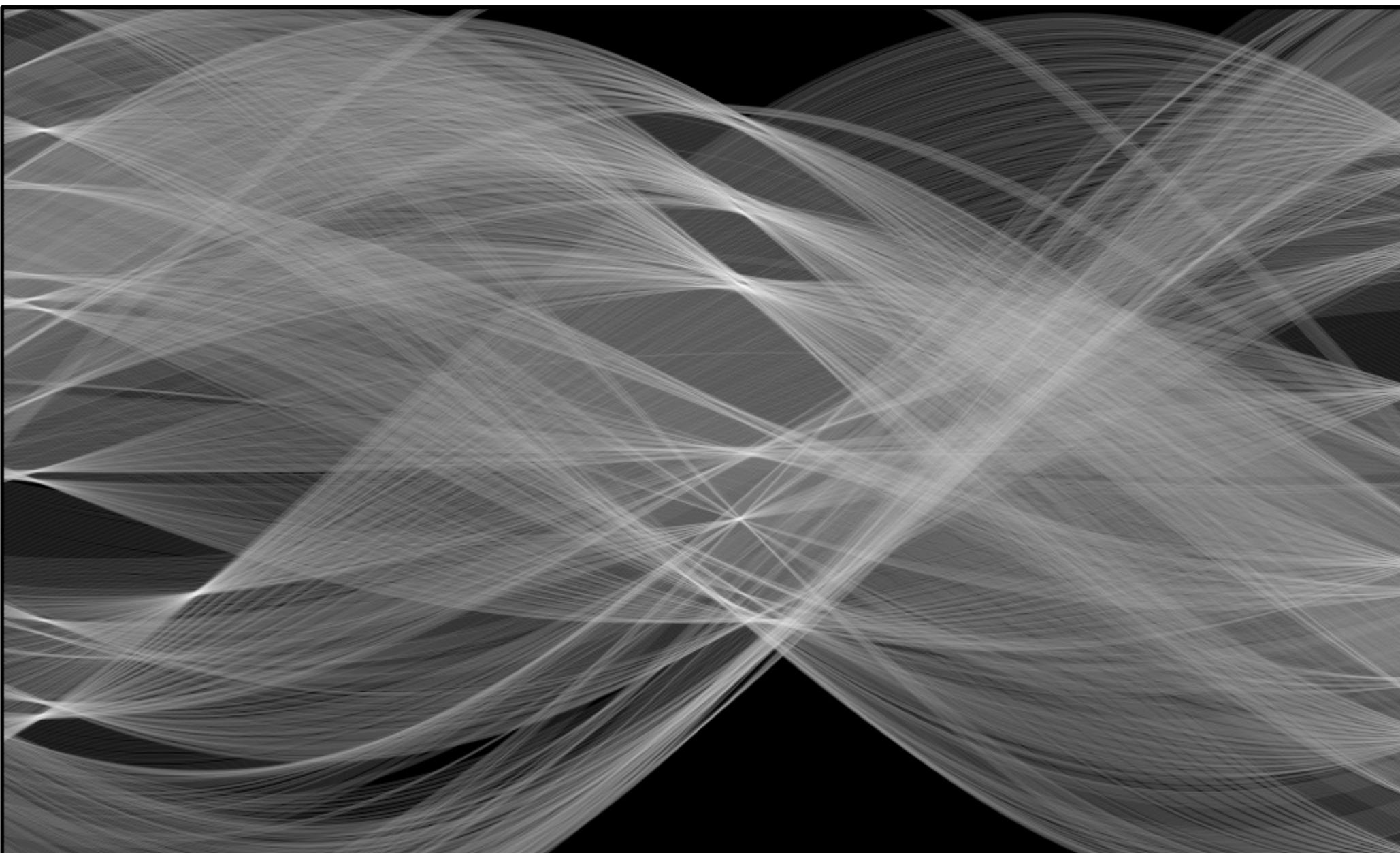


# Hough transform



# Overview of today's lecture

- Finding boundaries.
- Line fitting.
- Line parameterizations.
- Hough transform.
- Hough circles.
- Some applications.

# Slide credits

Most of these slides were adapted from:

- Kris Kitani (15-463, Fall 2016).

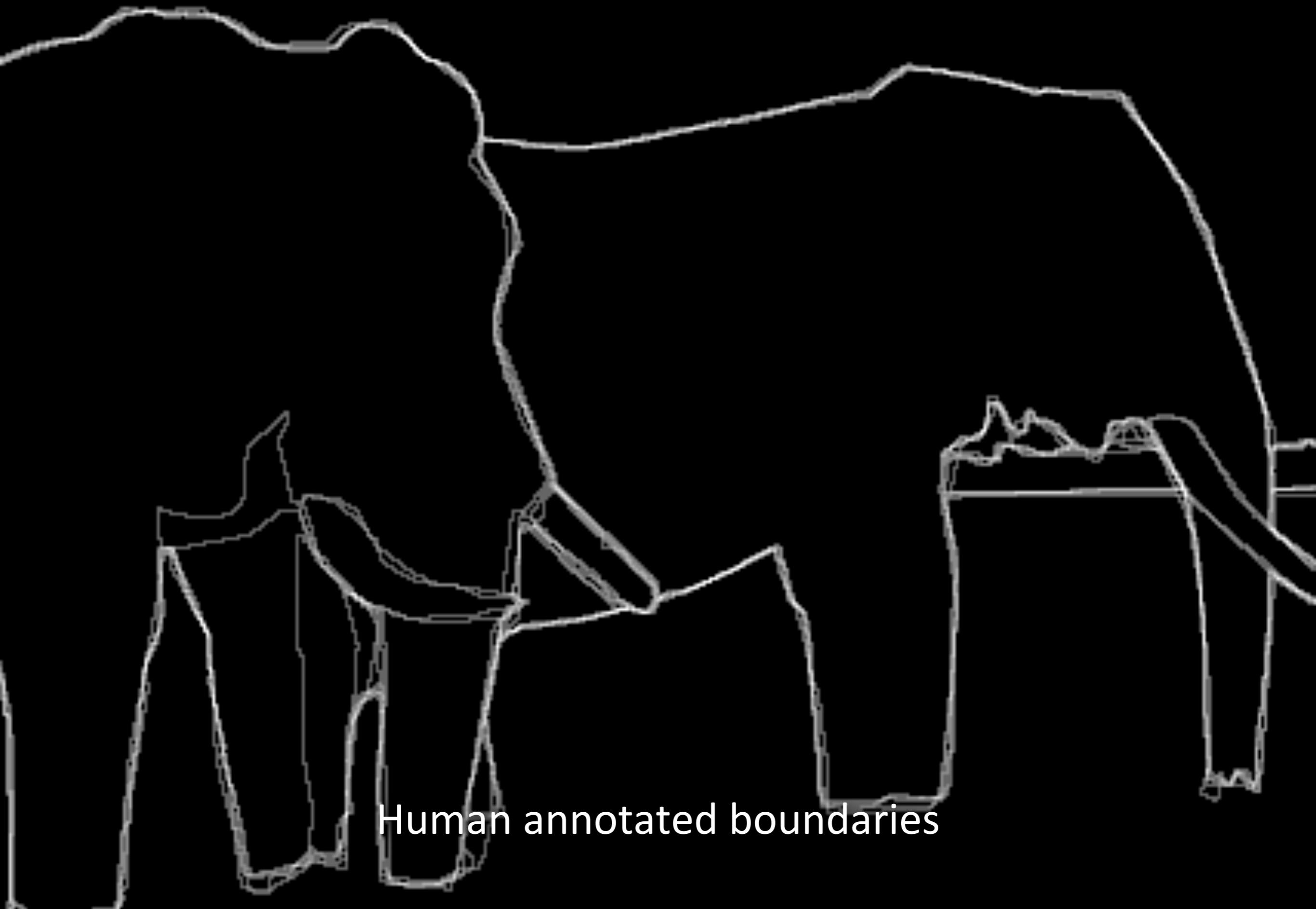
Some slides were inspired or taken from:

- Fredo Durand (MIT).
- James Hays (Georgia Tech).

# Finding boundaries



Where are the object boundaries?



Human annotated boundaries



Edge detection



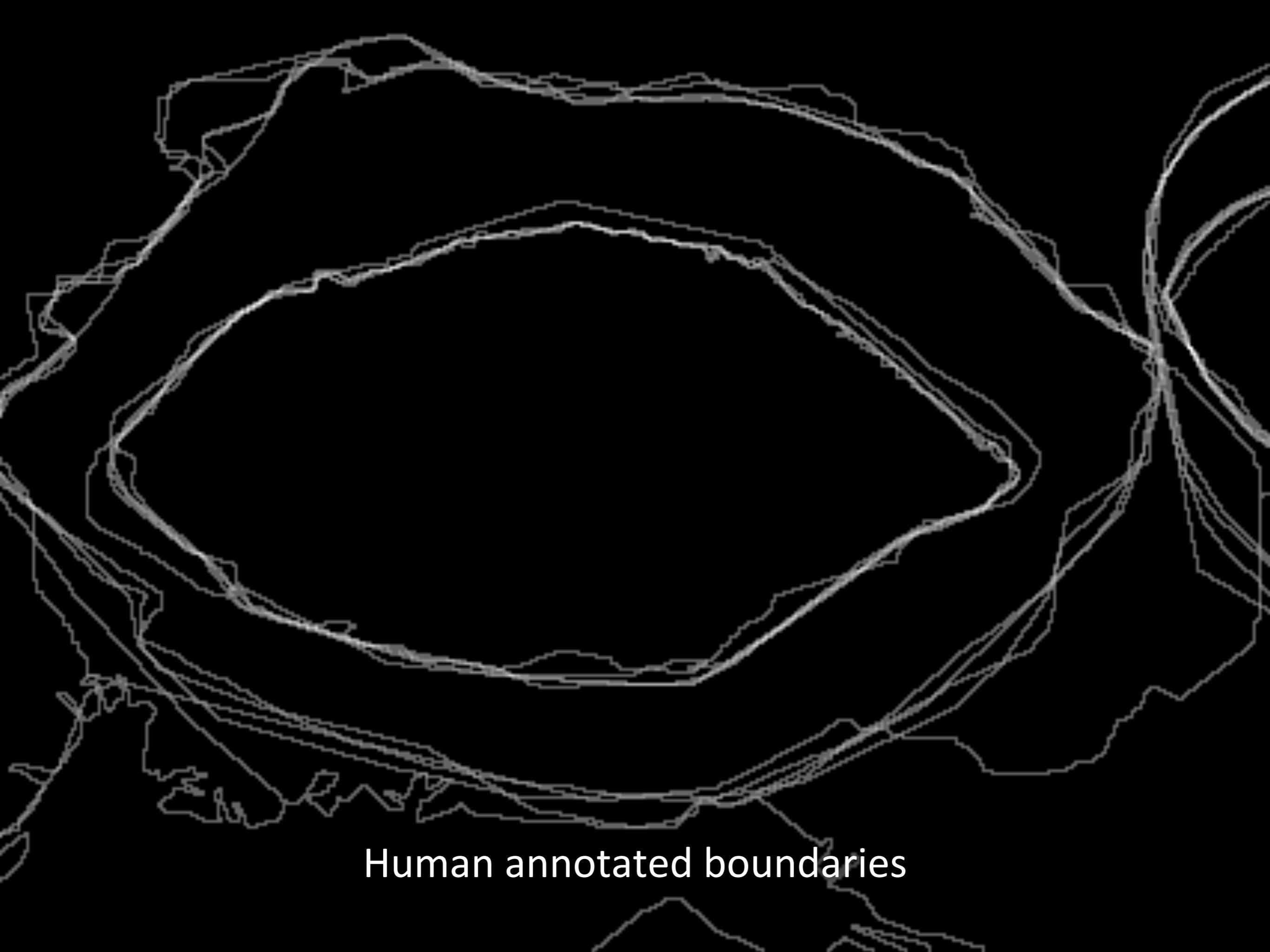
Multi-scale edge detection



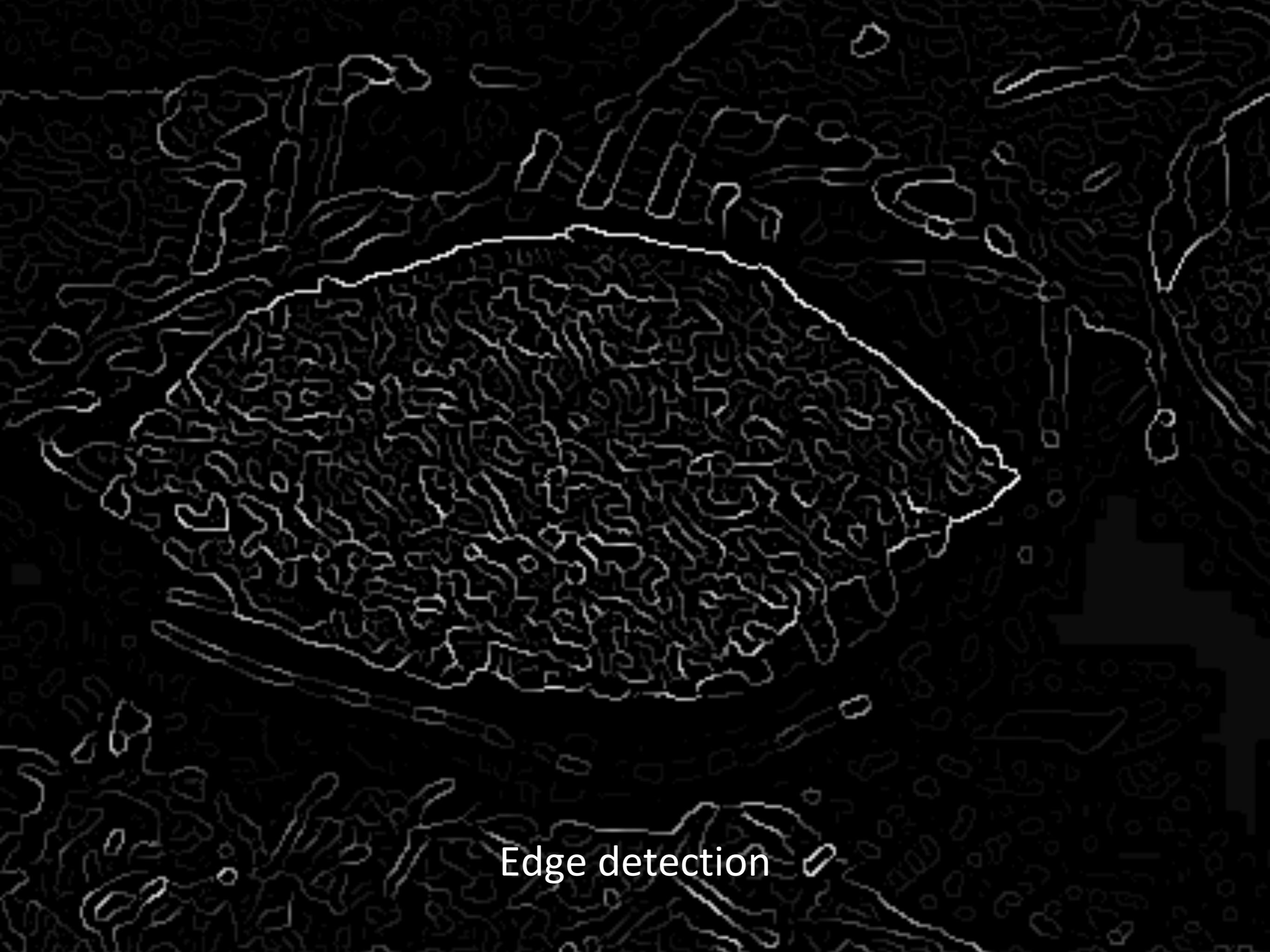
Edge strength does not necessarily correspond to our perception of boundaries



Where are the object boundaries?



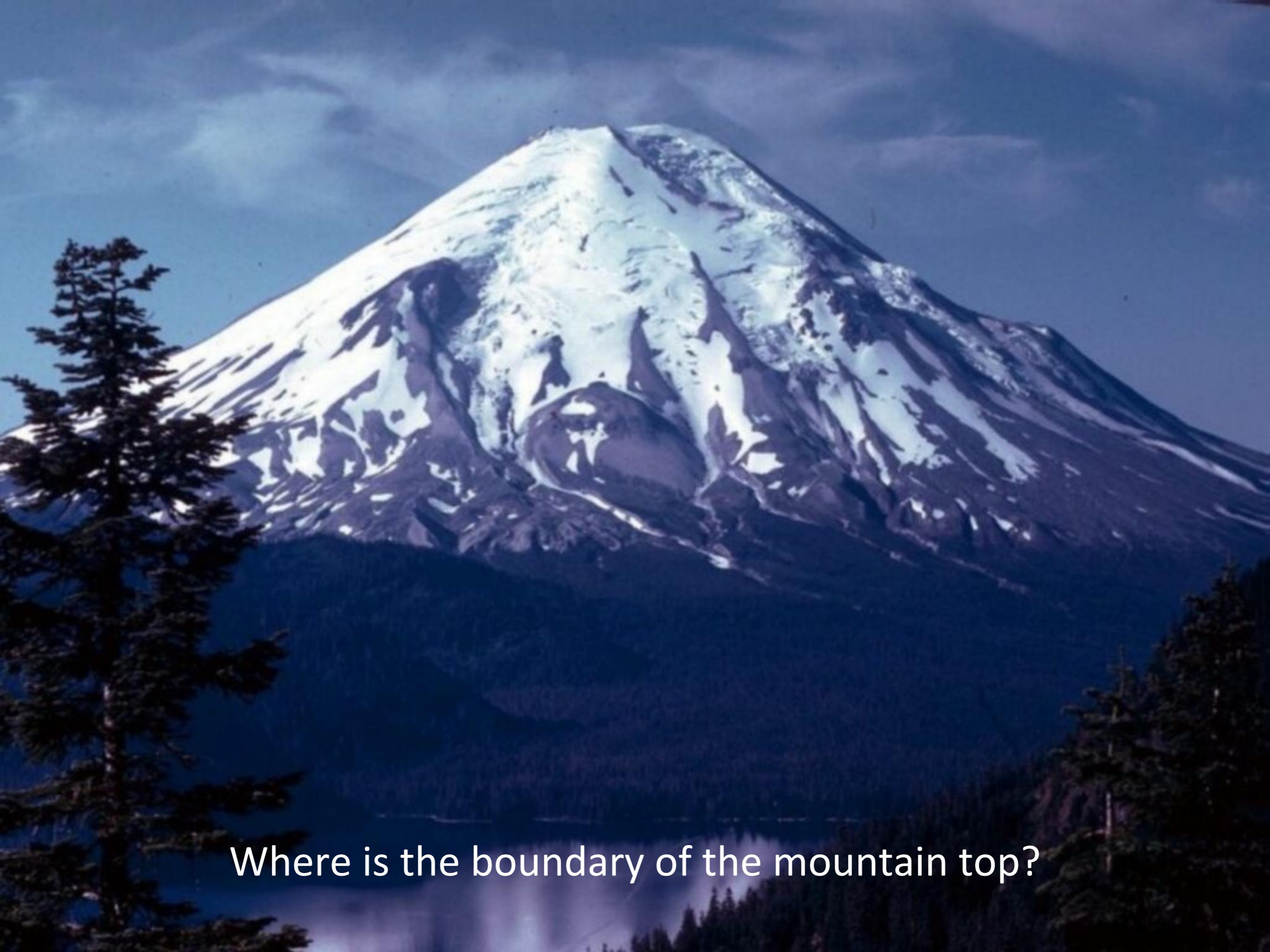
Human annotated boundaries



Edge detection

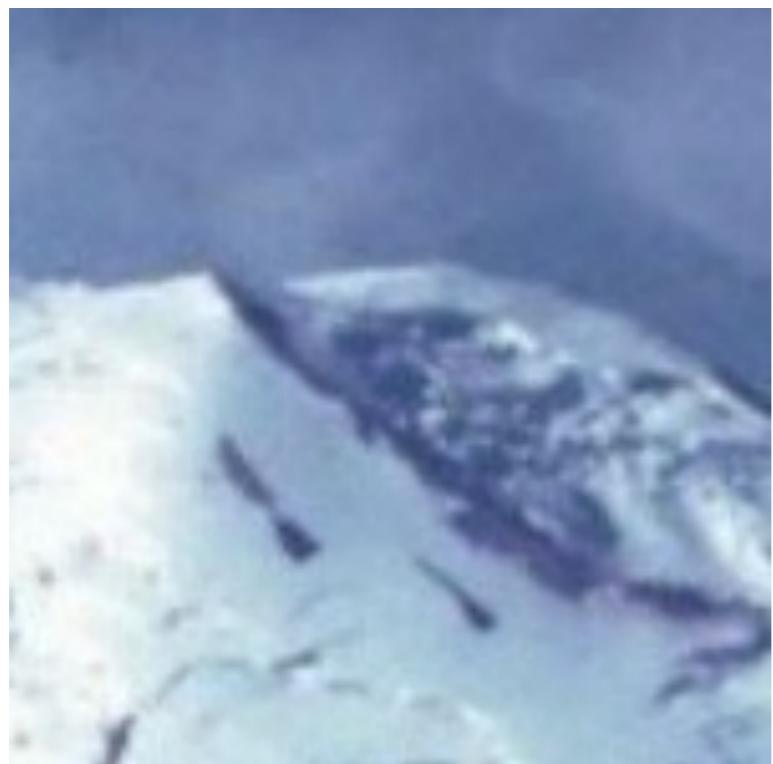
A detailed, high-magnification image of biological tissue. The tissue is composed of various types of cells, some appearing pinkish-red and others greenish-blue. A prominent feature is a network of thin, white, branching vessels, likely blood vessels, which are more clearly visible in the lower half of the image. The overall texture is somewhat granular and complex.

Defining boundaries are hard for us too

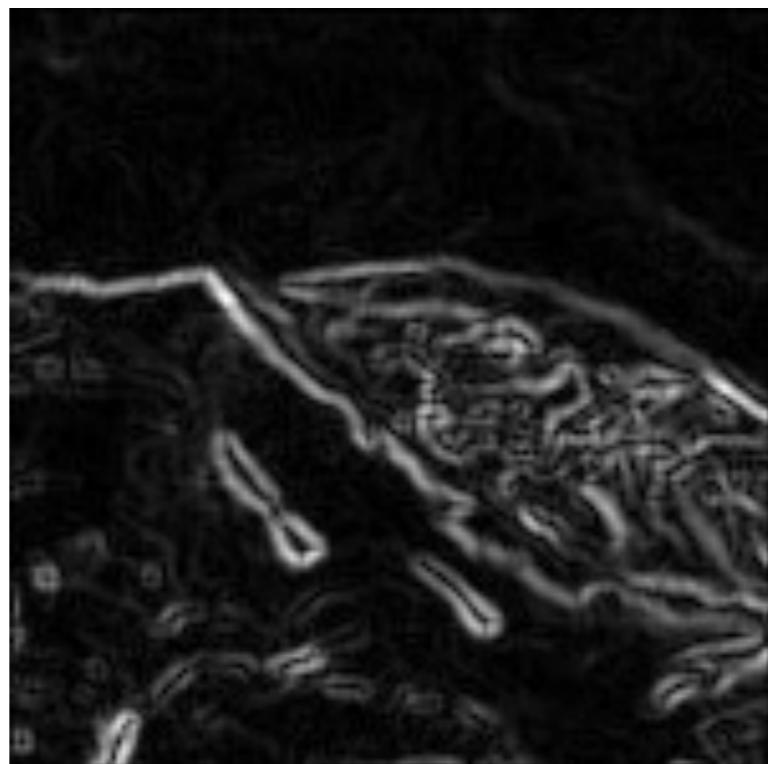


Where is the boundary of the mountain top?

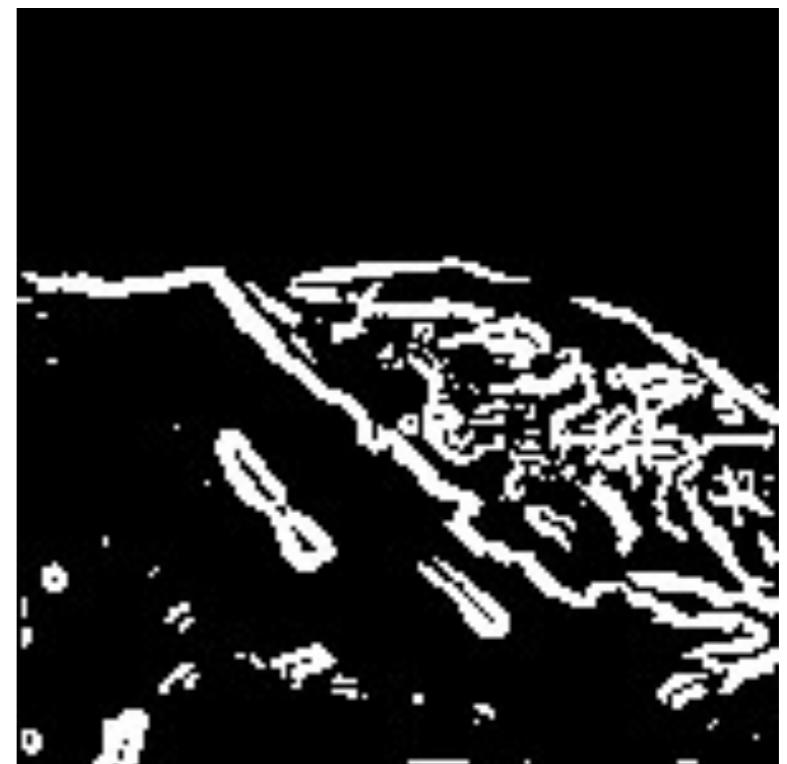
# Lines are hard to find



Original image



Edge detection



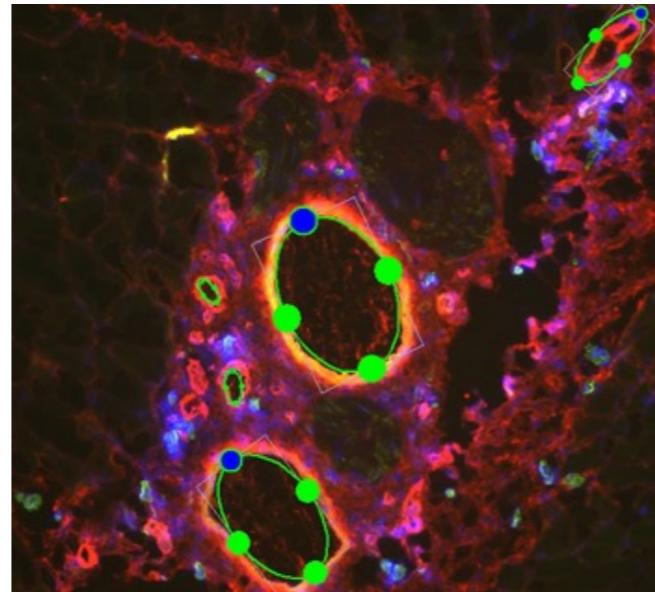
Thresholding

Noisy edge image  
Incomplete boundaries

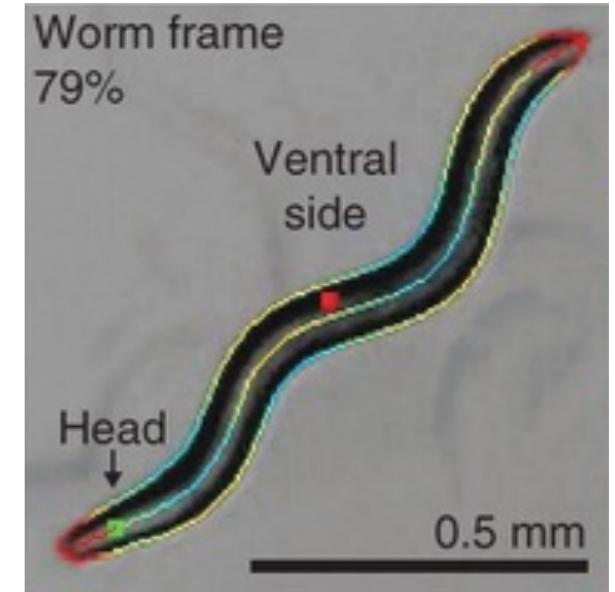
# Applications



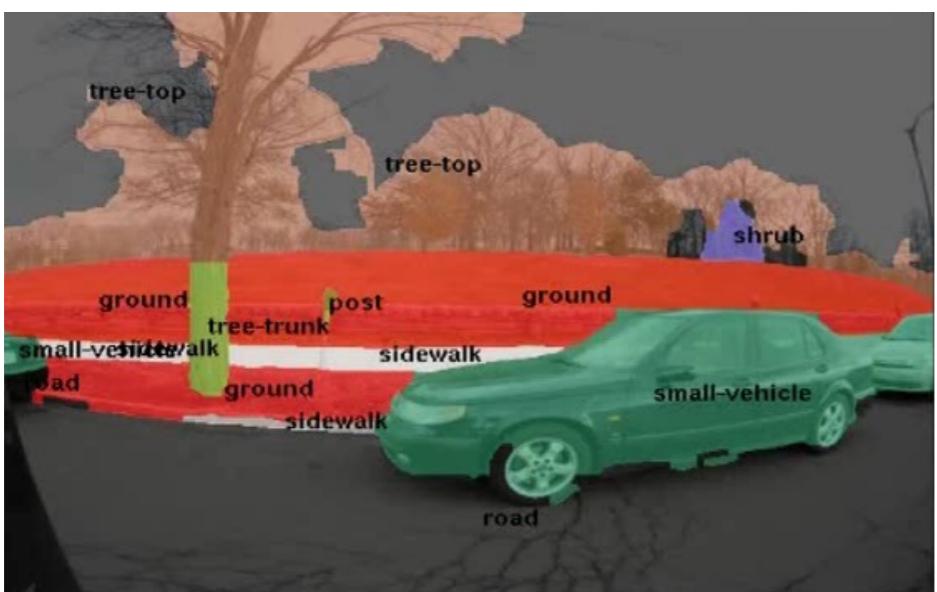
Autonomous Vehicles  
(lane line detection)



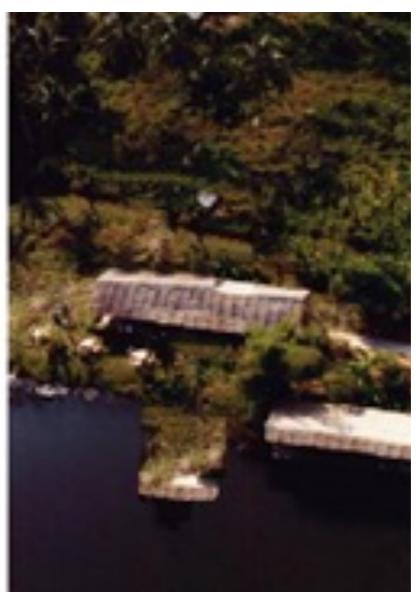
tissue engineering  
(blood vessel counting)



behavioral genetics  
(earthworm contours)



Autonomous Vehicles  
(semantic scene segmentation)



Computational Photography  
(image inpainting)

# Line fitting

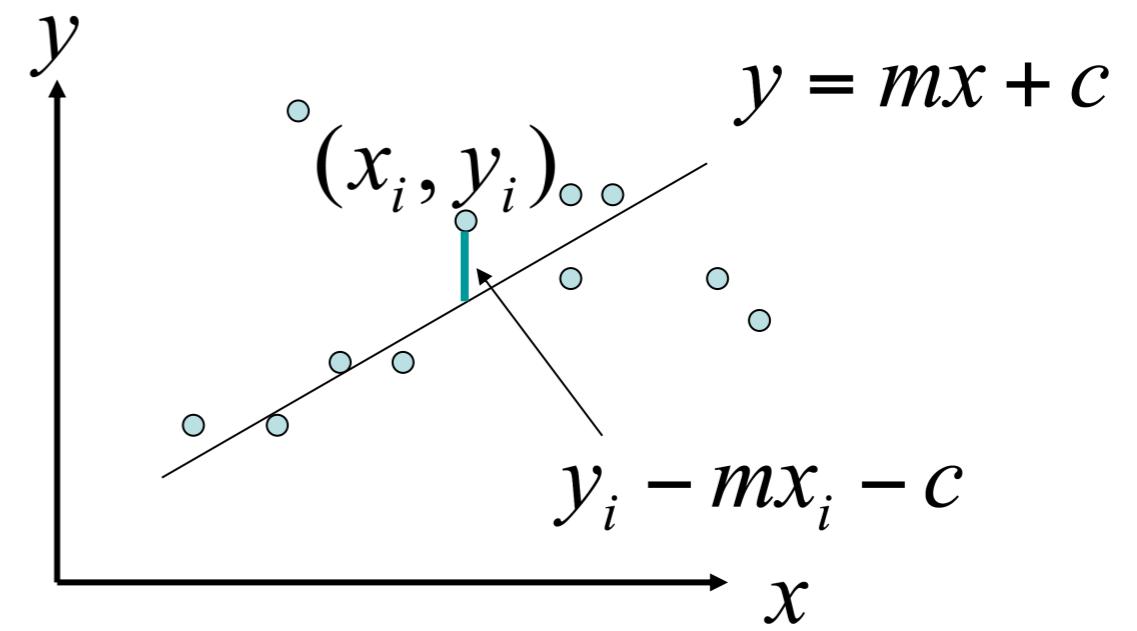
# Line fitting

Given: Many  $(x_i, y_i)$  pairs

Find: Parameters  $(m, c)$

Minimize: Average square distance:

$$E = \sum_i \frac{(y_i - mx_i - c)^2}{N}$$



*How can we solve this minimization?*

# Line fitting

Given: Many  $(x_i, y_i)$  pairs

Find: Parameters  $(m, c)$

Minimize: Average square distance:

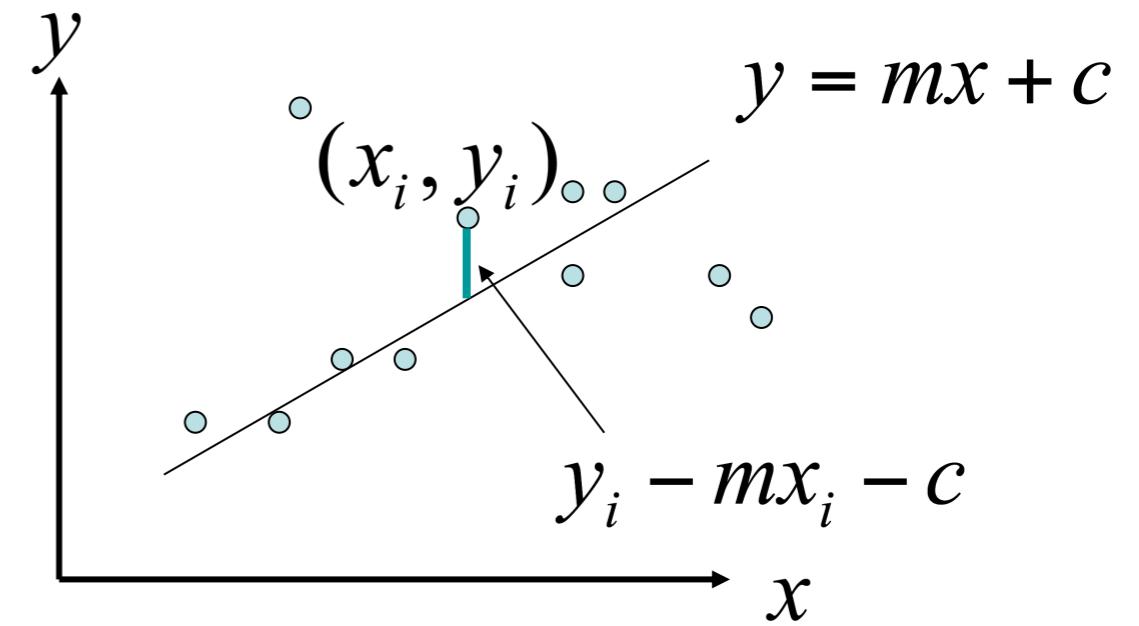
$$E = \sum_i \frac{(y_i - mx_i - c)^2}{N}$$

Using:

$$\frac{\partial E}{\partial m} = 0 \quad \& \quad \frac{\partial E}{\partial c} = 0$$

Note:

$$\bar{y} = \frac{\sum_i y_i}{N} \quad \bar{x} = \frac{\sum_i x_i}{N}$$



$$c = \bar{y} - m \bar{x}$$

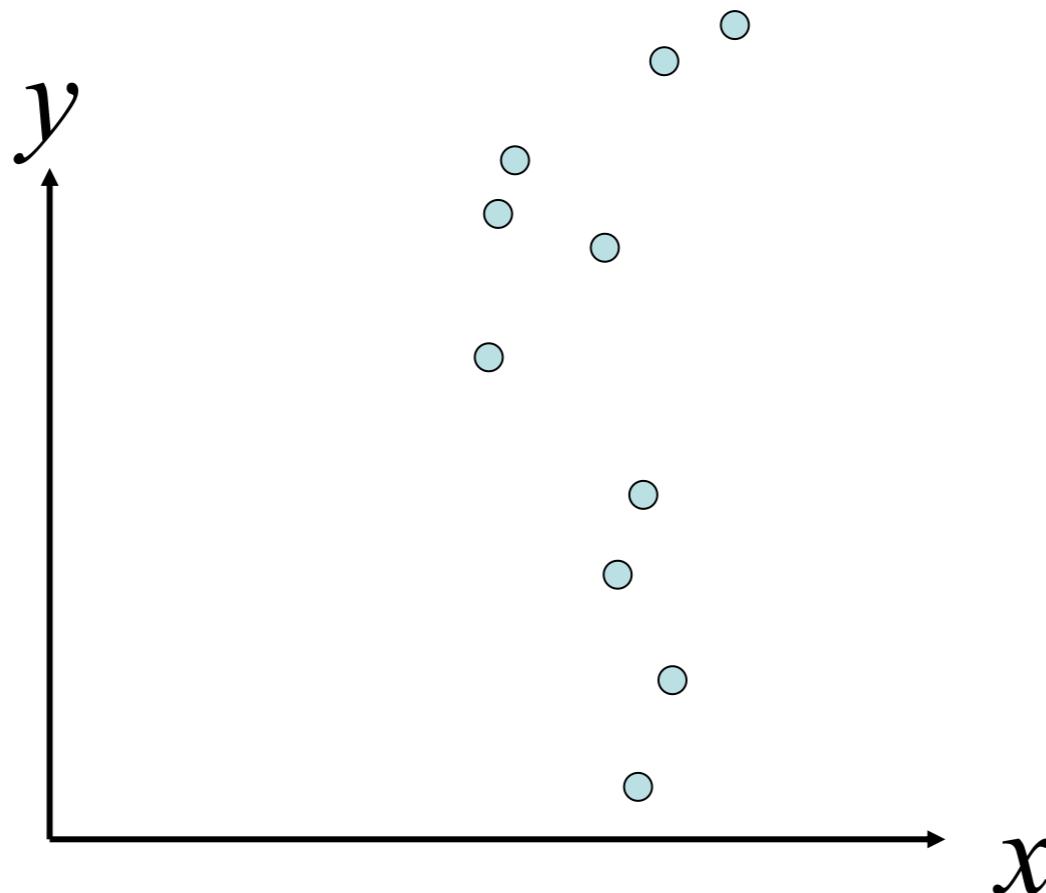
$$m = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sum_i (x_i - \bar{x})^2}$$

*What are some problems with the approach?*

# Problems with parameterizations

Where is the line that minimizes E?

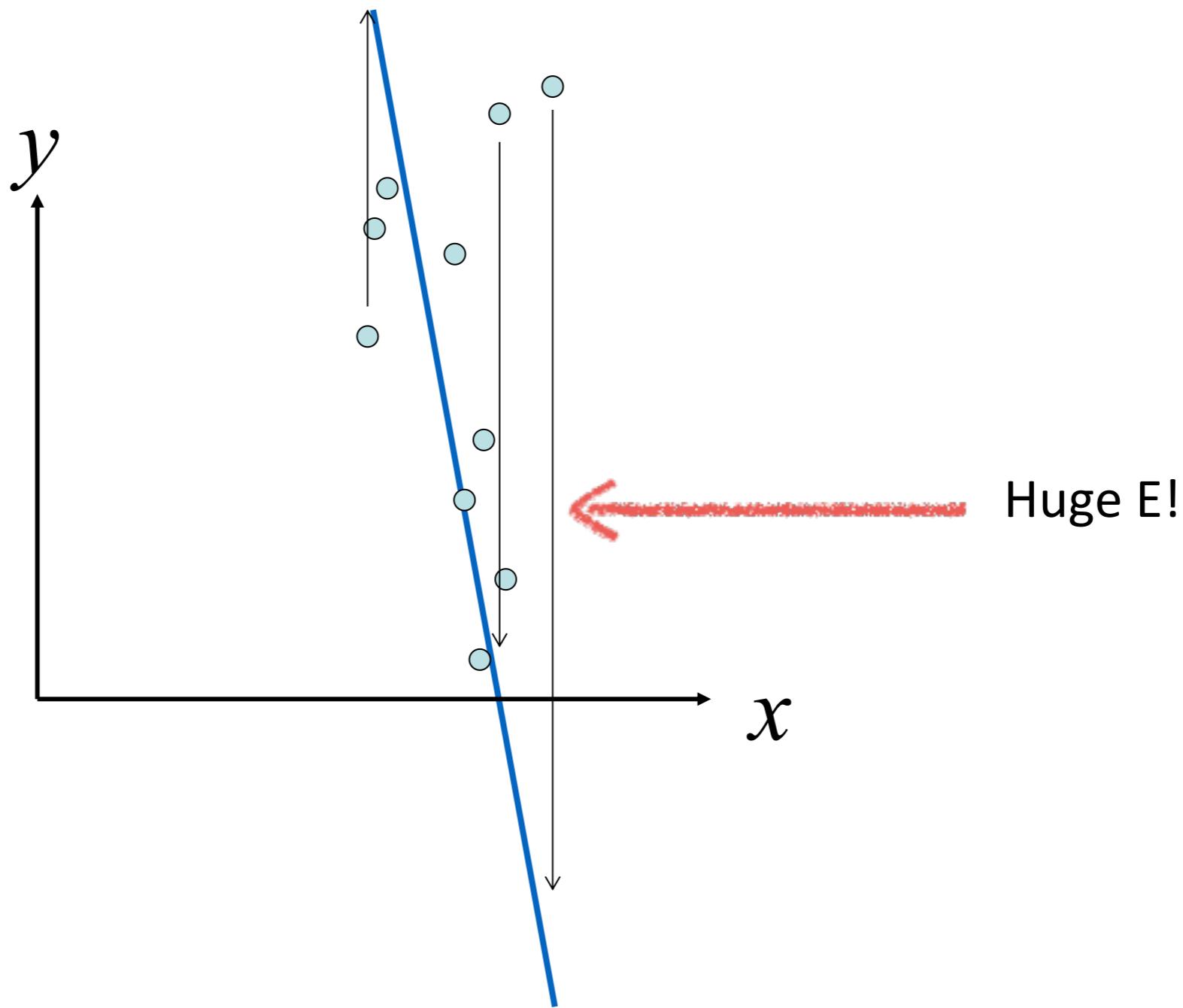
$$E = \sum_{i=1}^n (y_i - mx_i - b)^2$$



# Problems with parameterizations

Where is the line that minimizes E?

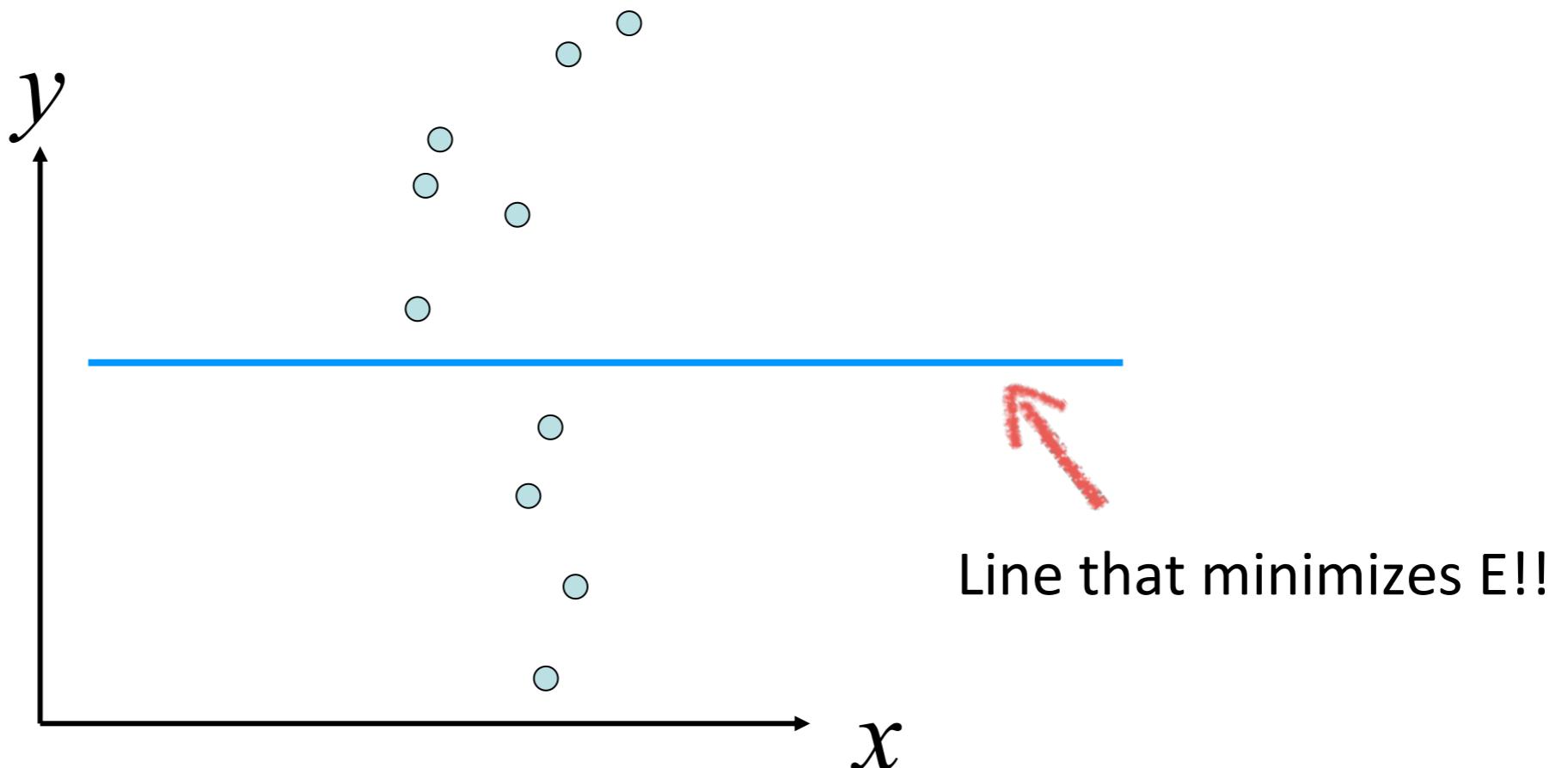
$$E = \sum_{i=1}^n (y_i - mx_i - b)^2$$



# Problems with parameterizations

Where is the line that minimizes E?

