Computational Vision

Binary Images

Horn (Robot Vision)

Ch.3 : pages 46-53, and section 3.4.

Ch.4: pages 65-71

Binary Images

Binary Image b(x,y): Obtained from gray-level g(x,y) by thresholding.

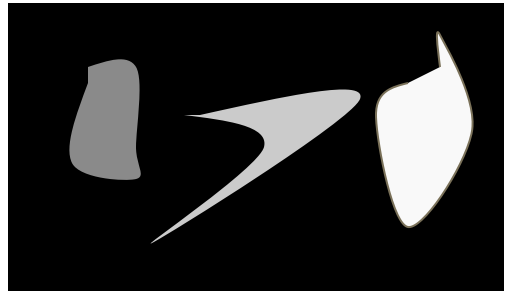
Characteristic Function:

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

1) Geometrical Properties
2) Discrete Binary Images
3) Multiple objects
4) Sequential and Iterative Processing.

From gray-scale to binary

g(x,y)



Gray-scale

<u>Given T</u>

For all pixels (x,y):

If g(x,y) < T

b(x,y)=0

else

b(x,y)=1

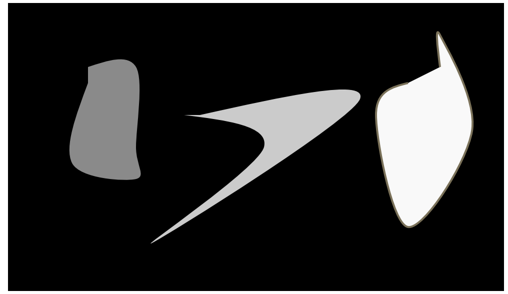
endif

b(x,y)

Binary

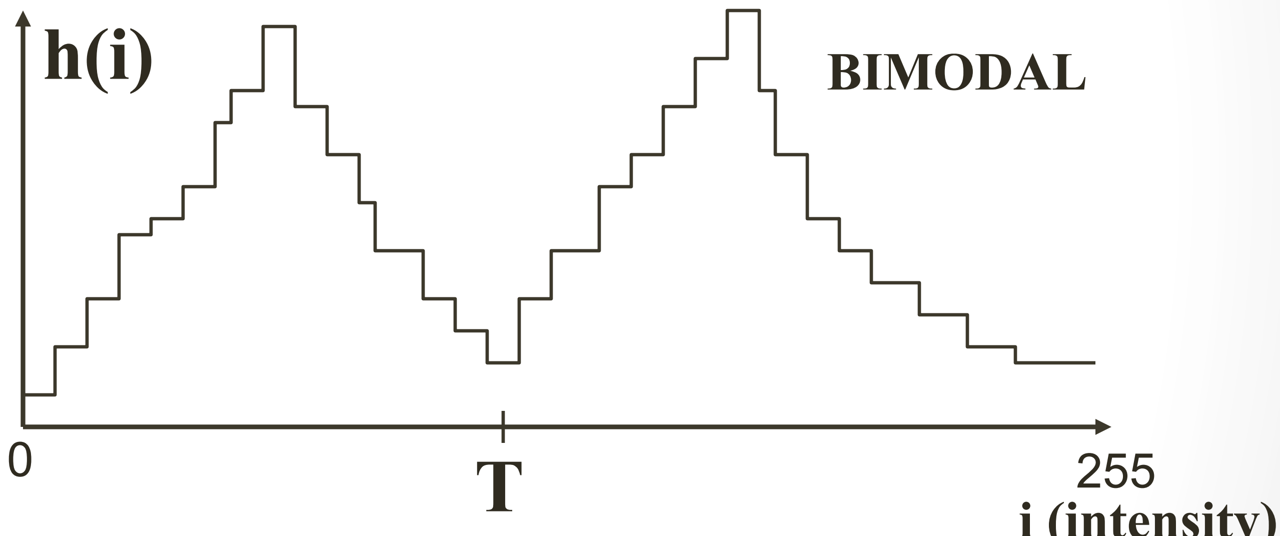
Selecting the Threshold (T)

g(x,y)

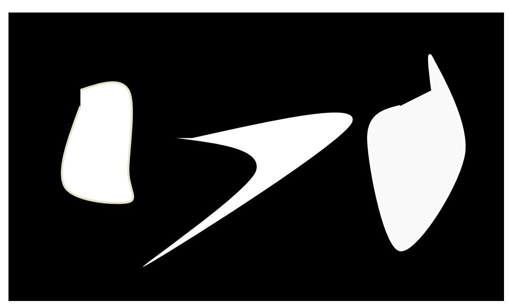


HISTOGRAM

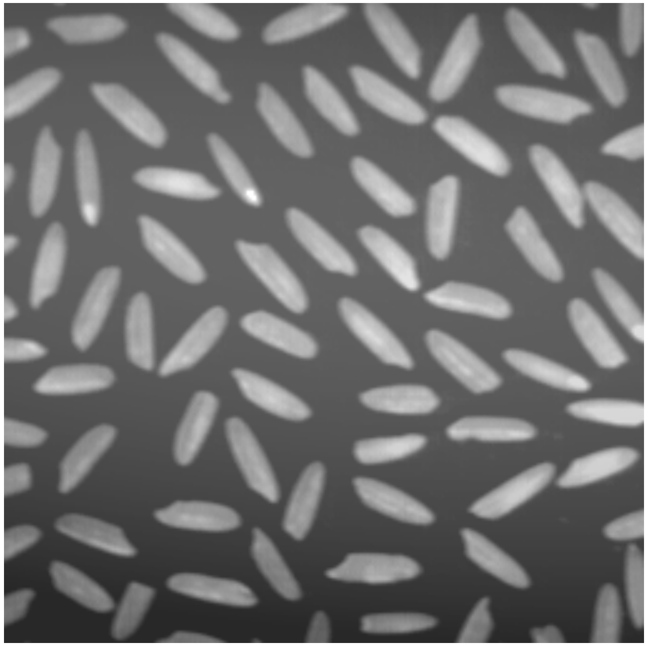
h(i) =
Number
of pixels with
value i,
i=0, …, 255
(8-bit image).



b(x,y)

