Digital Image Processing

Segmentation

Topics

- Sample filtering in frequency domain (continued from last week)
- Introduction
- Segmentation of Binary Images
  - Blob Coloring
- Segmentation of Images with Multiple Gray Levels
  - Thresholding
  - Region Growing
  - Split and Merge

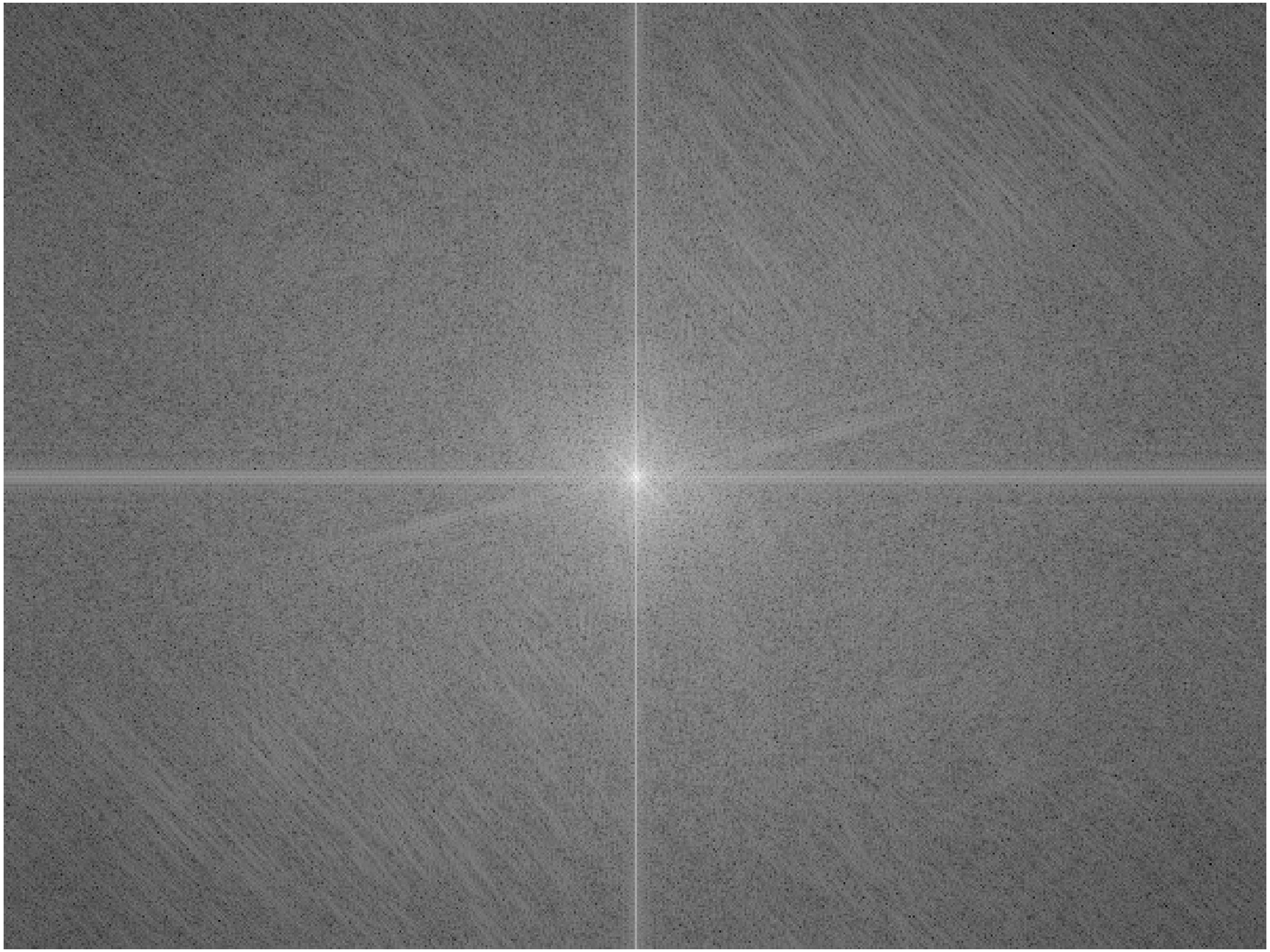
Fourier Transform in MATLAB

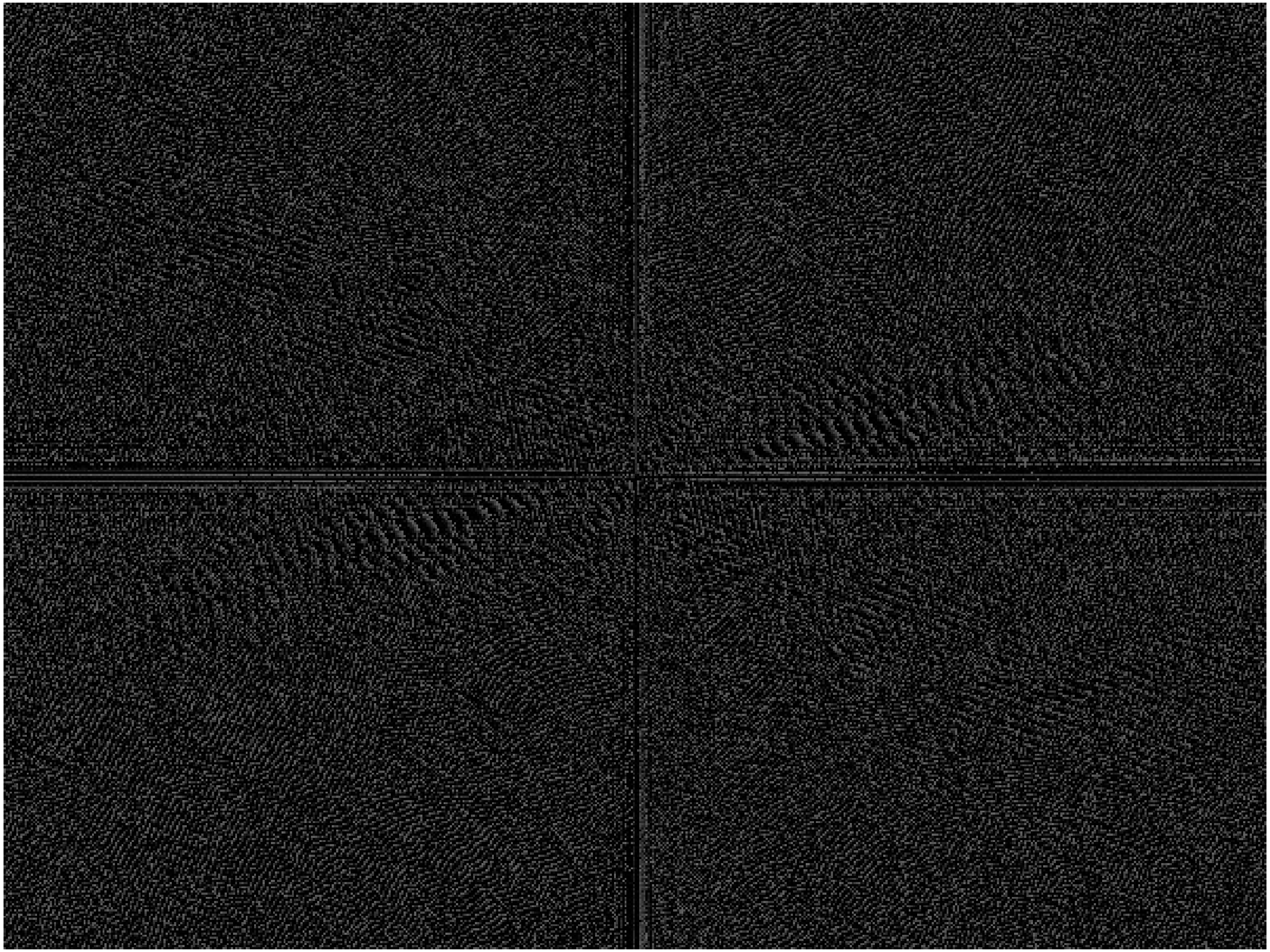
- Use fft2(.) for forward fast Fourier transform
- Use ifft2(.) for inverse fast Fourier transform
- Use abs(.) to get the magnitude of the Fourier transform
- Use angle(.) to find the phase of the Fourier transform
- To find the Fourier transform from magnitude and phase values use: Z = R.*exp(i*theta)
- Use real(.) and Imag(.) to get the real and imaginary parts of the Fourier transform resepectively
- Use fftshift(.) to shift the origin of the transform to the center.