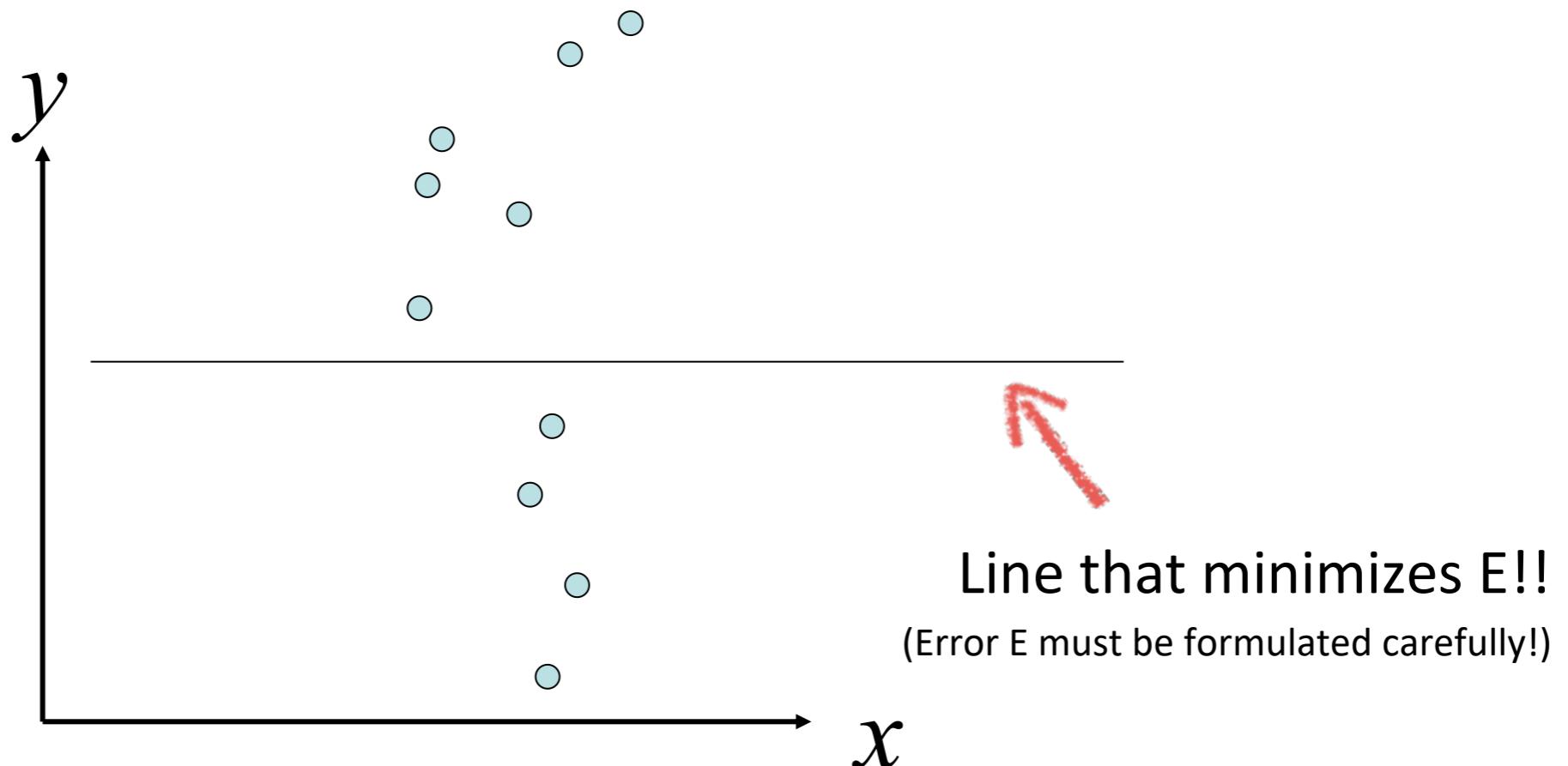
$$E = \sum_{i=1}^n (y_i - mx_i - b)^2$$



# Problems with parameterizations

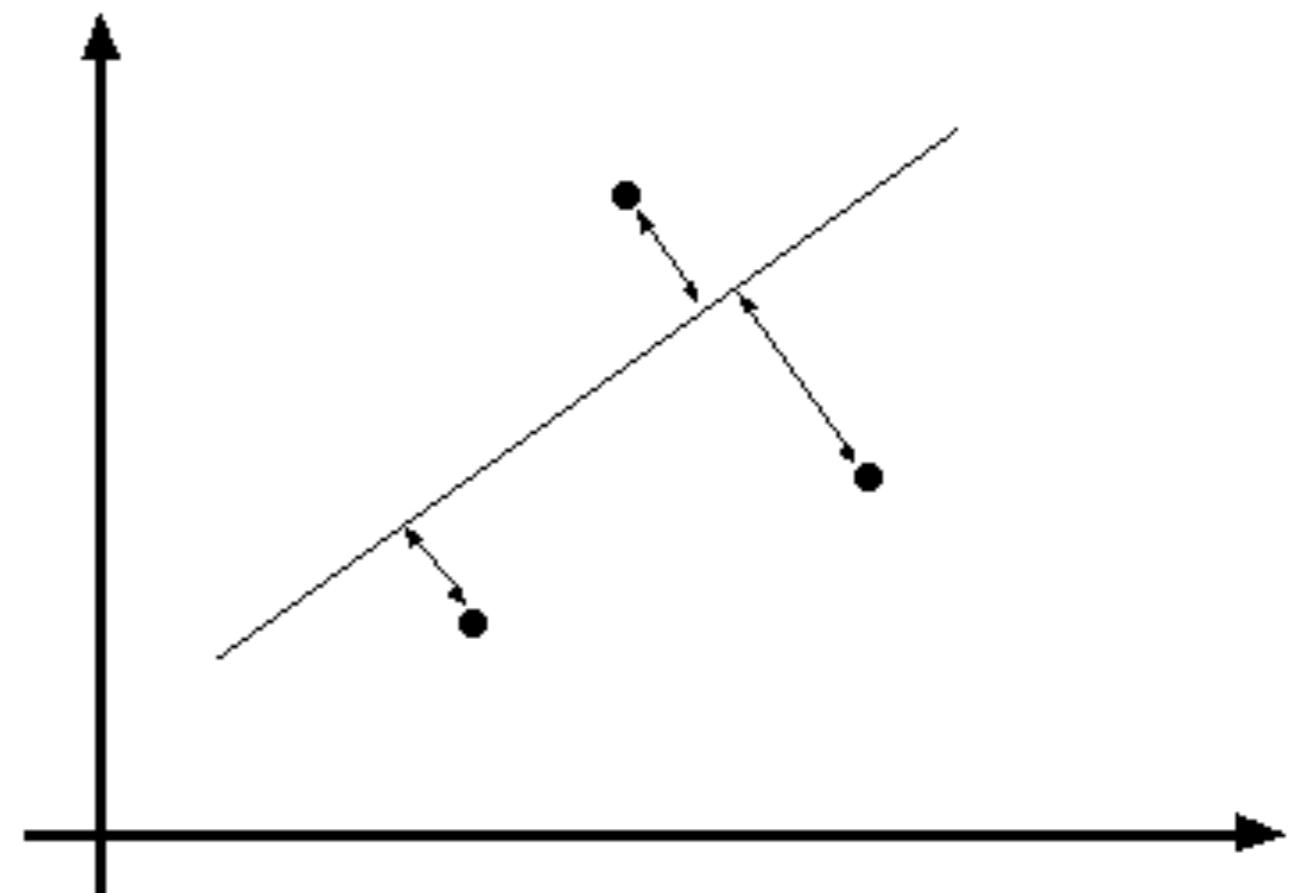
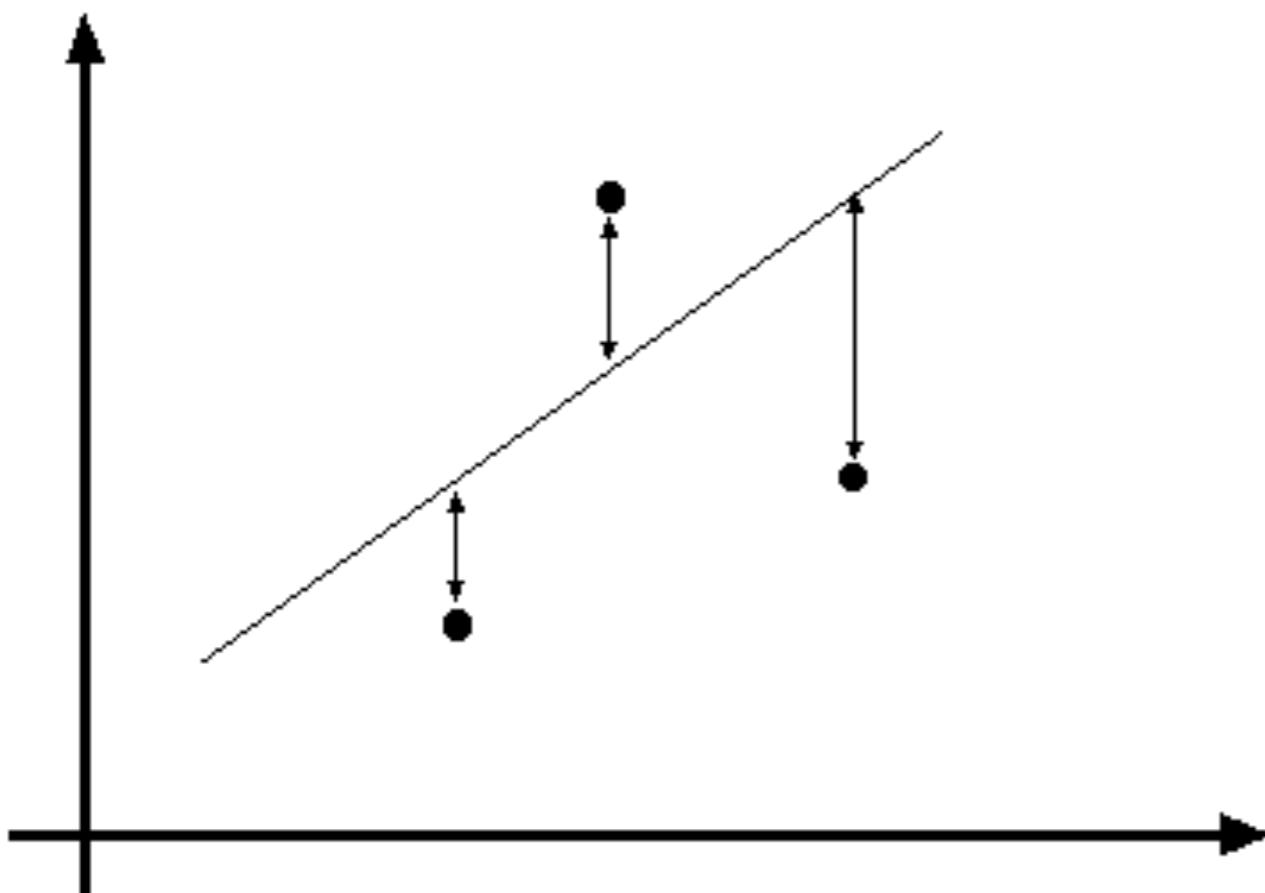
Where is the line that minimizes E?

$$E = \sum_{i=1}^n (y_i - mx_i - b)^2$$



*How can we deal with this?*

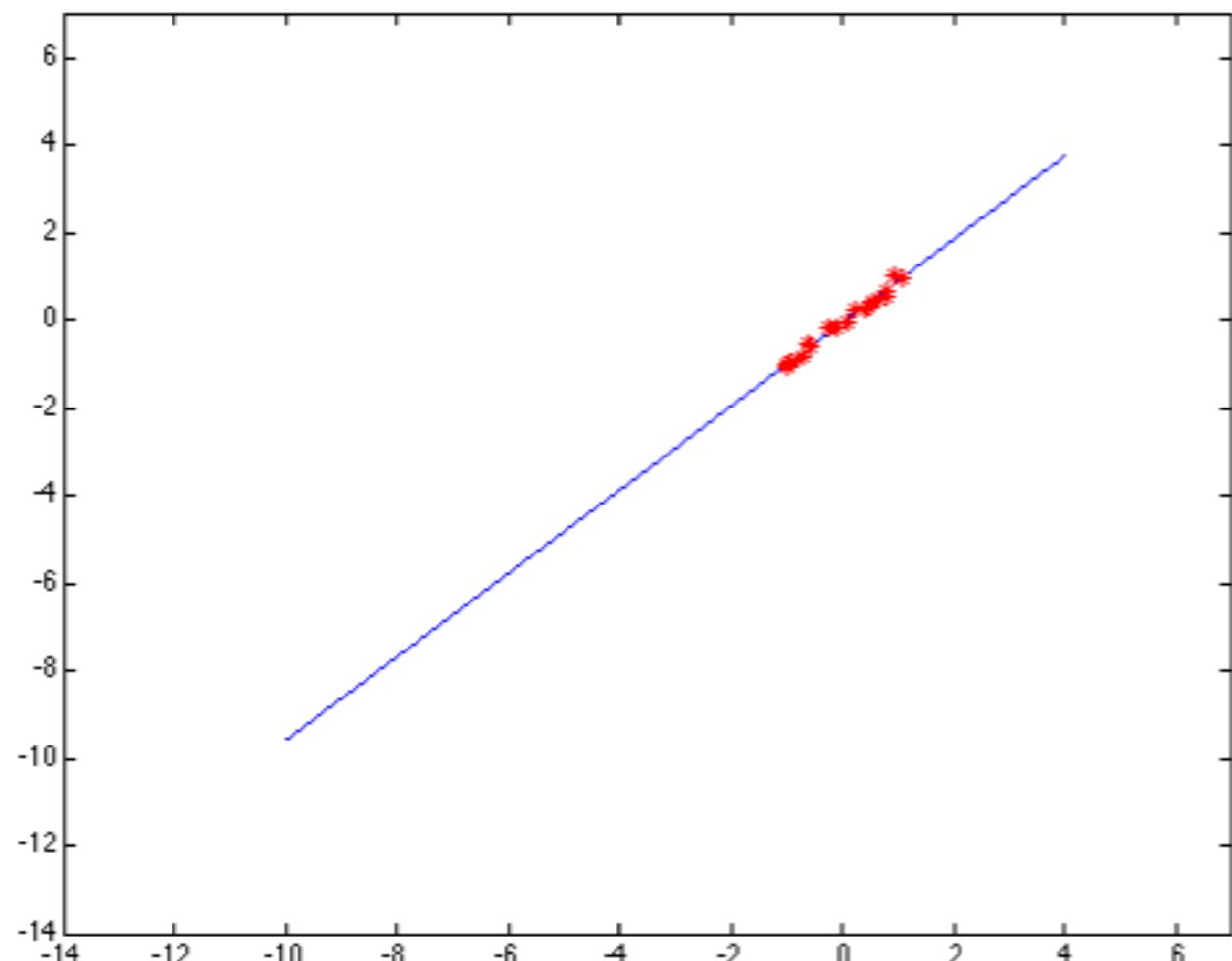
Line fitting is easily setup as a maximum likelihood problem  
... but choice of model is important



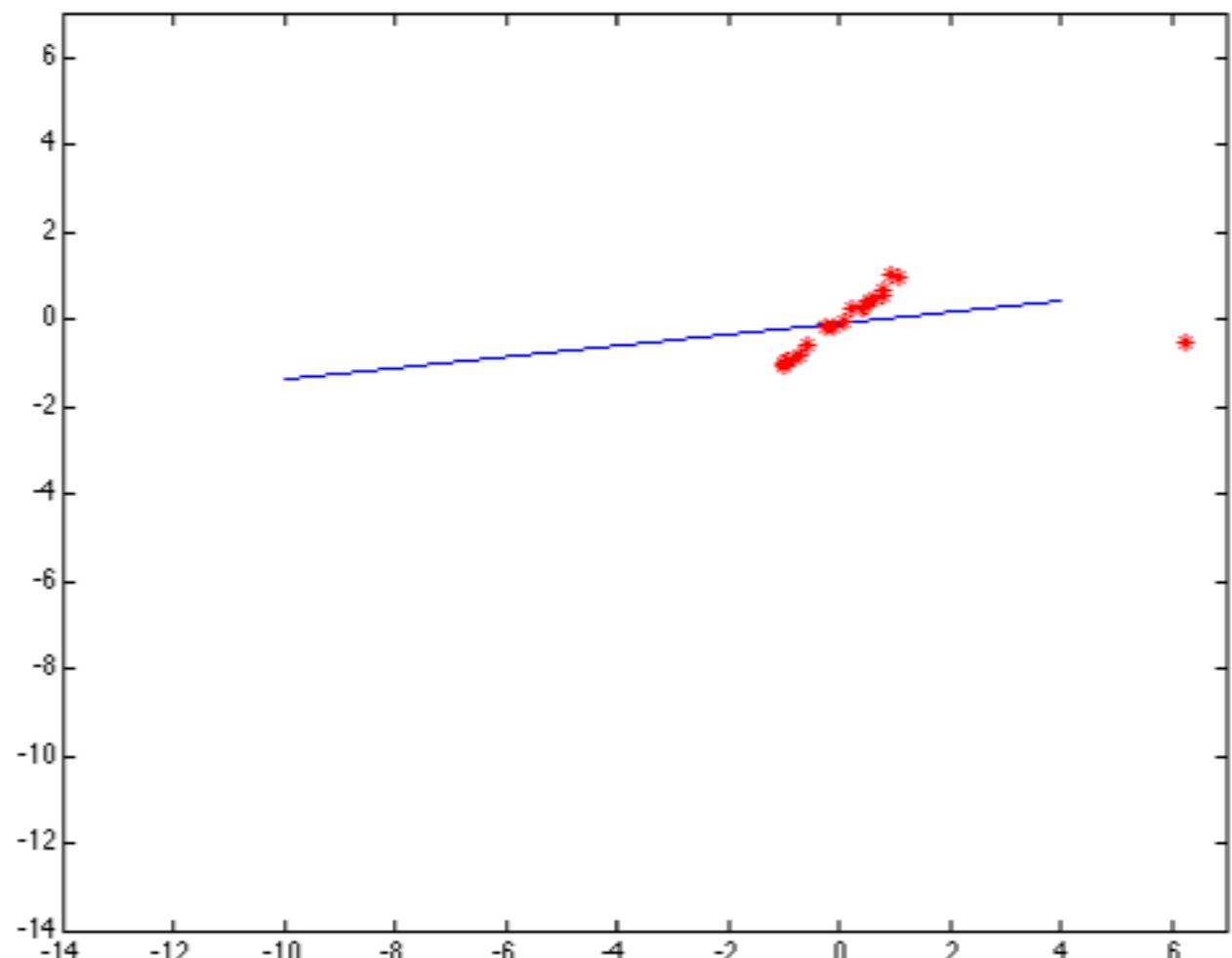
$$E = \sum_{i=1}^n (y_i - mx_i - b)^2$$

*What optimization are we solving here?*

# Problems with noise



Least-squares error fit



Squared error heavily penalizes outliers

# Model fitting is difficult because...

- **Extraneous data:** clutter or multiple models
  - We do not know what is part of the model?
  - Can we pull out models with a few parts from much larger amounts of background clutter?
- **Missing data:** only some parts of model are present
- **Noise**
- Cost:
  - It is not feasible to check all combinations of features by fitting a model to each possible subset

*So what can we do?*

# Line parameterizations

# Slope intercept form

$$y = mx + b$$

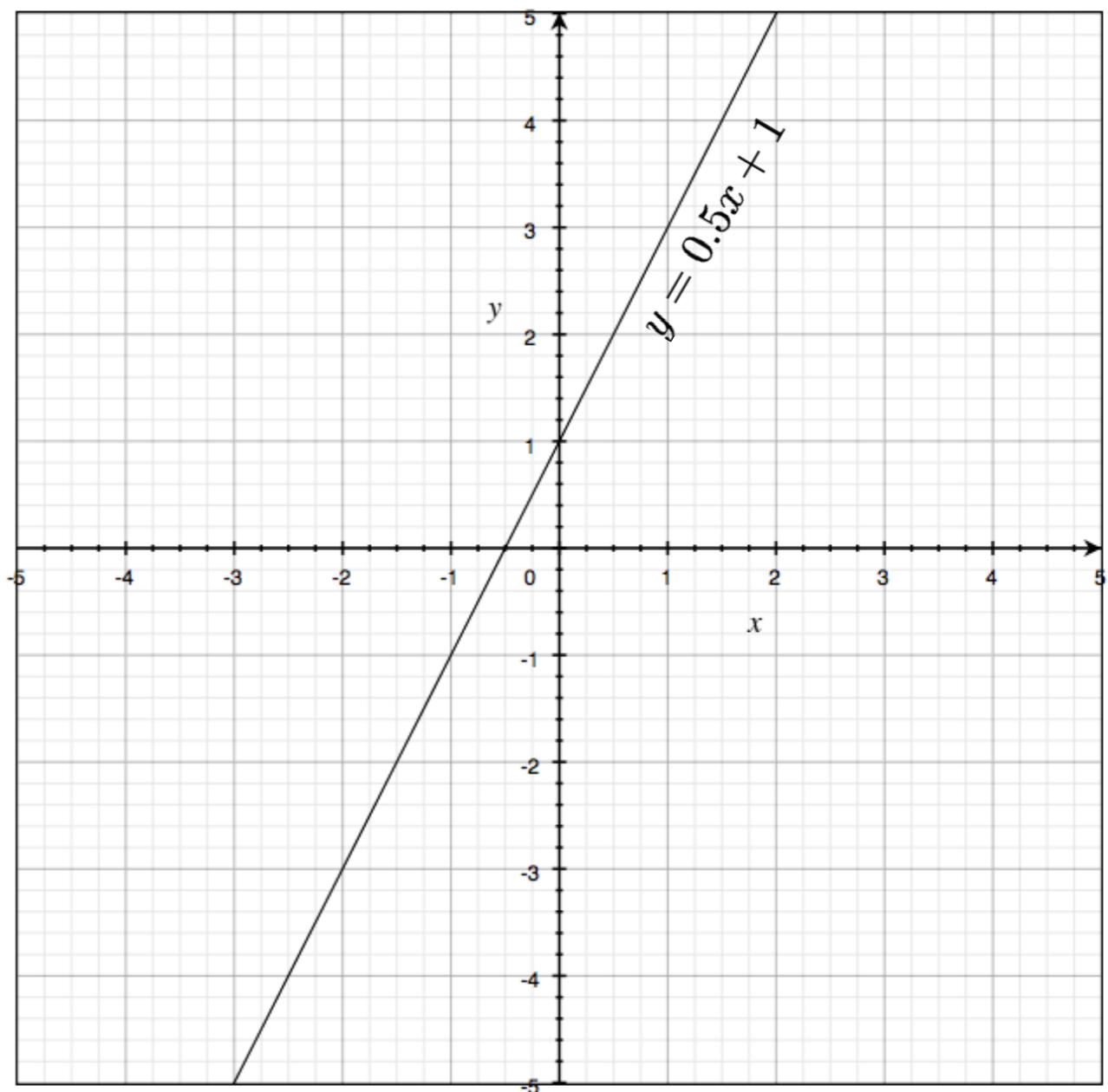

slope      y-intercept

*What are  $m$  and  $b$ ?*

# Slope intercept form

$$y = mx + b$$

slope      y-intercept



# Double intercept form

$$\frac{x}{a} + \frac{y}{b} = 1$$

x-intercept      y-intercept

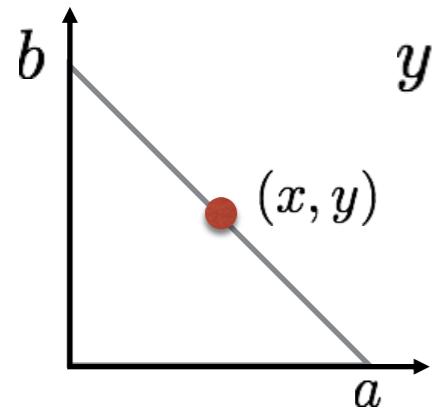
*What are a and b?*

# Double intercept form

$$\frac{x}{a} + \frac{y}{b} = 1$$

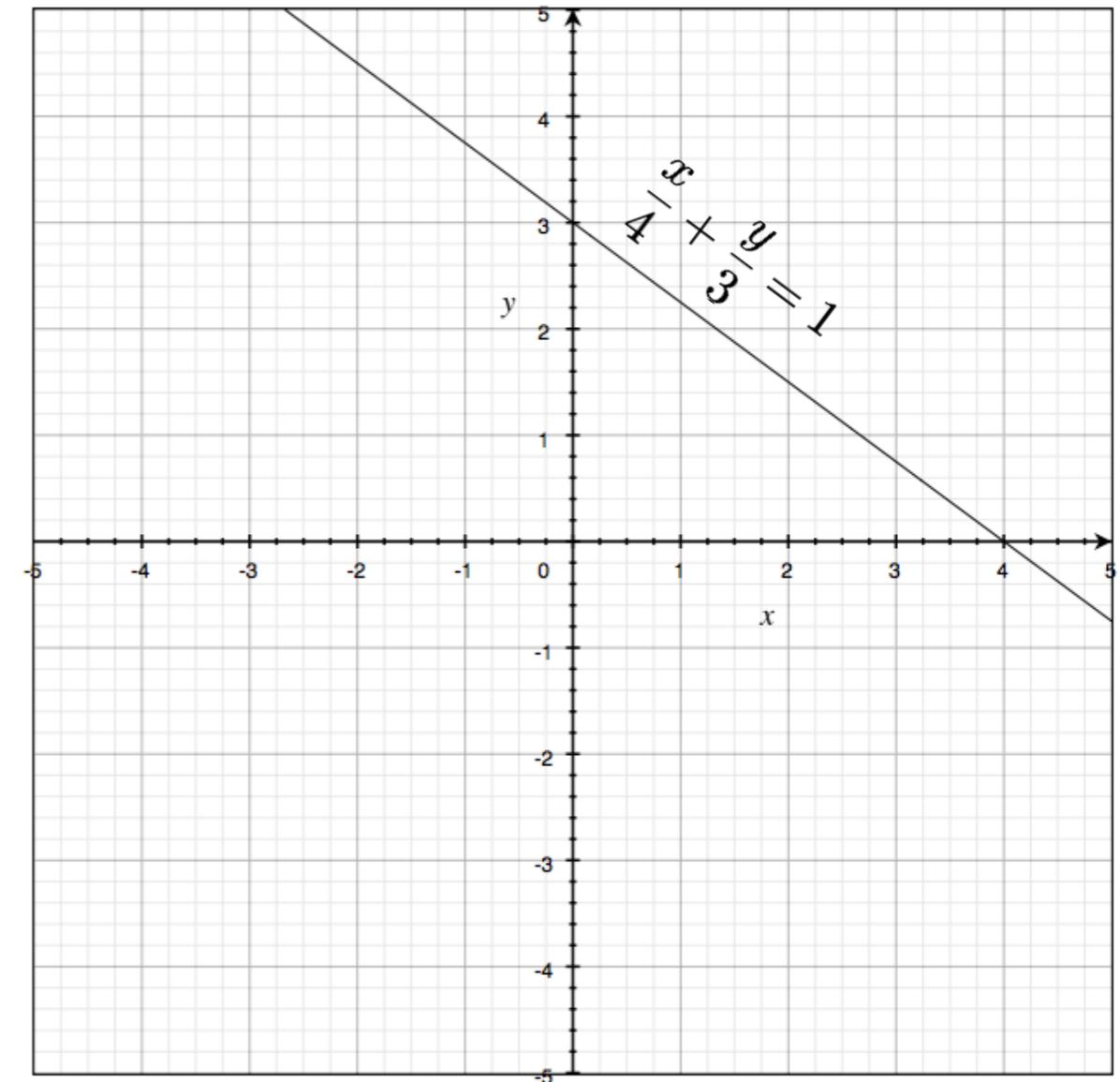
x-intercept      y-intercept

Derivation:



(Similar slope)  $\frac{y - b}{x - 0} = \frac{0 - y}{a - x}$

$$ya + yx - ba + bx = -yx$$
$$ya + bx = ba$$
$$\frac{y}{b} + \frac{x}{a} = 1$$



# Normal Form

$$x \cos \theta + y \sin \theta = \rho$$

*What are rho and theta?*

# Normal Form

$$x \cos \theta + y \sin \theta = \rho$$

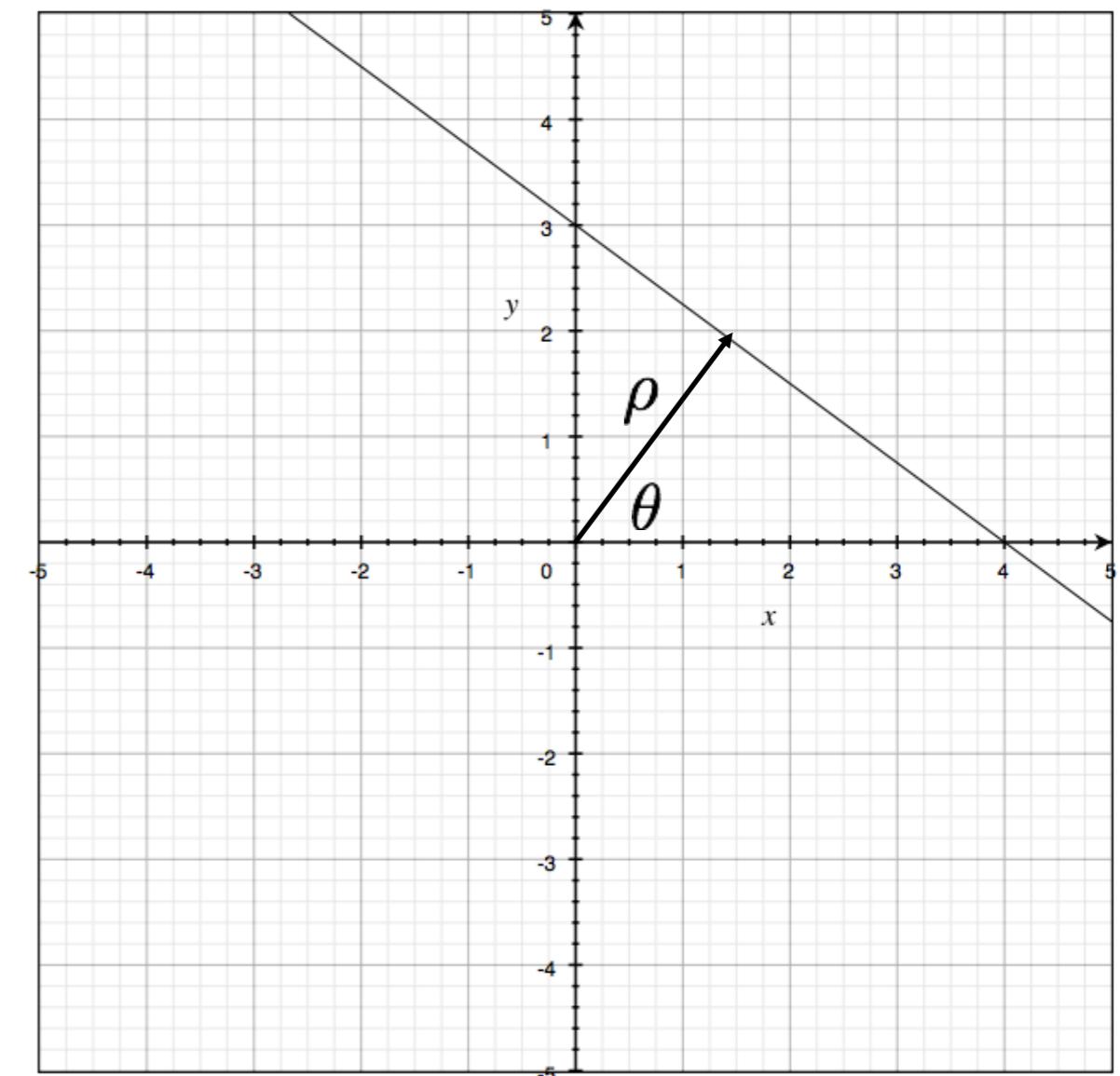
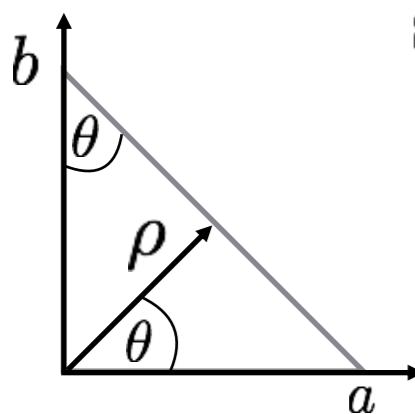
Derivation:

$$\cos \theta = \frac{\rho}{a} \rightarrow a = \frac{\rho}{\cos \theta}$$

$$\sin \theta = \frac{\rho}{b} \rightarrow b = \frac{\rho}{\sin \theta}$$

plug into:  $\frac{x}{a} + \frac{y}{b} = 1$

$$x \cos \theta + y \sin \theta = \rho$$



# Hough transform

# Hough transform

- Generic framework for detecting a parametric model
- Edges don't have to be connected
- Lines can be occluded
- Key idea: edges **vote** for the possible models

# Image and parameter space

variables  
 $y = mx + b$   
parameters

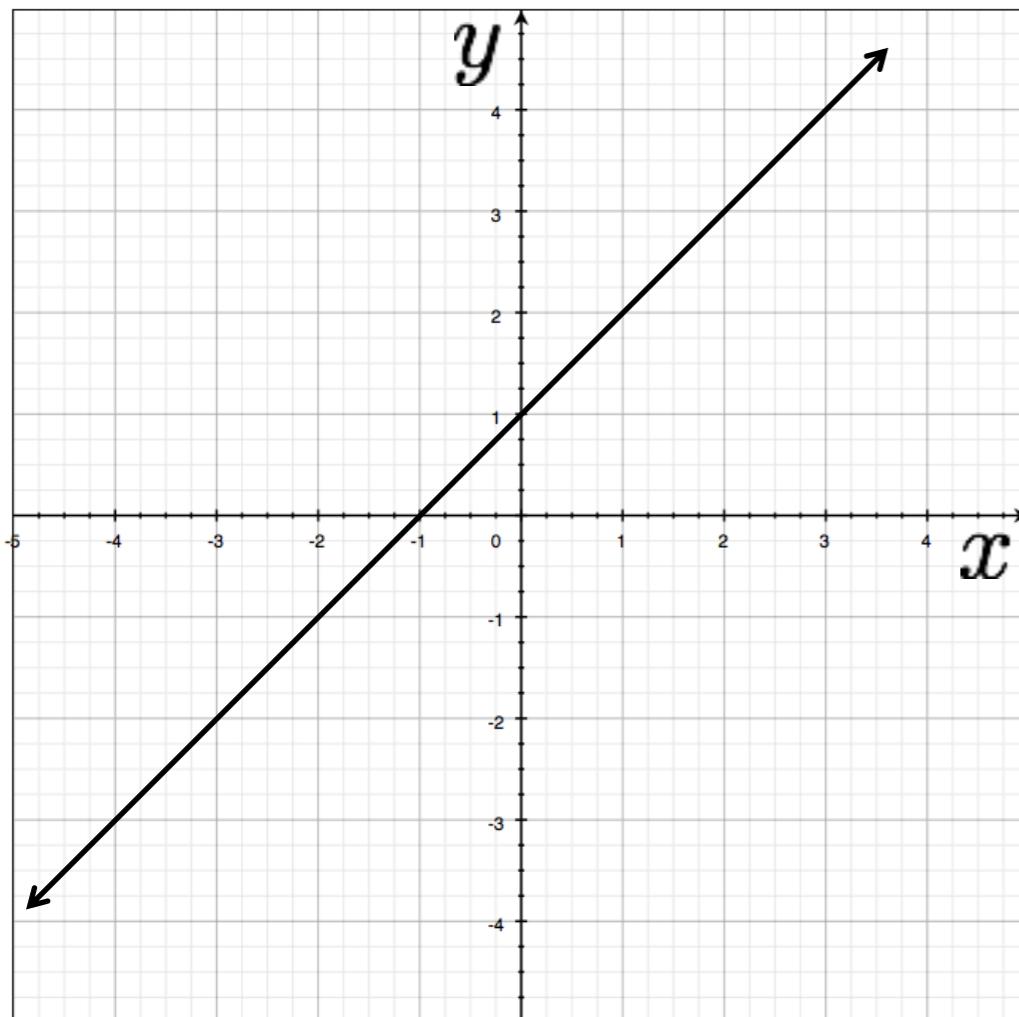
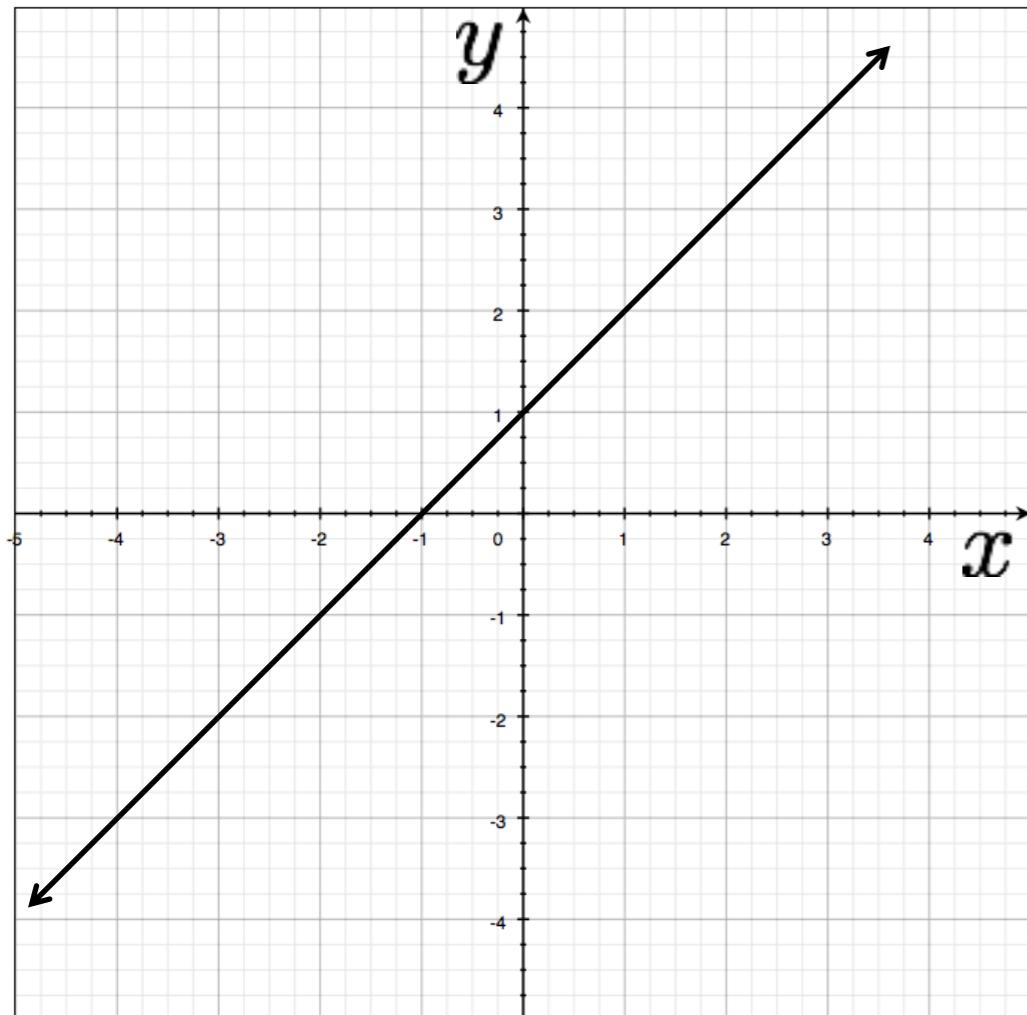


Image space

# Image and parameter space

variables  
 $y = mx + b$   
parameters



a line  
becomes a  
point

variables  
 $y - mx = b$   
parameters

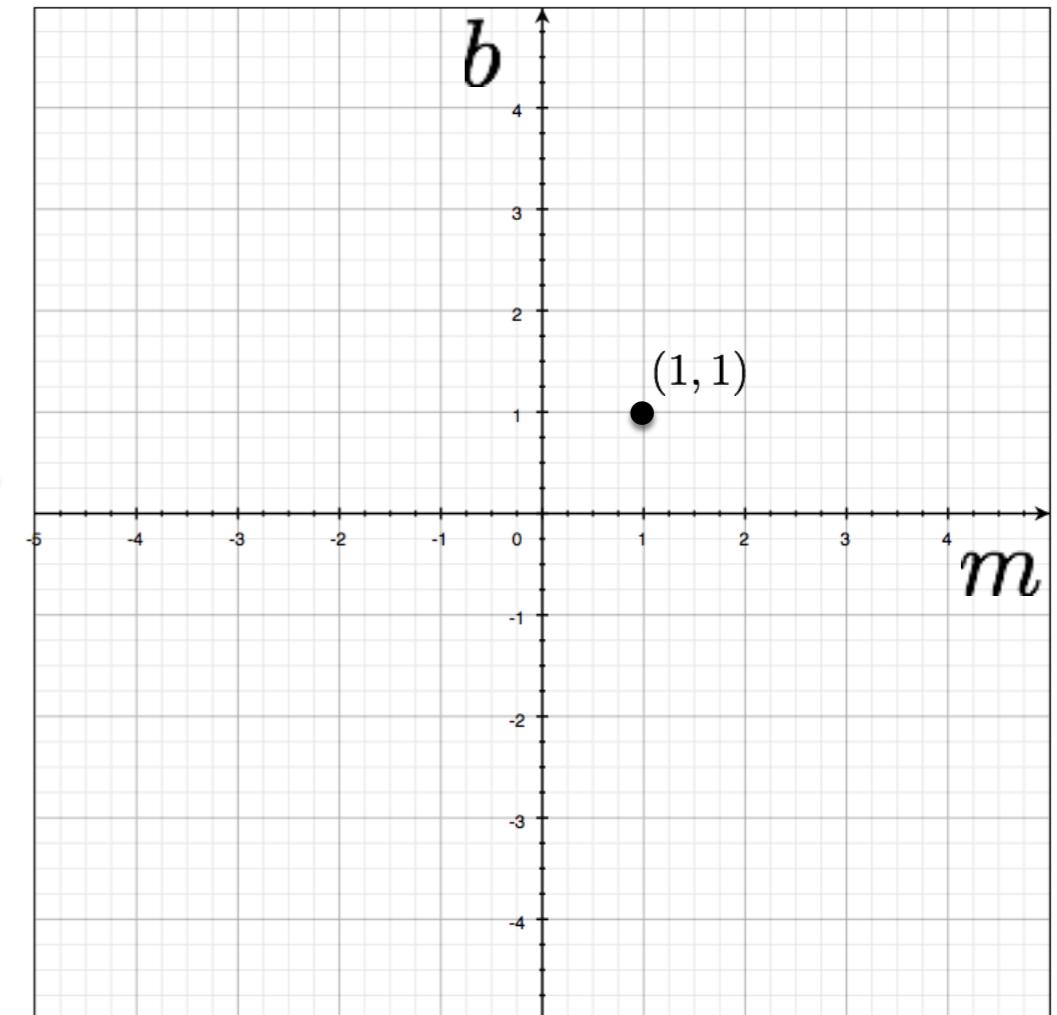


Image space

Parameter space

# Image and parameter space

variables  
 $y = mx + b$   
parameters

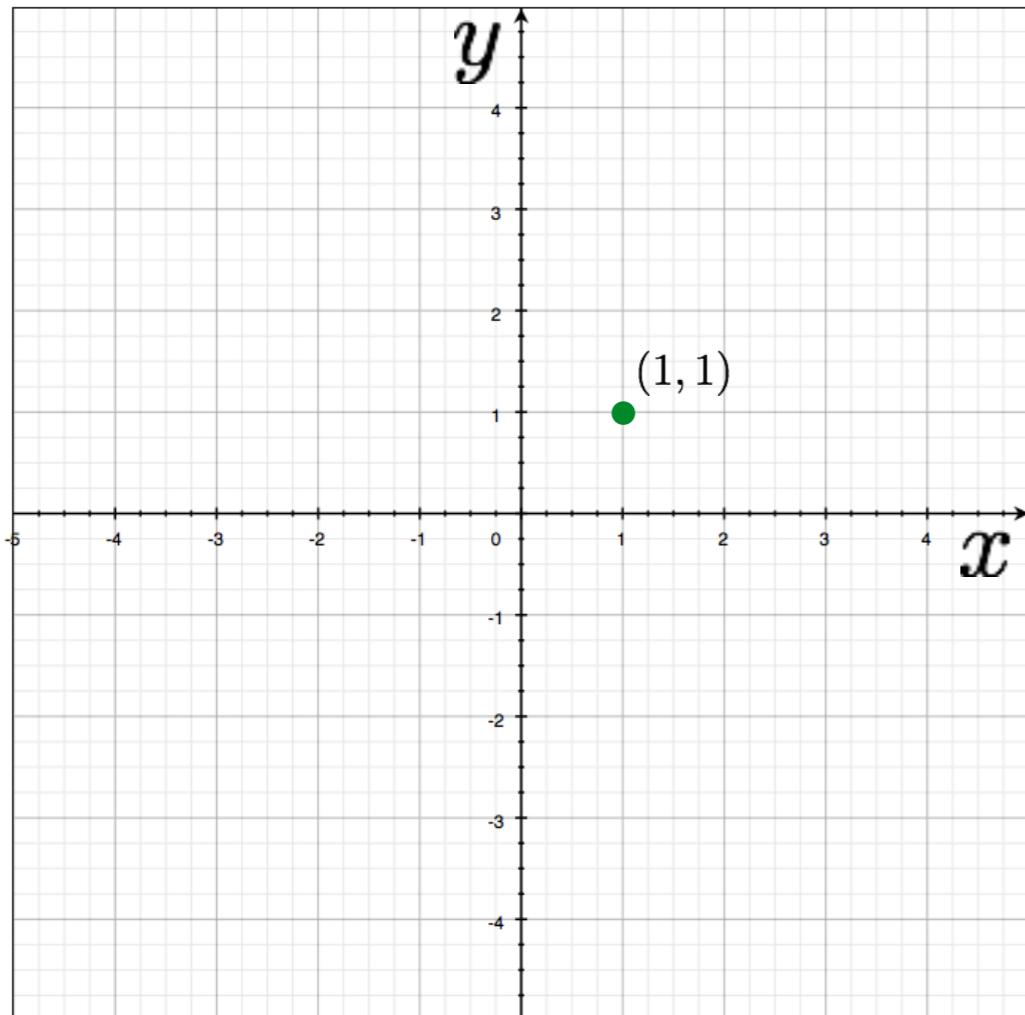
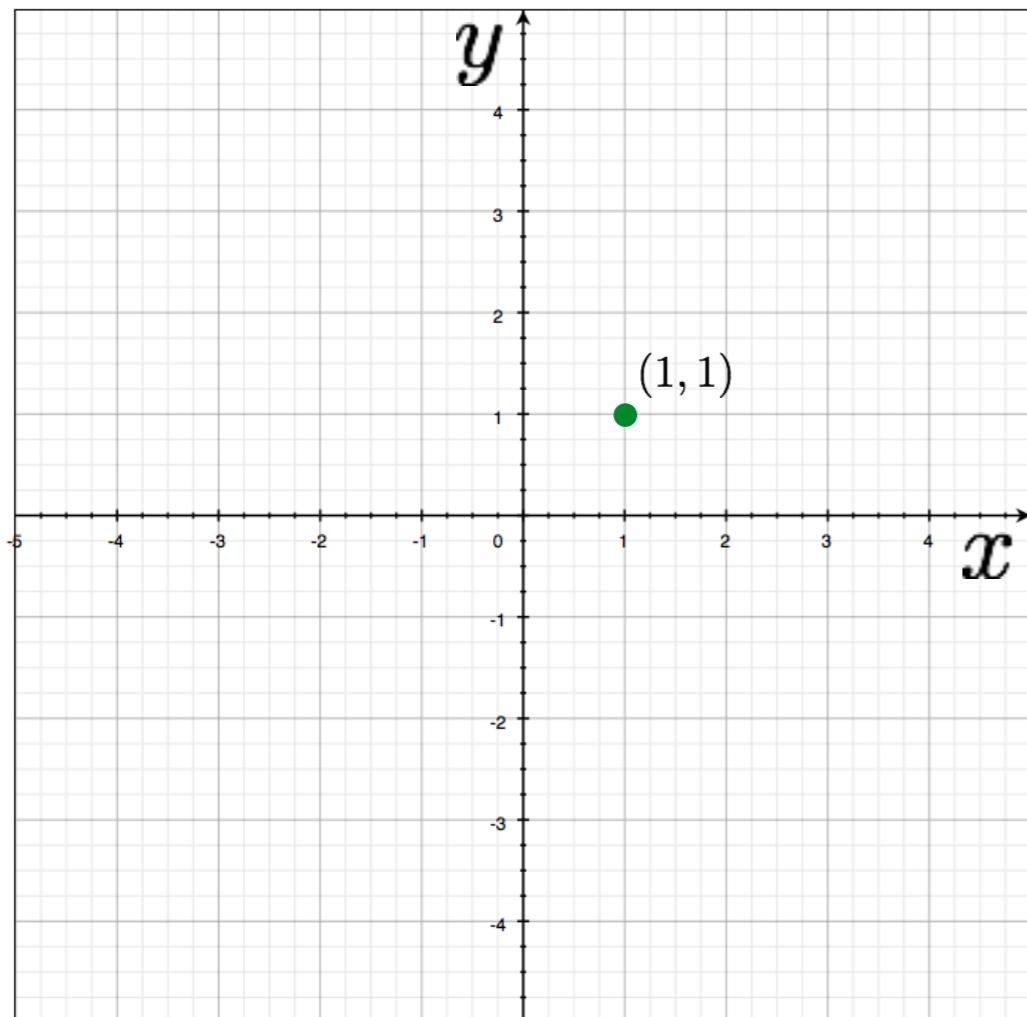


