CAS CS 585

Image and Video Computing

Lecture by Margrit Betke

January 30 and February 1, 2024

Traditional Segmentation Algorithms

Boundary Following Algorithm

Hausdoff Distance, Curvature Measures

Segmentation

Definition 1:

Segmentation = finding outline of object or region in image

Definition 2:

Segmentation = grouping of pixels into regions such that:

- Pixels in each region have a common property
- Pixels in adjacent regions do not share this property
- Exclusive Partitioning: Pi intersect Pj = {} for all i not equal to j
- Exhaustive Partitioning: Union of Pi = entire image

Definition 2:
Segmentation = grouping of pixels into regions such that:
Pixels in each region have a common property
Pixels in adjacent regions do not share this property
Exclusive Partitioning: Pi intersect Pj = {} for all i not equal to j
Exhaustive Partitioning: Union of Pi = entire image

Common Property:

Easy to test:

- Same (or similar) color
- Same grey value
- Brightness above some threshold
- Black versus white

Deep Learned:

"Typical" object appearance

"Semantic Segmentation"

common property =

same type of object, e.g. cars

"Instance Segmentation"

common property = object instance, e.g., a specific car (my car)

Definition 2:
Segmentation = grouping of pixels into regions such that:
Pixels in each region have a common property
Pixels in adjacent regions do not share this property
Exclusive Partitioning: Pi intersect Pj = {} for all i not equal to j
Exhaustive Partitioning: Union of Pi = entire image

Common Property:

Same color
Same grey value
Brightness above some threshold
Black versus white

Segmentation Pipeline of traditional (not-deep-learned) methods:

Color image -> e.g., R+G+B/3
Greyscale image ->
Segmentation algorithm
e.g., thresholding
-> Binary image ->
Boundary Following Algorithm

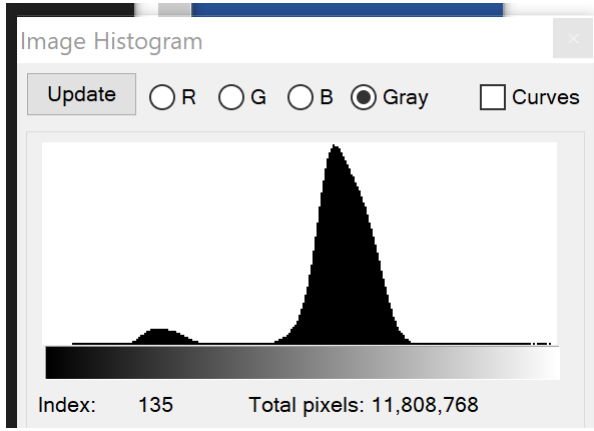
Segmentation Methods

1. <u>Absolute Thresholding</u>

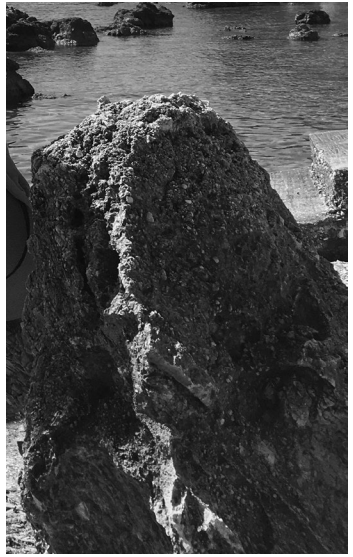
   Algorithm:

   1) Determine by inspection of the grey value histogram of the image which threshold T separates bright and dark pixels
   2) Scan image and assign foreground and background labels to objects/regions

   Disadvantage ?



Segmentation Methods



1. Absolute Thresholding

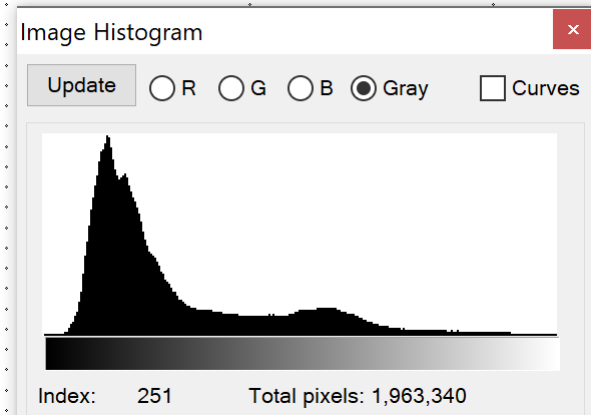
   Algorithm:

   1) Determine by inspection of the grey value histogram of the image which threshold T separates bright and dark pixels
   2) Scan image and assign foreground and background labels to objects/regions

   Disadvantage: Does not work well if

   object is illuminated unevenly

   MRI scanner has a bias



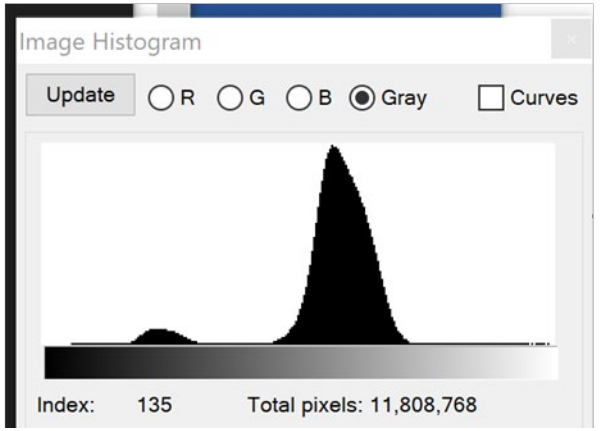
Segmentation Methods

2. Percentile Method

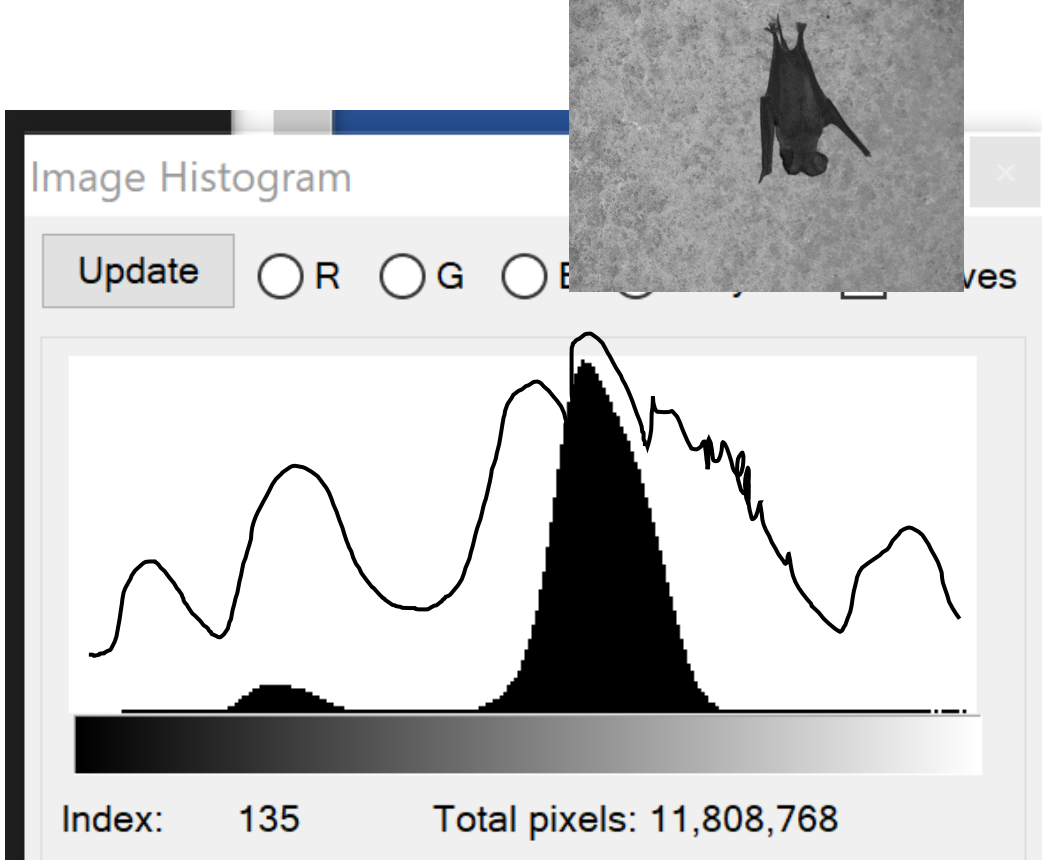
   Assumption: Single (dark) object occupies p% of image

