Image segmentation

- Introduction
- Detection of Discontinuities
  - Point detection
  - Line detection
  - Edge detection
  - Combined detection
  - Edge linking and boundary detection
- Thresholding
  - Adaptive thresholding
  - Threshold selection based on boundary characteristics
- Region-oriented segmentation
  - Region growing by pixel aggregation
  - Region splitting and merging

1. Introduction

- The objective is to subdivide an image into its constituent parts or objects for subsequent processing such as recognition.
- It is one of the most important steps leading to the analysis of processed image data.

Complete v.s. patrial segmentation

- In *complete segmentation,*
  - ❑ Disjoint regions segmented are uniquely corresponding with objects in the input image.
  - ❑ Cooperation with higher processing levels which use specific knowledge of the problem domain is necessary.
- In *partial segmentation*,
  - ❑ Regions segmented do not correspond directly with image objects.

- Totally correct and complete segmentation of complex scenes usually can't be achieved.
- A reasonable aim is to use partial segmentation as an input to higher level processing.

Applications:

- Simple segmentation problems:
  1. Contrasted objects on a uniform background
  2. Simple assembly tasks, blood cells, printed characters, etc.

How to achieve segmentation?

- Image is divided into separate regions that are homogeneous with respect to a chosen property such as color, brightness, texture, etc.
- Segmentation algorithms generally are based on 2 basic properties of gray level values:
  1. Discontinuity - isolated points, lines and edges of image.
  2. Similarity - thresholding, region growing, region splitting and merging.
- Segmentation methods:
  1. Global approaches such as thresholding
  2. Edge-based segmentation
  3. Region-based segmentation

2. *Detection of Discontinuities:*

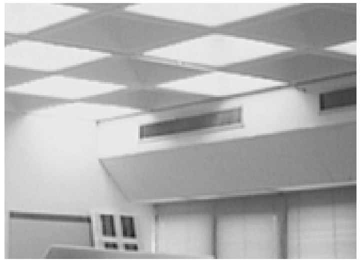
- There are 3 basic types of discontinuities: points, lines and edges.
- The detection is based on convoluting the image with a spatial mask.
- A general 3x3 mask $\begin{bmatrix} w_{-1,-1} & w_{-1,0} & w_{-1,1} \\ w_{0,-1} & w_{0,0} & w_{0,1} \\ w_{1,-1} & w_{1,0} & w_{1,1} \end{bmatrix}$
- The response of the mask at any point (x,y) in the image is $R_{x,y} = \sum_{i=-1}^{1} \sum_{j=-1}^{1} p(x-i, y-j)\, w(i, j)$

2.1 Point detection

- A point has been detected at the location p(i,j) on which the mask is centered if $|R| > T$, where T is a nonnegative threshold, and R is obtained with the following mask.

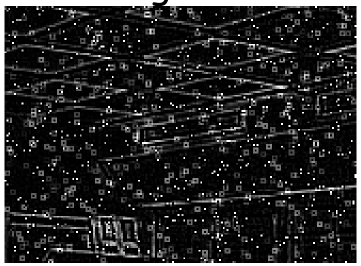
$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

- The idea is that the gray level of an isolated point will be quite different from the gray level of its neighbors.

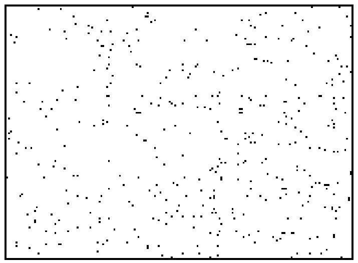


Original

Noise-added



Filtered o/p



Thresholded o/p

2.2 Line detection

- Line masks

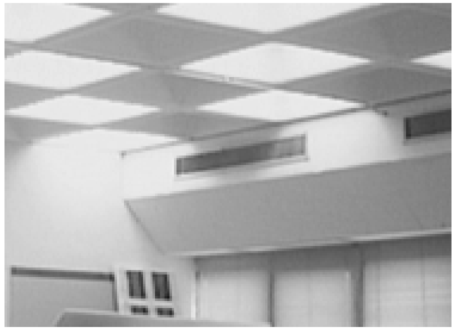
Horizontal line $\begin{bmatrix} -1 & -1 & -1 \\ 2 & 2 & 2 \\ -1 & -1 & -1 \end{bmatrix}$