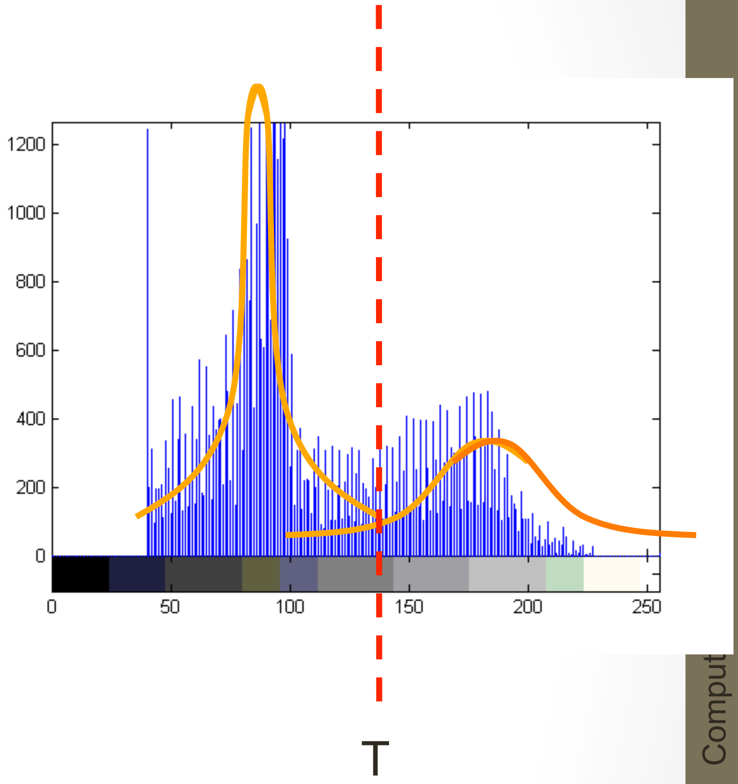


Histogram Thresholding:example



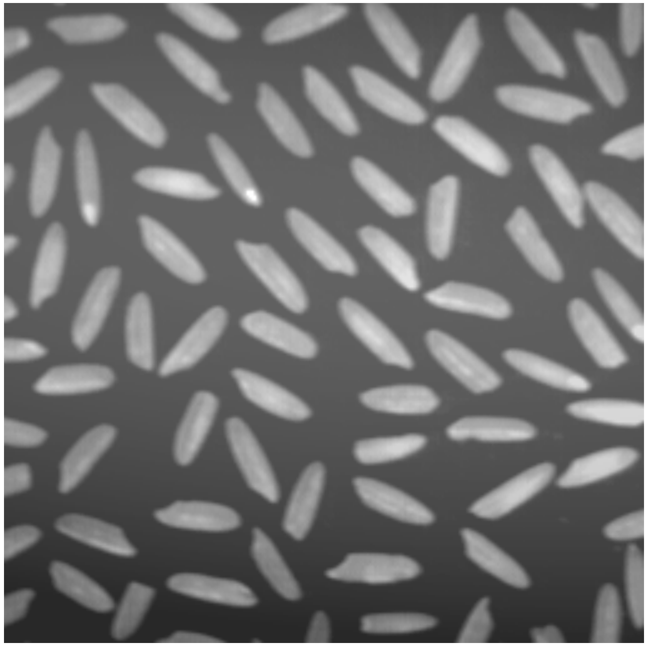
$g(x,y)$

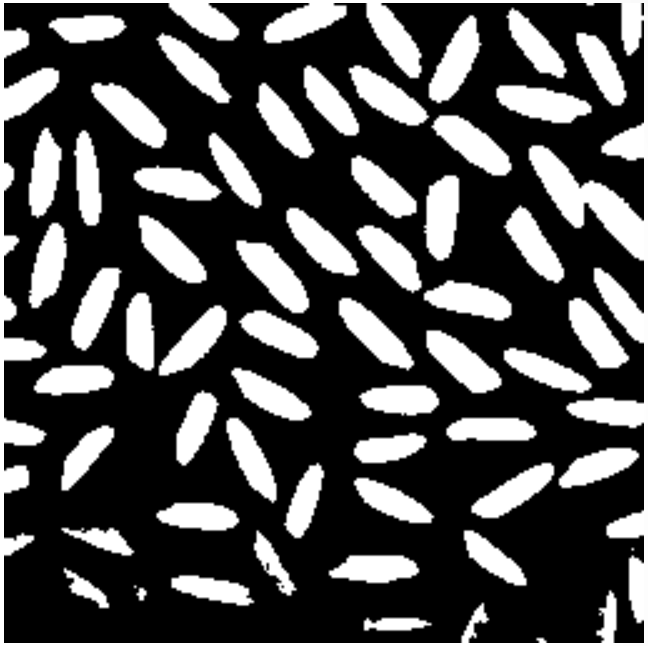
Gray-scale image



Histogram and threshold

Histogram Thresholding:example





g(x,y)

b(x,y)

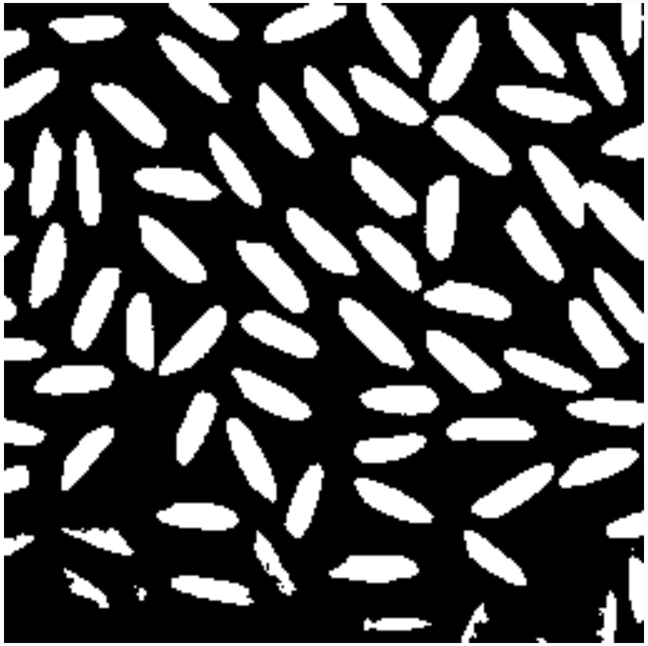
Gray-scale image

Binary image

Differences?

Binary Images

What can we do with a binary image?



b(x,y)

Binary Images

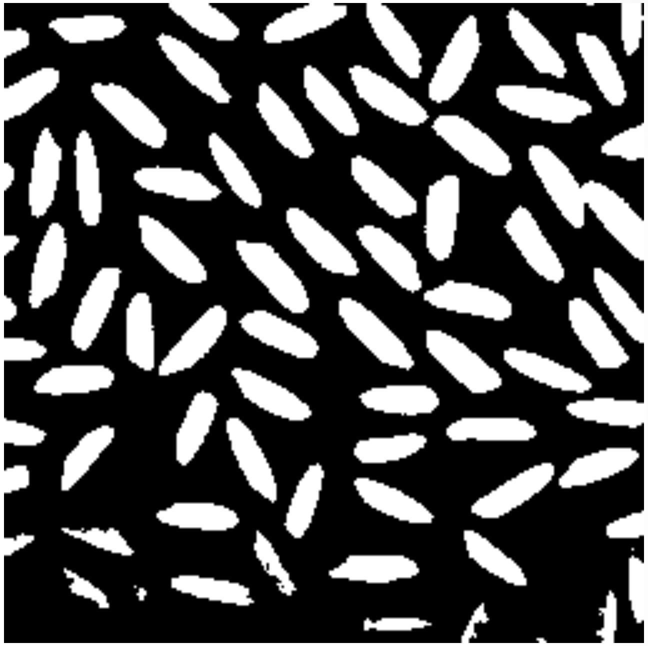
What can we do with a binary image?

Note that: We see only silhouettes.

-- We can count number of objects

-- We can measure
 -- size of objects
 -- orientation of objects
 -- …

Need some theory…



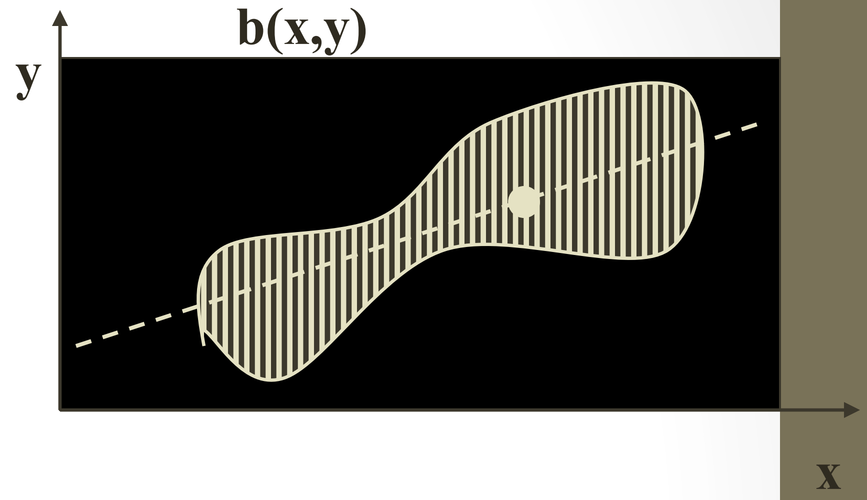
b(x,y)

