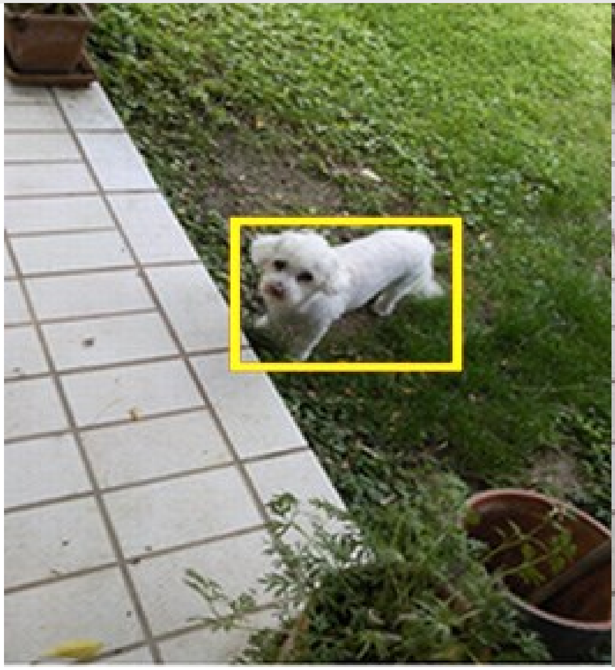
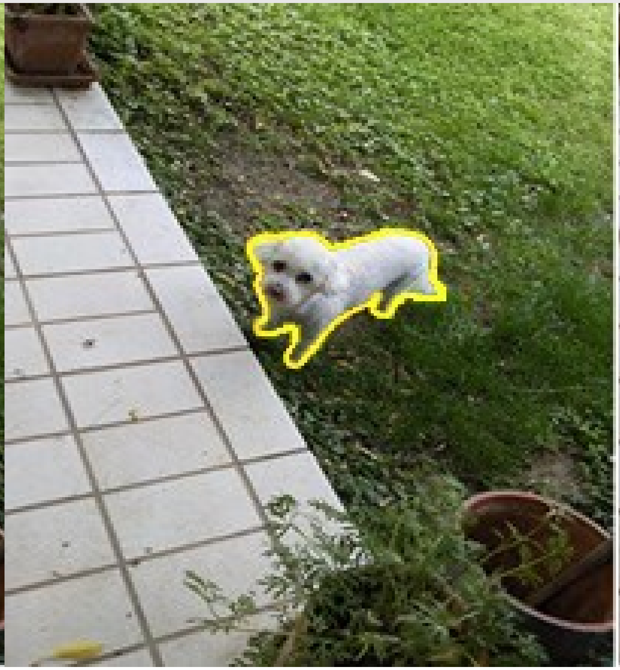
Computer vision

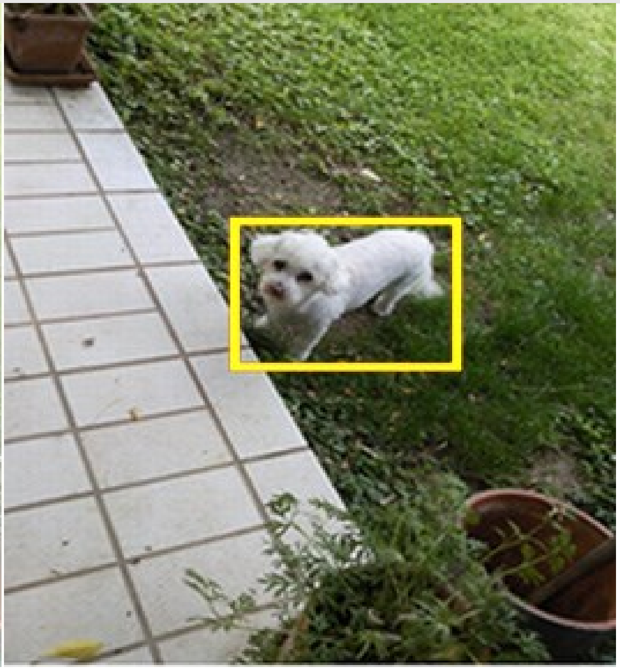
Image segmentation

Doc. Ing. Vanda Benešová, PhD.

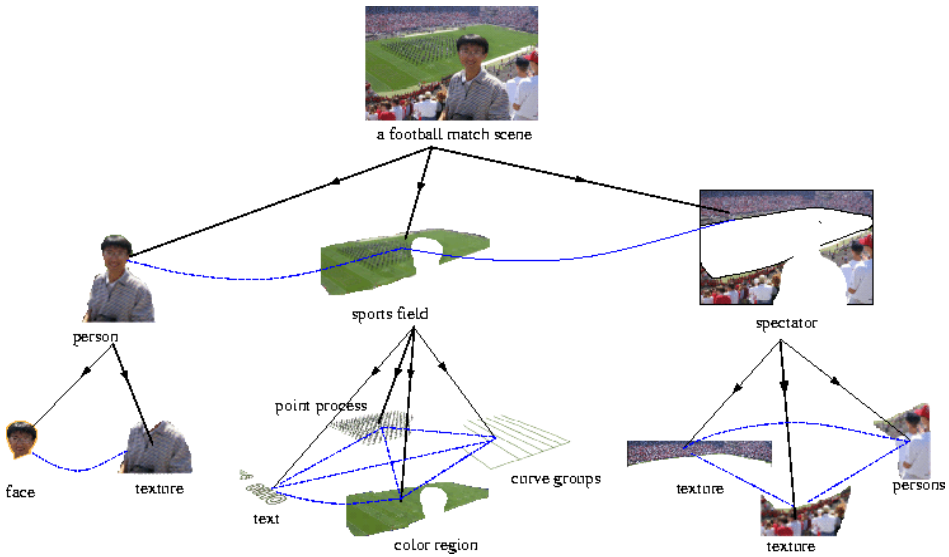
Type of task


Object detection

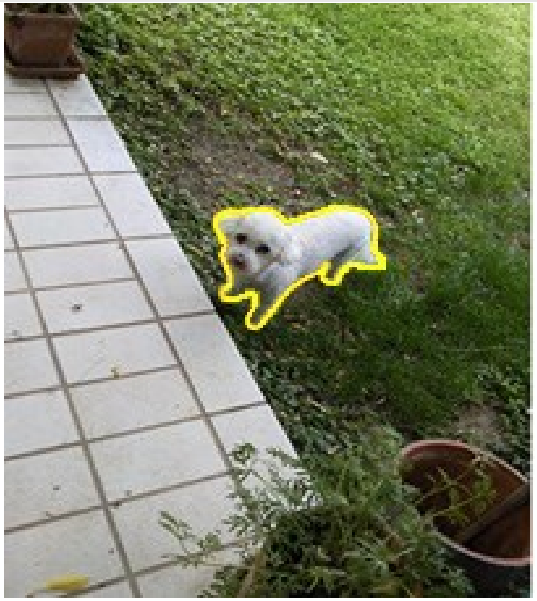

Object segmentation


Object detection and recognition

Complete segmentation/ Partial segmentation



Complete segmentation



Partial segmentation

Segmentation methods

Edge-based methods

- Snake Model
- Graph-cut

Region based methods

- Tresholding
- Growing regions
- Clustering (unsupervised classification)
- Split/Merge

Morphological methods – watershed

Knowledge-based

Template matching

Statistical methods

Artificial intelligence, Neural networks

Segmentation is a complex task

Different tasks → specifics methods

No universal method for all tasks !

Why it is so difficult ?

Important factor – quality of the image

Complex shapes of the objects

Overlapping objects

Shadows

Pre-processing could be necessary!

Additional knowledge is important

Whenever additional knowledge is available for boundary detection -

it should be used!

Thresholding

Simple segmentation tasks

Contrasted objects on a uniform background

simple assembly tasks:

blood cells segmentation

printed characters segmentation… etc.

Global thresholding

In global thresholding, the threshold value is held constant throughout the image

Global thresholding is of the form:

$$g(x,y) = \begin{cases} 0 & f(x,y) < \text{Threshold} \\ 1 & f(x,y) >= \text{Threshold} \end{cases}$$

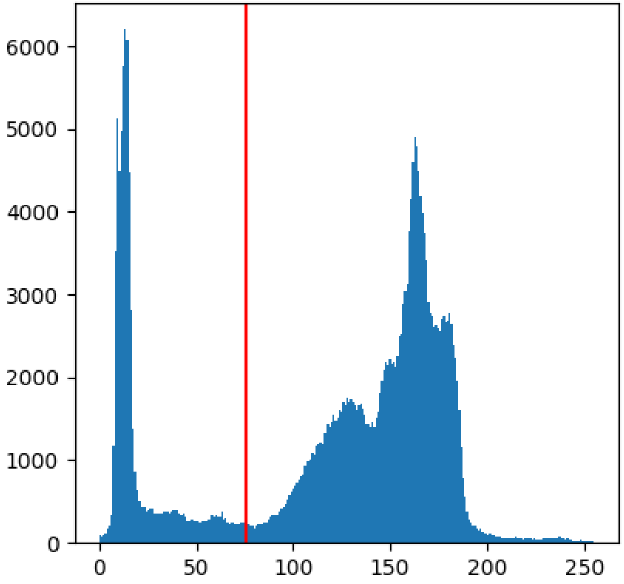
Output of thresholding -> binary mask

Thresholding

? Threshold value?

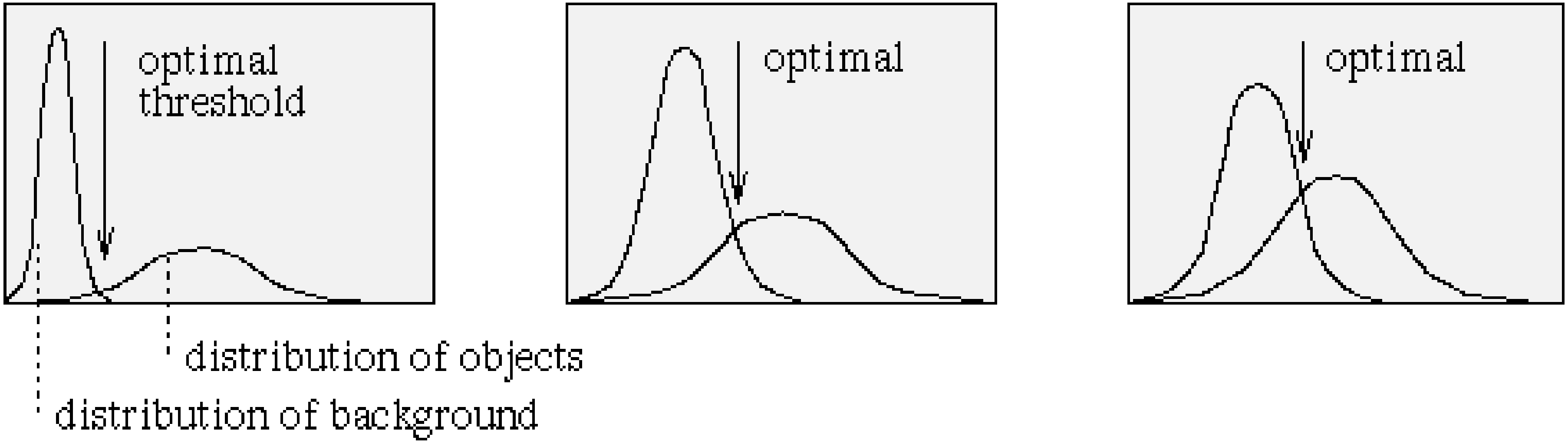
Bimodal histogram

Original



Thresholded

Optimal threshold



Probability distributions of background and object – optimal threshld results in minimum error segmentation

(Discrete probability distribution is represented by histogram)

Thresholding - Multimodal Histogram

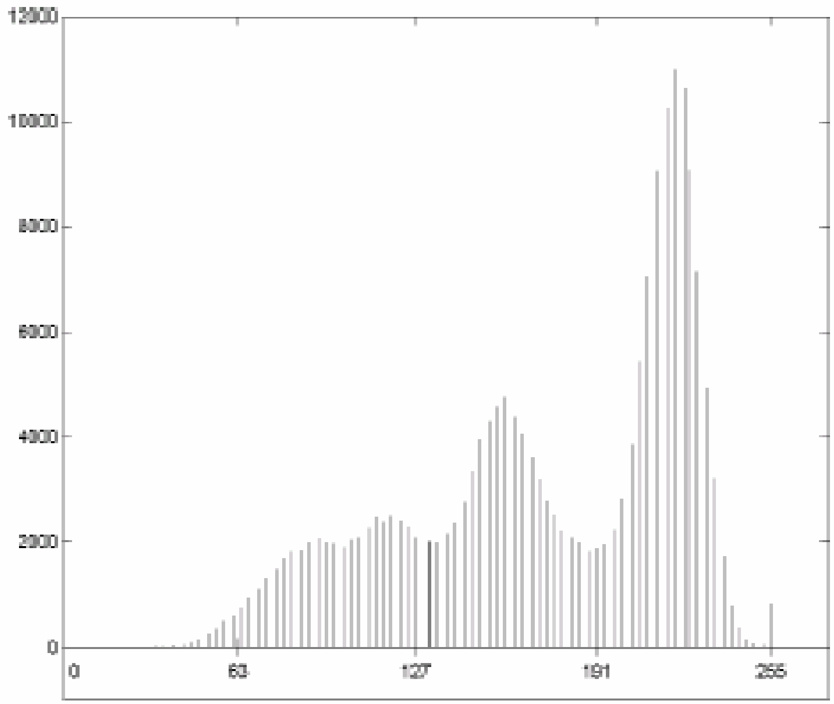
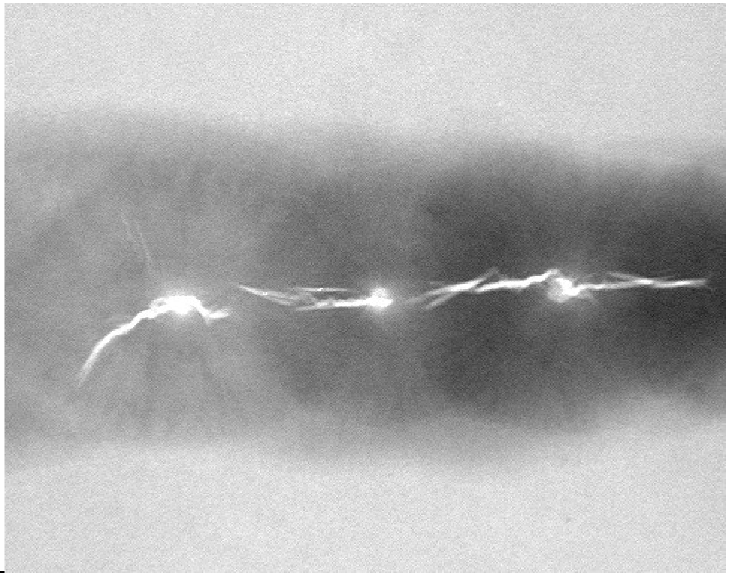


FIGURE 10.41
Histogram of Fig. 10.40(a).



Source: Digital Image Processing

by Rafael C. Gonzalez, Richard E. Woods

Example: MR images

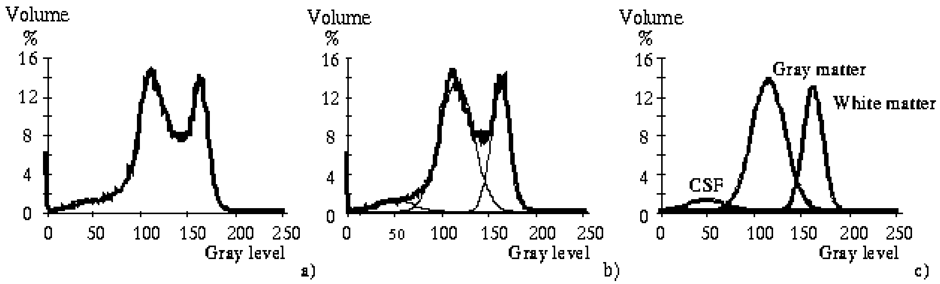


Figure 5.5 *Segmentation of 3-D T1-weighted MR brain image data using optimal thresholding: (a) Local gray level histogram, (b) fitted Gaussian distributions; global 3-D image fit, (c) Gaussian distributions corresponding to WM, GM, and CSF. Courtesy R.J. Frank, T.J. Grabowski, Human Neuroanatomy and Neuroimaging Laboratory, Department of Neurology, The University of Iowa.*

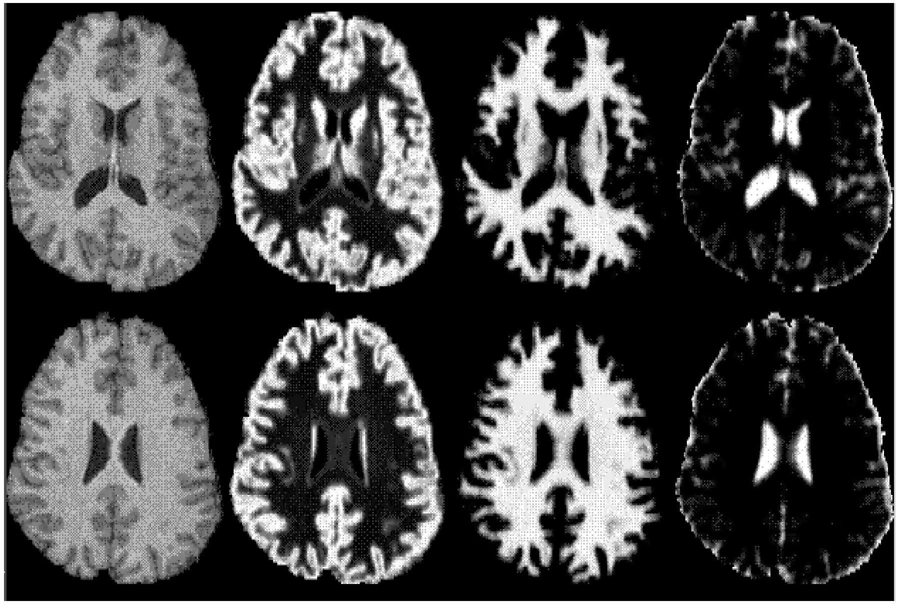


Figure 5.6 *Optimal MR brain image segmentation. Left column: Original T1-weighted MR images, two of 120 slices of the 3-D volume. Middle left: Partial-volume maps of gray matter. The brighter the voxel, the higher is the partial volume percentage of gray matter in the voxel. Middle right: Partial-volume maps of white matter. Right column: Partial-volume maps of cerebro-spinal fluid. Courtesy R.J. Frank, T.J. Grabowski, Human Neuroanatomy and Neuroimaging Laboratory, Department of Neurology, The University of Iowa.*

Otsu's method

Otsu’s thresholding chooses the threshold to minimize the intraclass variance and maximize the interclass variance between a segmented-foreground object and background.