   Algorithm:

   1) n = number of pixels
   2) m = number of object pixels = n ∗ p%
   3) Scan image and find m dark pixels and assign them to belong to desired object



Segmentation Methods



3. Mode Method

Assumptions: *n* objects and background have different and relatively uniform grey values; objects are not "very small"

Algorithm for 1 Object:

1) Find local maxima at some minimum distance apart, e.g., maximum at g_i & g_j
2) g_k = minimum between g_i & g_j
3) Peakiness $p(i,j,k) = [\min \{H(g_i), H(g_j)\}] / H(g_k)$
4) Use (i,j,k) with highest peakiness p: $T = g_k$

Algorithm for *n* Objects:

Move through histogram to find *n* valleys based on their peakiness

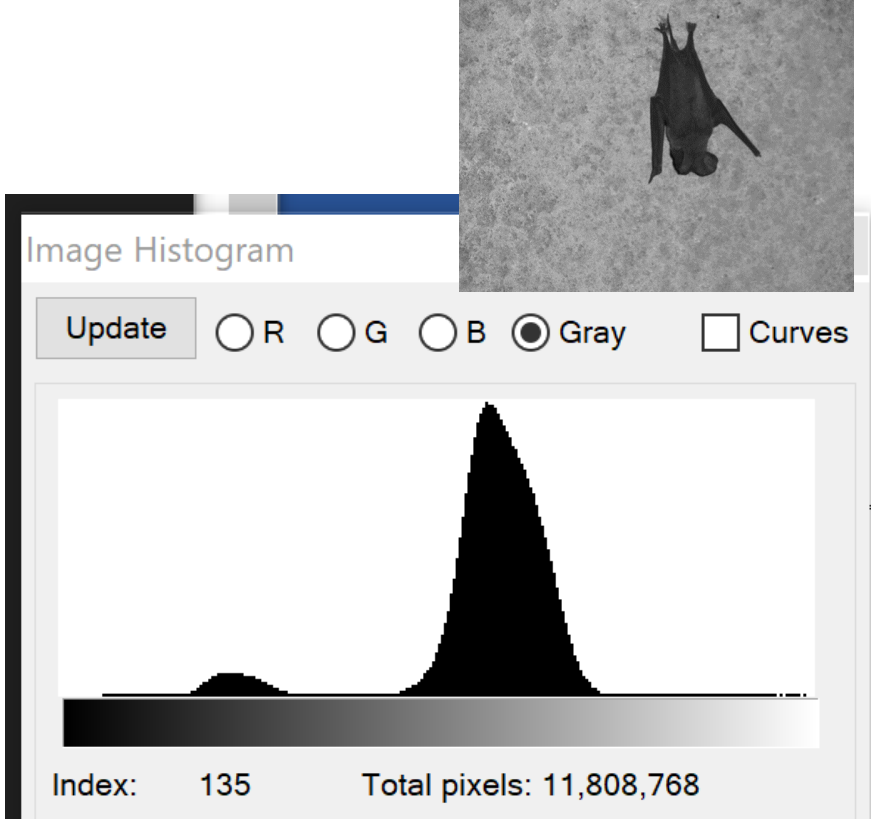
Segmentation Methods

4. Adaptive Thresholding

Adapt T to image region

Algorithm: Create different thresholds for different image regions

Segmentation Methods



5. Iterative Threshold Method

- T = average grey value in image
- Partition image into sets of pixels with grey value above and below T
- Compute the average grey value in each set
- T = average of these averages
- Repeat procedure until averages do not change much anymore

What if you segmented the object of interest into several regions instead of one region?

Merging of Adjacent Regions

Merging: Combine adjacent regions with similar characteristics:

Mean of grey values of region 1: $\mu_1 = 1/n_1 \sum_{\text{i in region1}} g_i$

Mean of grey values of region 2: $\mu_2 = 1/n_2 \sum_{\text{i in region2}} g_i$

Method 1: Merge based on means

If $|\mu_1 - \mu_2| <$ threshold, merge regions.

Method 2: Merge based on likelihood ratio (use standard deviations)

$\sigma_1^2 = 1/n_1 \sum_{\text{i in region1}} (g_i - \mu_1)^2$

$\sigma_2^2 = 1/n_2 \sum_{\text{i in region2}} (g_i - \mu_2)^2$

$L = P(g_1, g_2, \ldots, g_{n1+n2} | H_0) / P(g_1, g_2, \ldots, g_{n1+n2} | H_1)$

$= \sigma_0^{m1+m2} / (\sigma_1^{m1} \sigma_2^{m2})$

Assume the regions in an image have constant grey values that are corrupted by independent additive zero-mean Gaussian noise.
If L < threshold, then it is highly likely that there is only one region => merge

Boundary Following Algorithm

Goal: Given a segmented region (or object), find the pixels on the closed boundary of the region (or object).

Tool: Neighborhood definitions

N_4

0	1	0
1	?	1
0	1	0

N_8

1	1	1
1	?	1
1	1	1

Notation: R=region, $\underline{R}$=background,

b=pixel at boundary (outside object/region)

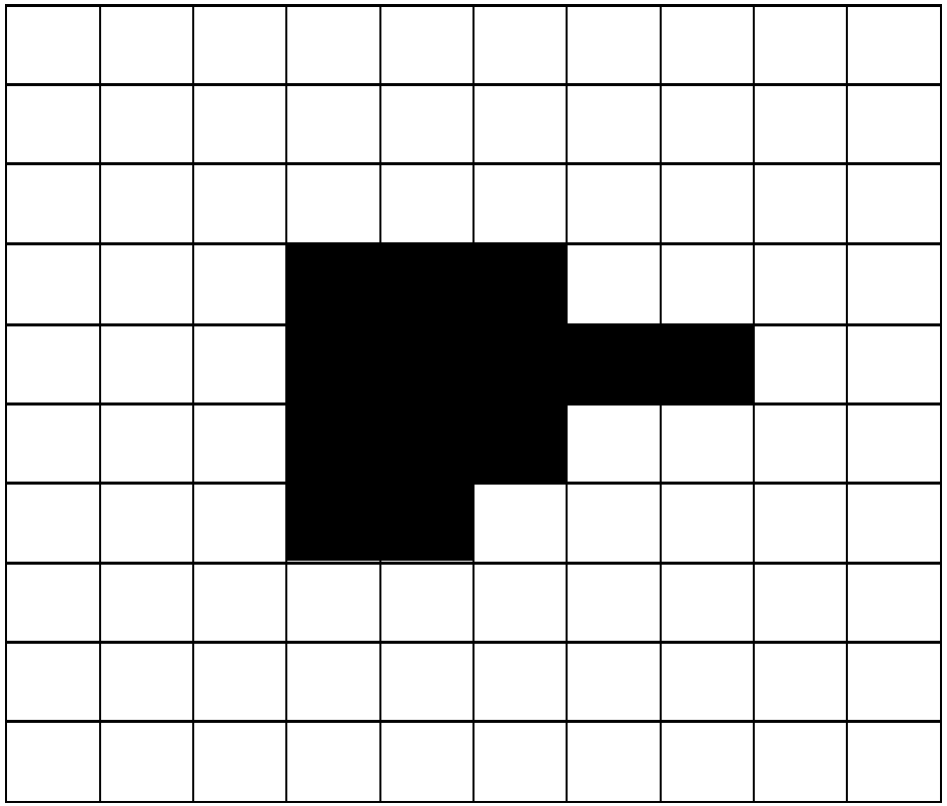
c= “current pixel” inside object/region on boundary