Geometric Properties

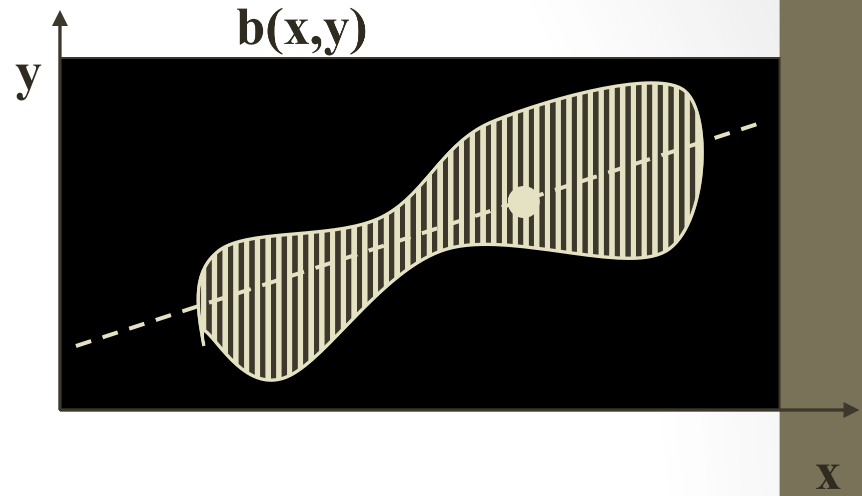
Assume:

1) b(x,y) is continuous

2) Only one object



Geometric Properties



Assume:

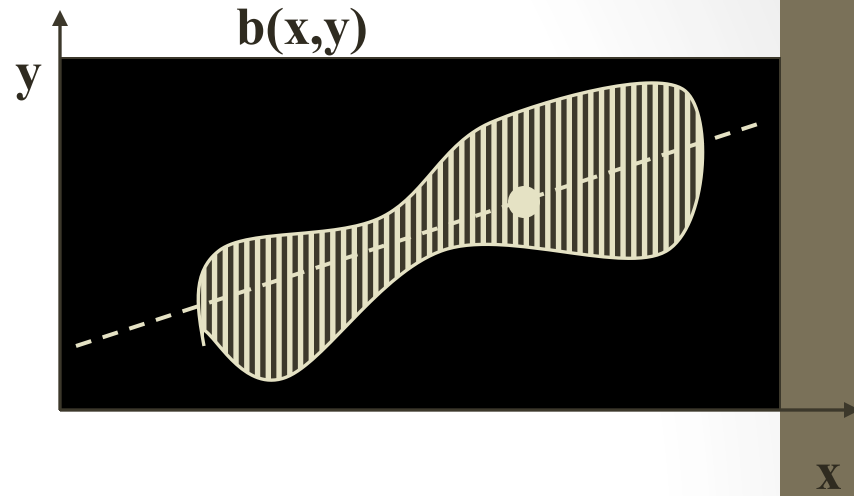
1) **b(x,y) is continuous**
2) **Only one object**

Area:

$$A = \iint_I b(x,y)\,dx\,dy$$

(Zeroth Moment)

Geometric Properties



Assume:

1) b(x,y) is continuous
2) Only one object

Area:

$$A = \iint_I b(x, y)\,dx\,dy$$

(Zeroth Moment)

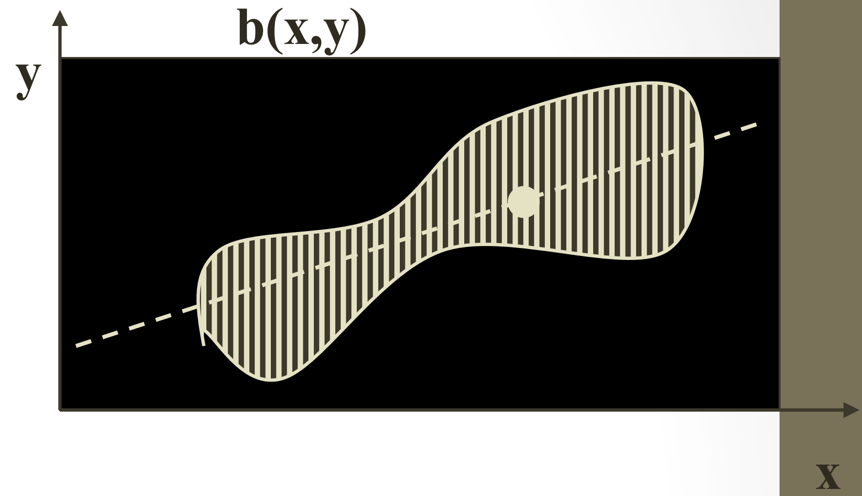
Position: "Center" (x,y) of Area

$$\bar{x} = (1/A)\iint_I x * b(x, y)\,dx\,dy$$

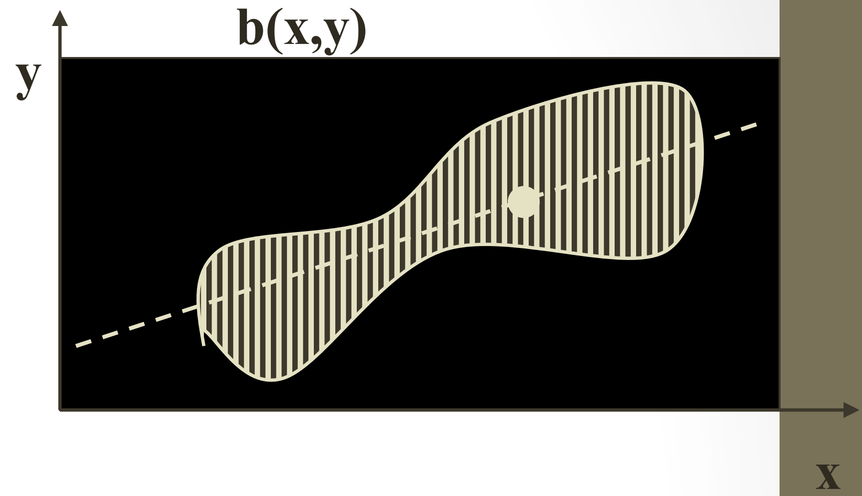
$$\bar{y} = (1/A)\iint_I y * b(x, y)\,dx\,dy$$

(First Moment)

Orientation?



Orientation?



Difficult to define!