All possible thresholds (0-255) are evaluated in this way, and that one that maximises the criterion is chosen as the optimal threshold.

Gaussian Mixture Models (GMM) Mixtture of Gaussians (MOG)

probability model determination ->

we can suppose to have mixture of Gaussian distribution

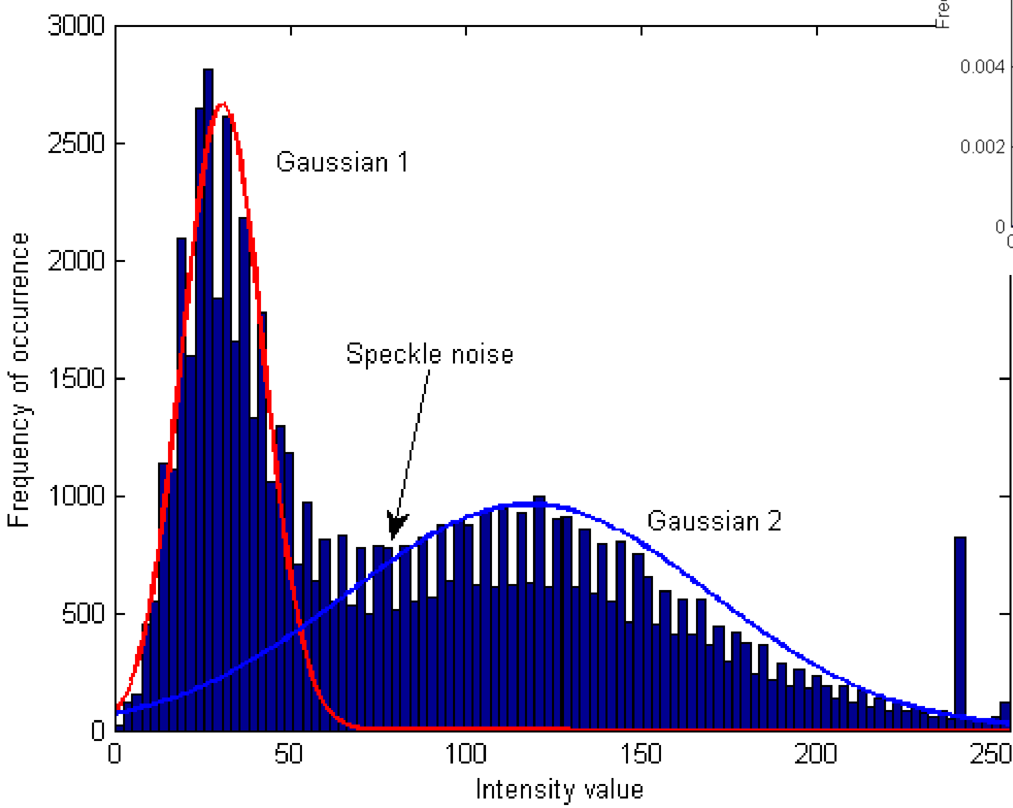
$$f(x) = \sum_{i=1}^{k} p_i N(x|\mu_i, \sigma_i^2)$$

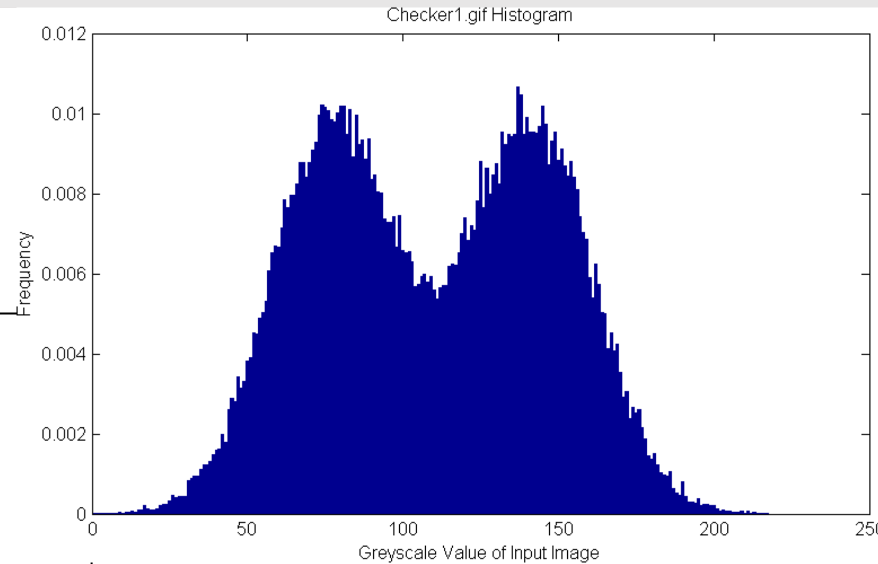
Where k is the number of regions and pi > 0 are weights such that $\sum_{i=1}^{k} p_i = 1$

$$N(\mu_i, \sigma_i^2) = \frac{1}{\sigma\sqrt{2pi}} exp \frac{-(x-\mu_i)^2}{2\sigma_i^2}$$

Where μi,σi are mean, standard deviation of class i.

Examples : Gaussian Mixture Models (GMM) Mixtture of Gaussians (MOG)





Using prior knowledge: p-tile thresholding

If some property of an image after correct segmentation is known a priori, the task of threshold selection can be simplified..

Example : a printed text sheet

if we know that characters of the text cover aprox. 1/p of the sheet area.

This method requires knowledge about the area or size of the objects present in the image. Let us assume that we have dark objects against a light background.

If, for example, the objects occupy p% of the image area, an appropriate threshold can be chosen by partitioning the histogram

Global / local / adaptive thresholding

In global thresholding, the threshold value is held constant throughout the image

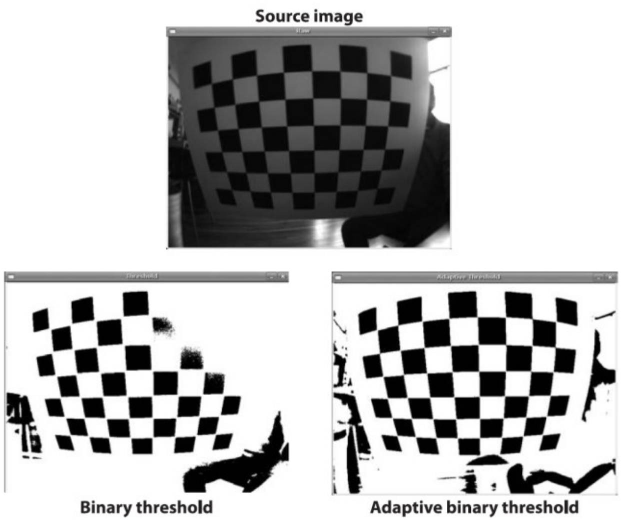
Global thresholding is of the form:

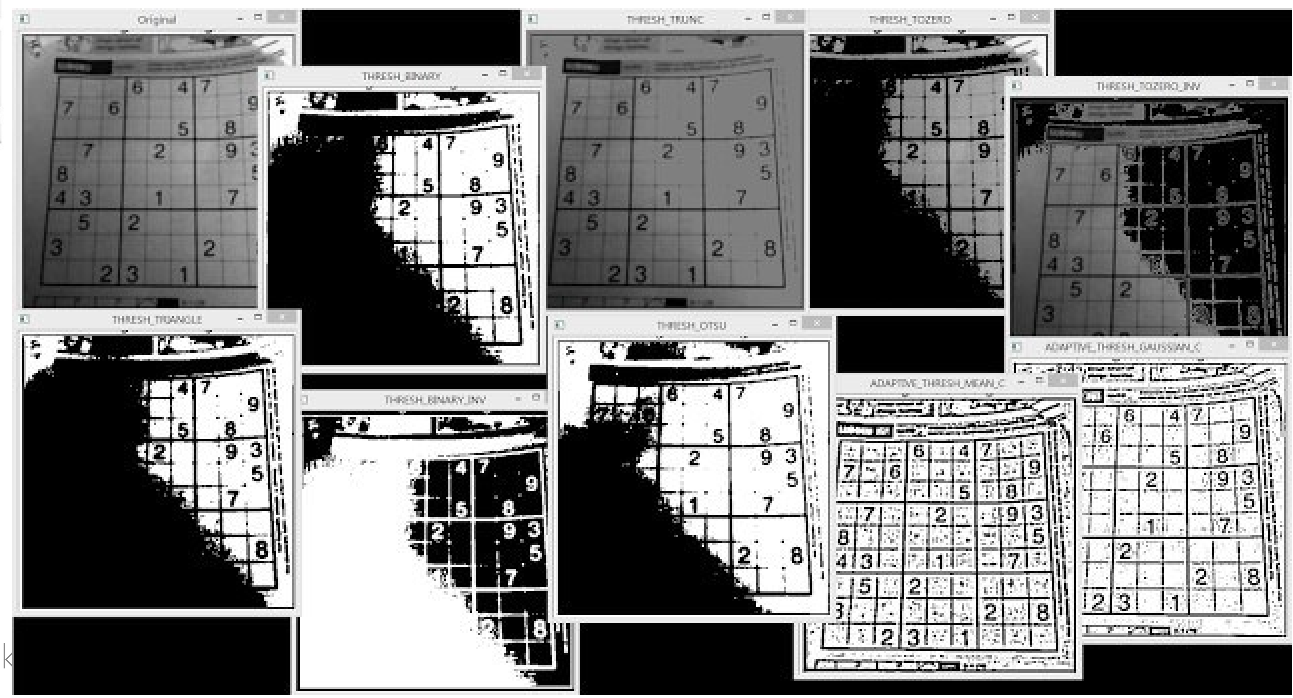
$$g(x,y) = \begin{array}{lll} 0 & f(x,y) & < \text{Threshold} \\ 1 & f(x,y) & >= \text{Threshold} \end{array}$$

Local threshold –> eliminate changes in illumination

Adaptive thresholding - technique in which the threshold level is variable.

Global binary threshold versus adaptive binary threshold





Multispectral thresholding

…multispectral , multimodal (MRI – T1,T2,flair) images

- Image fusion
  - combines the input images
- Information fusion
  - thresholds independently in each spectral band and
  - combines them into a single segmented mask image

Segmentation based on edge detection

Segmentation based on edge detection

Segmentation based on edge detection

Determination of object boundaries from a edge image (Sobel, Laplace, Canny detector...)

Problems:

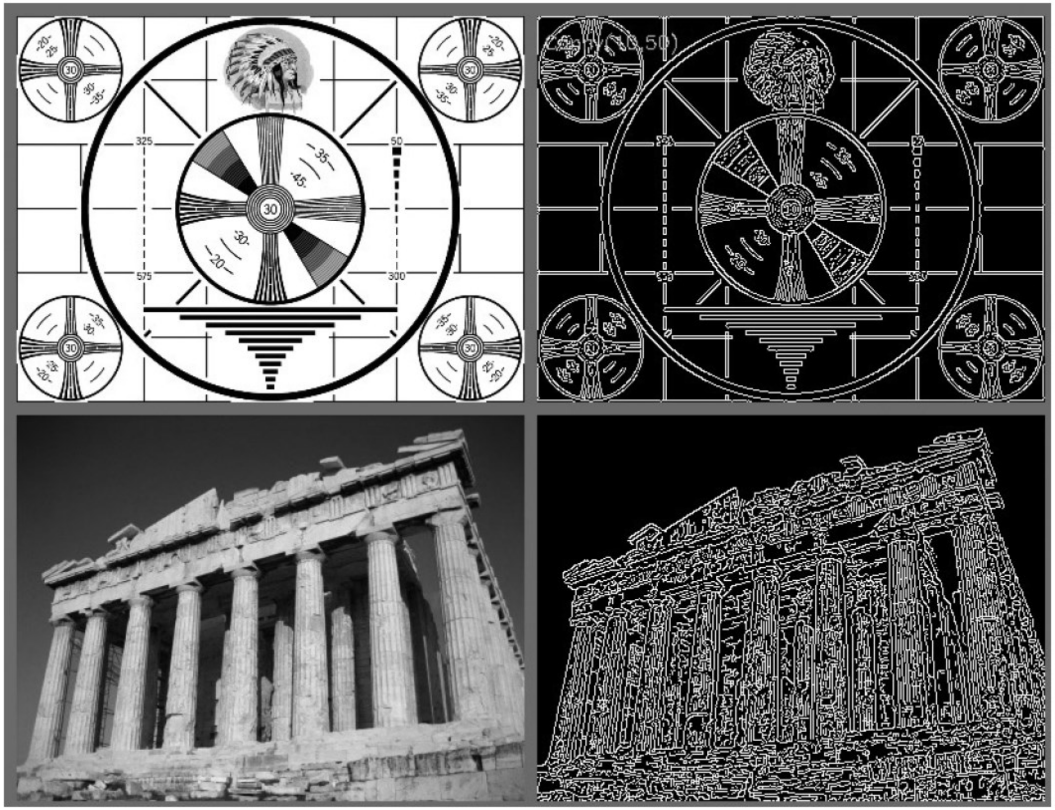
Edges are overlapped, not continuous...

Non relevant edges have to be avoid

Pre-processing and post-processing could be necessary...

An Example - Results of Canny edge detection

Alternatively, non-maximal suppression and hysteresis thresholding can be used as was introduced in the Canny edge detector.



Segmentation using edge detection

typical problem edges are not closed

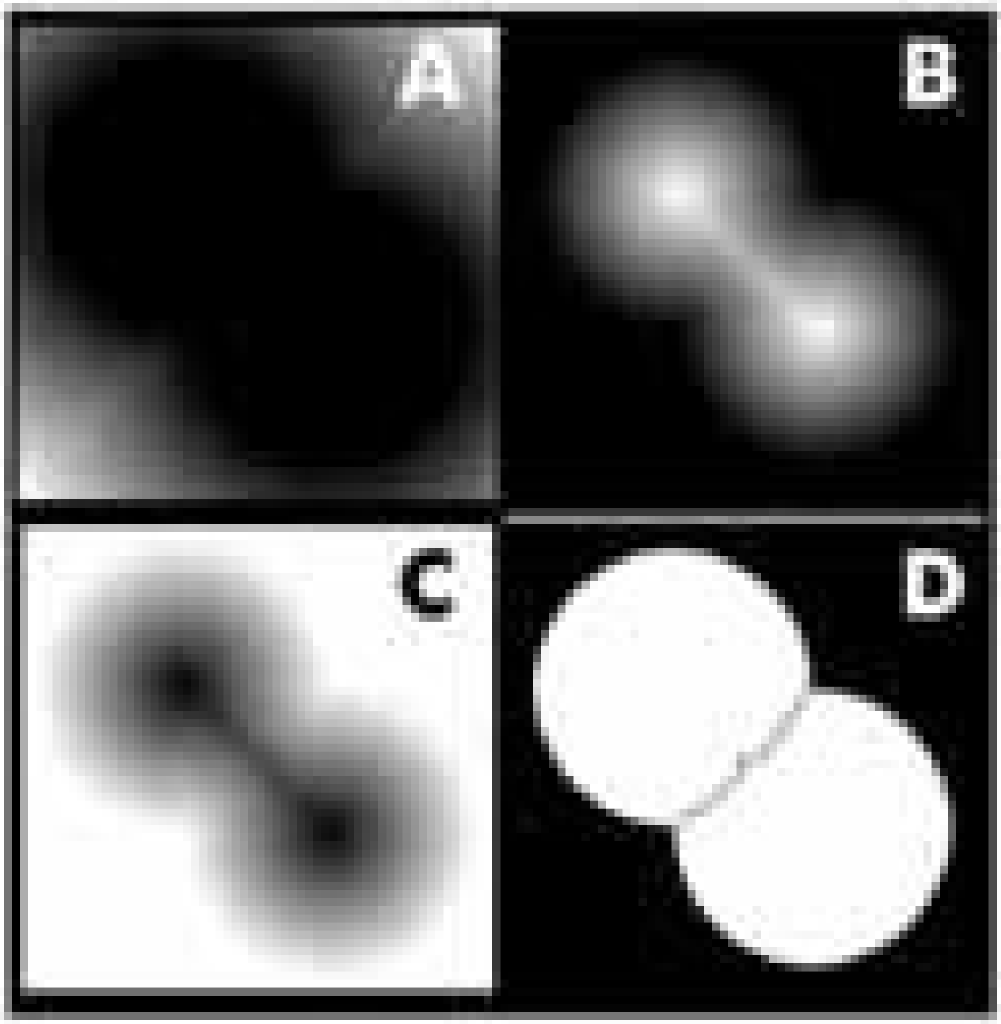
One possible solution →

Using the methods:

1: Distance transform on the edge image

2. Watershed transform on the distance image

Watershed Segmentation for closing the edges



B: Distance transform

C: Image's Complement

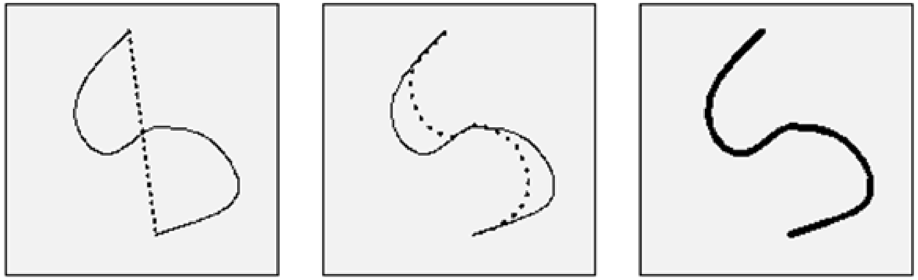
D: Watershed

Active contour model-snake

Active contour model- snake

The active contour model- snake, is defined as an energy minimizing spline - the snake's energy depends on its shape and location within the image.

