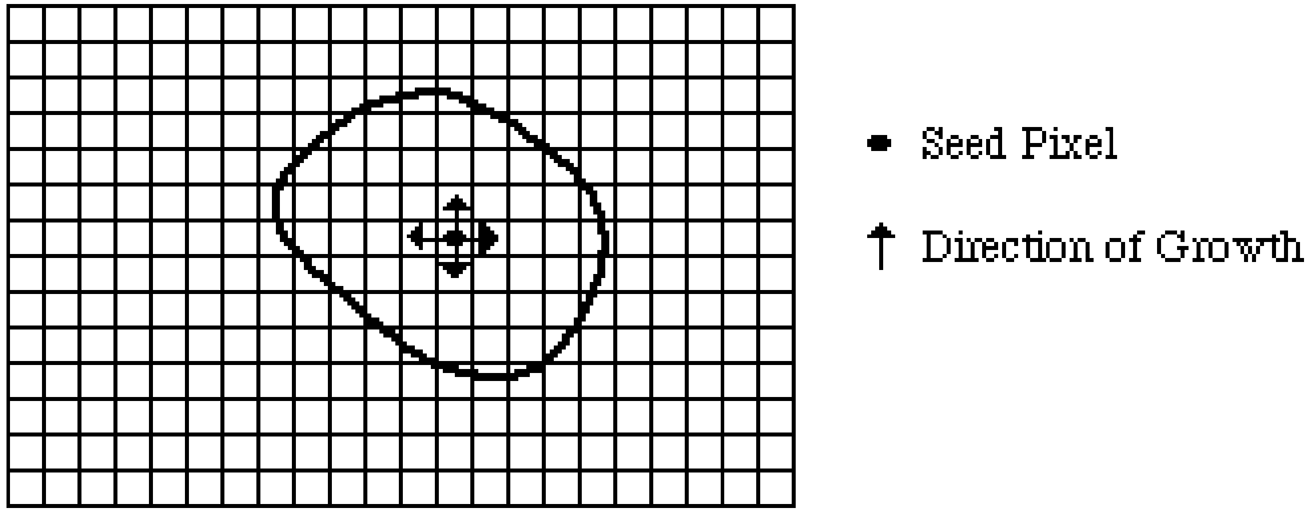
Let us pick up an arbitrary pixel from the domain of the image to be segmented. This pixel is called seed pixel and belongs to a particular region.

Now, examine the nearest neighbours (4- or 8-) of the seed pixel one by one, and a neighbouring pixel is accepted as a pixel of the same region as the seed pixel if they together satisfy the homogeneity property of a region.

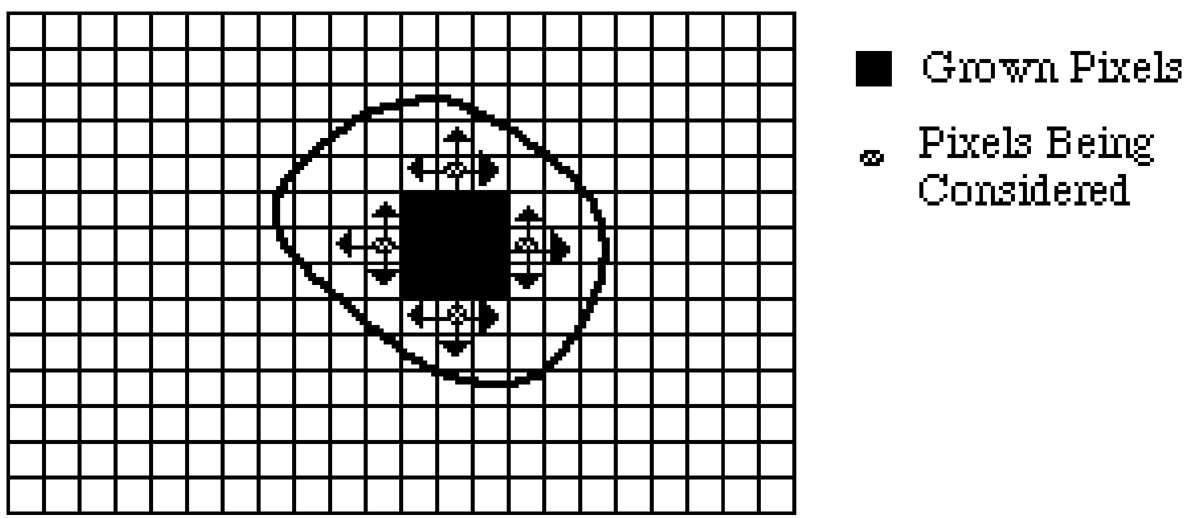
Once a new pixel is accepted as a member of the current region the nearest neighbours of this new pixel are examined.

