THE UNIVERSITY of TEXAS
HEALTH SCIENCE CENTER AT HOUSTON
SCHOOL of HEALTH INFORMATION SCIENCES

Image Operations II

For students of HI 5323
“Image Processing”

Willy Wriggers, Ph.D.
School of Health Information Sciences

http://biomachina.org/courses/processing/04.html

Edge-based Segmentation

- Attempt to find the edges directly by their high gradient magnitude
- Edge-based segmentations rely on edges found in an image by edge detecting operators
- These edges mark image locations of discontinuities in gray level, color, texture, etc

What is an Edge?

- Edges are those places in an image that correspond to object boundaries
- Edges are pixels where image brightness changes abruptly
- An edge is a property attached to an individual pixel and is calculated from the image function behavior in a neighborhood of the pixel
- It is a **vector variable** (**magnitude** of the gradient, **direction** of an edge**)**

Gradient

- The gradient is a vector, whose components measure how rapidly pixel values are changing with distance in the *x* and *y* directions
- The components of the gradient may be found using the following approximation

$$\frac{\partial f(x,y)}{\partial x} = \triangle x = \frac{f(x+d_x, y) - f(x,y)}{d_x}$$

$$\frac{\partial f(x,y)}{\partial y} = \triangle y = \frac{f(x, y+d_y) - f(x,y)}{d_y}$$

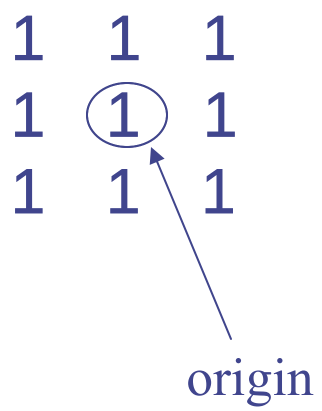
Where d_x and d_y measure distance along the *x* and *y* directions respectively

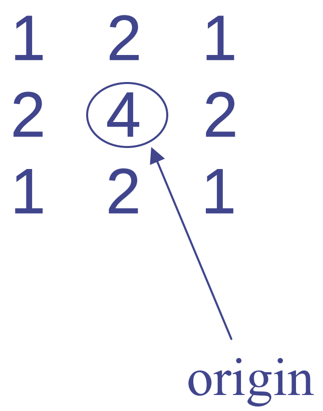
Magnitude and Direction

- For a continuous image *f(x,y)* , where *x* and *y* are the row and column coordinates respectively, we typically consider the two directional derivatives $\partial x\ f(x,y)$ and $\partial y\ f(x,y)$
- Two functions that can be expressed these directional derivatives
  - Gradient magnitude $$\left|\nabla f(x,y)\right| = \sqrt{(\partial_x f(x,y))^2 + (\partial_y f(x,y))^2}$$
  - Gradient orientation $$= \mathrm{ArcTan}\left(\partial_y f(x,y) / \partial_x f(x,y)\right)$$

Mask

- A mask is a small matrix whose value is called *weights*
- Each mask has an *origin*, which is usually one of its positions
- The origins of symmetric mask are usually its center pixel position
- For non-symmetric masks, any pixel position can be choose as a origins





Mask Operations

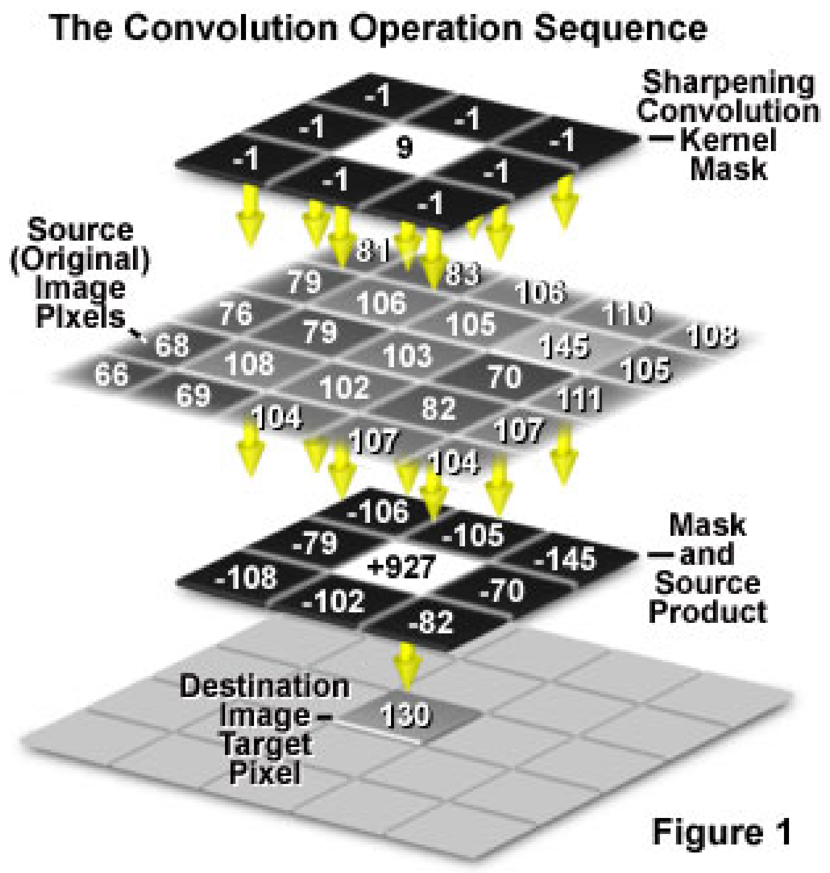
- A pixel value is computed from its old value and the value of the pixels of its neighborhoods
- More costly operations than simple point operations, but more powerful
- Applications:
  - Convolution
  - Cross correlation
  - Non-liner filtering (erosion, dilation)

Edge Detection Methods

- Many are implemented with ***convolution mask*** and based on discrete approximations to differential operators
- Differential operations measure the rate of change in the image brightness function
- Some operators return orientation information. Other only return information about the existence of an edge at each point

Convolution Operation

- Convolution kernels typically feature an odd number of rows and columns in the form of a square
  - a 3 x 3 pixel mask (convolution kernel) being the most common form



©micro.magnet.fsu.edu/primer/java/digitalimaging/processing/kernelmaskoperation

Convolution Operation (cont.)

- The convolution operation is performed individually on each pixel of the original **input** image, and involves three sequential operations
  - First, The operation begins when the convolution kernel is overlaid on the original image in such a manner that the center pixel of the mask is matched with the single pixel location to be convolved from the input image
  - Second, each pixel integer value in the original image is multiplied by the corresponding value in the overlying mask
  - Third, the sum of products from the second step is computed
  - Finally, the gray level value of the target pixel is replaced by the sum of all the products determined in the third step

  To perform a convolution on an entire image, this operation must be repeated for each pixel in the original image

Edge Detection Operators

- Most edge detection opreators are based in the way on measuring the intensity gradient at a point in the image
  - The Roberts Edge Operator
  - The Sobel Edge Operator
  - The Prewitt Edge Operator
  - The Kirsch Edge Operator
  - Laplacian Edge Operator

Roberts Edge Operators

- The operator consists of a pair of 2×2 convolution mask, One mask is simply the other rotated by 90°
- One local differential operator
- Mark edge point only
- No information about edge orientation
- Work best with binary images
- Primary disadvantage:
  - High sensitivity to noise
  - Few pixels are used to approximate the gradient

+1	0
0	-1

Gx

0	+1
-1	0

Gy

Sobel Edge Operators

- The operator consists of a pair of 3×3 convolution masks, one mask is simply the other rotated by 90°
- An advantage of using a larger mask size is that errors due to the effects of noise are reduced by local averaging within the neighborhood of the mask

-1	0	+1
-2	0	+2
-1	0	+1

Gx

+1	+2	+1
0	0	0
-1	-2	-1

Gy

Edge Magnitude = $\sqrt{Gx^2 + Gy^2}$ Edge Direction = $\mathrm{ArcTan}\,(Gy / Gx)$

Prewitt Edge Operators

- Similar to the Sobel, with slightly different mask coefficients

-1	0	+1
-1	0	+1
-1	0	+1

Gx

-1	-1	1
0	0	0
+1	+1	+1

Gy

Edge Magnitude = $\sqrt{Gx^2 + Gy^2}$ Edge Direction = $\mathrm{ArcTan}\ (Gy \,/\, Gx)$

Kirsch Edge Operators

- Taking a single mask and rotating it to 8 major compass orientations: N, NW, W, SW, S, SE, E, and NE
- The edge magnitude = The maximum value found by the convolution of each mask with the image
- The edge direction is defined by the mask that produces the maximum magnitude

$$N=\begin{array}{|c|c|c|}\hline +5 & +5 & +5 \\ \hline -3 & 0 & -3 \\ \hline -3 & -3 & -3 \\ \hline \end{array} \quad NE=\begin{array}{|c|c|c|}\hline -3 & +5 & +5 \\ \hline -3 & 0 & +5 \\ \hline -3 & -3 & -3 \\ \hline \end{array} \quad E=\begin{array}{|c|c|c|}\hline -3 & -3 & +5 \\ \hline -3 & 0 & +5 \\ \hline -3 & -3 & +5 \\ \hline \end{array} \quad SE=\begin{array}{|c|c|c|}\hline -3 & -3 & -3 \\ \hline -3 & 0 & +5 \\ \hline -3 & +5 & +5 \\ \hline \end{array}$$

$$S=\begin{array}{|c|c|c|}\hline -3 & -3 & -3 \\ \hline -3 & 0 & -3 \\ \hline +5 & +5 & +5 \\ \hline \end{array} \quad SW=\begin{array}{|c|c|c|}\hline -3 & -3 & -3 \\ \hline +5 & 0 & -3 \\ \hline +5 & +5 & -3 \\ \hline \end{array} \quad W=\begin{array}{|c|c|c|}\hline +5 & -3 & -3 \\ \hline +5 & 0 & -3 \\ \hline +5 & -3 & -3 \\ \hline \end{array} \quad NW=\begin{array}{|c|c|c|}\hline +5 & +5 & -3 \\ \hline +5 & 0 & -3 \\ \hline -3 & -3 & -3 \\ \hline \end{array}$$

Laplacian Edge Operator

- Edge magnitude is approximated in digital images by a convolution sum
- A very popular operator approximating the second derivative which gives the gradient magnitude only
  - Masks for 4 and 8 neighborhoods

0	-1	0
-1	4	-1
0	-1	0

-1	-1	-1
-1	8	-1
-1	-1	-1

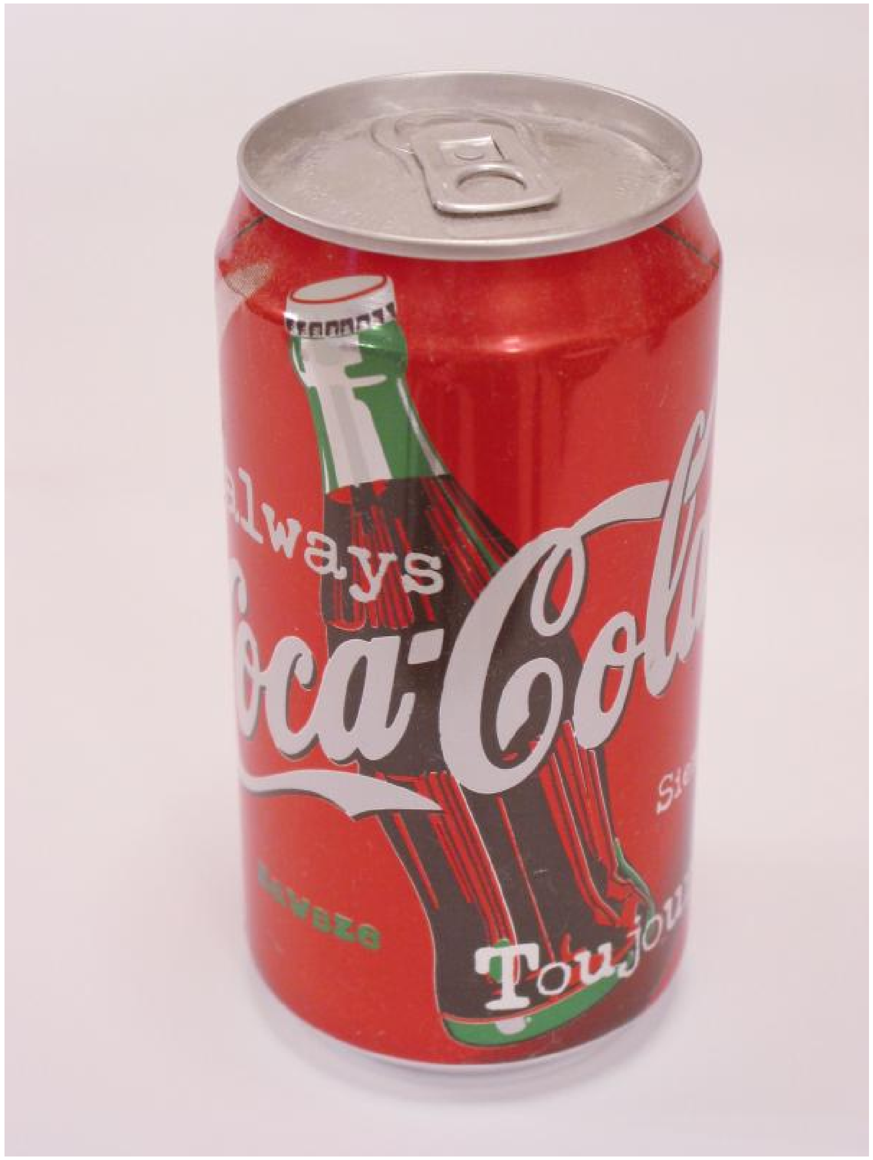
  - Mask with stressed significance of the central pixel or its neighborhood