Image space

*What would a point in image space become in parameter space?*

# Image and parameter space

variables  
 $y = mx + b$   
parameters



a point becomes a line

variables  
 $y - mx = b$   
parameters

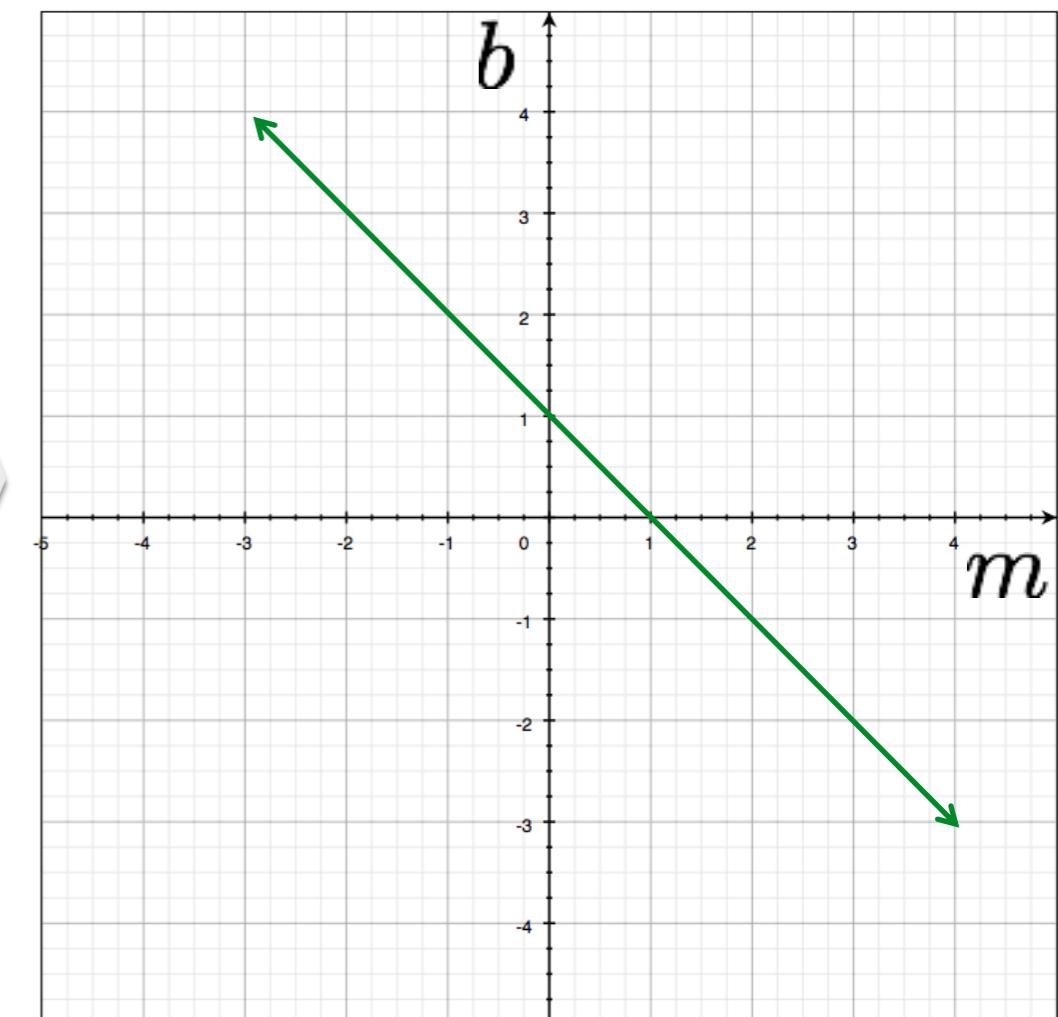
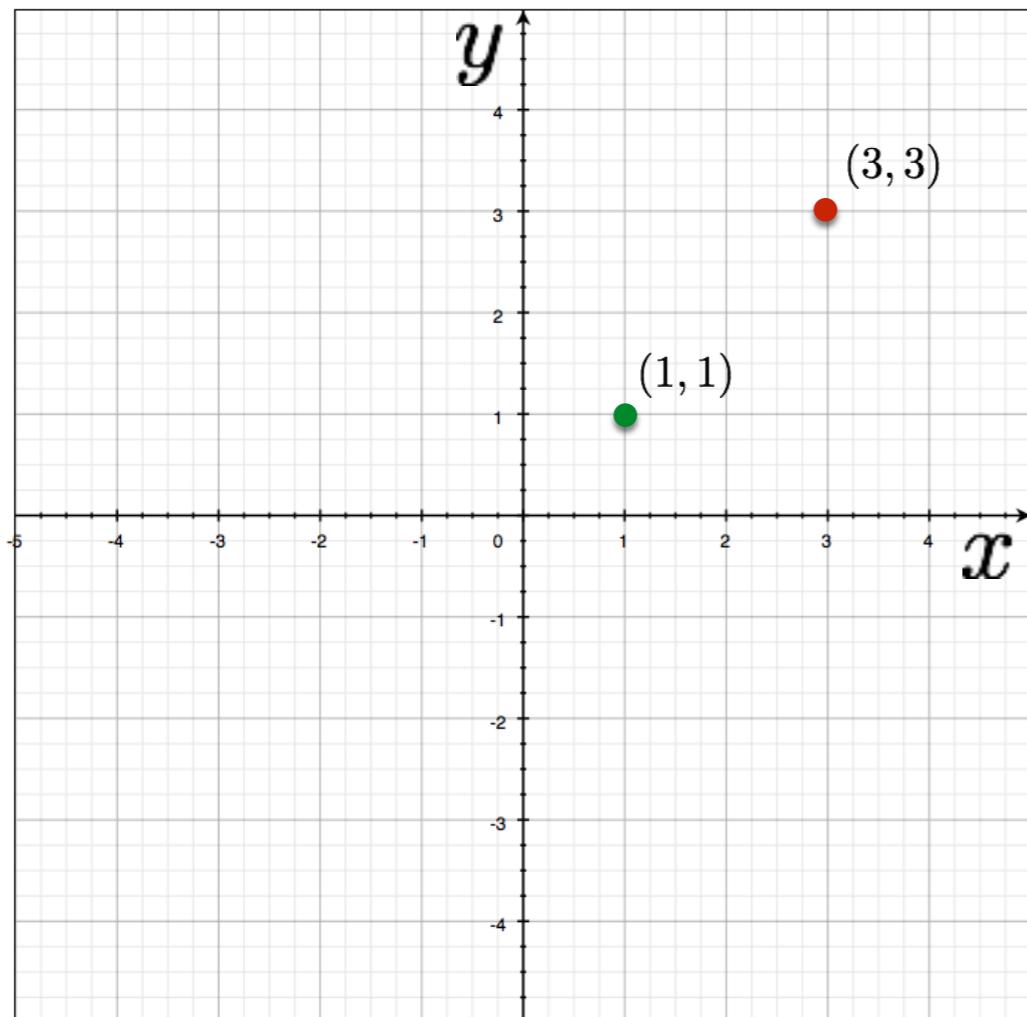


Image space

Parameter space

# Image and parameter space

variables  
 $y = mx + b$   
parameters



two points  
become  
?

variables  
 $y - mx = b$   
parameters

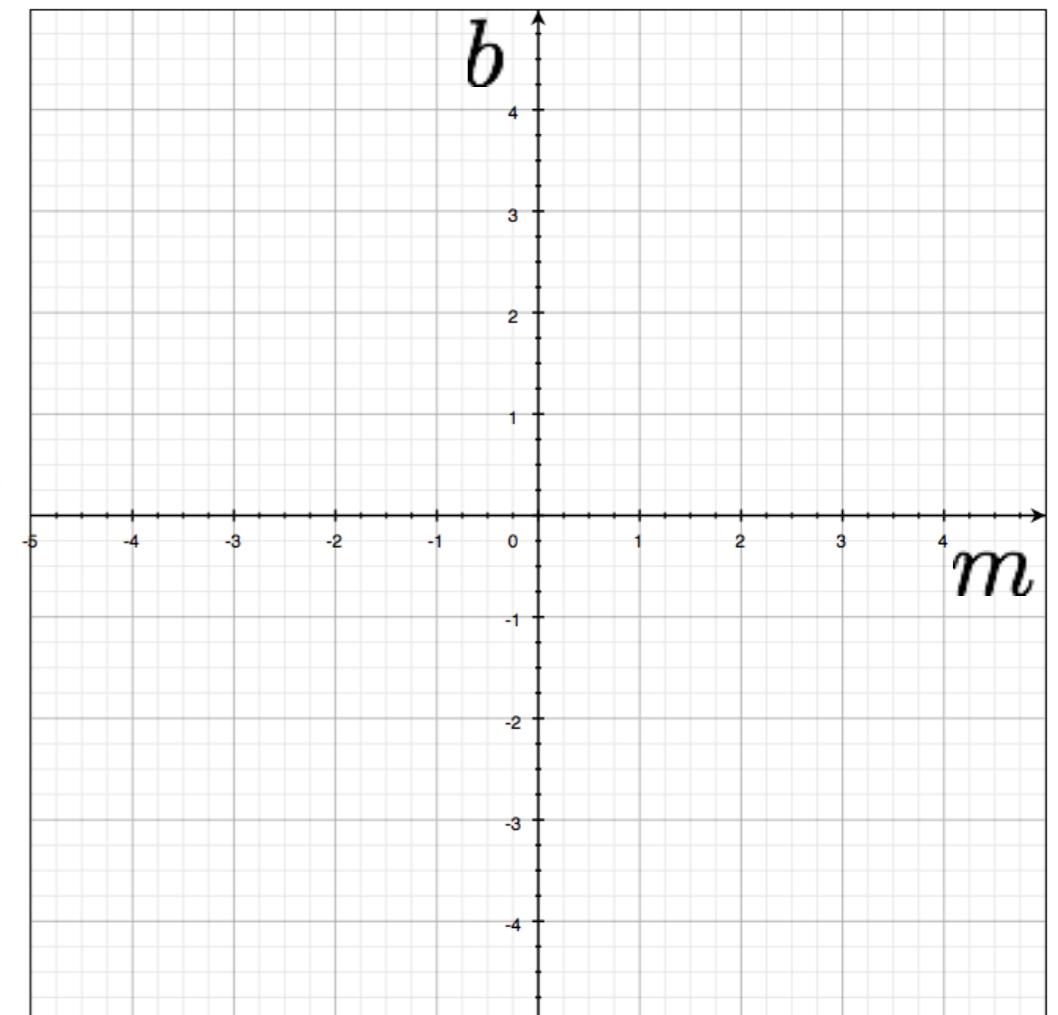
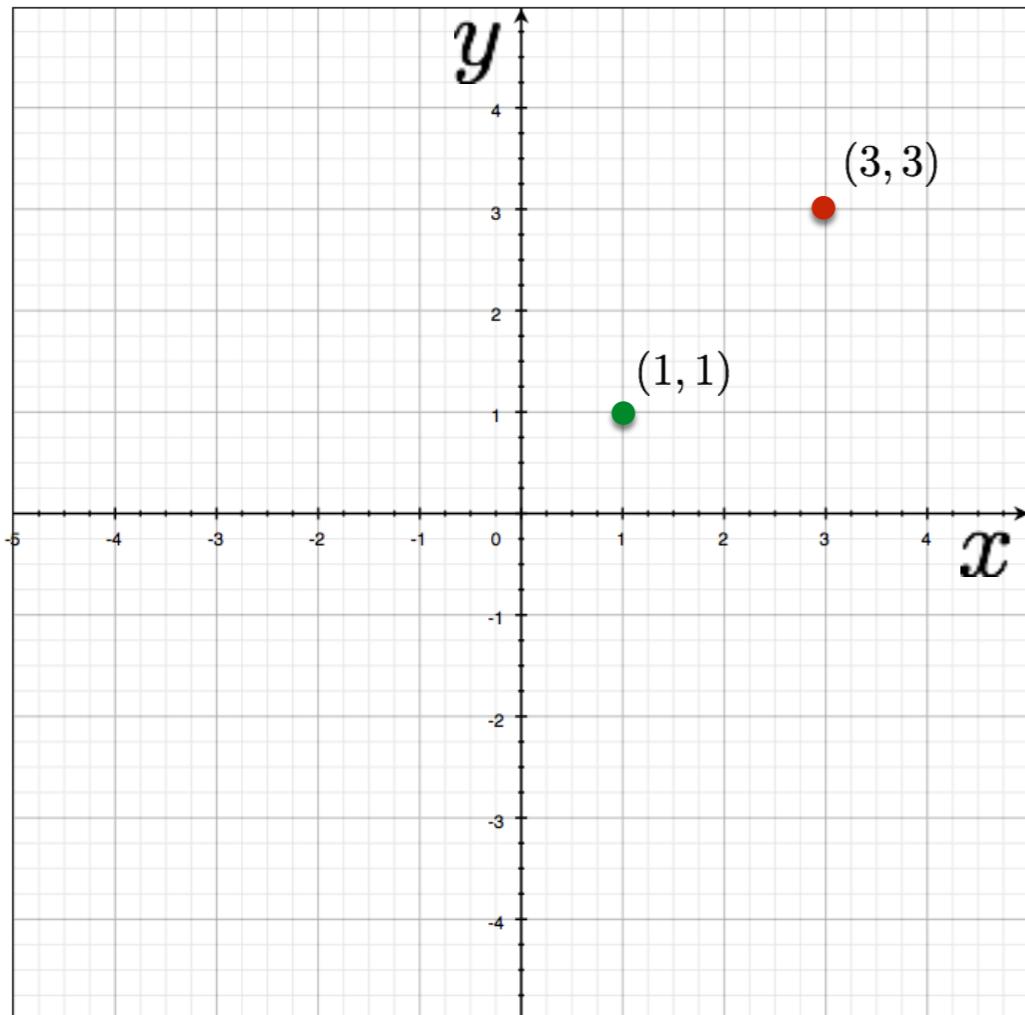


Image space

Parameter space

# Image and parameter space

variables  
 $y = mx + b$   
parameters



two points  
become  
?

variables  
 $y - mx = b$   
parameters

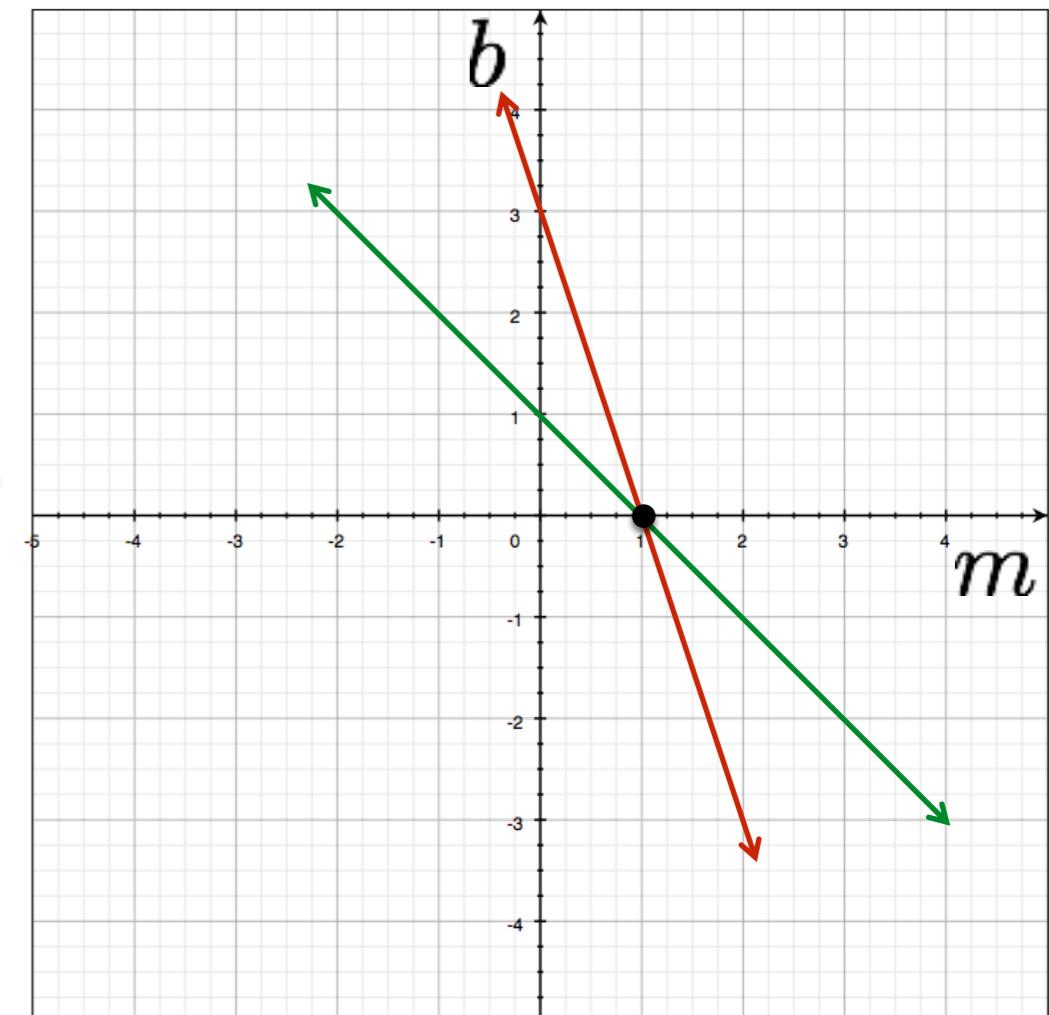
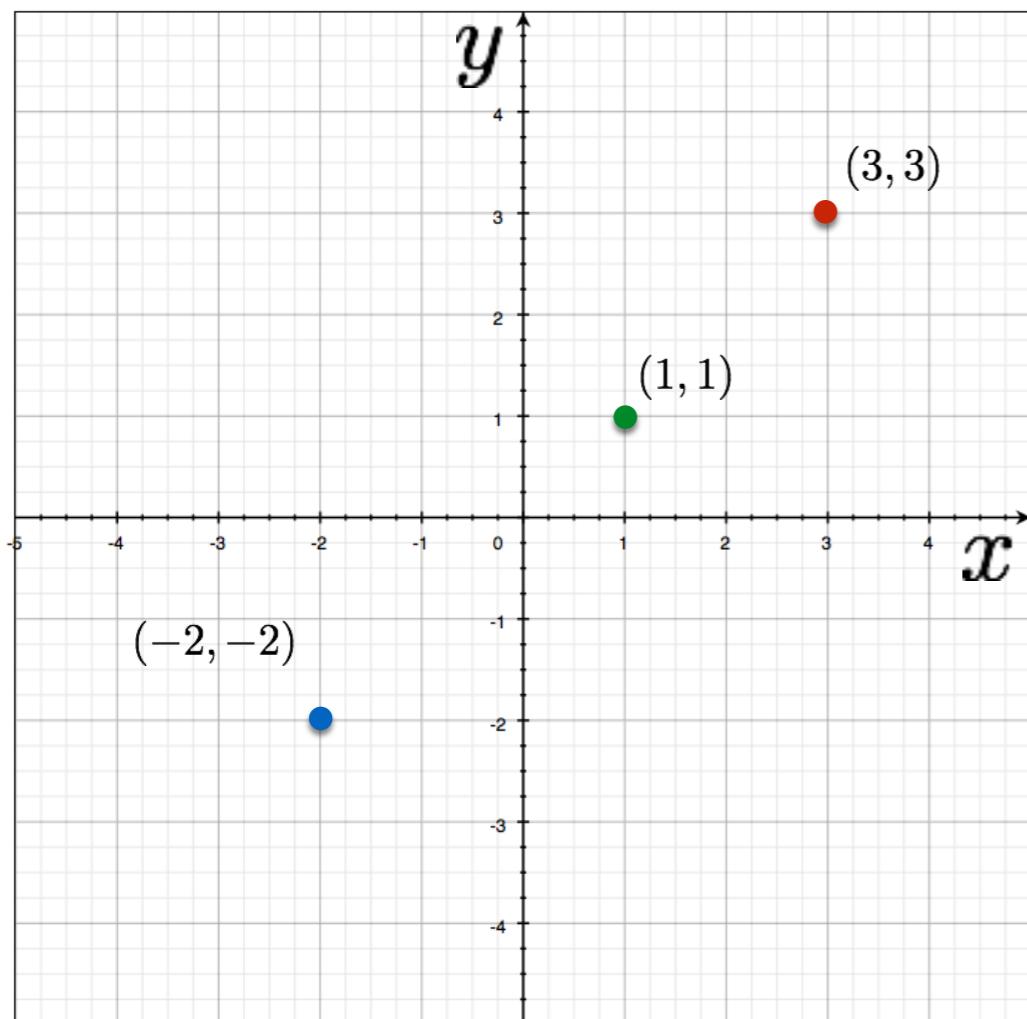


Image space

Parameter space

# Image and parameter space

variables  
 $y = mx + b$   
parameters



three points  
become  
?

variables  
 $y - mx = b$   
parameters

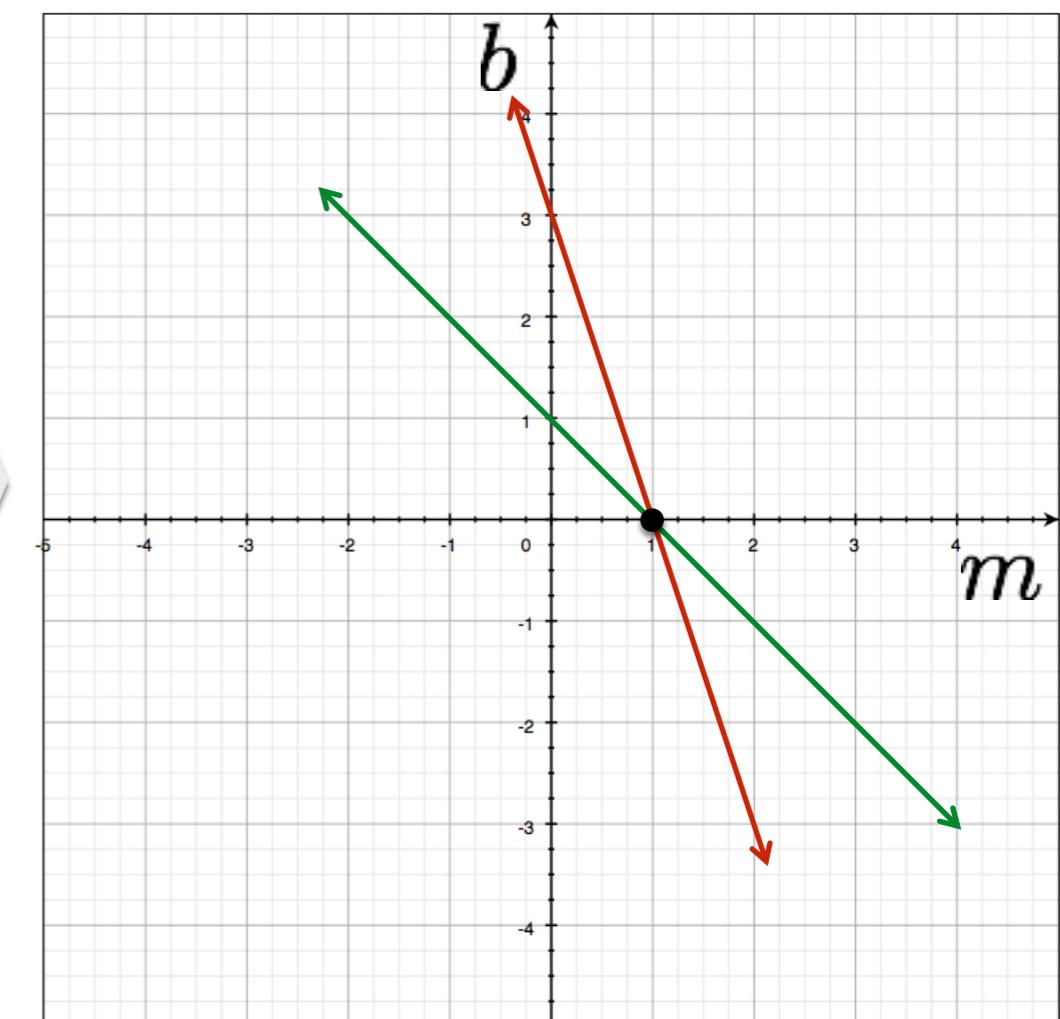
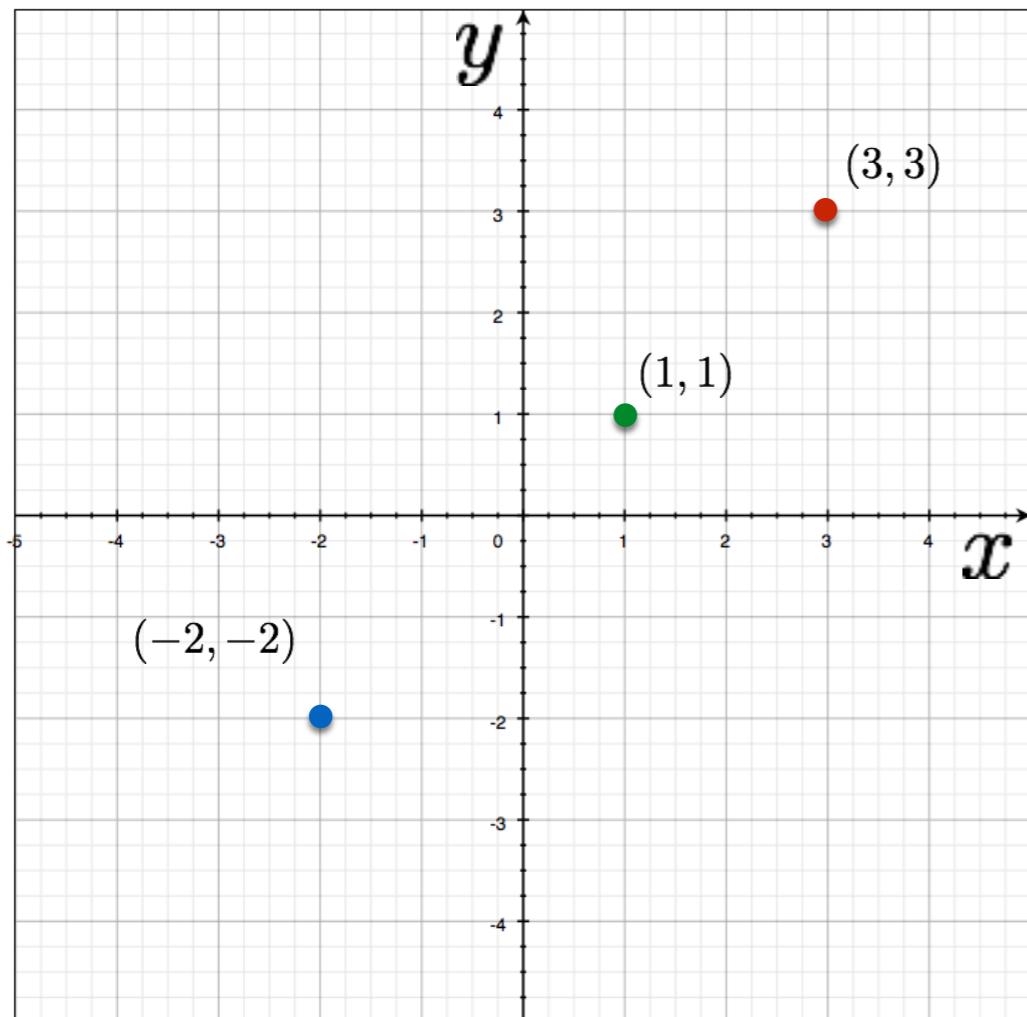


Image space

Parameter space

# Image and parameter space

variables  
 $y = mx + b$   
parameters



three points  
become  
?

variables  
 $y - mx = b$   
parameters

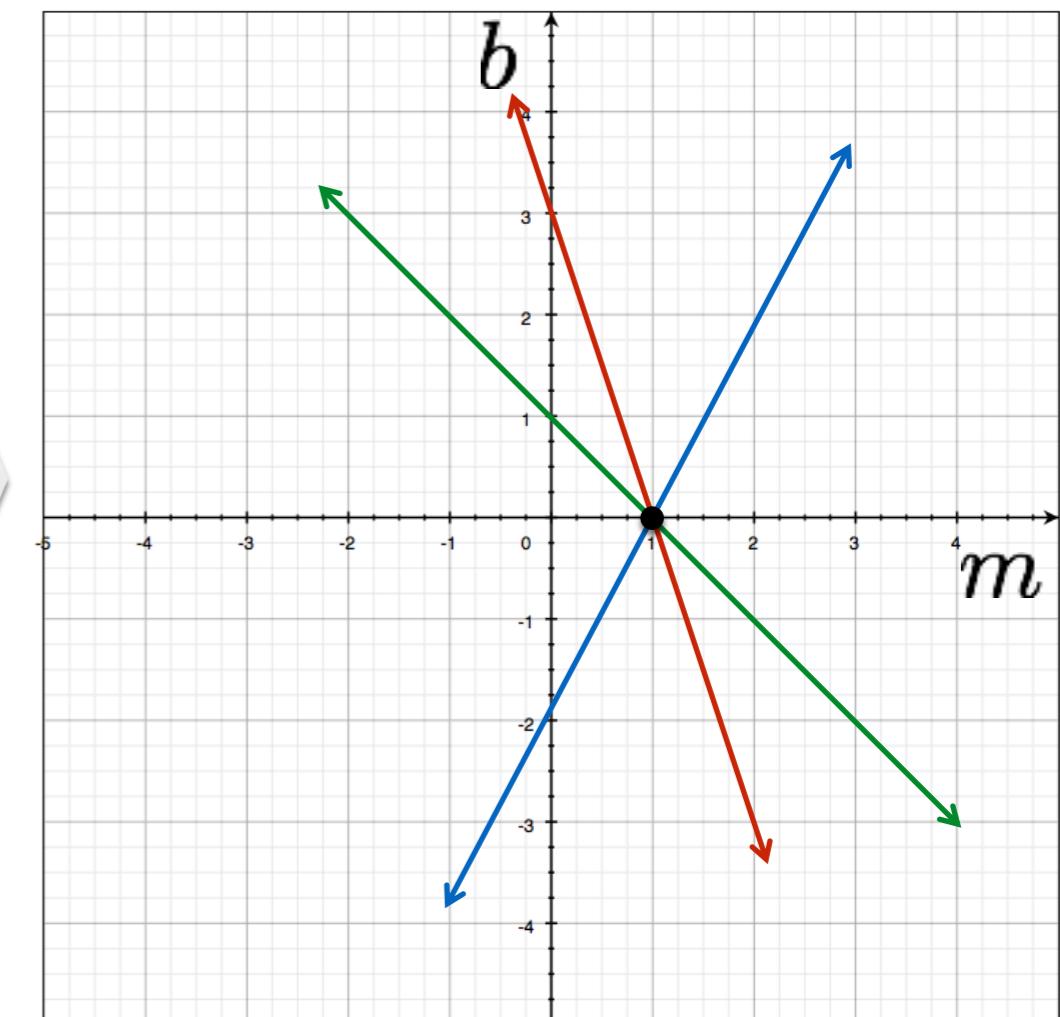
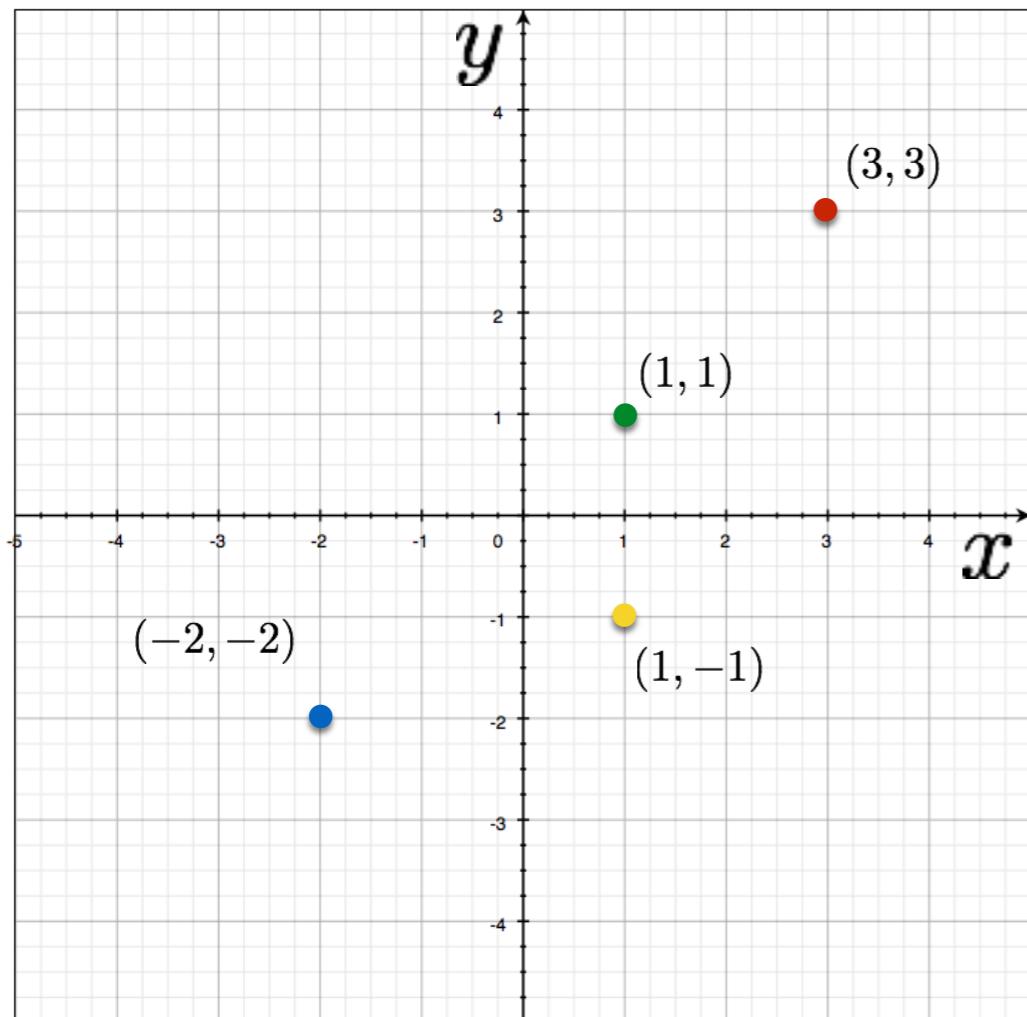


Image space

Parameter space

# Image and parameter space

variables  
 $y = mx + b$   
parameters



four points  
become  
?

variables  
 $y - mx = b$   
parameters

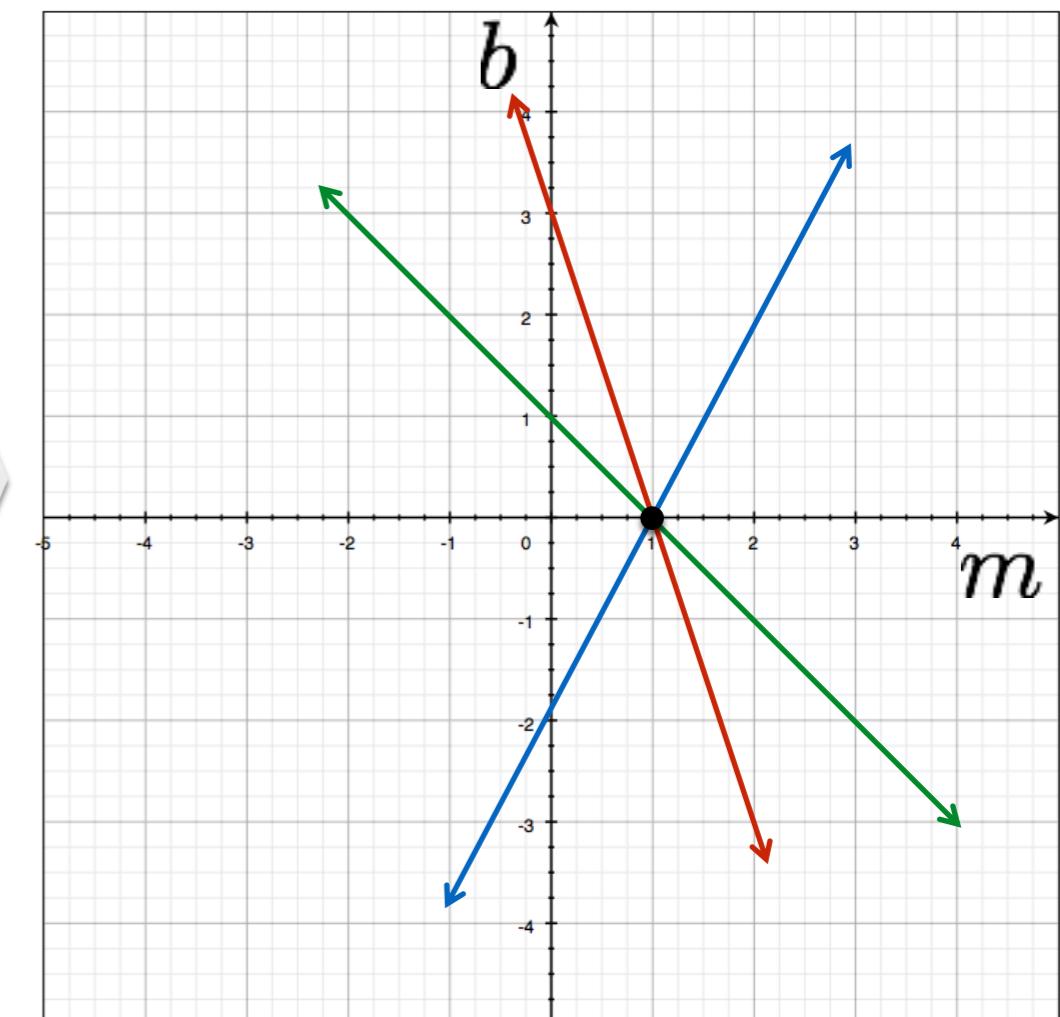
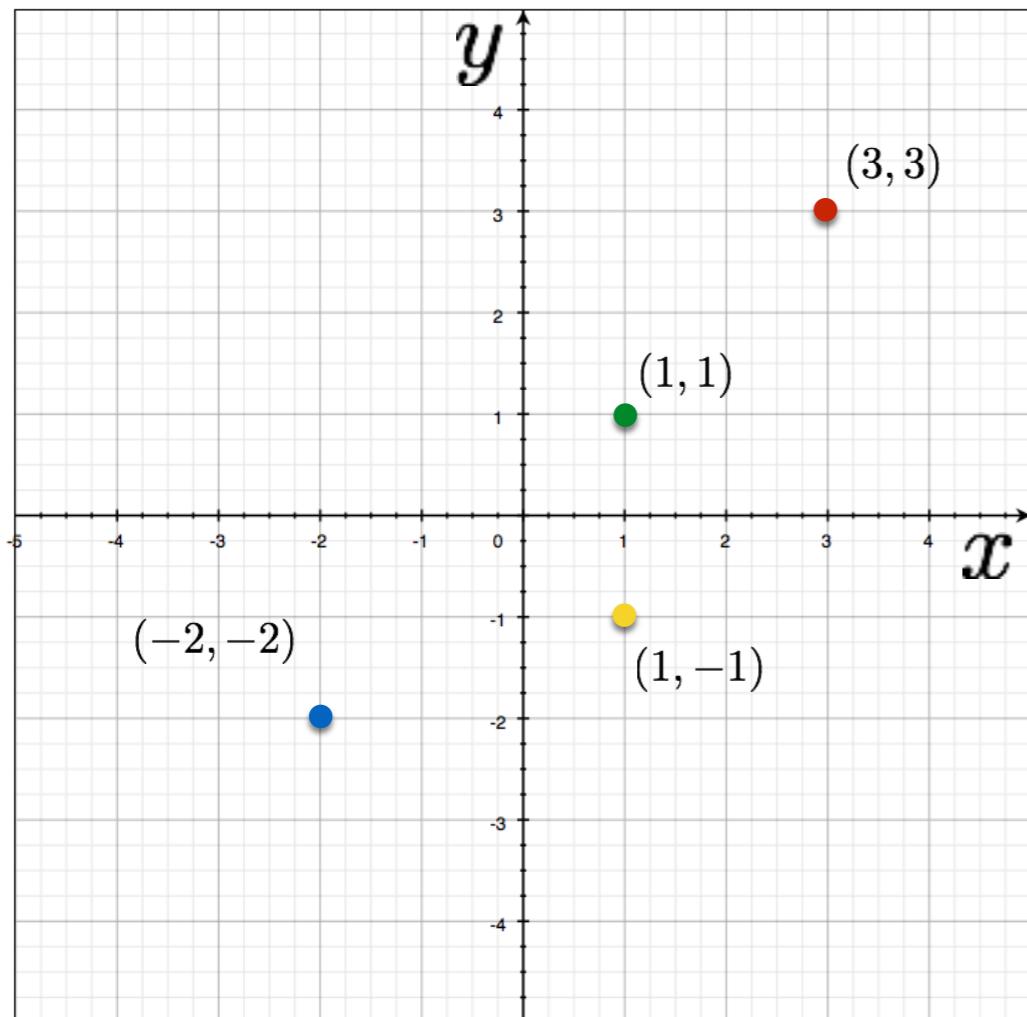


Image space

Parameter space

# Image and parameter space

variables  
 $y = mx + b$   
parameters



four points  
become  
?

variables  
 $y - mx = b$   
parameters

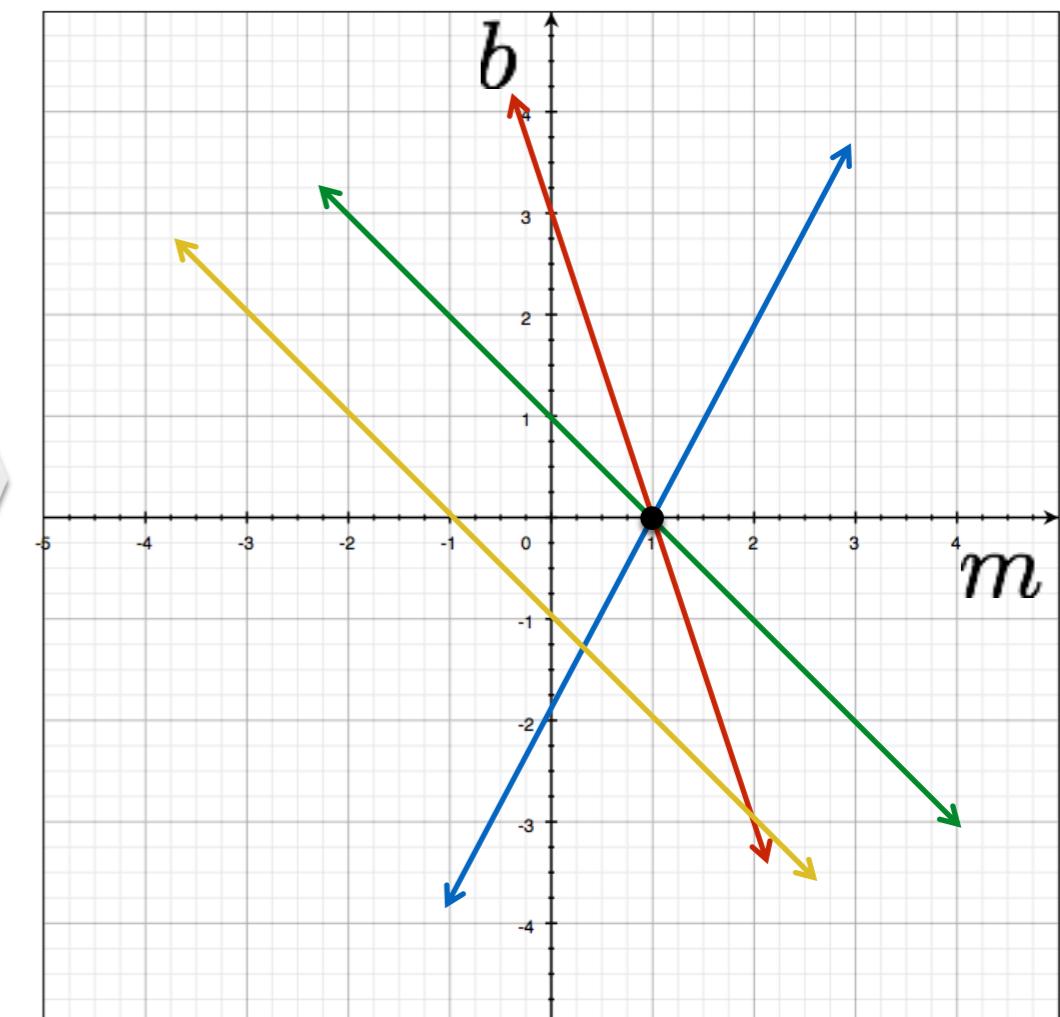


Image space

Parameter space

# How would you find the best fitting line?

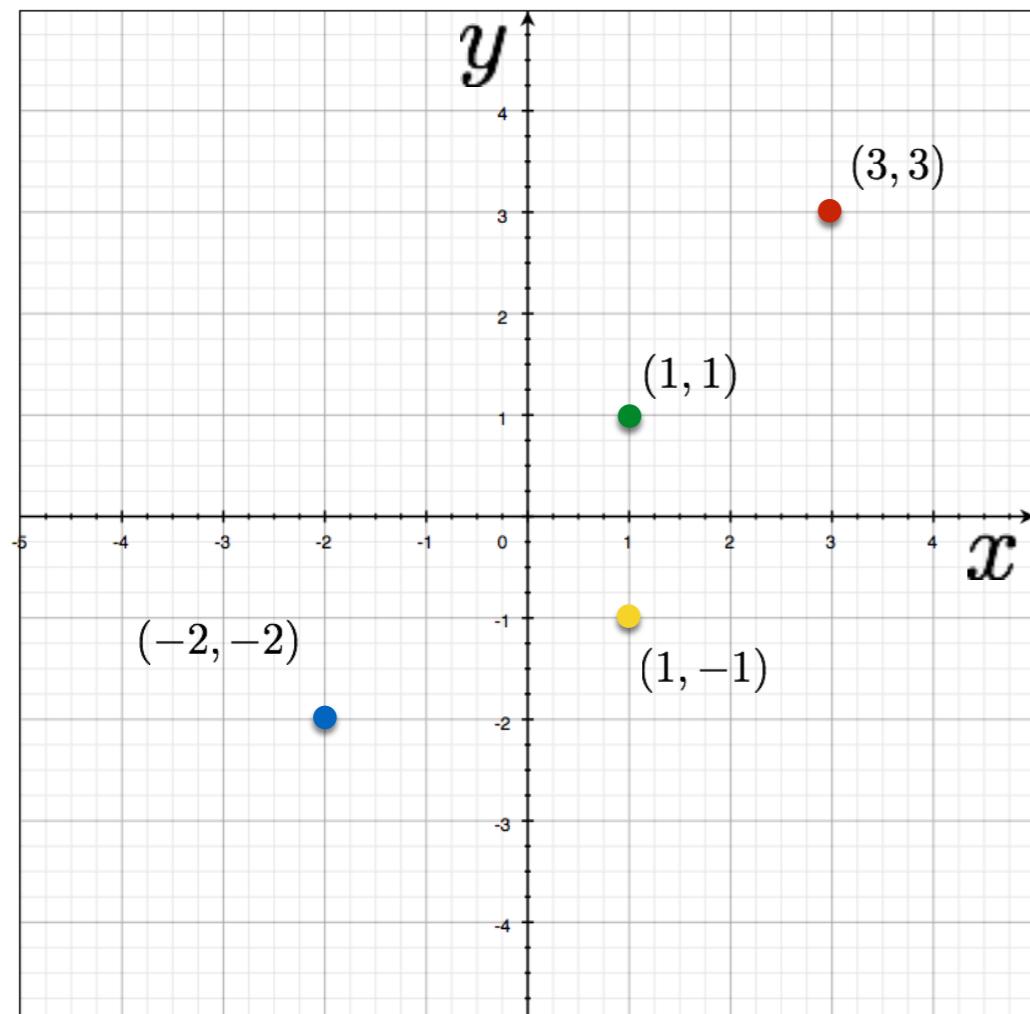
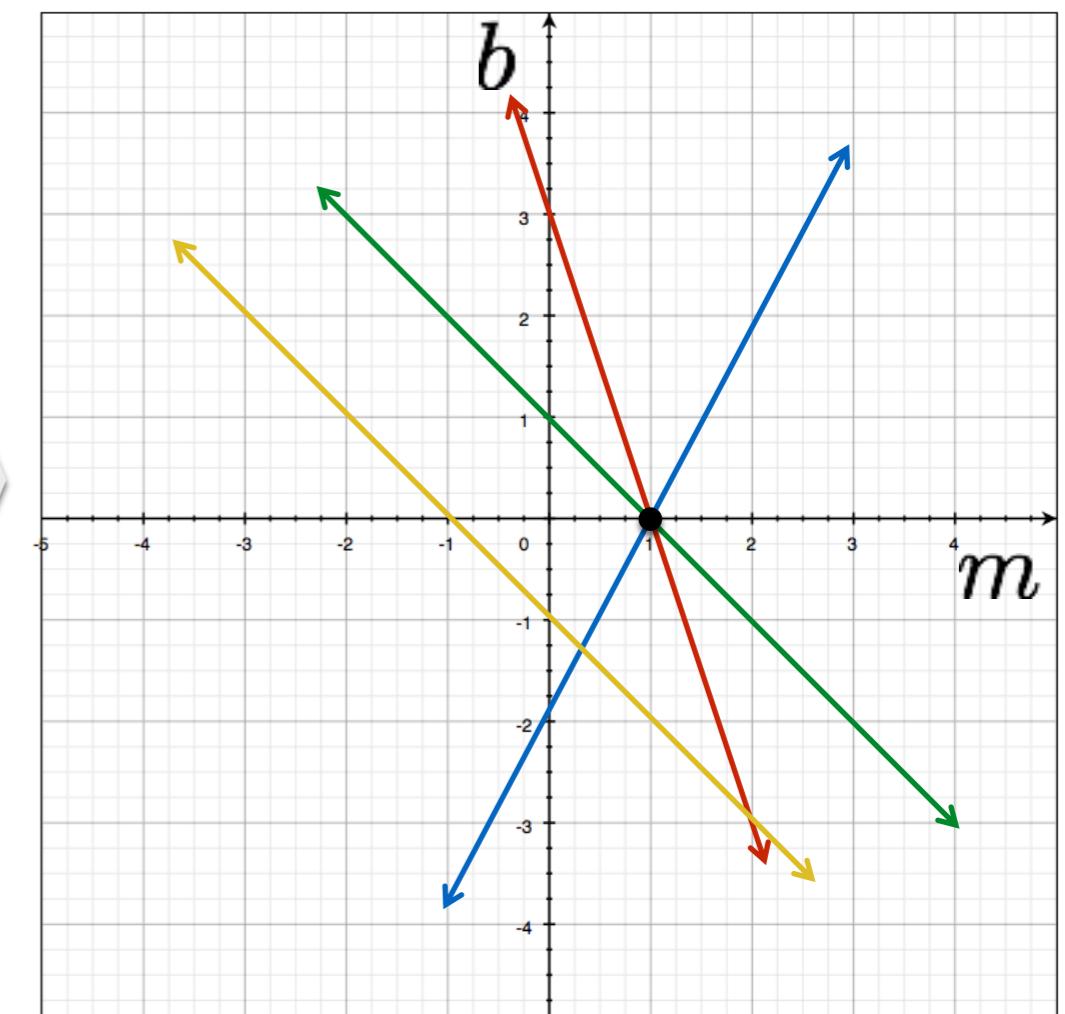


