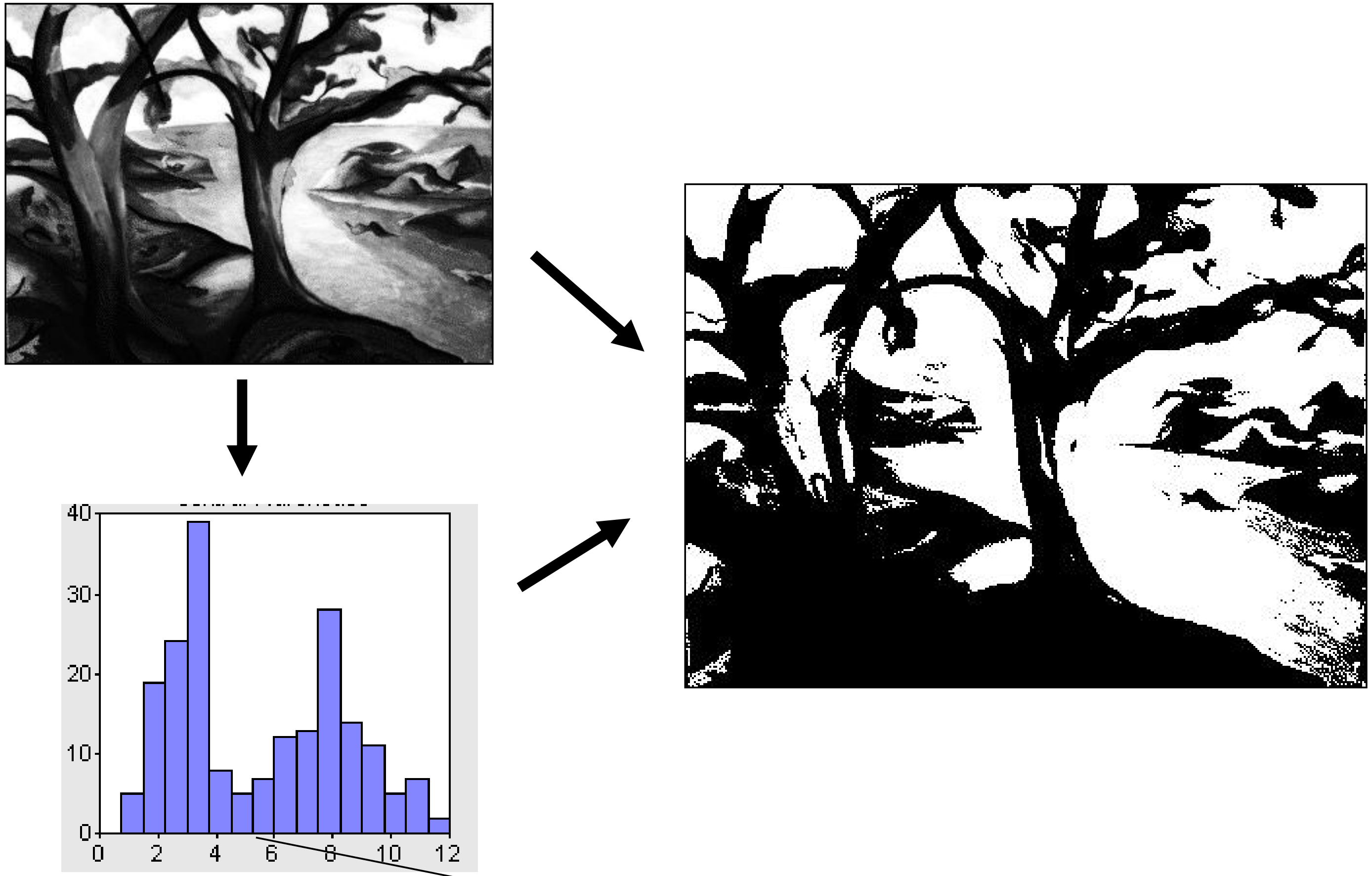
Binary Images

- Images with only two values (0 or 1)
- Simple to process and analyze
- Very useful for industrial applications