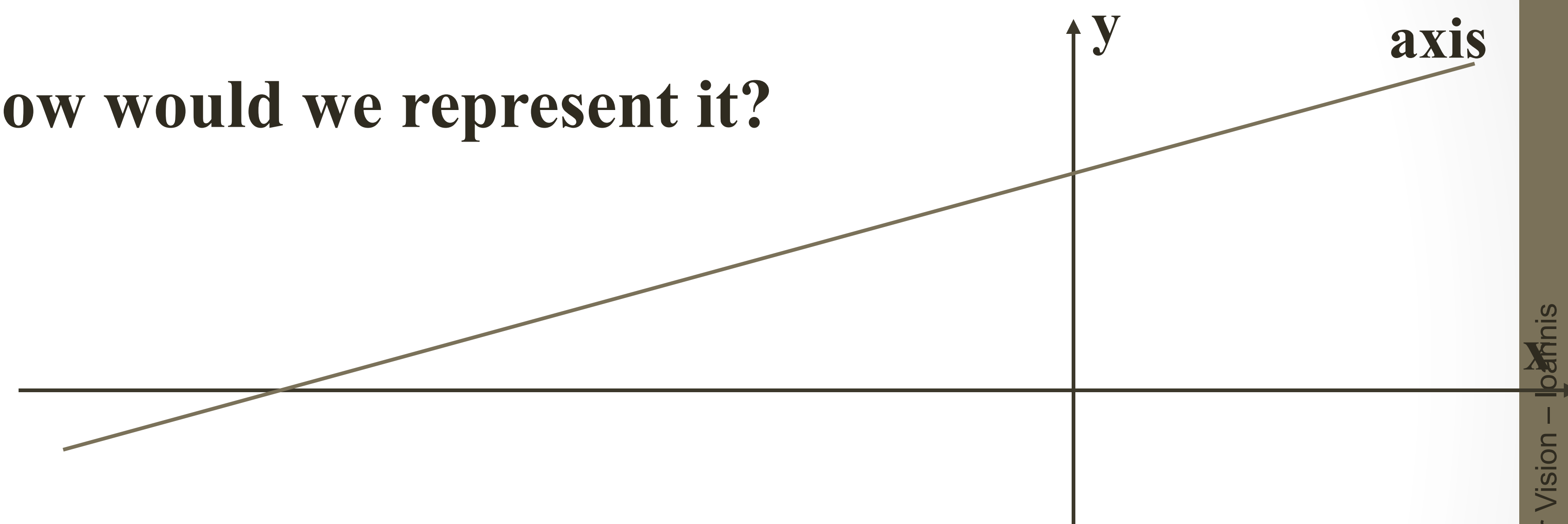
It is a line.

It is easy to imagine for elongated objects.

We will provide an intuitive formulation.

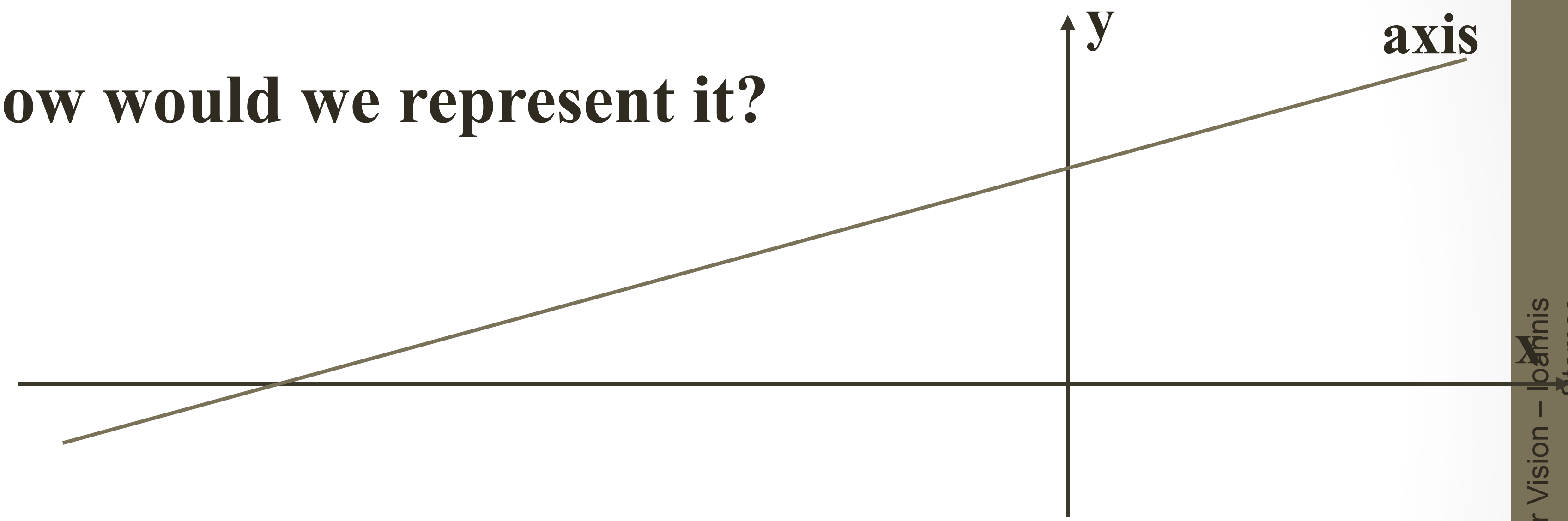
Line in 2D

How would we represent it?



Line in 2D

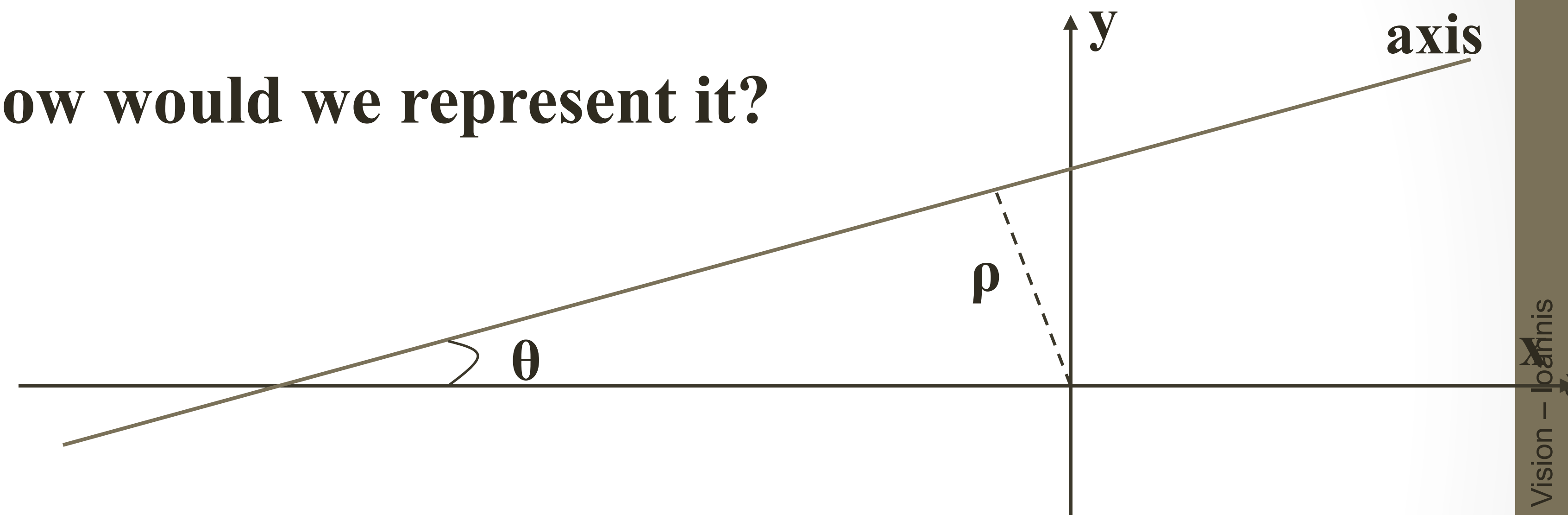
How would we represent it?



Equation of axis: y=mx+b ?   0<= m <= infinity

Line in 2D

How would we represent it?