Applying Low-Pass Filtering in Frequency Domain

- Open the image: Im = imread('test.jpg');
- Find Fourier transform of the image FIm = fft2(Im);
- Create a filter
  - h=[1   1   1; 1   1   1; 1   1   1];
  - h=h/9;
  - H=fft2(h, size(Im,1), size(Im,2));
- Multiply Fourier transform of the image by the Fourier transform of the filter
  - R = FIm.*H;
- Find inverse of the result: r = ifft(R);
- Keep the real part of the result: r = real(r);
- Display: imshow(uint8(r));

Example





Segmentation

Introduction

- Image segmentation is the process of partitioning the digital image into multiple regions that can be associated with the properties of one or more objects

- It is an initial and vital step in pattern recognition-a series of processes aimed at overall image understanding.

Definition

In mathematical sense the segmentation of the image *I*, which is a set of pixels, is partitioning *I* into *n* disjoint sets $R_1, R_2, \ldots, R_n$, called segments or regions such that the union of all regions equals *I*.

$$I = R_1 \cup R_2 \cup \ldots \cup R_n$$

Segmentation of Binary Images

- Since binary images contain only black or white pixels, segmenting objects from the background is trivial.

- Separating objects from each other is based on the neighborhood relationship of the pixels.

Blob Coloring

- Blob coloring is applied to a binary image for segmenting and labeling each object using a different color.