Selecting a Threshold

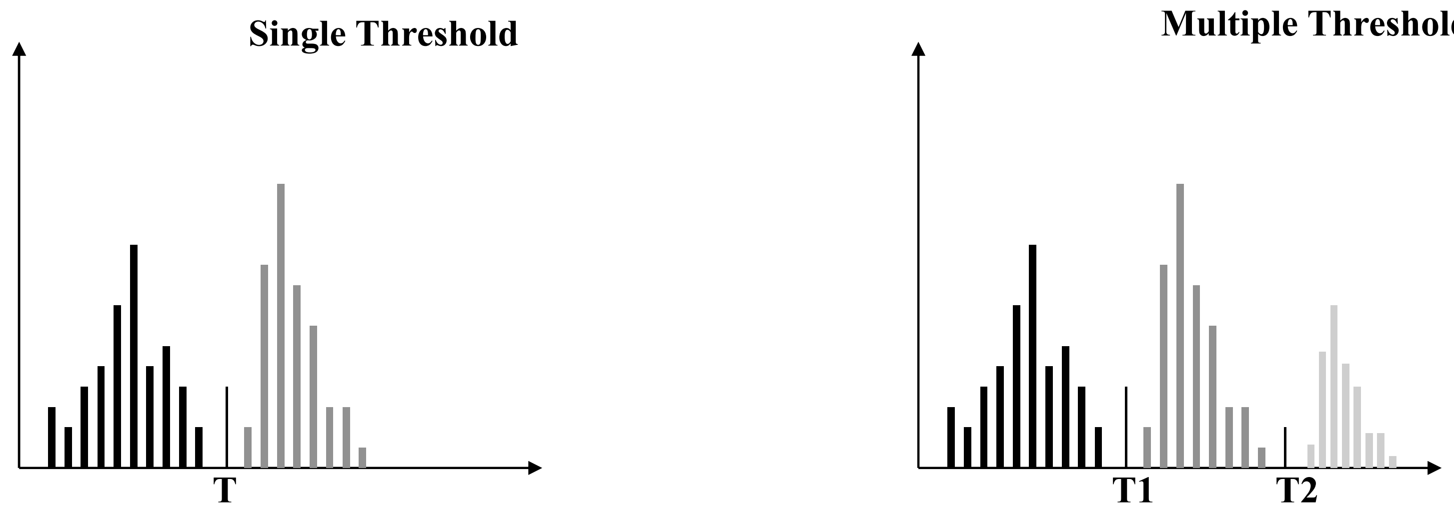


Threshold

Bimodal Histogram

Thresholding Techniques

- Thresholding is one of the most important approaches to image segmentation
- In this method, pixels that are alike in grayscale (or some other feature) are grouped together



Thresholding

- Often a image histogram is used to determine the best setting for the threshold(s)
- Some images (such as scanned text) will tend to be bimodal and a single threshold is suitable
- Other images may have multiple modes and multiple thresholds may be helpful
- In general multilevel thresholding is less reliable than single level thresholding. Mostly because it is very difficult to determine thresholds that adequately separate objects of interest

Thresholding Decision

- 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 graylevel at the point (x,y) and p(x,y) denotes some local property of the point (such as the average gray level of a neighbourhood centred on (x,y).

- A thresholded image is defined as

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

Thus pixels labelled 1, say, correspond to objects, and pixels labelled 0, say, correspond to the background

Local versus Global Thresholding

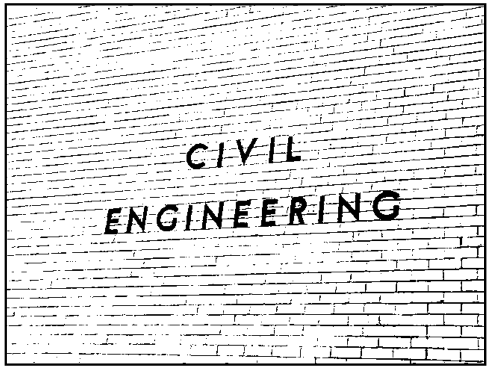
- If T depends only on f(x,y), the threshold is called *global*
- If T depends on both f(x,y) and neighbourhood p(x,y) the threshold is called *local*
- If, in addition, T depends on the spatial coordinates x and y, the threshold is called *dynamic*

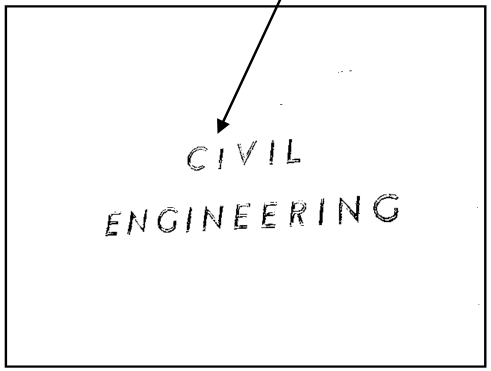
Example of Global Thresholding

Global Thresholding

Foreground "washed out"

Increasing Threshold





The Role of Illumination

- The formation of an image can be viewed as the product of a reflectance component r(x,y) and an illumination component i(x,y)
- Even though the reflectance of an object may have a histogram that allows easy separation of foreground and background, the effect of non-uniform illumination is to smear the histogram making simple thresholding ineffective
- If we know the illumination function (through calibration say), then we can sometimes eliminate this effect making global thresholding a practical method