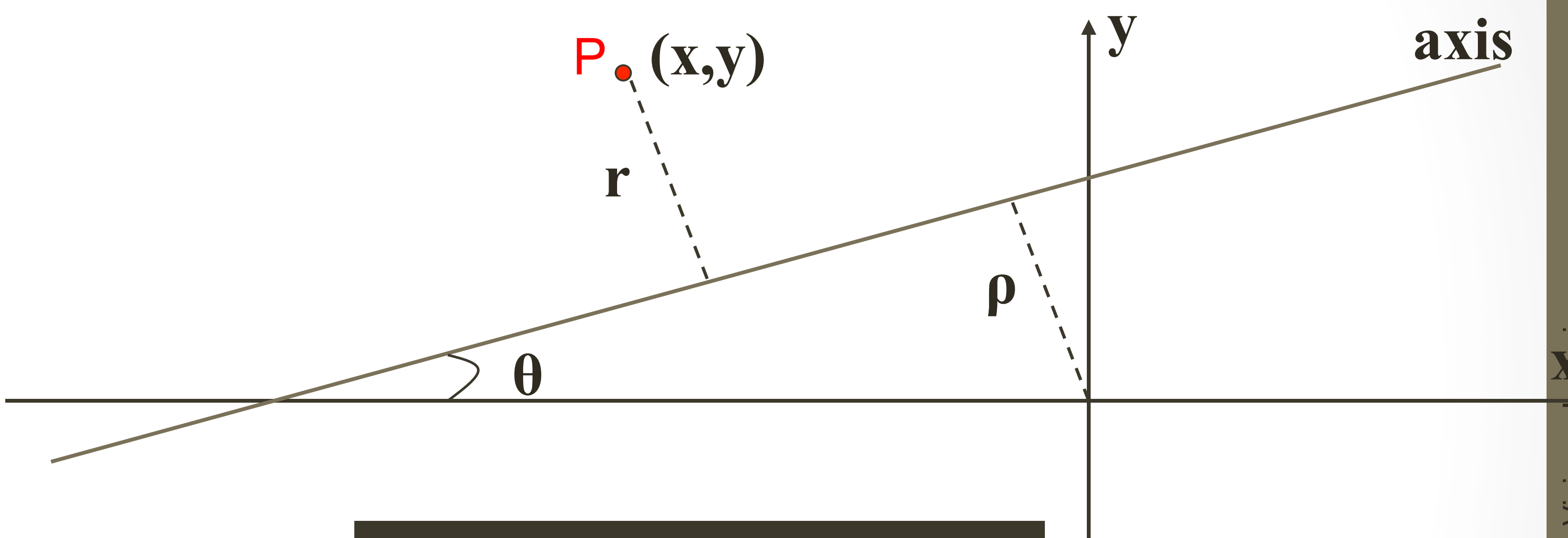


We representing the axis with angle θ and distance from origin ρ
So an axis is defined by the pair (ρ, θ).

$$\text{Equation of line: } x \sin\theta - y\cos\theta + \rho = 0$$

Orientation

We use axis of least second moment



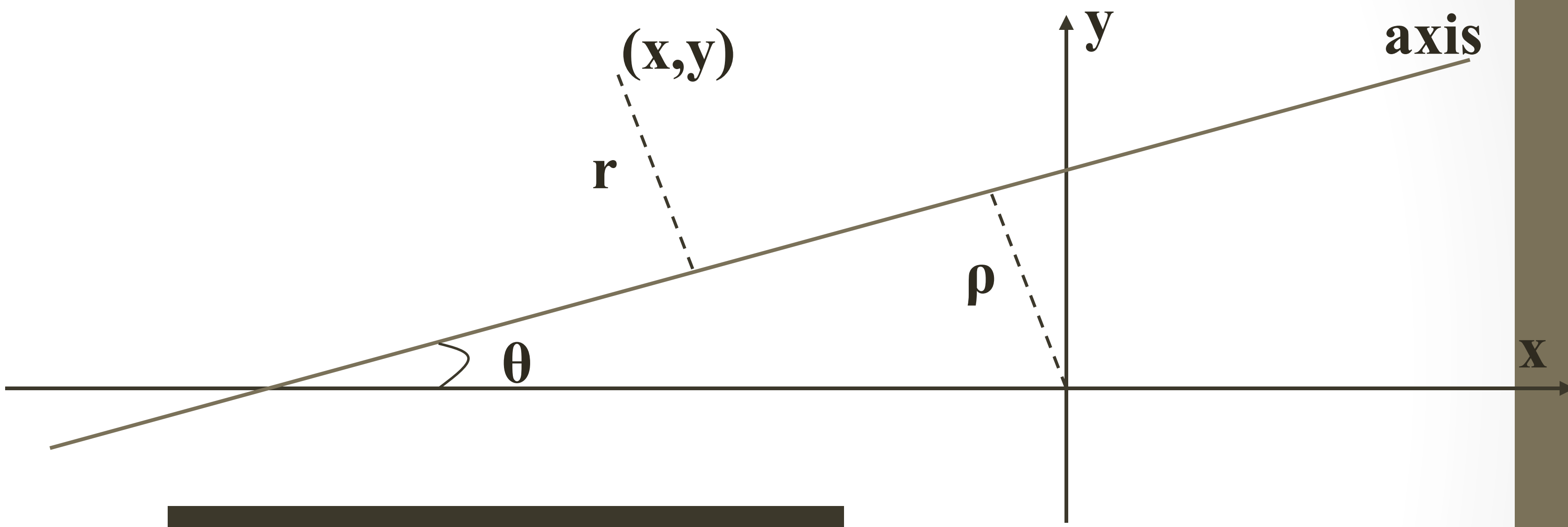
Minimize:

$$E = \iint_I r^2 b(x,y)\,dx\,dy$$

Find line that produces the minimum sum of squared distances from all points on the object.

Orientation

Difficult to define!
We use axis of least second moment
For mass distribution: Axis of minimum Inertia.



Minimize: $$E = \iint_I r^2 b(x,y)\,dx\,dy$$

Equation of Axis: y=mx+b ???   0<= m <= infinity

We use: $$x\sin(\theta) - y\cos(\theta) + \rho = 0$$ **(ρ, θ) finite.**

Find (ρ, θ) that minimize E for a given b(x,y)

We can show that: $$r^2 = (x\sin(\theta) - y\cos(\theta) + \rho)^2$$

So: $$E = \iint_I (x\sin(\theta) - y\cos(\theta) + \rho)^2 b(x,y)\,dxdy$$

We can show that: $$r^2 = (x\sin(\theta) - y\cos(\theta) + \rho)^2$$

So: $$E = \iint_I (x\sin(\theta) - y\cos(\theta) + \rho)^2 b(x,y)\,dx\,dy$$

Using <u>dE/dρ=0</u> we get: $$A(\bar{x}\sin(\theta) - \bar{y}\cos(\theta) + \rho) = 0$$

We can show that: $$r^2 = (x\sin(\theta) - y\cos(\theta) + \rho)^2$$

So: $$E = \iint_I (x\sin(\theta) - y\cos(\theta) + \rho)^2 b(x,y)\,dx\,dy$$

Using <u>dE/dρ=0</u> we get: $$A(\bar{x}\sin(\theta) - \bar{y}\cos(\theta) + \rho) = 0$$

Note: Axis passes through center ($\bar{x}$,$\bar{y}$)!!

So, change coordinates: $x' = x - \bar{x}$, $y' = y - \bar{y}$

We can show that: $$r^2 = (x\sin(\theta) - y\cos(\theta) + \rho)^2$$

So: $$E = \iint_I (x\sin(\theta) - y\cos(\theta) + \rho)^2 b(x,y)\,dx\,dy$$

Using <u>dE/dρ=0</u> we get: $$A(\bar{x}\sin(\theta) - \bar{y}\cos(\theta) + \rho) = 0$$

Note: Axis passes through center $(\bar{x},\bar{y})$!!