- 4-neighborhood or 8-neighborhood can be used for segmentation

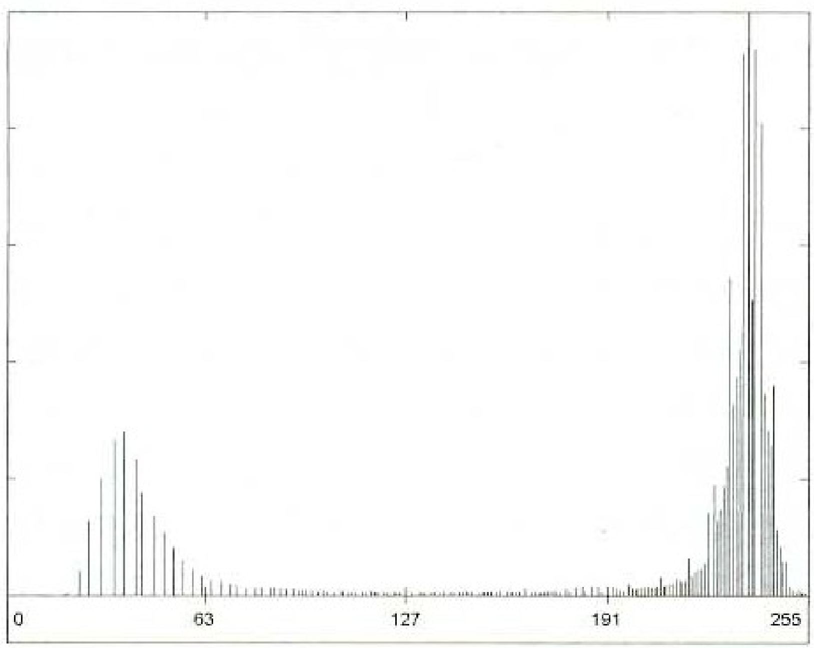
Blob Coloring Algorithm

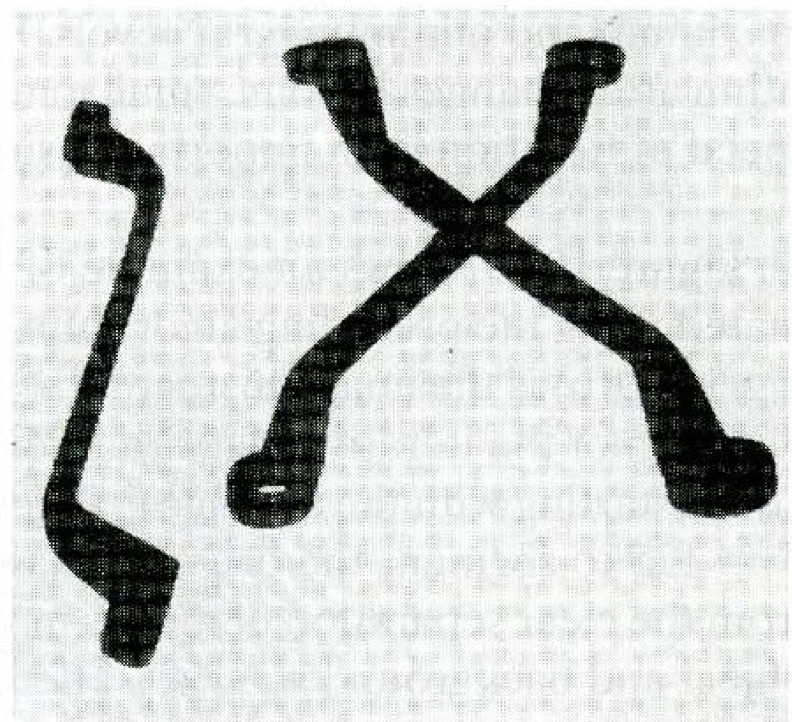
- Let the initial color k=1, scan the image from left to right and top to bottom
- If $f(x_c) = 0$ then continue
- Else
  - If($f(x_u) = 1$and $f(x_L) = 0$)
    - Color x_c = color x_u
  - If($f(x_L) = 1$ and $f(x_u) = 0$)
    - Color x_c = color x_L
  - If($f(x_L = 1$ and $f(x_u) = 1$)
    - Color x_c = color x_L