Example

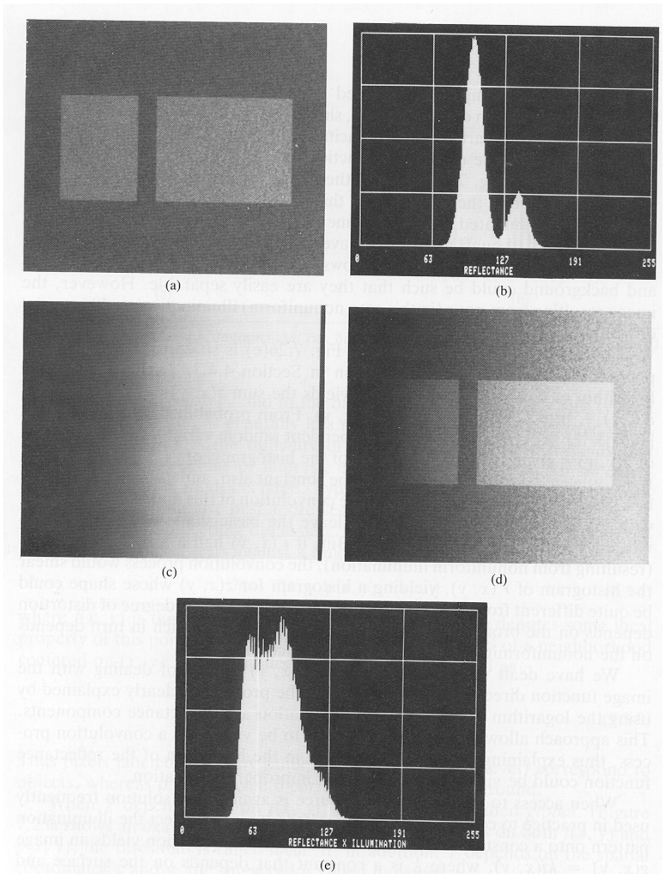
Reflectance Function

Reflectance Histogram

Illumination Function

Image

Image Histogram



Simple Global Thresholding

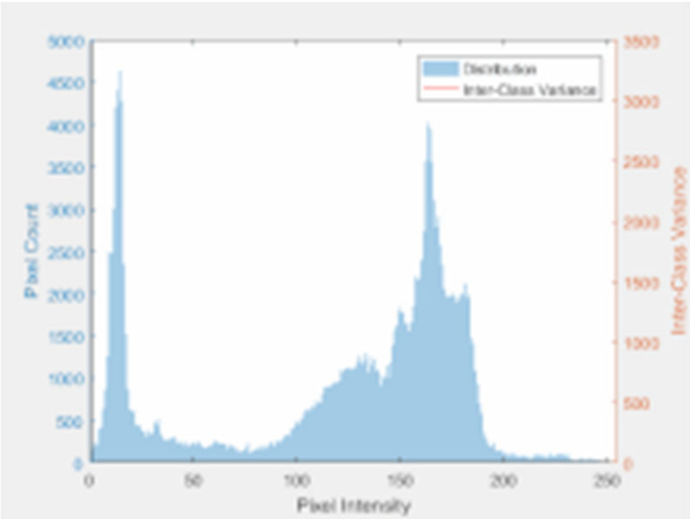
- In practice, global thresholding can be expected to be successful in highly controlled environments such as industrial inspection applications, where illumination control is feasible
- Note that uniform illumination is required for this method to work, or at least some sort of compensation for non-uniform illumination
- Usually a successful segmentation is highly dependent on the choice of thresholds
- There are many methods for automatic determination of these thresholds

Automatic Thresholding – Otsu's Method

Otsu's method is used to perform automatic image thresholding. In the simplest form, the algorithm returns a single intensity threshold that separate pixels into two classes, foreground and background.

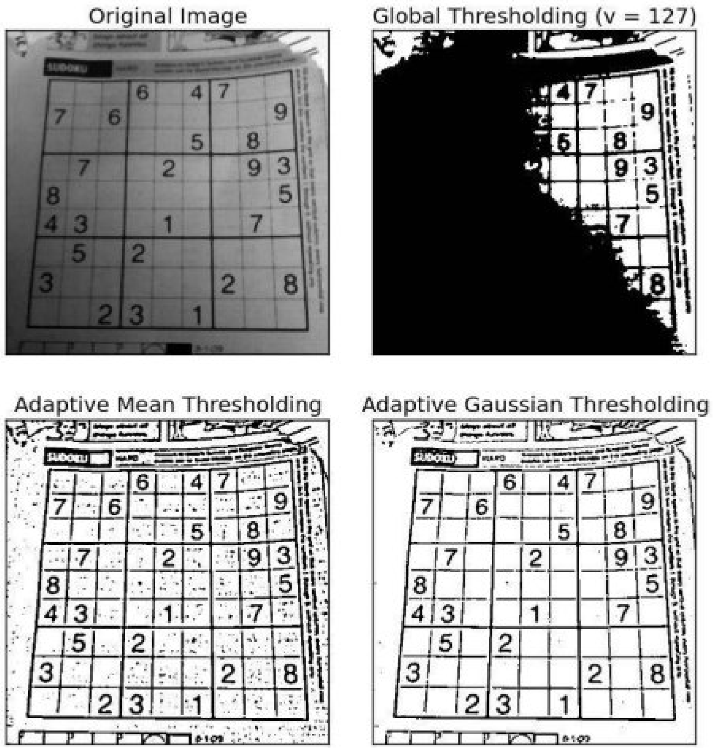
Matlab graythresh()

Otsu doesn't work well if foreground area is small compared to background area



Nobuyuki Otsu (1979). "A threshold selection method from gray-level histograms". IEEE Trans. Sys. Man. Cyber. 9 (1): 62–66.