So, change coordinates: $x' = x - \bar{x}$, $y' = y - \bar{y}$

We get:

$$E = a\sin^2(\theta) - b\sin(\theta)\cos(\theta) + c\cos^2(\theta)$$

$$a = \iint_I (x')^2 b(x,y)\,dx'\,dy'$$

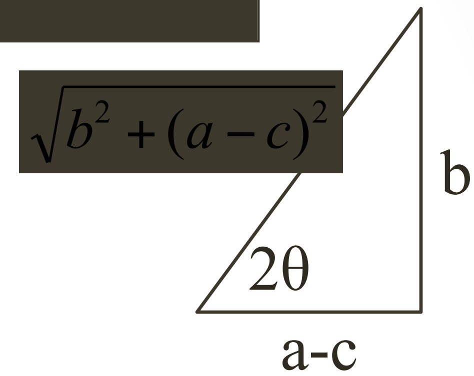
$$b = 2\iint_I (x'y')b(x,y)\,dx'\,dy'$$

$$c = \iint_I (y')^2 b(x,y)\,dx'\,dy'$$

Second Moments w.r.t. $(\bar{x}, \bar{y})$

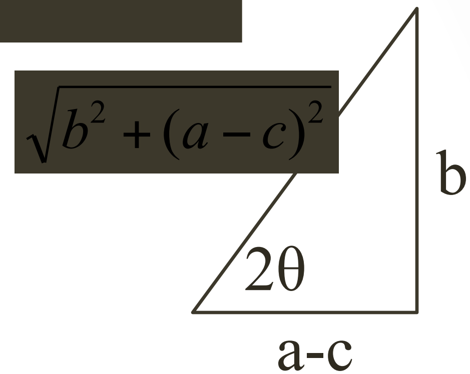
$$E = a\sin^2(\theta) - b\sin(\theta)\cos(\theta) + c\cos^2(\theta)$$

Using $dE/d\theta=0$, we get: $\tan(2\theta)=b/(a-c)$



$$E = a\sin^2(\theta) - b\sin(\theta)\cos(\theta) + c\cos^2(\theta)$$

Using dE/dθ=0, we get: tan(2θ)=b/(a-c)



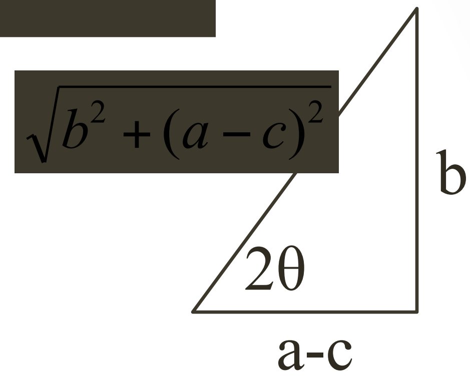
So:

$$\sin(2\theta) = \pm b / \sqrt{b^2 + (a-c)^2}$$

$$\cos(2\theta) = \pm (a-c) / \sqrt{b^2 + (a-c)^2}$$

$$E = a\sin^2(\theta) - b\sin(\theta)\cos(\theta) + c\cos^2(\theta)$$

Using $dE/d\theta=0$, we get: $\tan(2\theta)=b/(a-c)$



So:

$$\sin(2\theta) = \pm b / \sqrt{b^2 + (a-c)^2}$$

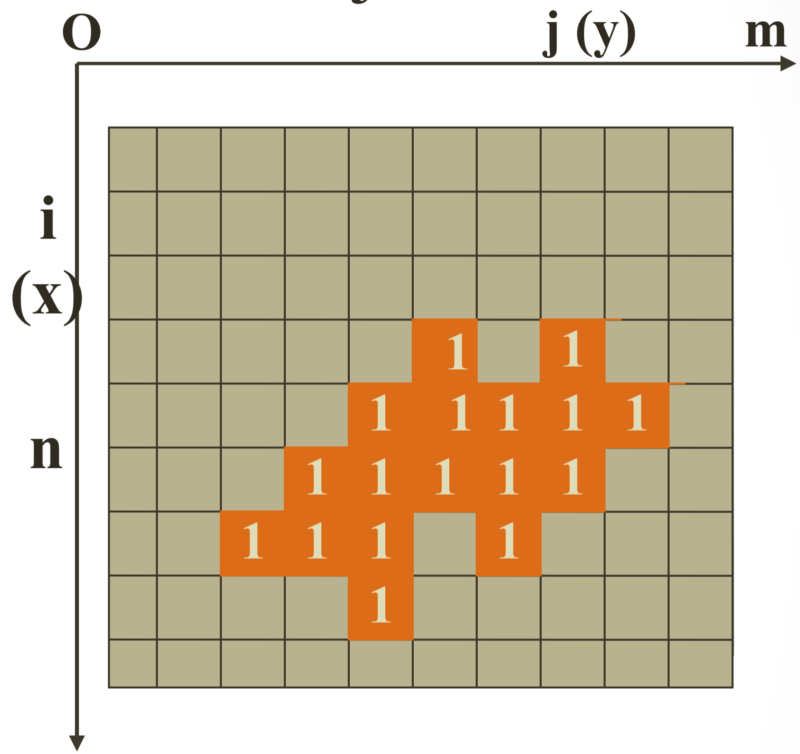
$$\cos(2\theta) = \pm (a-c) / \sqrt{b^2 + (a-c)^2}$$

Solutions with positive sign may be used to find θ that minimizes E (why??).

Emin/Emax -> (ROUDNESS of the OBJECT).

Discrete Binary Images

bij: value at pixel in row i & column j.



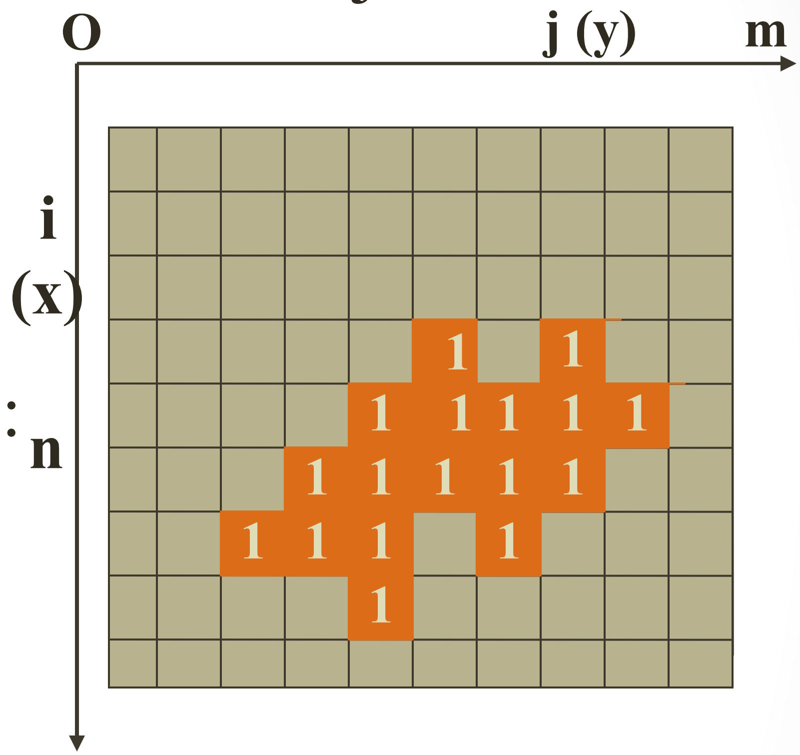
Discrete Binary Images

bij: value at pixel in row i & column j.

Assume pixel area is 1.