Local minima of this energy then correspond to desired image properties.



(a) (b) (c)

a) initial snake position

b-c) iteration steps of snake energy minimization

Parametric Snake Model

- The total energy of the snake is defined as

$$E_{\text{total}} = E_{\text{internal}} + E_{\text{external}}$$

Encourages smoothness or a particular shape

Encourages curve onto image structures (eg edges)

Snake Model - Internal Energy

For a curve represented as a set of points.

- The elastic (length) energy can be expressed as

$$E_{in\_elasticity} = \alpha \sum_{i=0}^{n-1} L_i^2$$

$$= \alpha \sum_{i=0}^{n-1} (x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2$$

Snake Model - External Energy

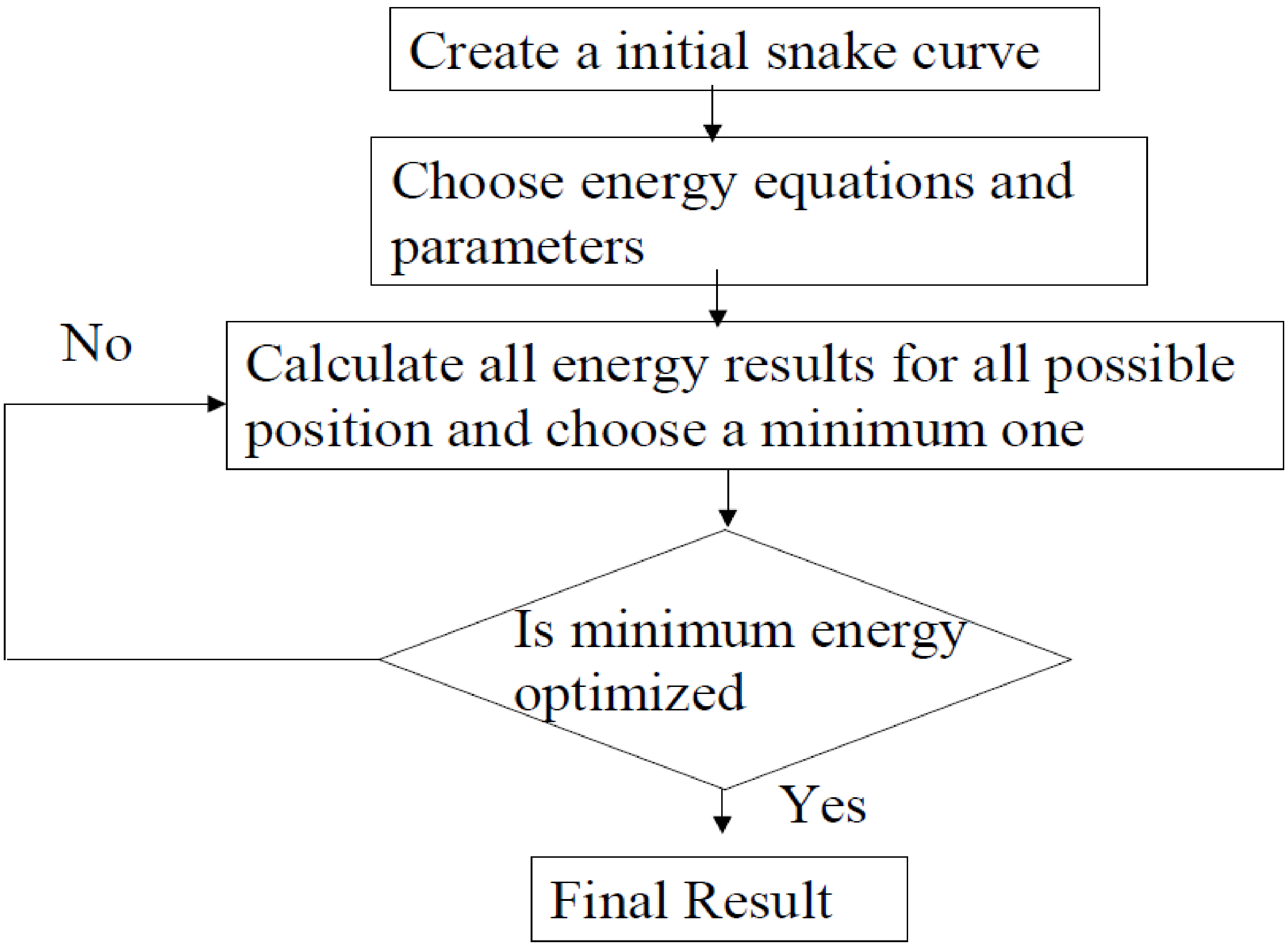
The external energy describes how well the curve matches the image data locally.

Numerous forms can be used, attracting the curve toward different image features.

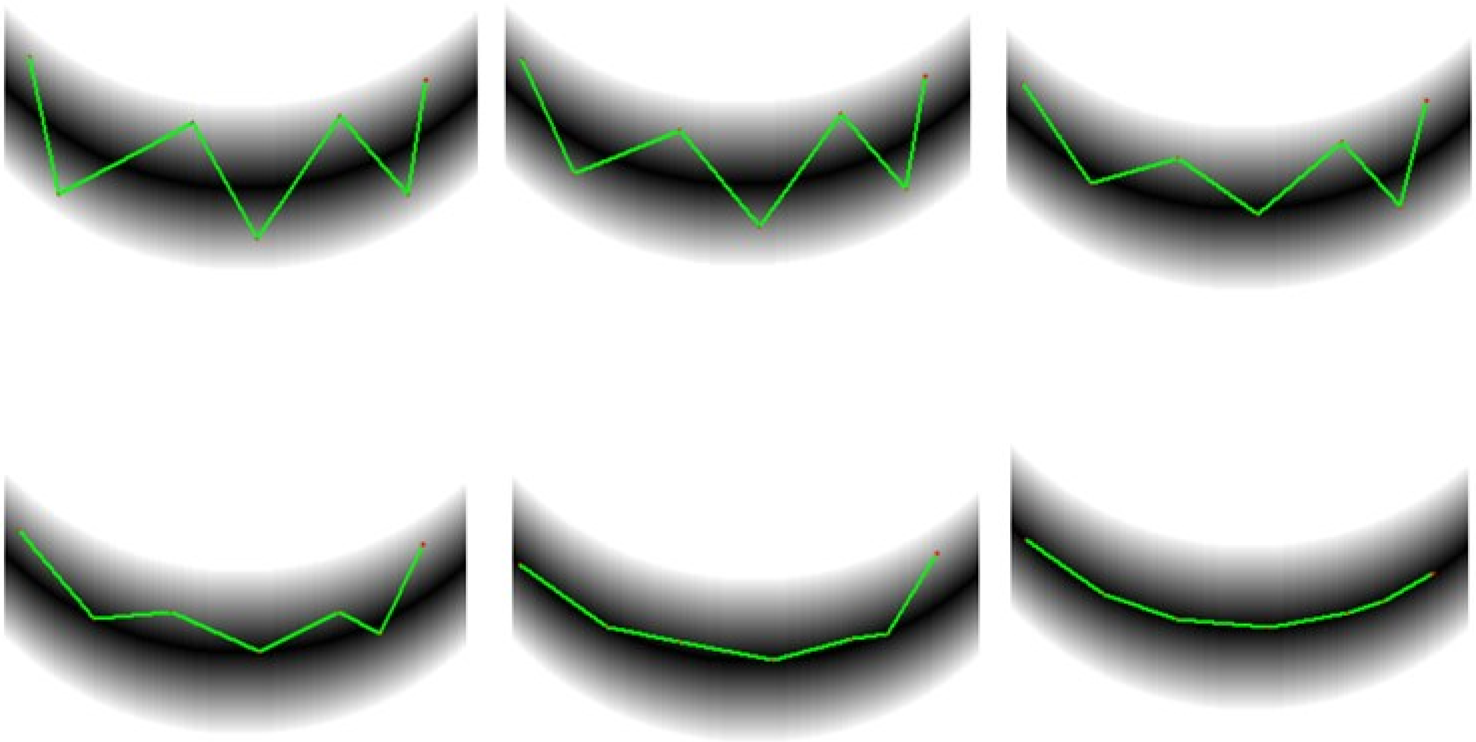
1. Gradient energy form:

$$E_{ext\_grad} = -\lambda_1 \sum_{i=0}^{n-1} \left(\left| G_x(x_i, y_i) \right|^2 + \left| G_y(x_i, y_i) \right|^2 \right)$$

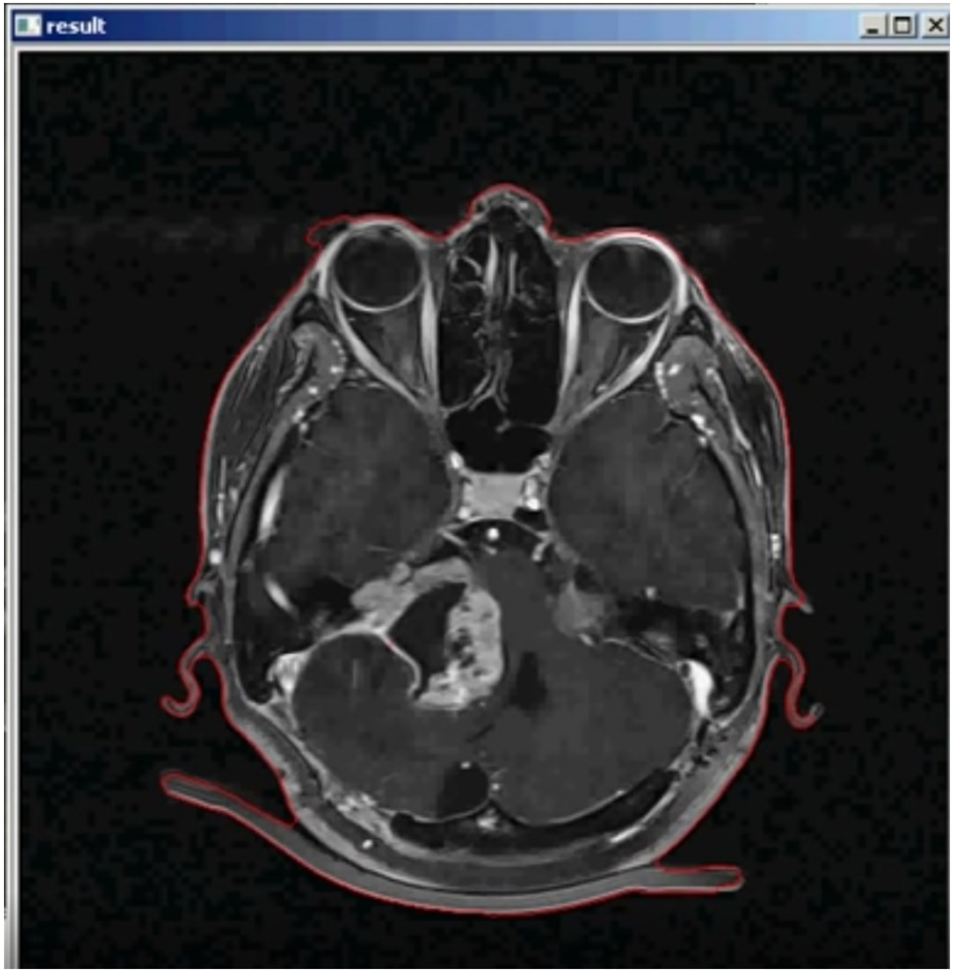
Snake Model - Flow Chart



Snake Model - Demo Pictures

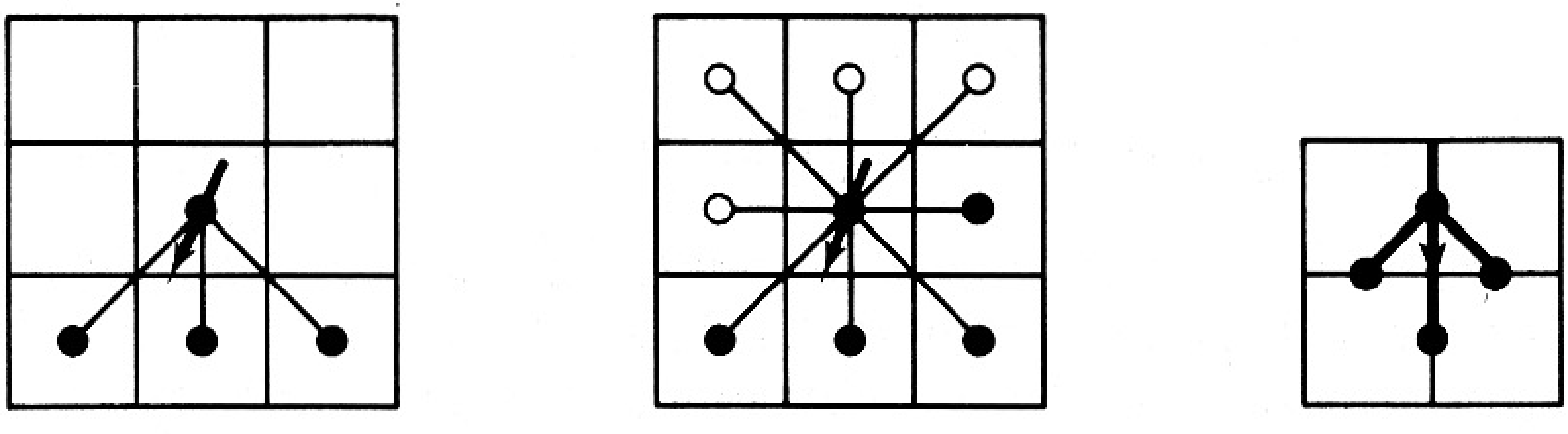


Snake Model - Demo



https://www.youtube.com/watch?v=CeU_yZjdVqY

Heuristic search of baundaries - example



(a) (b) (c)

1. Scan the stroke (edge) array for the most prominent edge.
2. Search in front of the edge until no more successors exist (i.e., a gap is encountered).
3. Search behind the edge until no more predecessors exist.
4. If the bidirectional search generates a path of 3 or more strokes, the path is a streak. Store it in a streak list and go to step 1.

Considering edge properties in the context Edge relaxation

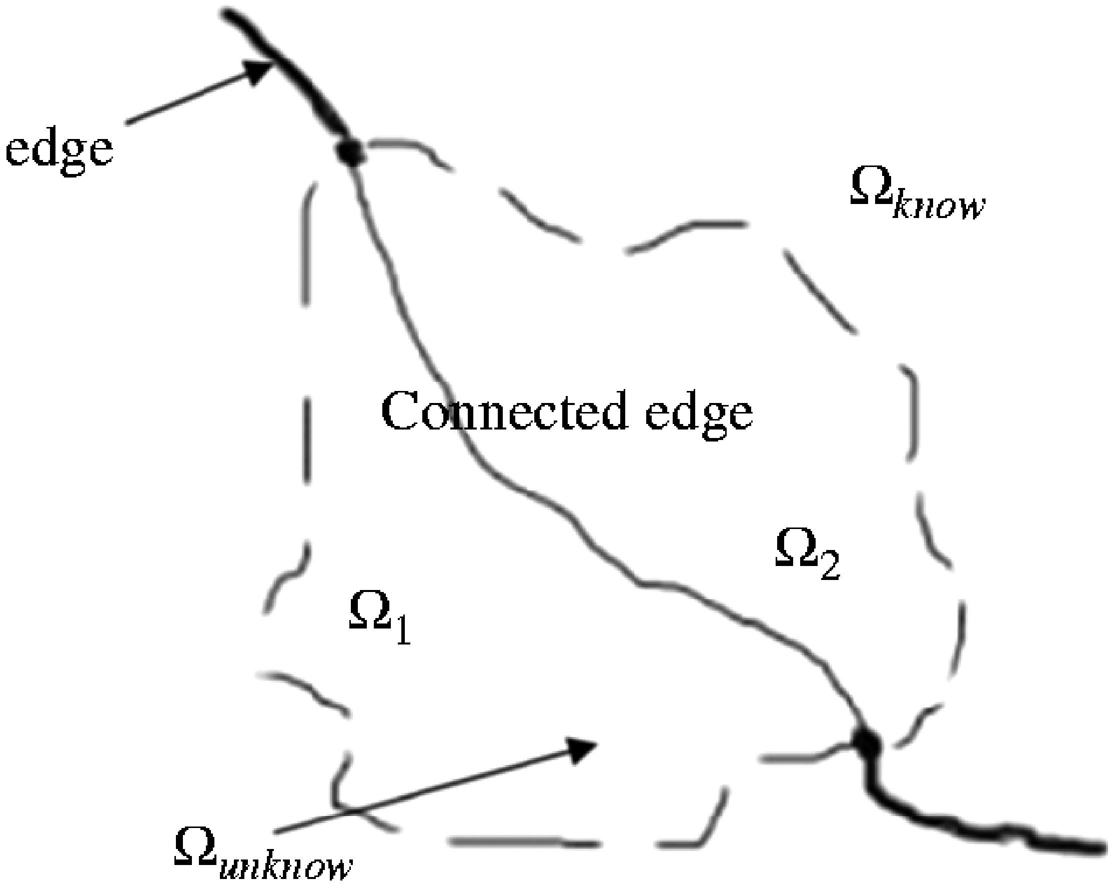
- Borders are strongly affected by image noise,
  often missing important parts.

- All the image properties, are iteratively evaluated with more precision until the edge context is totally clear
  - based on the strength of edges in a specified local neighbourhood, the confidence of each edge is either increased or decreased.

- A weak edge positioned between two strong edges provides an example of context;
  - it is highly probable that this inter-positioned weak edge should be a part of a resulting boundary.

- If, on the other hand, an edge (even a strong one) is positioned by itself with no supporting context,
  - it is probably not a part of any border.