Adaptive Thresholding

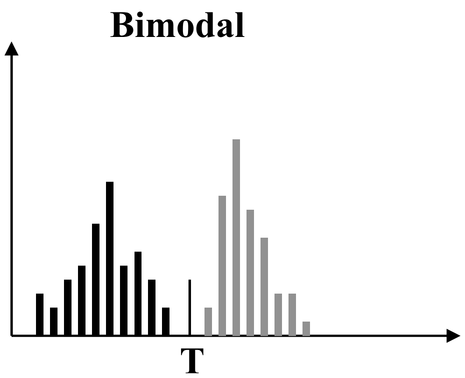


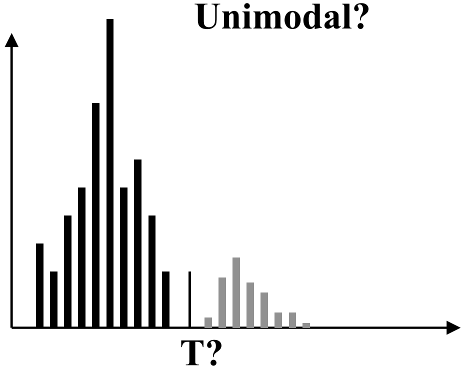
Adaptive Mean Threshold
threshold value is the mean of neighbourhood area

Adaptive Gaussian Threshold
threshold value is the weighted sum of neighbourhood values where weights are a gaussian window

Problems with Histogram

- What if the number of pixels in the foreground is much smaller than the number of pixels in the background?
- Perhaps we should only consider pixels near edges (high gradient)