Boundary Following Algorithm

1. Find start pixel s (through scanning binary image), c=s
2. Find N_4–neighbor b in R to west of c
3. Among N_8–neighbors of c, starting with b, search clockwise for first n_i in R
4. $c = n_i$, $b = n_{i-1}$
5. Repeat steps 3 & 4 until c=s

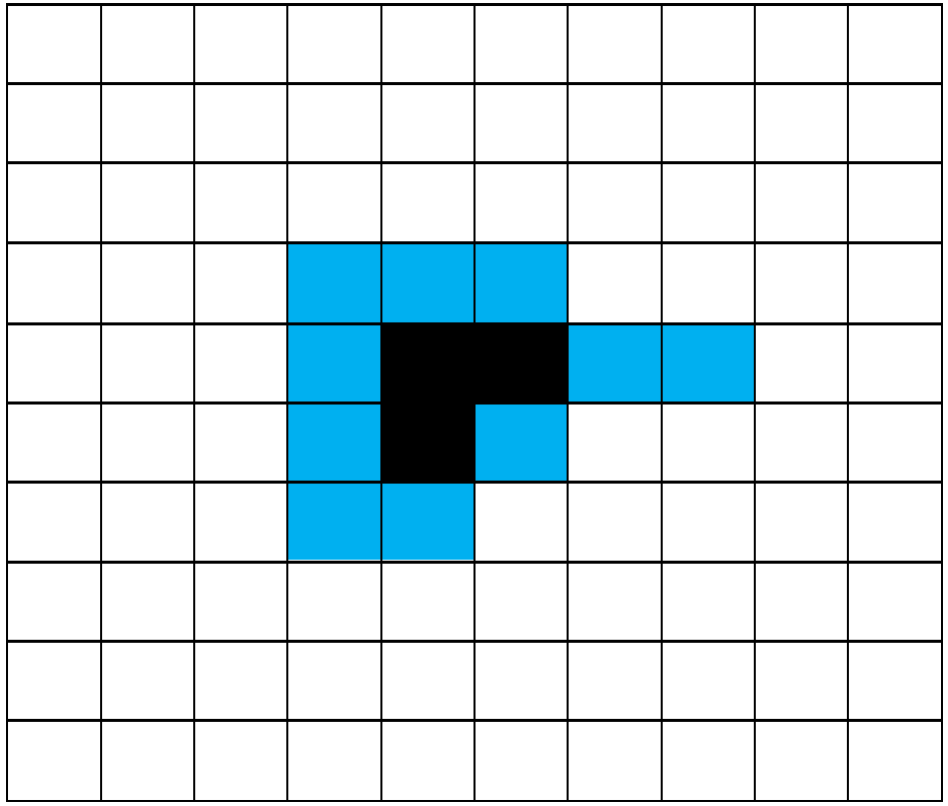
Boundary Following Algorithm

1. Find start pixel s (through scanning binary image), c=s
2. Find N_4–neighbor b in R to west of c
3. Among N_8–neighbors of c, starting with b, search clockwise for first n_i in R
4. $c = n_i$, $b = n_{i-1}$
5. Repeat steps 3 & 4 until c=s



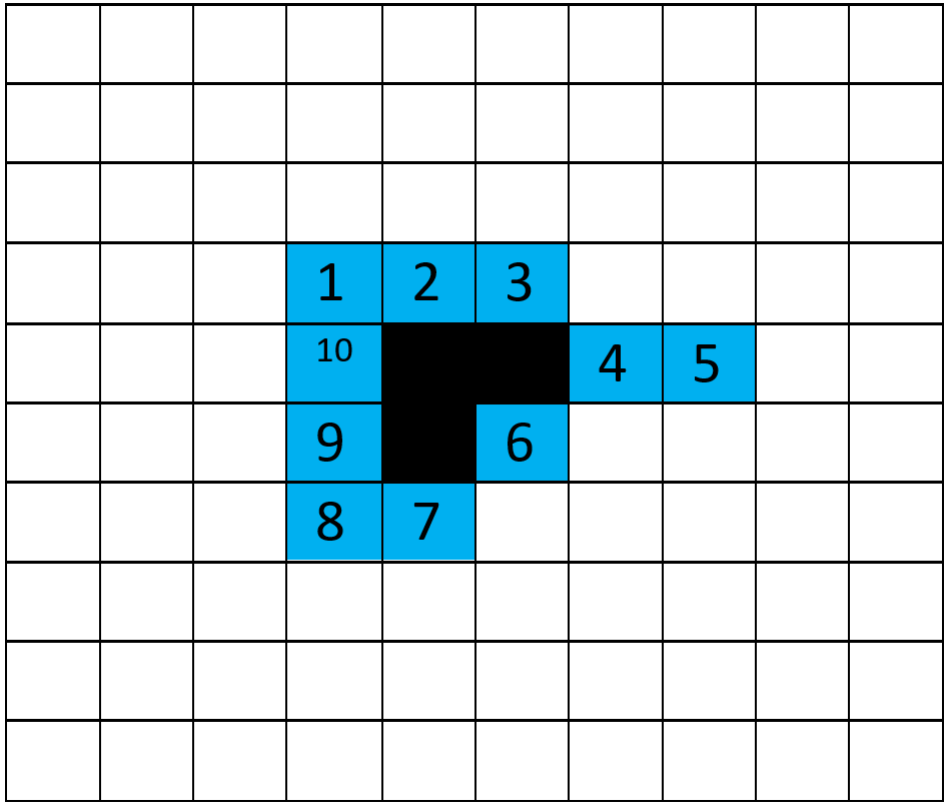
Boundary Following Algorithm

1. Find start pixel s (through scanning binary image), c=s
2. Find N_4–neighbor b in R to west of c
3. Among N_8–neighbors of c, starting with b, search clockwise for first n_i in R
4. $c = n_i$, $b = n_{i-1}$
5. Repeat steps 3 & 4 until c=s