45° line $\begin{bmatrix} -1 & -1 & 2 \\ -1 & 2 & -1 \\ 2 & -1 & -1 \end{bmatrix}$

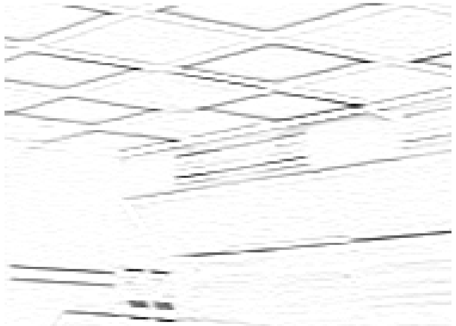
Vertical line $\begin{bmatrix} -1 & 2 & -1 \\ -1 & 2 & -1 \\ -1 & 2 & -1 \end{bmatrix}$

-45° line $\begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix}$

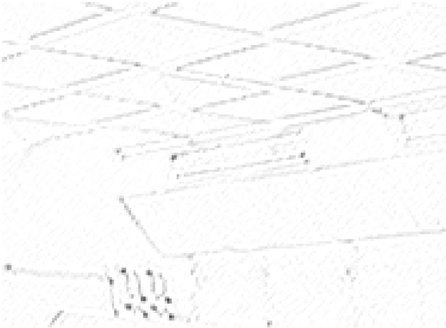
- If, at a certain point in the image, $|R_i|>|R_j|$ for all $j \neq i$, that point is said to be more likely associated with a line in the direction of mask *i*.



Original

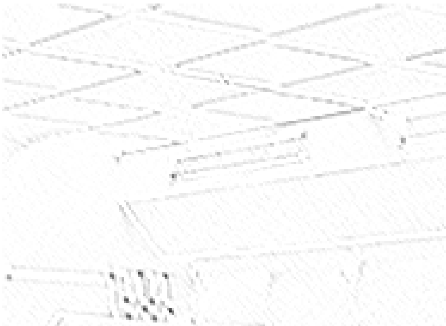


Horizontal line



45° line

Vertical line



-45° line

2.3 Edge detection

- It locates sharp changes in the intensity function.
- Edges are pixels where brightness changes abruptly.
- A change of the image function can be described by a gradient that points in the direction of the largest growth of the image function.
- An edge is a property attached to an individual pixel and is calculated from the image function behavior in a neighborhood of the pixel.
- Magnitude of the first derivative detects the presence of the edge.
- Sign of the second derivative determines whether the edge pixel lies on the dark sign or light side.

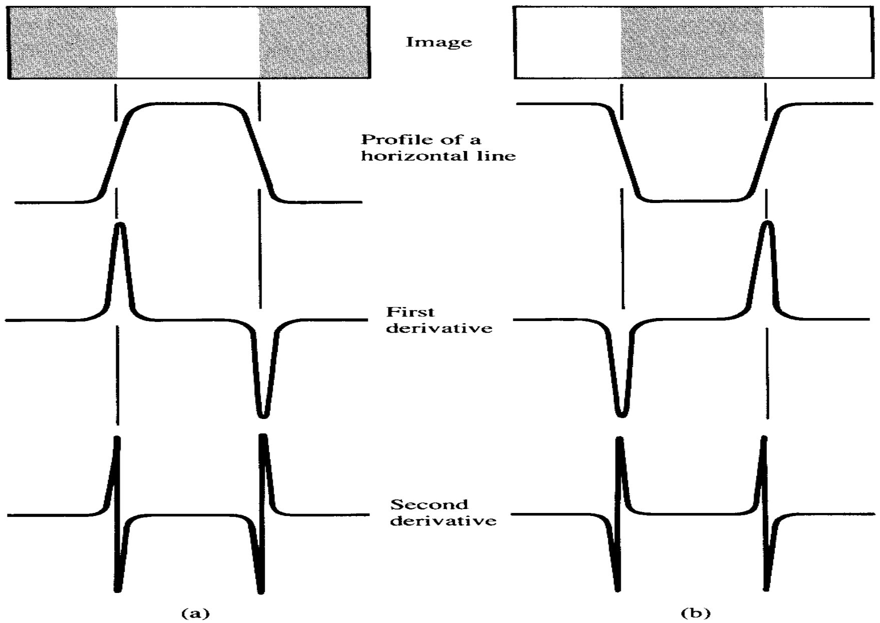


Fig 1. Edge detection by derivative operators: (a) light stripe on a dark background; (b) dark stripe on a light background.