Image space



Parameter space

*Is this method robust to measurement noise?*

*Is this method robust to outliers?*

# Line Detection by Hough Transform

## Algorithm:

1. Quantize Parameter Space  $(m, c)$

2. Create Accumulator Array  $A(m, c)$

3. Set  $A(m, c) = 0 \quad \forall m, c$

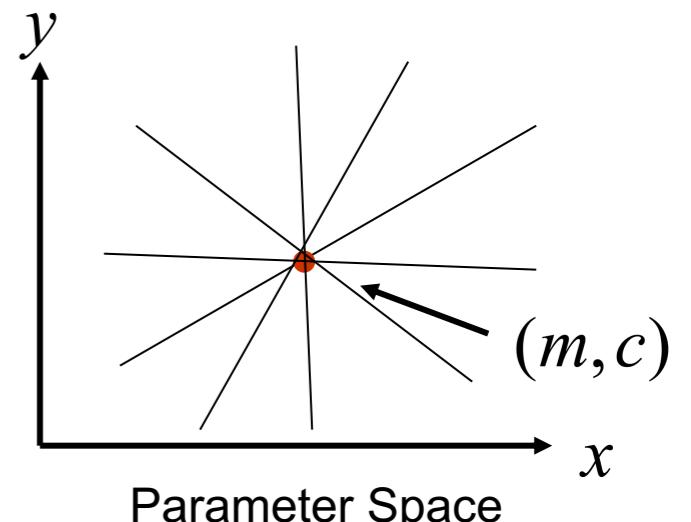
4. For each image edge  $(x_i, y_i)$

For each element in  $A(m, c)$

If  $(m, c)$  lies on the line:  $c = -x_i m + y_i$

Increment  $A(m, c) = A(m, c) + 1$

5. Find local maxima in  $A(m, c)$



$A(m, c)$

1				1	
	1			1	
		1	1		
				2	
		1	1		
	1			1	
1				1	

# Problems with parameterization

*How big does the accumulator need to be for the parameterization  $(m, c)$ ?*

$$A(m,c)$$

# Problems with parameterization

*How big does the accumulator need to be for the parameterization  $(m, c)$ ?*

$$A(m,c)$$

# The space of $m$ is huge!

$$-\infty \leq m \leq \infty$$

# The space of c is huge!

$$-\infty \leq c \leq \infty$$

# Better Parameterization

Use normal form:

$$x \cos \theta + y \sin \theta = \rho$$

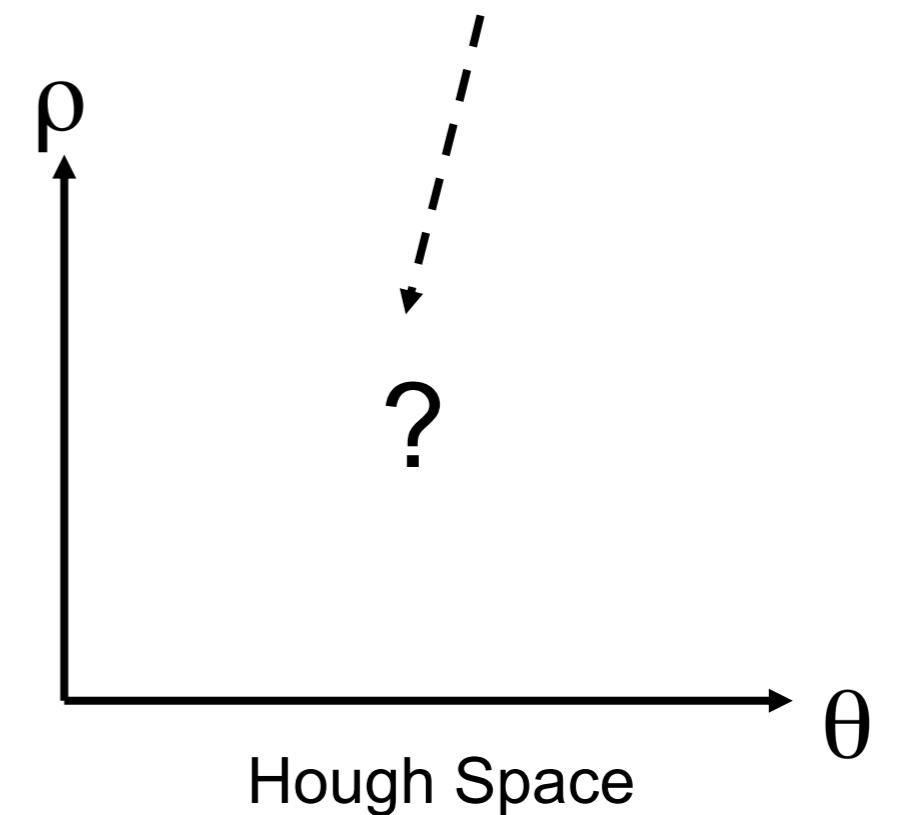
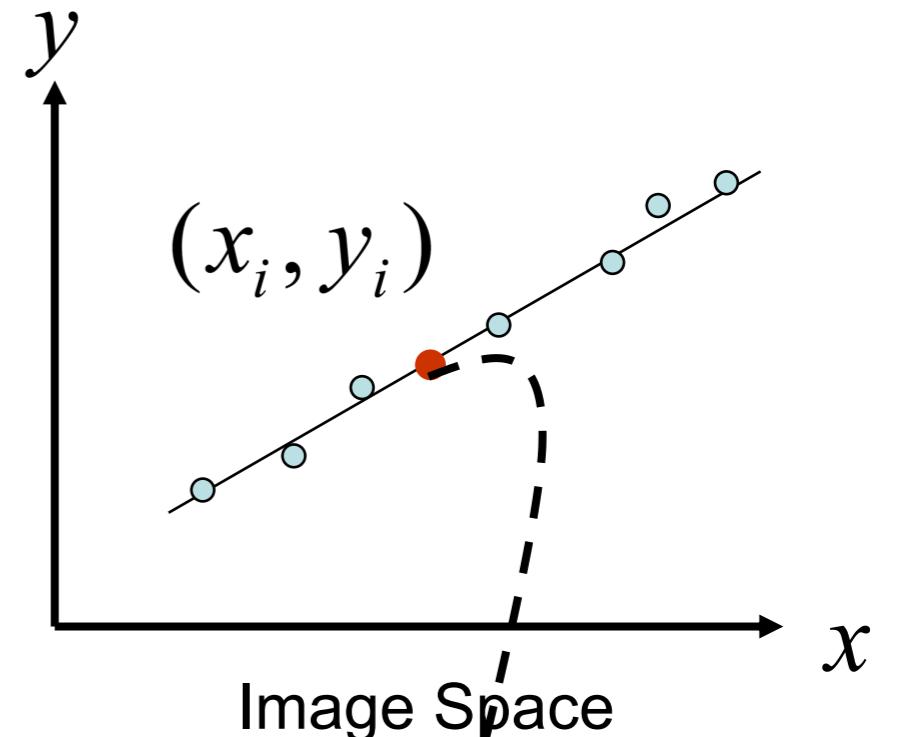
Given points  $(x_i, y_i)$  find  $(\rho, \theta)$

Hough Space Sinusoid

$$0 \leq \theta \leq 2\pi$$

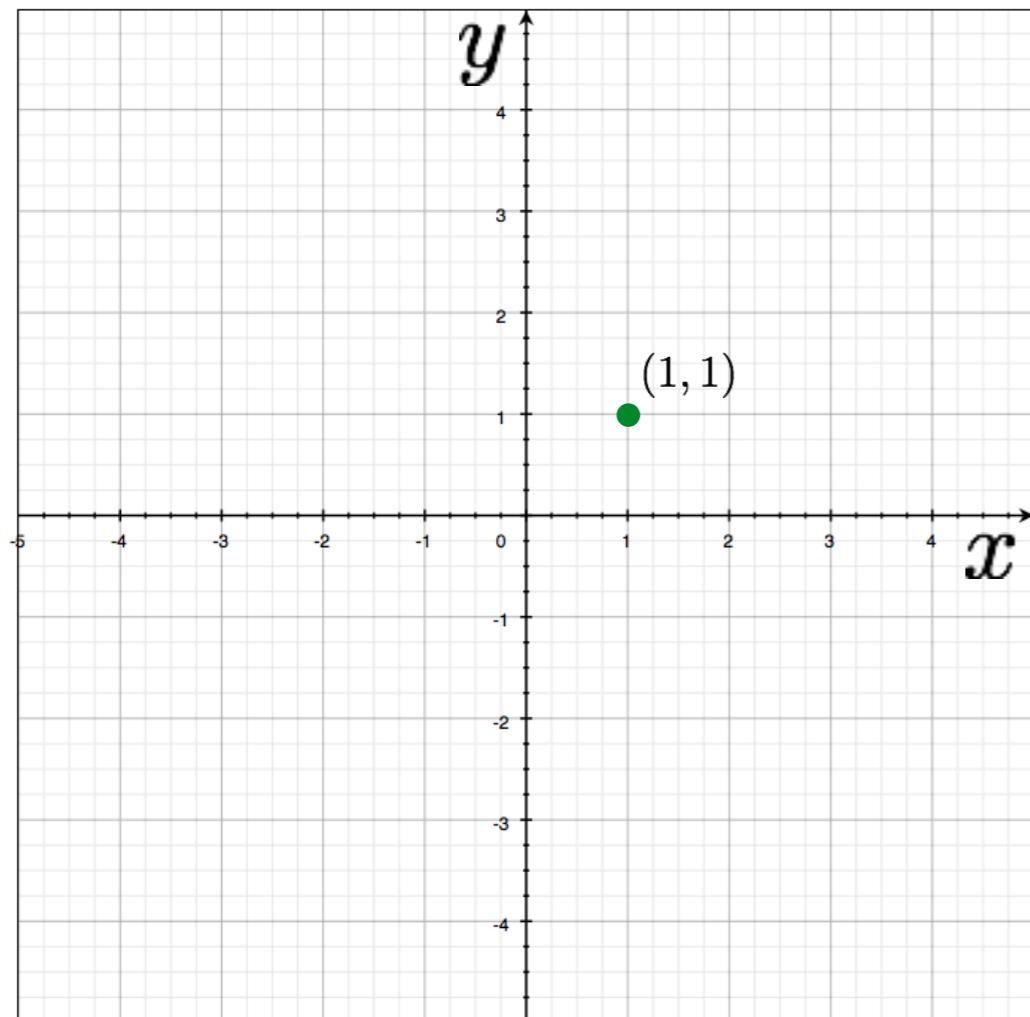
$$0 \leq \rho \leq \rho_{\max}$$

(Finite Accumulator Array Size)



# Image and parameter space

variables  
 $y = mx + b$   
parameters



a point becomes?

parameters  
 $x \cos \theta + y \sin \theta = \rho$   
variables

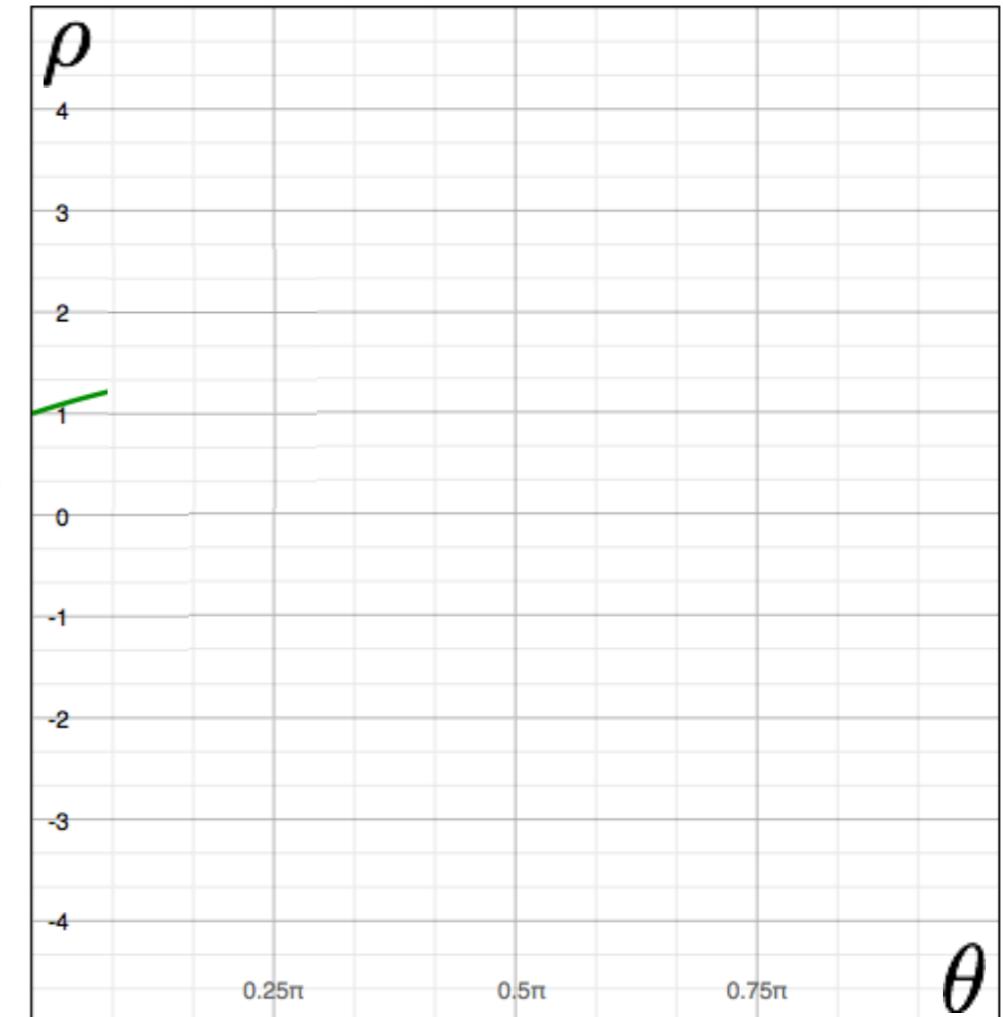
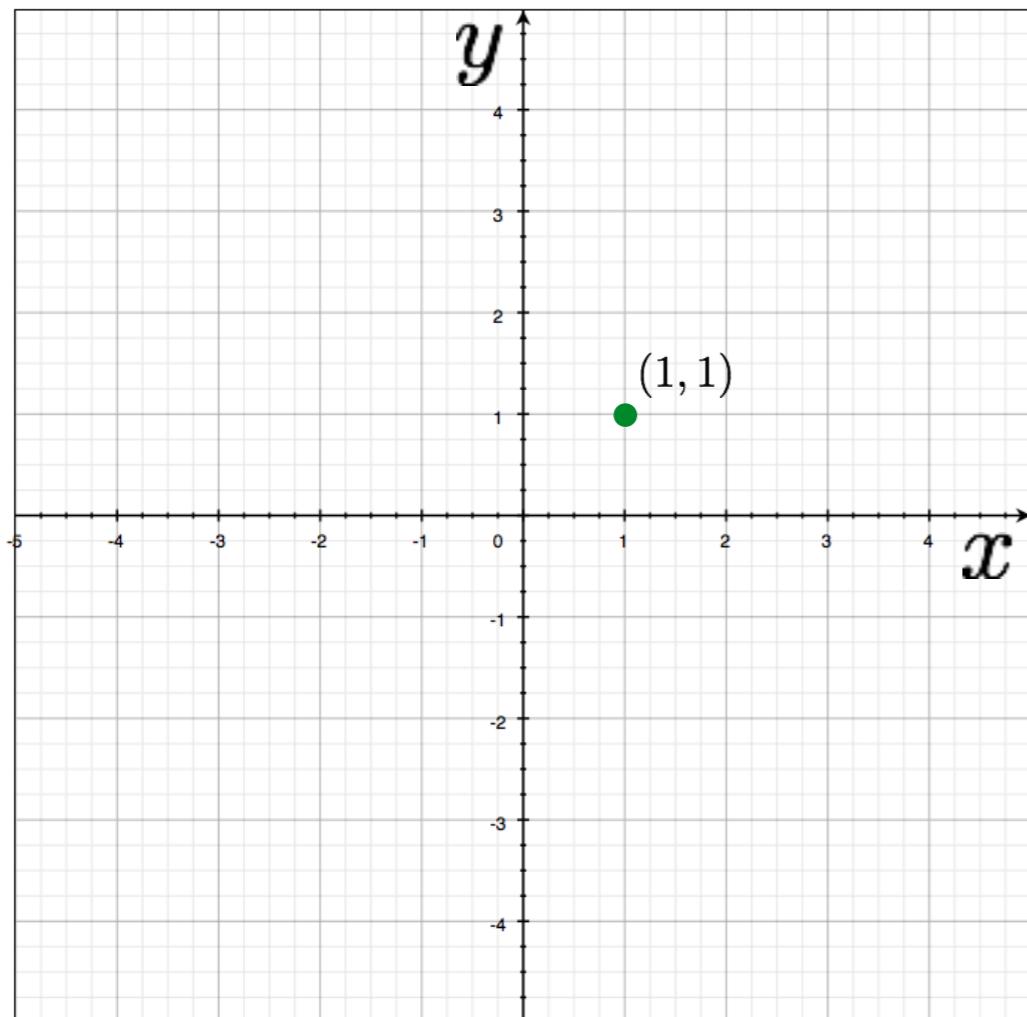


Image space

Parameter space

# Image and parameter space

variables  
 $y = mx + b$   
parameters



a point becomes a wave

parameters  
 $x \cos \theta + y \sin \theta = \rho$   
variables

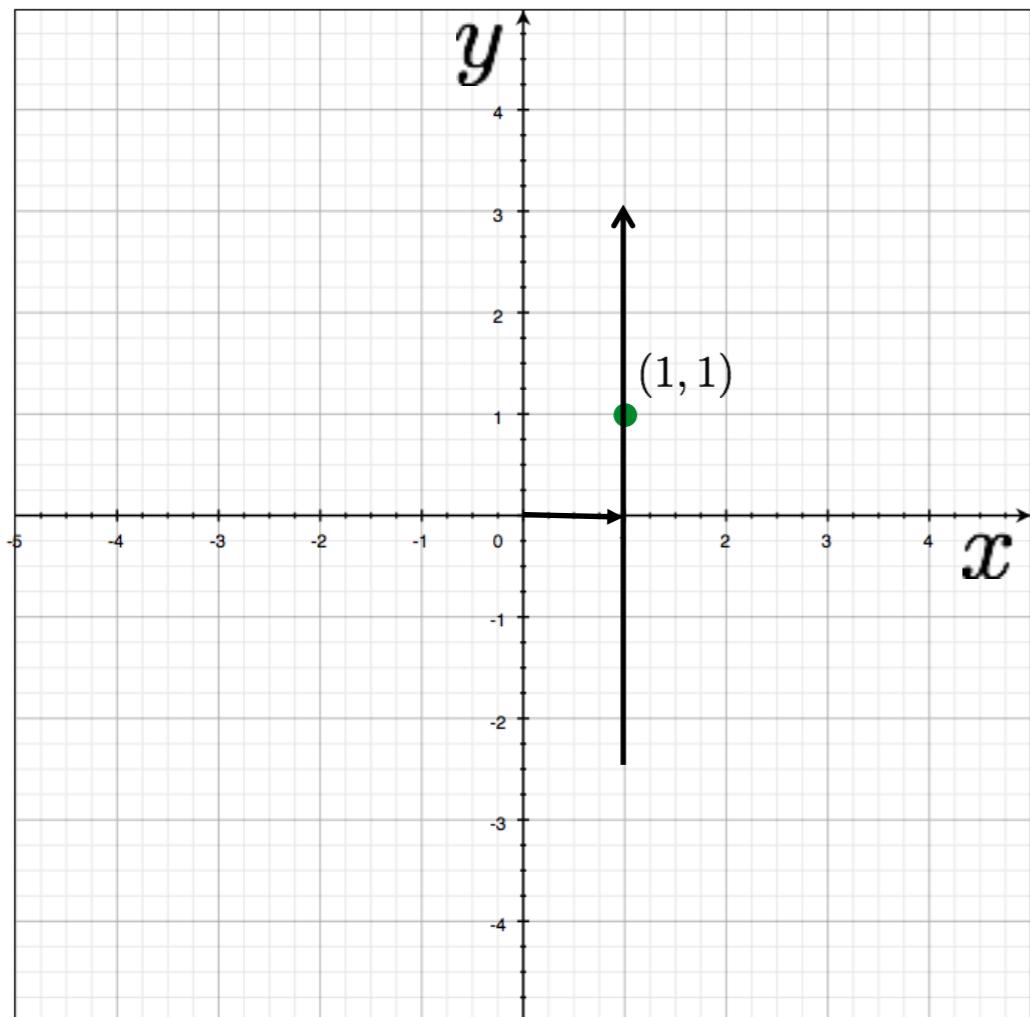


Image space

Parameter space

# Image and parameter space

variables  
 $y = mx + b$   
parameters



a line becomes?

$$x \cos \theta + y \sin \theta = \rho$$

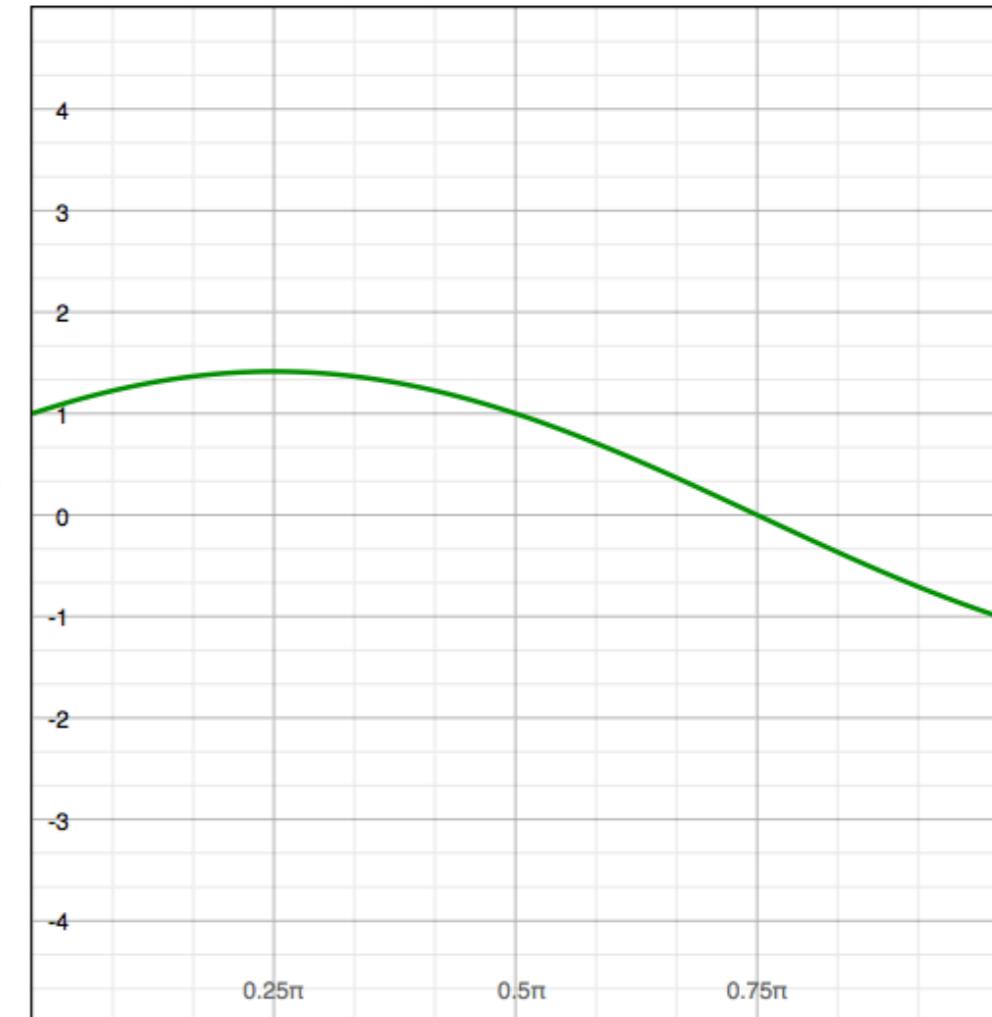


Image space

Parameter space

# Image and parameter space

variables  
 $y = mx + b$   
parameters

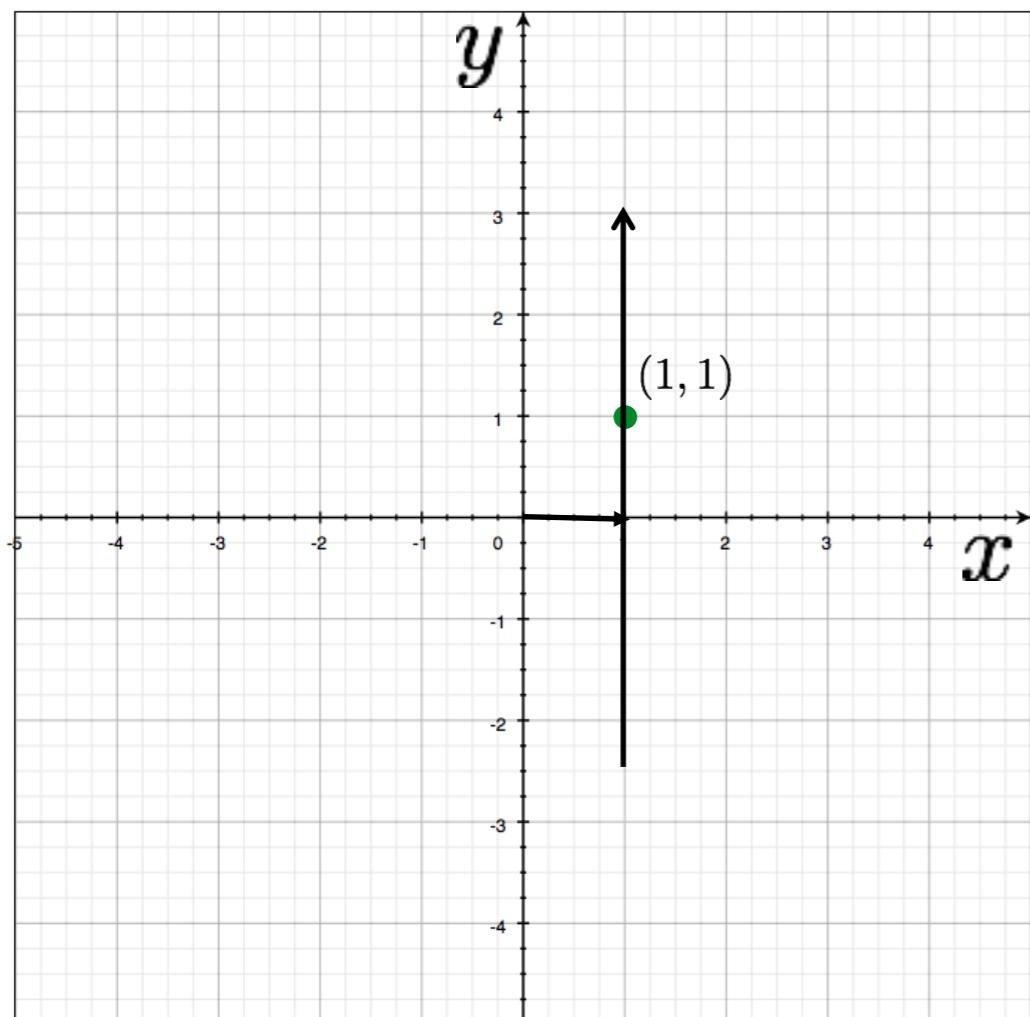
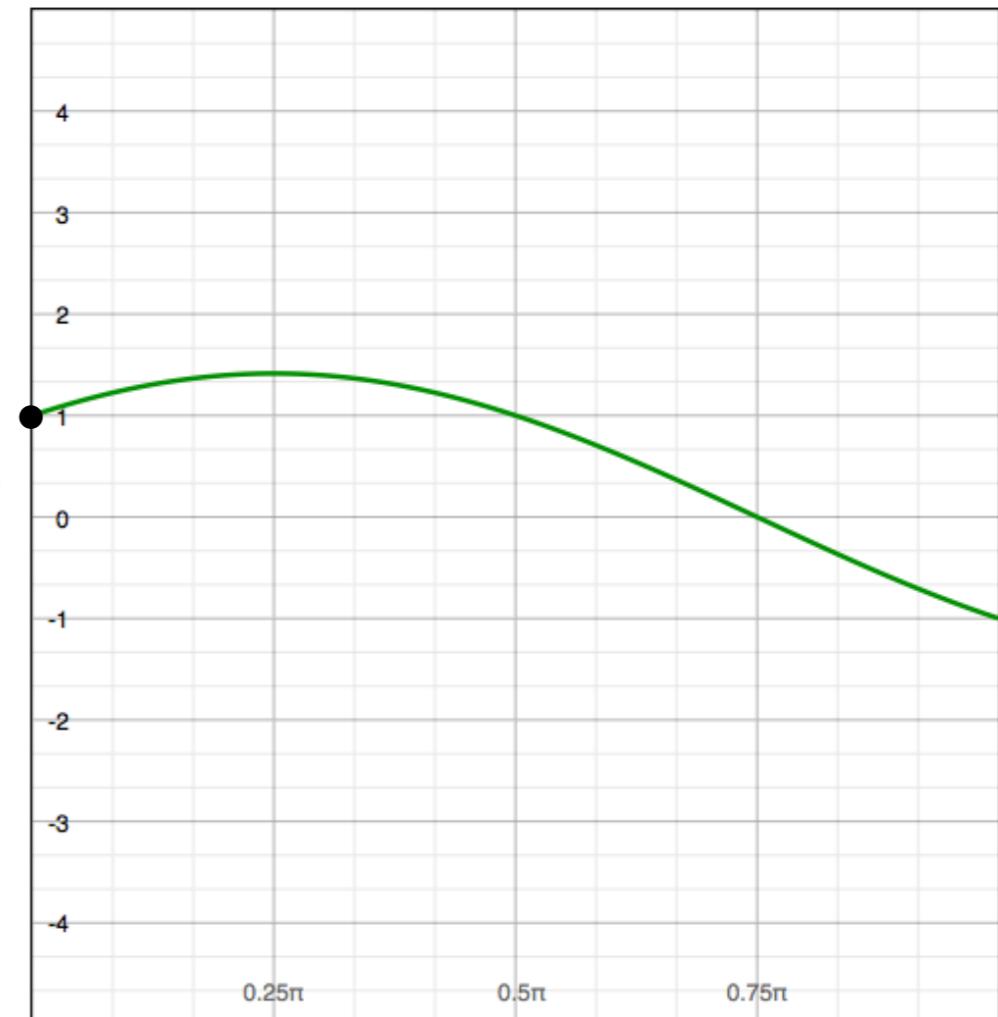


Image space

$$x \cos \theta + y \sin \theta = \rho$$

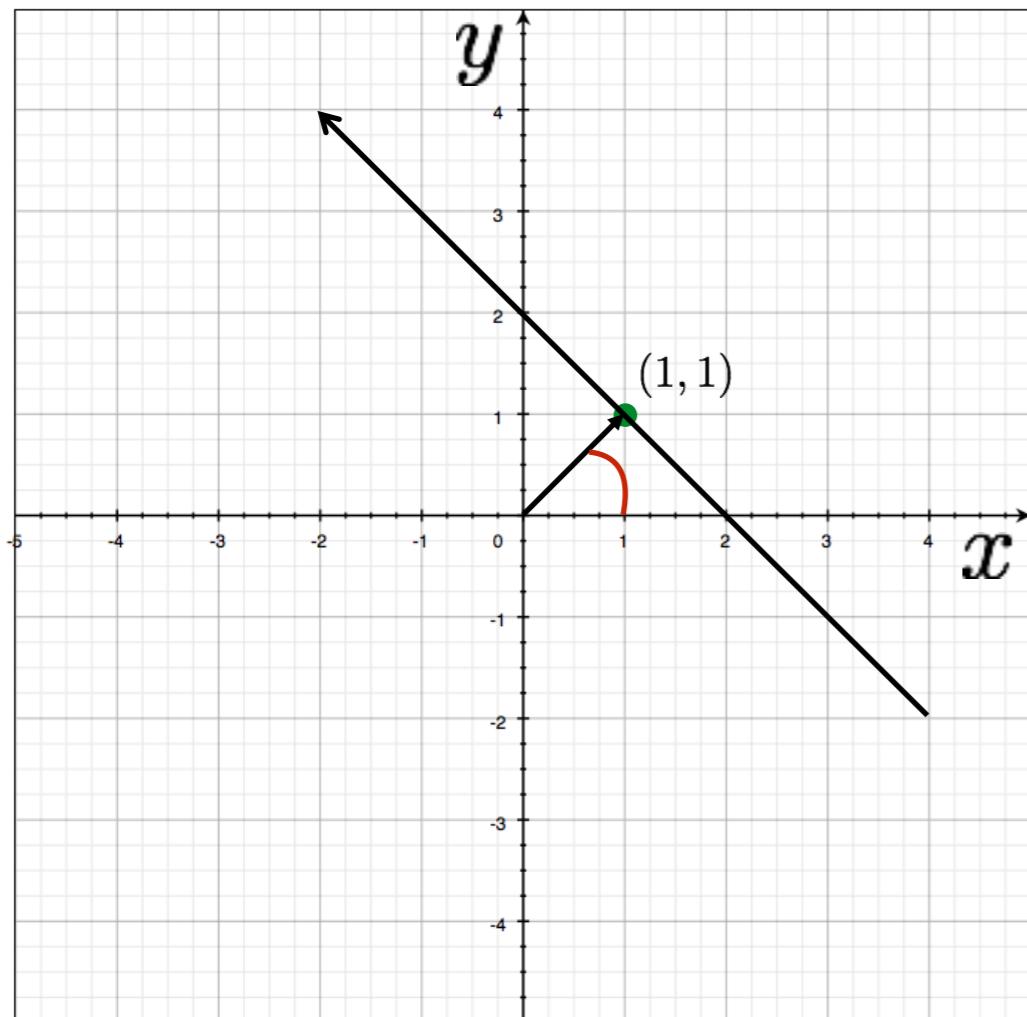
a line  
becomes a  
point



Parameter space

# Image and parameter space

variables  
 $y = mx + b$   
parameters



a line  
becomes?

$$x \cos \theta + y \sin \theta = \rho$$

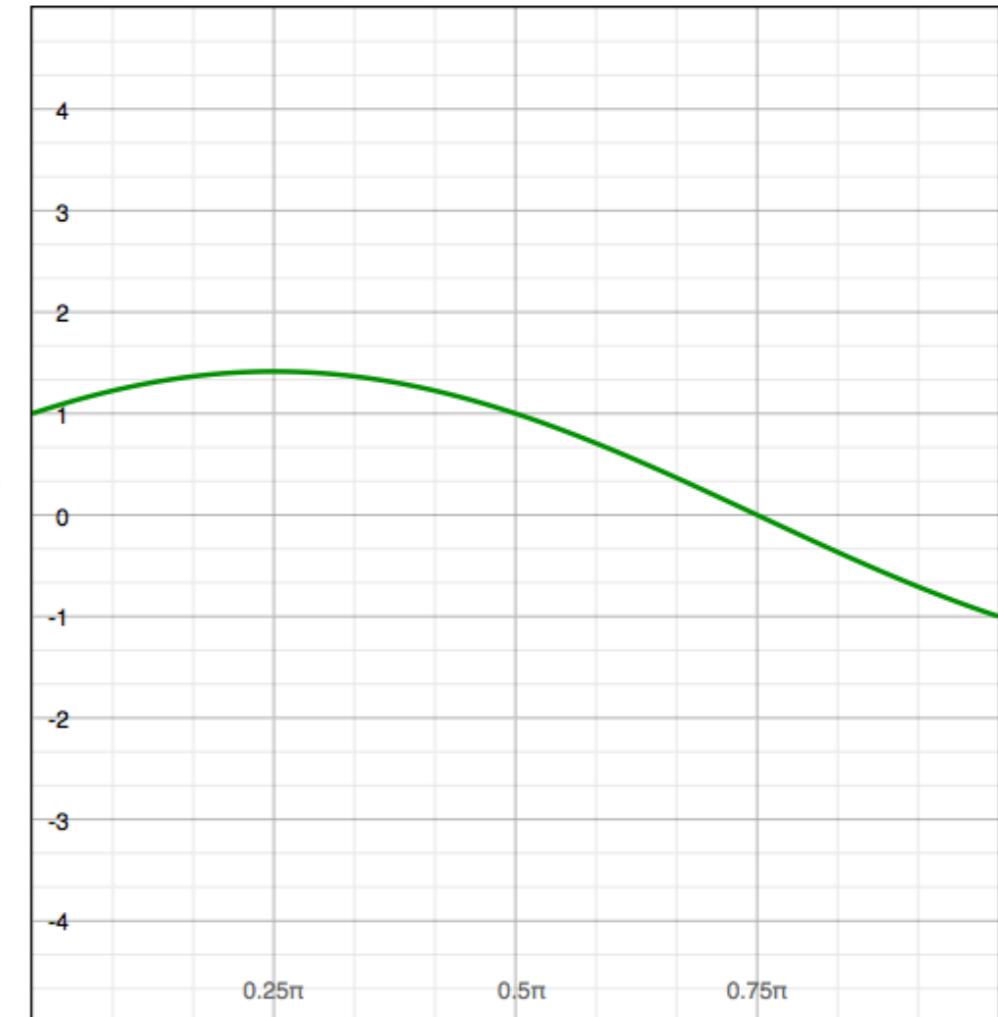


Image space

Parameter space

# Image and parameter space

variables  
 $y = mx + b$   
parameters

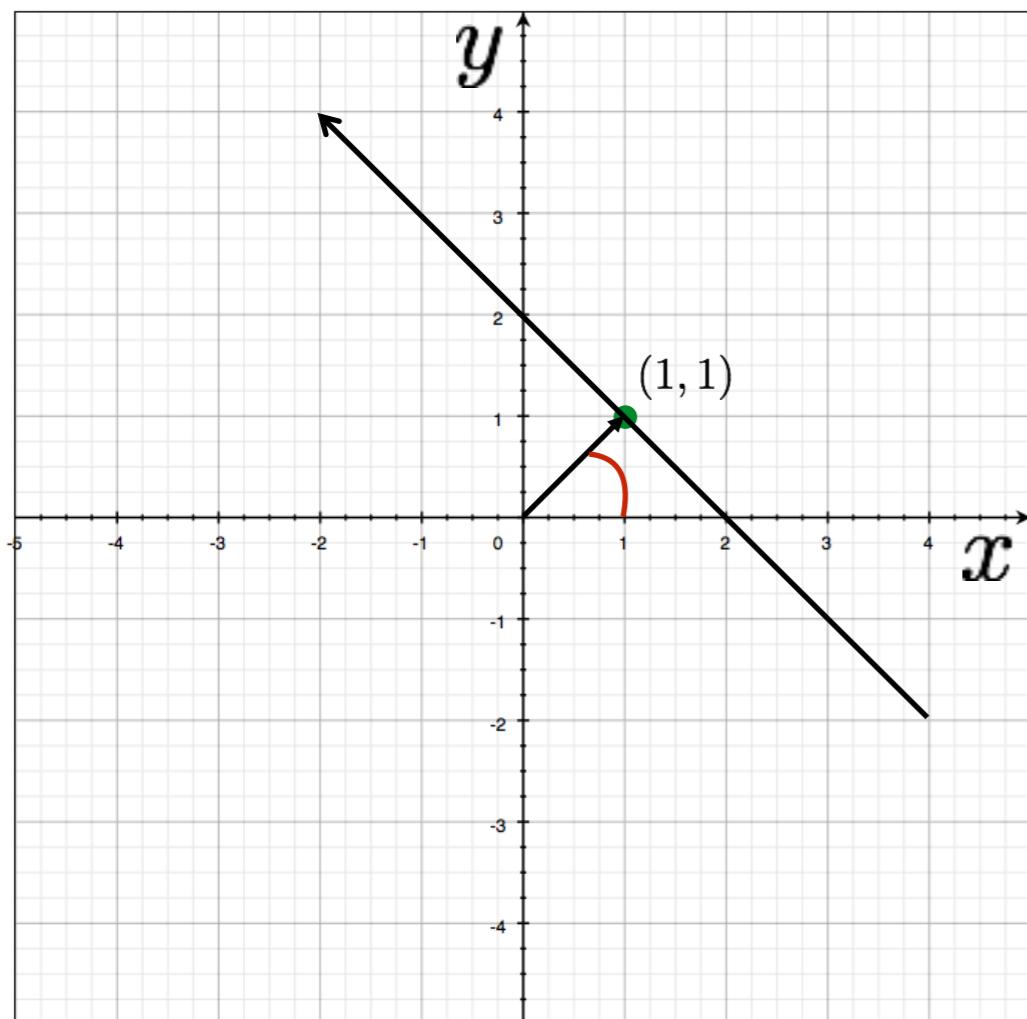
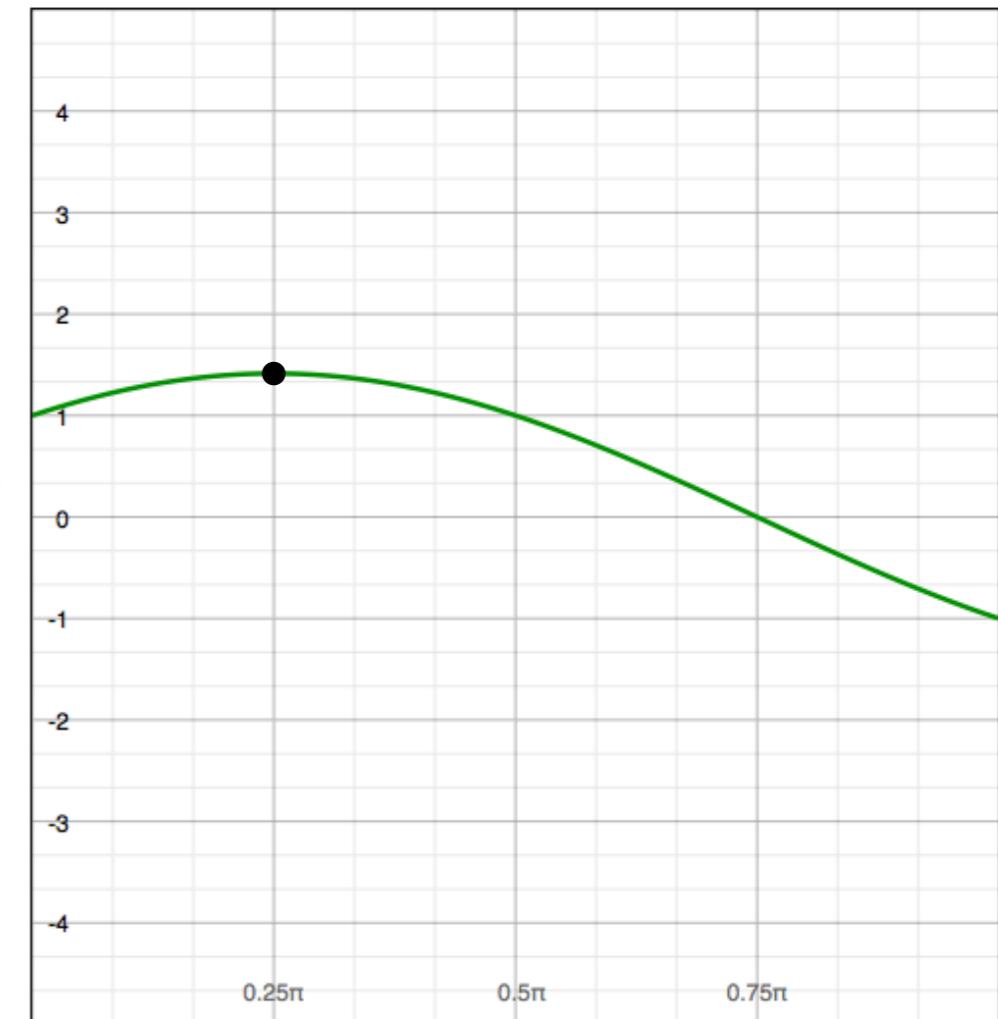


Image space

$$x \cos \theta + y \sin \theta = \rho$$

a line  
becomes a  
point



Parameter space

# Image and parameter space

variables  
 $y = mx + b$   
parameters

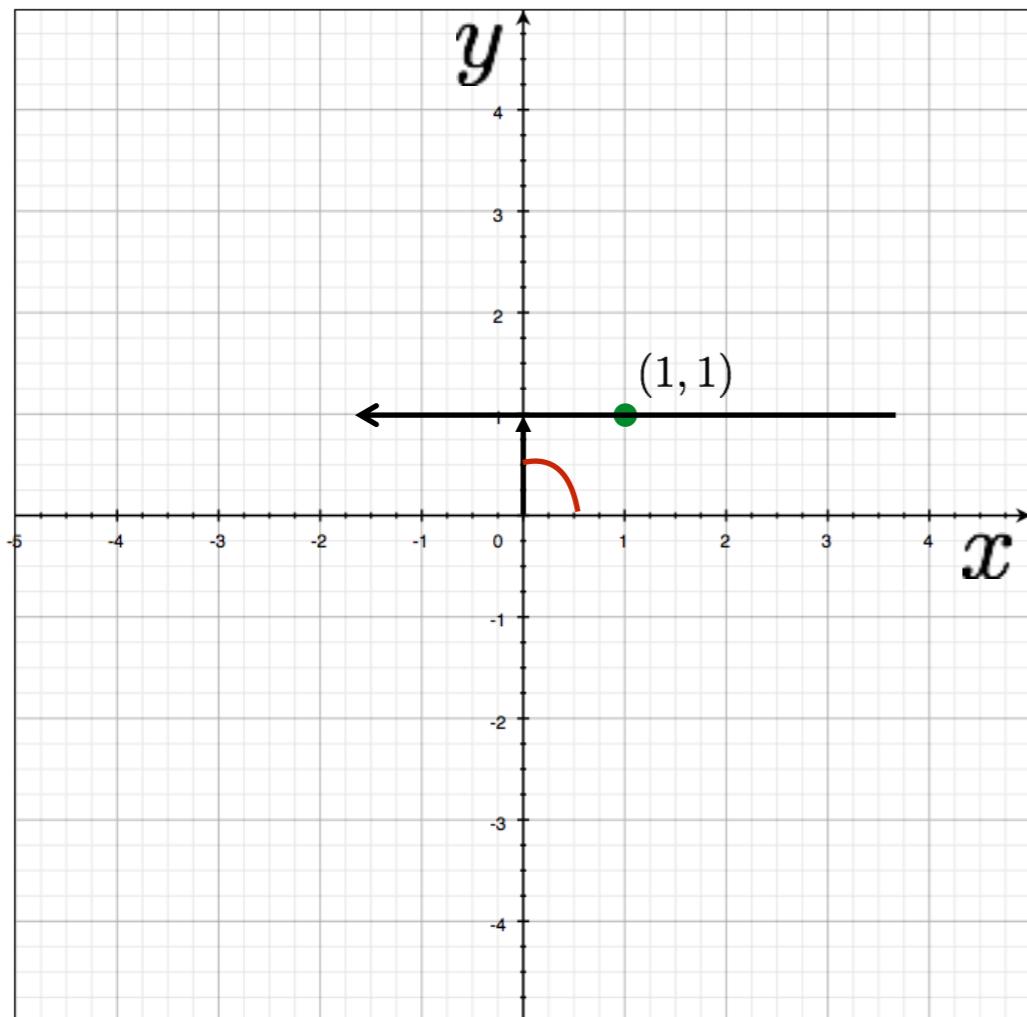
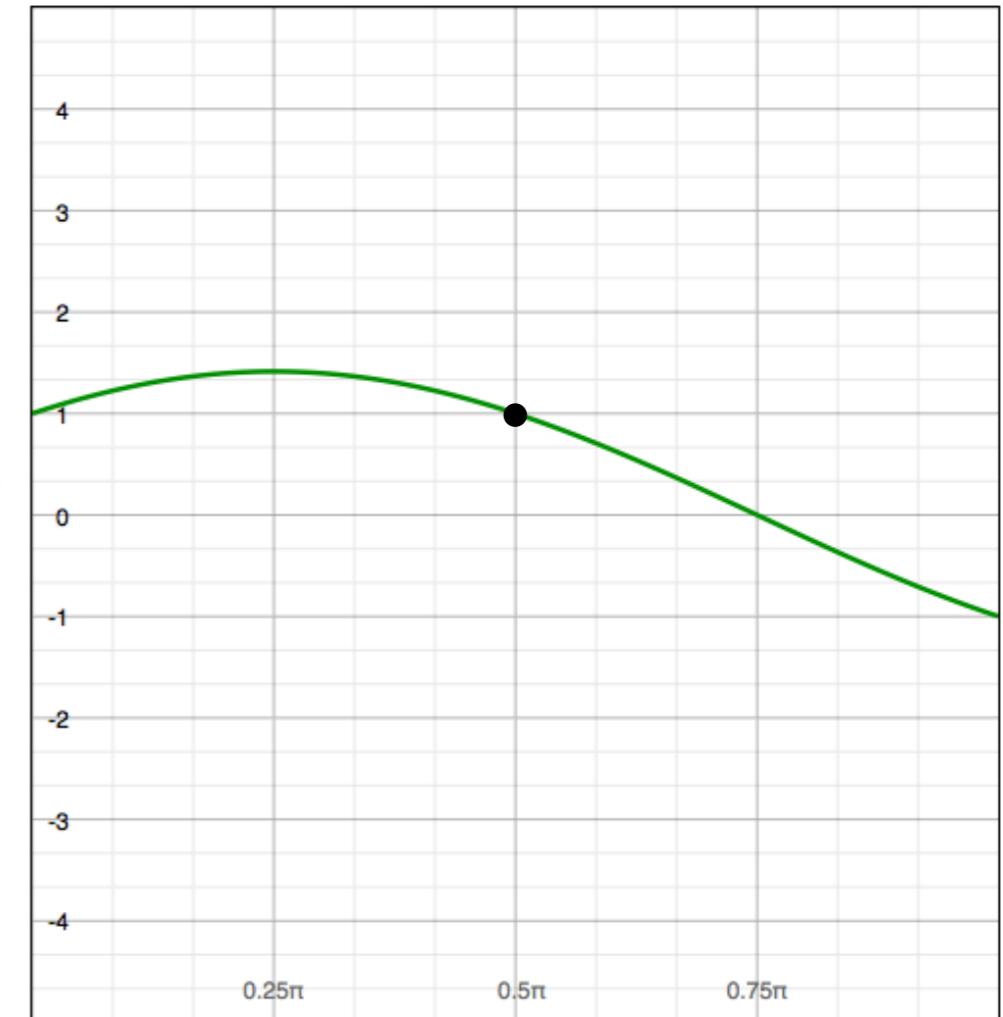