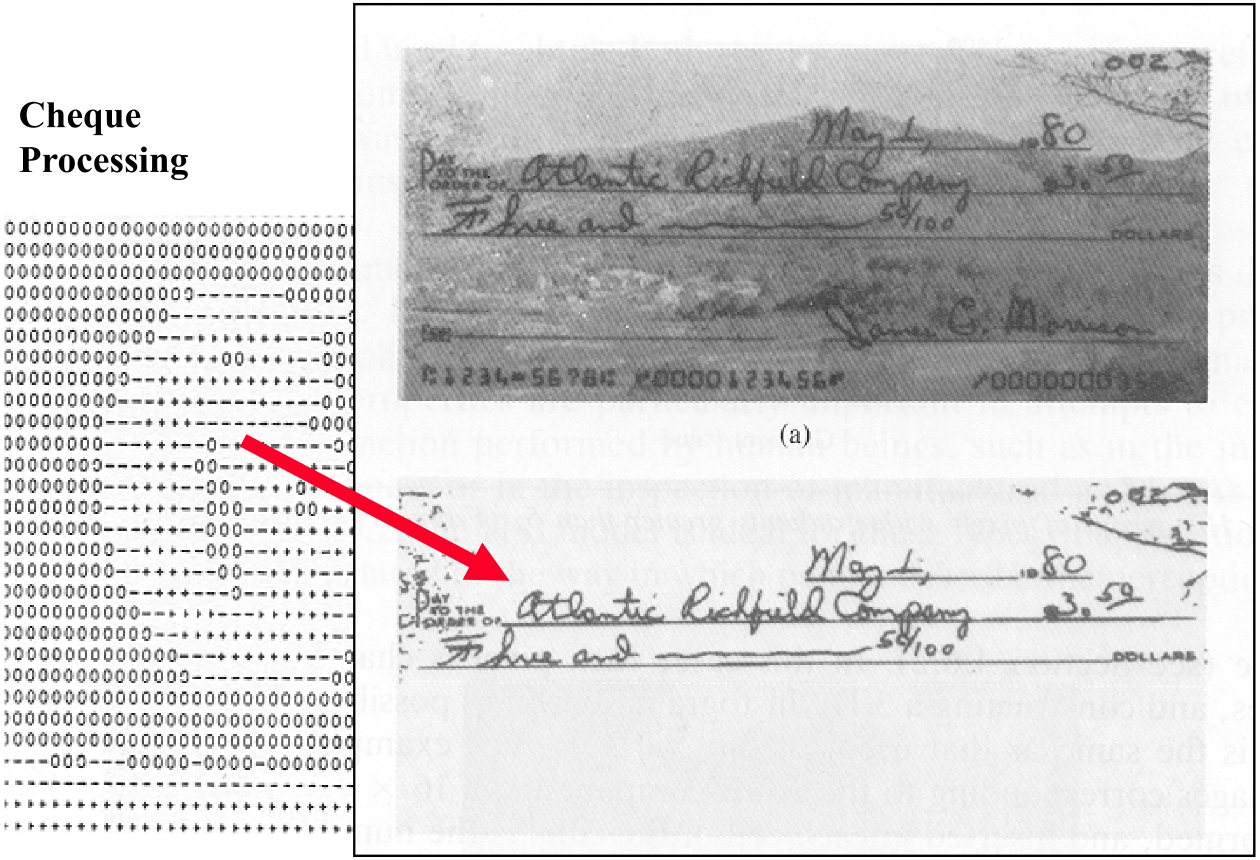


Example of Using Gradient

- Convert image into three-level image as follows
- $$s(x,y)=\begin{cases}0 \text{ if } \nabla f < T \\ + \text{ if } \nabla f \geq T \text{ and } \nabla^2 f \geq 0 \\ - \text{ if } \nabla f \geq T \text{ and } \nabla^2 f < 0\end{cases}$$
- This image can be converted to binary as follows
  - Scanning along either horizontally or vertically a transition from light to dark must correspond to a -+ sequence, the interior is either labelled 0 or +, and the transition from dark to light will correspond to the sequence +-
  - Thus a string of the form (…)(-+)(0 or +)(+-)(…) would have the inner parenthesis labelled as object (1).
  - From this string parsing each pixel can be labelled either 0 or 1

Example (White and Rohrer)

Cheque Processing



(a)

Looks like global thresholding because of constant T, but is actually local due to gradient operation

Motion in Segmentation

- Motion is a powerful cue used by humans to extract objects of interest from the background
- For fixed camera situations, a very common technique is simple frame differencing to detect change
- This technique is only applicable when the two frames are registered and the illumination is relatively constant
- Often isolated points in the difference image arise from noise, so some noise rejection may be necessary

Frame Differencing as an Edge Detector

- Frame differencing is just a method for differentiating (finding the gradient) of an image temporally rather than the more usual spatial filters.
- In that case, why not use some other edge detectors that have better properties?
- Significant improvements can be made by using more sophisticated filter kernels.

Frame differencing kernel

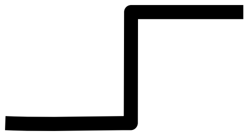
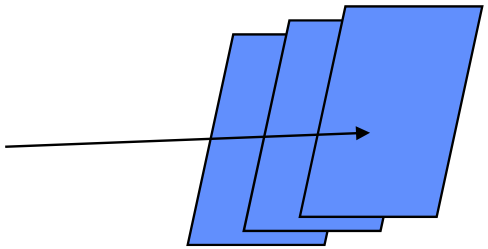
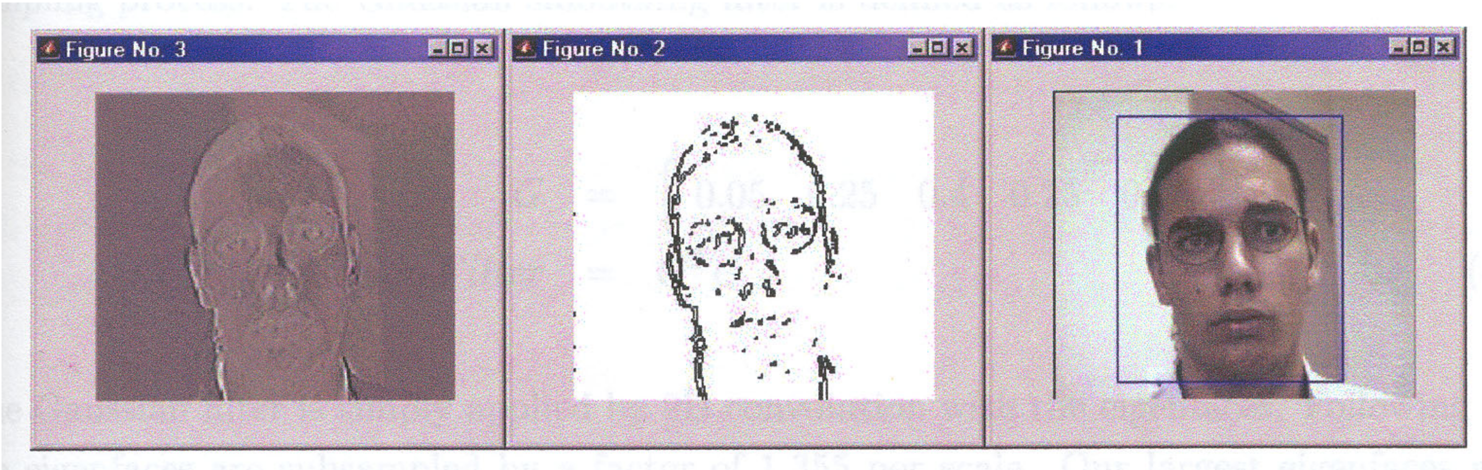


Image Sequence



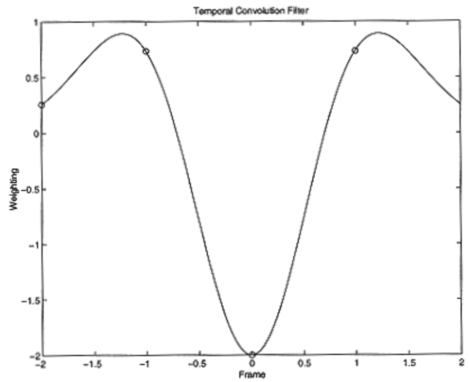
Temporal Filtering Example



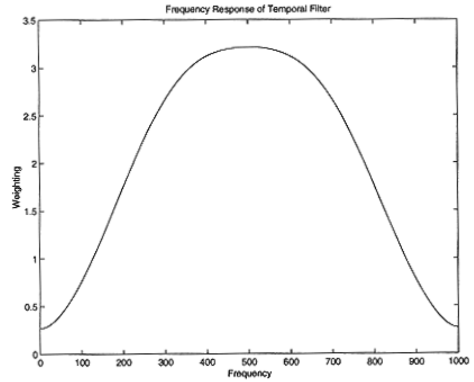
- This example uses the zero crossings of the Laplacian of Gaussian edge detector implemented against time over several frames to detect head movement.