This process goes on recursively until no more pixel is accepted.

All the pixels of the current region are marked with a unique label.




(a) Start of Growing a Region




(b) Growing Process After a Few Iterations

# Region Growing Example

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 5 | 6 | 6 | 6 | 7 | 7 | 6 | 6 |
| 6 | 7 | 6 | 7 | 5 | 5 | 4 | 7 |
| 6 | 6 | 4 | 4 | 3 | 2 | 5 | 6 |
| 5 | 4 | 5 | 4 | 2 | 3 | 4 | 6 |
| 1 | 3 | 2 | 3 | 3 | 2 | 4 | 7 |
| 0 | 0 | 1 | 0 | 2 | 2 | 5 | 6 |
| 1 | 1 | 0 | 1 | 0 | 3 | 4 | 4 |
| 1 | 0 | 1 | 0 | 2 | 3 | 5 | 6 |

Input

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 5 | 6 | 6 | 6 | 7 | 7 | 6 | 6 |
| 6 | 7 | 6 | 7 | 5 | 5 | 4 | 7 |
| 6 | 6 | 4 | 4 | 3 | 2 | 5 | 6 |
| 5 | 4 | 5 | 4 | 2 | 3 | 4 | 6 |
| 1 | 3 | 2 | 3 | 3 | 2 | 4 | 7 |
| 0 | 0 | 1 | 0 | 2 | 2 | 5 | 6 |
| 1 | 1 | 0 | 1 | 0 | 3 | 4 | 4 |
| 1 | 0 | 1 | 0 | 2 | 3 | 5 | 6 |

Output

Property ($R$): $\max\{g(r, c)\} - \min\{g(r, c)\} \leq 3$

# Region Splitting

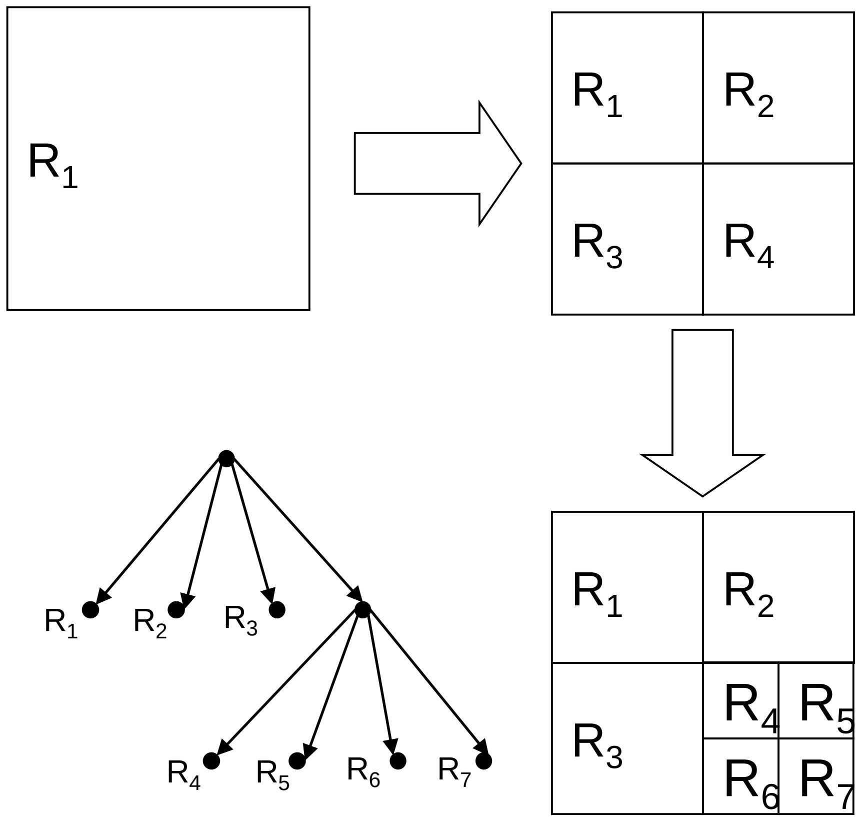
## Algorithm:

- One initial set that includes the **whole image.**

- Similarity criteria.

- Iteratively **split** regions into sub-regions.

- Stop when no more splittings are possible.



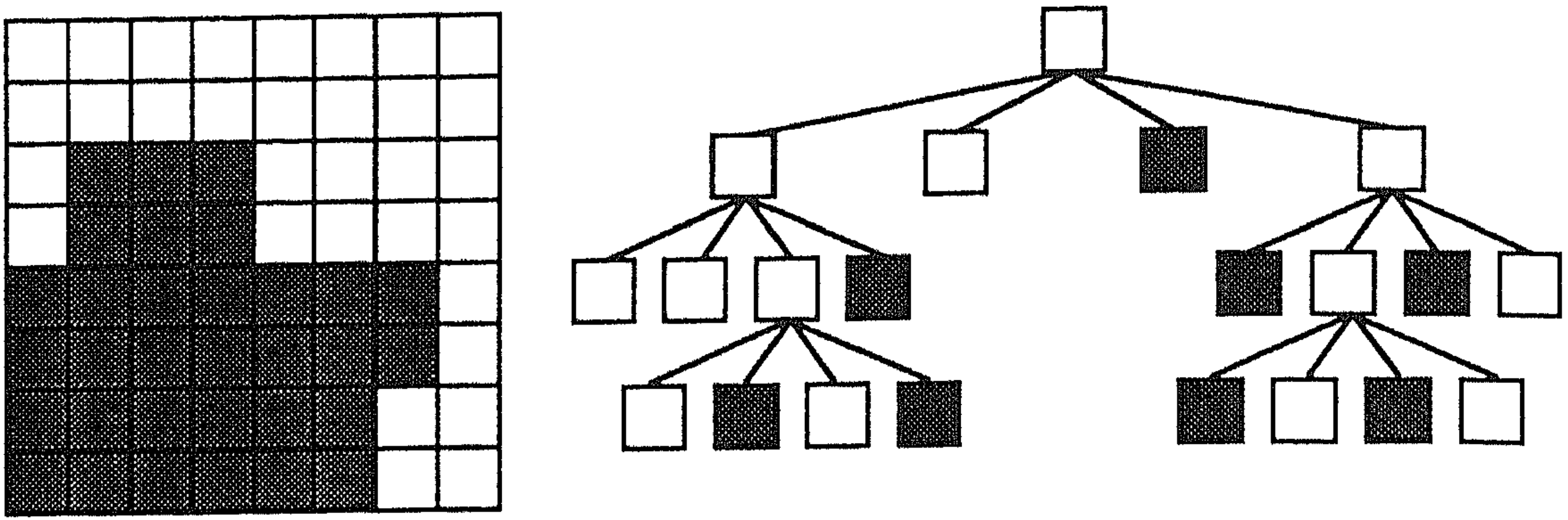


**Figure 5.23** A quad-tree representation of an $8 \times 8$ binary image.

# Region Splitting Example

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 5 | 6 | 6 | 6 | 7 | 7 | 6 | 6 |
| 6 | 7 | 6 | 7 | 5 | 5 | 4 | 7 |
| 6 | 6 | 4 | 4 | 3 | 2 | 5 | 6 |
| 5 | 4 | 5 | 4 | 2 | 3 | 4 | 6 |
| 1 | 3 | 2 | 3 | 3 | 2 | 4 | 7 |
| 0 | 0 | 1 | 0 | 2 | 2 | 5 | 6 |
| 1 | 1 | 0 | 1 | 0 | 3 | 4 | 4 |
| 1 | 0 | 1 | 0 | 2 | 3 | 5 | 6 |