(a) Gradient operator

- For a function *f(x,y)*, the gradient of *f* at coordinates (x',y') is defined as the vector

$$\nabla f(x',y') = \left.\begin{bmatrix} \dfrac{\partial f}{\partial x} \\ \dfrac{\partial f}{\partial y} \end{bmatrix}\right|_{(x',y')}$$

- Magnitude of vector $\nabla f(x',y')$:

$$\left|\nabla f(x',y')\right| = \left.\left(\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2\right)^{1/2}\right|_{(x',y')}$$

- Direction of the vector $\nabla f(x',y')$:

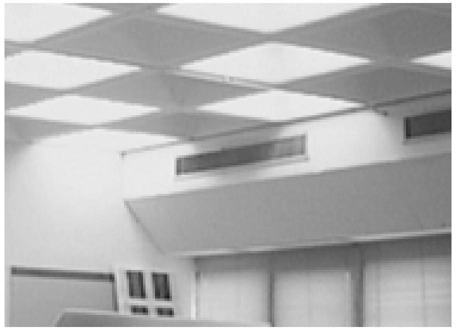
$$\alpha(x',y') = \tan^{-1}\left.\left(\frac{\partial f}{\partial y}\Big/\frac{\partial f}{\partial x}\right)\right|_{(x',y')}$$

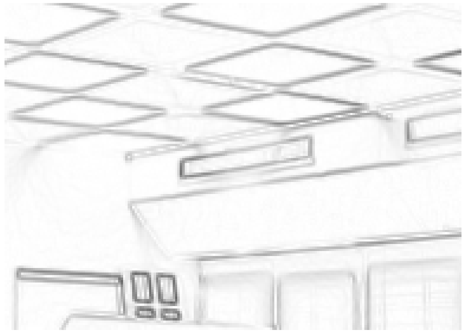
- Its magnitude can be approximated in the digital domain in a number of ways, which result in a number of operators such as Roberts, Prewitt and Sobel operators for computing its value.

Sobel operator:

- It provides both a differentiating and a smoothing effect, which is particularly attractive as derivatives typically enhance noise.

$$G_x : \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \qquad G_y : \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$


Original


Processed image

(b) Laplacian Operator

- The Laplacian of a 2D function f(x,y) is a 2nd-order derivative defined as

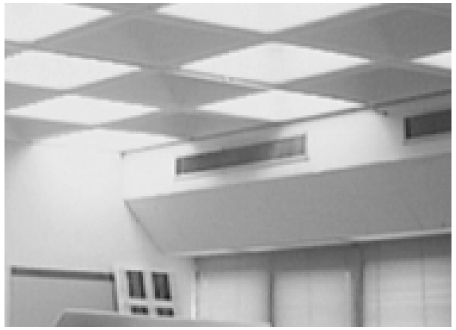
$$\nabla^2 f(x', y') = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}\bigg|_{(x', y')}$$

- The Laplacian has the same properties in all directions and is therefore invariant to rotation in the image.

- It can also be implemented in digital form in various ways.

- For a 3x3 region, the mask is given as

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$



Original

Processed image

- It is seldom used in practice for edge detection for the following reasons:
  1. As a 2nd-order derivative, it is unacceptably sensitive to noise.
  2. It produces double edges and is unable to detect edge direction.

- The Laplacian usually plays the secondary role of detector for establishing whether a pixel is on the dark or light side of an edge.

2.4 *Combined Detection:*

- Detection of combinations of points, lines and edges can be achieved by using sets of orthogonal masks.

- A set of 9 3X3 masks were proposed by Frei and Chen (1977)

Basis of edge subspace :

$$W_1=\frac{1}{2\sqrt{2}}\begin{bmatrix} 1 & \sqrt{2} & 1 \\ 0 & 0 & 0 \\ -1 & -\sqrt{2} & -1 \end{bmatrix} \quad W_2=\frac{1}{2\sqrt{2}}\begin{bmatrix} 1 & 0 & -1 \\ \sqrt{2} & 0 & -\sqrt{2} \\ 1 & 0 & -1 \end{bmatrix}$$

$$W_3=\frac{1}{2\sqrt{2}}\begin{bmatrix} 0 & -1 & \sqrt{2} \\ 1 & 0 & -1 \\ -\sqrt{2} & 1 & 0 \end{bmatrix} \quad W_4=\frac{1}{2\sqrt{2}}\begin{bmatrix} \sqrt{2} & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & -\sqrt{2} \end{bmatrix}$$