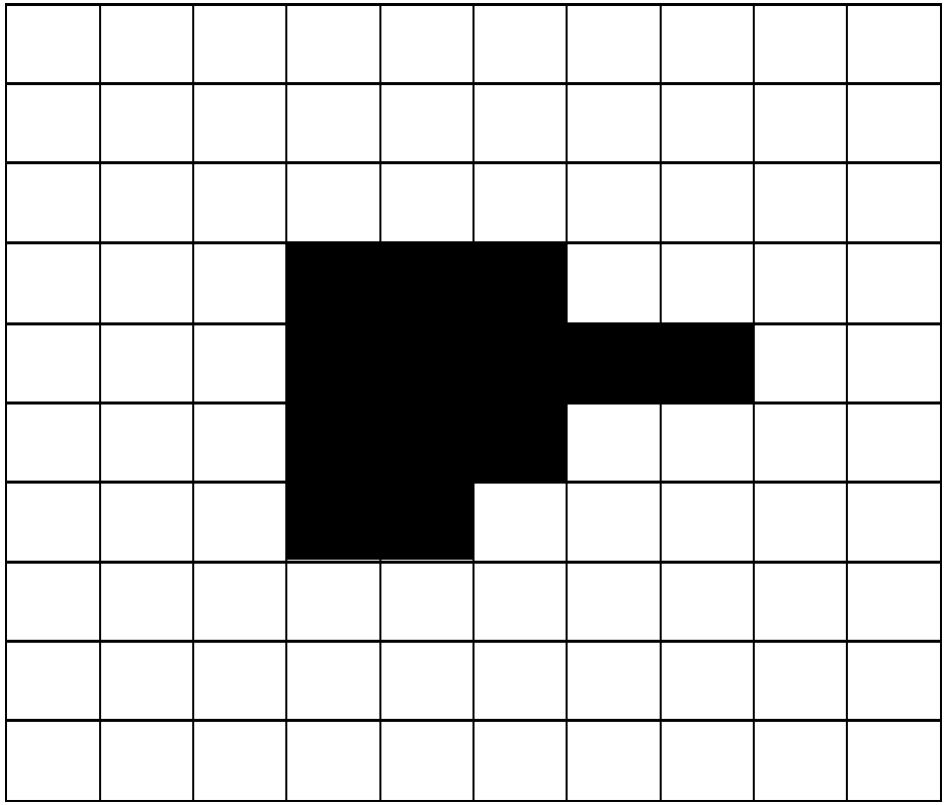
Boundary Following Algorithm

1. Find start pixel s (through scanning binary image), c=s
2. Find N_4–neighbor b in R to west of c
3. Among N_8–neighbors of c, starting with b, search clockwise for first n_i in R
4. $c = n_i$, $b = n_{i-1}$
5. Repeat steps 3 & 4 until c=s



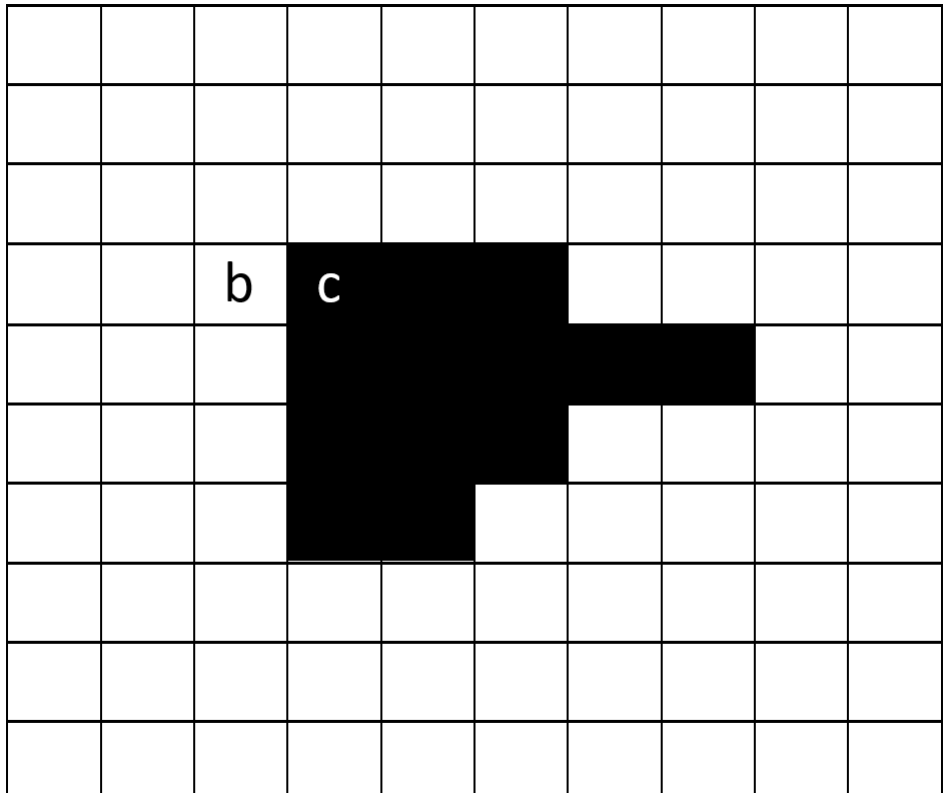
Boundary Following Algorithm

1. Find start pixel s (through scanning binary image), c=s
2. Find N_4–neighbor b in R to west of c
3. Among N_8–neighbors of c, starting with b, search clockwise for first n_i in R
4. $c = n_i$, $b = n_{i-1}$
5. Repeat steps 3 & 4 until c=s



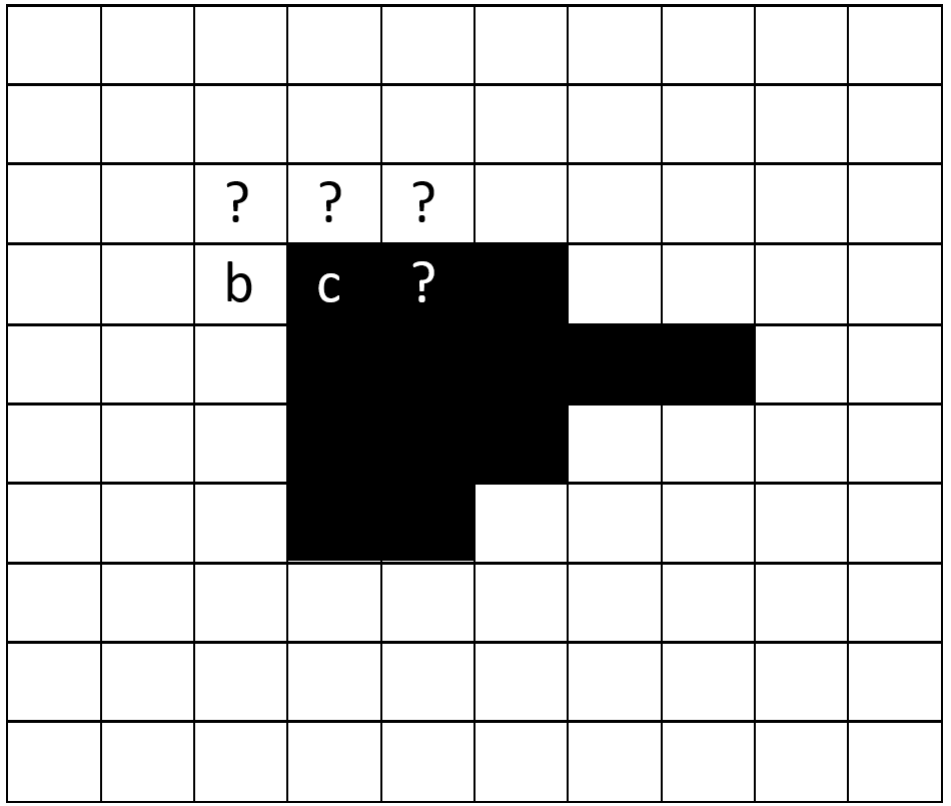
Boundary Following Algorithm

1. Find start pixel s (through scanning binary image), c=s
2. Find N_4–neighbor b in R to west of c
3. Among N_8–neighbors of c, starting with b, search clockwise for first n_i in R
4. $c = n_i$, $b = n_{i-1}$
5. Repeat steps 3 & 4 until c=s