Input

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 5 | 6 | 6 | 6 | 7 | 7 | 6 | 6 |
| 6 | 7 | 6 | 7 | 5 | 5 | 4 | 7 |
| 6 | 6 | 4 | 4 | 3 | 2 | 5 | 6 |
| 5 | 4 | 5 | 4 | 2 | 3 | 4 | 6 |
| 1 | 3 | 2 | 3 | 3 | 2 | 4 | 7 |
| 0 | 0 | 1 | 0 | 2 | 2 | 5 | 6 |
| 1 | 1 | 0 | 1 | 0 | 3 | 4 | 4 |
| 1 | 0 | 1 | 0 | 2 | 3 | 5 | 6 |

Output

Property ($R$): $\max\{g(r, c)\} - \min\{g(r, c)\} \leq 3$

# Region Merging

Algorithm:

- Divide image into an initial set of regions.
  - One region per pixel.
- Define a **similarity criteria** for merging regions.
- **Merge** similar regions.
- Repeat previous step until no more merge operations are possible.

# Region Merging Example

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 5 | 6 | 6 | 6 | 7 | 7 | 6 | 6 |
| 6 | 7 | 6 | 7 | 5 | 5 | 4 | 7 |
| 6 | 6 | 4 | 4 | 3 | 2 | 5 | 6 |
| 5 | 4 | 5 | 4 | 2 | 3 | 4 | 6 |
| 1 | 3 | 2 | 3 | 3 | 2 | 4 | 7 |
| 0 | 0 | 1 | 0 | 2 | 2 | 5 | 6 |
| 1 | 1 | 0 | 1 | 0 | 3 | 4 | 4 |
| 1 | 0 | 1 | 0 | 2 | 3 | 5 | 6 |

Input

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 5 | 6 | 6 | 6 | 7 | 7 | 6 | 6 |
| 6 | 7 | 6 | 7 | 5 | 5 | 4 | 7 |
| 6 | 6 | 4 | 4 | 3 | 2 | 5 | 6 |
| 5 | 4 | 5 | 4 | 2 | 3 | 4 | 6 |
| 1 | 3 | 2 | 3 | 3 | 2 | 4 | 7 |
| 0 | 0 | 1 | 0 | 2 | 2 | 5 | 6 |
| 1 | 1 | 0 | 1 | 0 | 3 | 4 | 4 |
| 1 | 0 | 1 | 0 | 2 | 3 | 5 | 6 |

Output

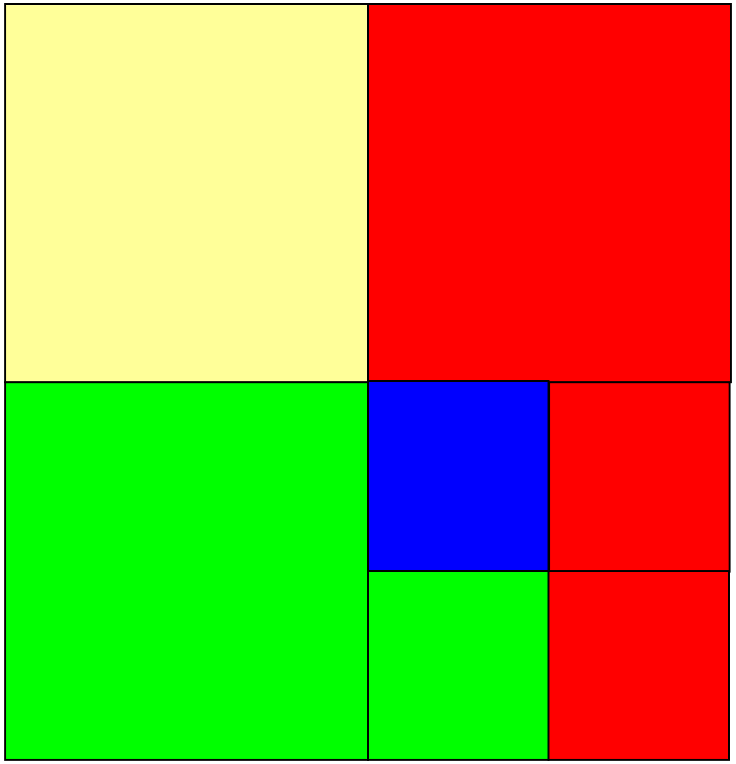
Property ($R$): $\max\{g(r, c)\} - \min\{g(r, c)\} \leq 3$

# Split and Merge

Combination of both algorithms.