    - Color x_L equivalent to Color x_u
  - If($f(x_L) = 0$ and $f(x_u) = 0$)
    - Color x_c = k
    - K=k+1

Segmentation by Thresholding

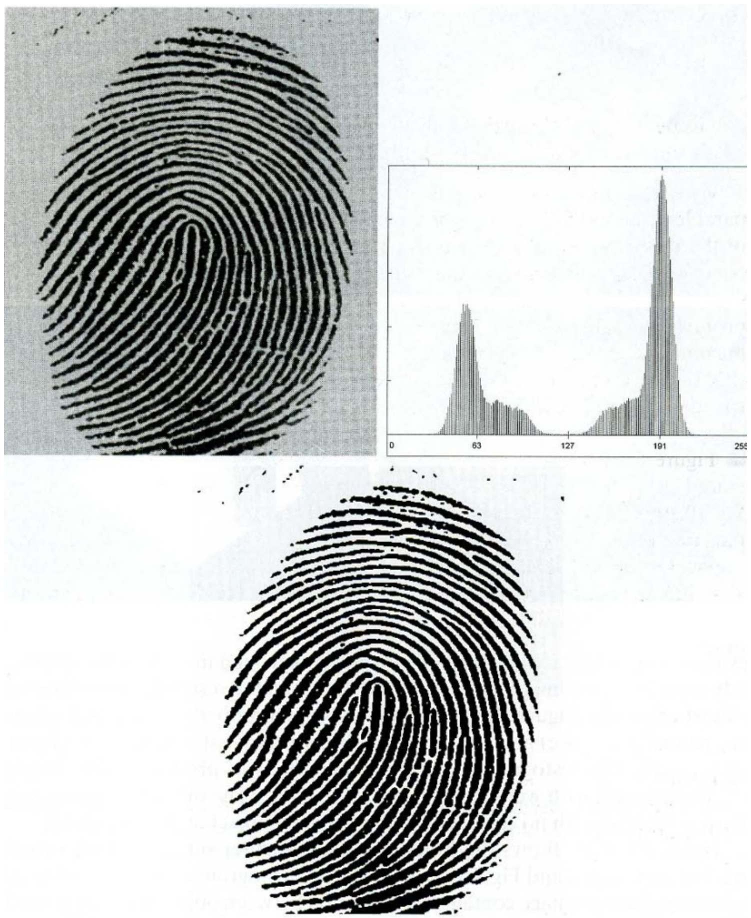
- Thresholding: segment scalar images by creating a binary partitioning of the image intensities.
- All pixels with a value greater than a threshold value are classified as pixels of the object and the others as the background (or vice-versa)
- Finding a suitable threshold value is not always simple