Basis of line subspace :

$$W_5=\frac{1}{2}\begin{bmatrix} -1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & -1 \end{bmatrix} \quad W_6=\frac{1}{2}\begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$W_7=\frac{1}{6}\begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix} \quad W_8=\frac{1}{6}\begin{bmatrix} -2 & 1 & -2 \\ 1 & 4 & 1 \\ -2 & 1 & -2 \end{bmatrix}$$

"Average" subspace : $W_9=\frac{1}{3}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$

- Given a 3x3 region represented by {f(i,j)|-2<i,j<2}, we have

$$R_m = \sum_{i=-1}^{1} \sum_{j=-1}^{1} f(i,j)\, w_m(i,j)$$

$$P_{line} = \left(\sum_{m=5}^{8} R_m^{\ 2} \right)^{1/2}$$

$$P_{edge} = \left(\sum_{m=1}^{4} R_m^{\ 2} \right)^{1/2}$$

$$P_{average} = \left| R_9 \right|$$

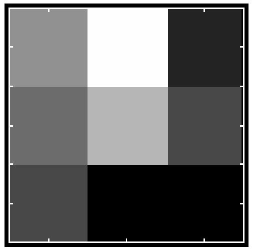
where P_{line}, $P_{average}$ and P_{edge} are the magnitudes of the projections onto edge, line and average subspaces respectively, which tell how likely it is associated with either an edge, a line or nothing.

- Example:

What's the attribute of the center of $\begin{bmatrix} 4 & 7 & 1 \\ 3 & 5 & 2 \\ 2 & 0 & 0 \end{bmatrix}$?

R_1 = 4.5607 R_2 = 2.2678
R_3 = -2.6213 R_4 = -0.8284
R_5 = -0.5000 R_6 = 1.0000
R_7 = 0.5000 R_8 = 3.0000
R_9 = 8.0000

P_{edge} = 5.7879
P_{line} = 3.2404
P_{aver} = 8.0000



Conclusion: It's likely not to be an edge or a line.

2.5 Edge linking and boundary detection

- The techniques of detecting intensity discontinuities yield pixels lying only on the boundary between regions.
- In practice, this set of pixels seldom characterizes a boundary completely because if noise, breaks in boundary from nonuniform illumination, and other effects that introduce spurious intensity discontinuities.
- Edge detection algorithms are typically followed by linking and other boundary detection procedures designed to assemble edge pixels into meaningful boundaries.

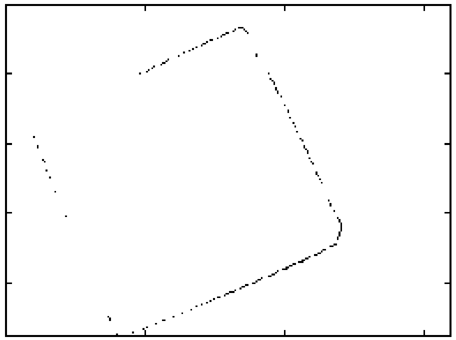
(a) *Local processing*

- Two principal properties used for establishing similarity of edge pixels in this kind of analysis are:
  1. The strength of the response of the gradient operator used to produce the edge pixel.
  2. The direction of the gradient.
- In a small neighborhood, e.g. 3x3, 5x5, all points with common properties are linked:

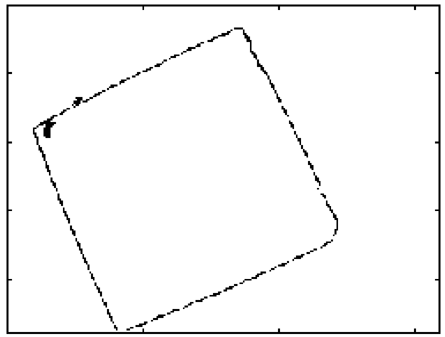
- A point (x',y') in the neighborhood of (x,y) is linked to the pixel at (x,y) if both the following magnitude and direction criteria are satisfied.

$$|\nabla f(x', y') - \nabla f(x, y)| \le Threshold\ \ T_m$$
$$|\alpha(x', y') - \alpha(x, y)| \le Threshold\ \ T_d$$

(a)



(b)