1	-2	1
-2	4	-2
1	-2	1

-2	1	-2
1	4	1
-2	1	-2

Examples

original image

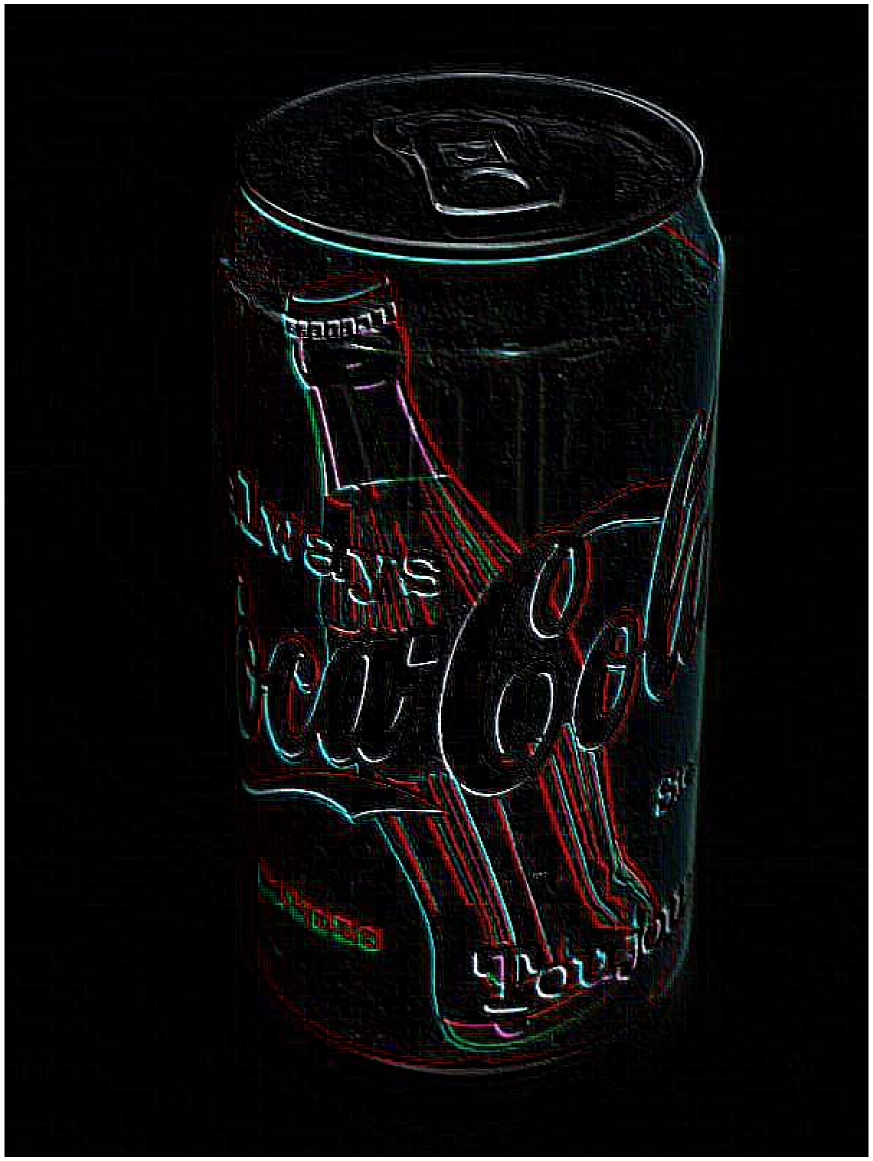


© www.pbase.com/lasm/convolution_matrix

Examples

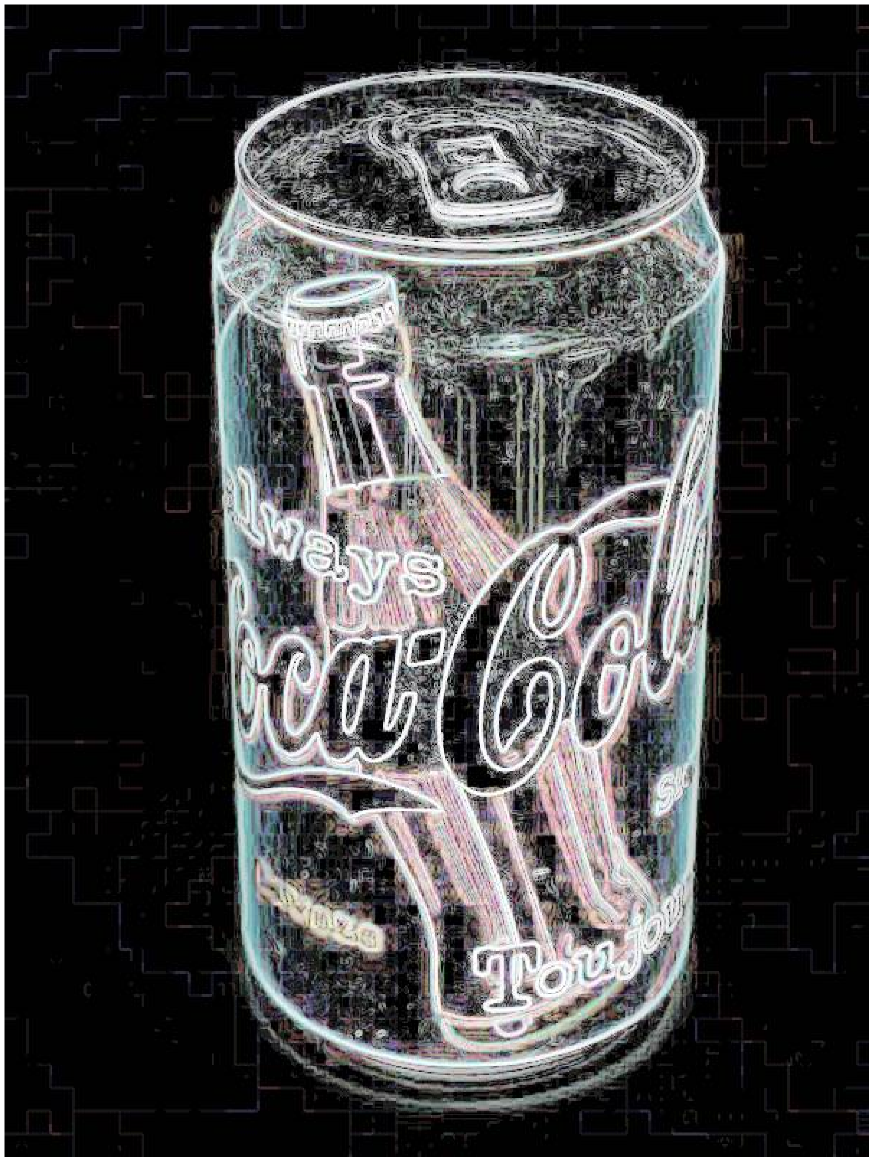
Roberts (Gy only)

Created with Adobe Photoshop



Examples

Sobel (Magnitude)