Laplacian of Gaussian

Temporal Convolution Filter



Frequency Response of Temporal Filter



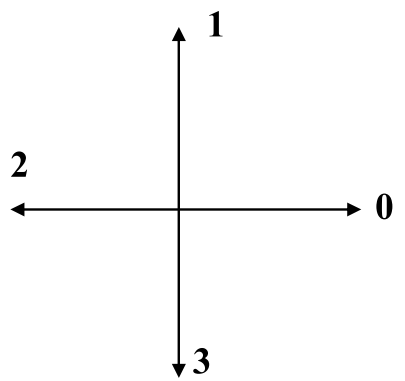
Note that this is basically a high pass filter response similar to a differentiator

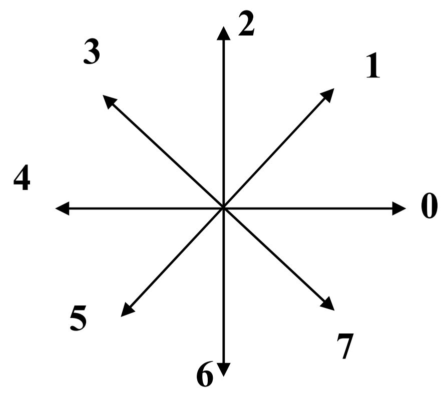
Image Representation

- Once an image is segmented into regions, the regions must generally be described in a suitable for for future processing
- There are generally two options
  - Represent the region in terms of its boundary
  - Represent the region in terms of its pixels
- The first method is often used if we are primarily concerned with the shape of a region
- The second method is more useful when we are concerned with internal properties such as colour and texture
- In general the features selected should be insensitive as possible to changes in size, translation, and rotation.

Chain Codes

- Chain codes are used to represent boundaries by a connected sequence of straight-line segments of specified length and direction.
- Typically this representation is based on the 4 or 8 connectivity of the segments.

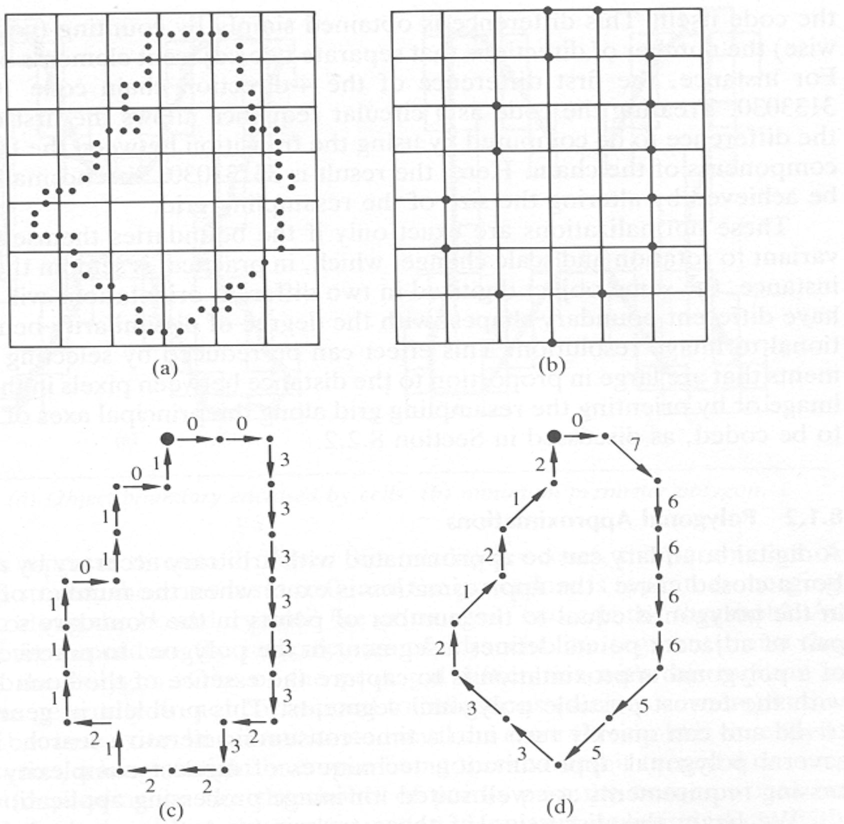




Chain Code Representation

- A chain code could be made by following the pixels around a boundary in, say, a clockwise direction
- The main problem with this method is that the chain codes are too long and the representation is much too sensitive to noise on the boundary
- Usually the boundary is resampled onto a larger grid spacing to avoid this problem
- The chain code will depend upon the starting point, but we can circumvent this problem by treating the code a circular sequence and rotating the code to form an integer of maximum magnitude
- We can make the code rotationally invariant by using relative angles rather than absolute
- Size normalization can be performed by scaling the resampling grid.