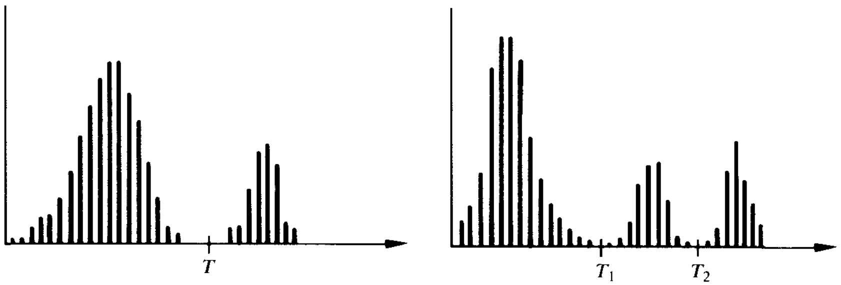
(c)

Fig 2. (a) Original image; (b) detection result without local processing; (c) detction result with local processing. (Tm = 0.15 x max(|∇f|) and Td = pi/9)

3. *Thresholding*

- Thresholding is one of the most important approaches to image segmentation.

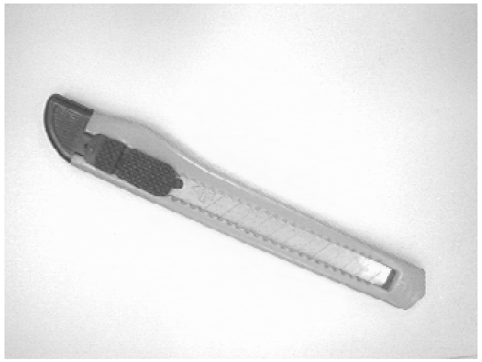
- If background and object pixels have gray levels grouped into 2 dominant modes, they can be separated with a threshold easily.



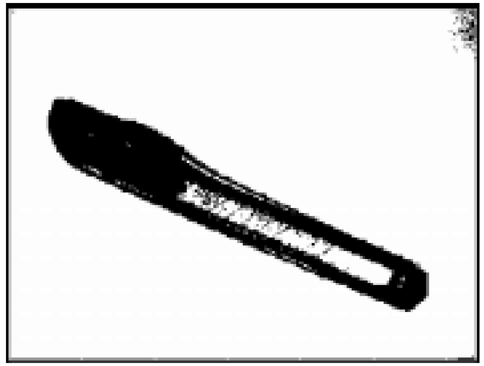
- Thresholding may be viewed as an operation that involves tests against a function T of the form T=T[x,y,p(x,y),f(x,y)], where f(x,y) is the gray level of point (x,y), and p(x,y) denotes some local property of this point such as the average gray level of a neighborhood centered on (x,y).

- Special cases:
  If T depends on
  1. f(x,y) only - global threshold
  2. Both f(x,y) & p(x,y) - local threshold
  3. (x,y) - dynamic threshold

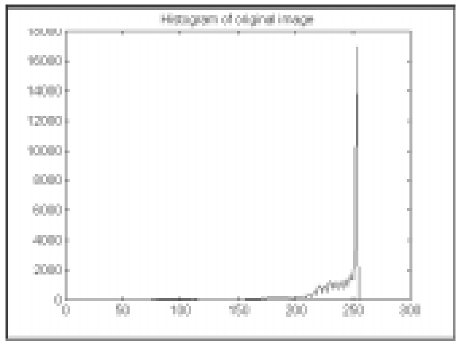
- Multilevel thresholding is in general less reliable as it is difficult to establish effective thresholds to isolate the regions of interest.



Original



Threshold result (T=200)



Histogram

Fig 3. Nonadaptive thresholding result

3.1 Adaptive thresholding

- The threshold value varies over the image as a function of local image characteristics.
- Image f is divided into subimages.
- A threshold is determined independently in each subimage.
- If a threshold can't be determined in a subimage, it can be interpolated with thresholds obtained in neighboring subimages.
- Each subimage is then processed with respect to its local threshold.