Can handle a larger variety of shapes.

- Simply apply previous algorithms consecutively.



# Region Formation

Split or merge, or split-merge technique results in non-overlapping rectangular homogeneous regions that satisfy all conditions except 5th one.

We use Region Adjacency Graph (RAG) to solve this problem.

# Region Formation

If the predicate is satisfied over union of adjacent regions then adjacent regions are grouped and labelled as a single region by some post-processing.

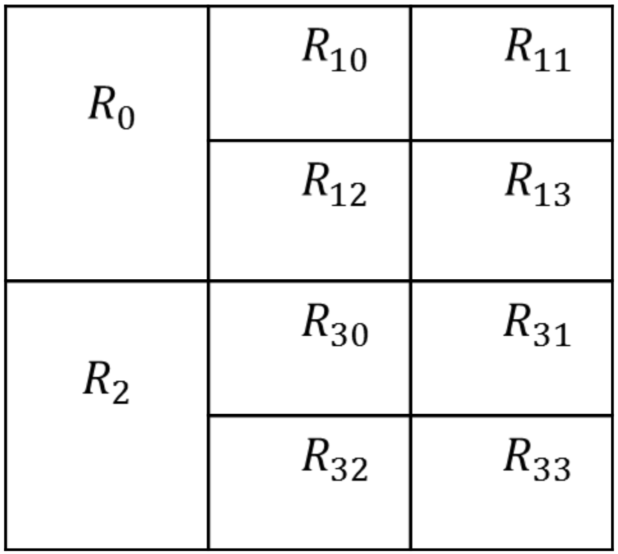
This post-processing method is described using a special type of data structure known as region adjacency graph (RAG).

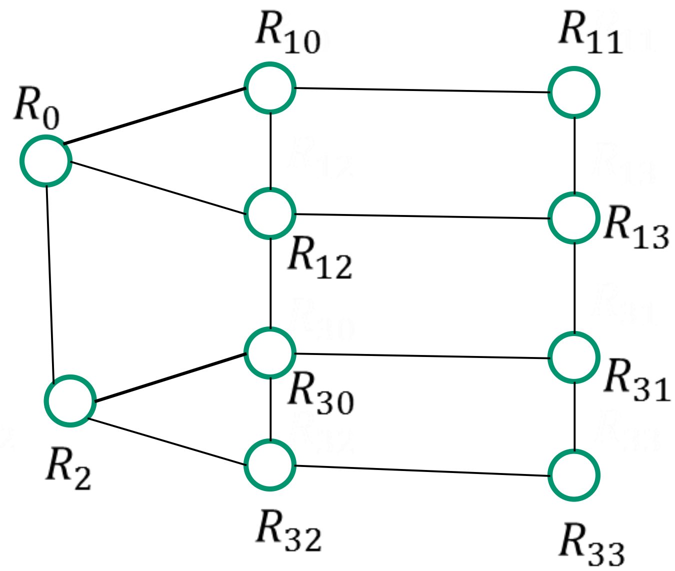
# Region Adjacency Graph

If two regions share a common boarder, they are said to be adjacent and corresponding nodes in the graph are connected by an edge.

The homogeneity property is tested over every pair of regions whose corresponding nodes in RAG directly connected by an edge.