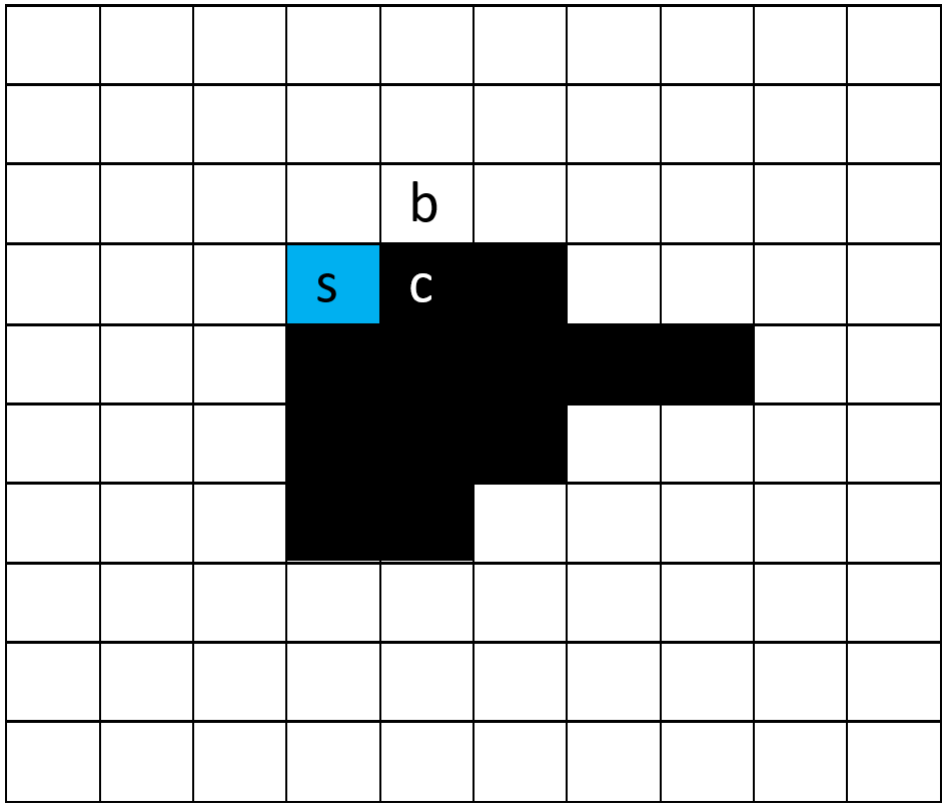
Boundary Following Algorithm

1. Find start pixel s (through scanning binary image), c=s
2. Find N_4–neighbor b in R to west of c
3. Among N_8–neighbors of c, starting with b, search clockwise for first n_i in R
4. $c = n_i$, $b = n_{i-1}$
5. Repeat steps 3 & 4 until c=s



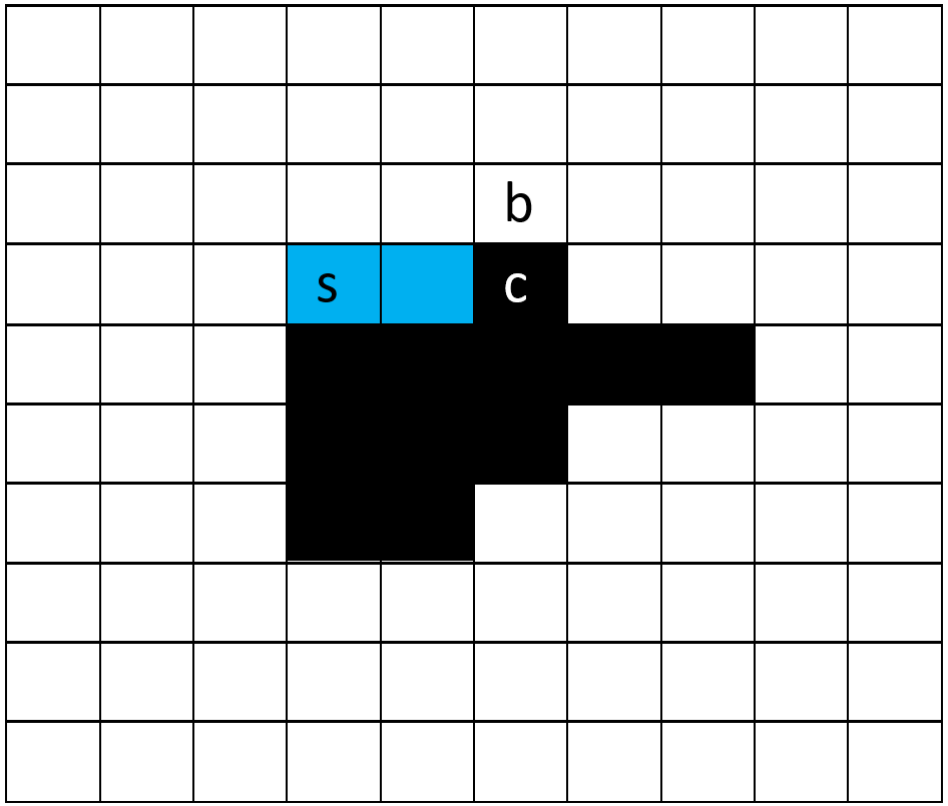
Boundary Following Algorithm

1. Find start pixel s (through scanning binary image), c=s
2. Find N_4–neighbor b in R to west of c
3. Among N_8–neighbors of c, starting with b, search clockwise for first n_i in R
4. $c = n_i$, $b = n_{i-1}$
5. Repeat steps 3 & 4 until c=s