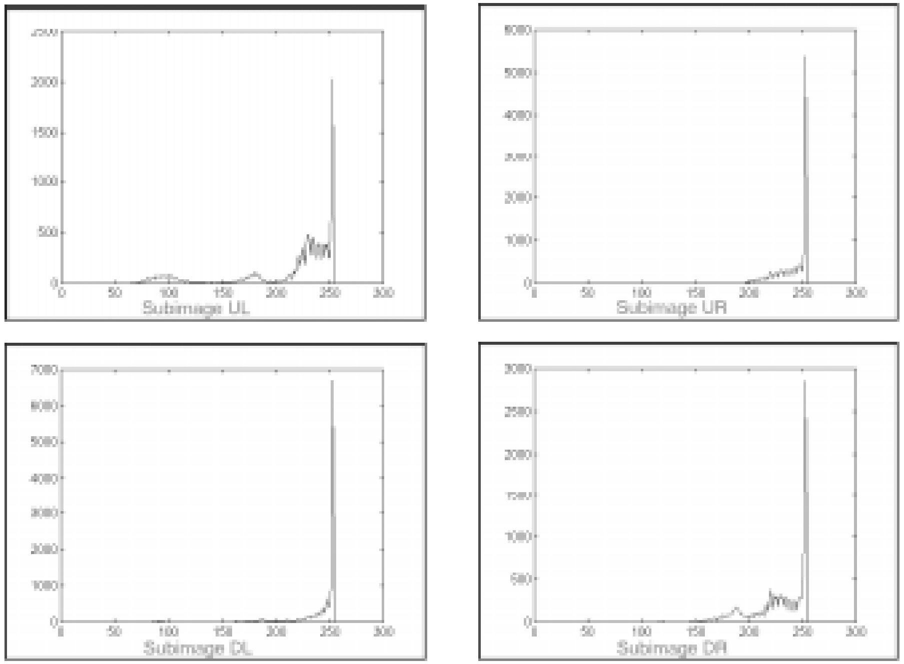


Fig 4. Histogram of the 4 subimages

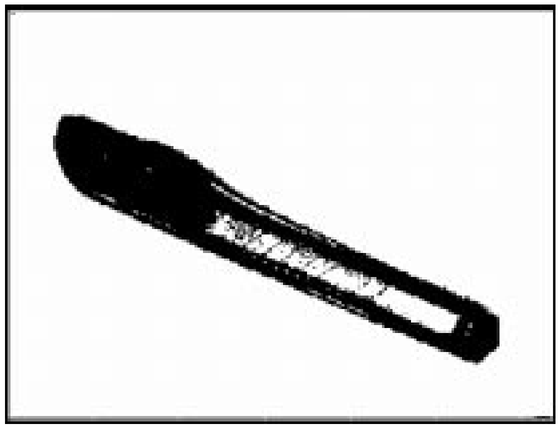


Fig 5. Adaptive thresholding result (T_{UL}=200,T_{UR}=100, T_{DL}=200,T_{DR}=200)

3.2 Threshold selection based on boundary characteristics

- A reliable threshold must be selected to identify the mode peaks of a given histogram.
- This capability is very important for automatic threshold selection in situations where image characteristics can change over a broad range of intensity distributions.
- We can consider only those pixels that lie on or near the boundary between objects and the background such that the associated histogram is well-shaped to provide a good chance for us to select a good threshold.
- The gradient can indicate if a pixel is on an edge or not.
- The Laplacian can tell if a given pixel lies on the dark or light (background or object) side of an edge.
- The gradient and laplacian can produce a 3-level image

$$s(x,y)=\begin{cases} 0 & \text{if } G[f(x,y)]<T \\ 1 \text{ if } G[f(x,y)]\geq T & \text{and } L[f(x,y)]\geq 0 \\ -1 \text{ if } G[f(x,y)]\geq T & \text{and } L[f(x,y)]<0 \end{cases}$$

where T is a threshold.