# Region Adjacency Graph



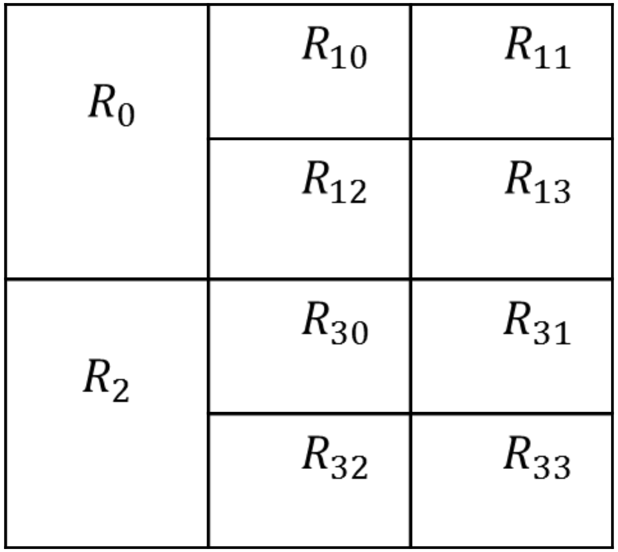
# Region Adjacency Graph

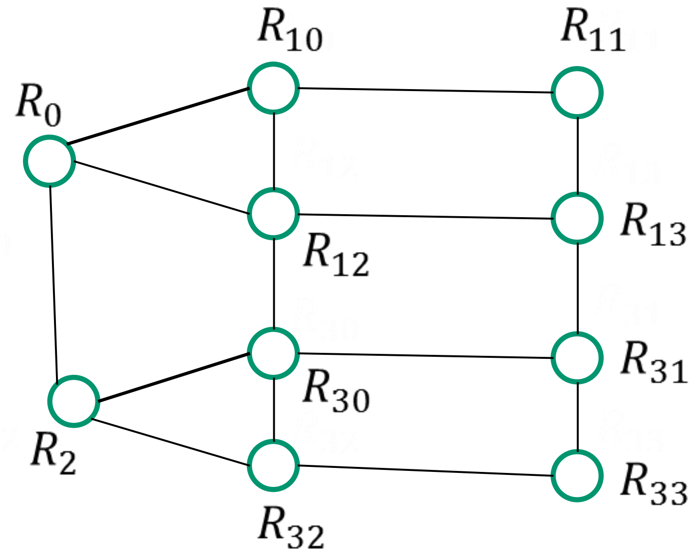
If homogeneity property between two nodes does not satisfy then the edge between these nodes is removed.

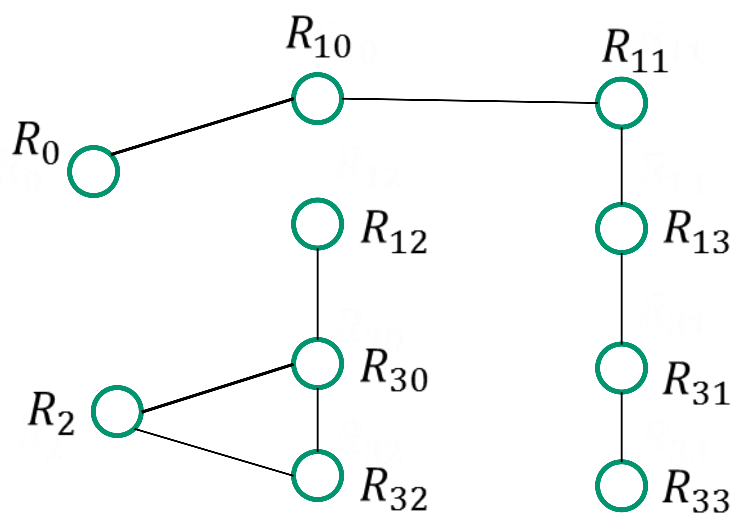
As a result, a set of disconnected graphs are formed.

Each of such graph represents a maximal region that is homogeneous.

# Region Adjacency Graph



# Region Adjacency Graph

The problem of this approach is that resultant sub-graphs depend on the order in which node pairs are examined for homogeneity.

*RAG* may also be used to merge small regions to adjacent large regions obtained from other segmentation techniques.

# Summary

We have begun looking at segmentation, and in particular region-based approaches.

We saw the basic region-based algorithms.

We have also seen the effect of different region-based approaches.