Area:

$$A = \sum_{i=1}^{n}\sum_{j=1}^{m} b_{ij}$$



Position (Center of Area):

$$\bar{x} = (1/A)\sum_{i=1}^{n}\sum_{j=1}^{m} i b_{ij}$$

$$\bar{y} = (1/A)\sum_{i=1}^{n}\sum_{j=1}^{m} j b_{ij}$$

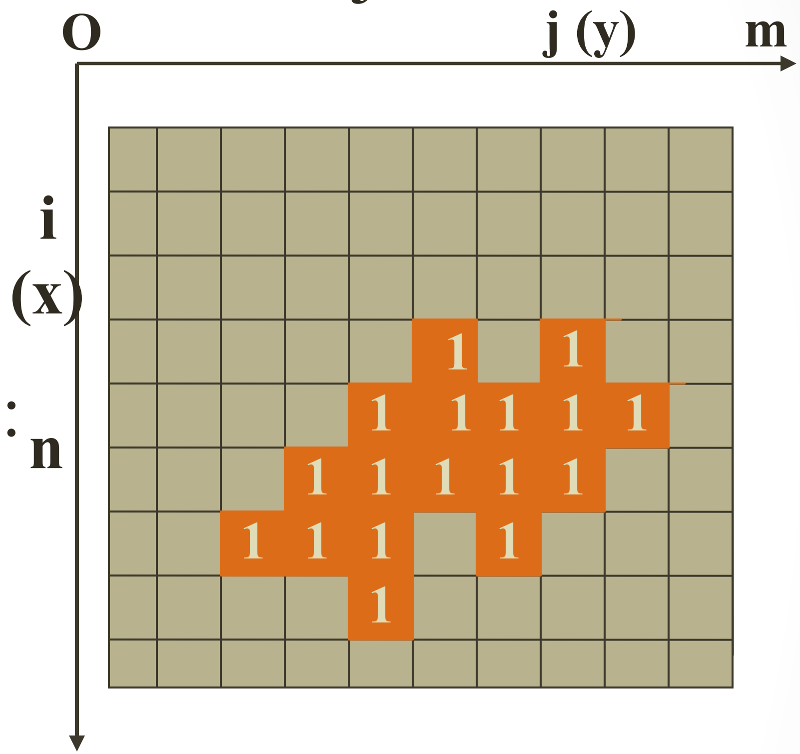
Discrete Binary Images

bij: value at pixel in row i & column j.

Assume pixel area is 1.

Area:

$$A = \sum_{i=1}^{n} \sum_{j=1}^{m} b_{ij}$$



Position (Center of Area):

$$\bar{x} = (1/A) \sum_{i=1}^{n} \sum_{j=1}^{m} i b_{ij}$$

$$\bar{y} = (1/A) \sum_{i=1}^{n} \sum_{j=1}^{m} j b_{ij}$$

Second Moments.

$$a' = \sum_{i=1}^{n} \sum_{j=1}^{m} i^2 b_{ij}, b' = 2 \sum_{i=1}^{n} \sum_{j=1}^{m} ij\ b_{ij}, c' = \sum_{i=1}^{n} \sum_{j=1}^{m} j^2 b_{ij}$$

Discrete Binary Images (cont)

Note: a', b', c' are second moments w.r.t ORIGIN
a,b,c (w.r.t "center") can be found from
($a', b', c', \bar{x}, \bar{y}, A$).

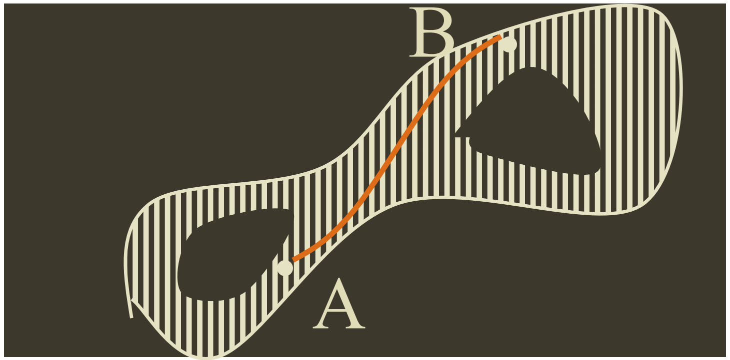
Note: UPDATE ($a', b', c', \bar{x}, \bar{y}, A$) during RASTER SCAN

Multiple Objects

Need to SEGMENT image into separate COMPONENTS (regions).

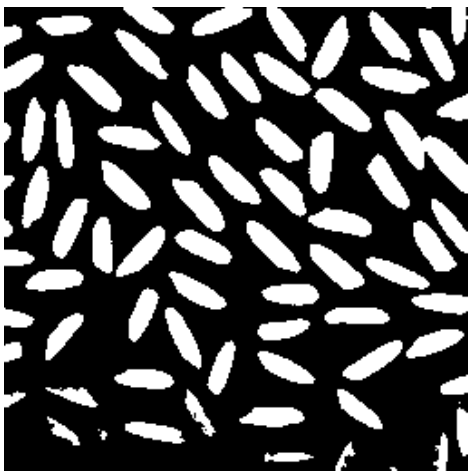
Connected Component:

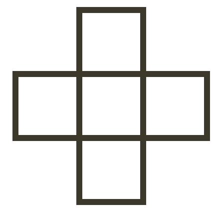
Maximal Set of Connected Points:



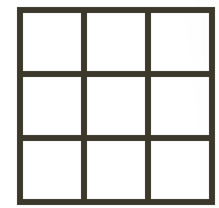
Points A and B are connected: Path exists between A and B along which b(x,y) is constant.

Connected Components



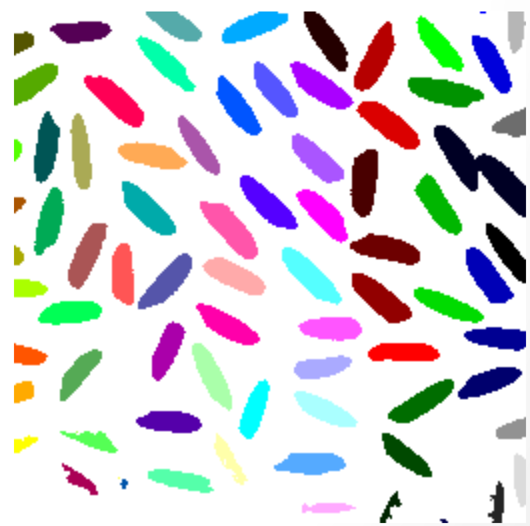


4-way connected



8-way connected

Label all pixels that are connected



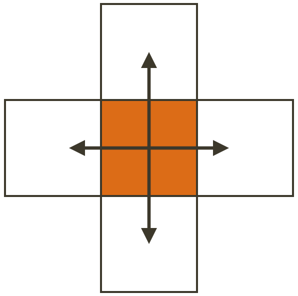
Region Growing Algorithm (Connected Component Labeling)

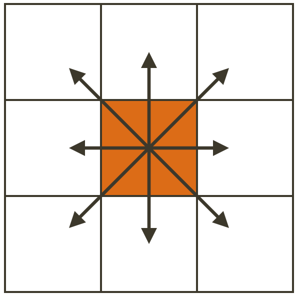
- Start with "seed" point where bij=1.
- Assign LABEL to seed point.
- Assign SAME LABEL to its NEIGHBORS (b=1).
- Assign SAME LABEL to NEIGHBORS of NEIGHBORS.

Terminates when a component is completely labeled.
Then pick another UNLABELED seed point.

What do we mean by NEIGHBORS?

Connectedness





8-Connectedness (8-c)

4-Connectedness (4-c)

Neither is perfect!

What do we mean by NEIGHBORS?