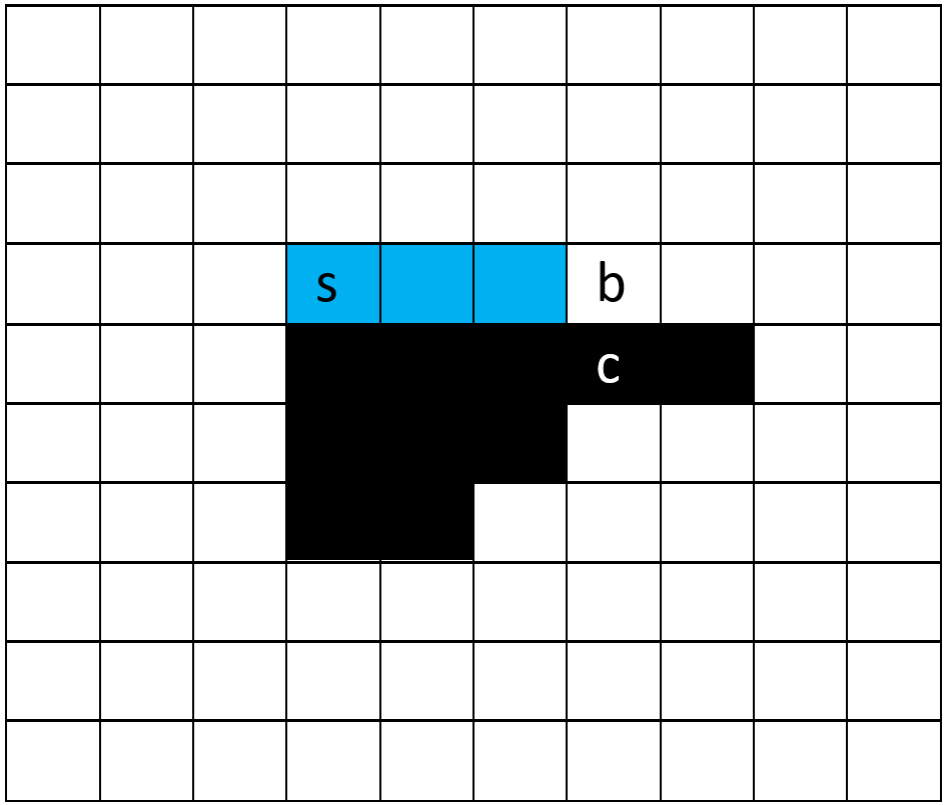
Boundary Following Algorithm

1. Find start pixel s (through scanning binary image), c=s
2. Find N_4–neighbor b in R to west of c
3. Among N_8–neighbors of c, starting with b, search clockwise for first n_i in R
4. $c = n_i$, $b = n_{i-1}$
5. Repeat steps 3 & 4 until c=s



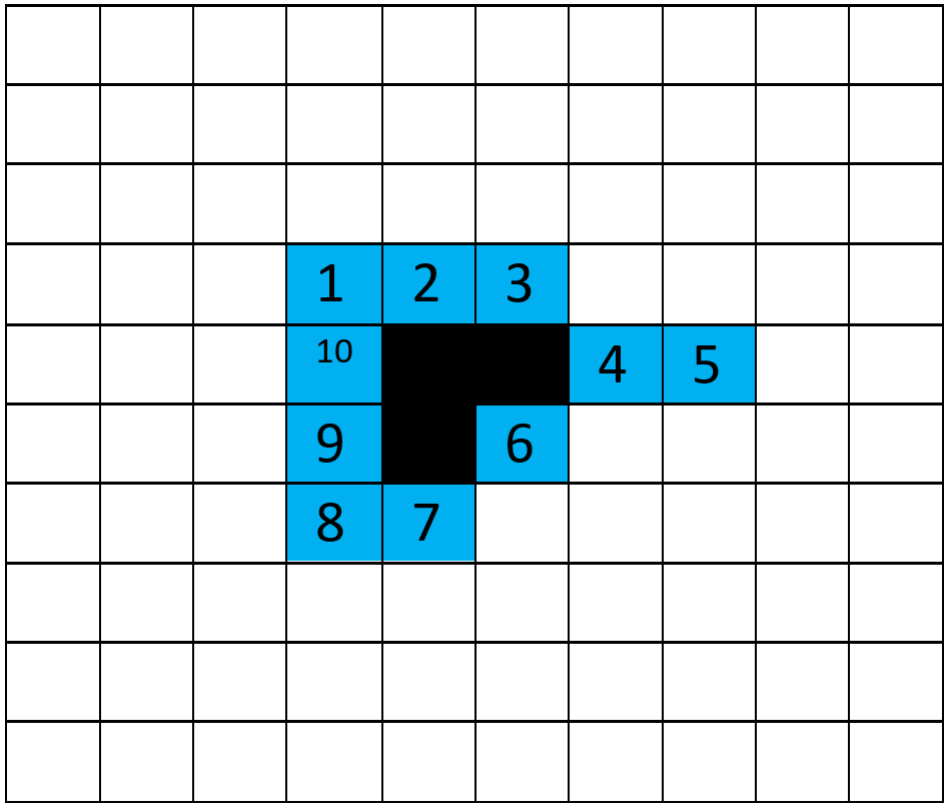
Boundary Following Algorithm

1. Find start pixel s (through scanning binary image), c=s
2. Find N_4–neighbor b in R to west of c
3. Among N_8–neighbors of c, starting with b, search clockwise for first n_i in R
4. $c = n_i$, $b = n_{i-1}$
5. Repeat steps 3 & 4 until c=s



Boundary Following Algorithm

1. Find start pixel s (through scanning binary image), c=s
2. Find N_4–neighbor b in R to west of c
3. Among N_8–neighbors of c, starting with b, search clockwise for first n_i in R
4. $c = n_i$, $b = n_{i-1}$
5. Repeat steps 3 & 4 until c=s



Comparison of two Boundaries (or Curves): using the Hausdorff Distance

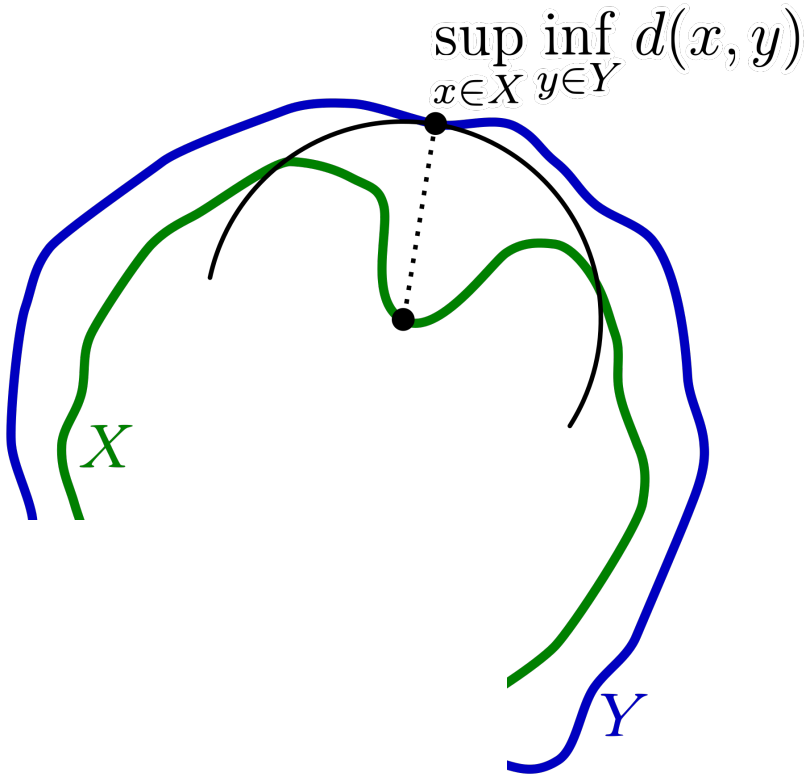
Idea: Two boundaries have a small Hausdorff distance if every pixel of either boundary is close to some pixel of the other boundary.