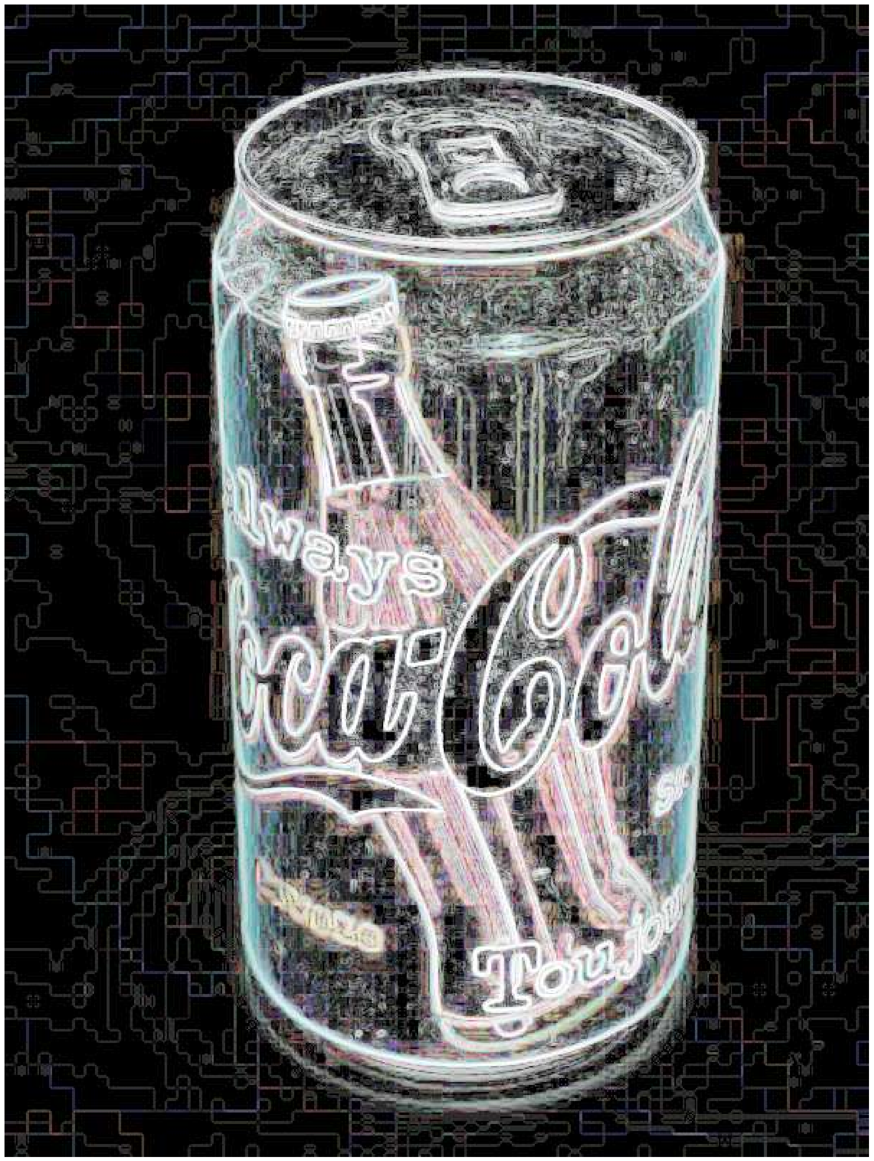


© www.pbase.com/lasm/convolution_matrix

Examples

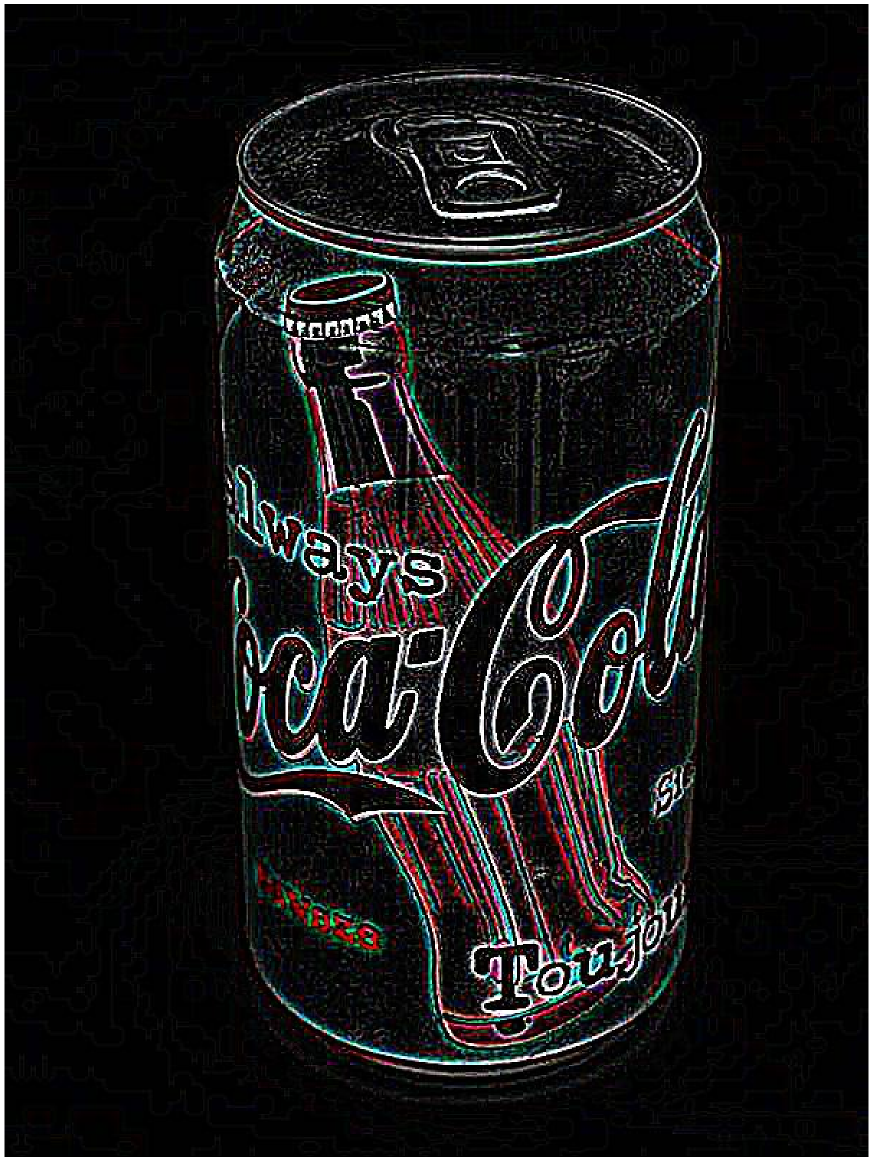
Kirsch

© www.pbase.com/lasm/convolution_matrix



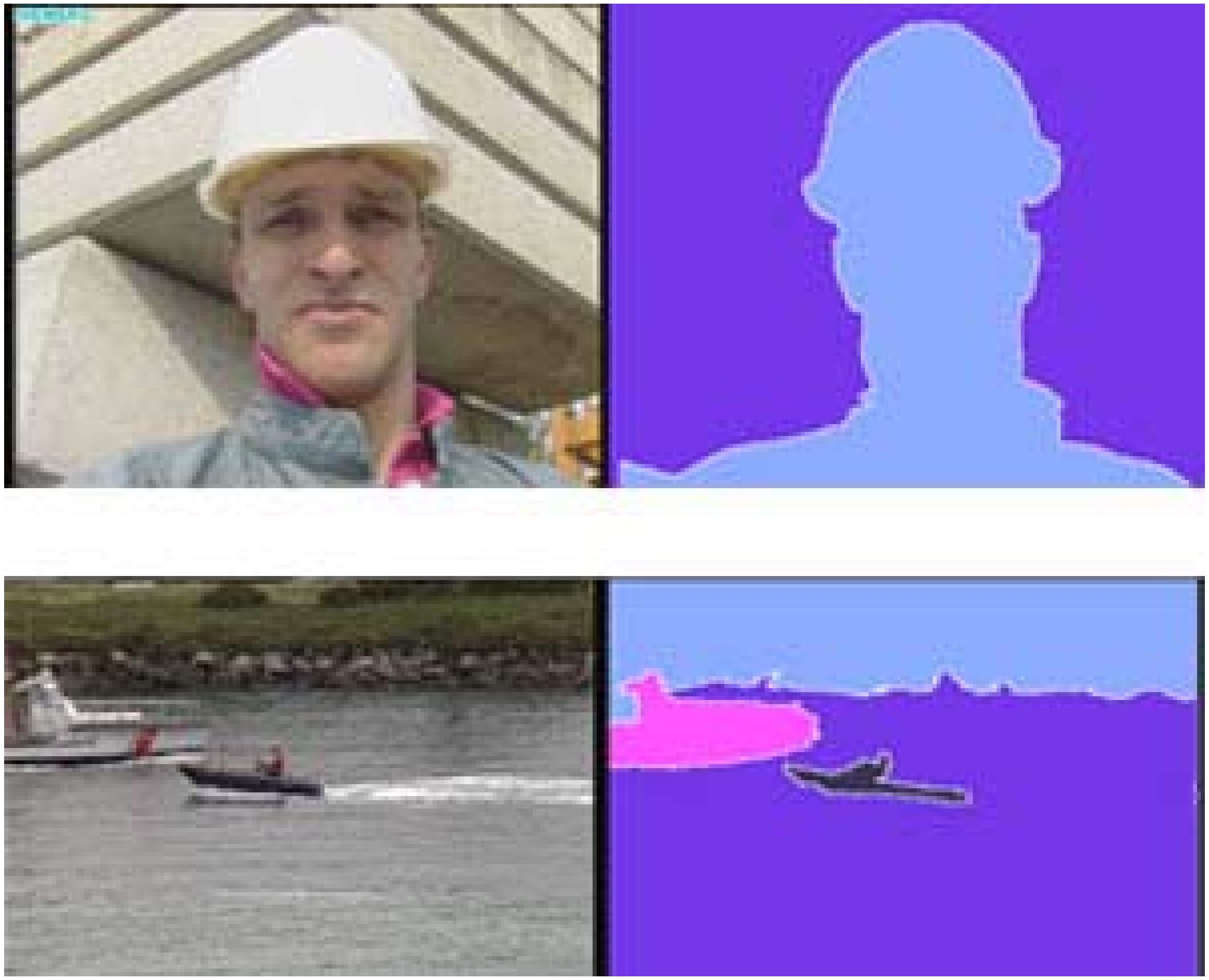
Examples

Laplacian



© www.pbase.com/lasm/convolution_matrix

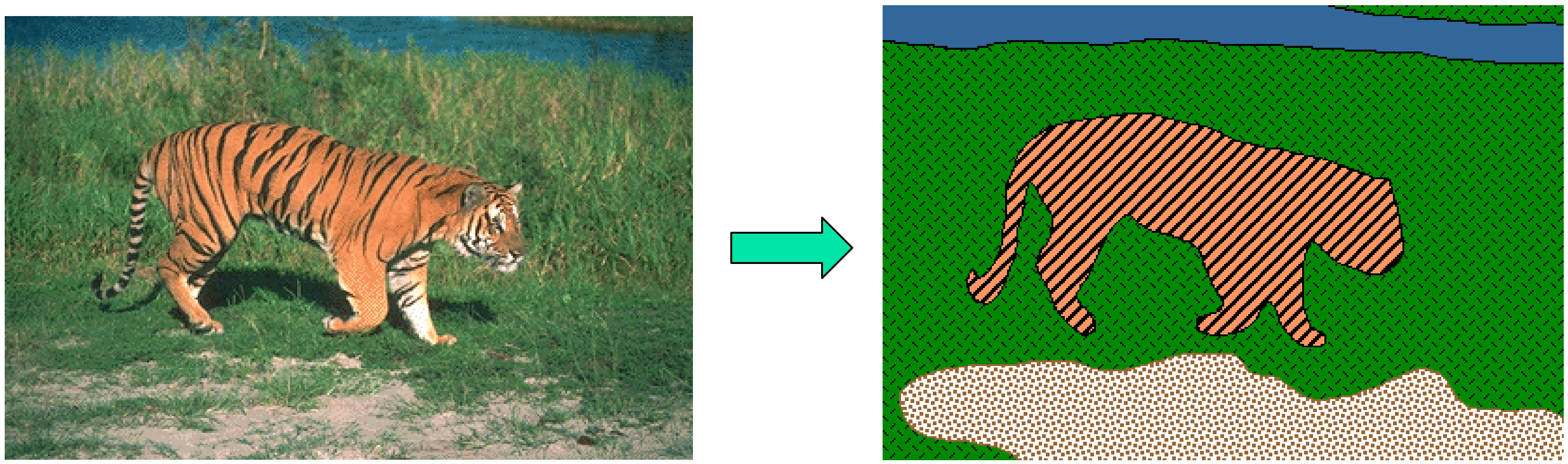
Image Segmentation



© http://ippr.csie.ntu.edu.tw:8080/image/Video_Object_Segmentation.jpg

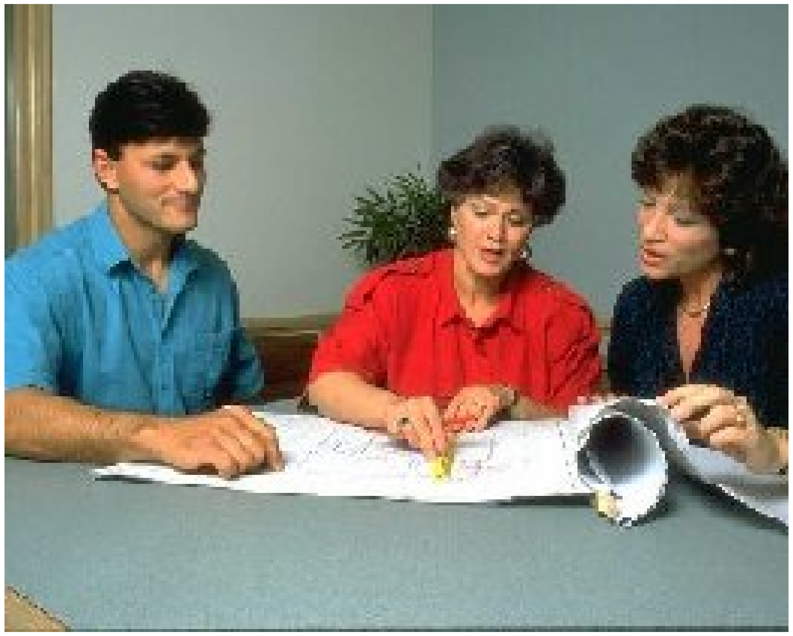
What is Segmentation?

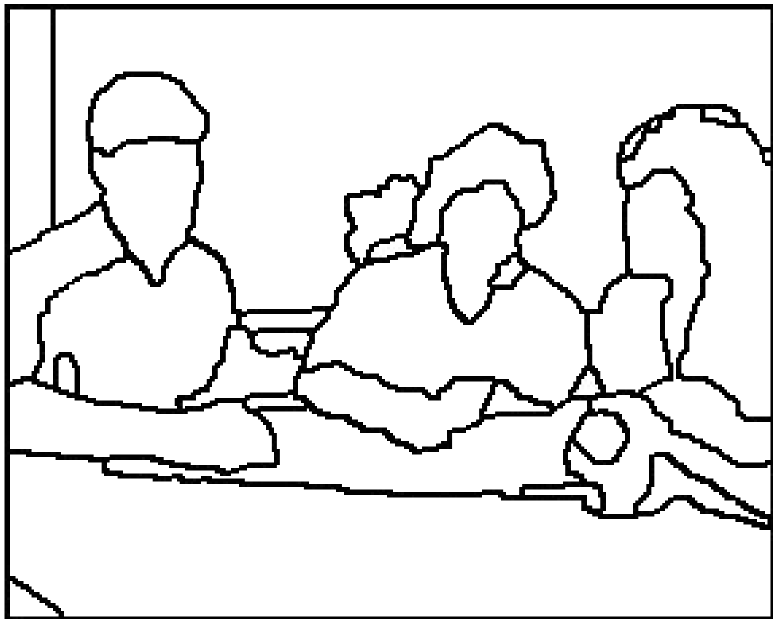
- Dividing image into its constituent regions or objects



© 2003, Ninad Thakoor, http://visionlab.uta.edu/~ninad/ other_files/Image_Segmentation.ppt

Boundary Tracking



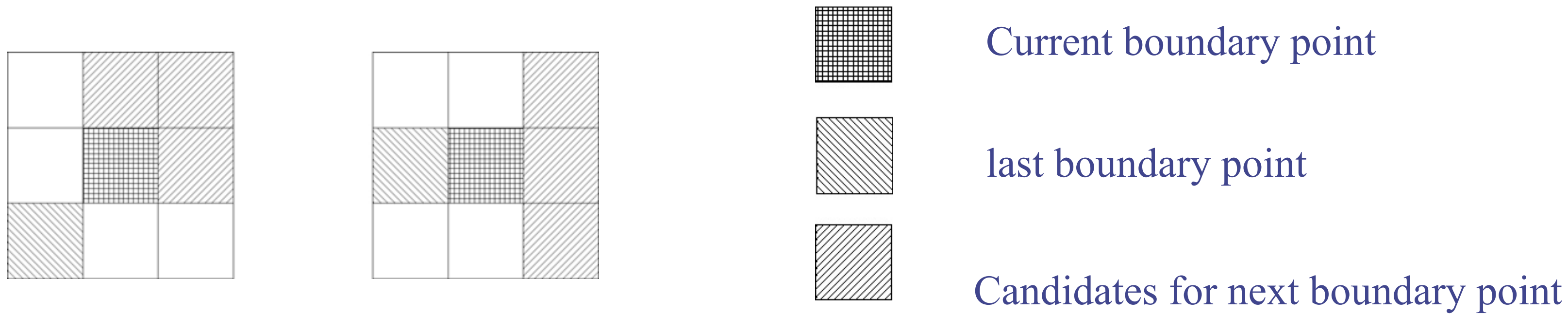


© http://www.stat.ucla.edu/~ztu/DDMCMC/color_people/results.htm

Boundary Tracking

- How it works:
  - Identify the pixel of the highest gray level as the first boundary point
  - Search the *3*x*3* neighborhood centered on the first boundary point, take the neighbor with the maximum gray level as the second boundary point
  - Finding the next boundary point iterative by give the current and previous boundary points

    To find the following border elements, edge gradient magnitudes and directions are usually computed in pixels of probable border continuation



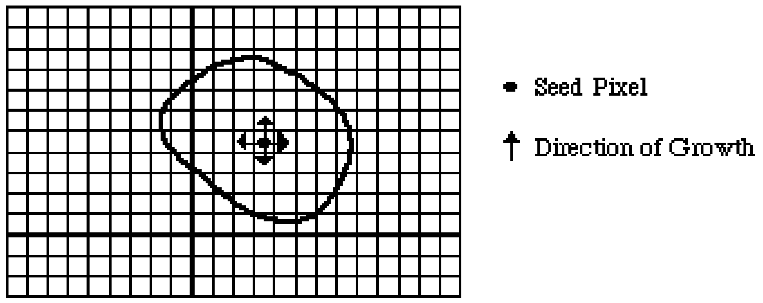
©Kenneth R. Castleman

This algorithm works well on noise free image

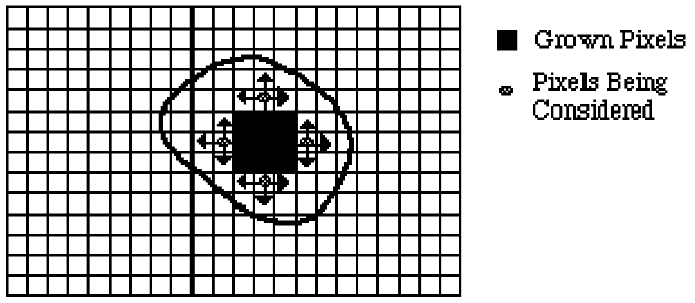
Remove the noise by smoothing the gradient image before tracking

Region Growing (Floodfill)

- Region growing - opposite of the split and merge approach
  - An initial set of small areas are iteratively merged according to similarity constraints
  - Start by choosing an arbitrary ***seed pixel*** and compare it with neighboring pixels
  - Region is ***grown*** from the seed pixel by adding in neighboring pixels that are similar, increasing the size of the region
  - When the growth of one region stops we simply choose another seed pixel which does not yet belong to any region and start again
  - This whole process is continued until all pixels belong to some region



(a) Start of Growing a Region

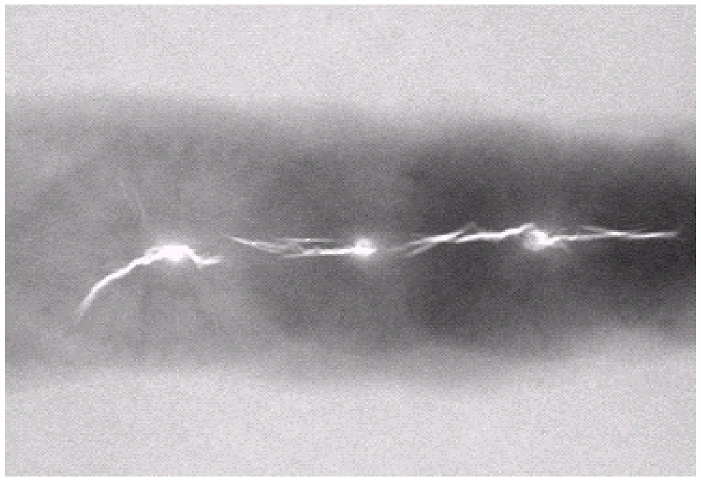


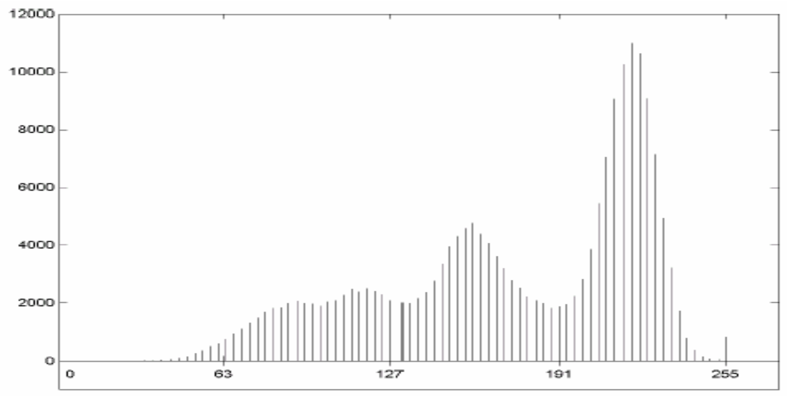
(b) Growing Process After a Few Iterations