Edge inpainting, edge following…

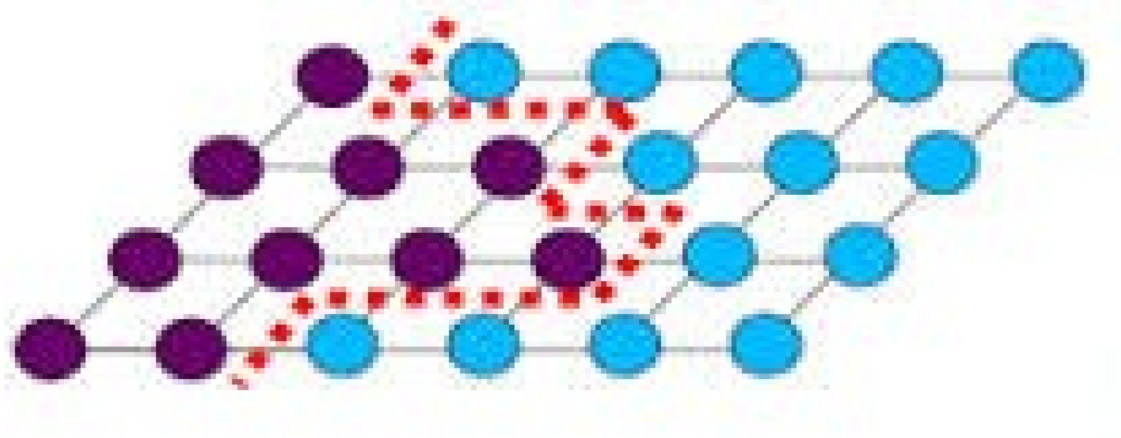


Graph-cut

Graph-cut

Graph-cut is an algorithm that finds a globally optimal segmentation solution.

- Min-cut
- Normalized cut

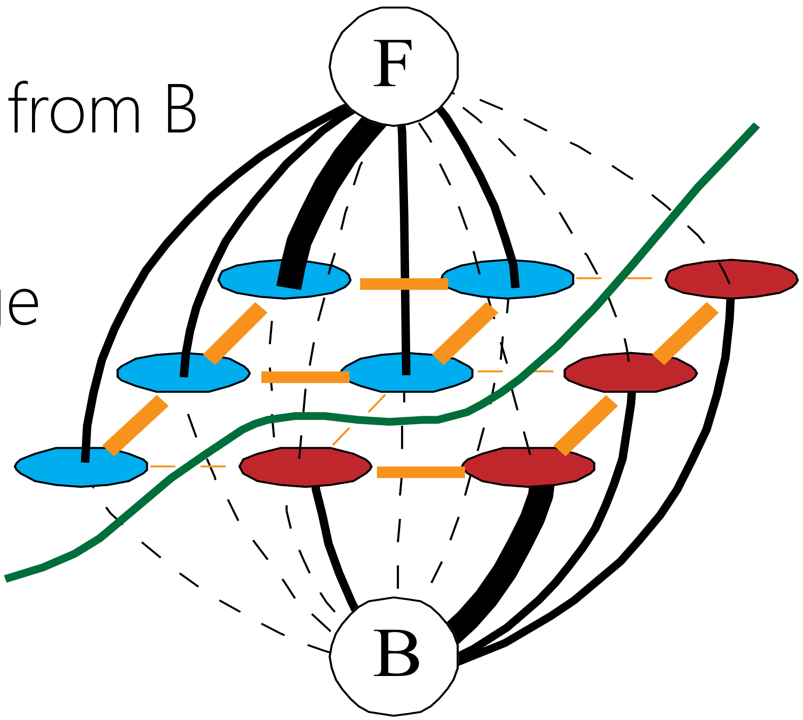


Graph min cut

Energy optimization equivalent to graph min cut

Cut: remove edges to disconnect F from B

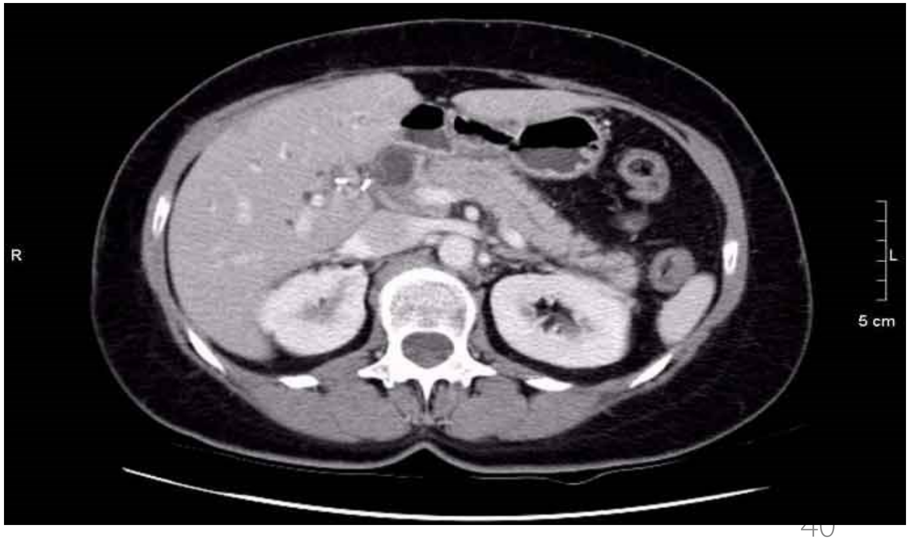
Minimum: minimize sum of cut edge weight



Interactive Segmentation using Graph Cuts

Motivation: for some applications: Fully automatic segmentation does not give sufficient results.

for example, in medical diagnostics, Doctors handlings a medical scan images where soft tissues, blood vessels, and bones are “blended” together , are very difficult to segment.



Interactive Segmentation application using Graph Cuts

User marks some pixels as seeds of “Background” and “Object”, and runs the application several times until a satisfactory result achieved.

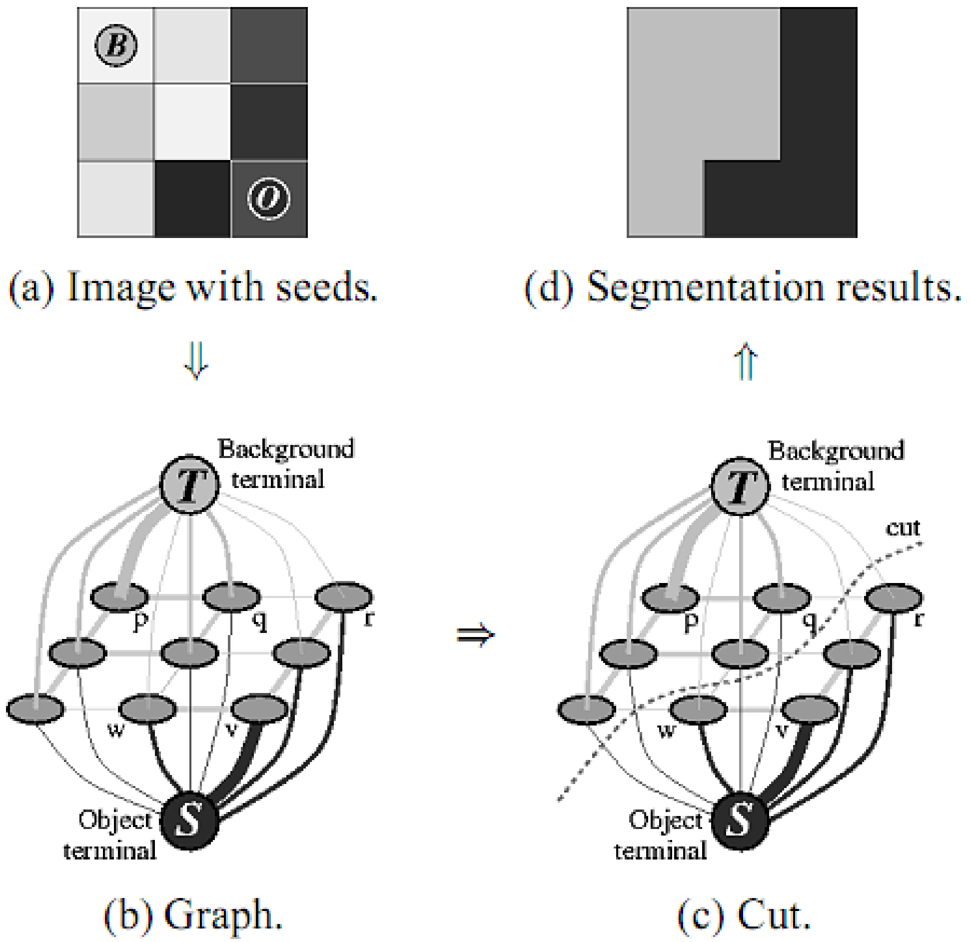
Interactive inputs:

“seeds” pixels

the ratio between regional importance VS boundary importance

Energy function (Intensities, distances, textures, etc..)

Seeds in the flow graph



(a) Image with seeds.
(d) Segmentation results.
(b) Graph.
(c) Cut.

Region based methods

Region based methods

Segmentation may be regarded as spatial clustering:

- clustering in the sense that pixels with similar values are grouped together

  and

- spatial in that pixels in the same category also form a single connected component.

Growing regions

Preprocessing is necessary!

Growing regions

– Select a start (seed) point

– Grow the point based on a certain property of homogeneity (Connectivity should be considered)

– Seed point selection:

for example - highlighted point (Due to specific property)

(openCv: FloodFill)

An example of growing regions

a b
c d

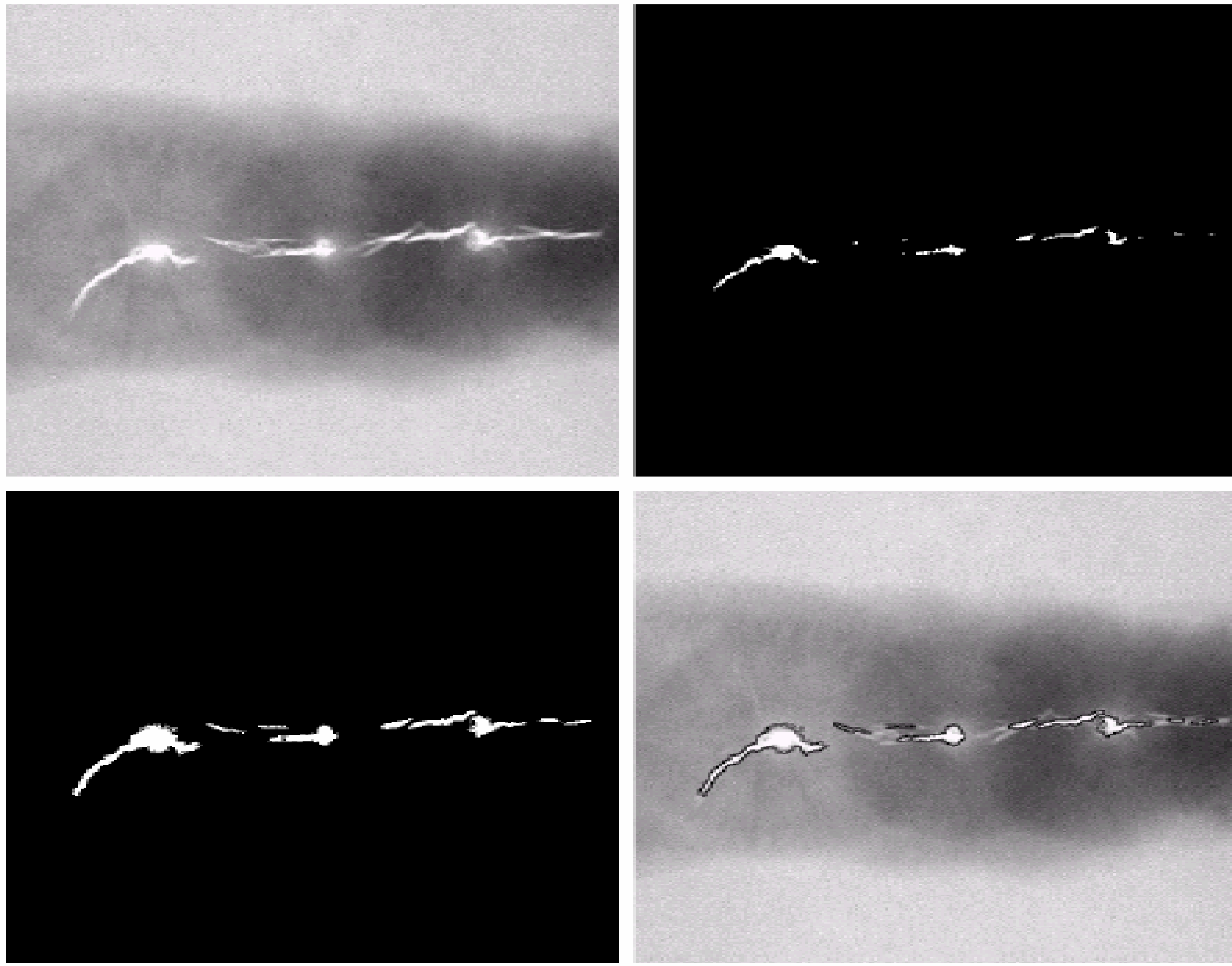
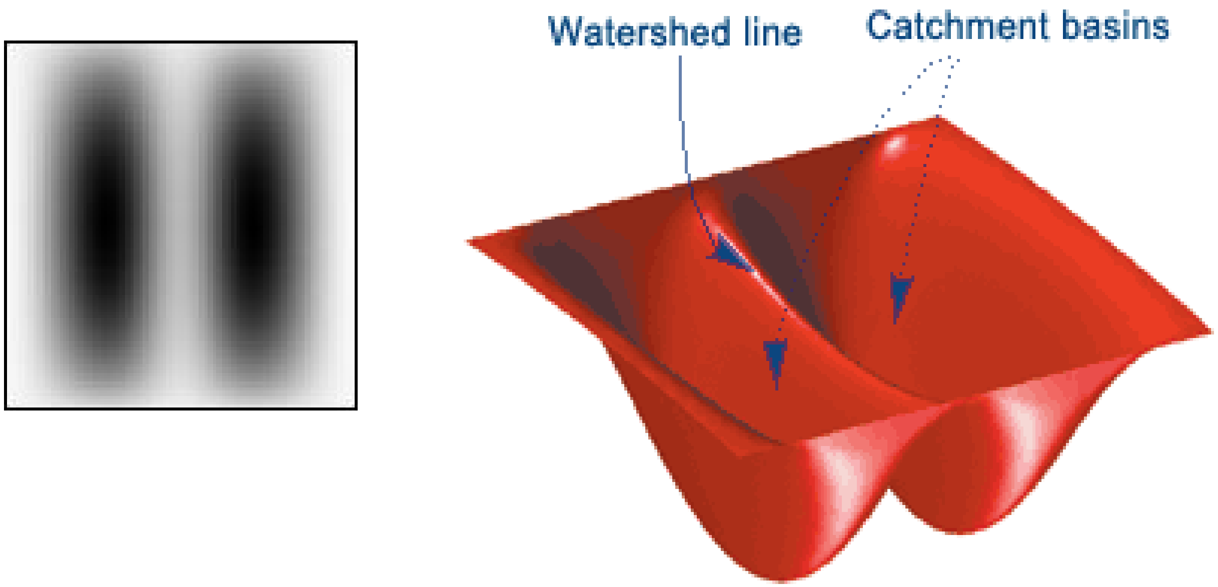


FIGURE 10.40 (a) Image showing defective welds. (b) Seed points. (c) Result of region growing. (d) Boundaries of segmented defective welds (in black). (Original image courtesy of X-TEK Systems, Ltd.).

Cor

Region based method- Watershed Segmentation

The watershed transform treats the intensity as a function defining 'hills' and 'valleys' and attempts to detect the valleys.



Segmentation by clustering

Clustering - unsupervised classification

Consider segmentation as a classification problem

An image consists of K regions.

Each pixel have to be classified into the one of this regions.

K-Means algorithms

Mean Shift algorithms

Fuzzy C-Means

Gaussian Mixture Models

and much more...

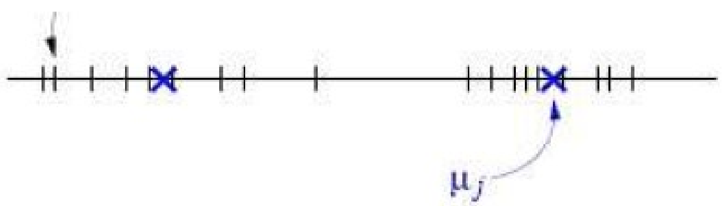
K-Means Clustering

K-Means Algorithm

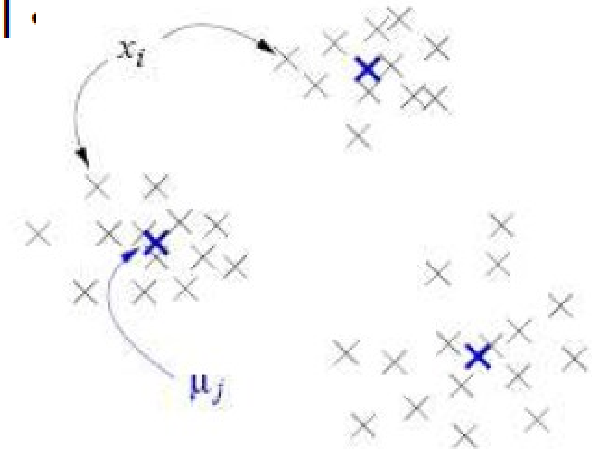
Each region is represented by a mean value – μj.

Minimum distance classification

$$O = \sum_{k=1}^{K} \sum_{\mathbf{x} \in C_k} \|\mathbf{x} - \boldsymbol{\mu}_k\|$$



2D



Počítačové videnie - Detekcia a Rozpoznávanie Objektov.pdf str.228

K-means Algorithm

D-dimension feature vectors xi.

The criterion is the minimum distance from the center of clusters.

$$H(y_1,\ldots,y_N,\boldsymbol{\mu}_1,\ldots,\boldsymbol{\mu}_k) = \sum_{i=1}^{N} \left\| \mathbf{x}_i - \boldsymbol{\mu}_{y_i} \right\|^2$$

$y_i \in \{1, \ldots, k\}$ … assignment to clusters..

Dividing of the data into compact clusters.

K-Means Algorithm

1.Randomly initialize centres of regions μ_i.