Using Histogram for Selecting the Threshold Value





Example



Estimating the Threshold Value

1. Select an initial estimate for T.
2. Segment the image using T. This will produce two groups of pixels: G_1 consisting of all pixels with gray level values $>T$ and G_2 consisting of pixels with values $\leq T$.
3. Compute the average gray level values μ_1 and μ_2 for the pixels in regions G_1 and G_2.
4. Compute a new threshold value:

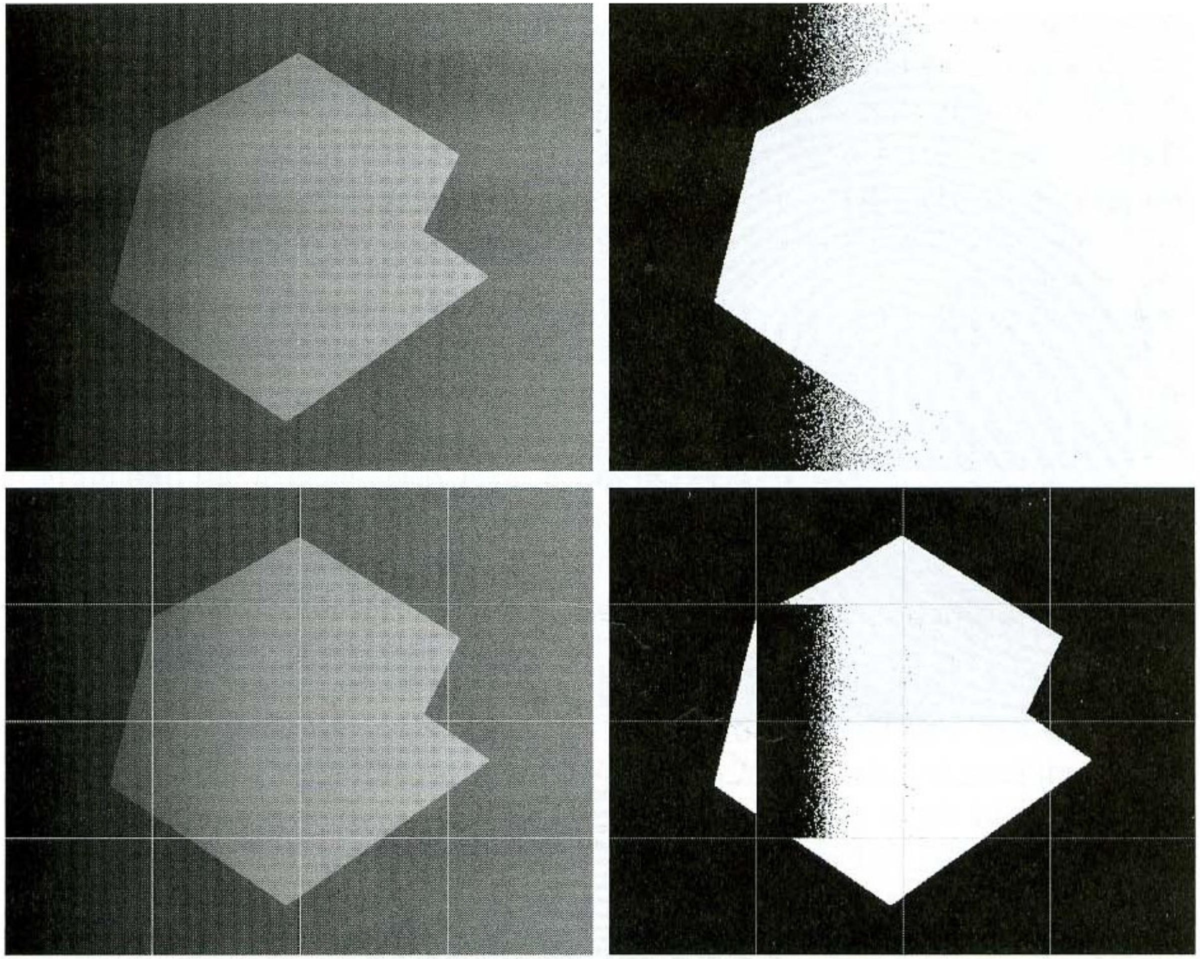
$$T = \frac{1}{2}(\mu_1 + \mu_2).$$

5. Repeat steps 2 through 4 until the difference in T in successive iterations is smaller than a predefined parameter T_o.

Adaptive Thresholding

- A single (global) threshold value may not be available for all images.

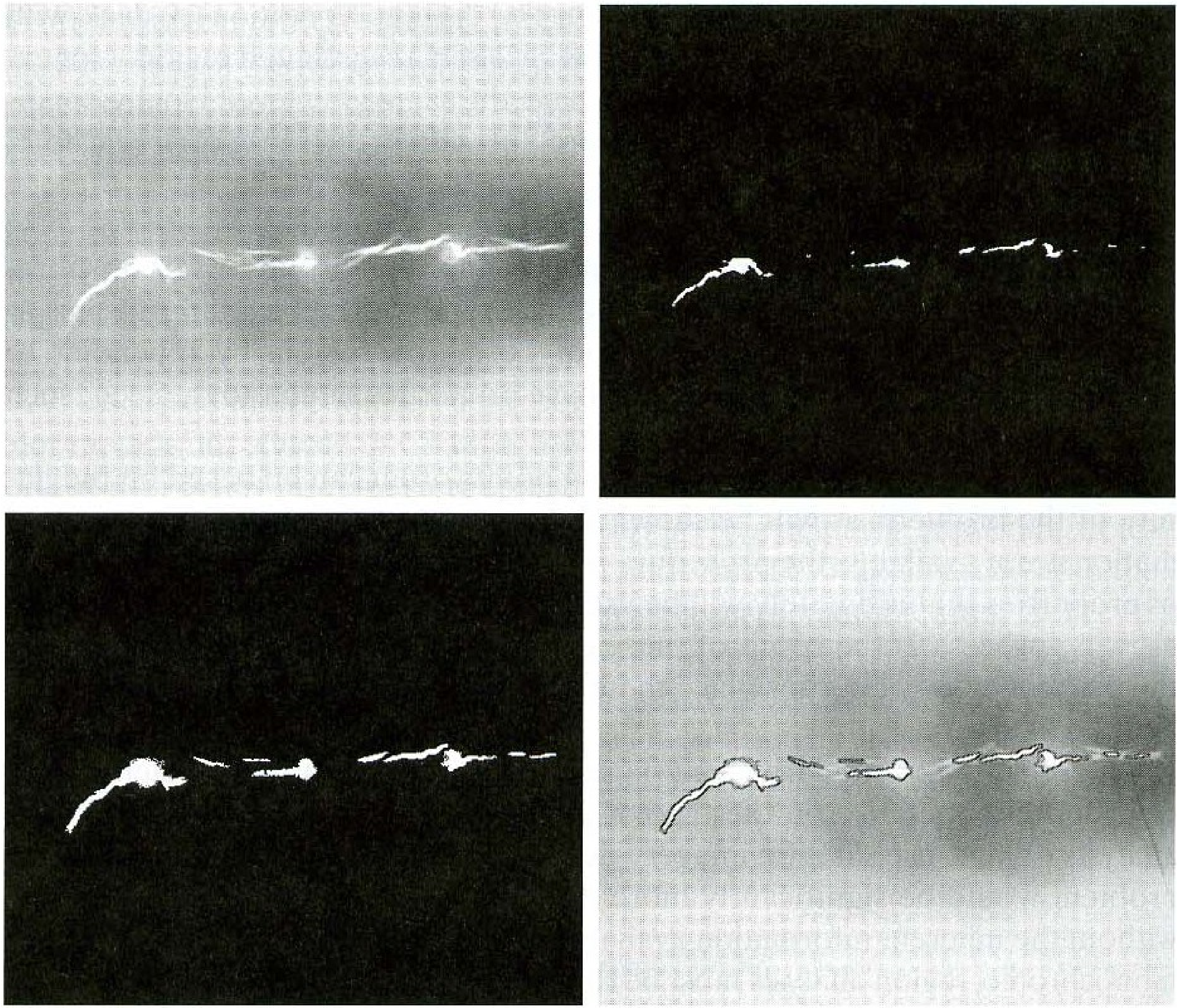
- A local threshold can be found from the local processing of the image.