©www.cs.cf.ac.uk/Dave/Vision_lecture/node35

Region Growing Example

- X ray of weld (the horizontal dark region) containing several cracks and porosities (the bright, white streaks running horizontally through the image)
- We need initial seed points to grow into regions
- On looking at the histogram and image, cracks are bright
- Select all pixels having value of 255 as seeds

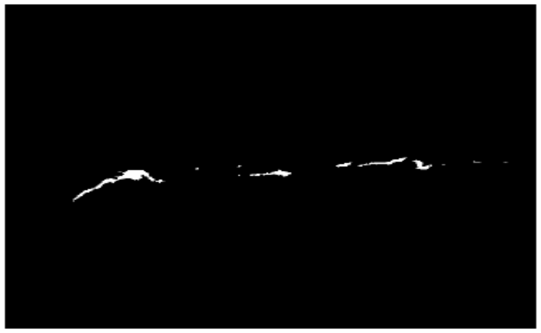


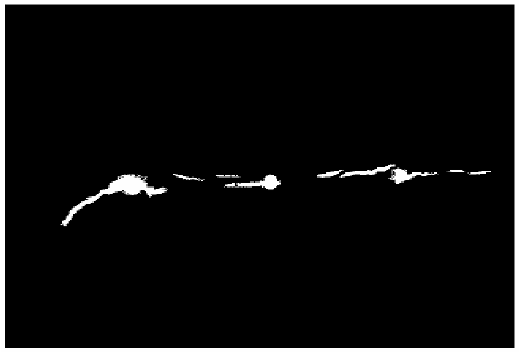


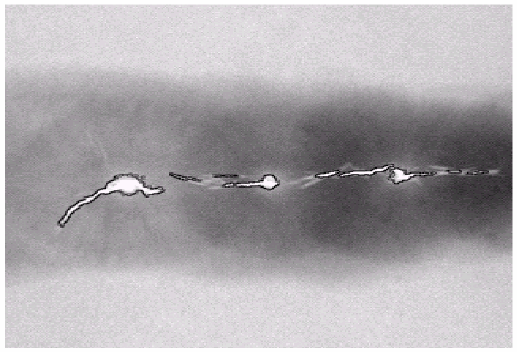
© 2003, Ninad Thakoor, http://visionlab.uta.edu/~ninad/ other_files/Image_Segmentation.ppt

Region Growing Example

- There is a valley at around 190 in the histogram
- A pixel should have a value > 190 to be considered as a part of region to the seed point
- The pixel should be a 8-connected neighbor to at least one pixel in that region





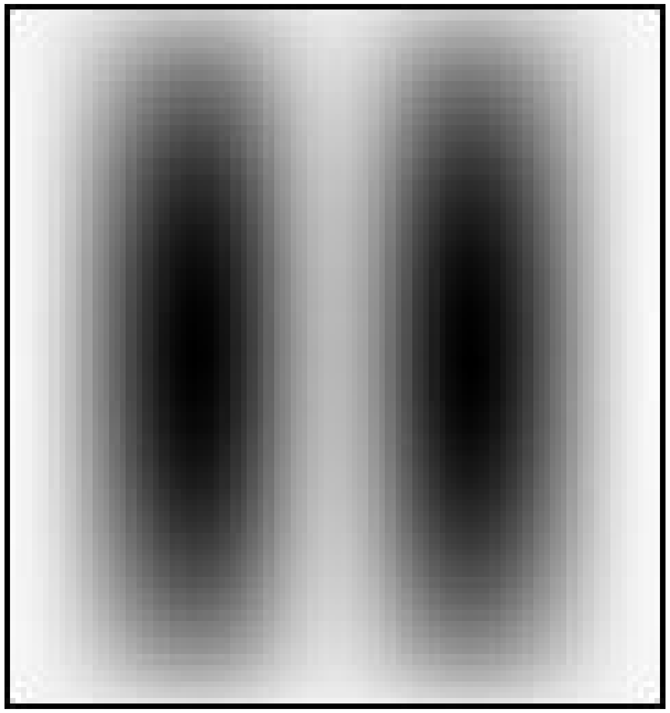


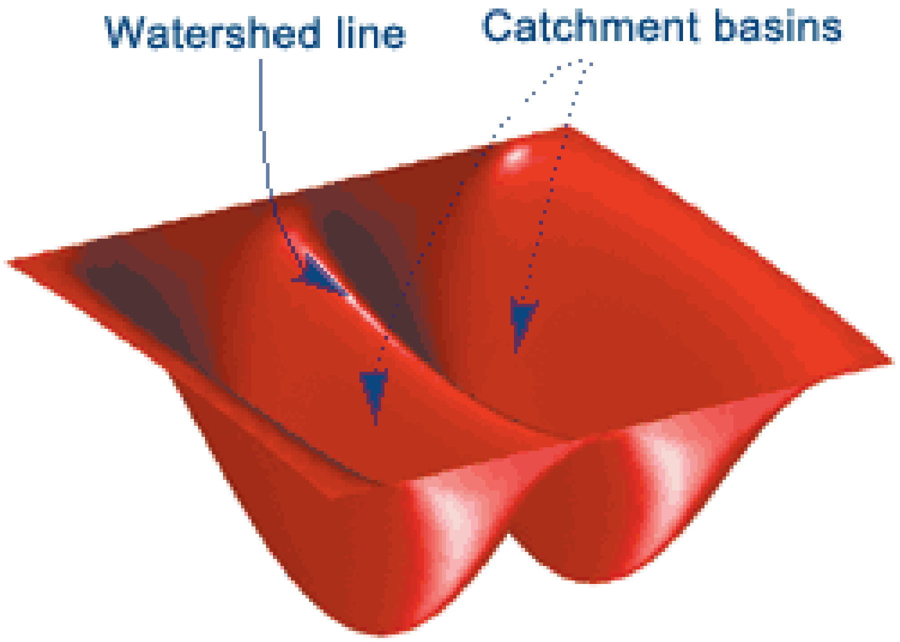
Watershed Segmentation

- Visualizes 2D image in three dimensions
- Three type of points
  - Points belonging to local minima
  - Points at which a drop of water, if placed at the location of any of these points, would fall with certainty to a single minimum
  - Points at which water will be equally likely to fall to more than one such minimum

© 2003, Ninad Thakoor, http://visionlab.uta.edu/~ninad/ other_files/Image_Segmentation.ppt

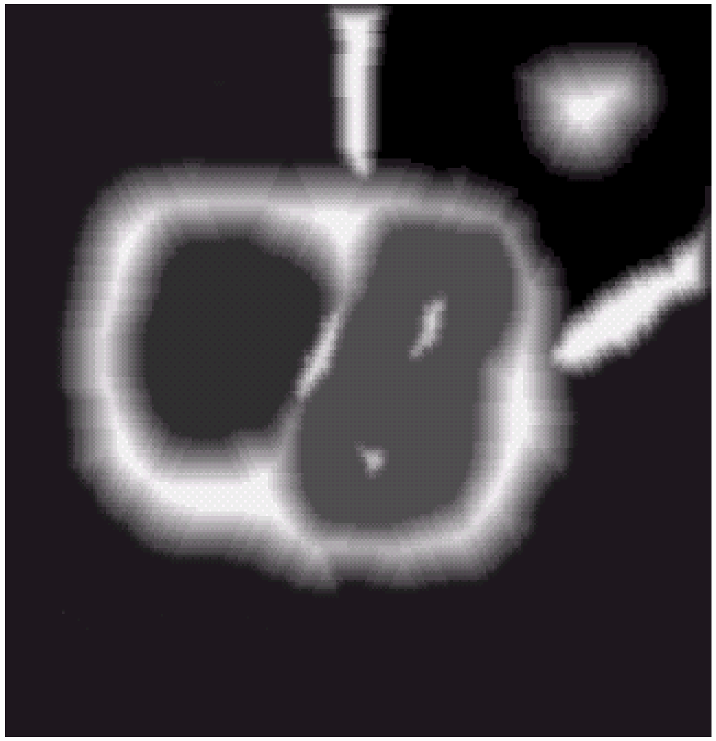
Watershed Segmentation

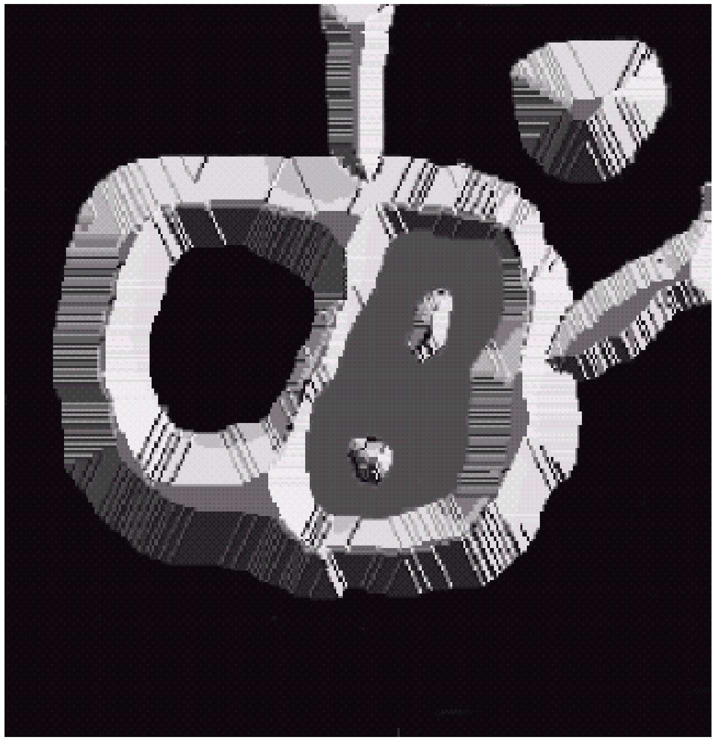




© 2003, Ninad Thakoor, http://visionlab.uta.edu/~ninad/ other_files/Image_Segmentation.ppt