2. Classify each vector xi to the nearest cluster:

$$y_i = \arg\min_{k \in \{1,..,K\}} \left\| x_i - \mu_k \right\|^2$$

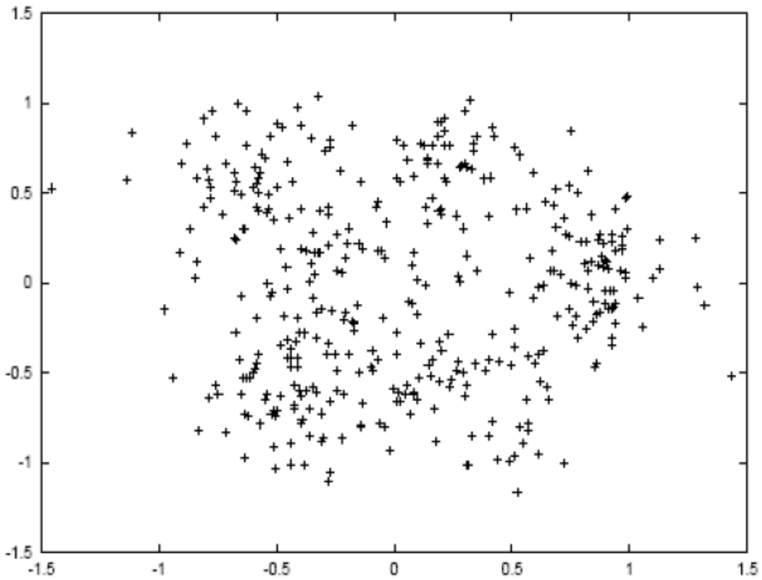
3.Calculate a new centres of regions

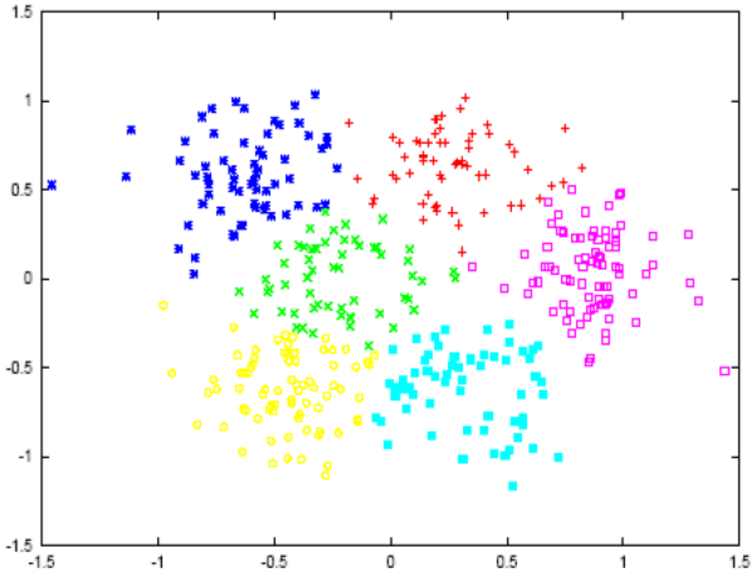
$$\mu'_k = \frac{1}{N_{y_i = k}} \sum_{i: y_i = k} x_i$$

Repeat steps (2) and (3) until the center of the cluster changes.

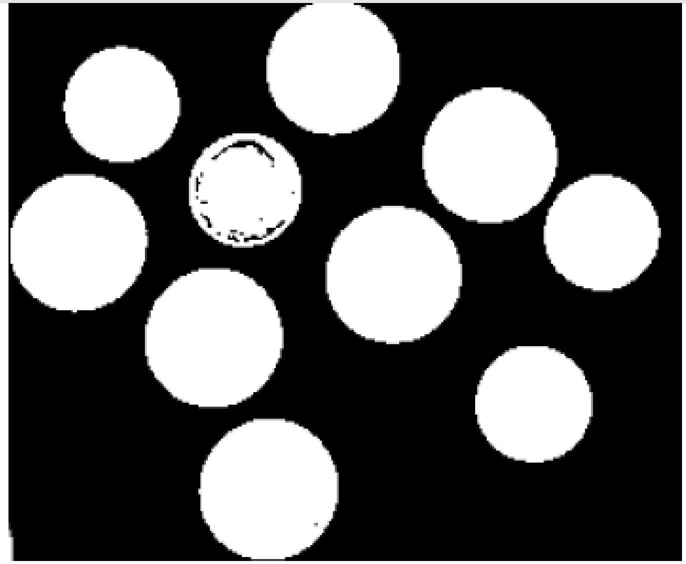
K-means Algorithm

Dividing of the data into compact clusters.

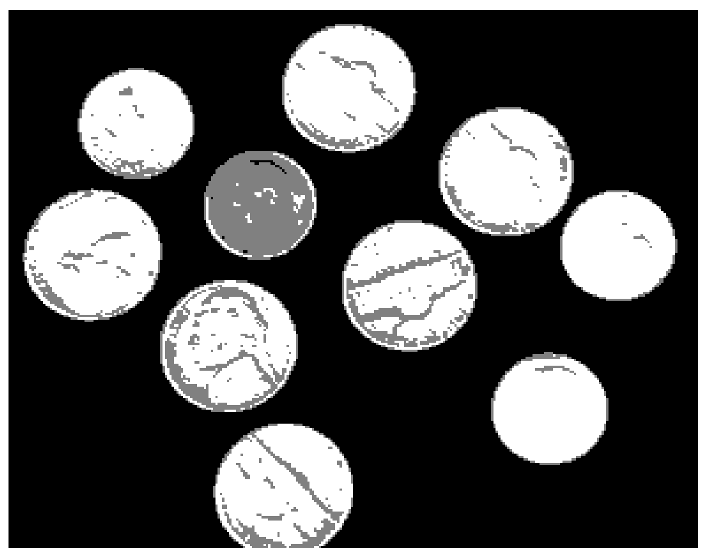




Example of K-Means Algorithm



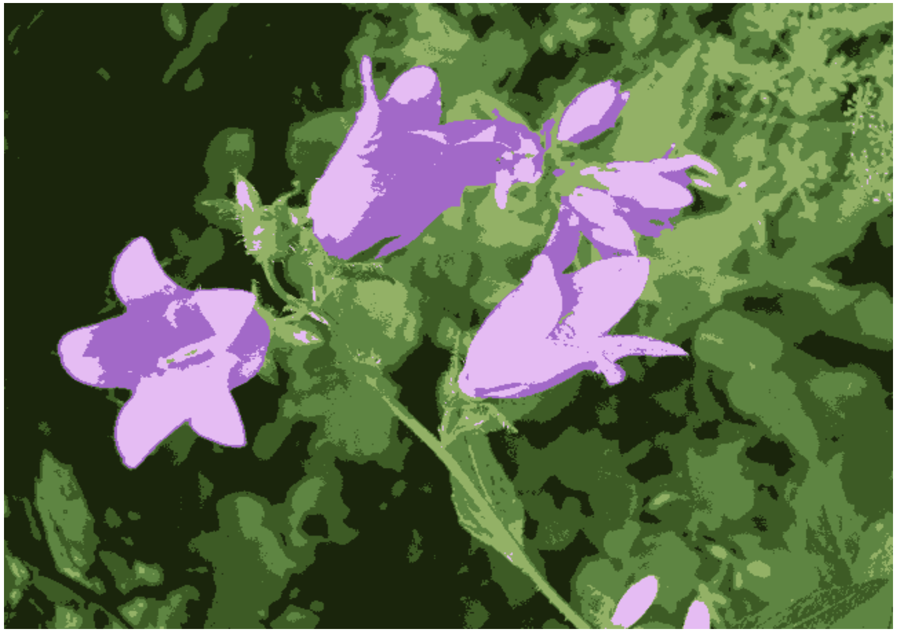
K=2



K=3

K-Means Clustering

K=6 in 3-Dim feature space: RGB color space:

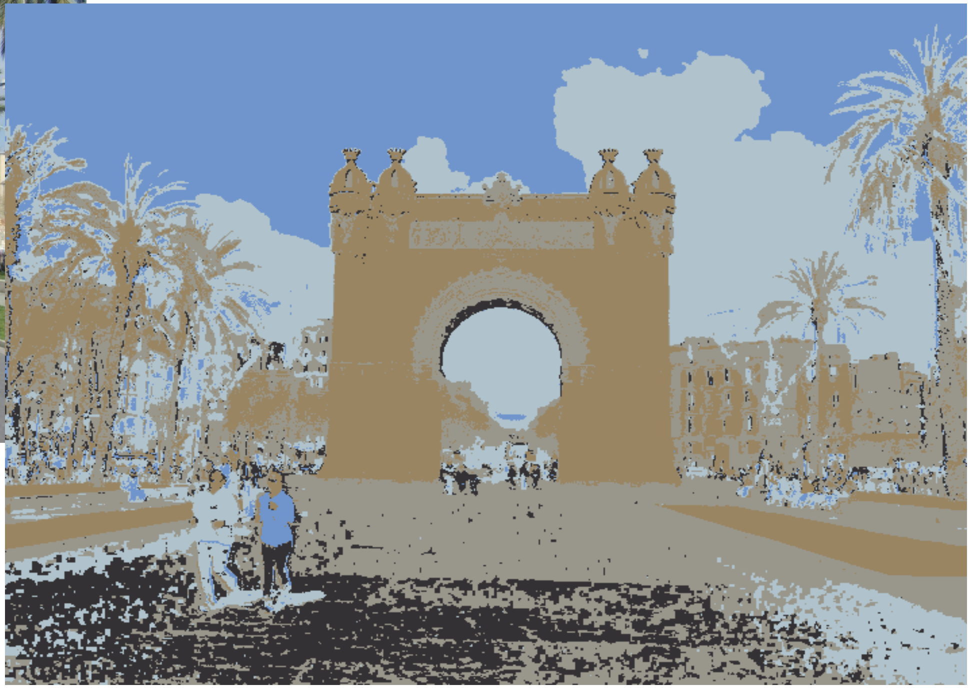


K-Means

K=5 in 3-Dim feature space: RGB color space:

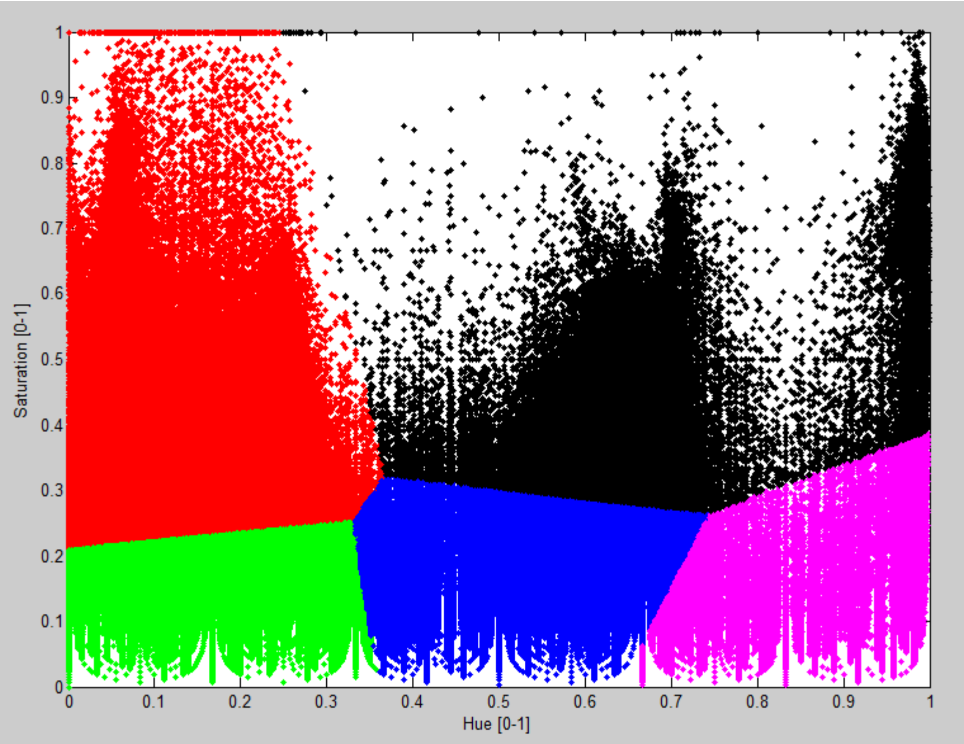
K-Means

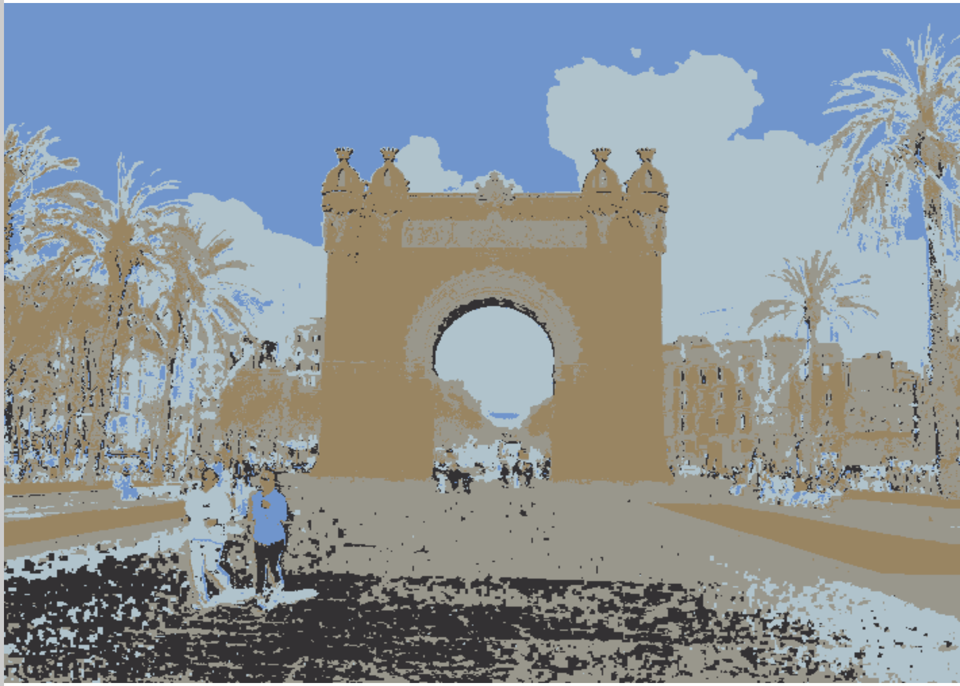
K=5 in 2-Dim feature space: Hue, Saturation



K-Means - Visualisation of the 2Dim feature space

K=5 in 2-Dim feature space: Hue, Saturation





K-Means Algorithm

If the K-means algorithm use only 1 feature (lightness), than is this operation only a modification of the thresholding method

extended feature vector:

Colour information

Position of the pixel

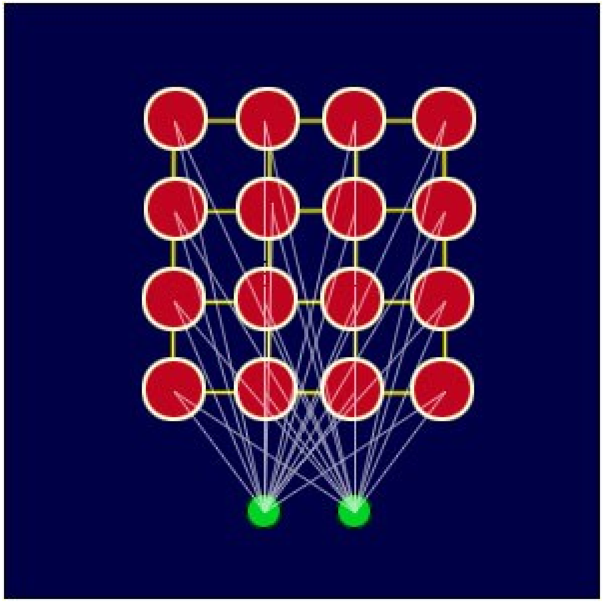
Texture description

etc.

Kohonen's Self Organizing Feature Maps

A self-organizing map (SOM) or self-organizing feature map (SOFM) is a type of artificial neural network (ANN) that is trained using unsupervised learning

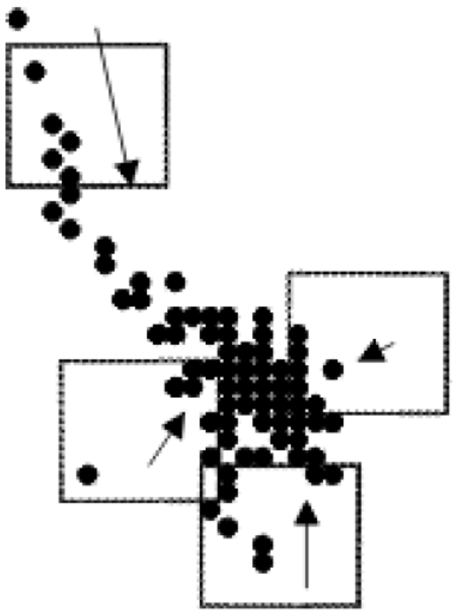
...more in detail in the next lectures...