The Hausdorff distance is the largest distance (supremum) between points on two curves when an adversary can select a point on one of the curves (goal to maximize the distance) while you are trying to minimize the distance (infimum) from that chosen point to a point on the other curve

$$\sup_{x \text{ in } X} \inf_{y \text{ in } Y} \text{Euclidean distance } (x,y)$$

Comparison of two boundaries (or curves): using the Hausdorff Distance

The Hausdorff distance is the largest distance between points on two curves when an adversary can select a point on one of the curves (goal to maximize the distance) while you are trying to minimize the distance from that chosen point to a point on the other curve



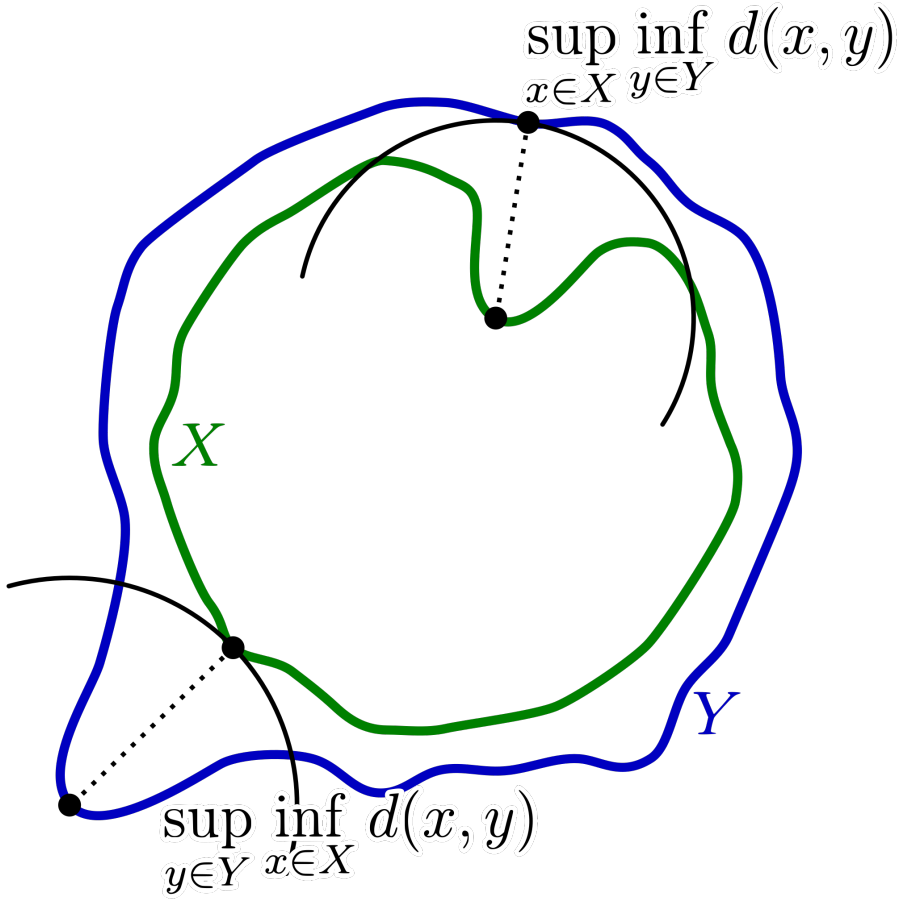
sup $_{x \text{ in } X}$ inf $_{y \text{ in } Y}$ Euclidean distance (x,y)

largest smallest

Image credit: Wikipedia, Rocchini

Comparison of two boundaries (or curves): using the Hausdorff Distance

The Hausdorff distance is the largest distance between points on two curves when an adversary can select a point on one of the curves (goal to maximize the distance) while you are trying to minimize the distance from that chosen point to a point on the other curve



sup $_{y \text{ in } Y}$ inf $_{x \text{ in } X}$ Euclidean distance (x,y)

largest smallest

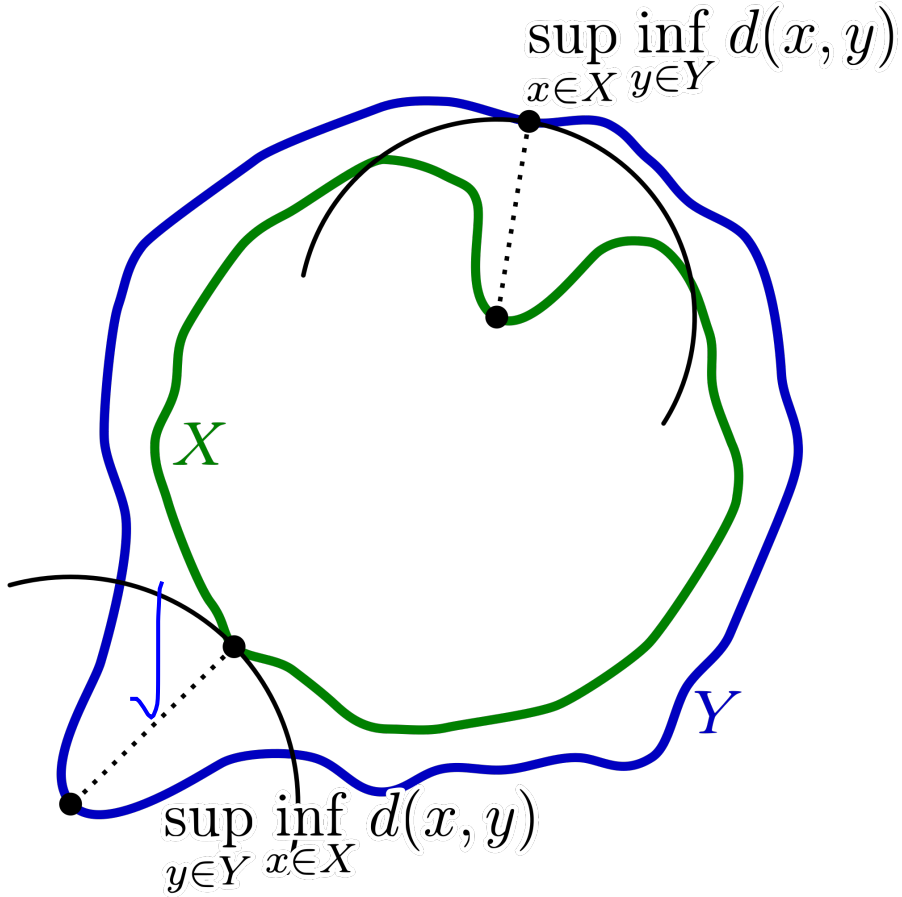
Image credit: Wikipedia, Rocchini

Comparison of two boundaries (or curves): Hausdorff Distance

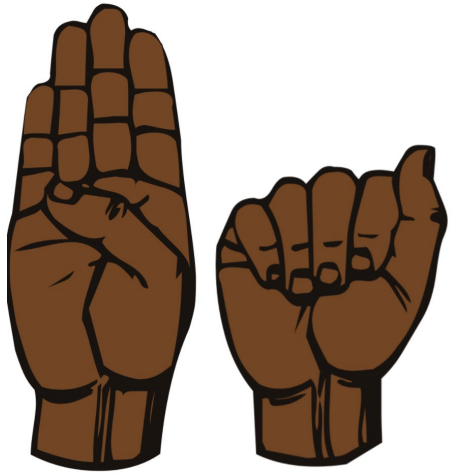
Hausdorff distance between curves X and Y =

max {sup $_{x \text{ in } X}$ inf $_{y \text{ in } Y}$ distance (x,y),

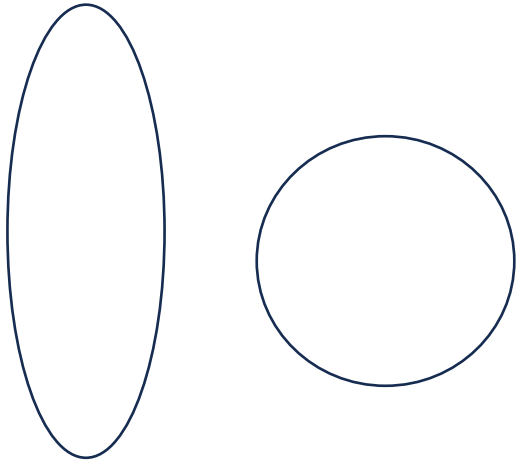
sup $_{y \text{ in } Y}$ inf $_{x \text{ in } X}$ distance (x,y) }