Jordan's Curve Theorem: Closed curve -> 2 connected regions

0	**1**	**0**
1	**0**	**1**
0	**1**	**0**

(4-c)

B1	**O1**	**B1**
O2	**B2**	**O3**
B1	**O4**	**B1**

Hole without closed curve!

What do we mean by NEIGHBORS?

Jordan's Curve Theorem: Closed curve -> 2 connected regions

0	1	0
1	0	1
0	1	0

(4-c)

B1	O1	B1
O2	B2	O3
B1	O4	B1

Hole without closed curve!

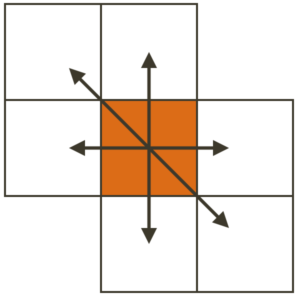
(8-c)

B	O	B
O	B	O
B	O	B

Connected background with closed ring!

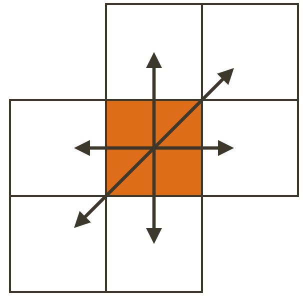
Solution: Introduce Assymetry

Use:



(a)

or



(b)

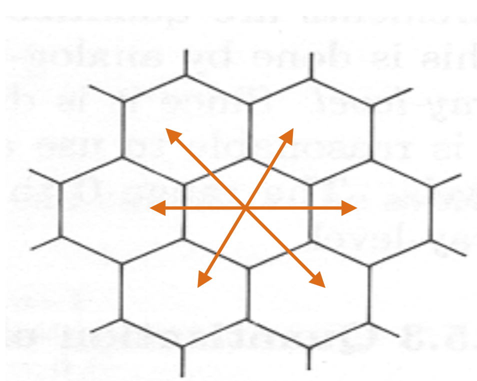
Using (a)

0	1	0
1	0	1
0	1	0

B	O1	B
O2	B	O1
B	O2	B

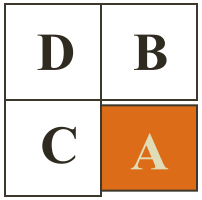
Two separate line segments.

Hexagonal Tesselation



Above assymetry makes SQUARE grid like HEXAGONAL grid.

Sequential Labeling Algorithm



Raster Scan

<u>Note: B,C,D are already labeled</u>

Sequential Labeling Algorithm

D	B
C	A

Raster Scan

<u>Note: B,C,D are already labeled</u>

a.

X	X
X	0

Label(A) = background

b.

D	X
X	1

Label(A) = label(D)

Sequential Labeling Algorithm

D	B
C	A

Raster Scan →

Note: B,C,D are already labeled

c.

0	0
C	1

Label(A) = label(C)

d.

0	B
0	1

Label(A) = label(B)

d.

0	B
C	1

If Label(B) = label(C), then Label(A)=Label(B)= Label(C)

Sequential Labeling Algorithm

D	B
C	A

Raster Scan →

Note: B,C,D are already labeled

a.

X	X
X	0

Label(A) = background

b.

D	X
X	1

Label(A) = label(D)

c.

0	0
C	1

Label(A) = label(C)

d.

0	B
0	1

Label(A) = label(B)

d.

0	B
C	1

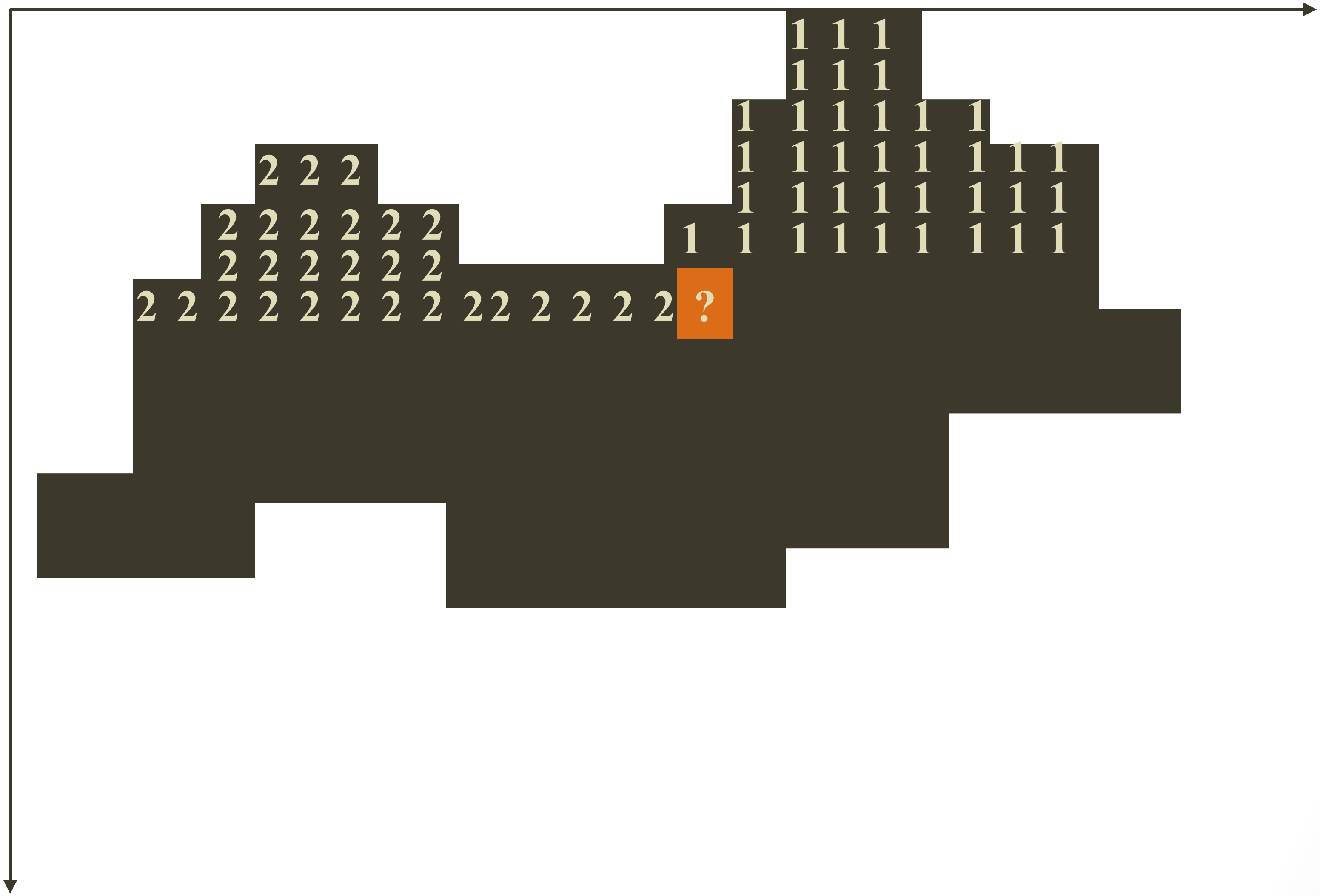
If Label(B) = label(C), then Label(A)=Label(B)= Label(C)

Sequential Labeling (Cont.)

What if B & C are labeled but label(B) [redacted] label(C) ?

Sequential Labeling (Cont.)

What if B & C are labeled but label(B) ≠ label(C) ?



Sequential Labeling (Cont.)

Solution: Let: Label(A)=Label(B) =2
& create an EQUIVALENCE TABLE.

Resolve Equivalences in SECOND PASS.

$2 \equiv 1$
$7 \equiv 3,6,4$
…

Next Class

- Edge Detection…