Image space

$$x \cos \theta + y \sin \theta = \rho$$

a line becomes a point



Parameter space

# Image and parameter space

variables  
 $y = mx + b$   
parameters

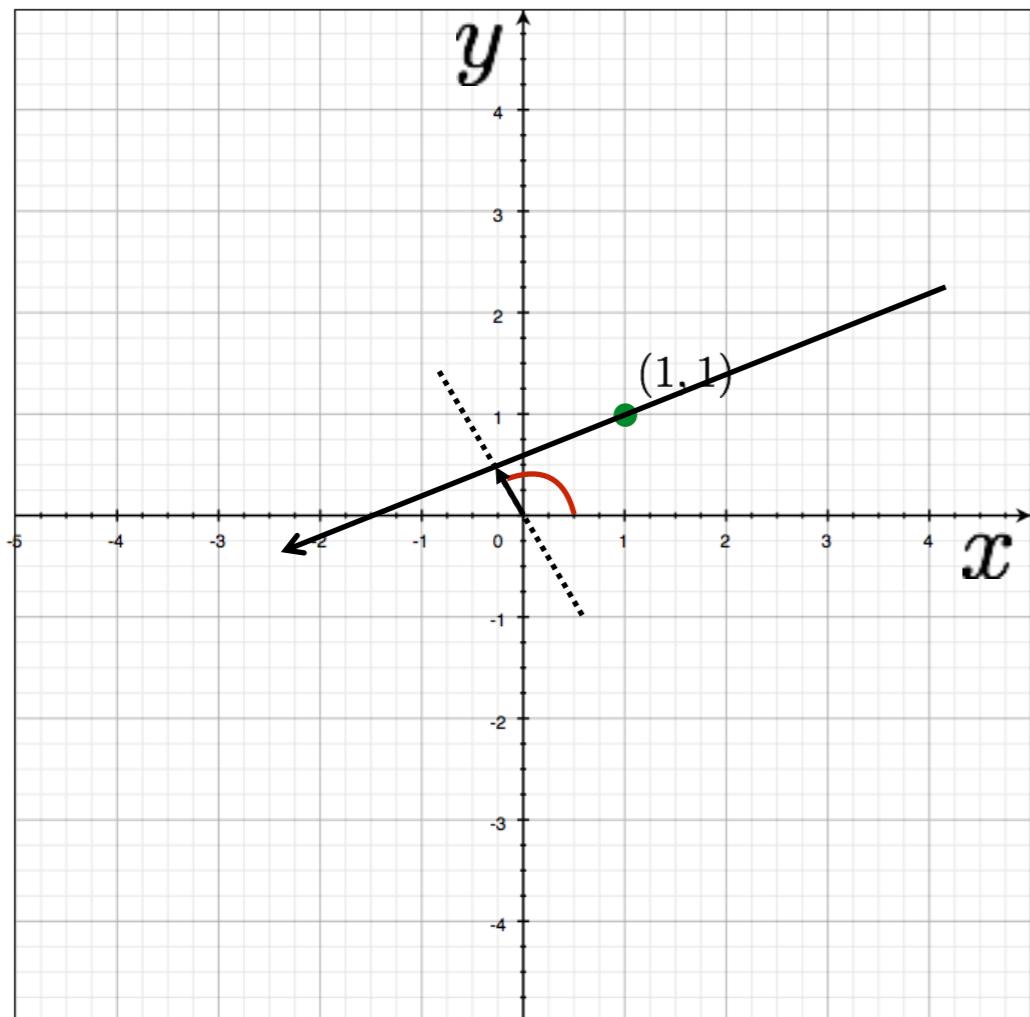
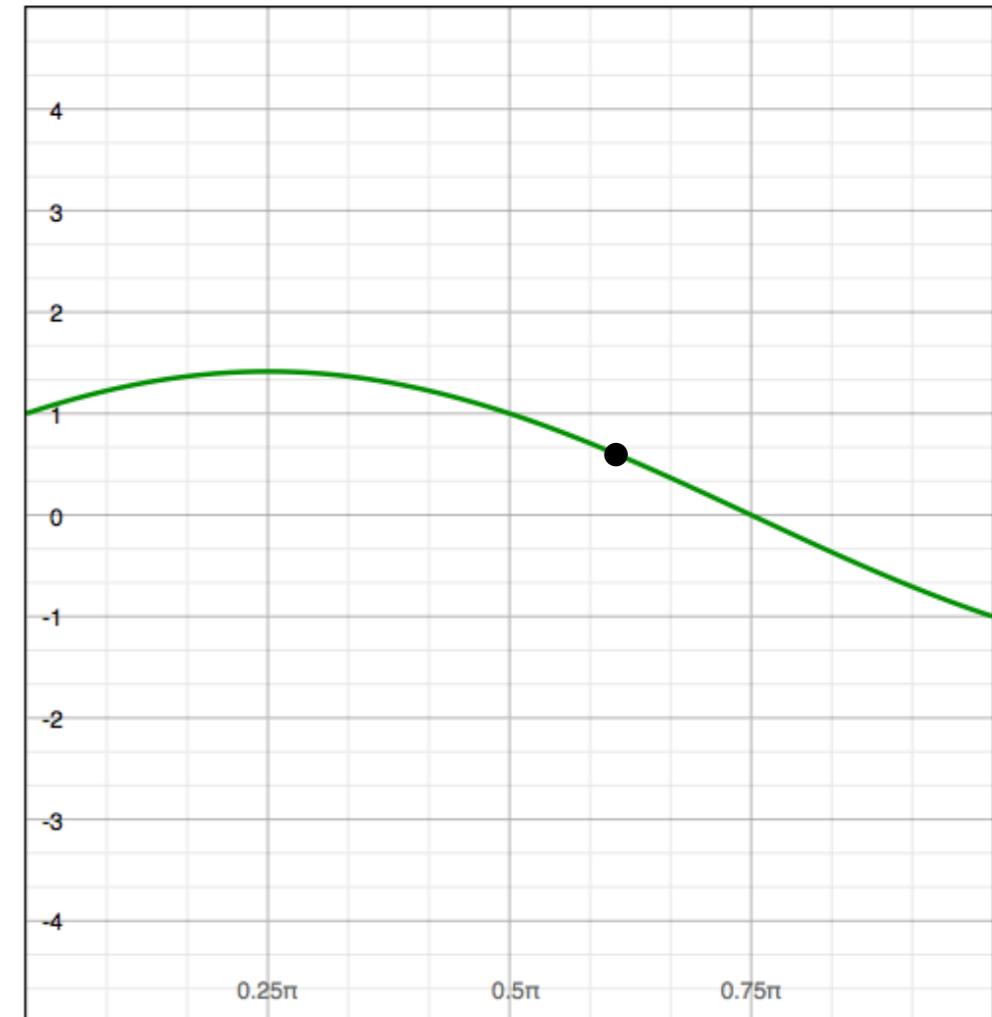


Image space

$$x \cos \theta + y \sin \theta = \rho$$

a line  
becomes a  
point



Parameter space

# Image and parameter space

variables  
 $y = mx + b$   
parameters

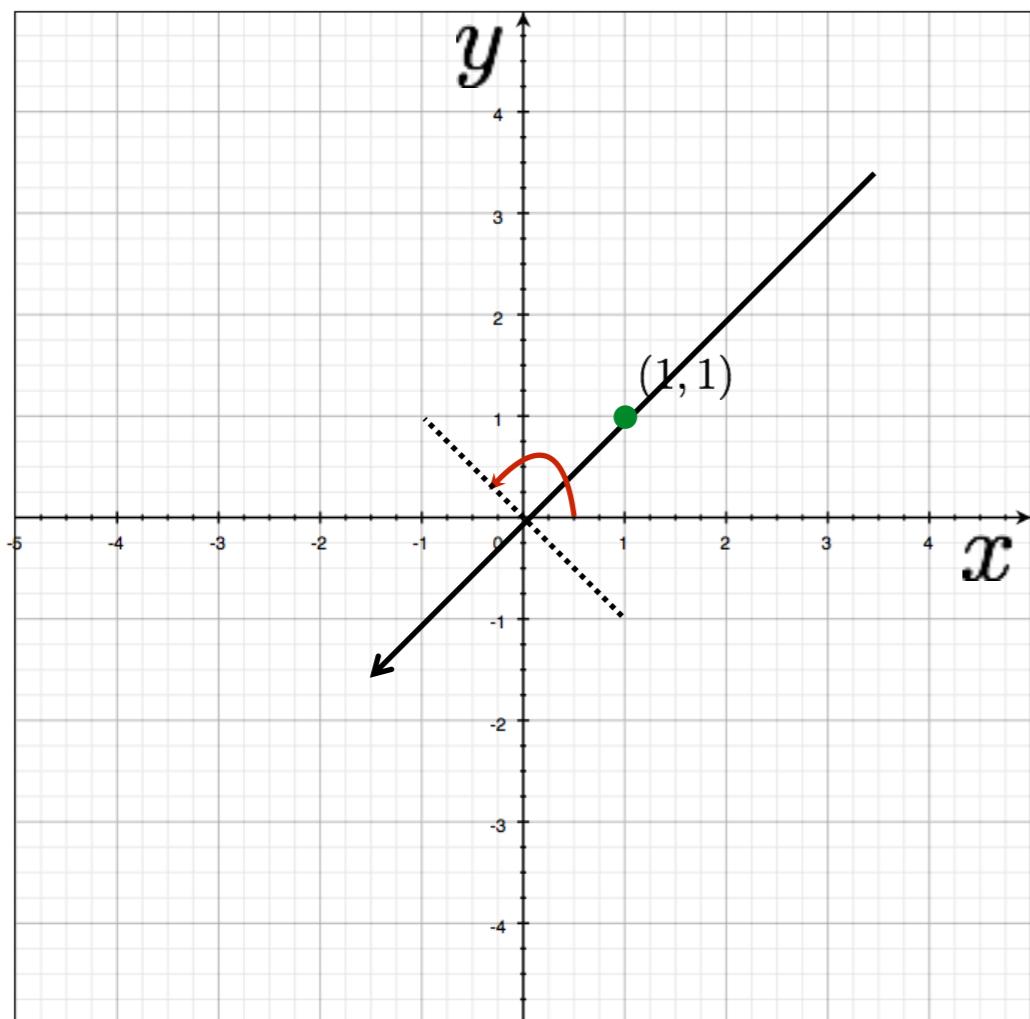
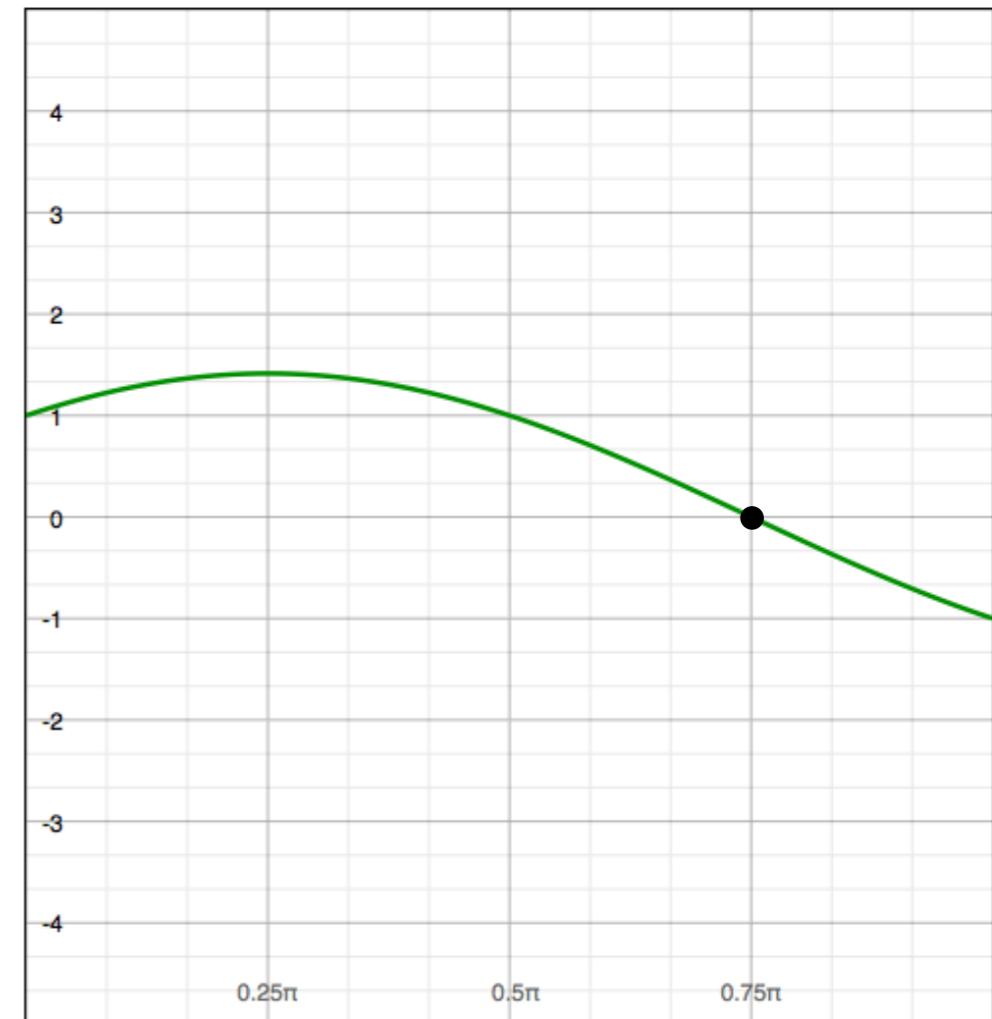


Image space

$$x \cos \theta + y \sin \theta = \rho$$

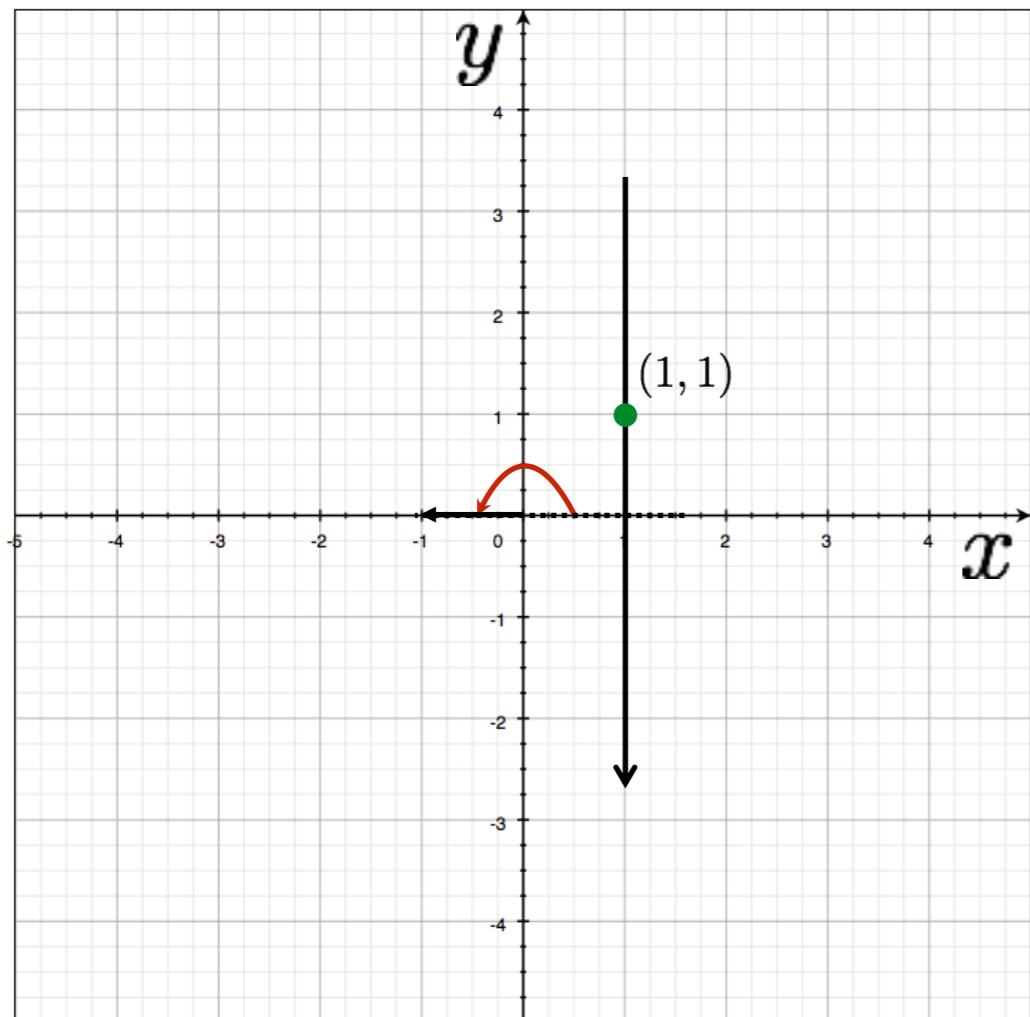
a line  
becomes a  
point



Parameter space

# Image and parameter space

variables  
 $y = mx + b$   
parameters



a line  
becomes a  
point

$$x \cos \theta + y \sin \theta = \rho$$

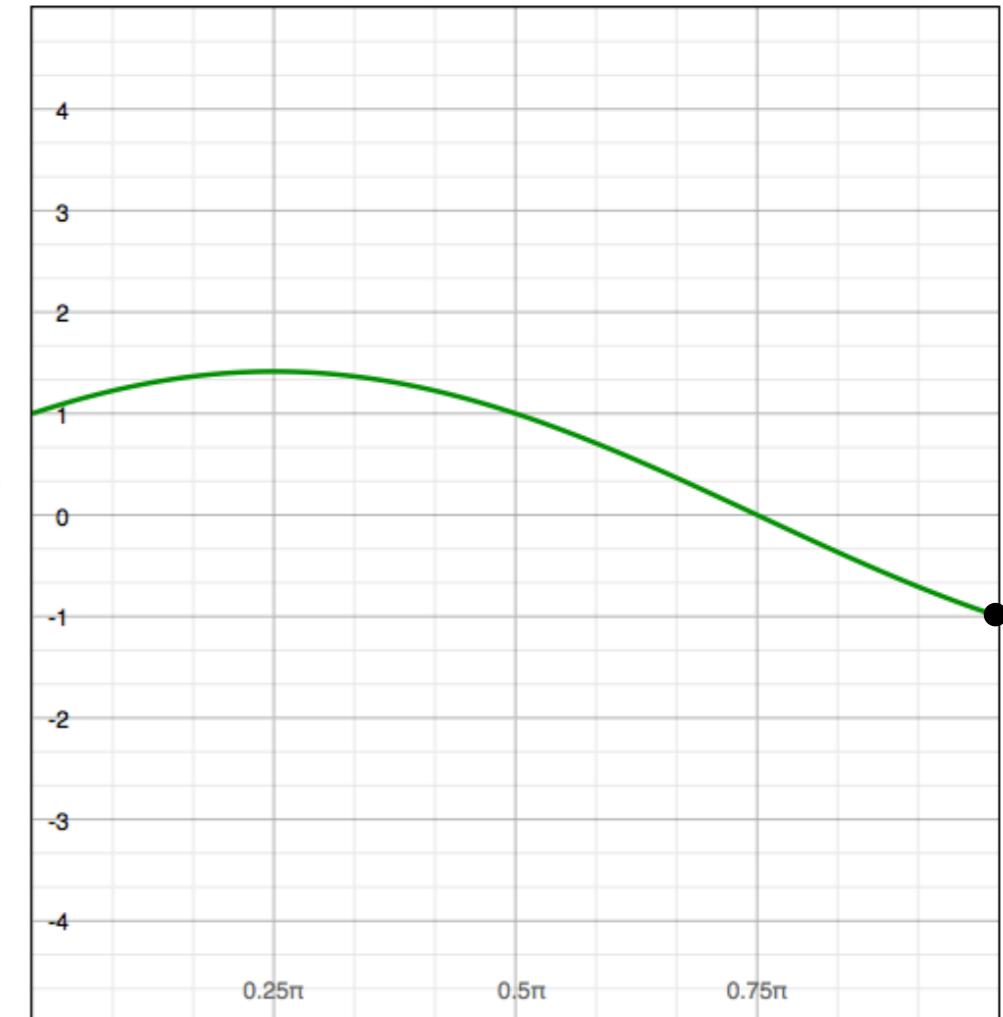
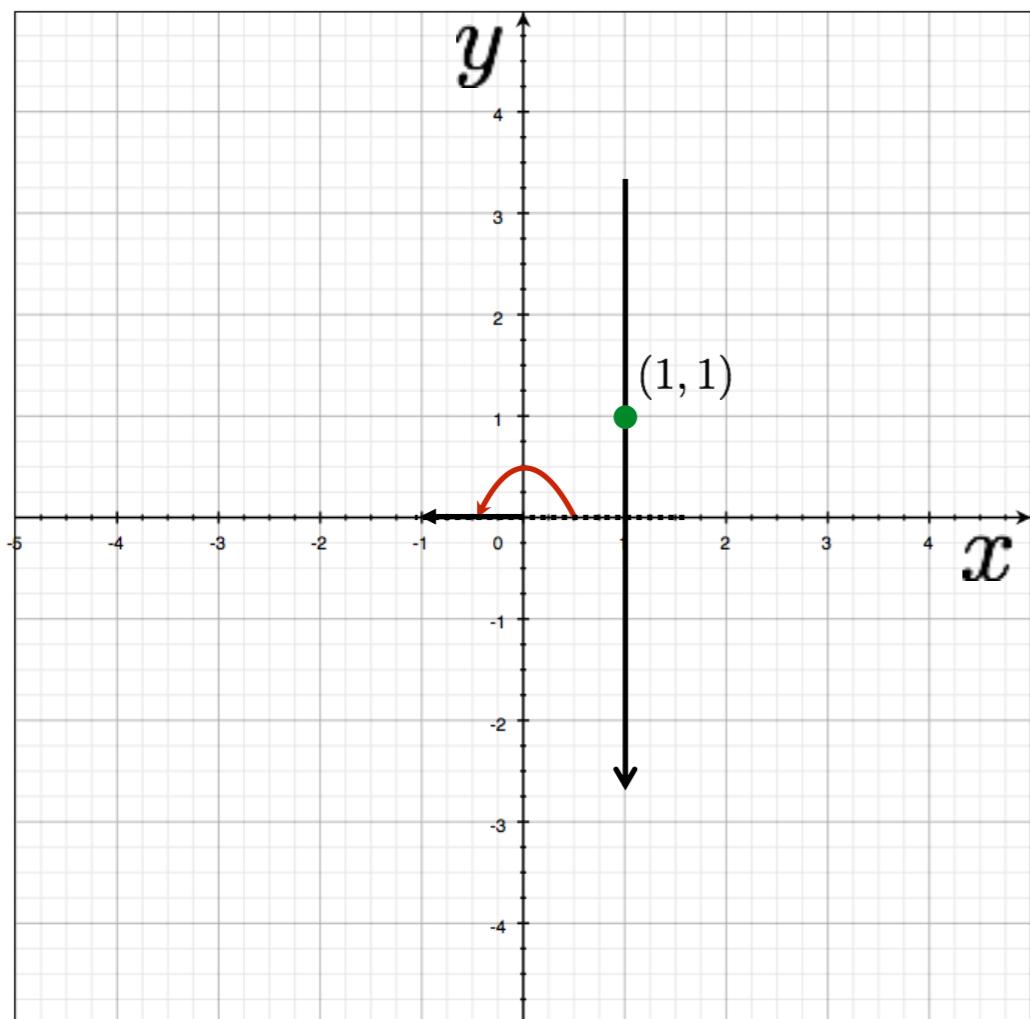


Image space

Parameter space

# Image and parameter space

variables  
 $y = mx + b$   
parameters



a line becomes a point

$$x \cos \theta + y \sin \theta = \rho$$

Wait ...why is rho negative?

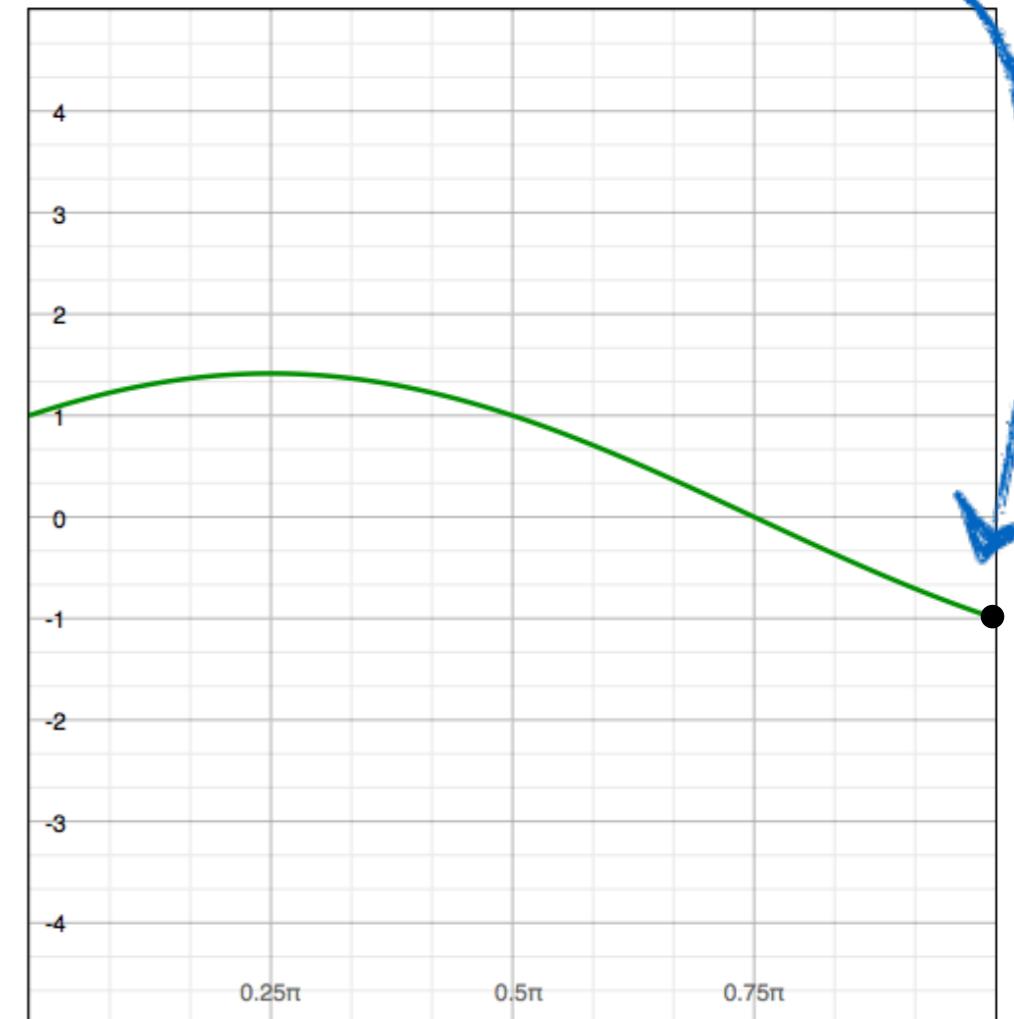
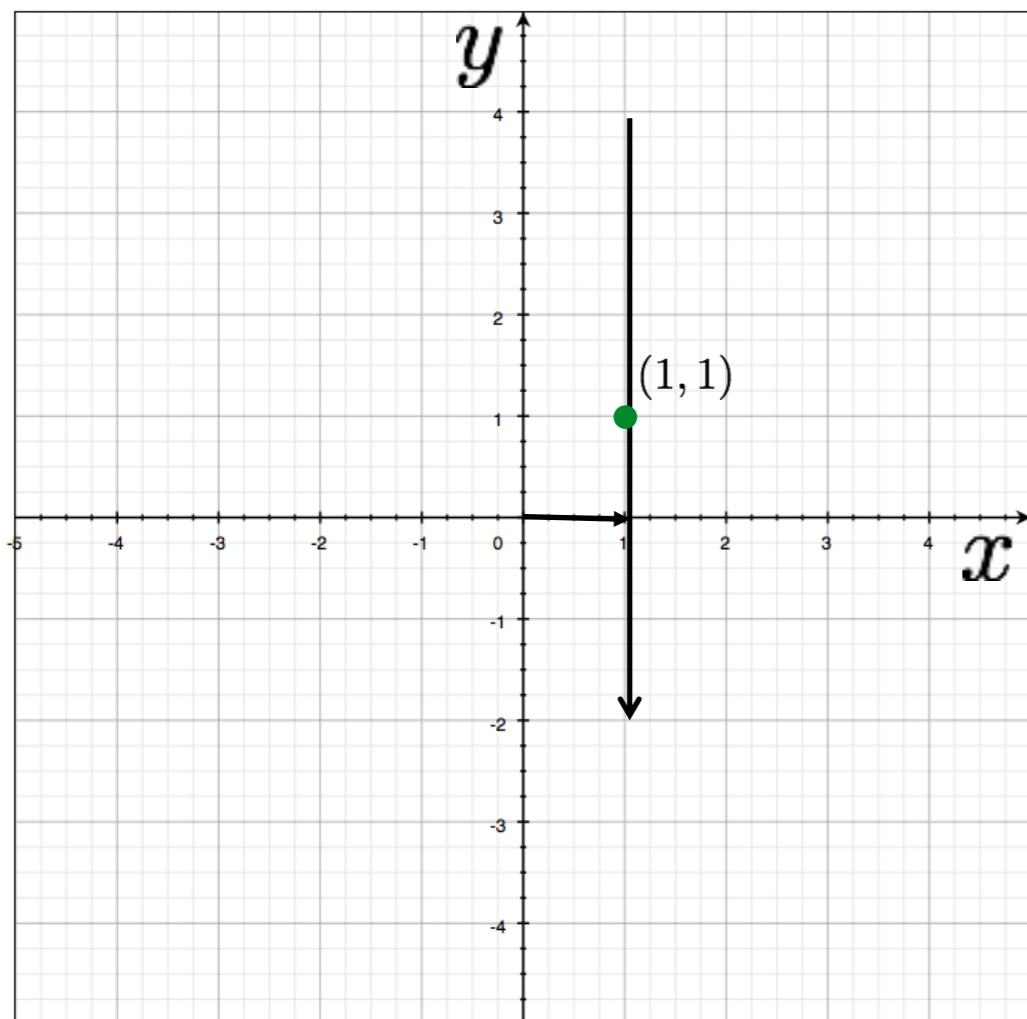


Image space

Parameter space

# Image and parameter space

variables  
 $y = mx + b$   
parameters



a line  
becomes a  
point

$$x \cos \theta + y \sin \theta = \rho$$

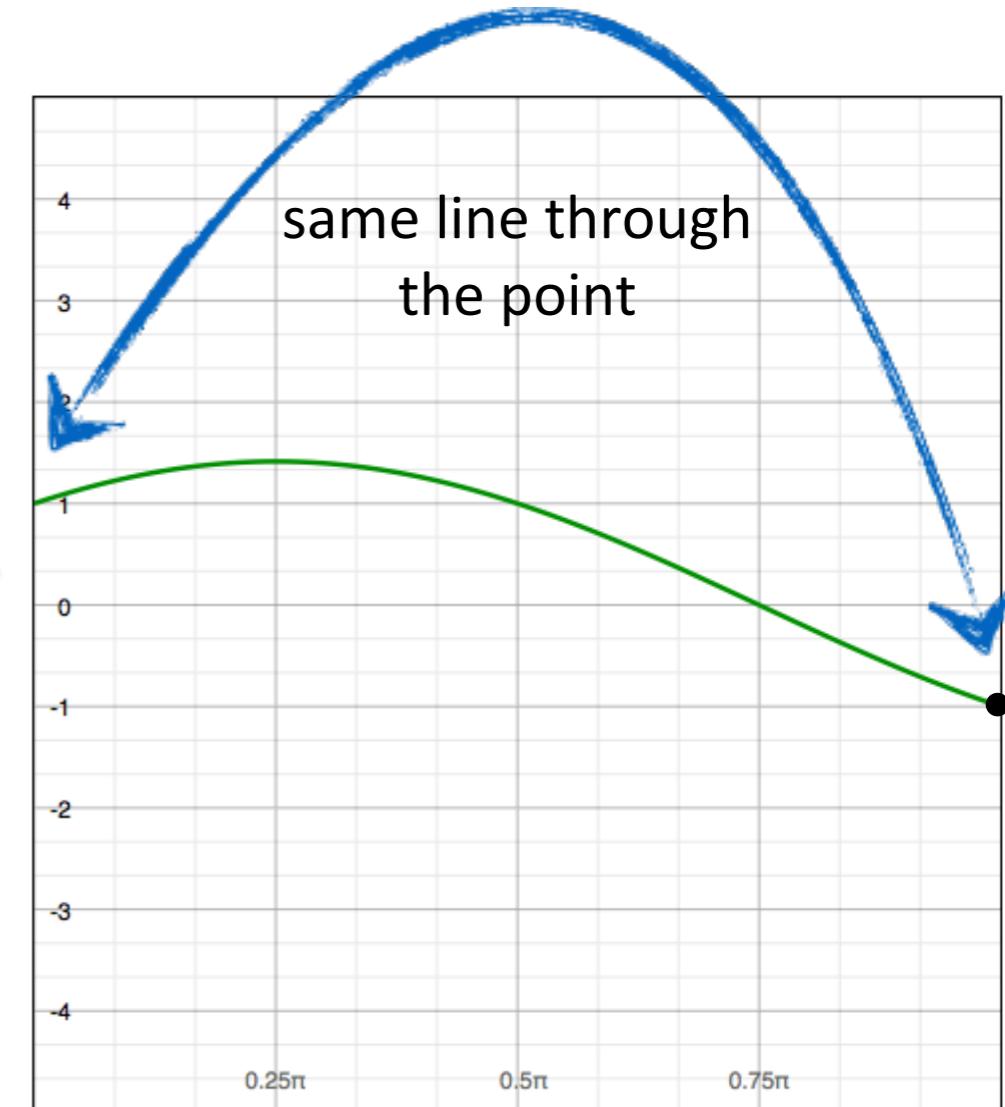


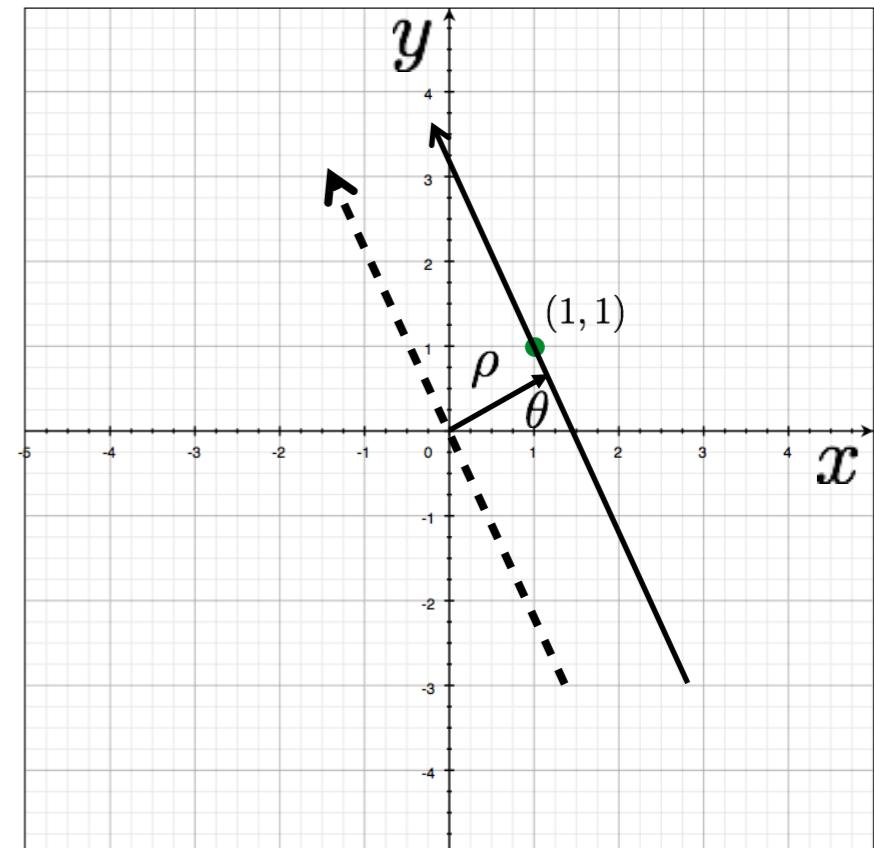
Image space

Parameter space

There are two ways to write the same line:

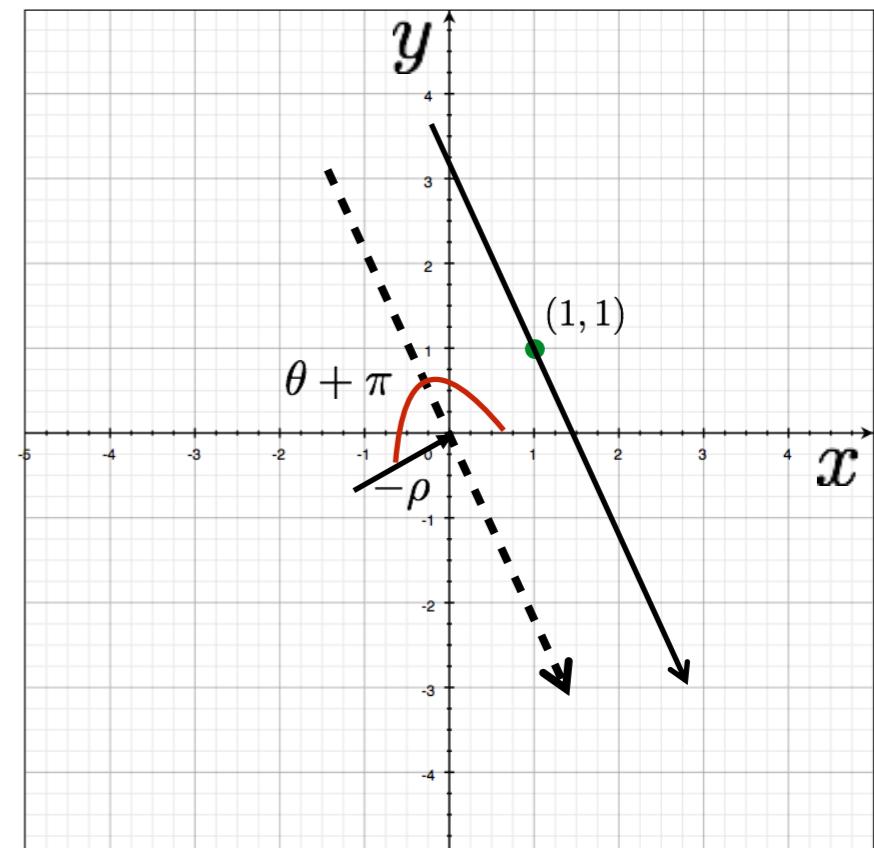
Positive rho version:

$$x \cos \theta + y \sin \theta = \rho$$



Negative rho version:

$$x \cos(\theta + \pi) + y \sin(\theta + \pi) = -\rho$$



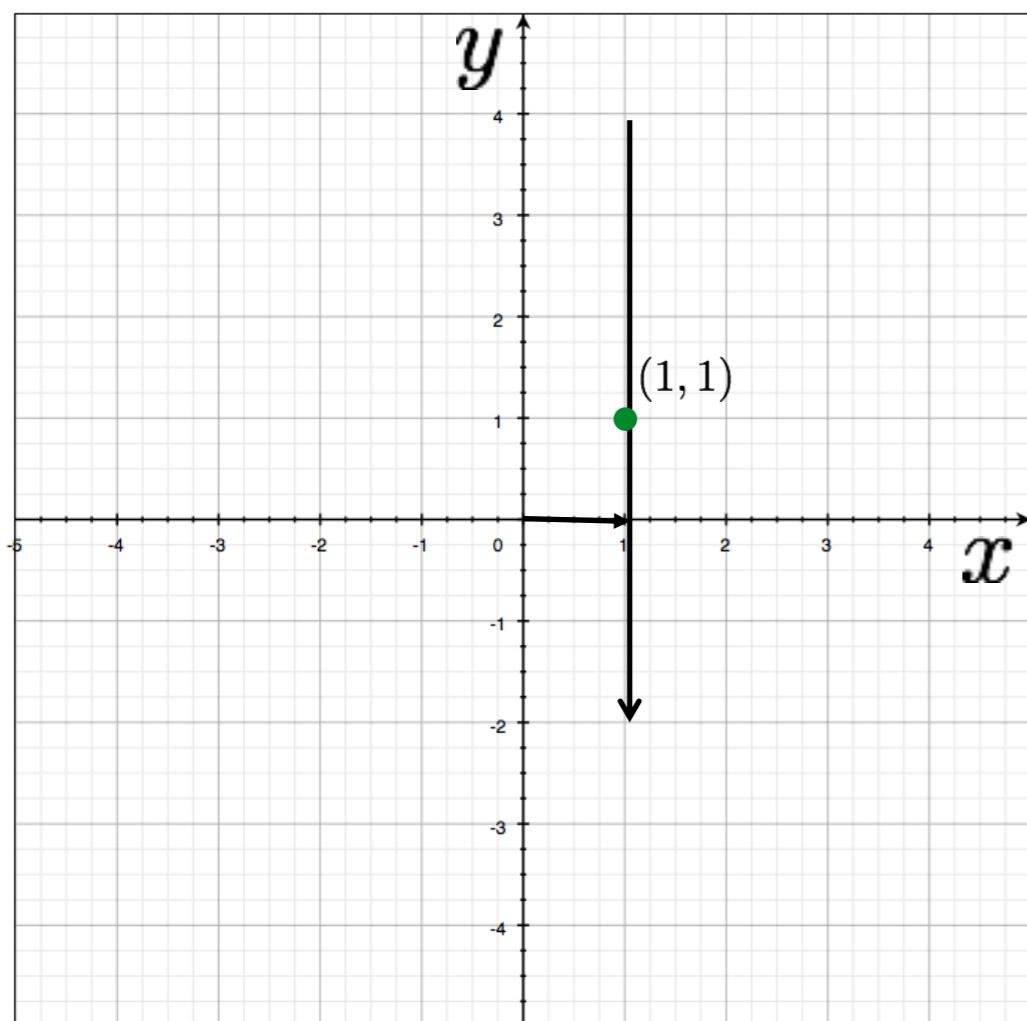
Recall:

$$\sin(\theta) = -\sin(\theta + \pi)$$

$$\cos(\theta) = -\cos(\theta + \pi)$$

# Image and parameter space

variables  
 $y = mx + b$   
parameters



a line  
becomes a  
point

$$x \cos \theta + y \sin \theta = \rho$$

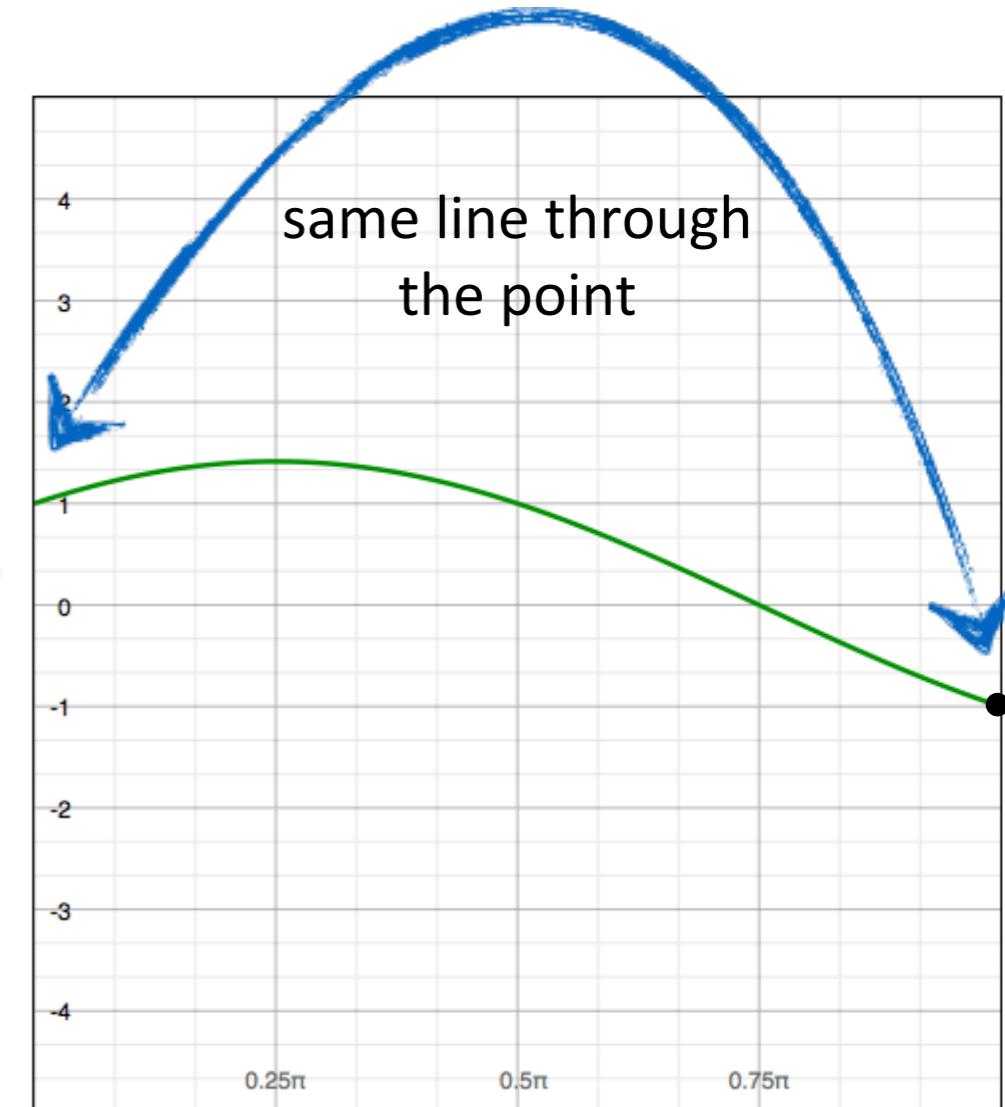
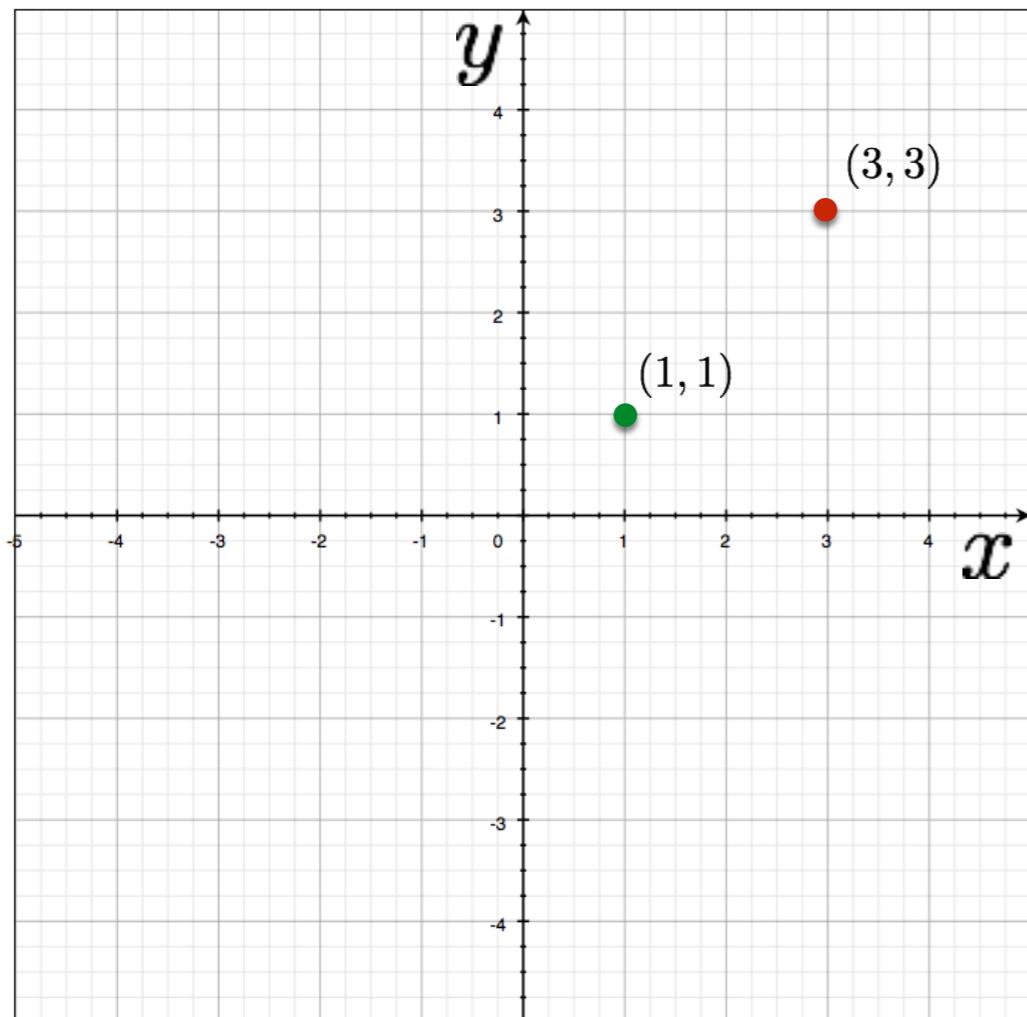


Image space

Parameter space

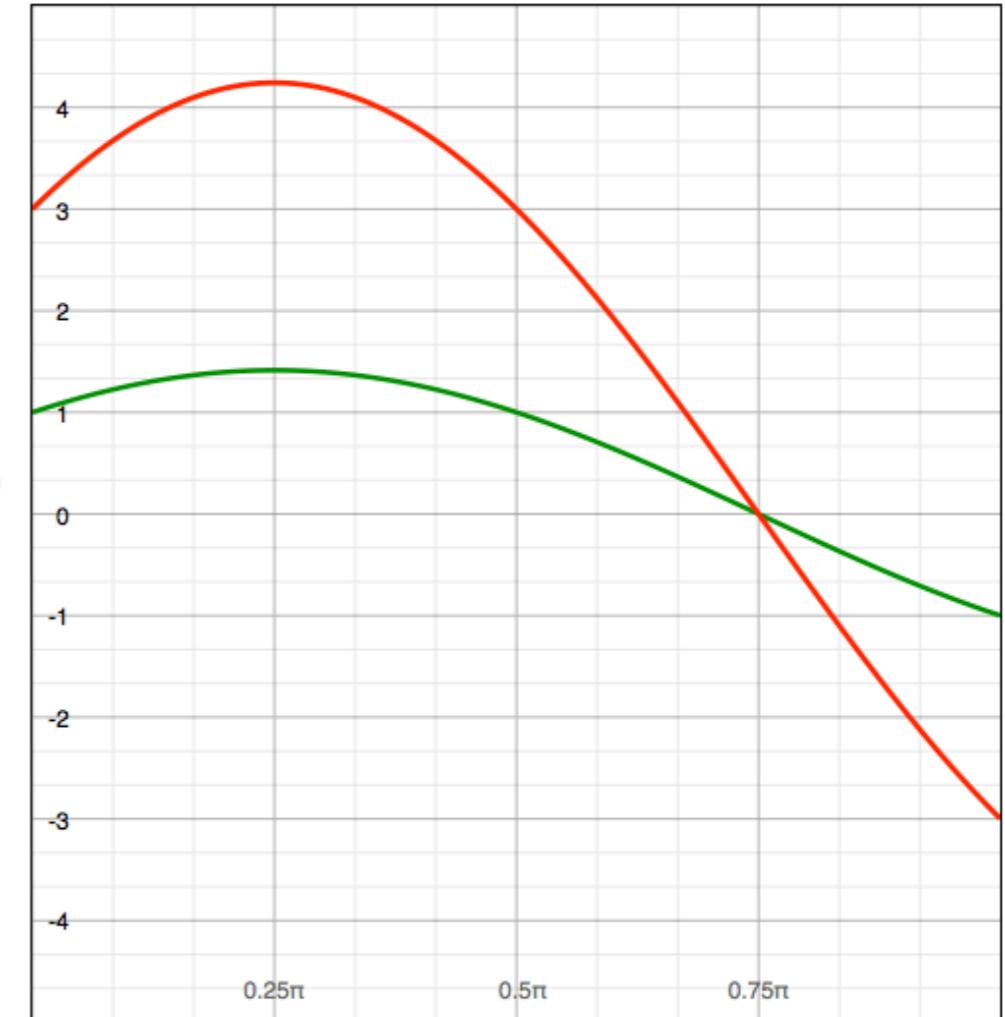
# Image and parameter space

variables  
 $y = mx + b$   
parameters



two points  
become  
?