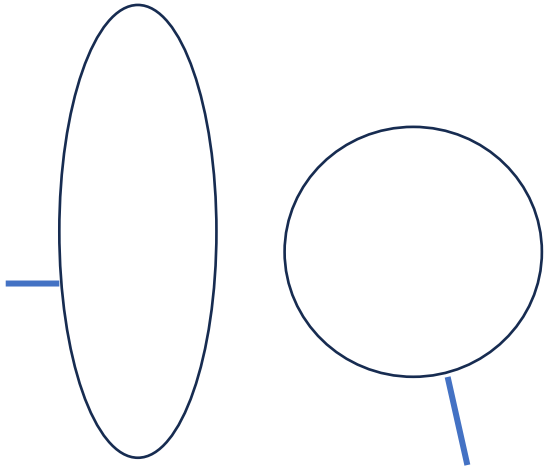
Application: Use Hausdorff Distance to compare hand shapes



ASL Sign: A B



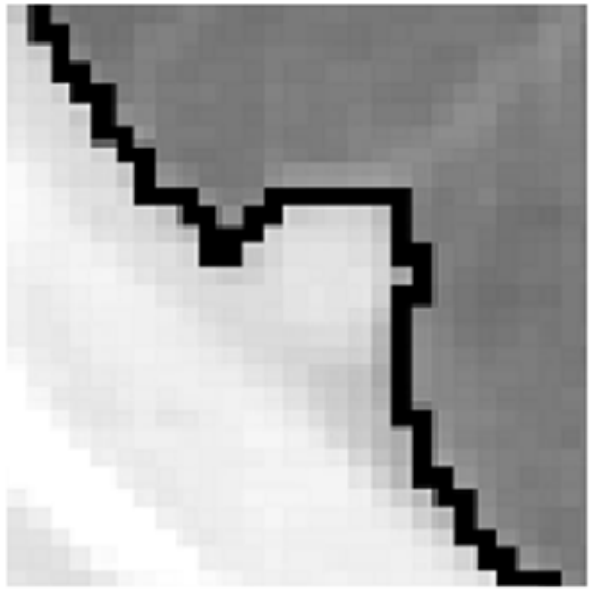
Template boundary of equivalent shape



Comparison with measured boundary

Curve Processing:
Lung Nodule on Boundary

Before: High curvature

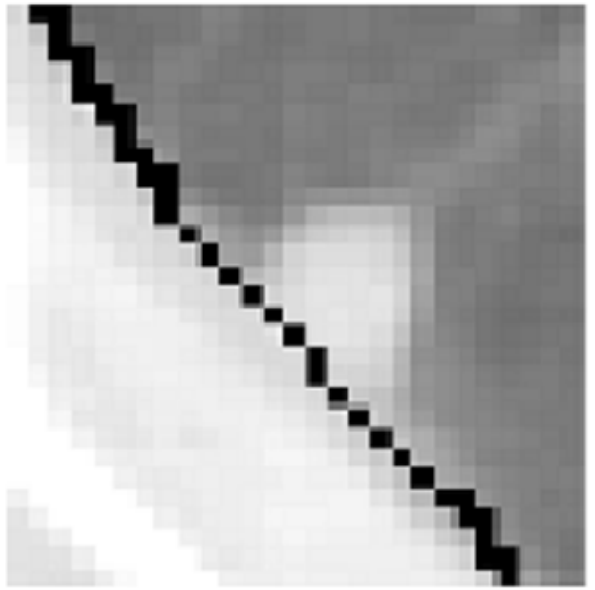


Black pixels:
Output of Boundary Following Algorithm.

Problem: Adjacent nodule excluded from Lung

After: Low curvature

Boundary is smoothed



Solution: Include nodule into lung region by
-searching for candidate boundary with appropriate curvature
-smoothing the boundary.

Evaluating a Curved Boundary in an Image: Curvature Measures

1) Parametric Definition

Intuition: *How does the angle of a tangent change when you move along the curve?*

Fast change -> high curvature

Slow change -> low curvature

Arc length s, tangential angle θ

Curvature = $d\theta/ds = \lim_{\delta s \to 0} \frac{\theta(s+\delta s) - \theta(s)}{\delta s}$

Discrete version: $\Delta\theta/\Delta s = (\theta_2 - \theta_1)/\Delta s$

Evaluating a Curved Boundary in an Image: Curvature Measures

2) Extrinsic Definition

Intuition: *The change of the tangential angle is written as a function of time and expressed by the derivatives of curve coordinates [x(t),(y(t)]*

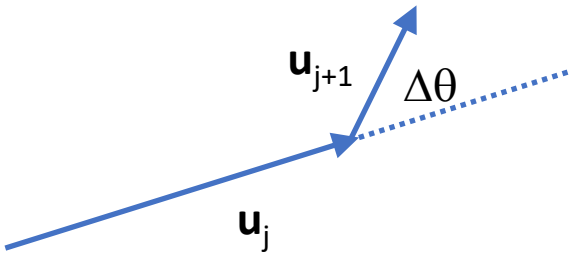
Arc length s, tangential angle θ

Curvature $K = d\theta/ds = (d\theta(t)/dt) / (ds/dt) =$

$$\frac{x'y'' - y'x''}{(x'^2 + y'^2)^{3/2}}$$

See derivation on blackboard

Evaluating a Curved Boundary in an Image: Curvature Measures



3) Consecutive Vectors on Curve

Intuition: *The change of the tangential angle is approximated by the angle between consecutive vectors*

$\mathbf{u}_j = (x_j - x_{j-1},\ y_j - y_{j-1})^T$

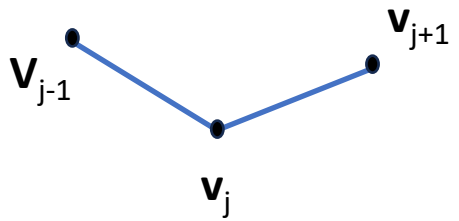
$\mathbf{u}_{j+1} = (x_{j+1} - x_j,\ y_{j+1} - y_j)^T$

$$\cos(\Delta\theta) = \frac{\mathbf{u}_j \cdot \mathbf{u}_{j+1}}{|\mathbf{u}_j|\ |\mathbf{u}_{j+1}|}$$

$\Delta\theta/\Delta s$ in $[0..\pi]/[0..1]$

Curvature in range $[0..\pi]$

Evaluating a Curved Boundary in an Image: Curvature Measures



4) Assuming Equally Spaced Vectors on Curve

Intuition: *The change of the arc length s is now 1.*

$$K = \left| \frac{d^2\mathbf{v}_j}{ds^2} \right|^2 \approx |(\mathbf{v}_{j+1}-\mathbf{v}_j)-(\mathbf{v}_j-\mathbf{v}_{j-1})|^2$$

$$= (x_{j-1}-2x_j+x_{j+1})^2 + (y_{j-1}-2y_j+y_{j+1})^2$$

Learning Objectives

- Can select and explain an appropriate “traditional” segmentation algorithms that uses analysis of greyscale histograms
- Can define N4 and N8 neighborhoods
- Can implement boundary following algorithm
- Can evaluate the similarity of a pair of curves based on their Hausdorff distance
- Can evaluate the curvature of an object boundary in an image using different measures of curvature