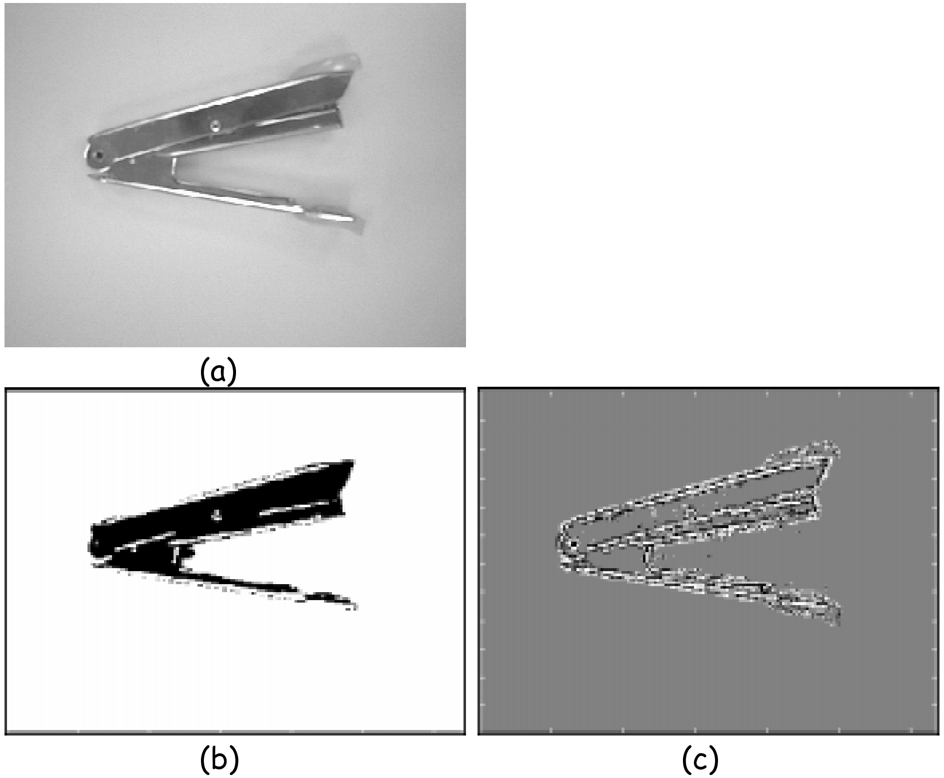


Fig 6. (a) Original, (b) processed result without using boundary characteristic and (c) processed result with using boundary characteristic.

4. *Region-oriented segmentation*

- In previous methods, we partition an image into regions by finding boundaries between regions based on intensity discontinuities.
- Here, segmentation is accomplished via thresholds based on the distribution of pixel properties, such as intensity or color.
- Basic formulation:
  Let R represents the entire image which is partitioned into subregions R_1, R_2,...R_n such that
  - $\bigcup_{i=1}^{n} R_i = R$
  - R_i is a connected region, i=1,2...n
  - $R_i \cap R_j = \{\}$ for all $i \neq j$,
  - $P(R_i)$=true for i=1,2..n
  - $P(R_i \cup R_j)$=false for $i \neq j$

  where $P(R_i)$ is a logical predicate over the points in set R_i.

- Physical meaning of the formulation:
  - The segmentation must be complete, i.e. every point must be in a region.
  - Points in a region must be connected.

  - The regions must be disjoint.
  - It deals with the properties that must be satisfied by the pixels in a segmented region - for example $P(R_i)$=true if all pixels in R_i have the same intensity.
  - Regions R_i and R_j are different in the sense of predicate P.

4.1 Region growing by pixel aggregation

- Region growing is a procedure that groups pixels or subregions into larger regions.
- Pixel aggregation starts with a set of "seed" points from those grows by appending to each seed point those neighboring pixels that have similar properties such as gray level, texture and color.

0	0	5	6	7
1	1	5	8	7
0	1	6	7	7
2	0	7	6	6
0	1	5	6	5

Original intensity array

a	a	b	b	b
a	a	b	b	b
a	a	b	b	b
a	a	b	b	b
a	a	b	b	b

Result of threshold=3

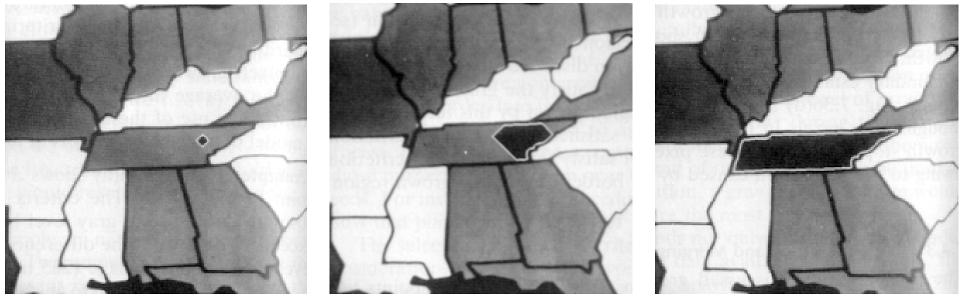
a	a	a	b	b
a	a	a	b	b
a	a	b	b	b
a	a	b	b	b
a	a	a	b	?