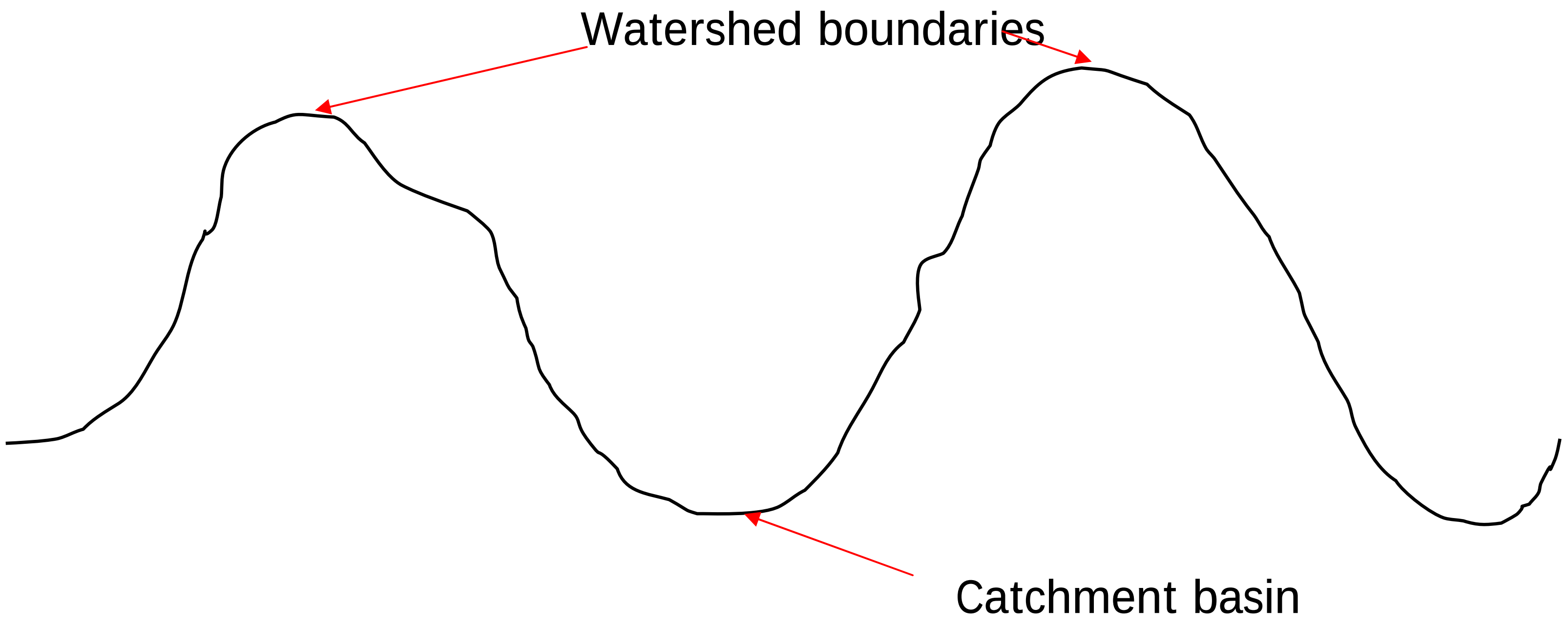
Example


Original Image


Topographic view of the image

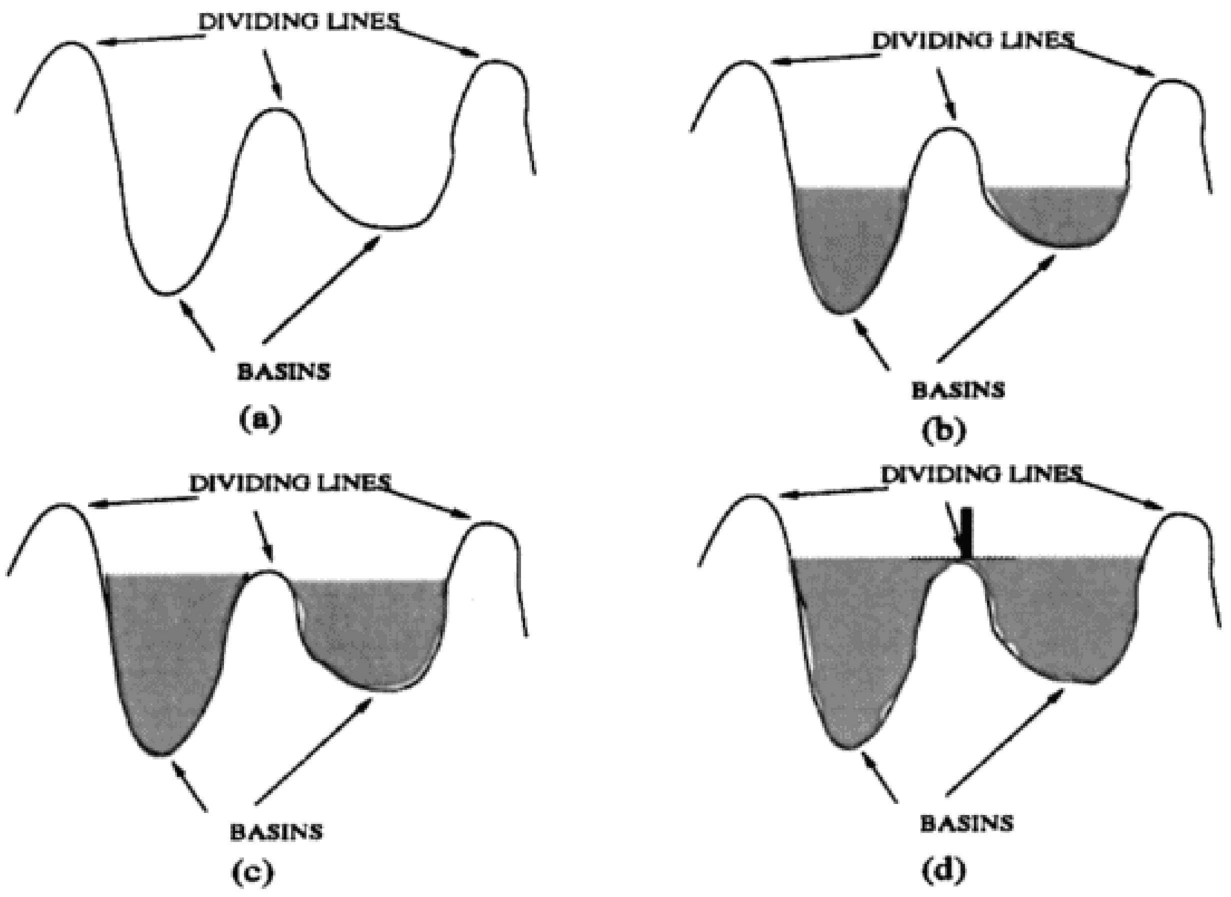
Immersion Algorithm

- To locate peaks in the image, we could poke holes in the bottom of the gradient magnitude image and lower the image into the water tub
- Basins would begins to fill – as two basins combined, we could mark these points as watershed boundaries



© 2003, Ninad Thakoor, http://visionlab.uta.edu/~ninad/ other_files/Image_Segmentation.ppt

Immersion Algorithm



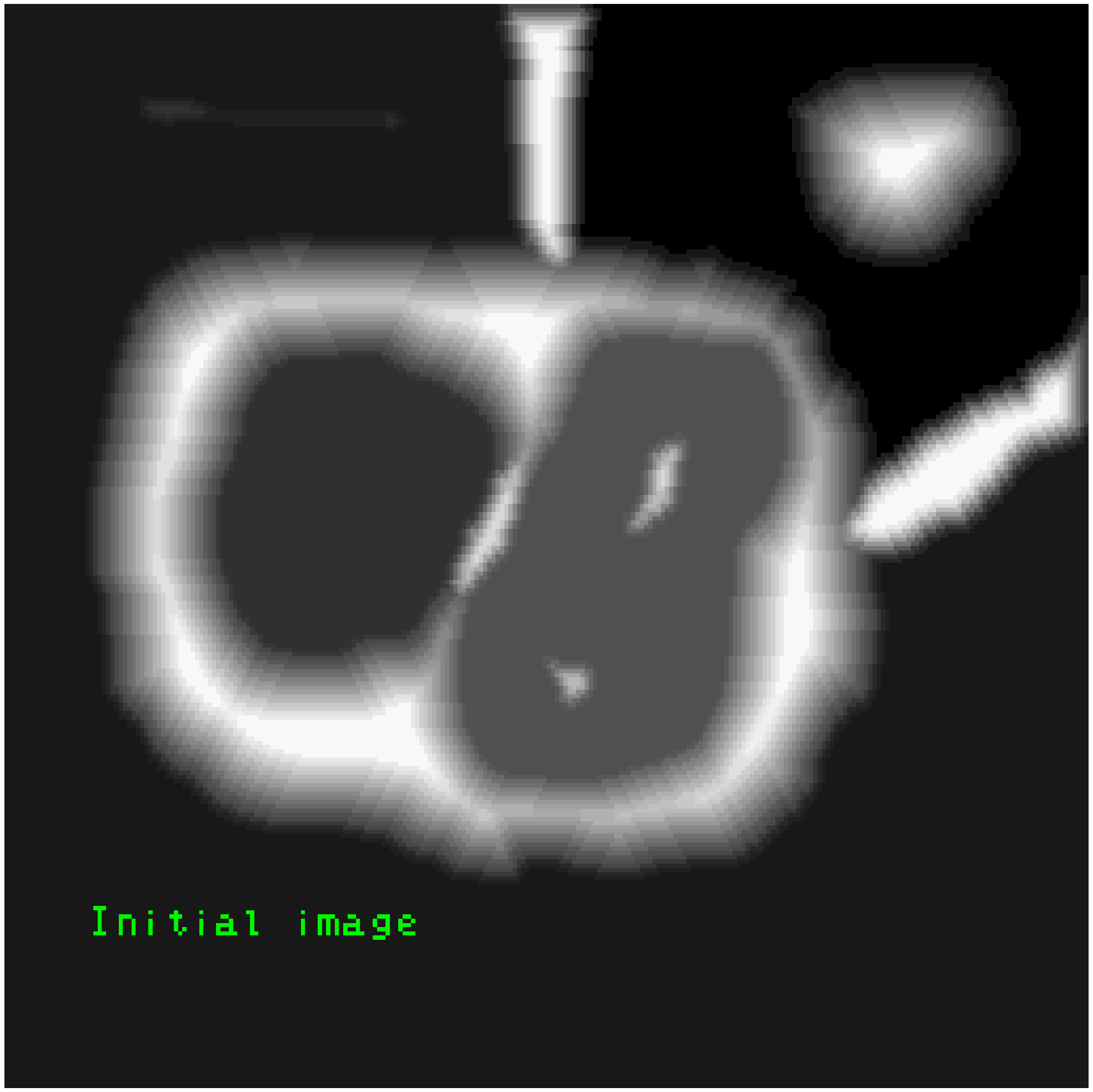
© 2003, Ninad Thakoor, http://visionlab.uta.edu/~ninad/ other_files/Image_Segmentation.ppt

Practical Algorithm

- Calculate the gradient magnitude image by evaluating the magnitude at each image point
- Start at threshold T=1
- Find pixels above threshold and pixels below or equal to T
- Let T=T+1
- If a region of pixels above T combines with a region of pixels below or equal to T, then mark as a watershed boundary
- Repeat until highest gradient magnitude is reached (T=K-1)

© 2003, Ninad Thakoor, http://visionlab.uta.edu/~ninad/ other_files/Image_Segmentation.ppt

Demo Movie of Watershed

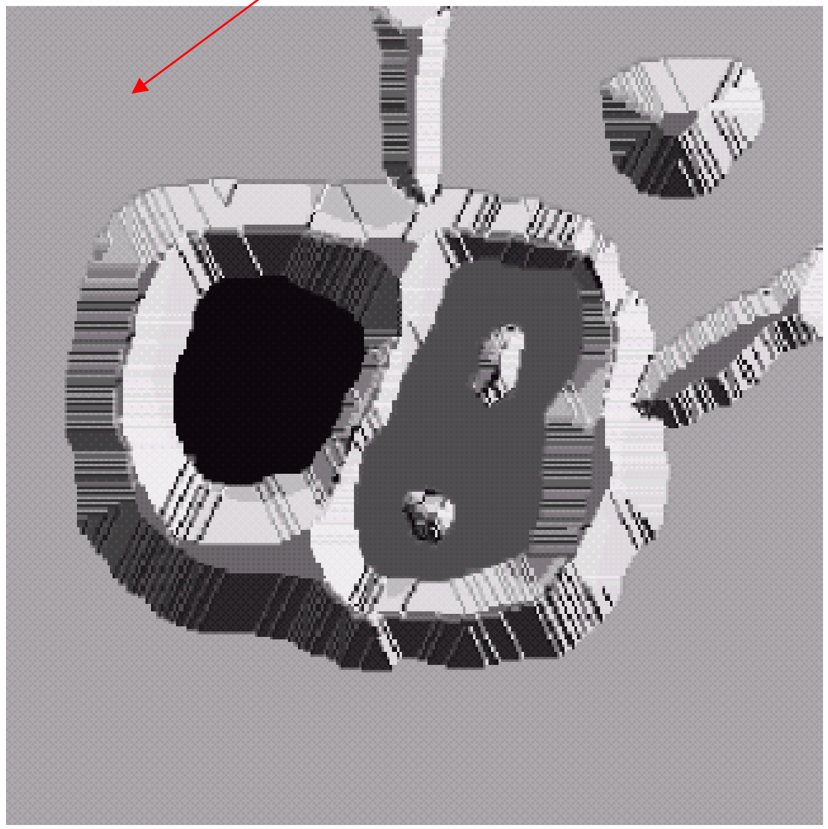


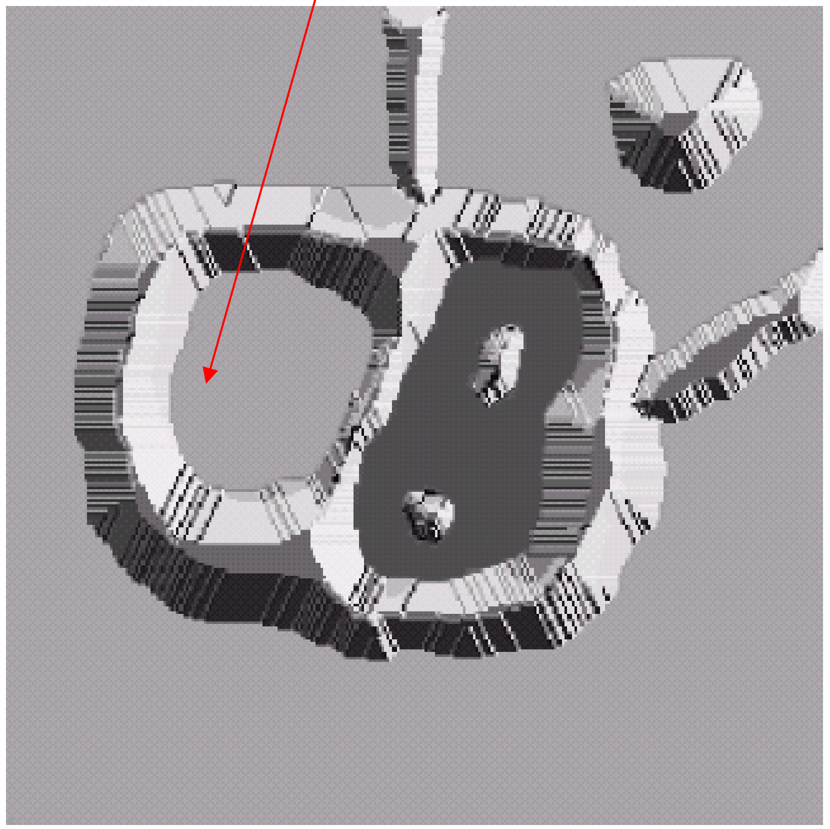
© 2004, Mingye Ding, http://www.imaging.robarts.ca/coders/ content/seminars/bme519/ws-bme519.ppt

First Stage of Flooding

Water floods the background first, as it is the darkest region

Water then floods this catchment basin

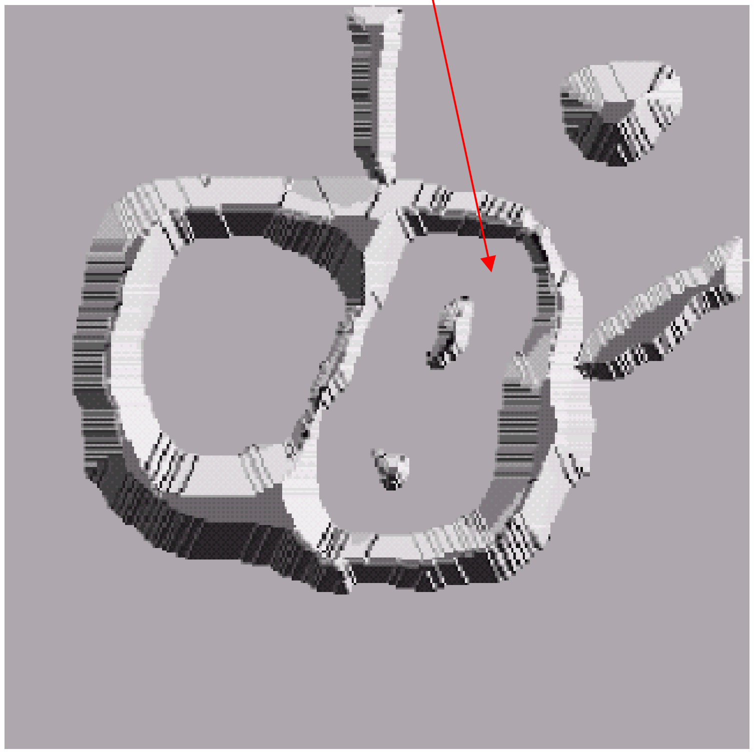


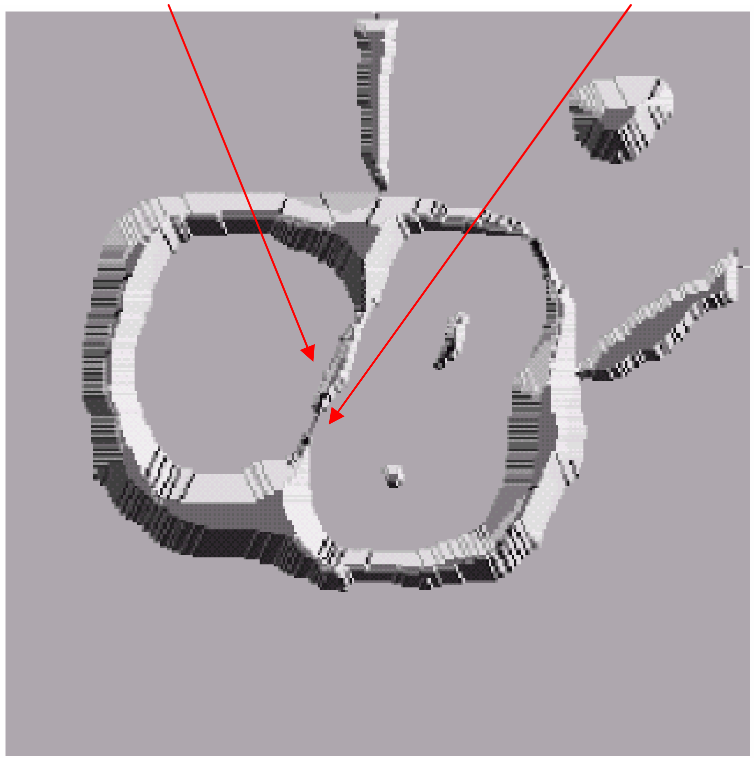


Further Flooding

Water now floods the next catchment basin

Two catchment basins are going to merge – Build dams on the merging line, which is the watershed boundary