Image space



Parameter space

# Image and parameter space

variables  
 $y = mx + b$   
parameters

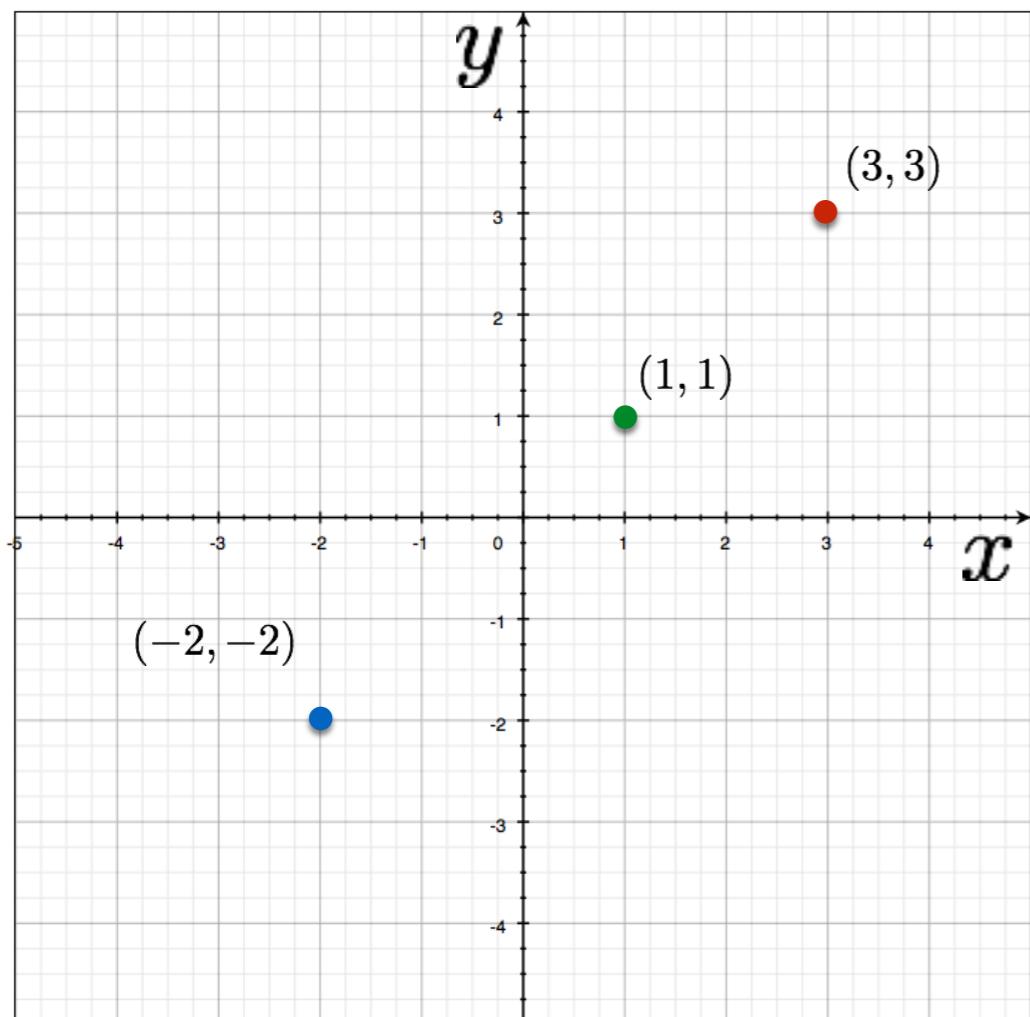
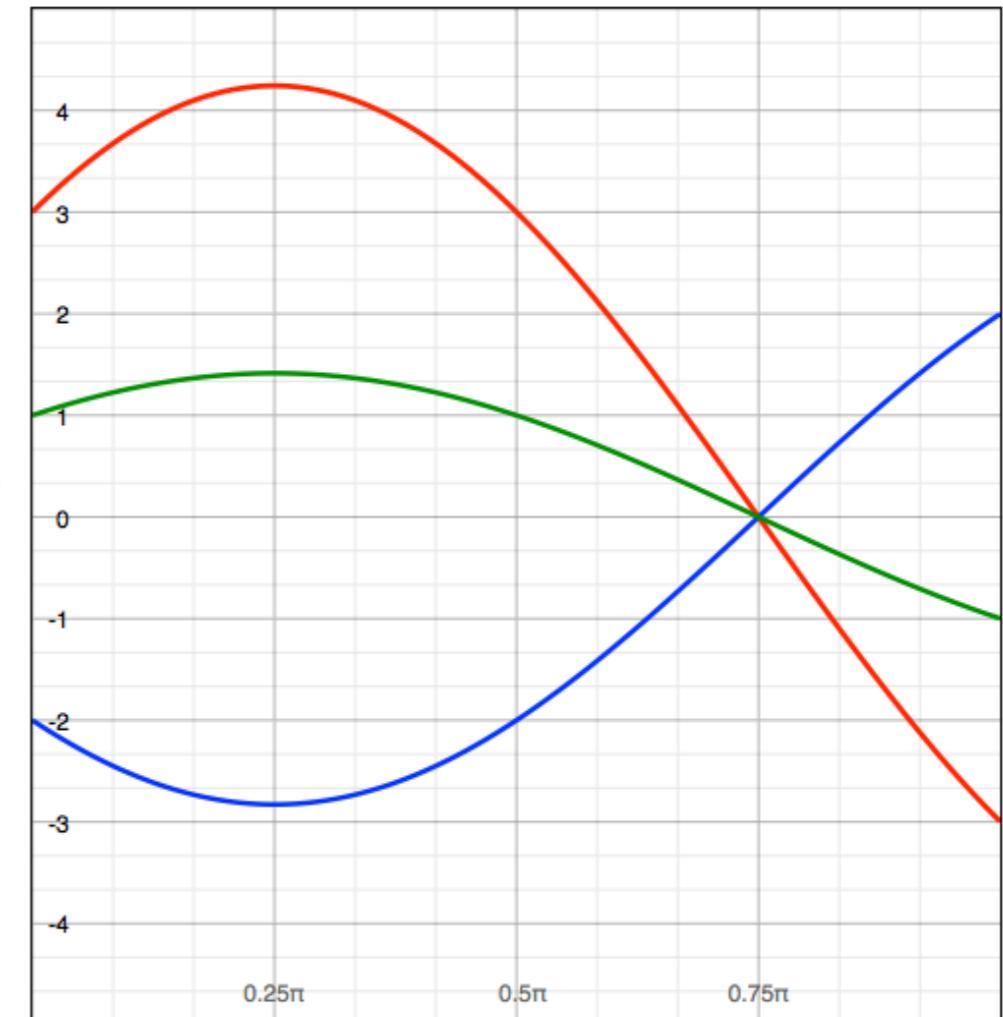


Image space

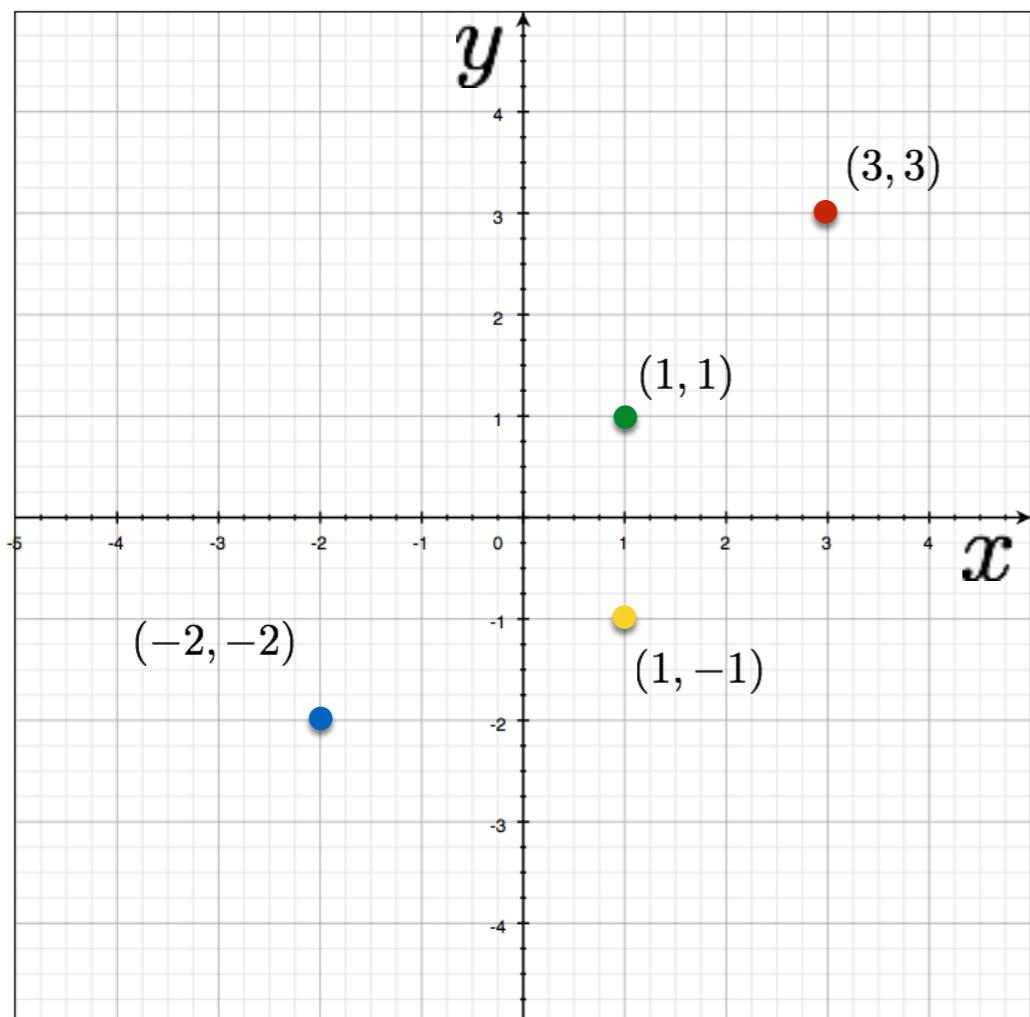
three points  
become  
?



Parameter space

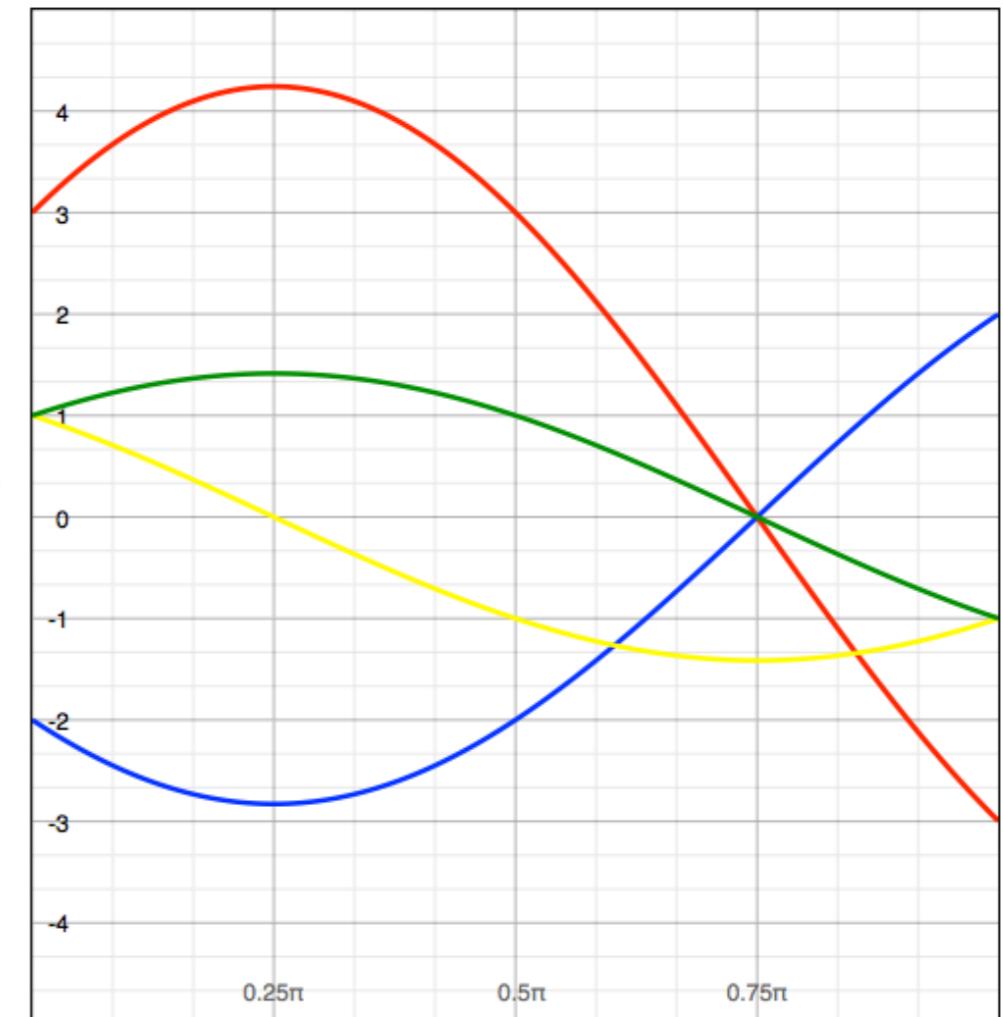
# Image and parameter space

variables  
 $y = mx + b$   
parameters



four points  
become  
?

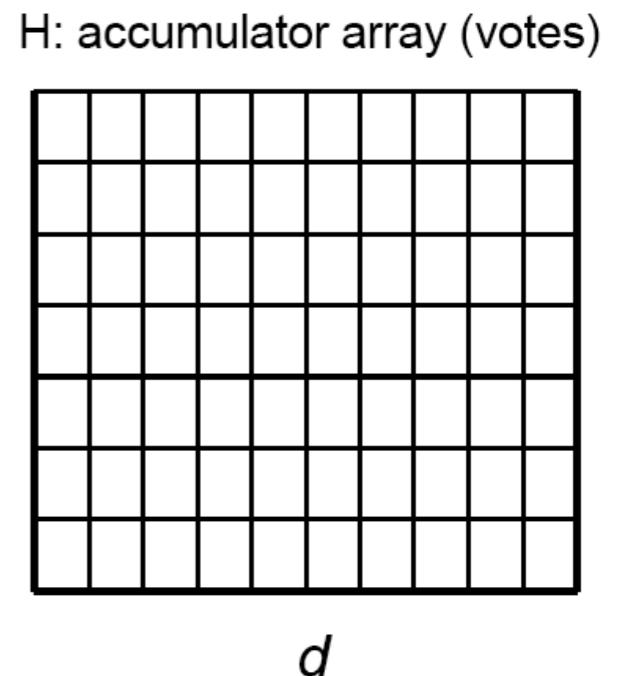
Image space



Parameter space

# Implementation

1. Initialize accumulator  $H$  to all zeros
2. For each edge point  $(x, y)$  in the image  
    For  $\theta = 0$  to  $180$   
         $\rho = x \cos \theta + y \sin \theta$   
         $H(\theta, \rho) = H(\theta, \rho) + 1$   
    end  
end
3. Find the value(s) of  $(\theta, \rho)$  where  $H(\theta, \rho)$  is a local maximum
4. The detected line in the image is given by  
$$\rho = x \cos \theta + y \sin \theta$$



NOTE: Watch your coordinates. Image origin is top left!

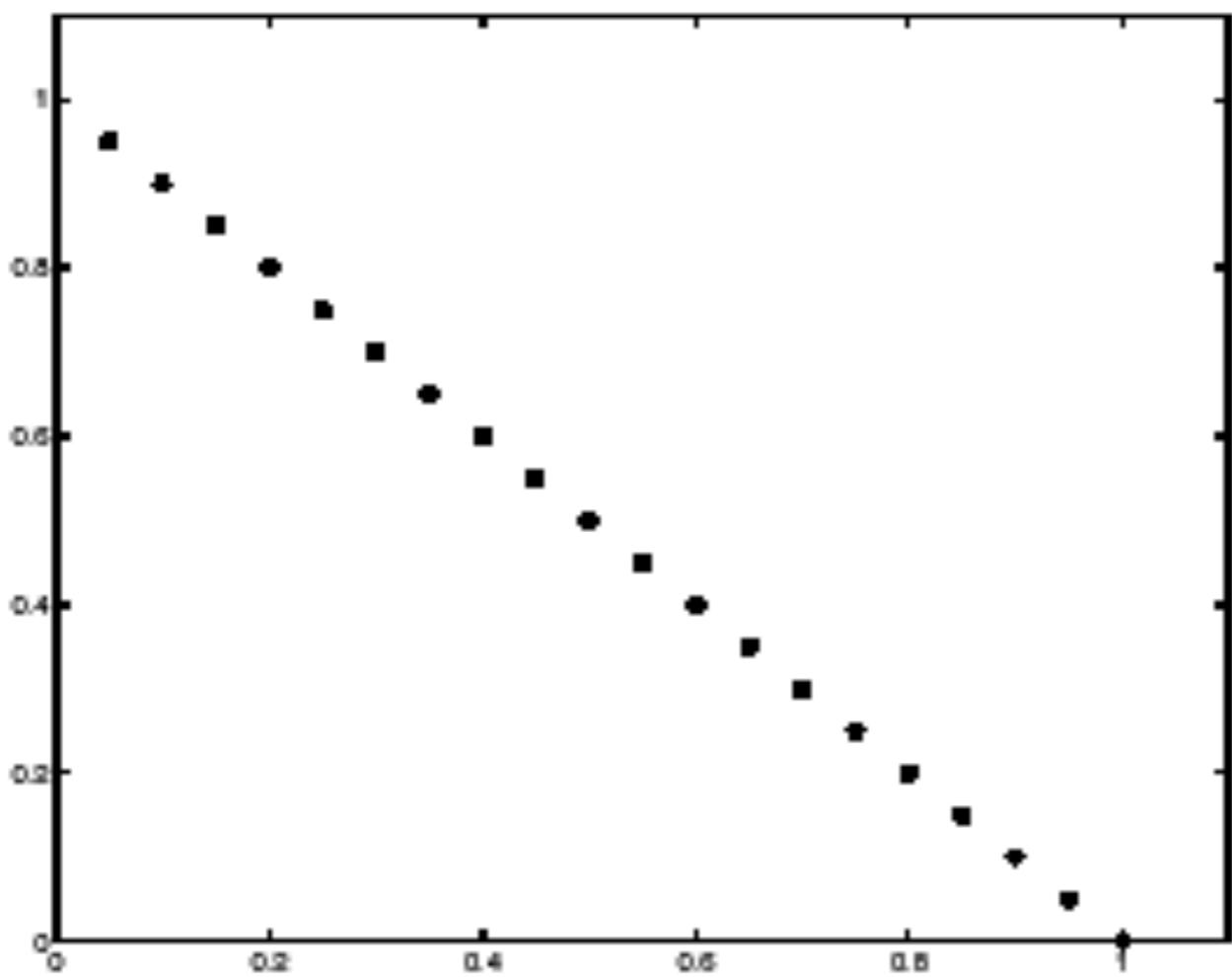
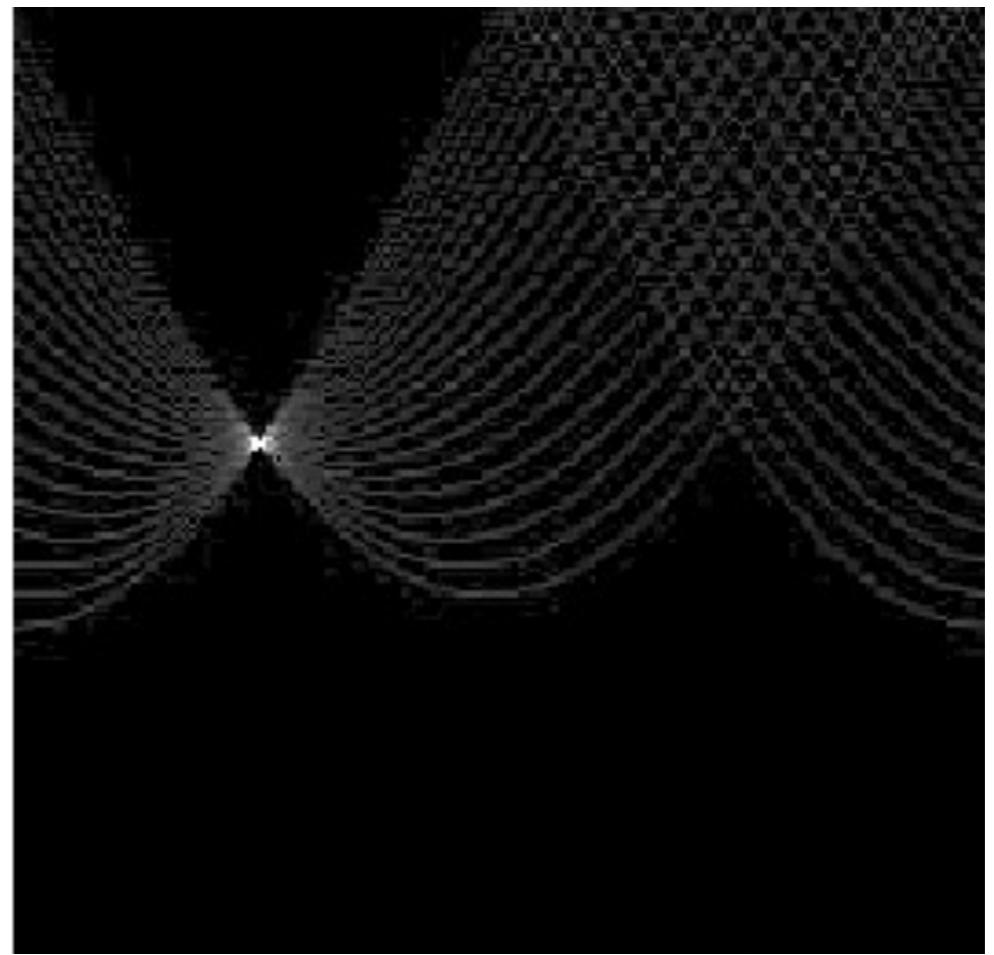


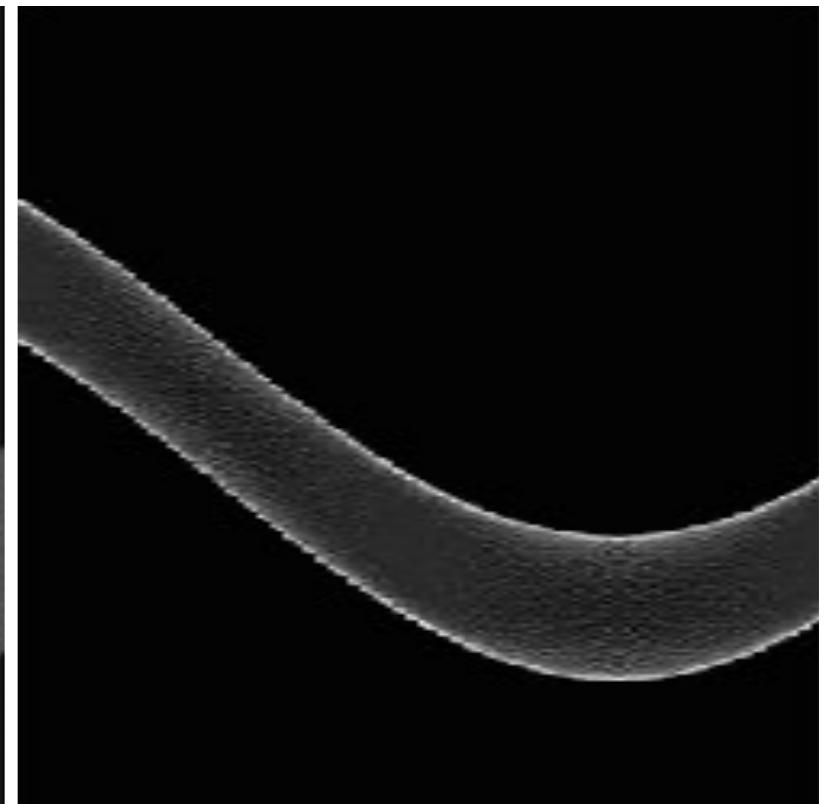
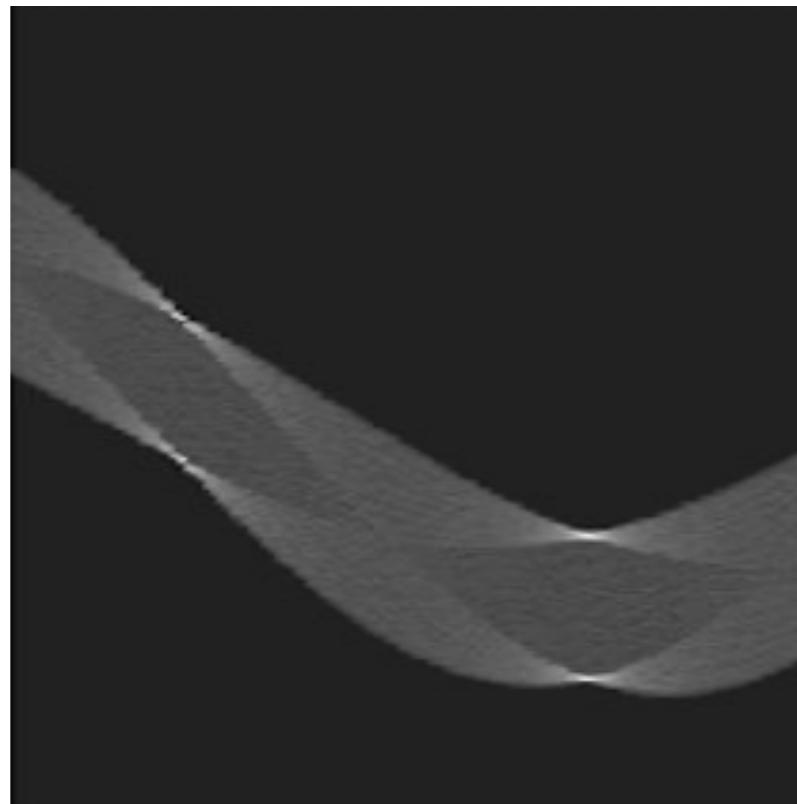
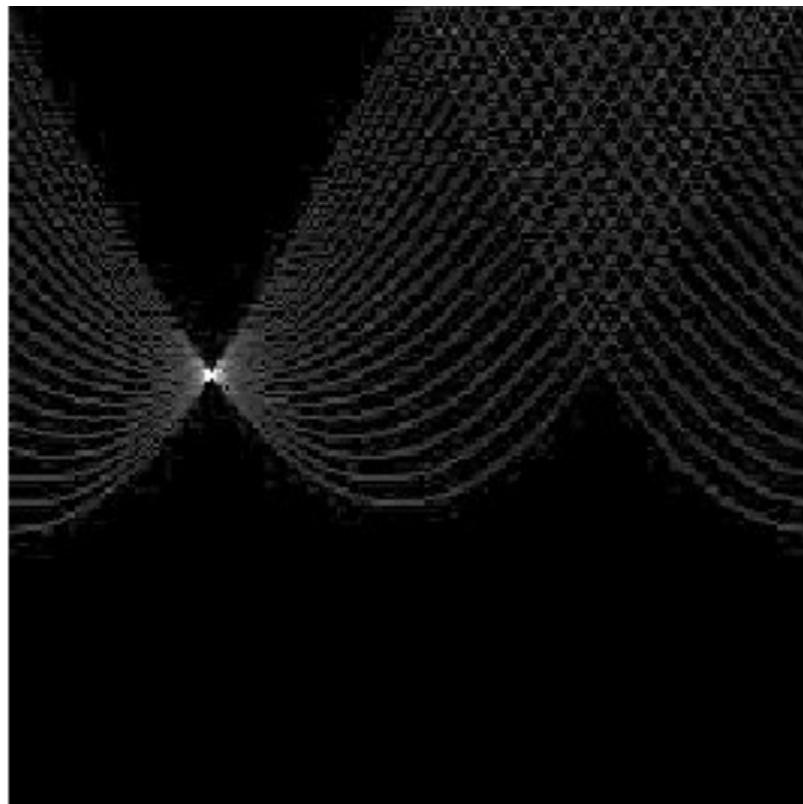
Image space



Votes

# Basic shapes

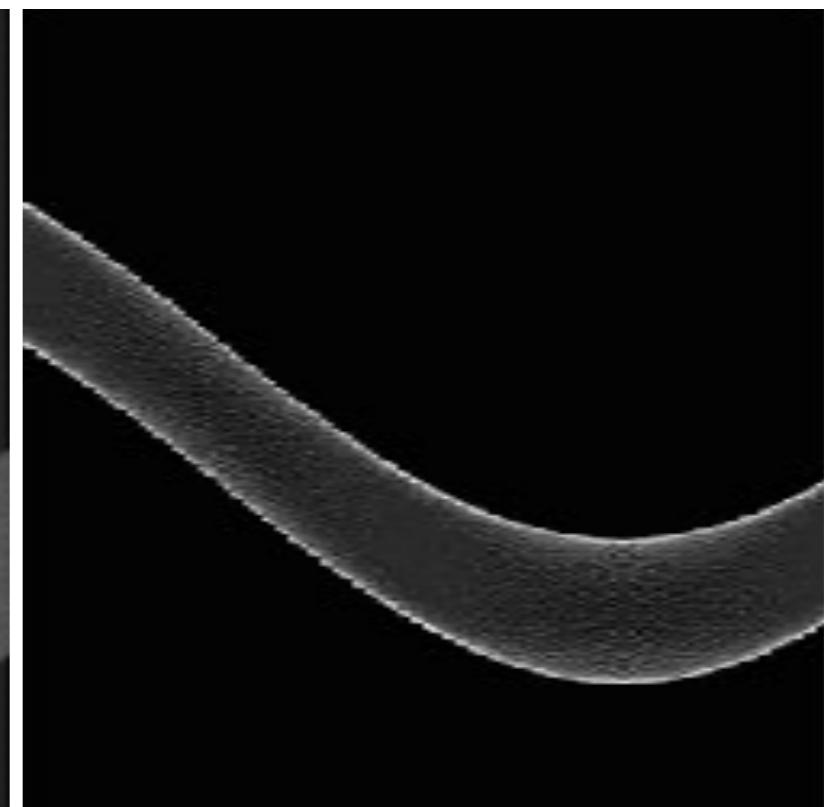
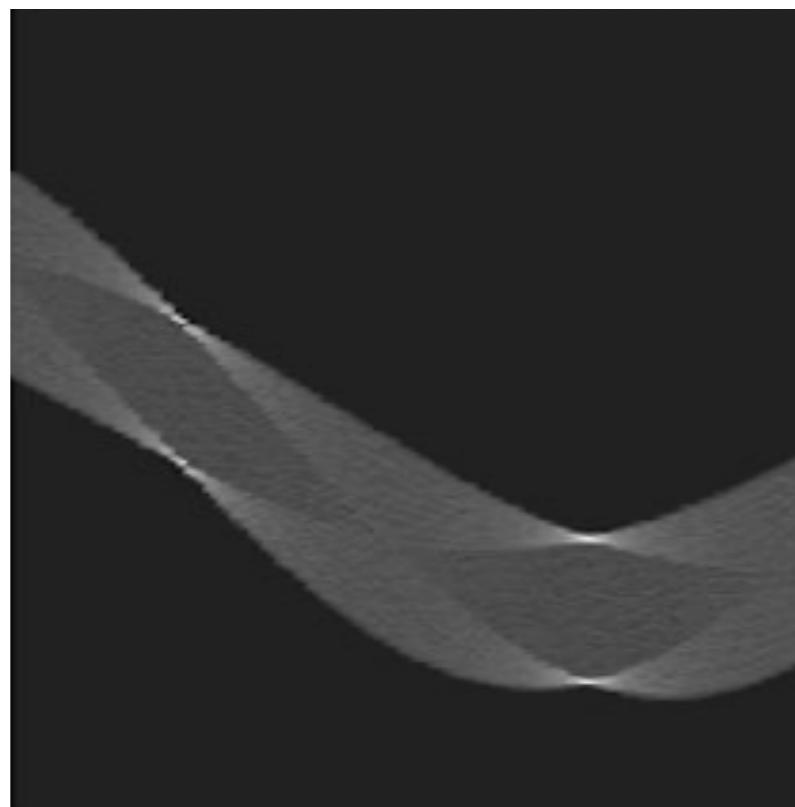
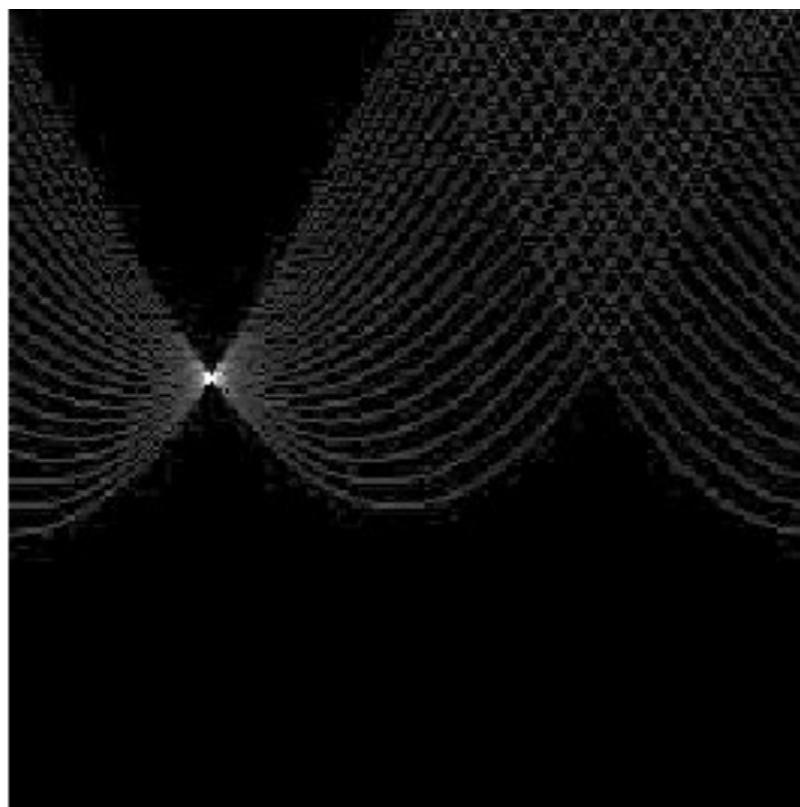
(in parameter space)



*can you guess the shape?*

# Basic shapes

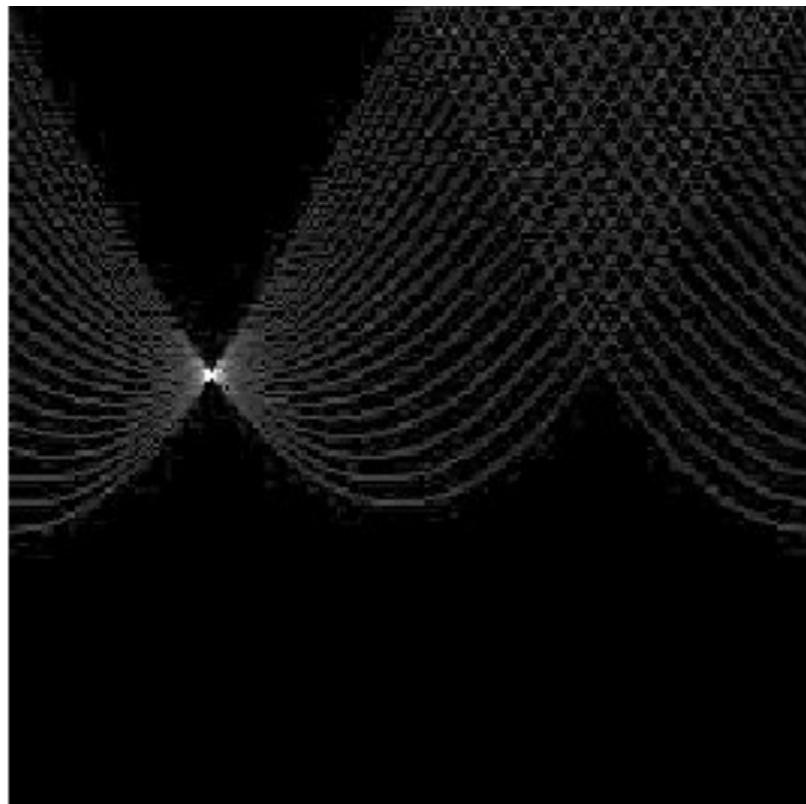
(in parameter space)



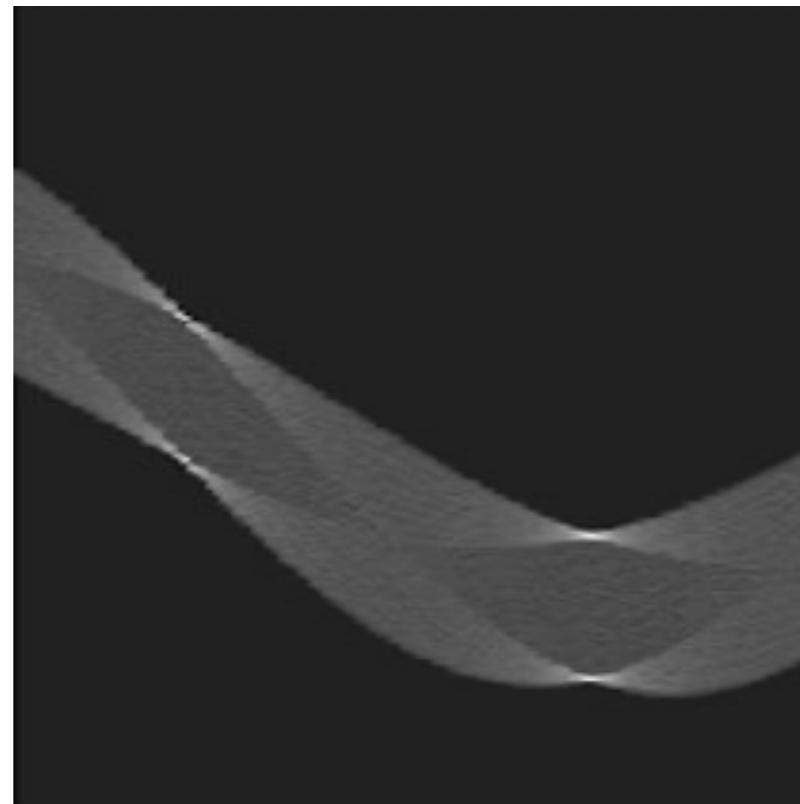
line

# Basic shapes

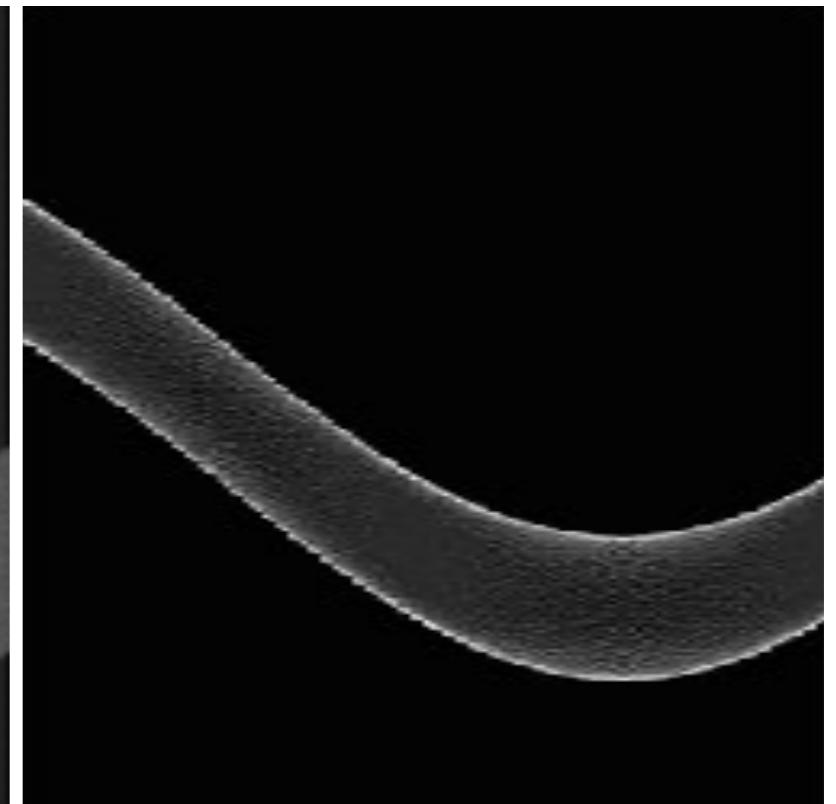
(in parameter space)



line

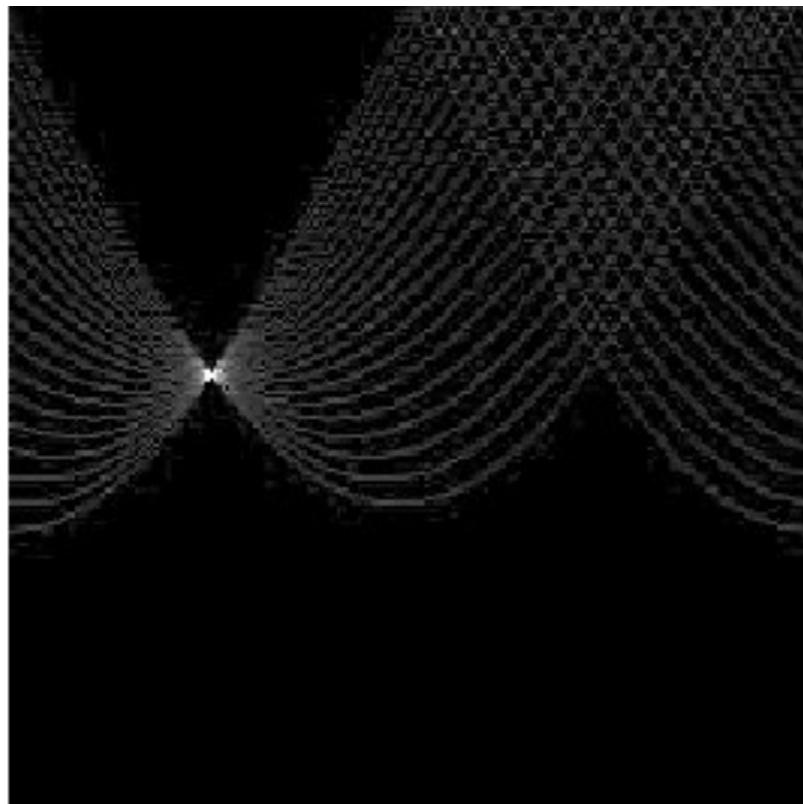


rectangle

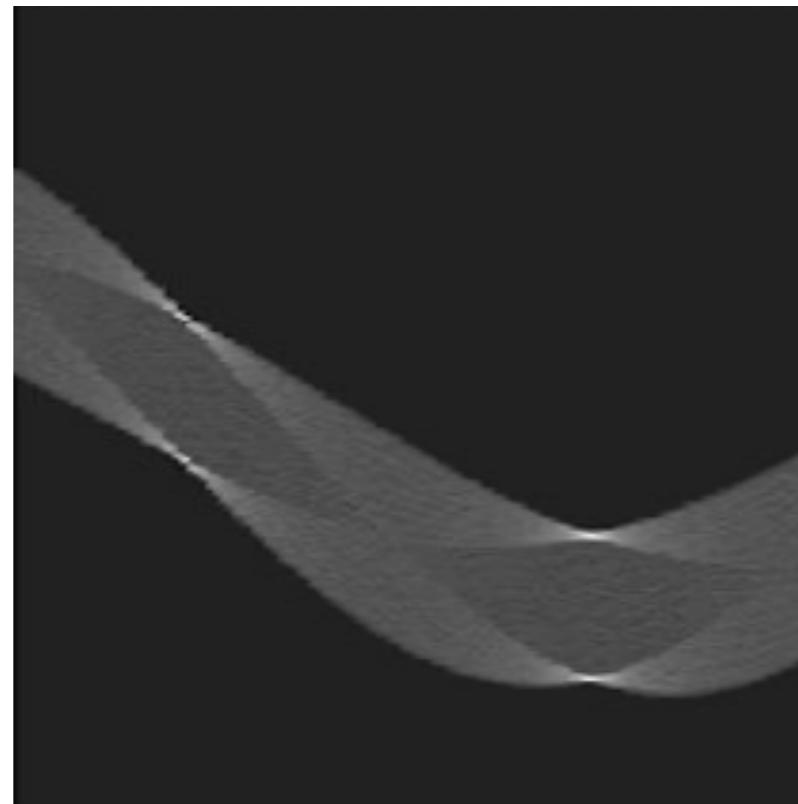


# Basic shapes

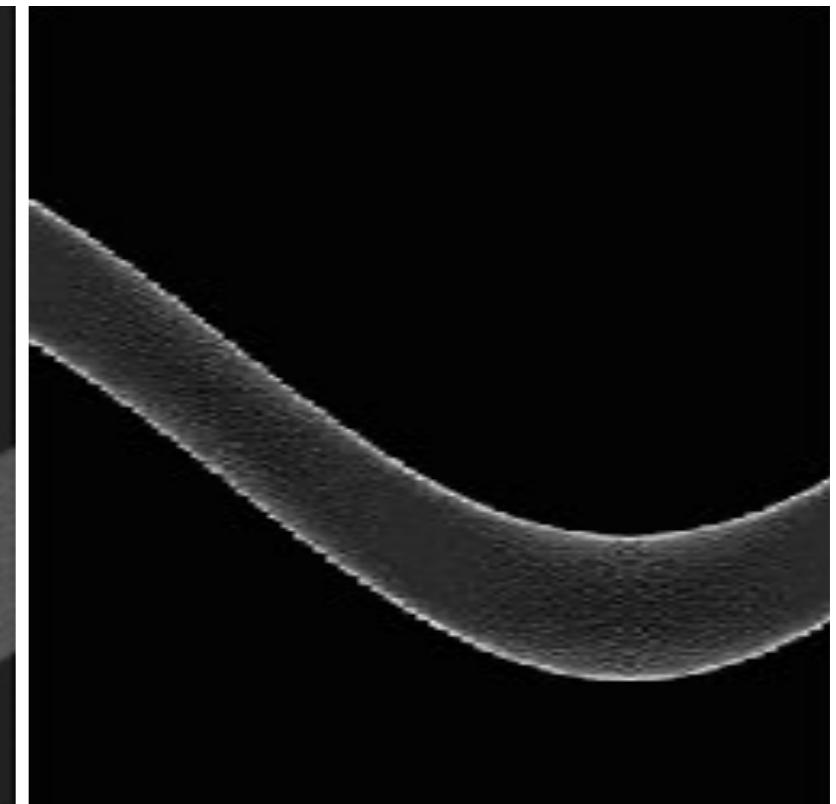
(in parameter space)