Region Growing

- Begins with a set of seed points and from them grows regions by appending neighboring pixels that have properties similar to initial seed.

- Gray level, texture, color, and other local features are used for measuring the similarity

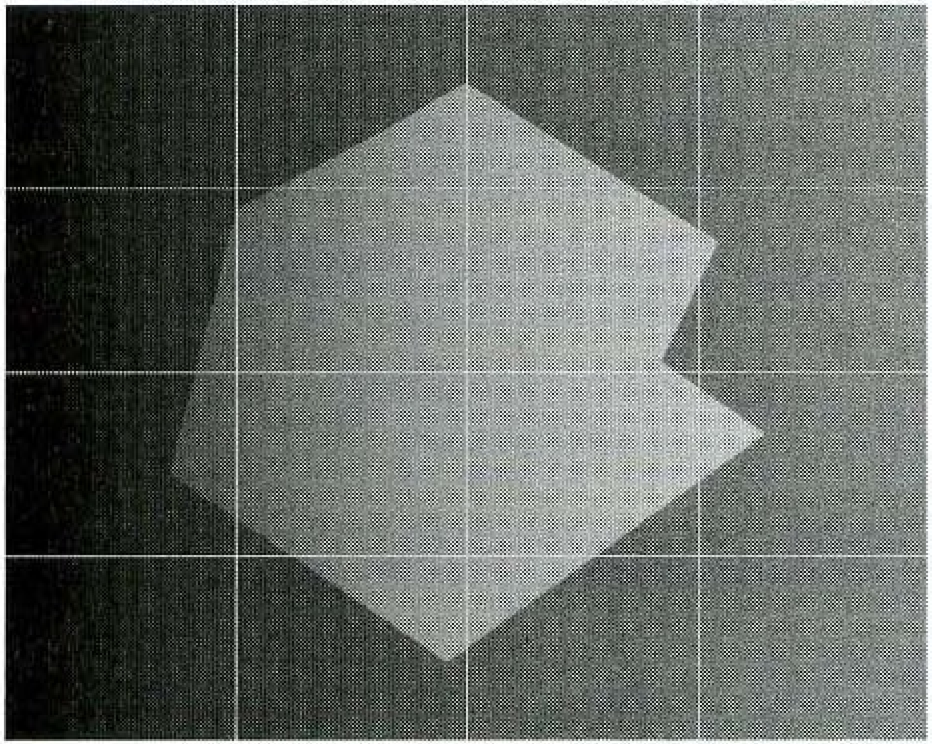


Region Growing Problems

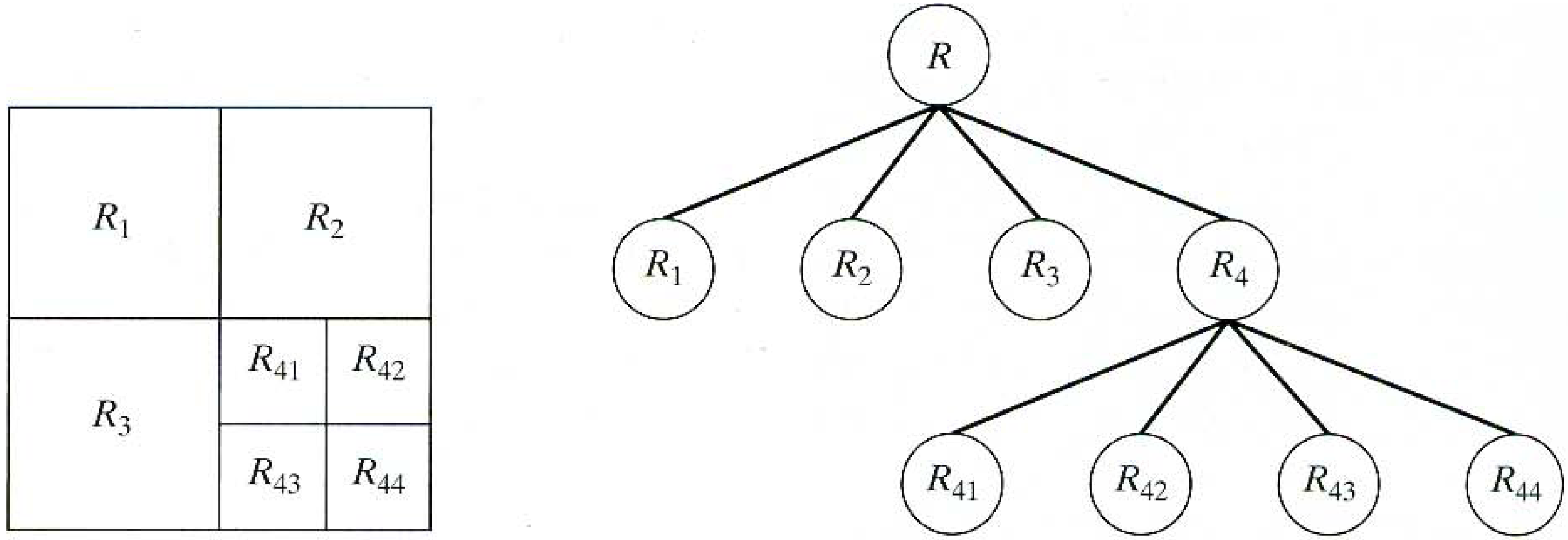
- Selecting initial seed
- Selecting suitable properties for including points
  - Example: In military applications using infra red images, the target of interest is slightly hotter than its environment

Region Split and Merge

- Divide the image into a set of arbitrary disjoint regions.
- Merge/split the regions



Quad-Tree Representation



The Use of Motion in Segmentation

- Compare two image taken at times t1 and t2 pixel by pixel (difference image)
- Non-zero parts of the difference image corresponds to the non-stationary objects

$$d_{ij}(x,y) = \begin{cases} 1 & \text{if } |f(x,y,t_1) - f(x,y,t_2)| > \theta \\ 0 & \text{otherwise} \end{cases}$$

Accumulating Differences

- A difference image may contain isolated entries that are the result of the noise
- Thresholded connectivity analysis can remove these points
- Accumulating difference images can also remove the isolated points

Questions?