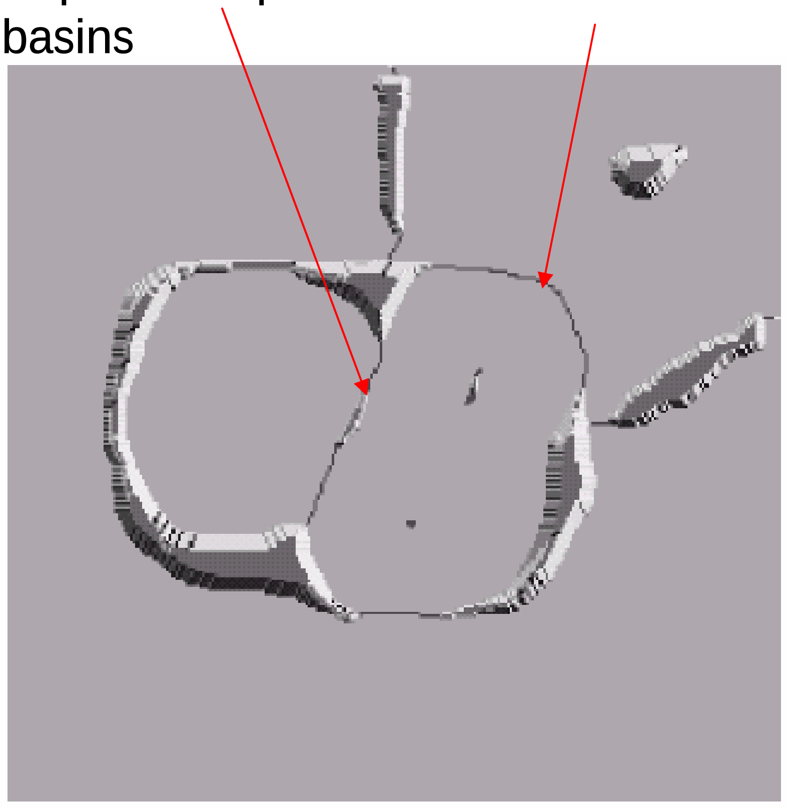




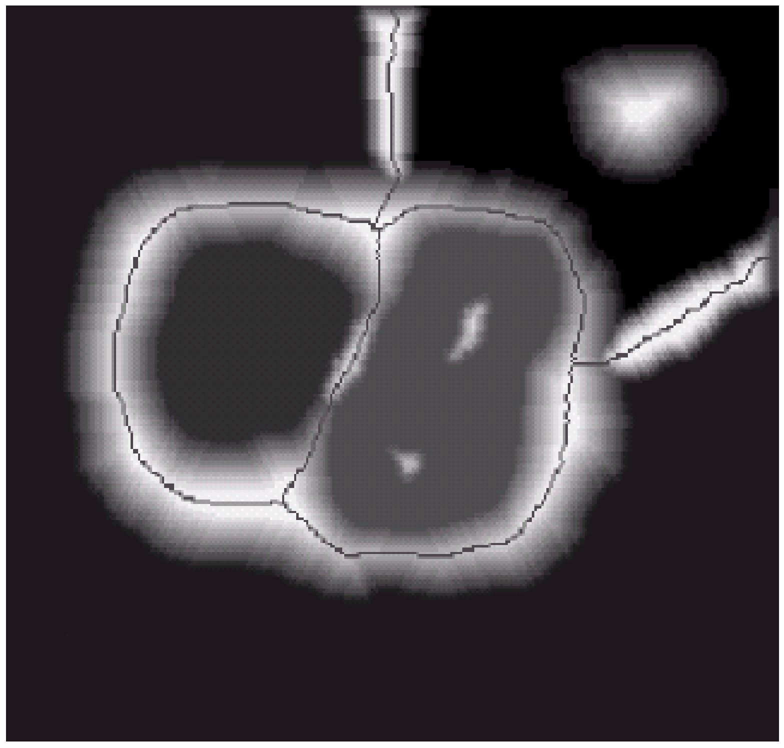
© 2003, Ninad Thakoor, http://visionlab.uta.edu/~ninad/ other_files/Image_Segmentation.ppt

Final Result

Further flooding – longer dams are required to prevent spill-over to other catchment basins

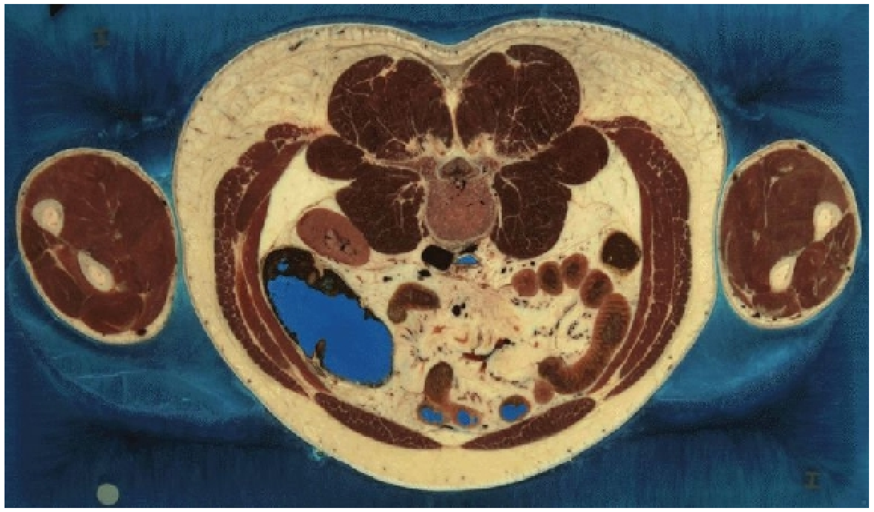


Final watershed boundary

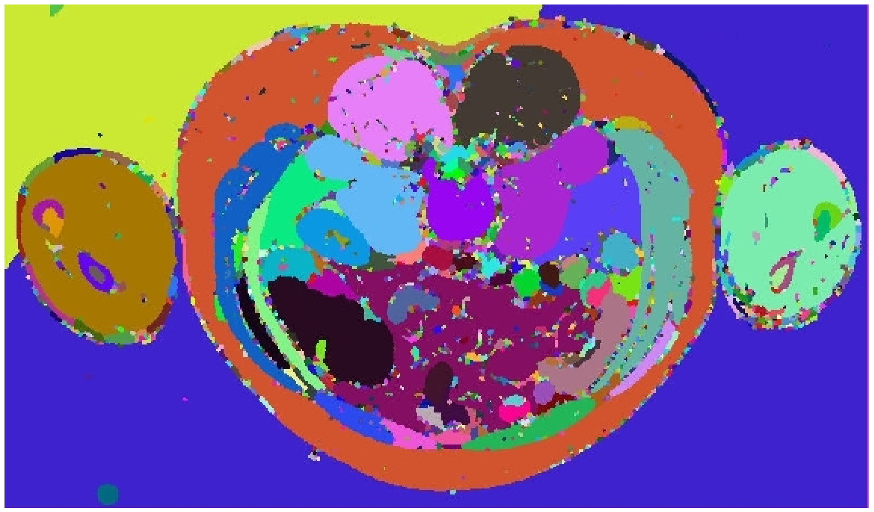


Example

24 bit RGB image taken from Visible Human data



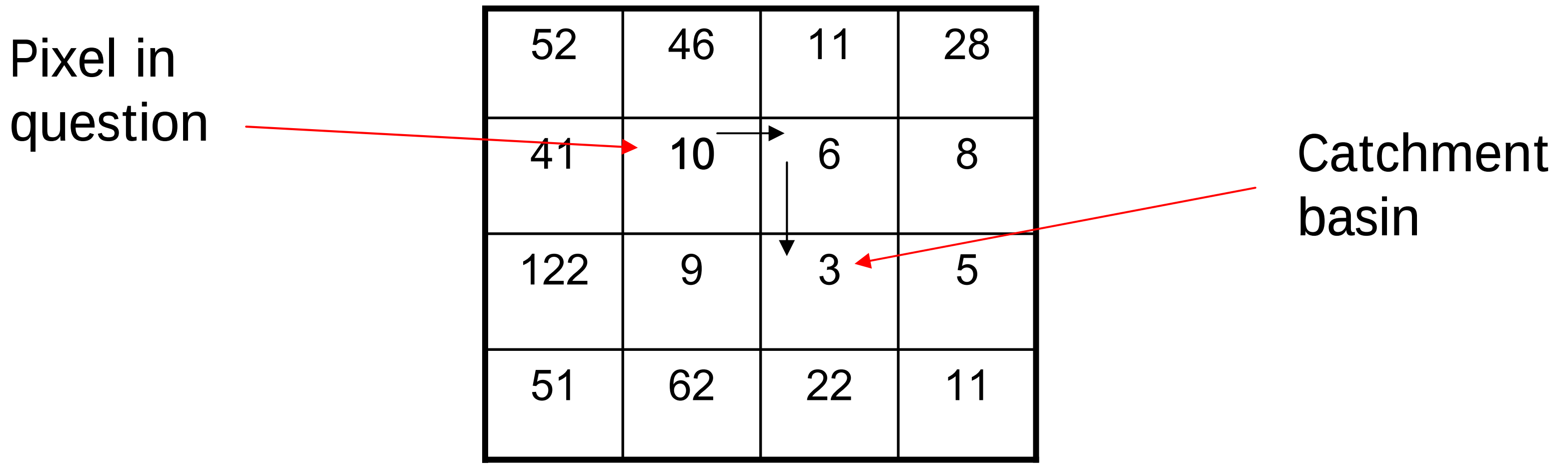
After Watershed Segmentation



© www.itk.org/HTML/WatershedSegmentationExample.html

Minimum Following Algorithm

- Using “runoff water” analogy, we could also compute the watershed regions by evaluating which pixels “drain” into which catchment basins
- In other words, we can trace the flow (in the direction of the steepest descent) to match each pixel to a local minimum (a catchment basin)



52	46	11	28
41	10	6	8
122	9	3	5
51	62	22	11

© 2003, Ninad Thakoor, http://visionlab.uta.edu/~ninad/ other_files/Image_Segmentation.ppt

Minimum Following Algorithm

- Calculate the gradient of the image
- Blur the image slightly
- Find all local minima (pixels that are less than all of neighbors) – number these; they are the catchment basins
- For each pixel in the image:
  - Find the minimum value in its neighborhood – continue following minimum values until a catchment basin or previously labeled pixel is encountered
  - Label the pixel with the catchment basin number
- The boundaries between the catchment basins define the watershed boundaries

© 2003, Ninad Thakoor, http://visionlab.uta.edu/~ninad/ other_files/Image_Segmentation.ppt

Limitations

- Drawback of raw watershed segmentation:
  - Oversegmentation
- Reasons:
  - Noise
  - Small catchment basins
- Solution:
  - Region growing post-processing using homogeneity criteria or region markers

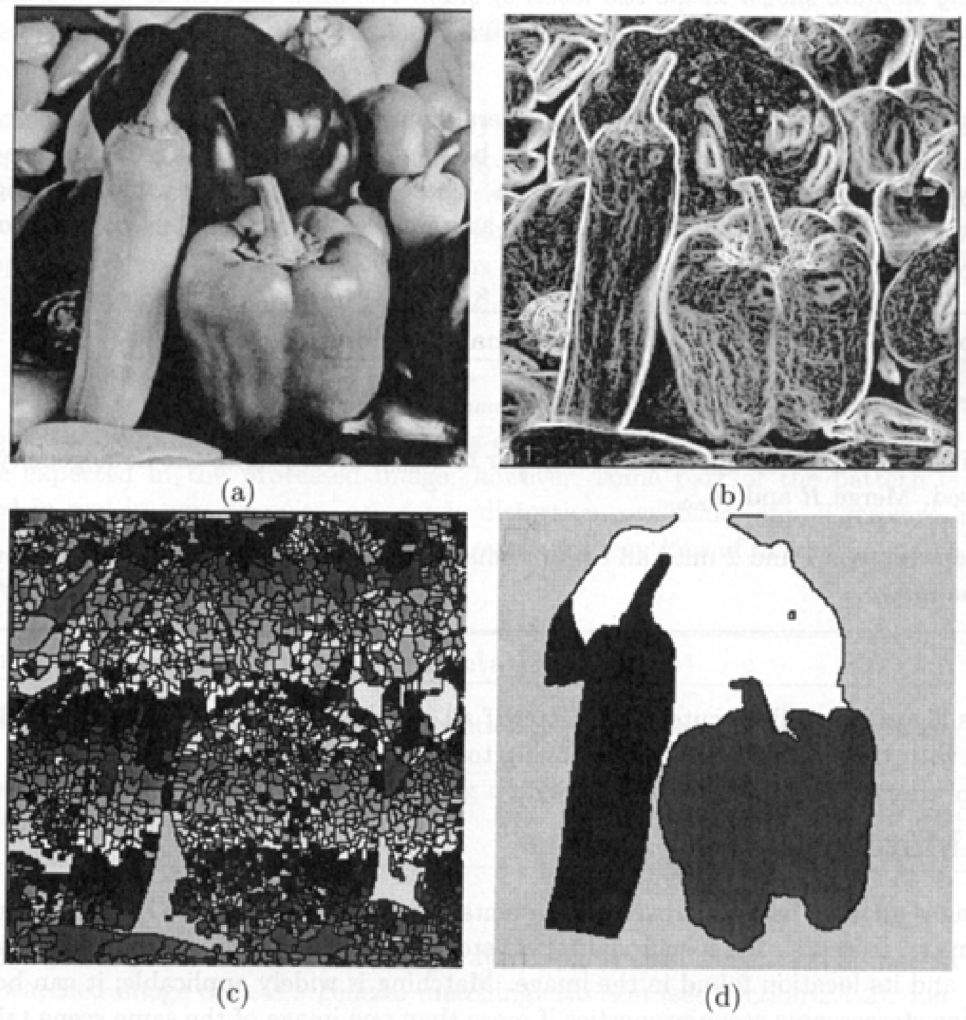


Figure 5.51: *Watershed segmentation: (a) original; (b) gradient image, 3×3 Sobel edge detection, histogram equalized; (c) raw watershed segmentation; (d) watershed segmentation using region markers to control oversegmentation. Courtesy W. Higgins, Penn State University.*