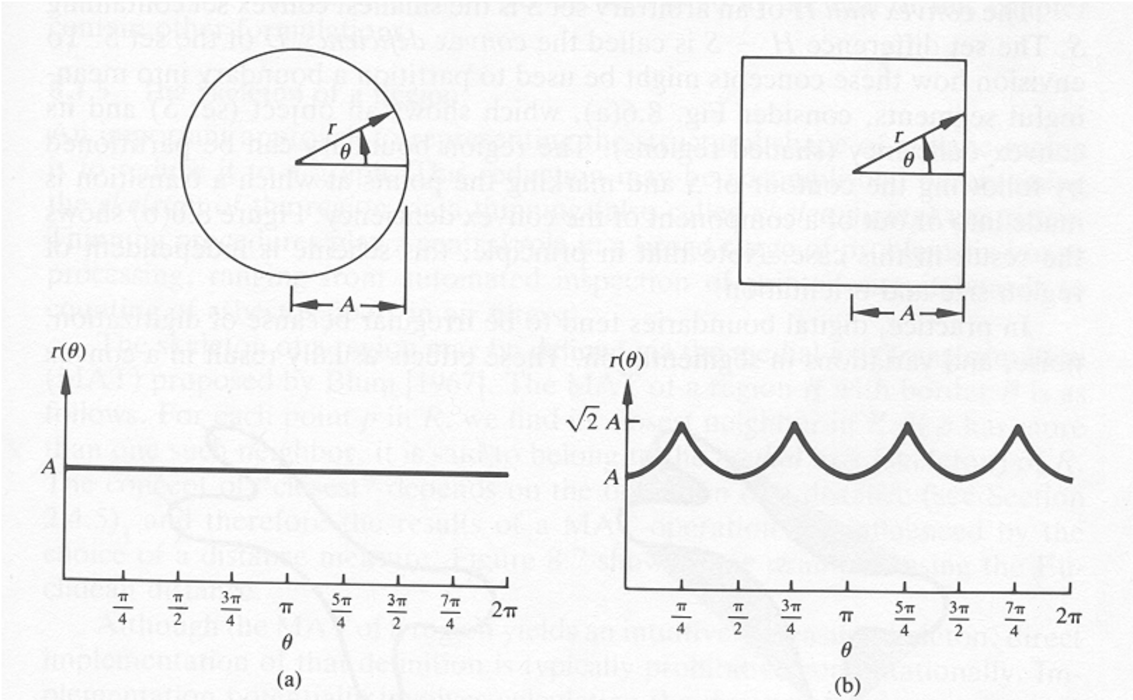
Chain Code Example



Signatures

- A signature is a 1D functional representation of a boundary and may be generated in a number of ways
- One of the simplest is to plot the distance from the centroid to the boundary as a function of angle
- Signatures generated by this approach are invariant to translation but depend on rotation and scaling
- Normalization to rotation can be achieved by selecting the point furthest from the centroid as the starting point – if this point is unique
- Another way is to pick the point on the principle eigen-axis furthest from the centroid

Example of Signature

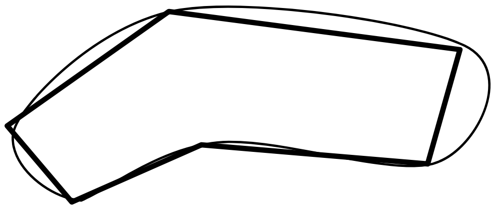


Problems: Robustness, normalisation, representing concave shapes

Many variations on this technique

Interesting Variations

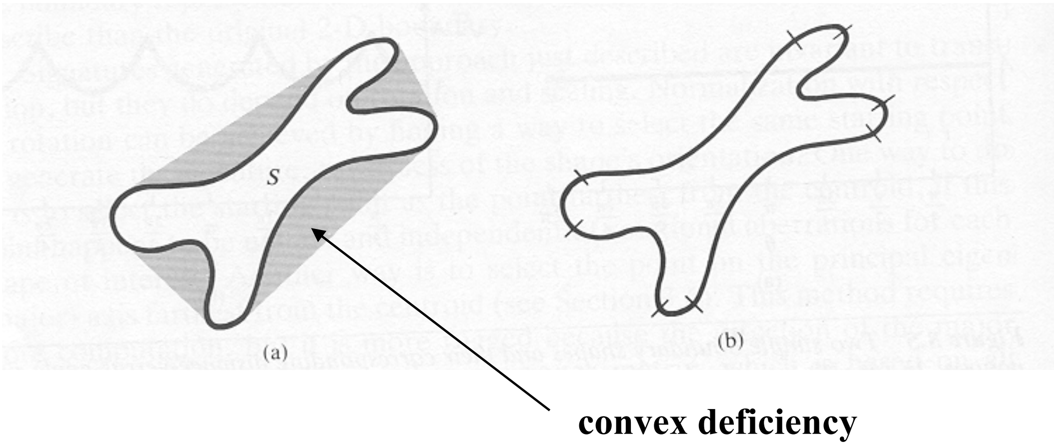
- To allow for a wider range of shapes (including concave), divide boundary into uniform intervals and record angle changes between line segments traversing boundary.
- Take Fourier Transform of coefficients to make representation rotationally and scale invariant
  - leads to called *Fourier Shape Descriptors* which can be quite powerful
  - shape can be decomposed into Fourier components with first second harmonics etc
  - Most reliable shape information is contained in the low frequency terms