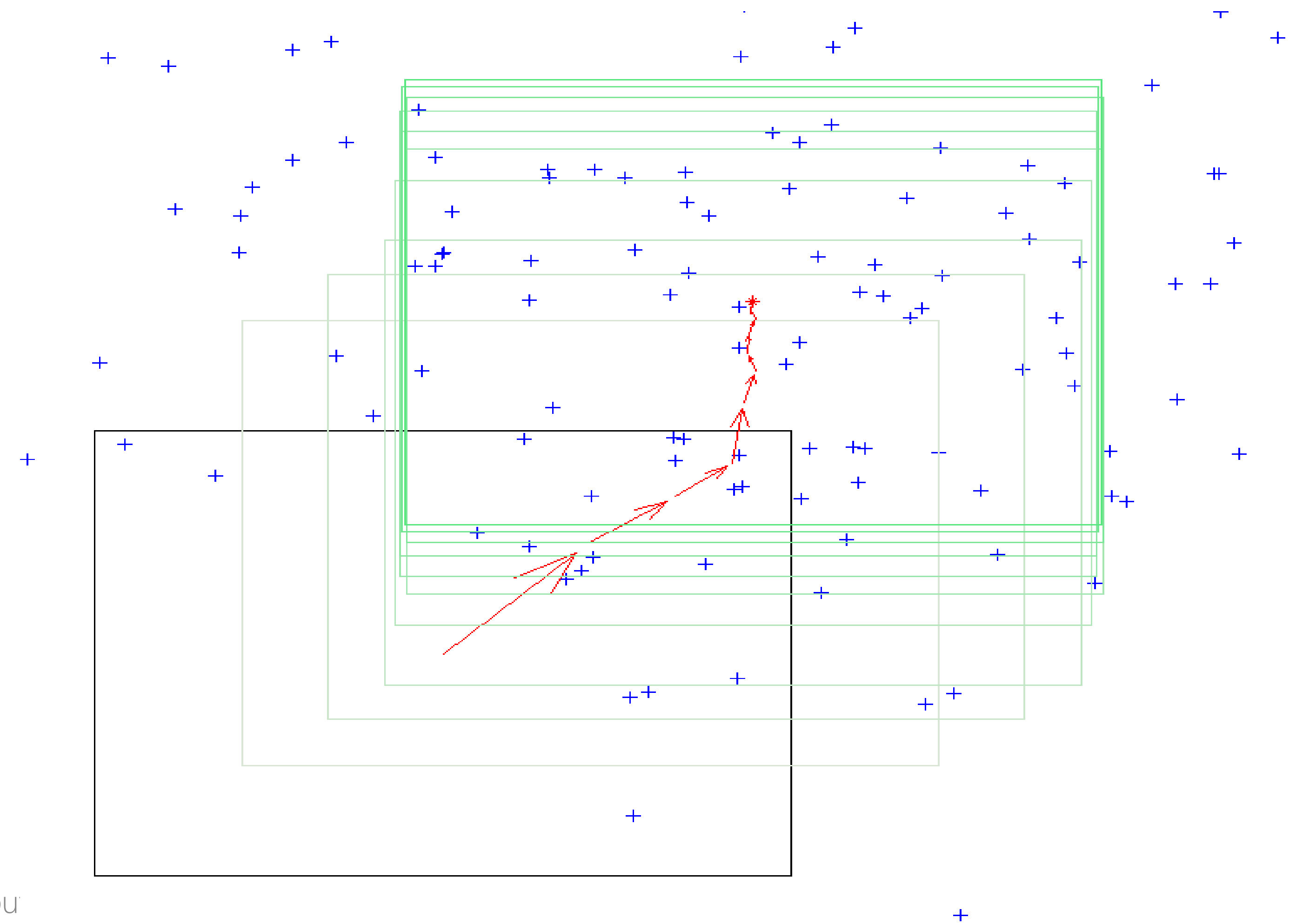
Mean Shift Algorithm

Mean Shift Algorithm

1. Choose a search window size.
2. Choose the initial location of the search window.
3. Compute the mean location (centroid of the data) in the search window.
4. Center the search window at the mean location computed in Step 3.
5. Repeat Steps 3 and 4 until convergence.



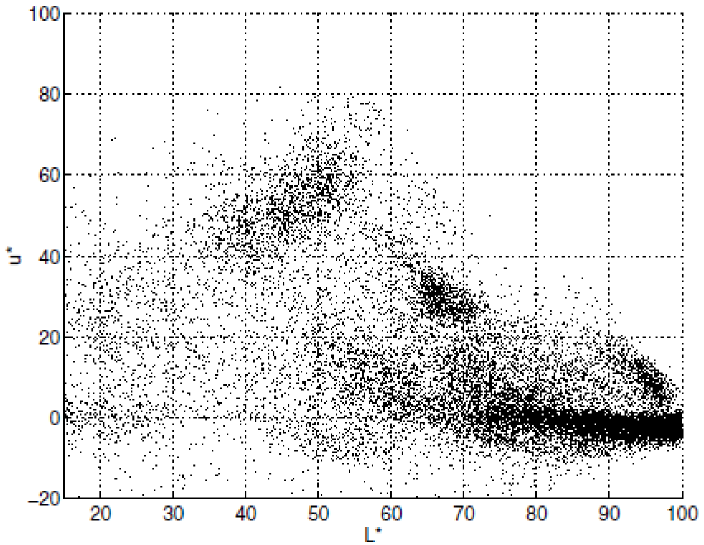
Mean Shift Algorithm

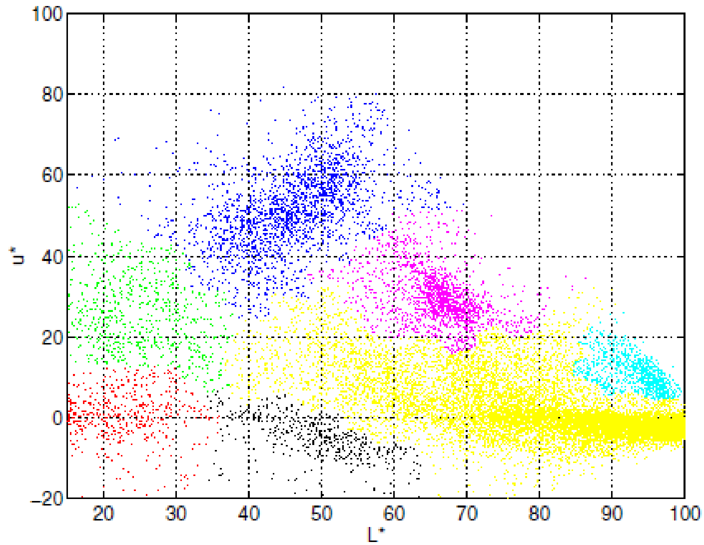


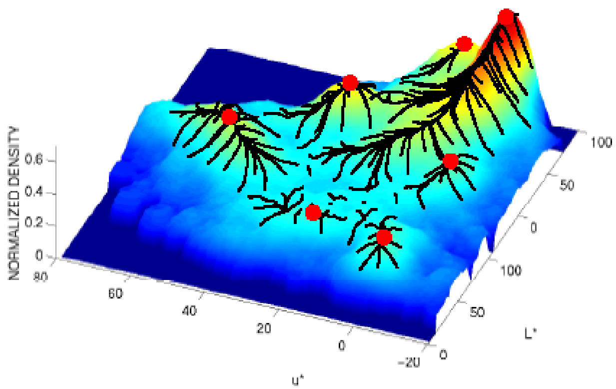
Mean Shift Segmentation Algorithm

1. Convert the image into the features (color, gradients, texture measures etc).
2. Choose initial search window locations uniformly in the data.
3. Compute the mean shift window location for each initial position.
4. Merge windows that end up on the same "peak" or mode.
5. The data these merged windows traversed are clustered together.

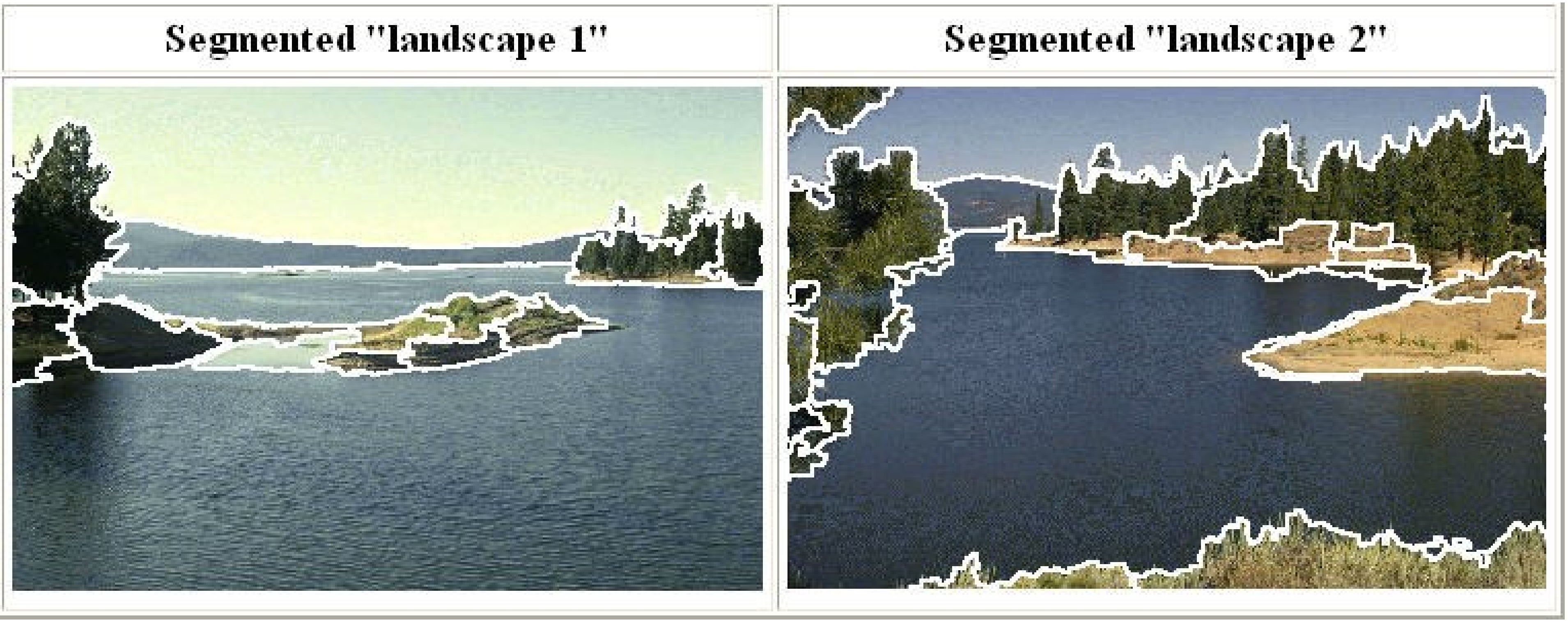
Mean Shift Algorithm (Features: L*u*)







Mean shift segmentation example



Split & Merge Algorithm

Split & Merge Algorithm
Step 1 – SPLIT

Recursively splits an image region (which starts out as the entire image) by dividing it into quarters. Feature vectors are computed for each block.

If the four blocks are judged to be similar, they are grouped back together and the process ends.

If not, each of the blocks are recursively divided and analyzed using the same procedure.

Split & Merge Algorithm

When we are finished, there will be many different sized blocks, each of which has a homogeneous texture according to our model.

However, the arbitrary division of the image into rectangles (called a „quad-tree'' decomposition) might have accidentally split up regions of homogeneous texture.

The merge step, then, tries to fix this by looking at adjacent regions that were not compared to each other during the split phase, and merging recursively splits an image region (which starts out as the entire image) by dividing it into quarters.

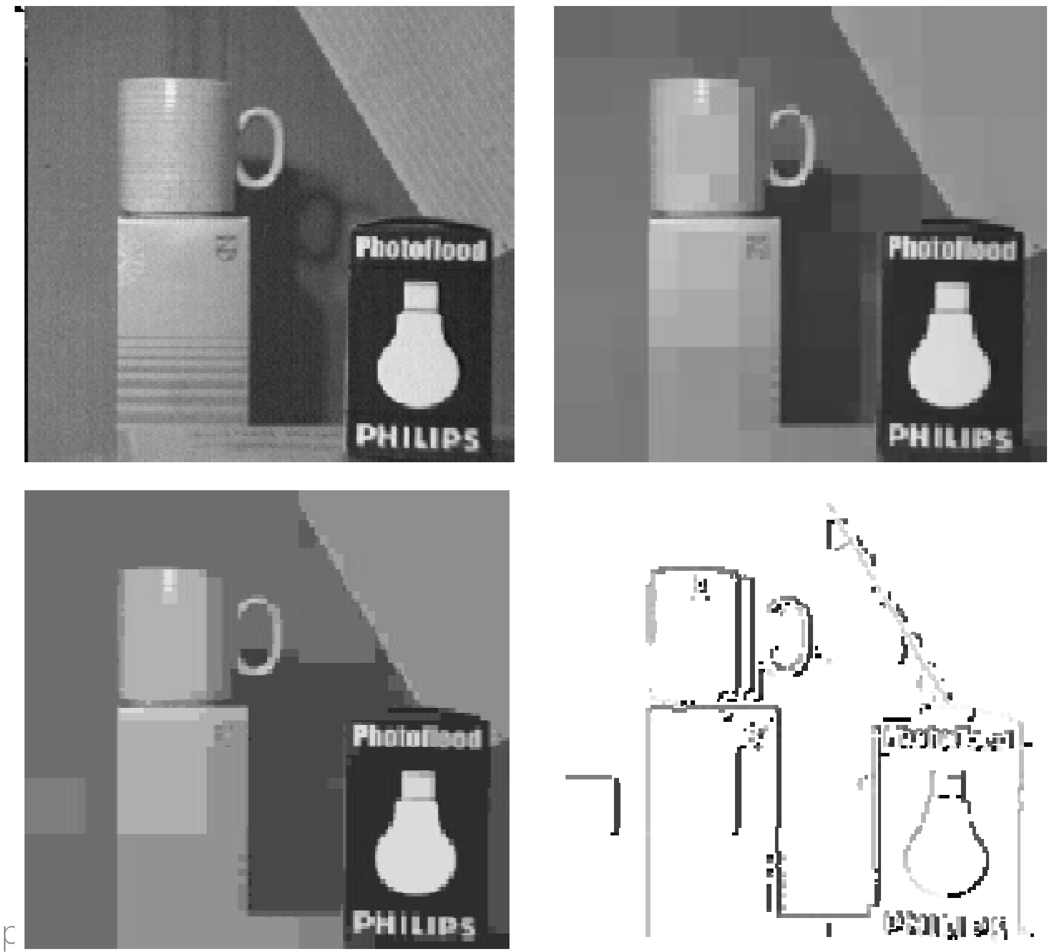
Split & Merge Algorithm
Step 2 - MERGE

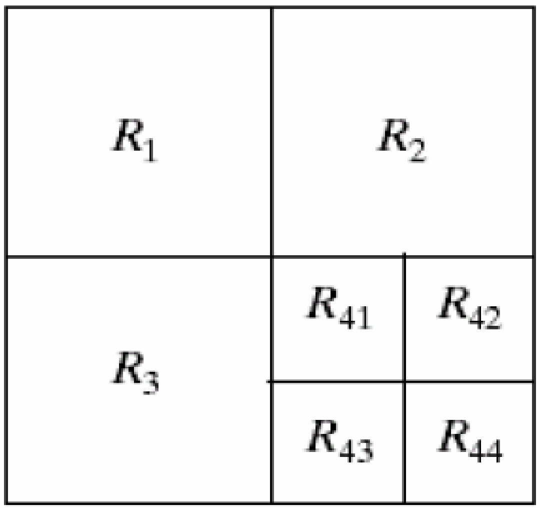
Feature vectors are computed for each block.

If the four blocks are judged to be similar, they are grouped back together and the process ends.

If not, each of the blocks are recursively divided and analyzed using the same procedure.

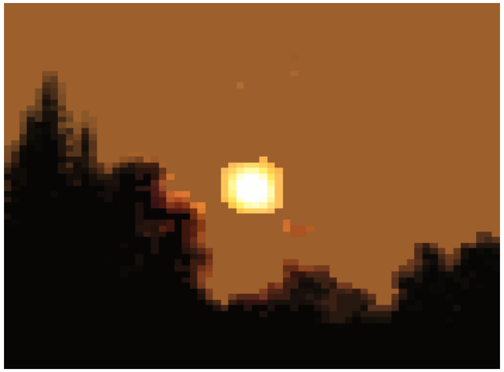
Split & Merge -Example





Split & Merge -Examples



Min size10x10 pixelov

Hough Transforms

Hough Transforms

The Hough transform is a method for finding lines, circles, or other simple forms in an image.

The purpose of the technique is to find instances of objects within a certain class of shapes by a **voting** procedure.

This **voting** procedure is carried out in a parameter space – accumulator.

Hough Line Transform

The underlying principle of the Hough transform is that there are an infinite number of potential lines that pass through any point, each at a different orientation.

The purpose of the transform is to determine which of these theoretical lines pass through most *points* in an image - that is, which lines fit most closely to the data in the image

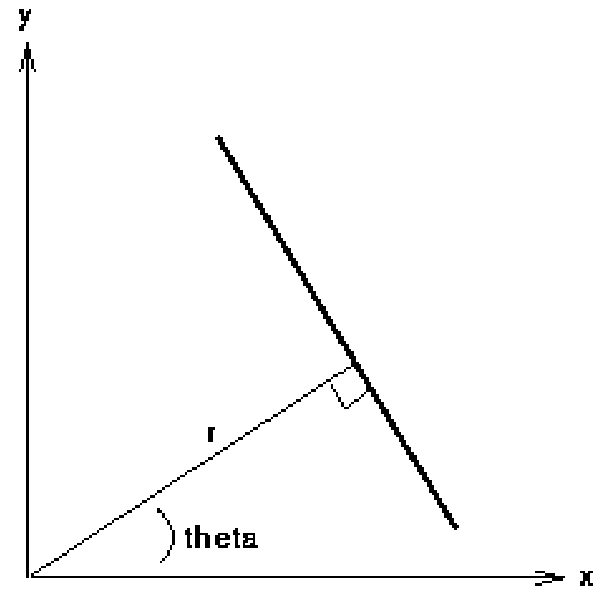
Hough Transform

In the standard Hough transform, each line is represented by two parameters, commonly called *r* and θ (*theta*), which represent the length and angle from the origin of a normal to the line.