line

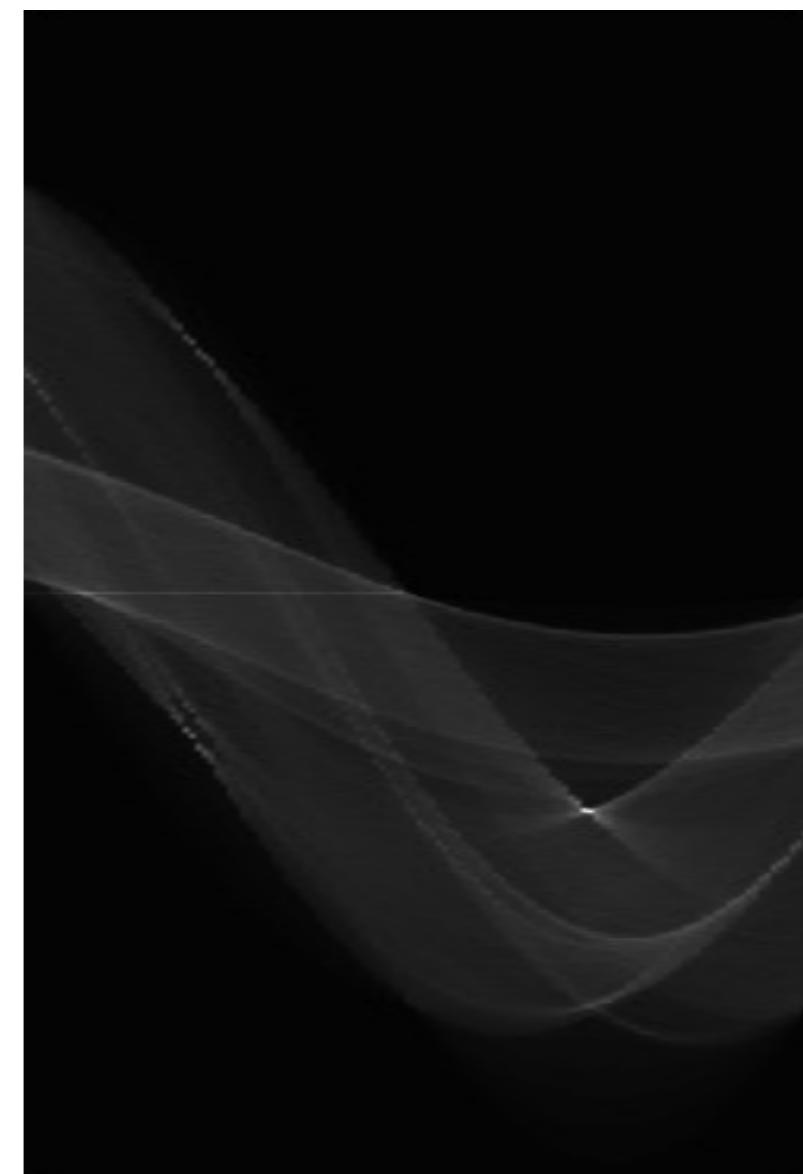
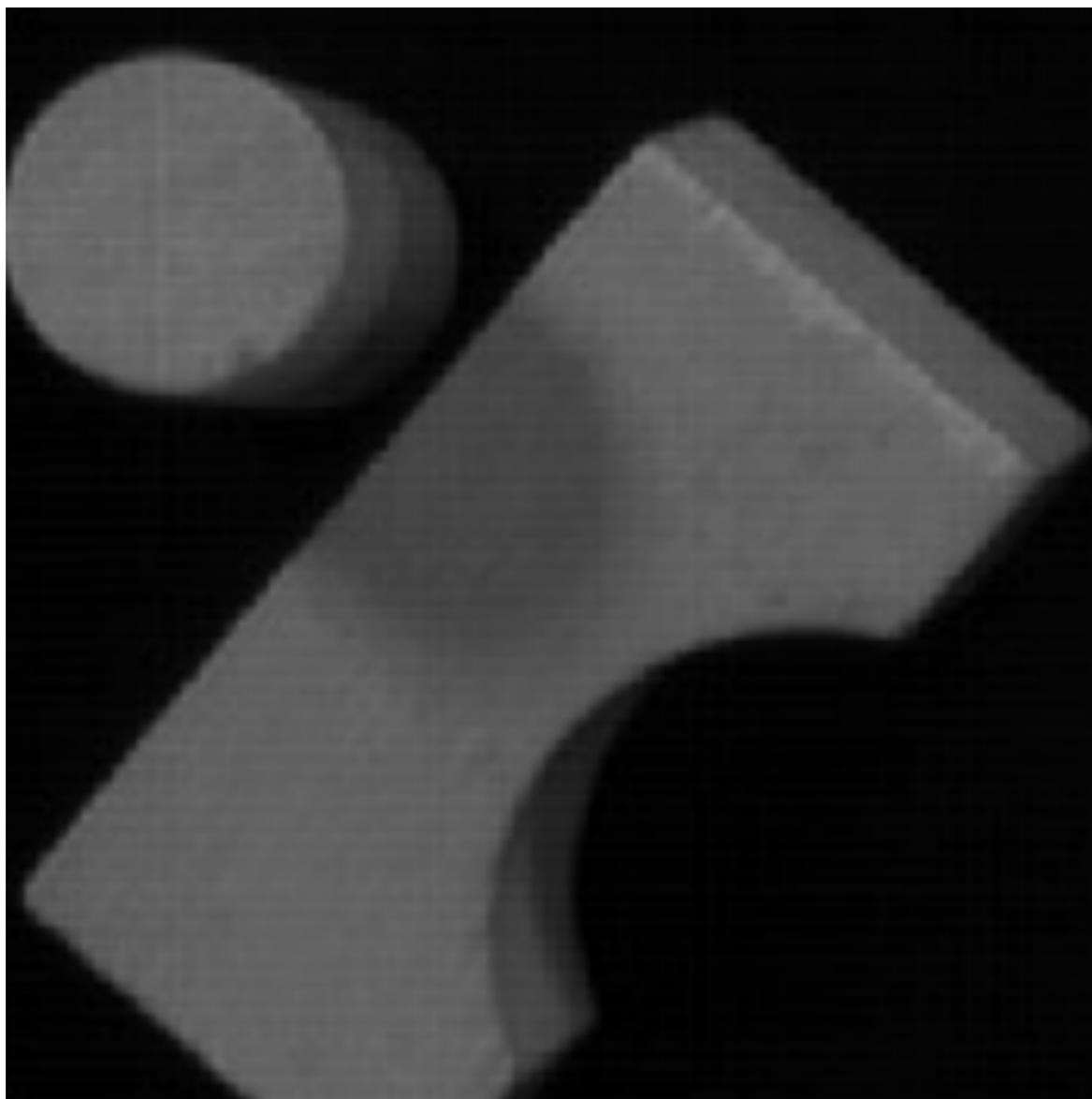


rectangle

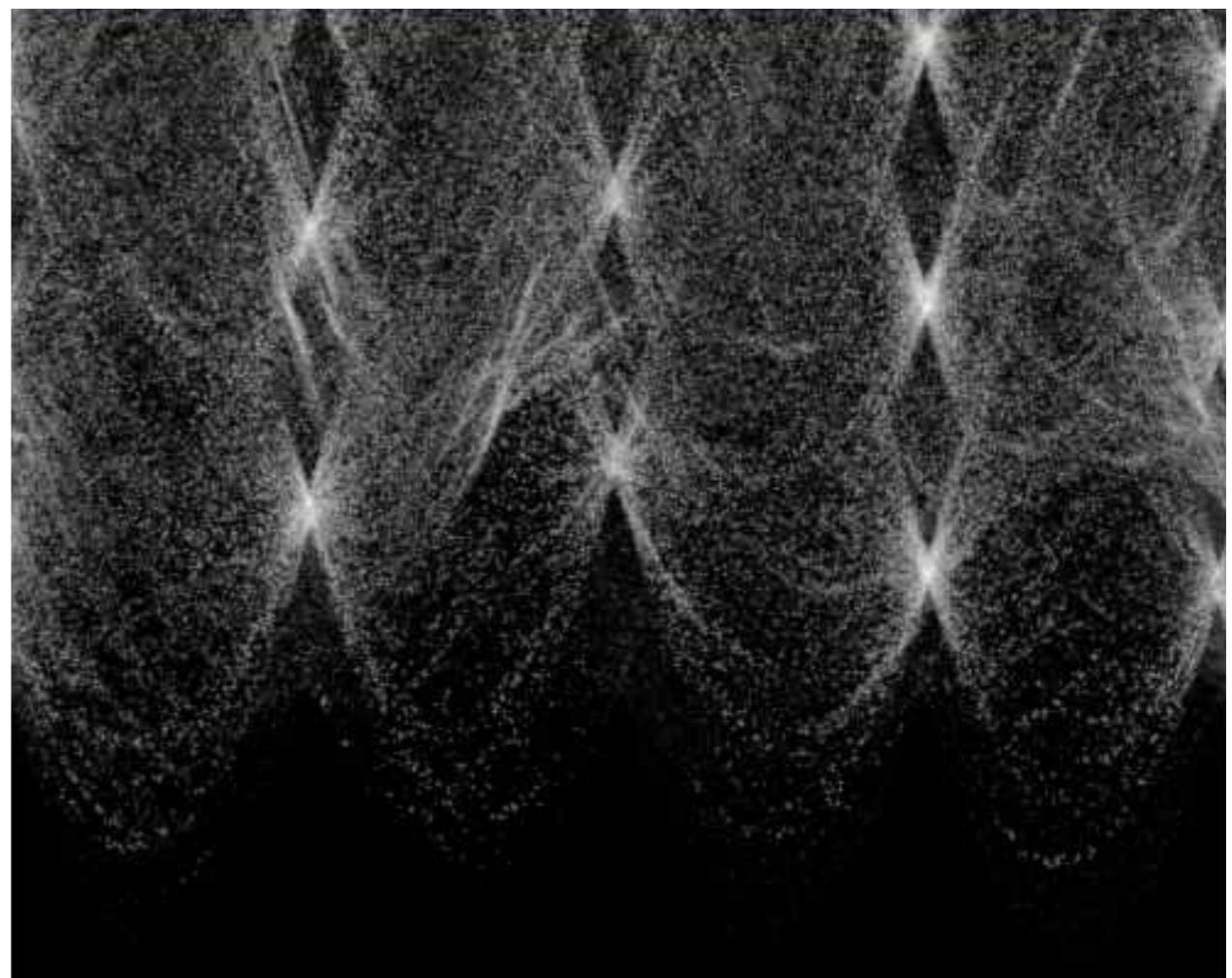


circle

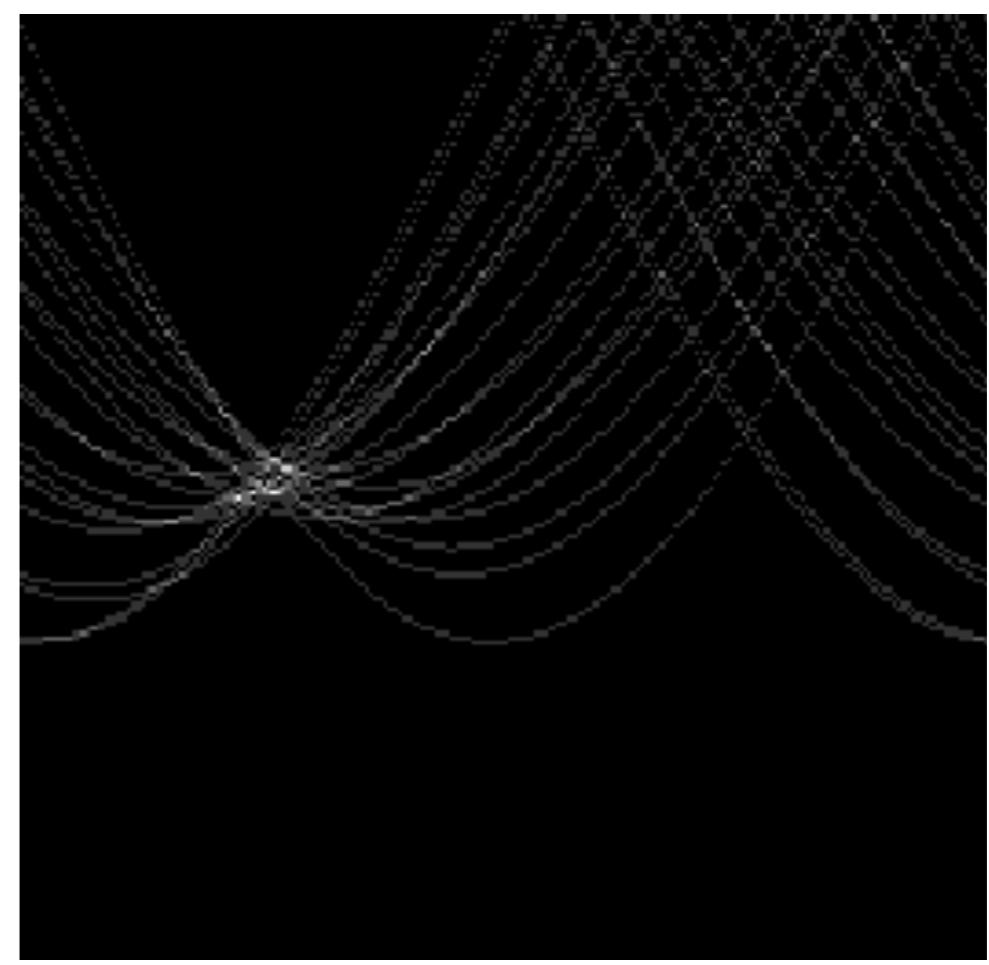
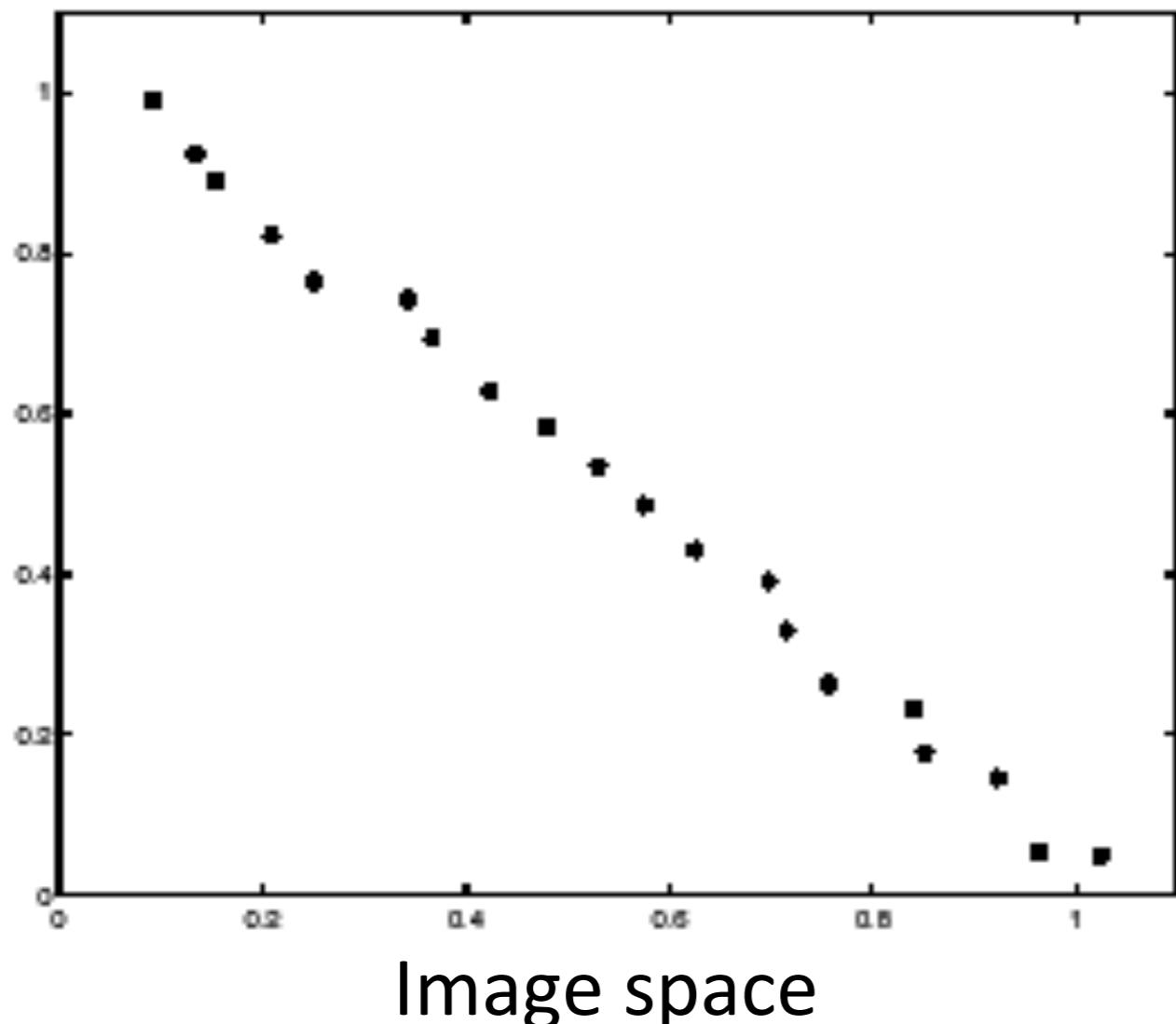
# Basic Shapes



# More complex image



In practice, measurements are noisy...



Votes

Too much noise ...

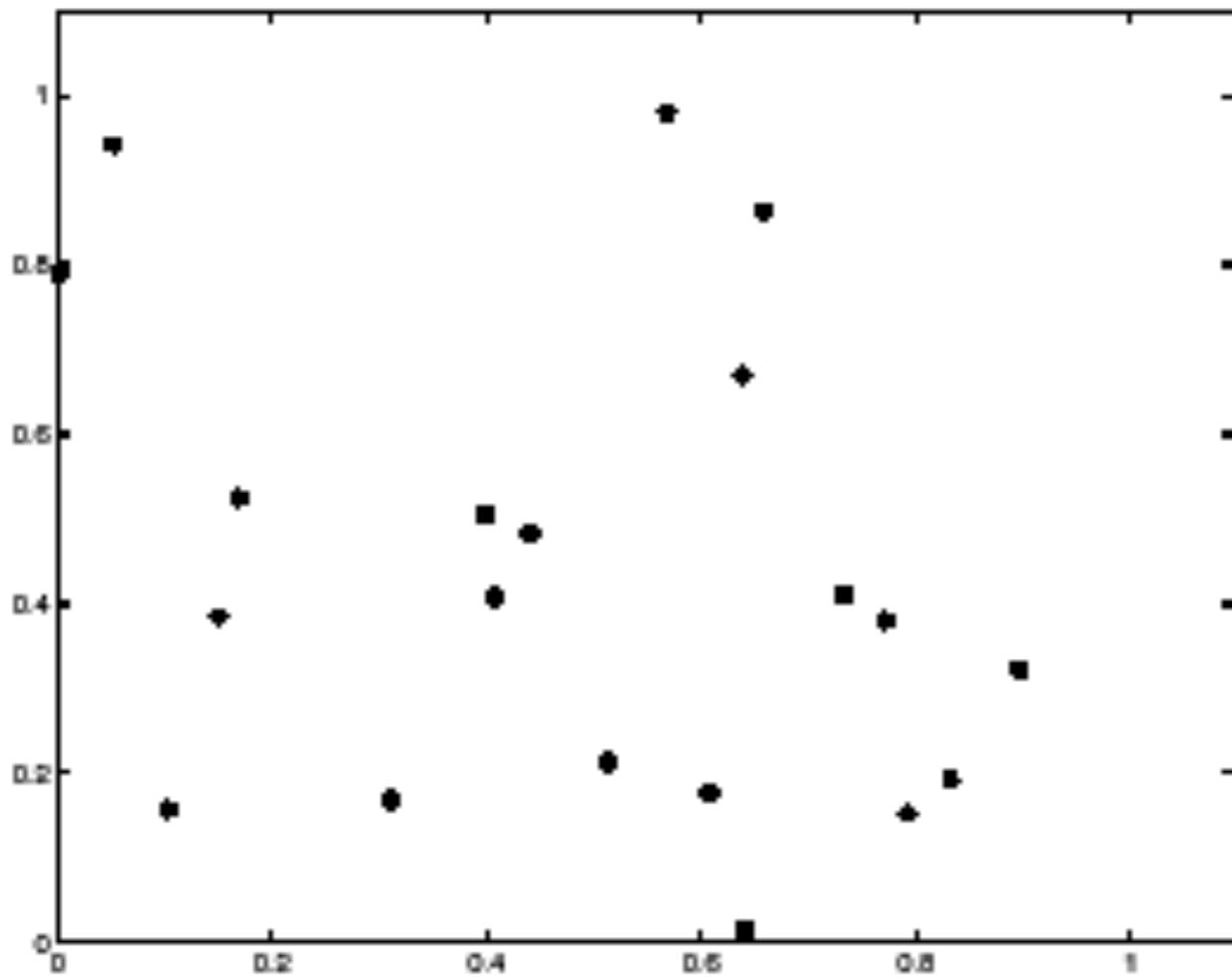
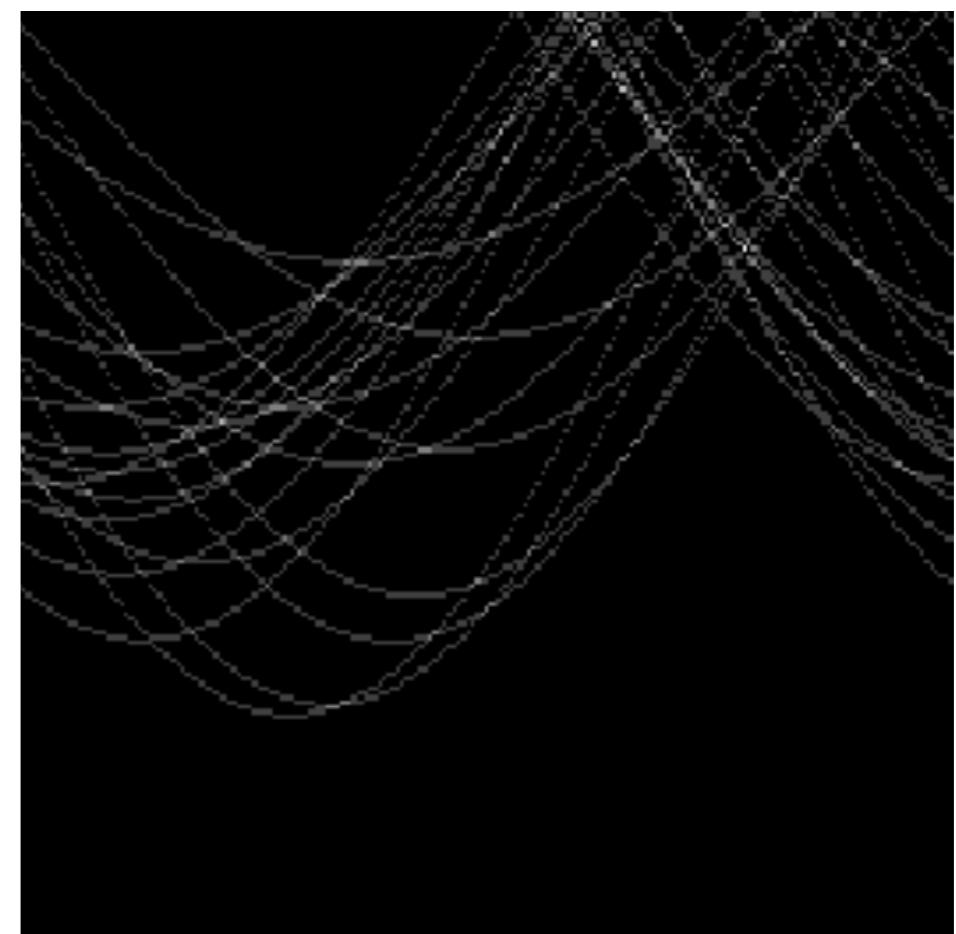


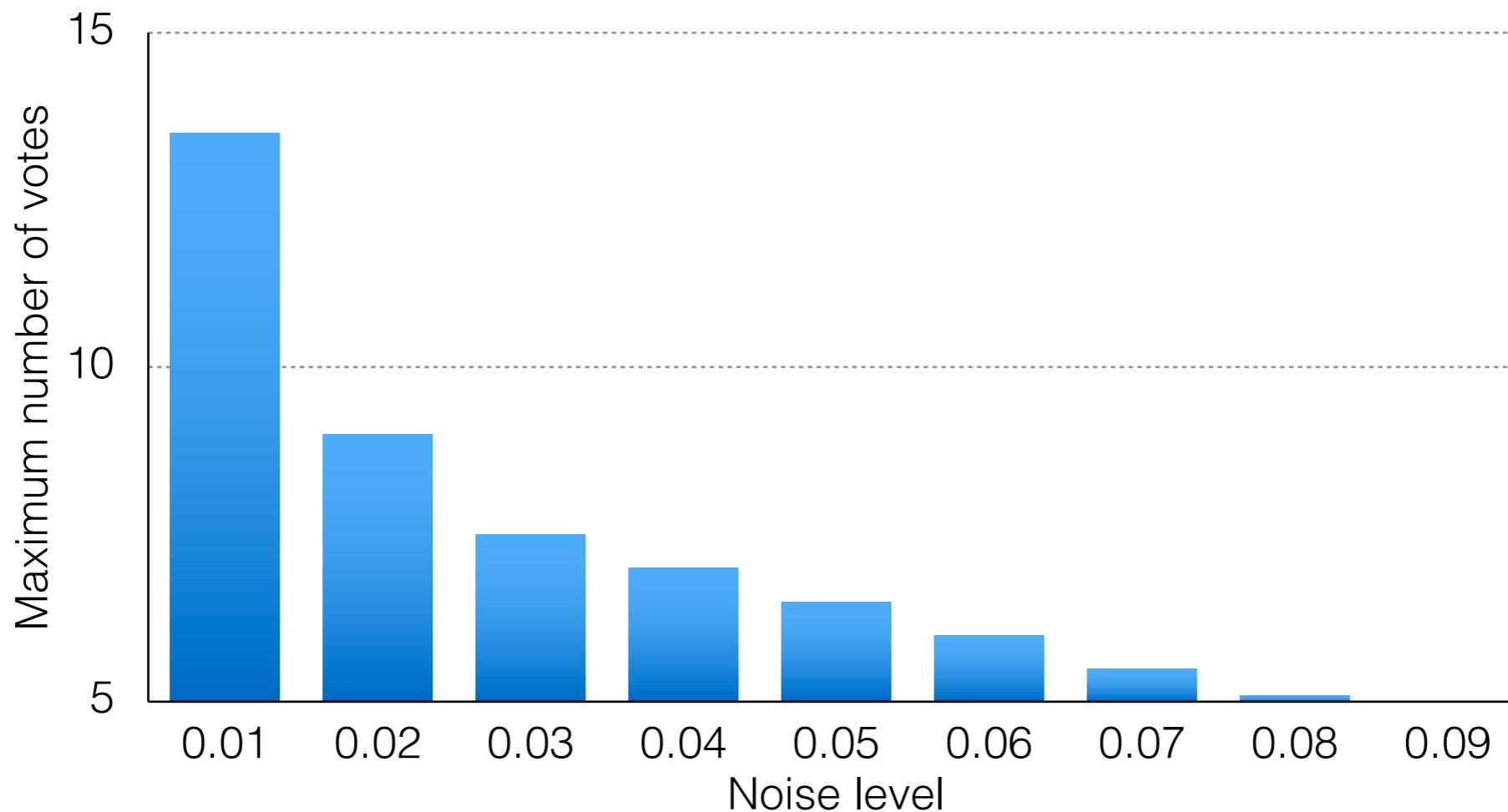
Image space



Votes

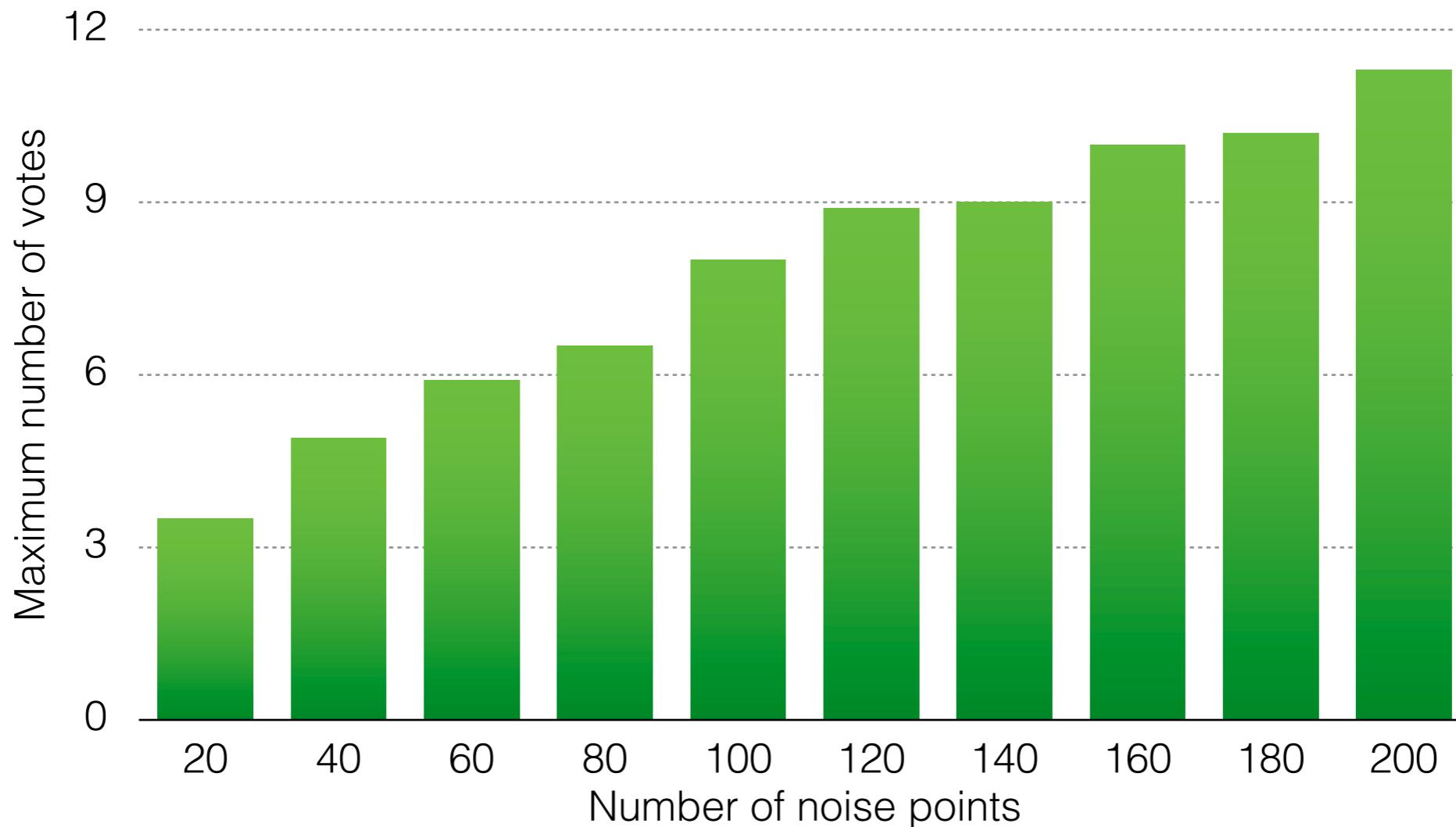
# Effects of noise level

Number of votes for a line of 20 points with increasing noise



More noise, fewer votes (in the right bin)

# Effect of noise points

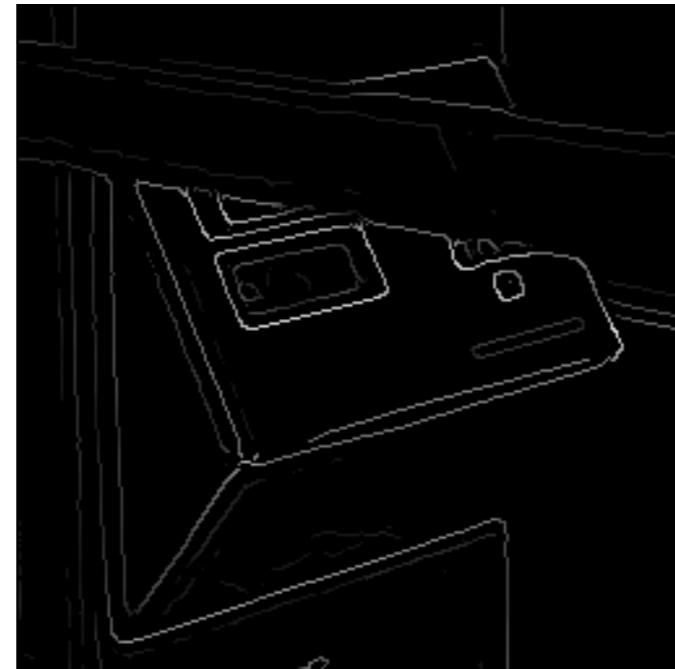


More noise, more votes (in the wrong bin)

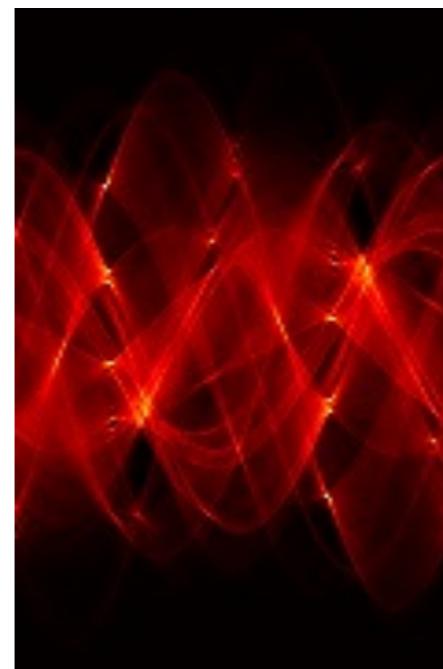
# Real-world example



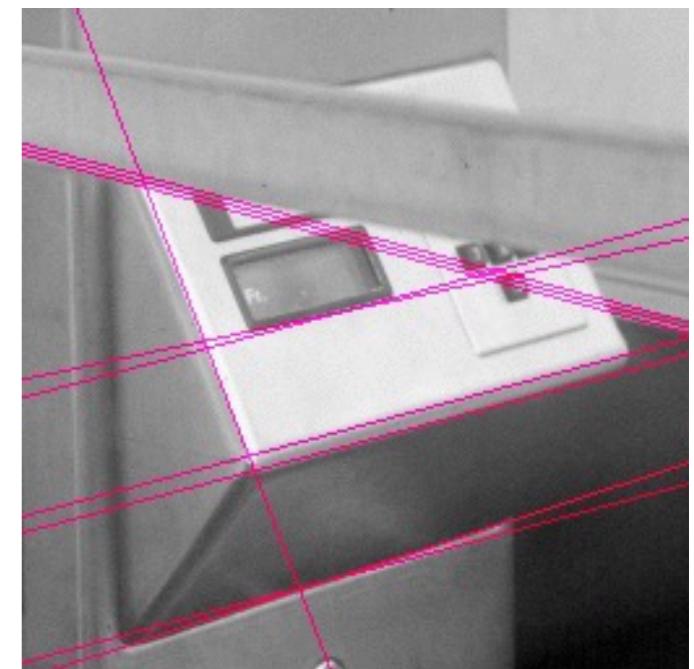
Original



Edges



parameter space



Hough Lines

# Hough Circles

# Let's assume radius known

$$(x - a)^2 + (y - b)^2 = r^2$$

parameters  
variables

$$(x - a)^2 + (y - b)^2 = r^2$$

parameters  
variables

*What is the dimension of the parameter space?*

parameters  
 $(x - a)^2 + (y - b)^2 = r^2$   
variables

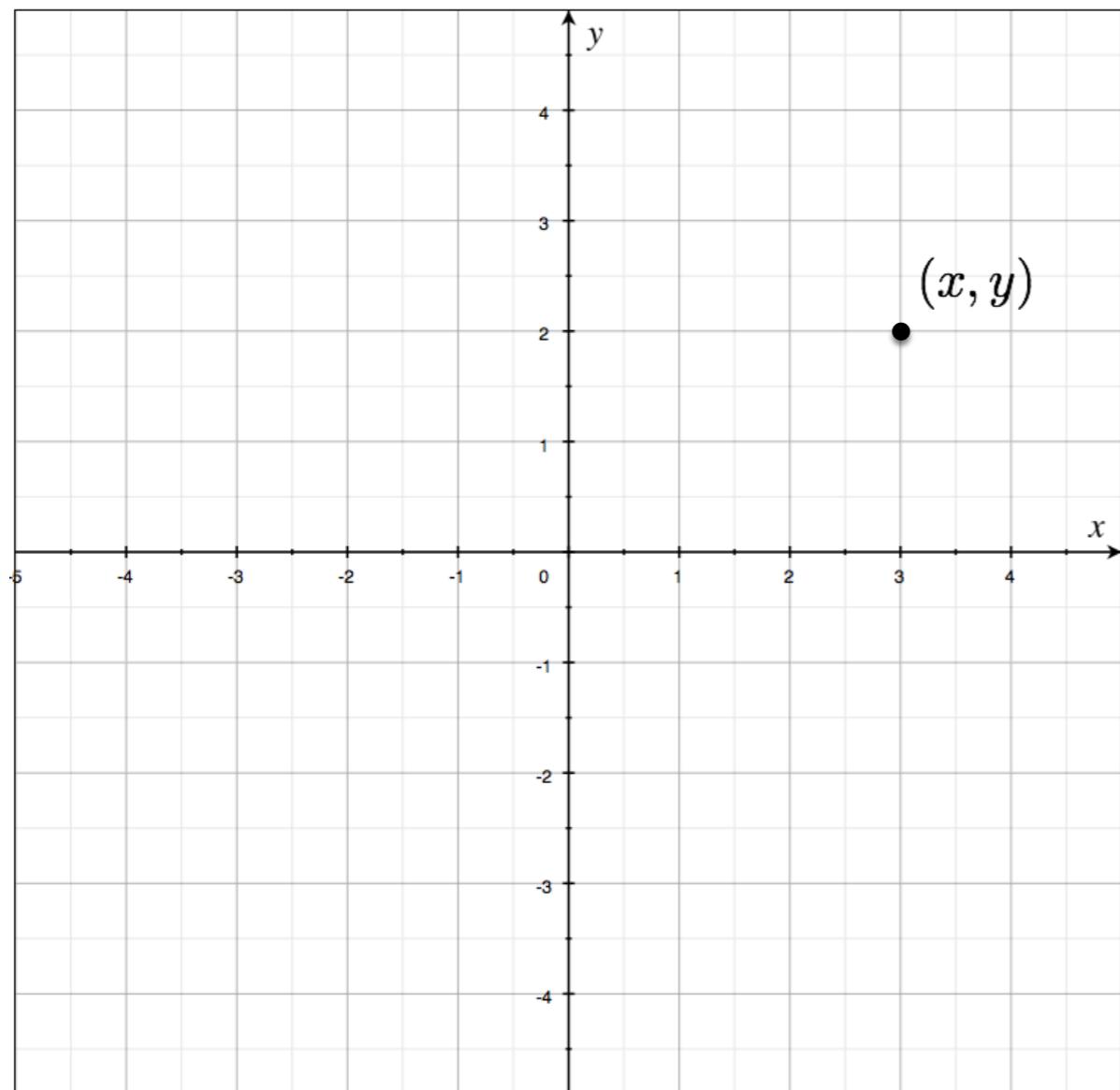
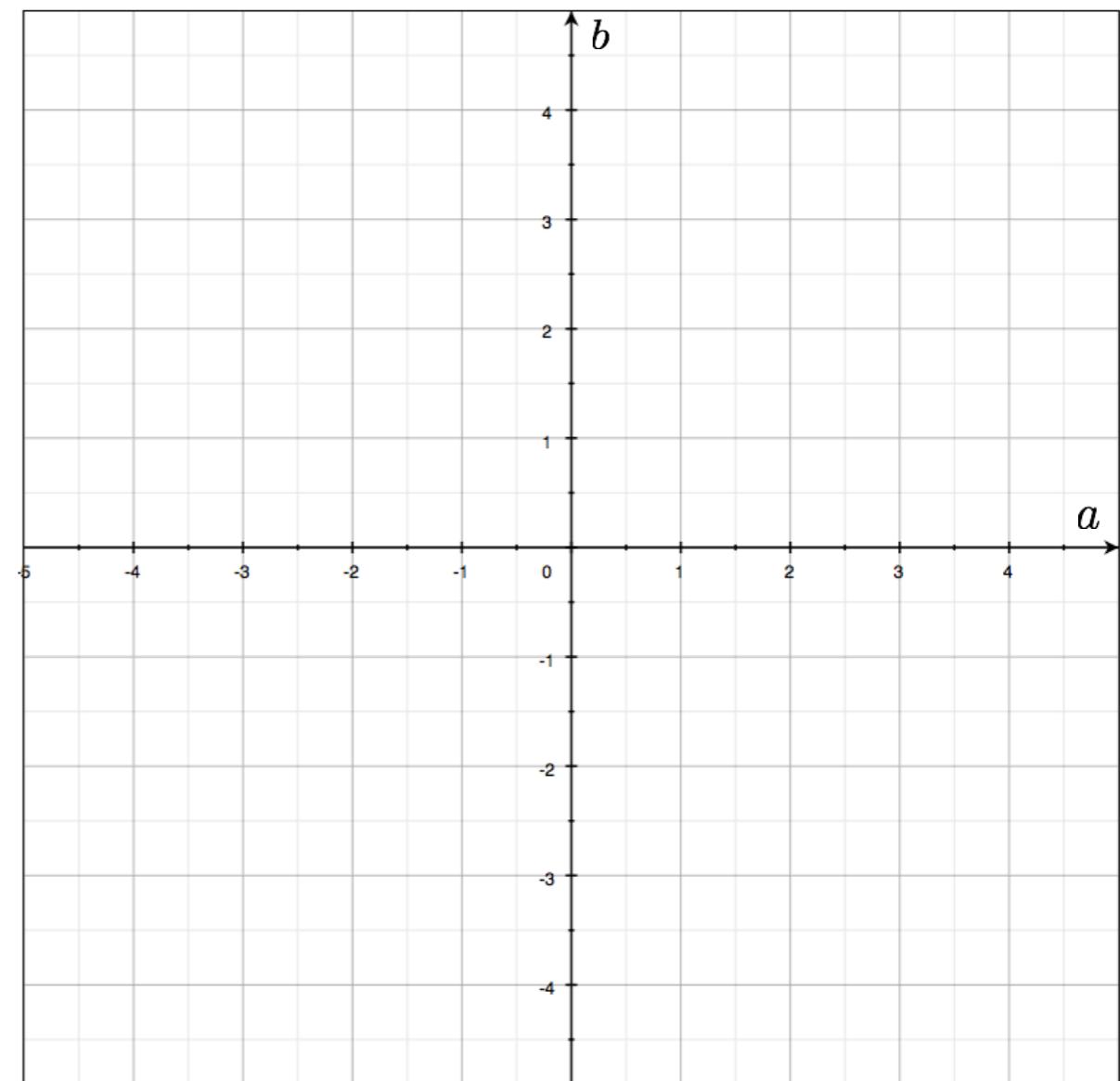


Image space

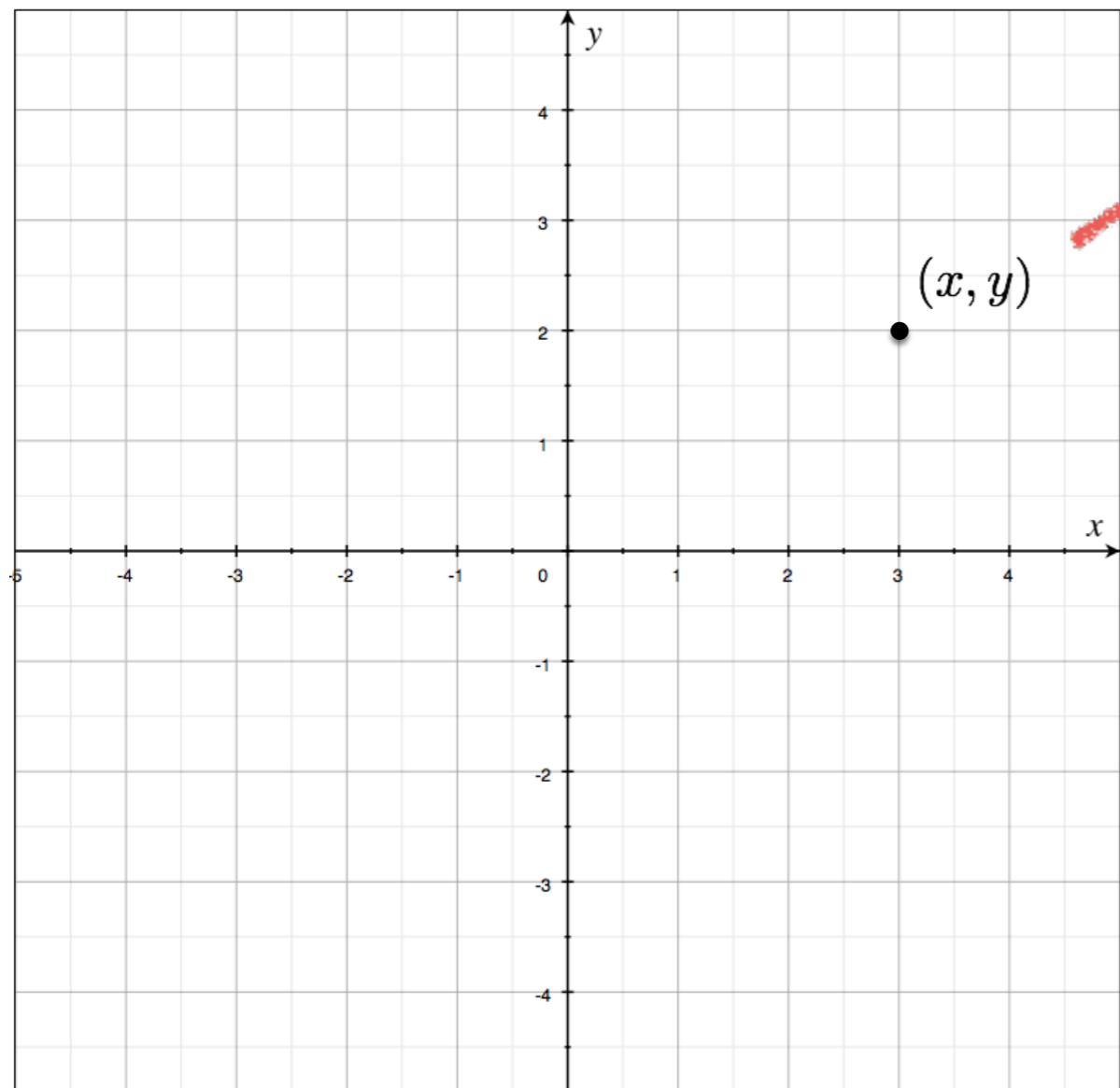
parameters  
 $(x - a)^2 + (y - b)^2 = r^2$   
variables