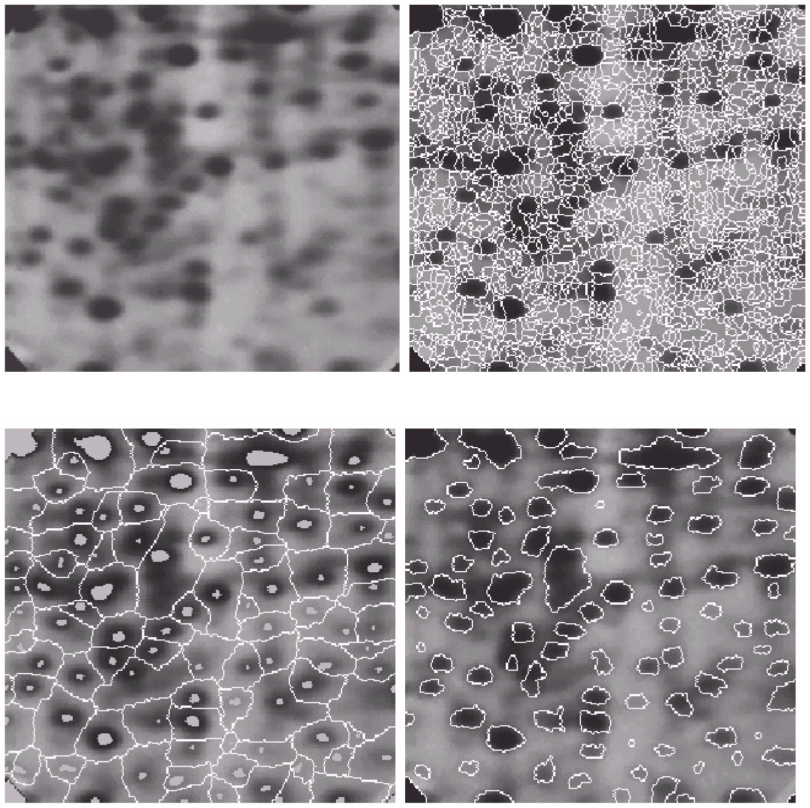
©ct.radiology.uiowa.edu/~jiangm/courses/dip/html/node138

Use of Markers

- Direct application of watersheds leads to over-segmentation due to noise and irregularities in gradient
- Marker is connected component belonging to an image
- Internal markers: related to objects of interest
- External markers: associated with background

Selection of Markers

- Preprocessing
  - Smoothing
- Set of criteria that markers must follow
  - Region surrounded by points of higher altitude
  - Points form connected components
  - Points of connected component have same gray value

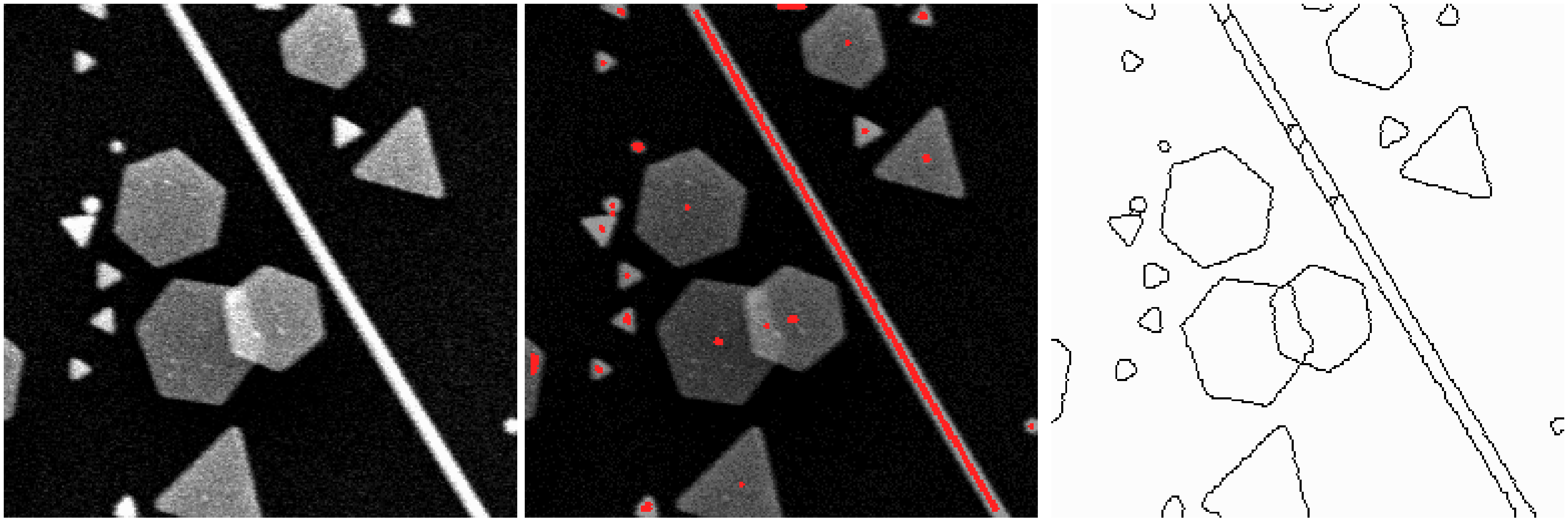


© 2003, Ninad Thakoor, http://visionlab.uta.edu/~ninad/ other_files/Image_Segmentation.ppt

Demo Movie of Marker-Based Watershed

© 2004, Mingye Ding, http://www.imaging.robarts.ca/coders/ content/seminars/bme519/ws-bme519.ppt

Marker-Based Watershed



© 2004, Mingye Ding, http://www.imaging.robarts.ca/coders/ content/seminars/bme519/ws-bme519.ppt

Binary Images

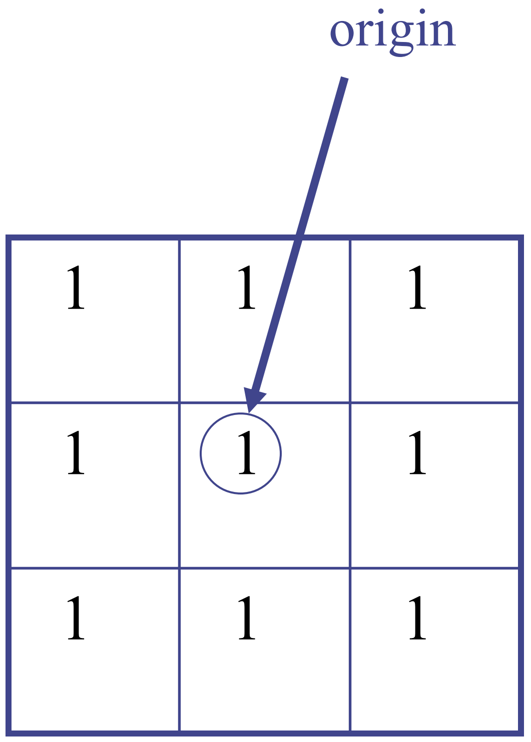
- Those have only two gray levels. A binary image normally results from an image segmentation operation
  - Advantages
    - Easy to acquire: simple digital cameras can be used together with very simple frame stores, or low-cost scanners, or thresholding may be applied to grey-level images
    - Low storage: no more than 1 bit/pixel, often this can be reduced as such images are very amenable to compression (e.g. run-length coding)
    - Simple processing: the algorithms are in most cases much simpler than those applied to grey-level images
  - Disadvantages
    - Limited application: as the representation is only a silhouette, application is restricted to tasks where internal detail is not required as a distinguishing characteristic

Binary Image Processing

- If the initial segmentation is not completely satisfactory, some processing can be done on binary image to improve the situation
  - Advantages:
    - Easy to perform
    - fast
  - Operations:
    - Morphological operations - add or remove pixels based on the pattern of neighboring pixels
    - Boolean operation - combine several binary images using logical operations on each pixel

Structuring Element

- The number of pixels added or removed from the objects in an image depends on the size and the shape of the ***structuring element*** used to process the image
- *Structuring element*, which consist of a matrix of 0's and 1's with arbitrary shape and size, is used to probe the input image



Structuring Element

Erosion

- The erosion of set B by set S (*structure element*), denoted $B \otimes S$, is the set intersection of all negative translates of set B by elements of set S
- Simple *erosion* is the process of eliminating all the boundary points (*a pixel that is located inside an object, but has at least one neighbor outside object*) from an object.
- Erosion makes the object smaller in area by one pixel all around its perimeter
- Erosion reduces the size of objects in relation to their background

Erosion is useful for removing from a segmented image objects that are too small to be of interest

Erosion (cont.)

- Erosion is defined by

$$E = B \otimes S = \{x,y \mid Sxy \subseteq B\}$$

The binary image **E** that results from eroding **B** *(binary image)* by **S** (*structure element*) is the set of points (***x, y***) such that

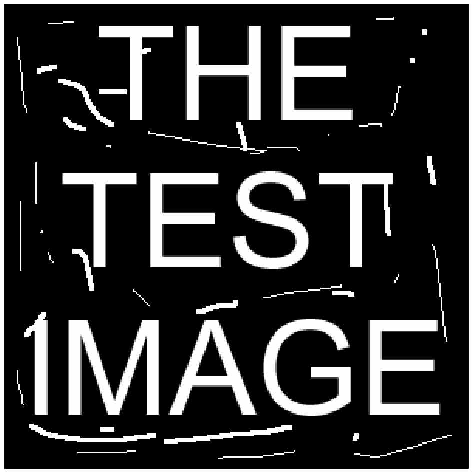
if

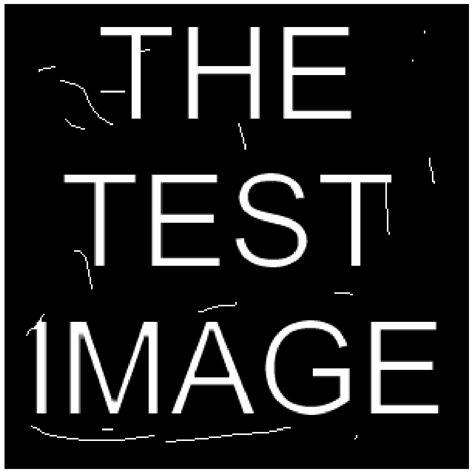
S is translated so that its origin is located at (***x, y***)

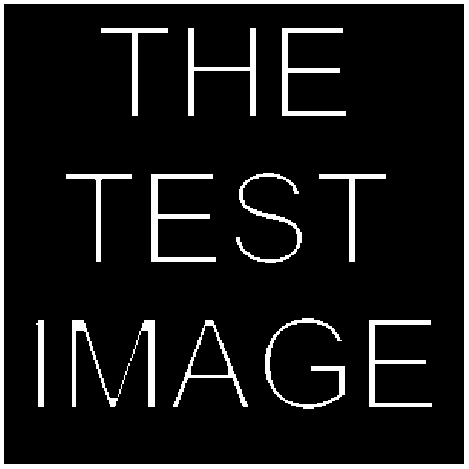
then

it is completely contained within **B**

Example of Erosion







Original image

Eroded image once

Eroded image twice

©www.cs.mdx.ac.uk/staffpages/peter/com3325

Dilation

- Binary dilation of set B by set S (*structuring element*), denoted $B \oplus S$, is the union of all translates of set B by elements of set S.
- Simple dilation is the process of incorporating into the object all the background points that touch it.
- Dilation leaves the object larger in area by one pixel all around its perimeter
- Dilation increases the size of objects in relation to their background

Dilation is useful for filling holes in segmented image objects

Dilation (cont.)