By transforming all the possible lines through a point into this coordinate system - i.e. calculating the value of *r* for every possible value of θ

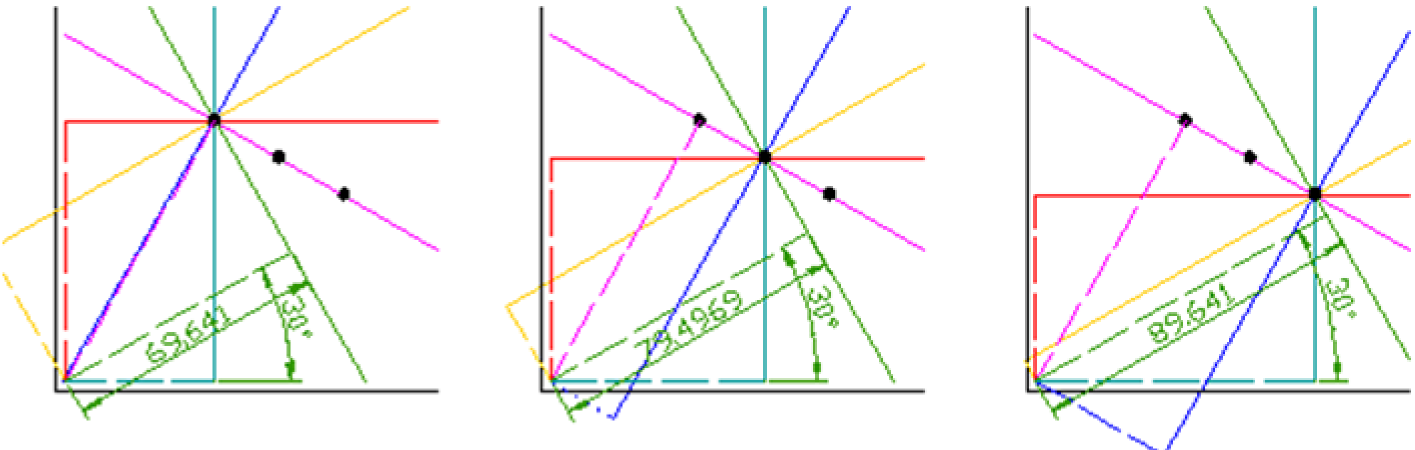
This representation of the two parameters is sometimes referred to as *Hough space* .

$r = x.\cos(\theta) + y.\sin(\theta)$



Hough Line Transform

Consider three data points, shown here as black dots.



Angle	Dist.
0	40
30	69.6
60	80.5
90	70
120	40.6
150	0.4

Angle	Dist.
0	57.1
30	79.5
60	80.5
90	60
120	23.4
150	−19.5

Angle	Dist.
0	74.6
30	89.6
60	80.5
90	50
120	6.0
150	−39.6

Hough Line Transform

For each data point, a number of lines are plotted going through it, all at different angles. These are shown here as solid lines.

For each solid line a line is plotted which is perpendicular to it and which intersects the origin. These are shown as dashed lines.

The length and angle of each dashed line is measured. In the diagram above, the results are shown in tables.

This is repeated for each data point.

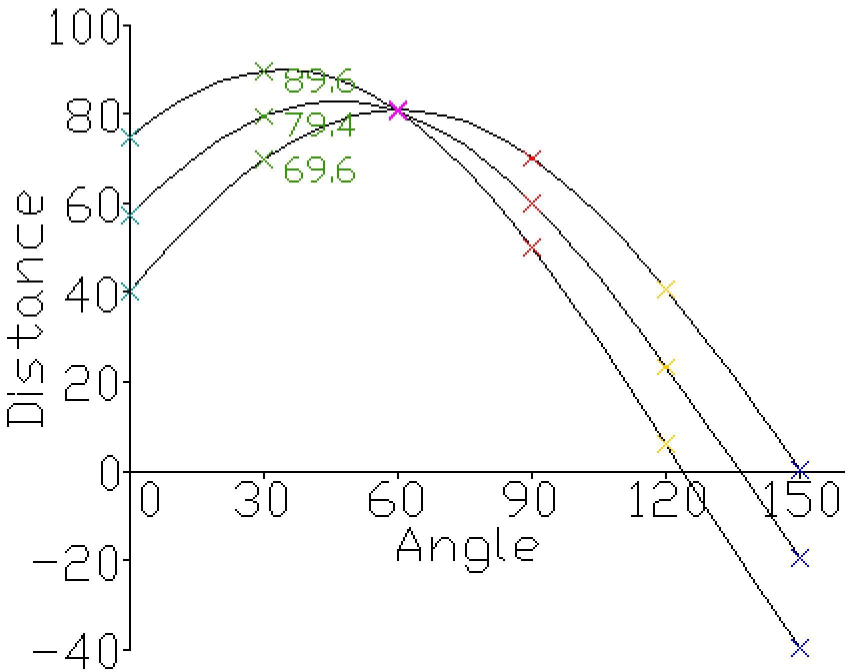
A graph of length against angle, known as a Hough space graph, is then created.

Hough Line Transform

The point where the lines intersect gives a distance and angle. This distance and angle indicate the line which bisects the points being tested.

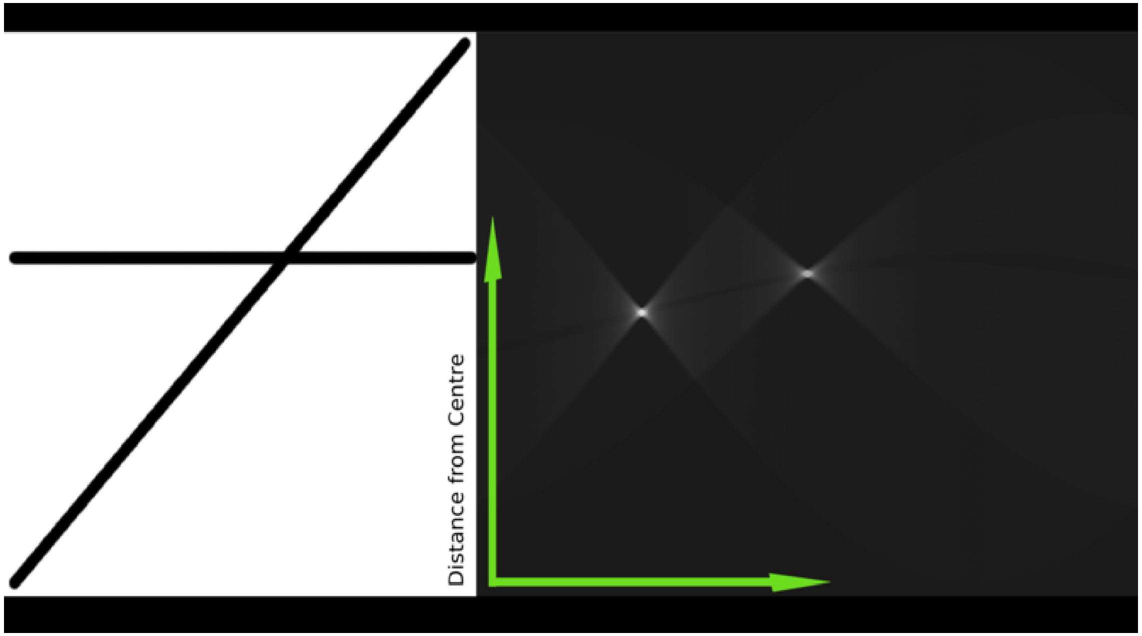
In the graph shown the lines intersect at the purple point; this corresponds to the solid purple line in the diagrams above, which passes through all three points.

Hough space plot example:



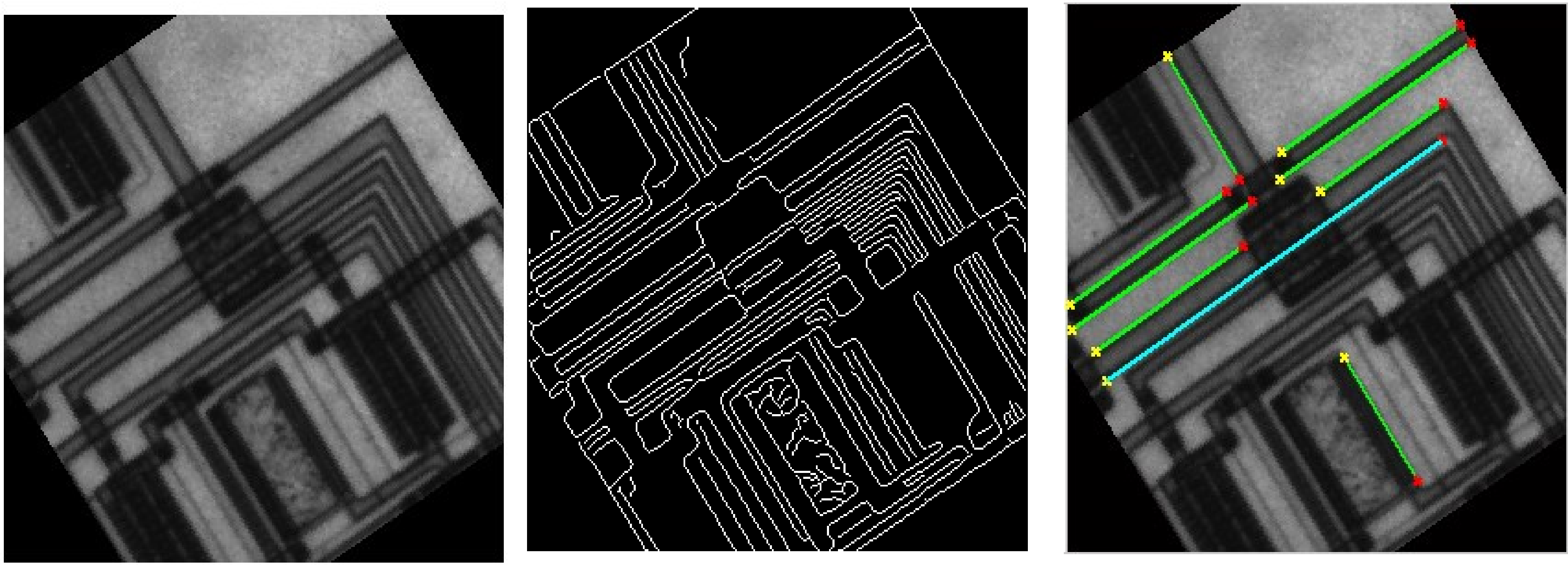
Example : raster image containing two thick lines

The results of this transform were stored in a matrix. Cell value represents the number of curves through any point. Higher cell values are rendered brighter. The two distinctly bright spots are the Hough parameters of the two lines. From these spots' positions, angle and distance from image center of the two lines in the input image can be determined.



Hough Transform - example

Highlight the longest line segments:



Hough transform of curves

A similar transform can be used for finding any shape which can be represented by a set of parameters.

A circle, for instance, can be transformed into a set of three parameters, representing its centre and radius, so that the Hough space becomes three dimensional.

Detection of 3D objects (Planes and cylinders)

Hough transform can also be used for the detection of 3D objects in range data or 3D point clouds.

The extension of classical Hough transform for plane detection is quite straight forward. A plane is represented by its explicit equation

z = ax*x + ay*y + d

for which we can use a 3D Hough space corresponding to ax, ay and d.

This formulation of the plane has been used for the detection of planes in the point clouds acquired from airborne laser scanning and works very well because in that domain all planes are nearly horizontal.

Detection of 3D objects (Planes and cylinders)

For generalized plane detection using Hough transform, the plane can be parametrized by its normal vector n (using spherical coordinates) and its distance from the origin ρ resulting in a three dimensional Hough space.

This results in each point in the input data voting for a surface in the Hough space. The intersection of these surfaces indicates presence of a plane.

Hough transform has also been used to find cylindrical objects in point clouds using a two step approach. The first step finds the orientation of the cylinder and the second step finds the position and radius.

Hough Transform

A classical Hough transform is most commonly used for the detection of regular curves such as lines, circles, ellipses, *etc.*

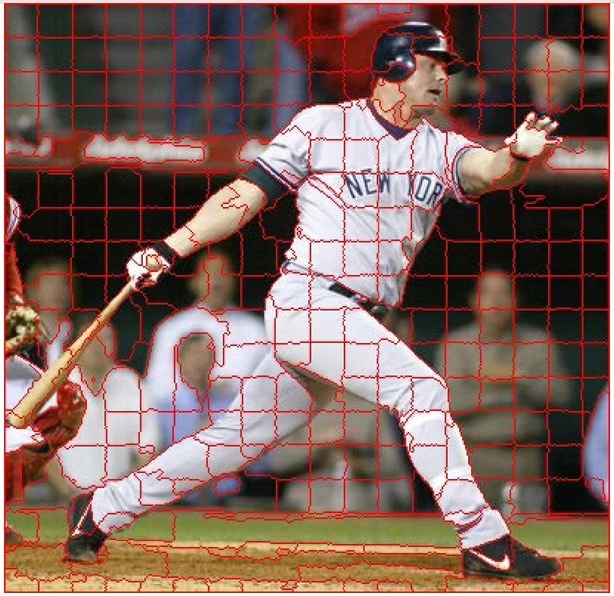
A *generalized* Hough transform can be employed in applications where a simple analytic description of a feature(s) is not possible

http://homepages.inf.ed.ac.uk/rbf/HIPR2/hough.htm

Superpixel segmentation

Superpixel segmentation

- results of an image oversegmentation
- regions in an image which can be used
  as basic units (primitives) in the
  next image processing



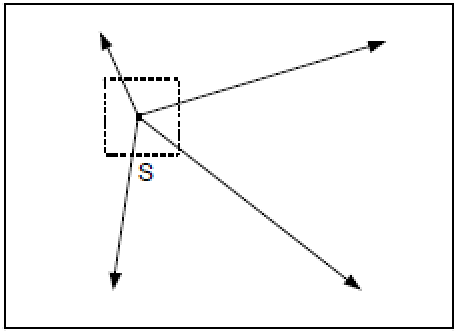
- distributed regularly with respect to the nature of the input image,
- the desirable variation of the size of superpixels is preferably small
- the boundary of superpixels has to be corresponding with the natural boundary of objects presented on the image

Simple Linear Iterative Clustering - SLIC

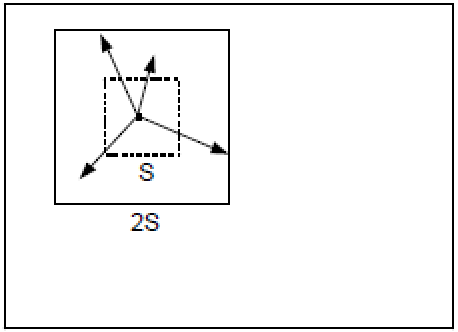
- color images in the CIELAB color space,
- the clustering procedure begins with an initialization step where k initial cluster centers Ci are sampled on a regular grid spaced S pixels apart.
- The centers are moved to seed locations corresponding to the lowest gradient position in a 3 × 3 neighborhood. (This is done to avoid centering a superpixel on an edge, and to reduce the chance of seeding a superpixel with a noisy pixel.)
- Ci = $[l \; a \; b \; x \; y]^{\hat{T}}$

Simple Linear Iterative Clustering - SLIC

- each pixel i is associated with the nearest cluster center whose search region overlaps its location



(a) standard *k*-means searches the entire image

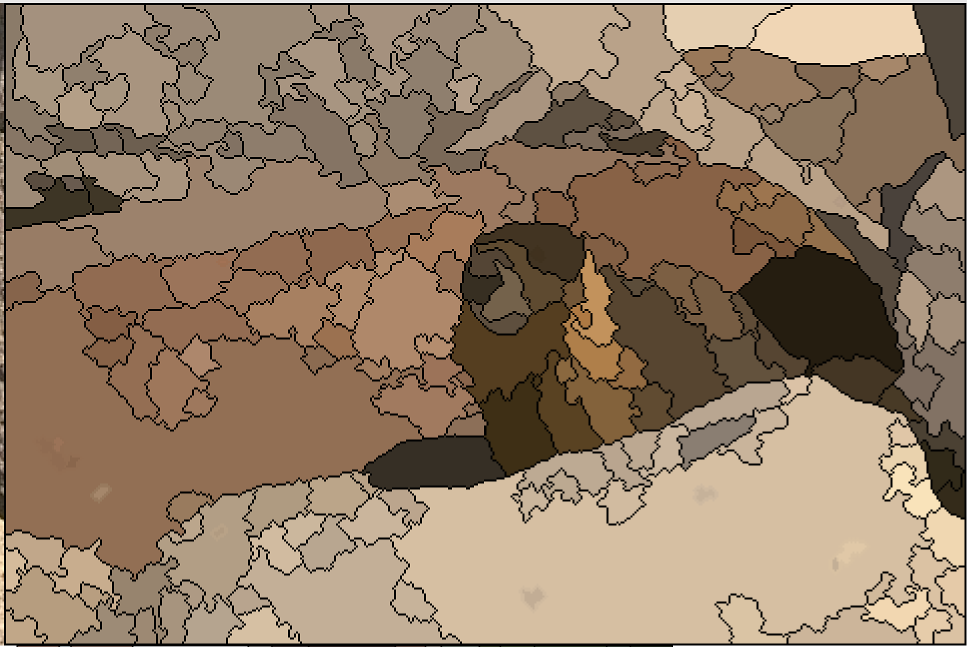


(b) SLIC searches a limited region

Simple Linear Iterative Clustering - SLIC

- Once each pixel has been associated to the nearest cluster center, an update step adjusts the cluster centers to be the mean [*L*a* b* x y*] vector of all the pixels belonging to the cluster.
- The *Euclidean distance* is used to compute a residual error *E* between the new cluster center locations and previous cluster center locations.
- The assignment and update steps can be repeated iteratively until the error converges, (10 iterations suffices for most images)
- Post-processing step enforces connectivity by re-assigning disjoint pixels to nearby superpixels – enforce connectivity.