Convex Hull

- Decomposing a boundary into segments can often be useful
- Decomposition reduces the boundary's complexity and thus simplifies the description process
- This technique is particularly attractive when the boundary contains one or more significant concavities that carry shape information
- The *convex hull* H of an arbitrary set S is the smallest convex set containing S.
- The set difference H-S is called the *convex deficiency* D of the set S

Example of Convex Hull

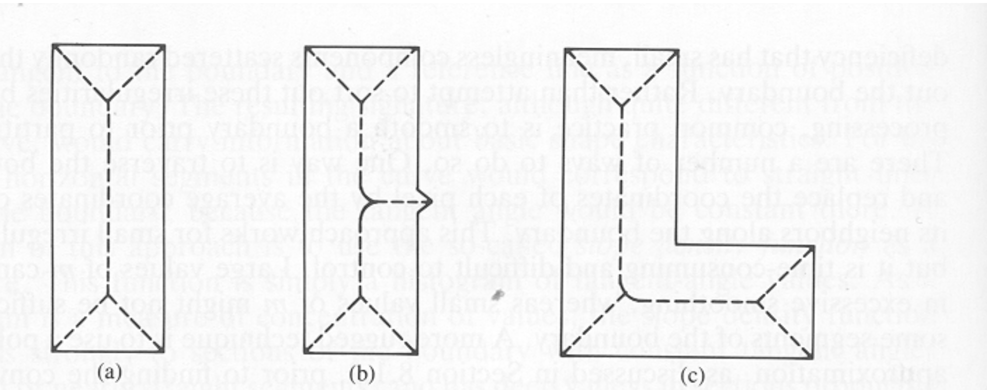


The region boundary can be partitioned by finding the contour of S and marking the points at which the transition is made into or out of a component of the convex defficiency

Skeleton of a Region

- An important technique for representing the structural shape of a plane region is to reduce it to a graph
- This may be accomplished by obtaining the skeleton of the region via a thinning (skeletonization) algorithm
- The skeleton may be defined by the medial axis transformation defined by Blum(1967).
- The MAT of a region R with border B is as follows
  - For each point p in R, we find its closest neighbour in B
  - if p has more than one such neighbour, it is said to belong to the medial axis of R
  - The definition of closest depends on the measure of distance. Frequently we use Euclidean distance.

Medial Axes of Three Regions



While the result is pleasing, the computation of MAT is expensive

Many algorithms have been proposed to approximate MAT at reduced computational load

Thinning Algorithm

- Approximate method successively deletes edge points subject to the constraint that it
  - does not remove end points
  - does not break connectedness
  - does not cause excessive erosion of the region
- Method consists of successive passes of two basic steps applied to the contour points of a region, where a contour point is a point having at least one 8-neighbour valued 0 (background)

Algorithm

- Step1
  - Flag a contour point p for deletion if the following conditions are satisfied
    a. number of 8-neighbours is between 2 and 6
    b. $S(p_1)=1$
    c. $p_2.p_4.p_6=0$
    d. $p_4.p_6.p_8=0$
    where $S(p_1)$ is the number of 0-1 transitions in the sequence $p_2,p_3,..p_9$
  - delete flagged points
- Step2
  - repeat Step 1 but change the last two conditions to
    c. $p_2.p_6.p_8=0$
    d. $p_2.p_4.p_8=0$
- Repeat until no more points to delete

p9	p2	p3
p8	p1	p4
p7	p6	p5

Comments

- Condition a is violated when contour point p has only 1 or 7 8-neighbours valued 1
  - having only one implies that p is an end-point of a skeleton and should not be deleted
  - deleting p with 7 neighbours would cause erosion into the region
- Condition b is violated when it is applied to points on a stroke 1 pixel wide
  - hence this conditions prevents disconnection of segments of a skeleton

More Comments

- Conditions c and d are satisfied simultaneously by the minimum set of values
  - (p4=0 or p6=0) or (p2=0 and p8=0)
  - a point which satisfies these conditions is an east or south boundary point or a northwest corner point in the boundary
  - in either case p is not part of the skeleton and should be removed
  - similar arguments apply for the conditions of Step2

p9	p2	p3
p8	p1	p4
p7	p6	p5

Skeletonization Example