- Dilation is defined by

$$D = B \oplus S = \{x, y \mid S_{xy} \cap B \neq \varnothing\}$$

The binary image **D** that results from dilating **B** by **S** is the set of points (***x, y***) such that

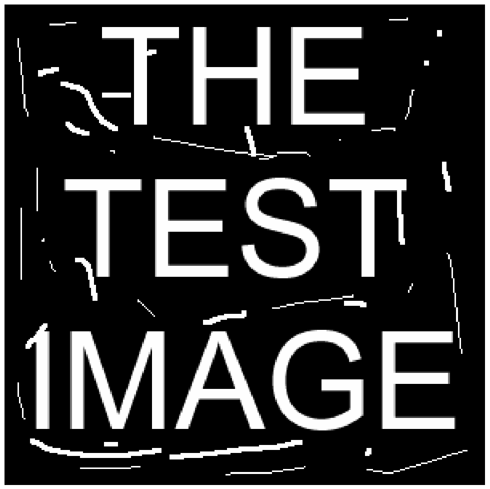
if

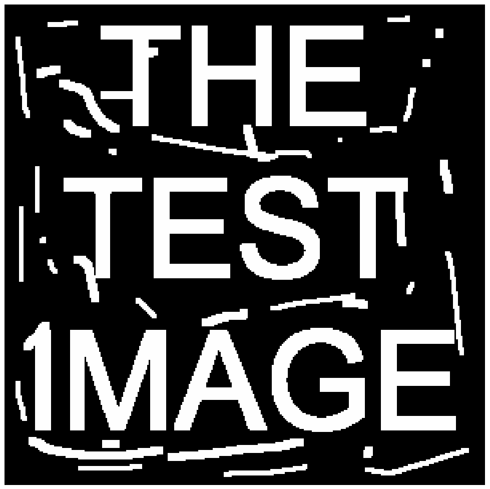
S is translated so that its origin is located at (***x, y***)

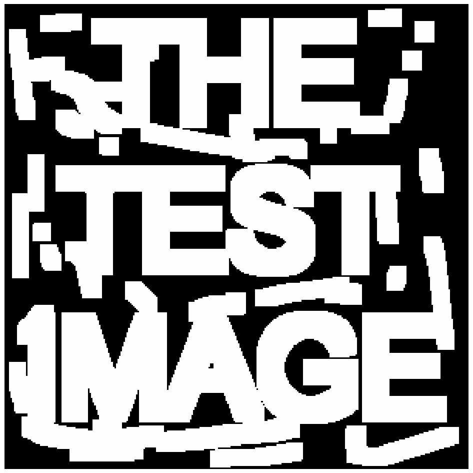
then

its intersection with **B** is not empty

Example of Dilation







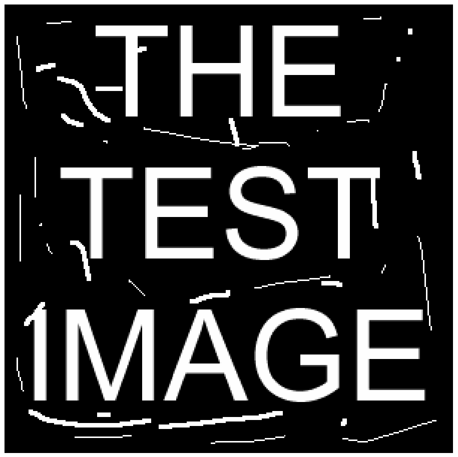
Original image Dilated image Dilated many times

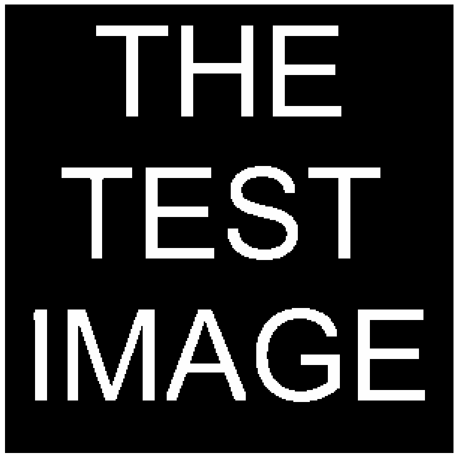
Opening & Closing

- The process of erosion followed by dilating is called opening
- Opening has the effect of elimination small and thin objects
- Opening is defined by $B \circ S = (B \otimes S) \oplus S$

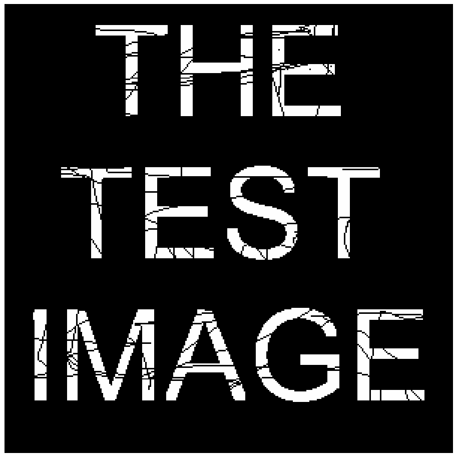
- The process of dilating followed by erosion is called closing
- Closing has the effect of filling small and thin holes in objects
- Closing is defined by $B \bullet S = (B \oplus S) \otimes S$

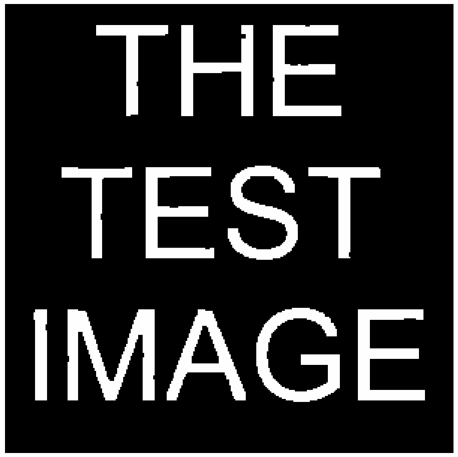
Example of Opening & Closing





The test image eroded twice and dilated twice (opened). Most noise is removed





The testimage dilated twice and eroded twice (closed). Most small gaps are removed.

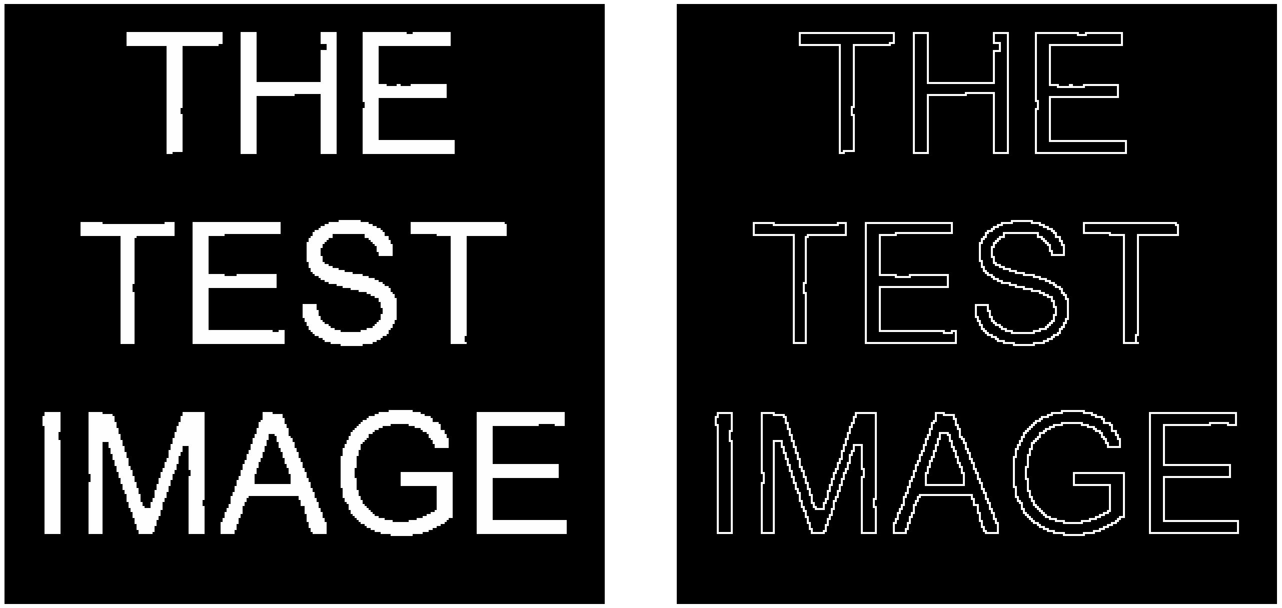
©www.cs.mdx.ac.uk/staffpages/peter/com3325

More Variants of Erosion and Dilation

- Shrinking : Erosion is implemented keep single-pixel objects intact
- Thinning : apply erosion in two steps
  - First step : mark the pixels for removal instead of removing them
  - Second step : remove the marked pixels without destroying connectivity
- Thickening : a process to apply dilation in two passes
- Outlining
- Skeletonization

Outlining

- Implemented by applying an erosion operation followed by subtraction of the eroded image from the original image



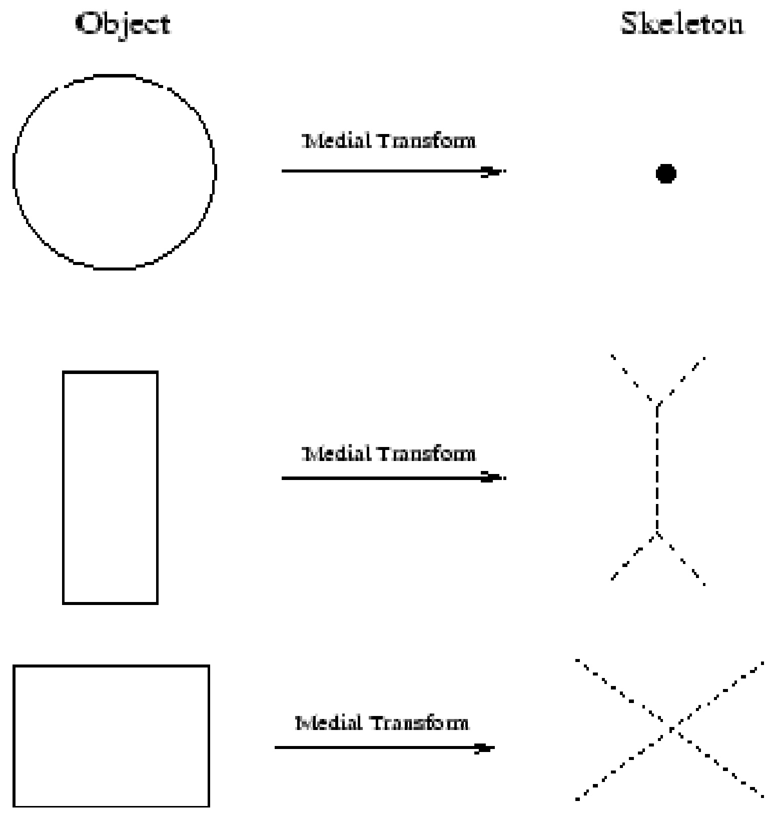
Skeletonization

- The process of peeling off as many pixels as possible without affecting the general shape of the pattern
- It is a specialized erosion that preserves an important characteristic of feature shapes
  - It is produced by an iterative erosion that removes all pixels that are not part of the skeleton
    - **Do not remove a single pixel**
    - **Do not break the connectivity**
    - **Do not remove an end-pixel**
- Skeleton hence obtained must have the following properties
  - as thin as possible
  - connected
  - centered
- Application: Compact shape representation, work best on long and thin features

Skeletonization

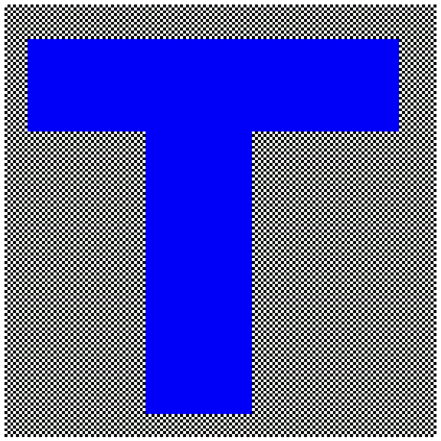
- Skeletonization of an image is obtained by a *medial axis transform* , where the medial axis is the locus of points such that any medial point is equidistant to at least two points on the boundary

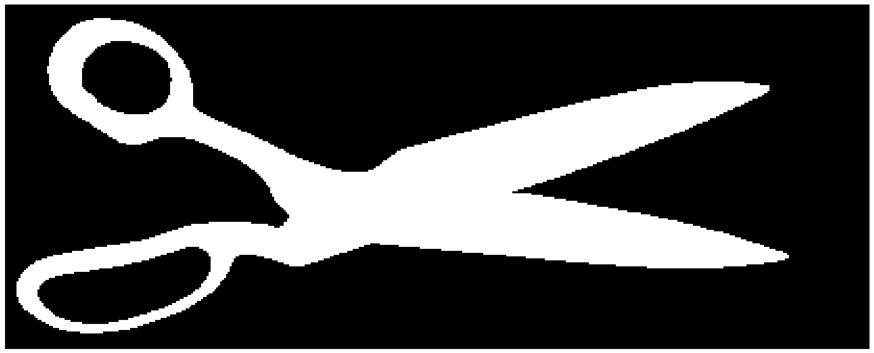
- Mathematically, the medial axis can be found by finding the largest circle that can completely fit into the object and still touches at least two boundary points. The center of the circle partly defines one of the points along the medial axis. Then find the next smaller circle that just fits inside the object, and the center points of all such touching circles define the entire medial axis



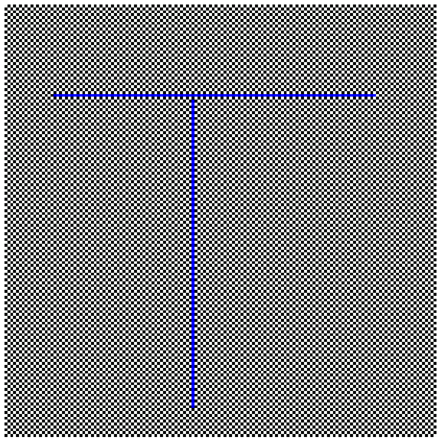
©utam.geophys.utah.edu/ebooks/gg527/impro/ch6/ch/node11

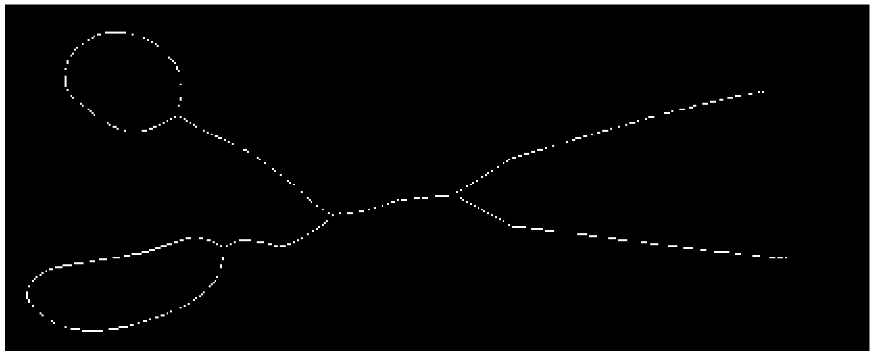
Examples of Skeletonization





©cgm.cs.mcgill.ca/~godfried/teaching/projects97/azar/skeleton





© http://bavaria.utcluj.ro/~alinm/image.html

Resources

Textbooks:

Kenneth R. Castleman, Digital Image Processing, Chapter 6,7,18

John C. Russ, The Image Processing Handbook, Chapter 3,4,6,7

Reading Assignment

Textbooks:

Kenneth R. Castleman, Digital Image Processing, Chapter 8, 9

John C. Russ, The Image Processing Handbook, Chapter 5