Result of Threshold=5.5

a	a	a	a	a
a	a	a	a	a
a	a	a	a	a
a	a	a	a	a
a	a	a	a	a

Result of threshold=9

Example of region growing using known starting points



(a) (b) (c)

Fig 7. Original image with seed point, (b) early stage of region growth, (c) final region.

- Problems have to be resolved:
  1. Selection of initial seeds that properly represent regions of interest.
  2. Selection of suitable properties for including points in the various regions during the growing process.
  3. The formulation of stopping rule.

4.2 Region splitting and merging

- To subdivide an image initially into a set of arbitrary, disjointed regions and then merge and/or split the regions in an attempt to satisfy the conditions stated above.
- A split and merge algorithm is summarized by the following procedure in which, at each step, we:
  (1) split into 4 disjointed quadrants any regions R_i where $P(R_i)$=false;
  (2) merge any adjacent regions R_j and R_k for which $P(R_i \cup R_j)$=true; and
  (3) stop when no further merging or splitting is possible.

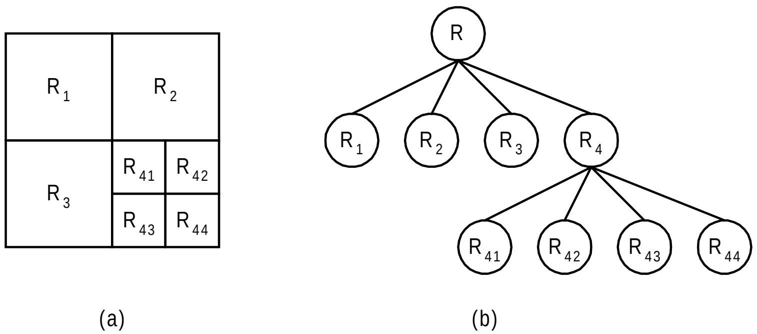
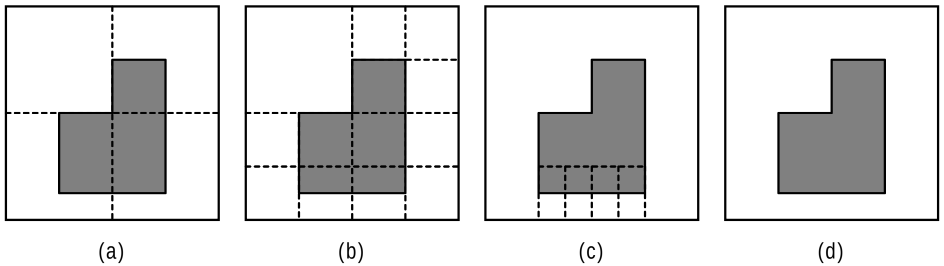


Fig 8. Partitioned image and corresponding quadtree

- Example:

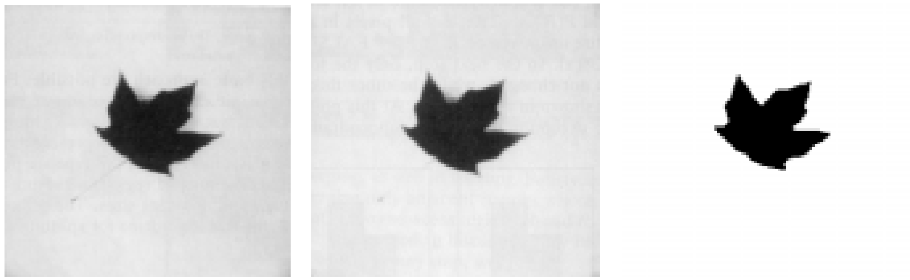


Example of split-and-merge algorithm

(a) The entire image is split into 4 quadrants.

(b) Only the top left region satisfies the predicate so it is not changed, while the other 3 quadrants are split into subquadrants.

(c) At this point several regions can be merged, with the exception of the 2 subquadrants that include the lower part of the object; these do not satisfy the predicate and must be split further.



(a) (b) (c)

Fig 9. (a) Original image, (b) Result of split and merge algorithm; (c) Result of thresholding (b).

- Image segmentation is a preliminary step in most automatic pictorial pattern-recognition and scene-analysis problems.
- The choice of one segmentation technique over another is dicated mostly by the peculiar characteristics of the problem being considered.