Merge superpixels



Match-based segmentation
Template segmentation

Match-based segmentation
Template segmentation

Match-based segmentation :

Evaluate a match criterion for each location and rotation of the pattern in the image

Local maxima of this criterion exceeding a present threshold represent pattern location in the image

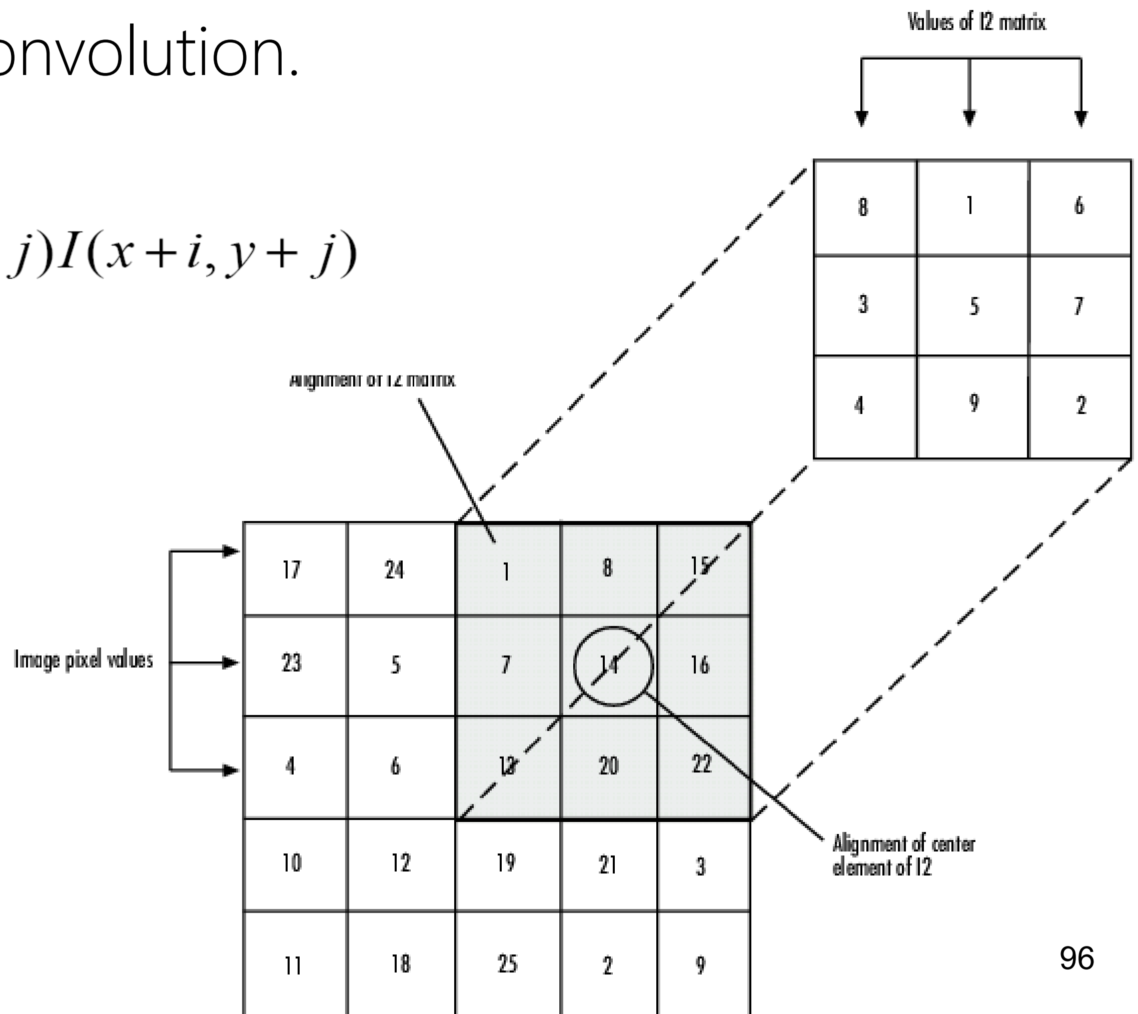
Matching criteria can be defined in many ways;

in particular, correlation between a pattern and the searched image data is a general matching criterion

Template matching
Correlation

The operation called correlation is closely related to convolution.

$$F \circ I(x,y) = \sum_{j=-N}^{N} \sum_{i=-N}^{N} F(i,j) I(x+i, y+j)$$



Template matching by correlation

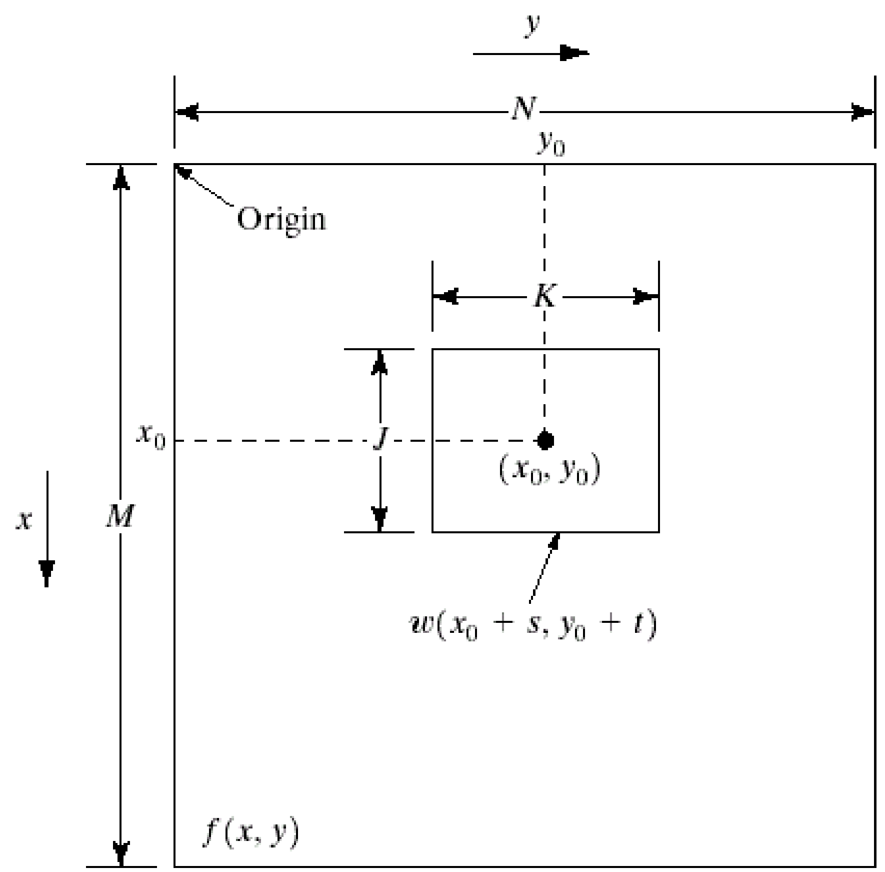
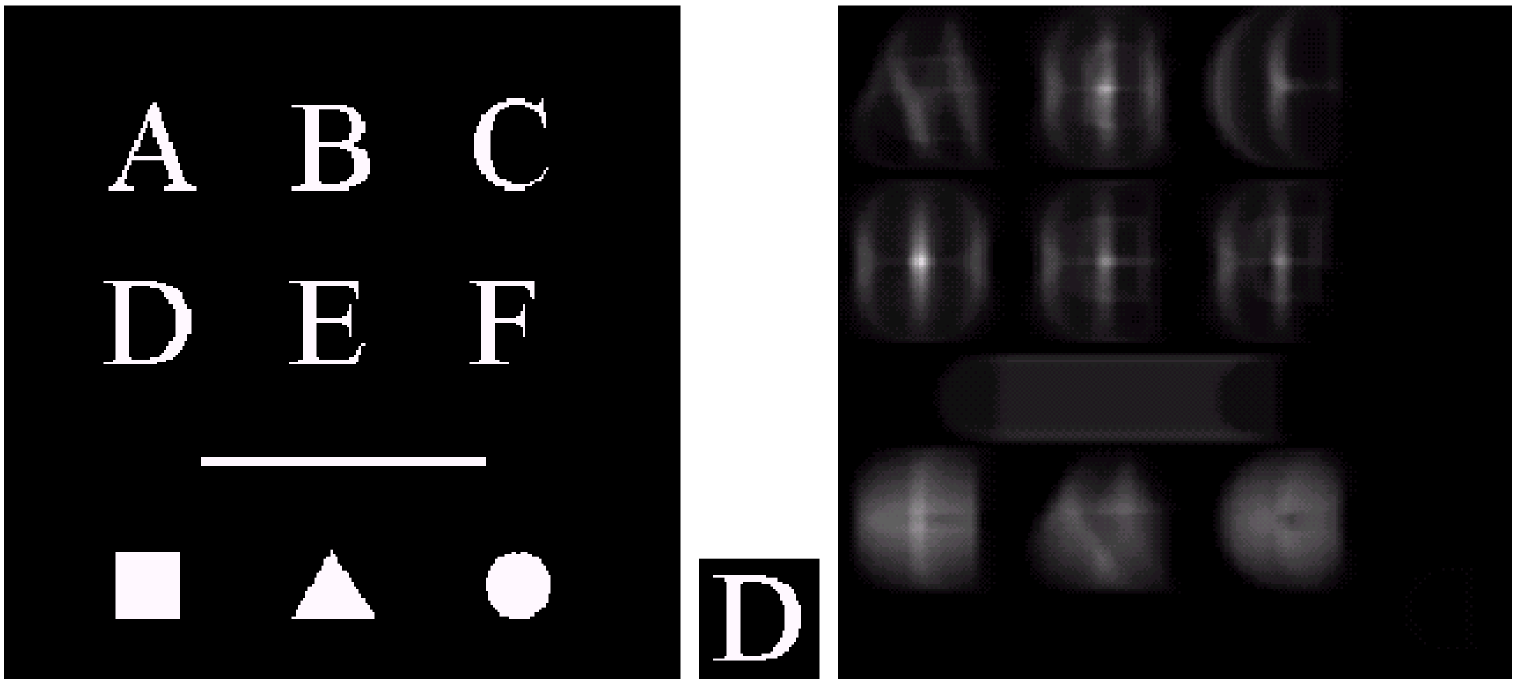


FIGURE 12.8 Arrangement for obtaining the correlation of f and w at point (x_0, y_0).

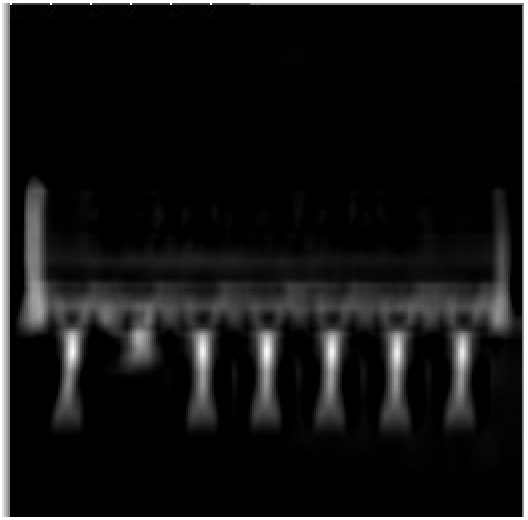
Demo of template matching using correlation



a b c

FIGURE 12.9
(a) Image.
(b) Subimage.
(c) Correlation coefficient of (a) and (b). Note that the highest (brighter) point in (c) occurs when subimage (b) is coincident with the letter "D" in (a).

Demo of template matching using correlation



http://bigwww.epfl.ch/demo/templatematching/tm_correlation/demo.html

Matching edge templates

Gradient-based representations: Matching edge templates

Input image | **Edges detected** | **Distance transform** | **Template shape** | **Best match**

At each window position, compute average min distance between points on template (T) and input (I).

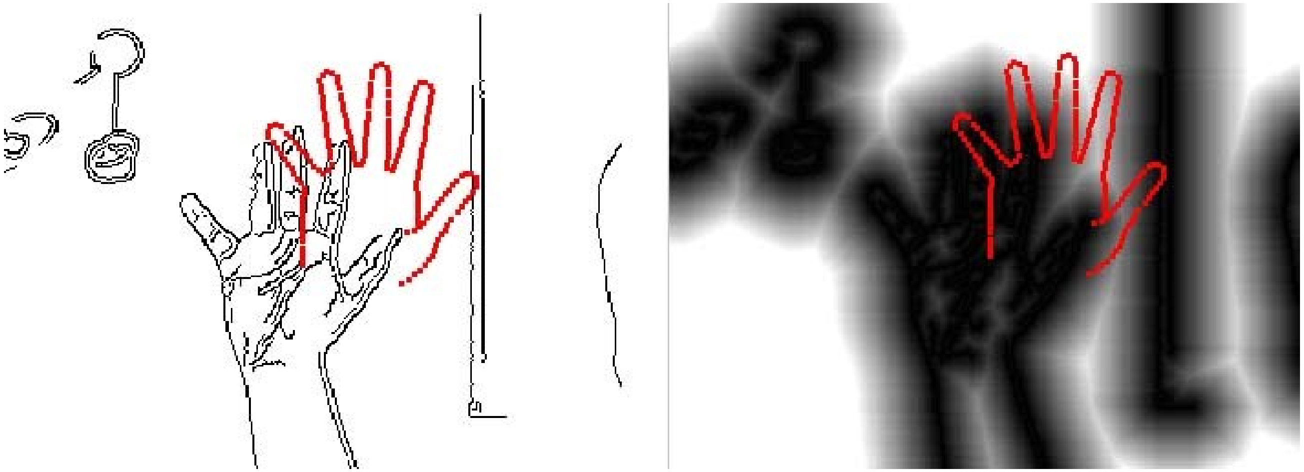
$$D_{chamfer}(T, I) \equiv \frac{1}{|T|} \sum_{t \in T} d_I(t)$$

Gavrila & Philomin ICCV 1999

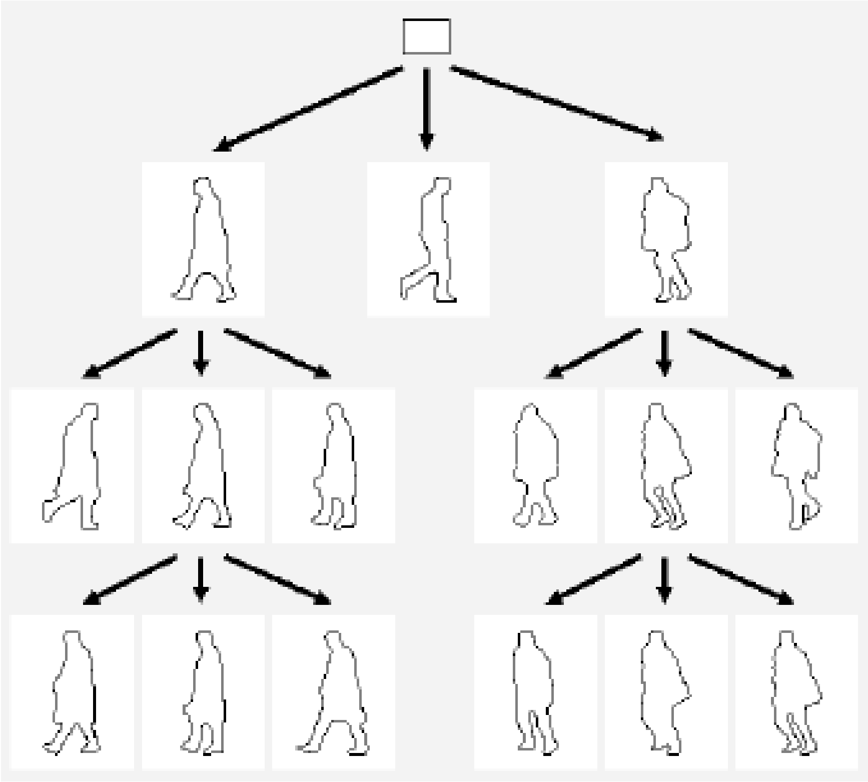
Example: Chamfer matching

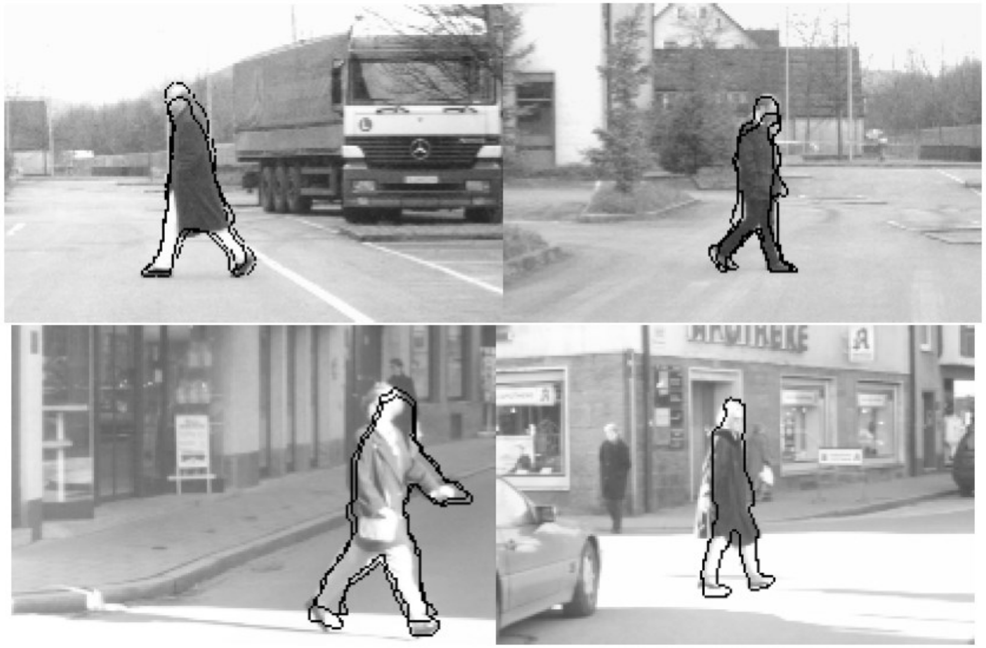
Chamfer Matching

- Chamfer score is average nearest distance from template points to image points
- Nearest distances are readily obtained from the distance image
- Computationally inexpensive



Matching edge templates





Hierarchy of templates

Features-based Segmentation

Other Features-based Segmentation

Frequency features

successful in the tasks of texture segmentation

PCA - Principal component analysis

....and more – will be presented in the next lessons about object detection

Segmentation in Video-sequences

Segmentation in Video-sequences

Foreground / Background detection

Background Subtraction

first must be “learned” a model of the background

this background model is compared against the current image and then the known background parts are subtracted away.

The objects left after subtraction are presumably new foreground objects.

Establishing a Reference Images

Establishing a Reference Images:

– Difference will erase static object:

• When a dynamic object move out completely from its position, the back ground in replaced.

a b c

FIGURE 10.50 Building a static reference image. (a) and (b) Two frames in a sequence. (c) Eastbound automobile subtracted from (a) and the background restored from the corresponding area in (b). (Jain and Jain.)