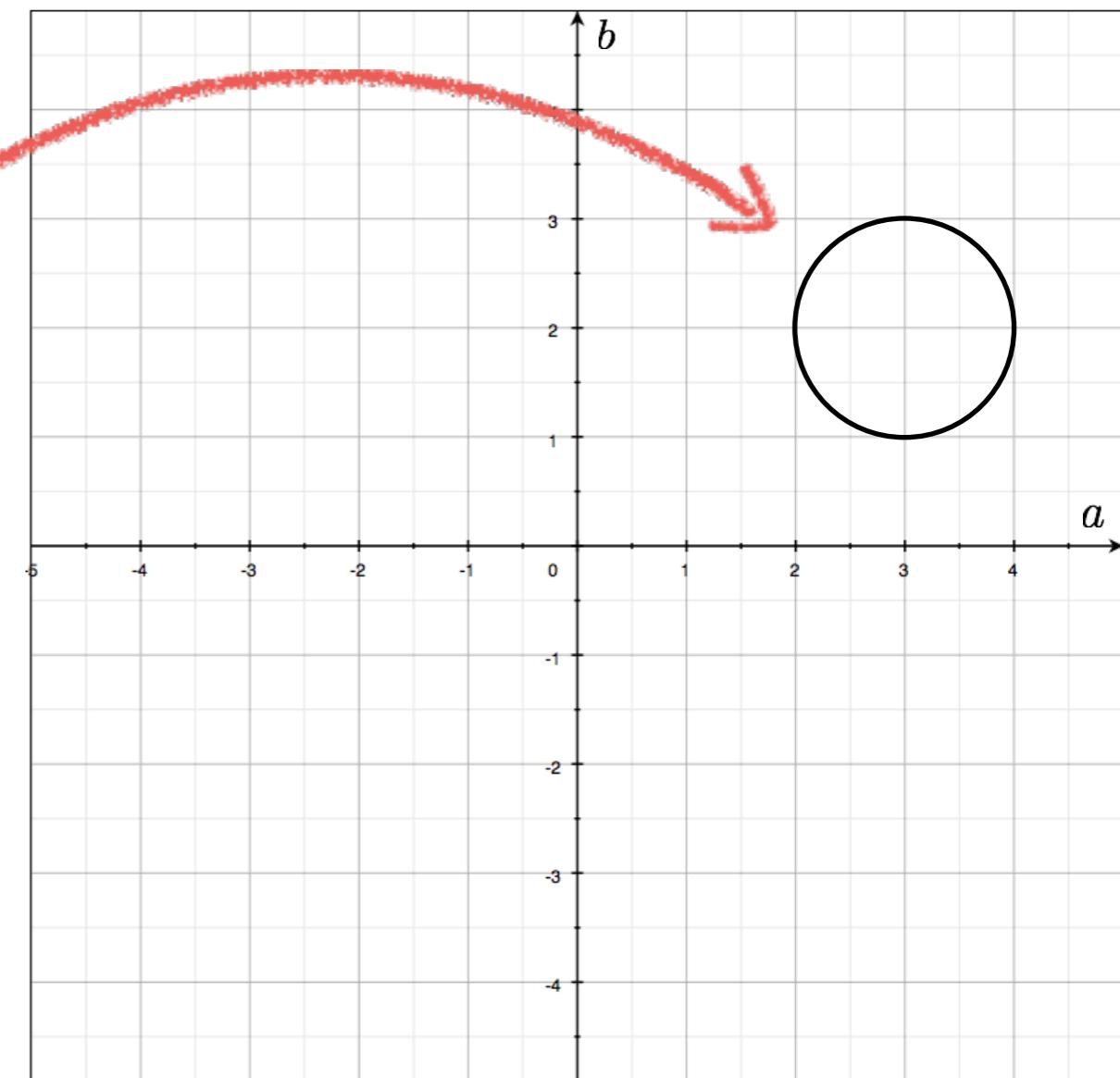
Parameter space

*What does a point in image space correspond to in parameter space?*

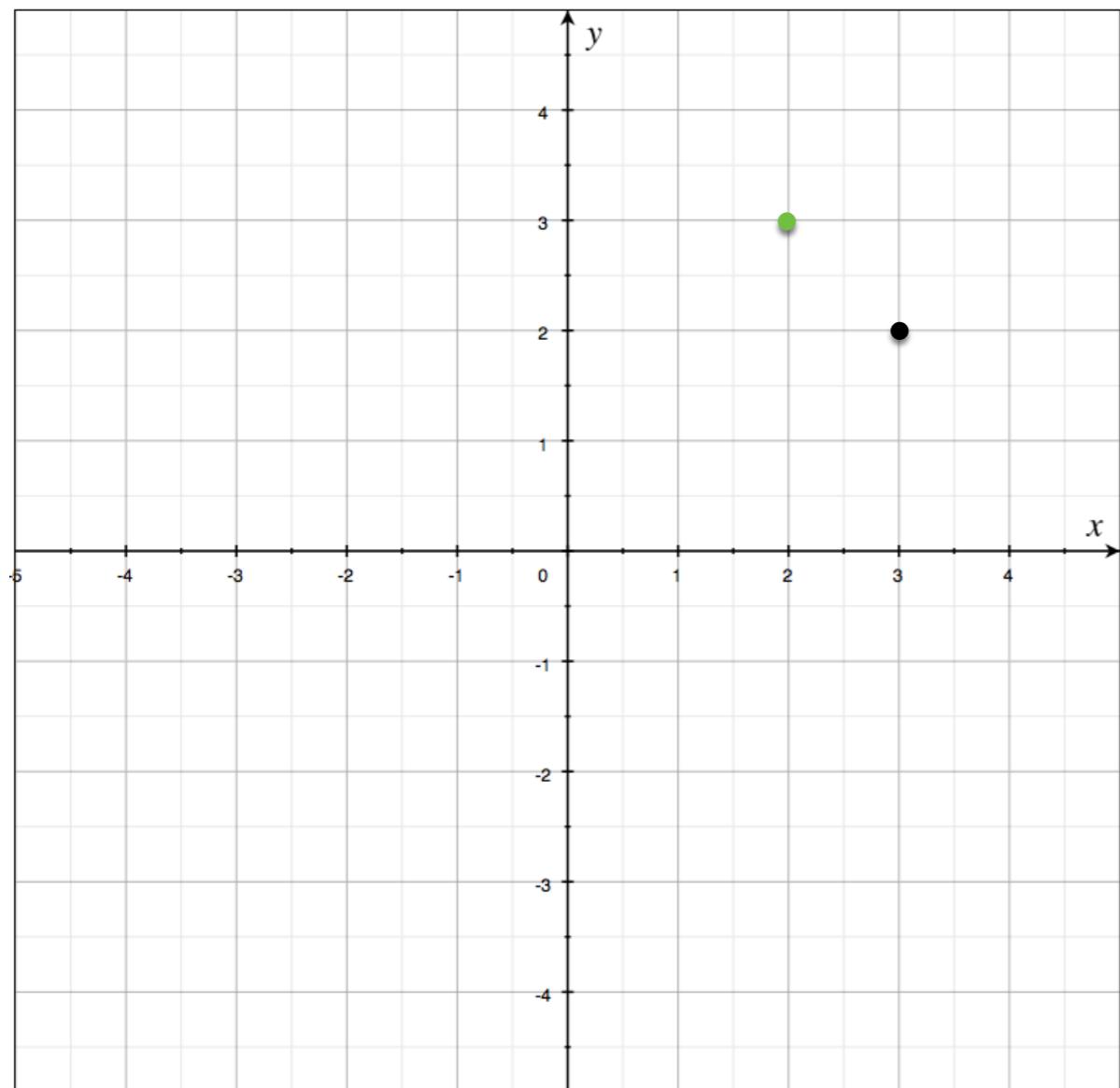
parameters  
 $(x - a)^2 + (y - b)^2 = r^2$   
variables



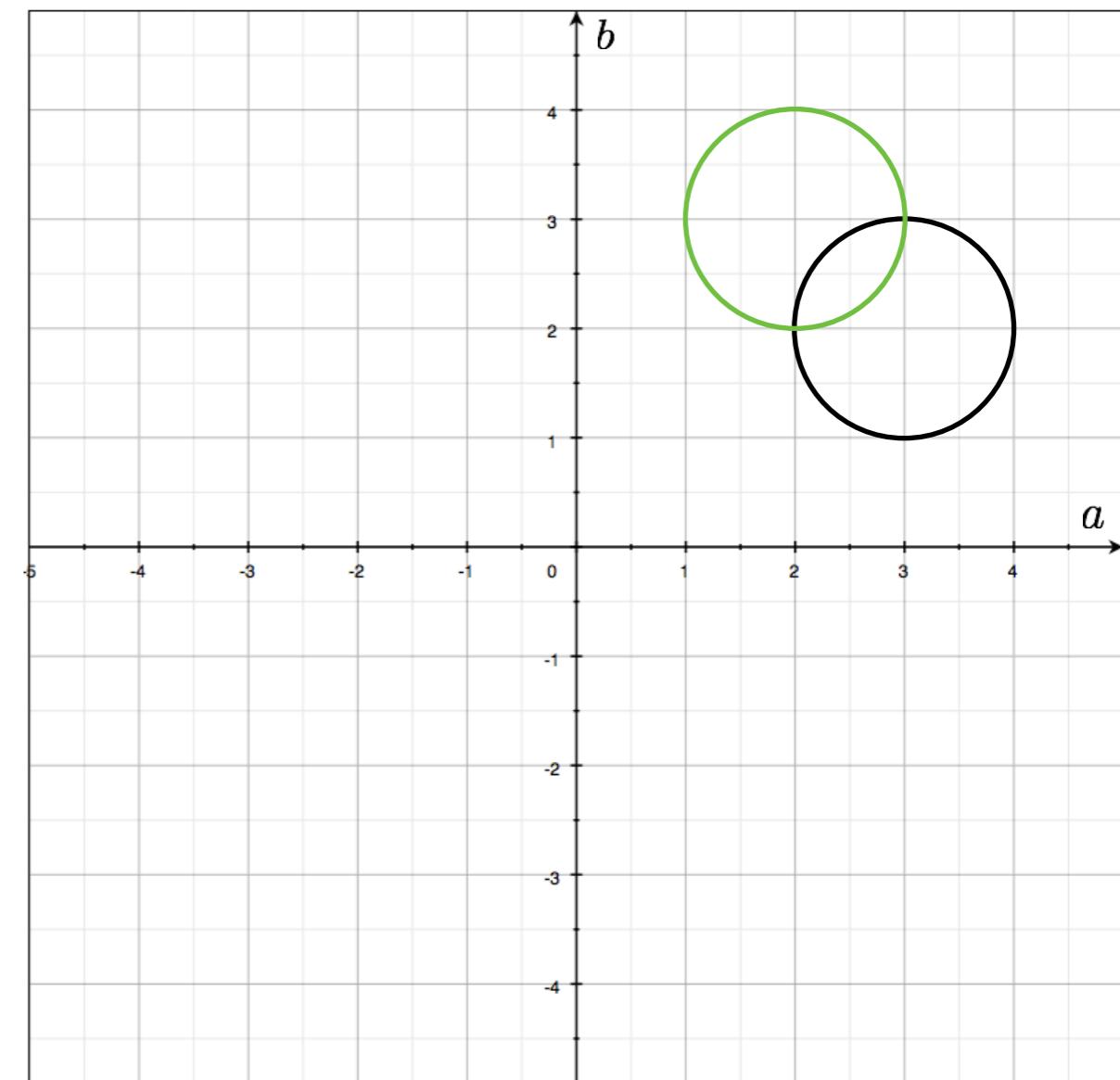
parameters  
 $(x - a)^2 + (y - b)^2 = r^2$   
variables



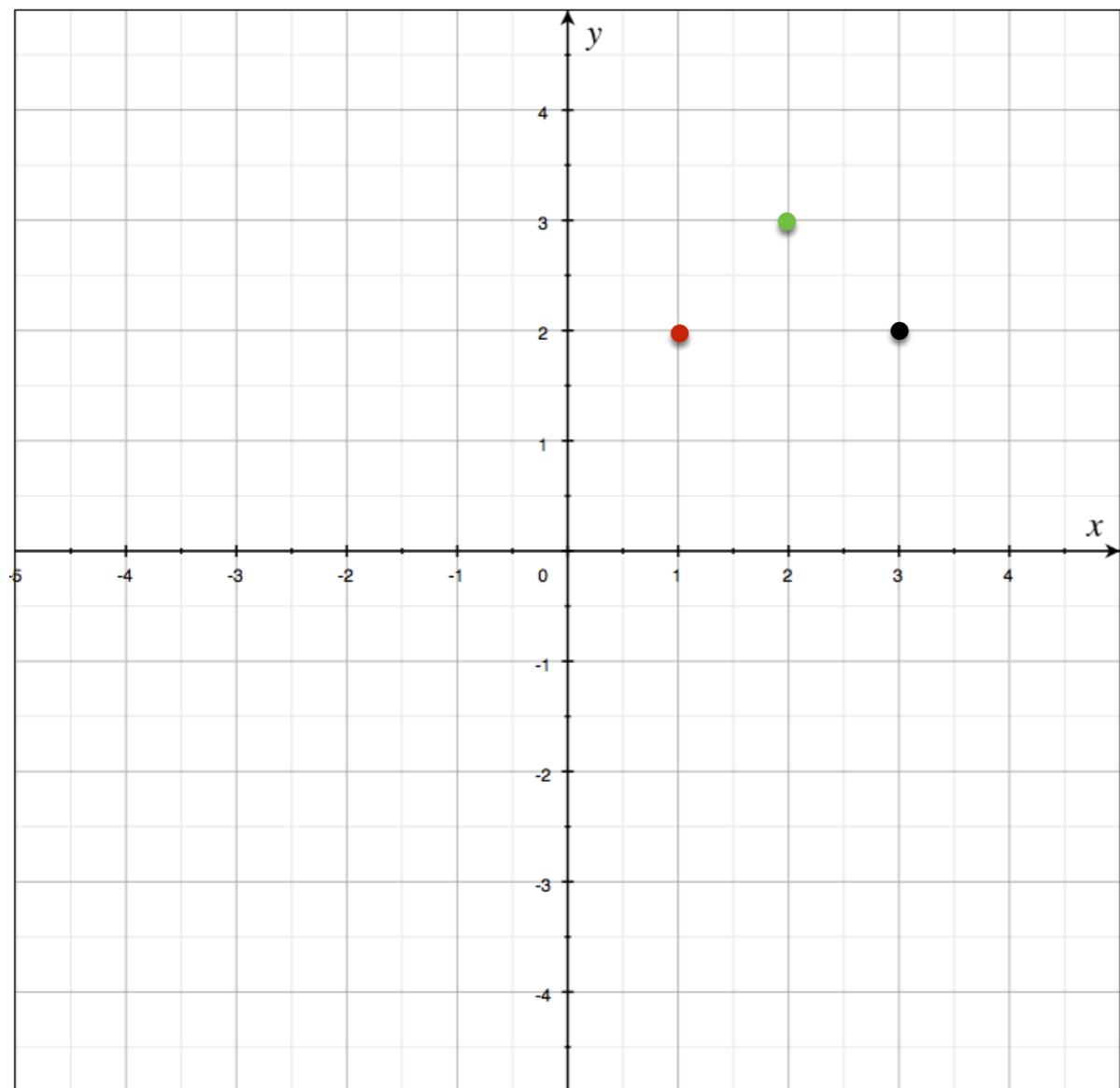
parameters  
 $(x - a)^2 + (y - b)^2 = r^2$   
variables



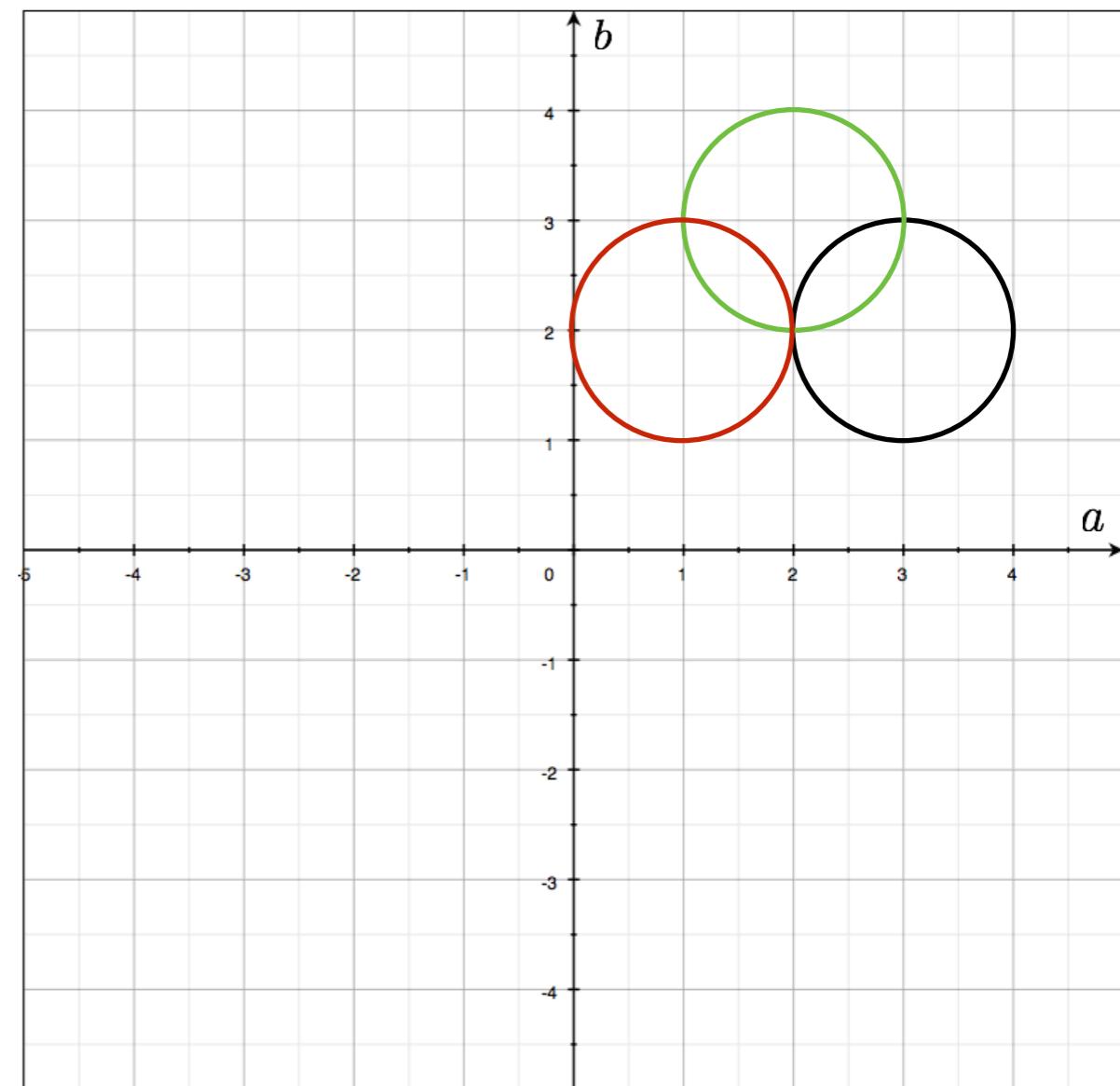
parameters  
 $(x - a)^2 + (y - b)^2 = r^2$   
variables



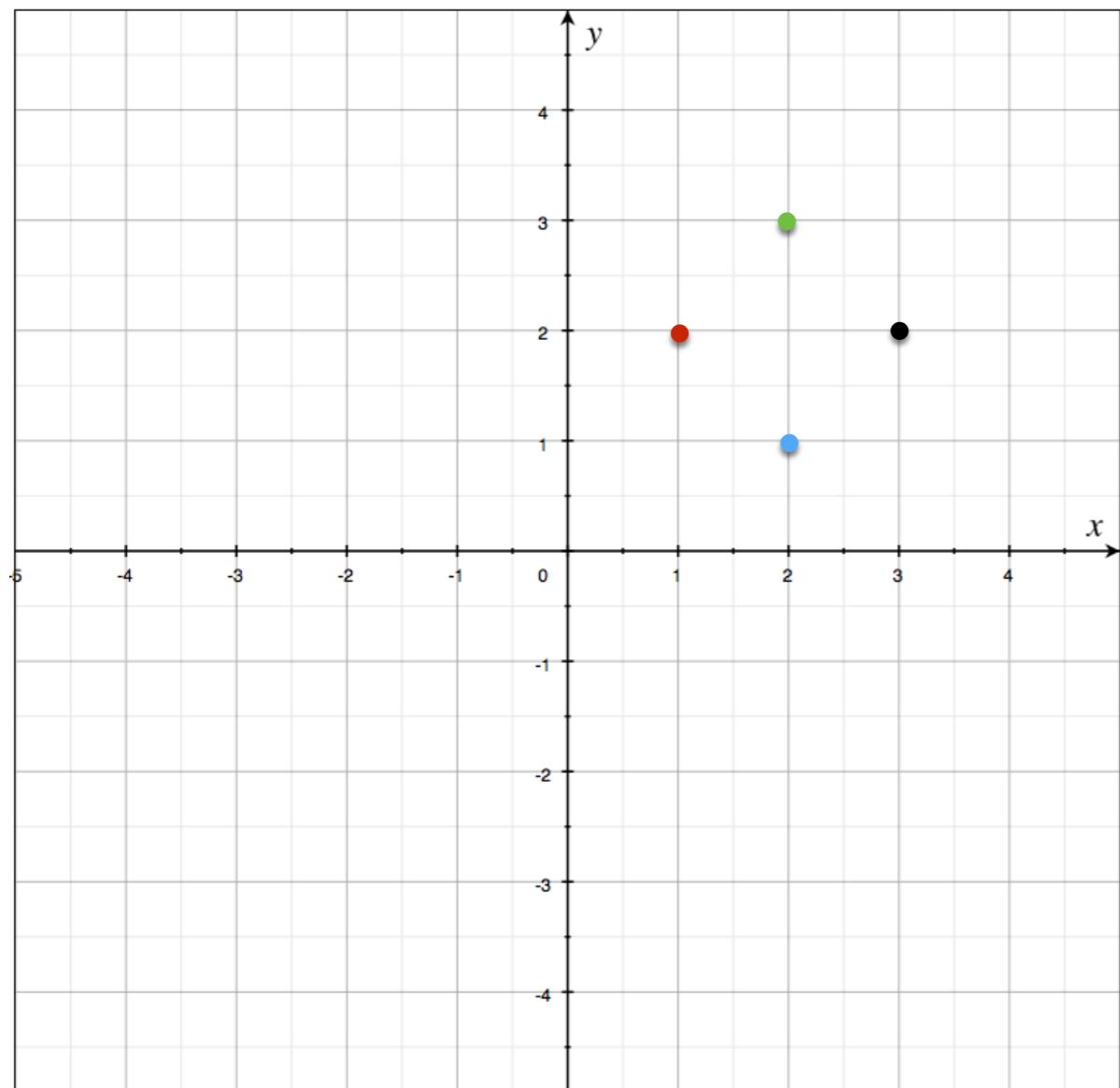
parameters  
 $(x - a)^2 + (y - b)^2 = r^2$   
variables



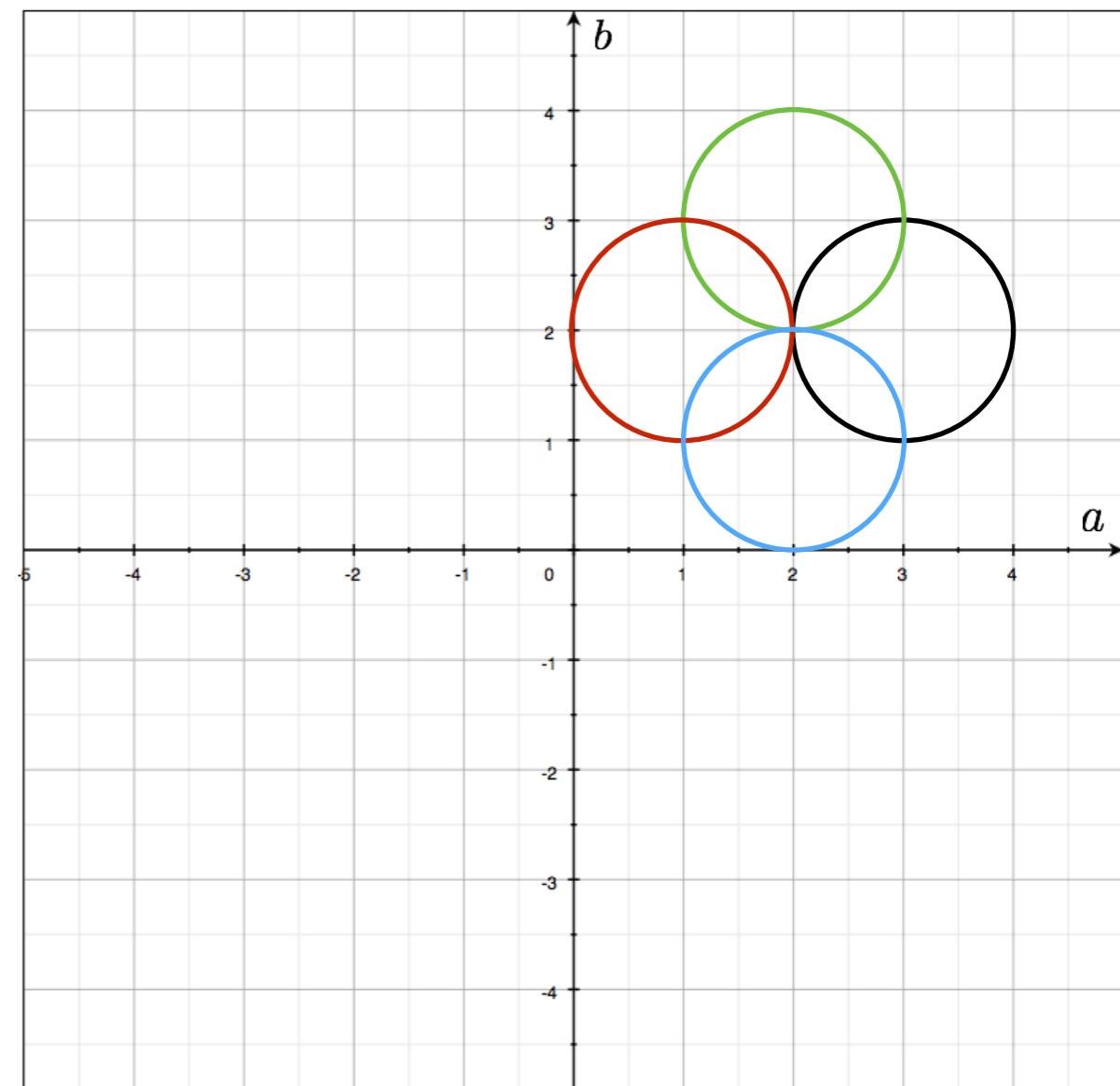
parameters  
 $(x - a)^2 + (y - b)^2 = r^2$   
variables



parameters  
 $(x - a)^2 + (y - b)^2 = r^2$   
variables



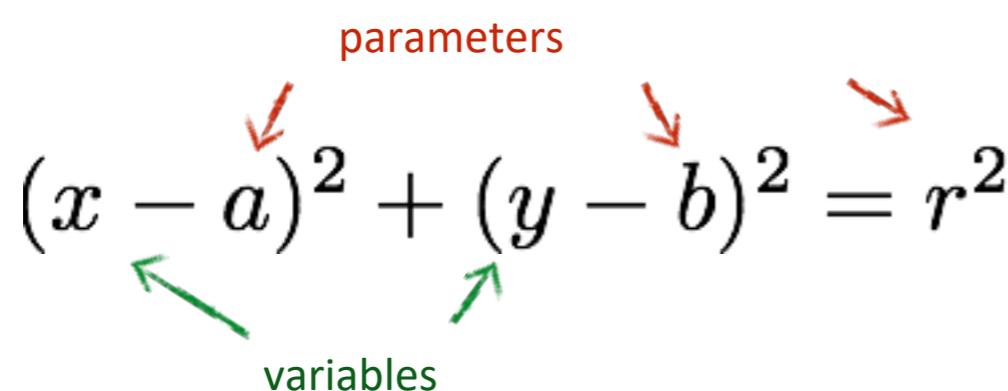
parameters  
 $(x - a)^2 + (y - b)^2 = r^2$   
variables



# What if radius is unknown?

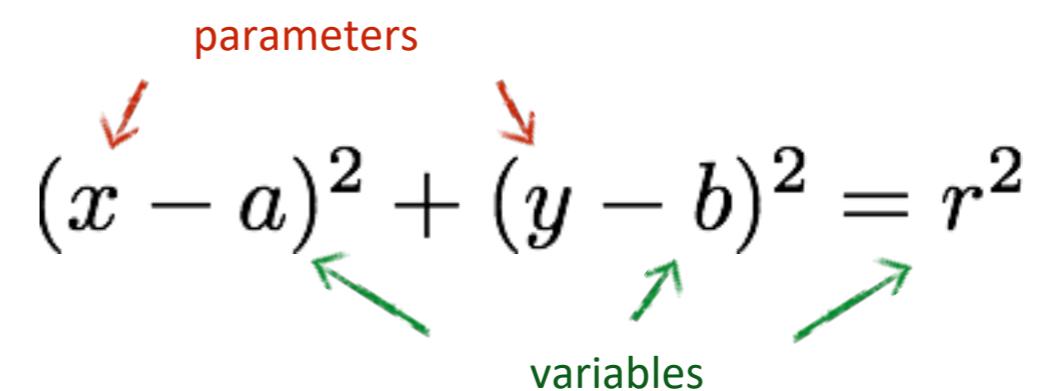
$$(x - a)^2 + (y - b)^2 = r^2$$

parameters  
variables



$$(x - a)^2 + (y - b)^2 = r^2$$

parameters  
variables



# What if radius is unknown?

$$(x - a)^2 + (y - b)^2 = r^2$$

parameters  
variables

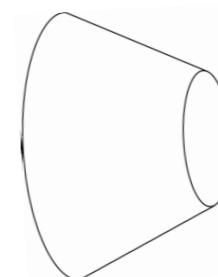
$$(x - a)^2 + (y - b)^2 = r^2$$

parameters  
variables

If radius is not known: 3D Hough Space!

Use Accumulator array  $A(a, b, r)$

Surface shape in Hough space is complicated

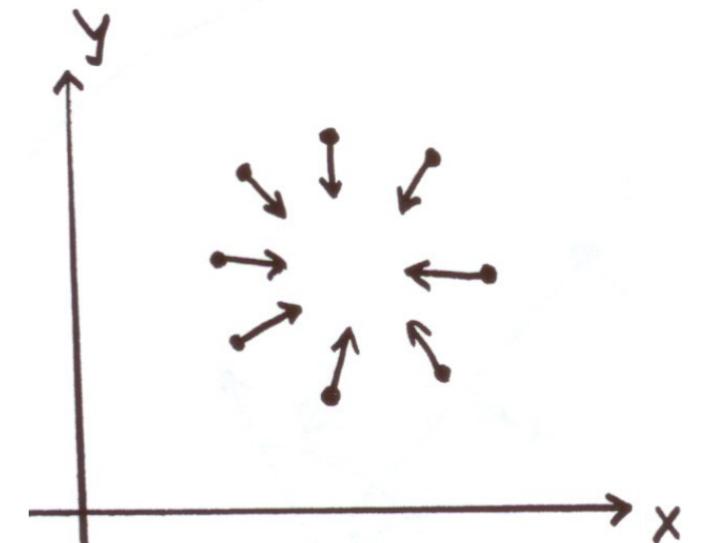


# Using Gradient Information

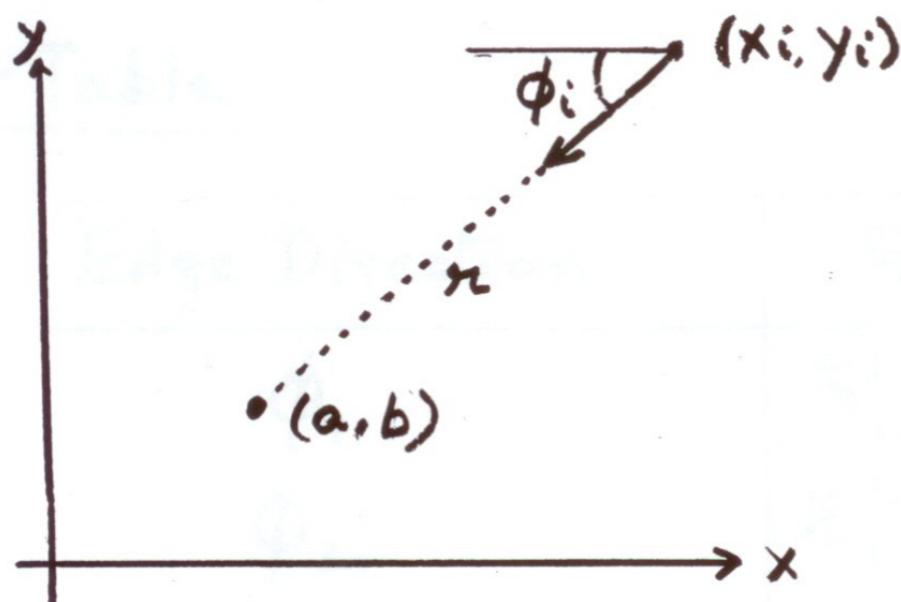
Gradient information can save lot of computation:

Edge Location  $(x_i, y_i)$

Edge Direction  $\phi_i$



Assume radius is known:



$$a = x - r \cos\phi$$

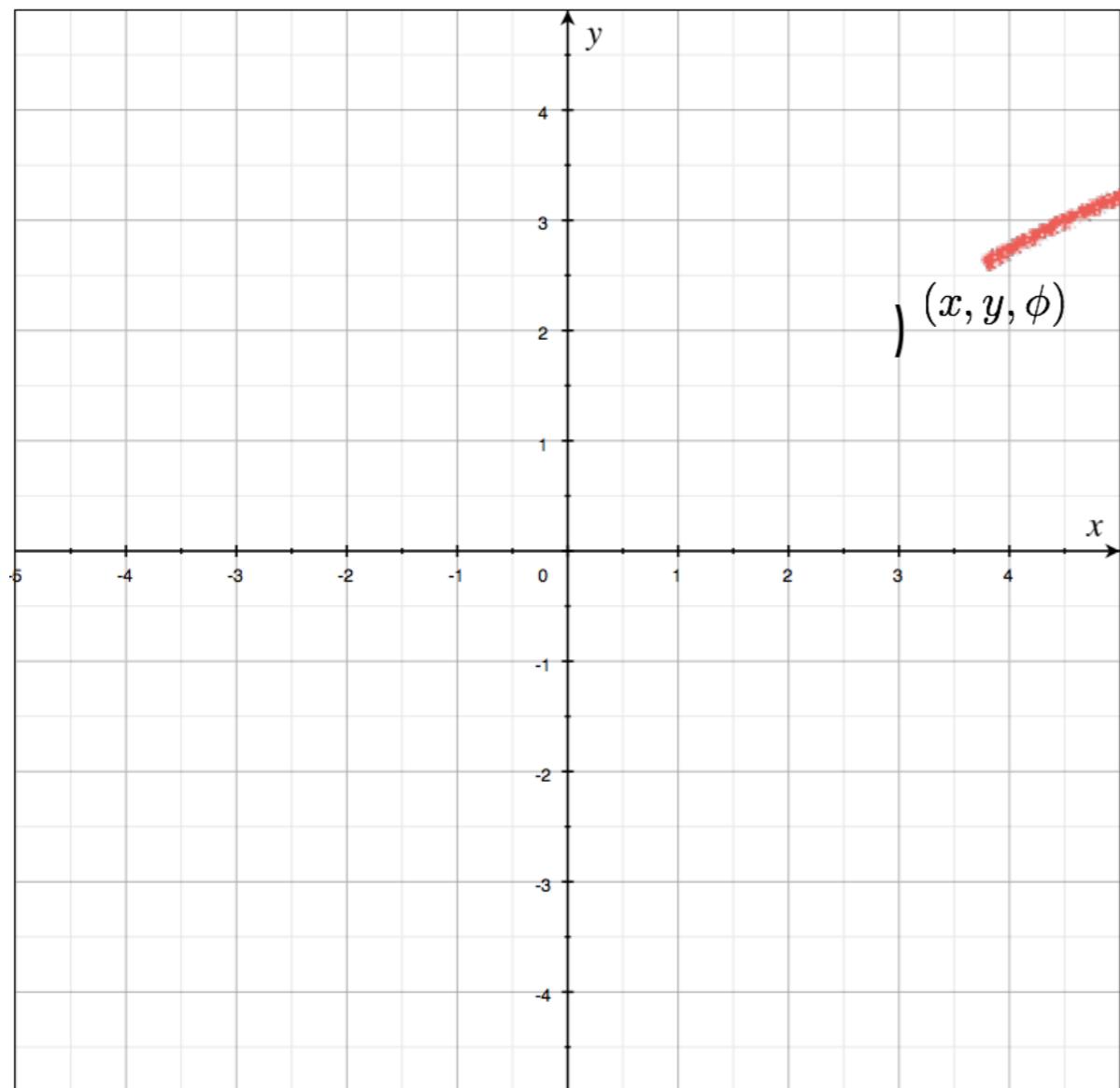
$$b = y - r \sin\phi$$

*Need to increment only one point in accumulator!*

parameters

$$(x - a)^2 + (y - b)^2 = r^2$$

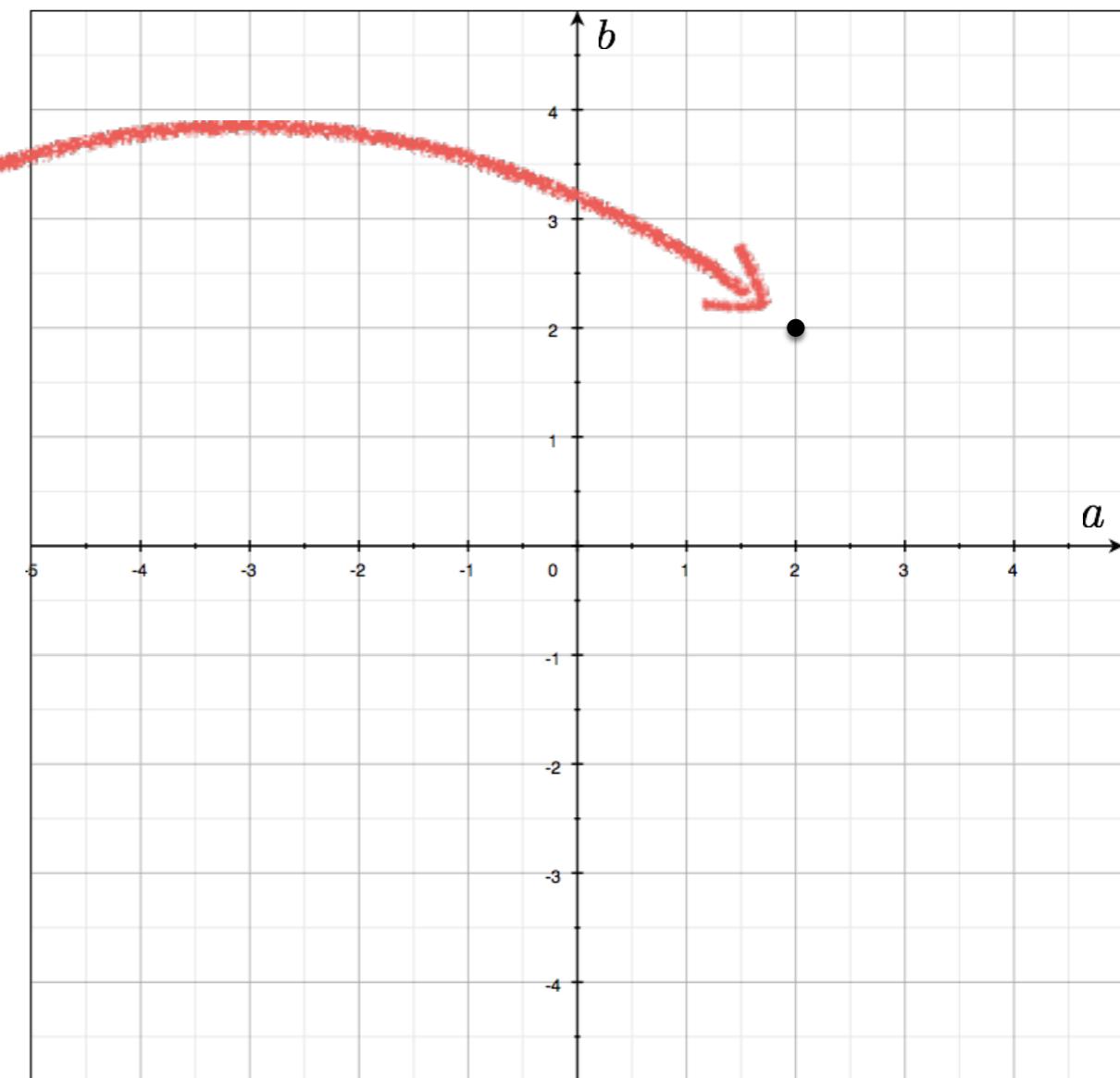
variables



parameters

$$(x - a)^2 + (y - b)^2 = r^2$$

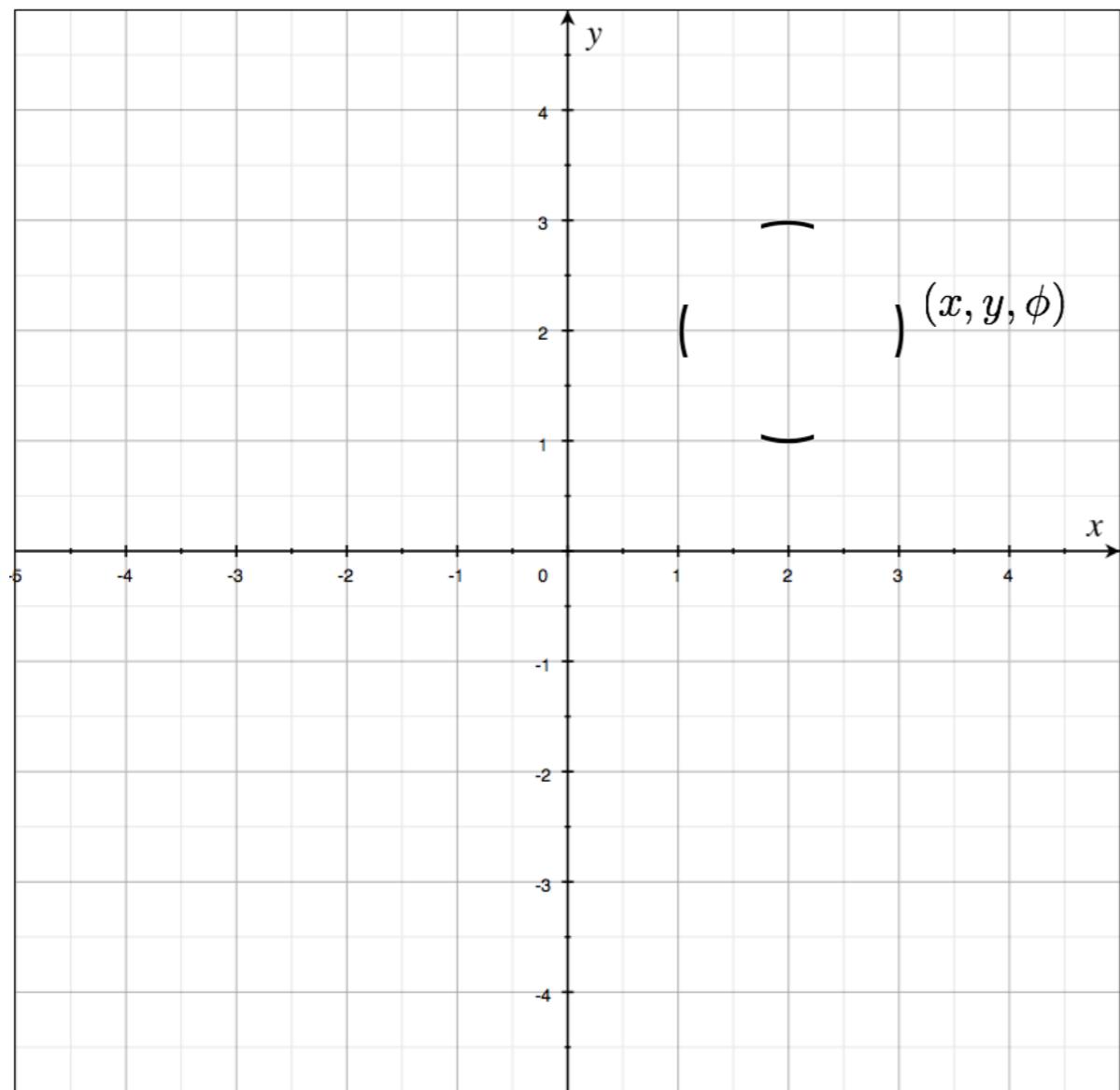
variables



parameters

$$(x - a)^2 + (y - b)^2 = r^2$$

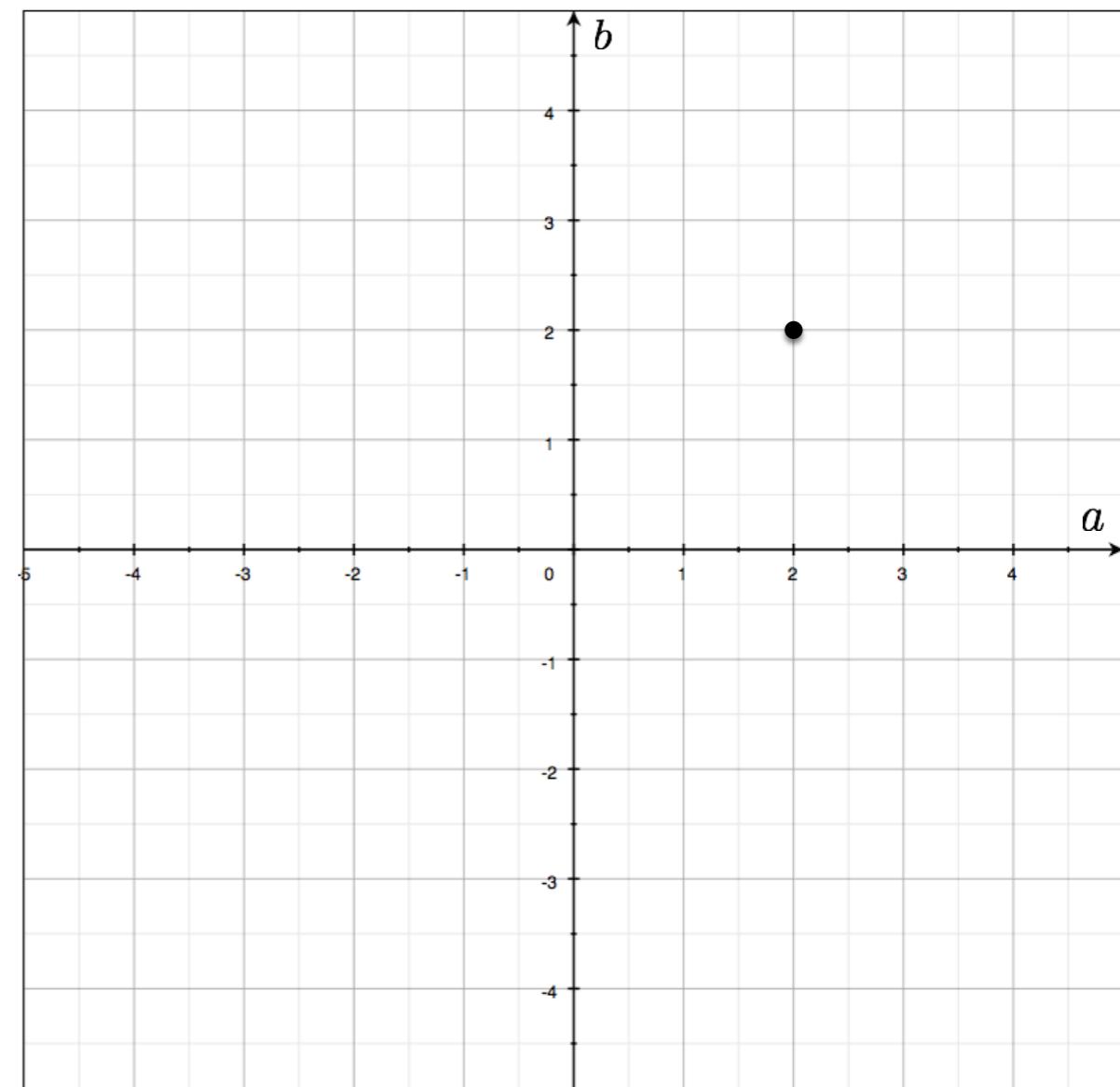
variables

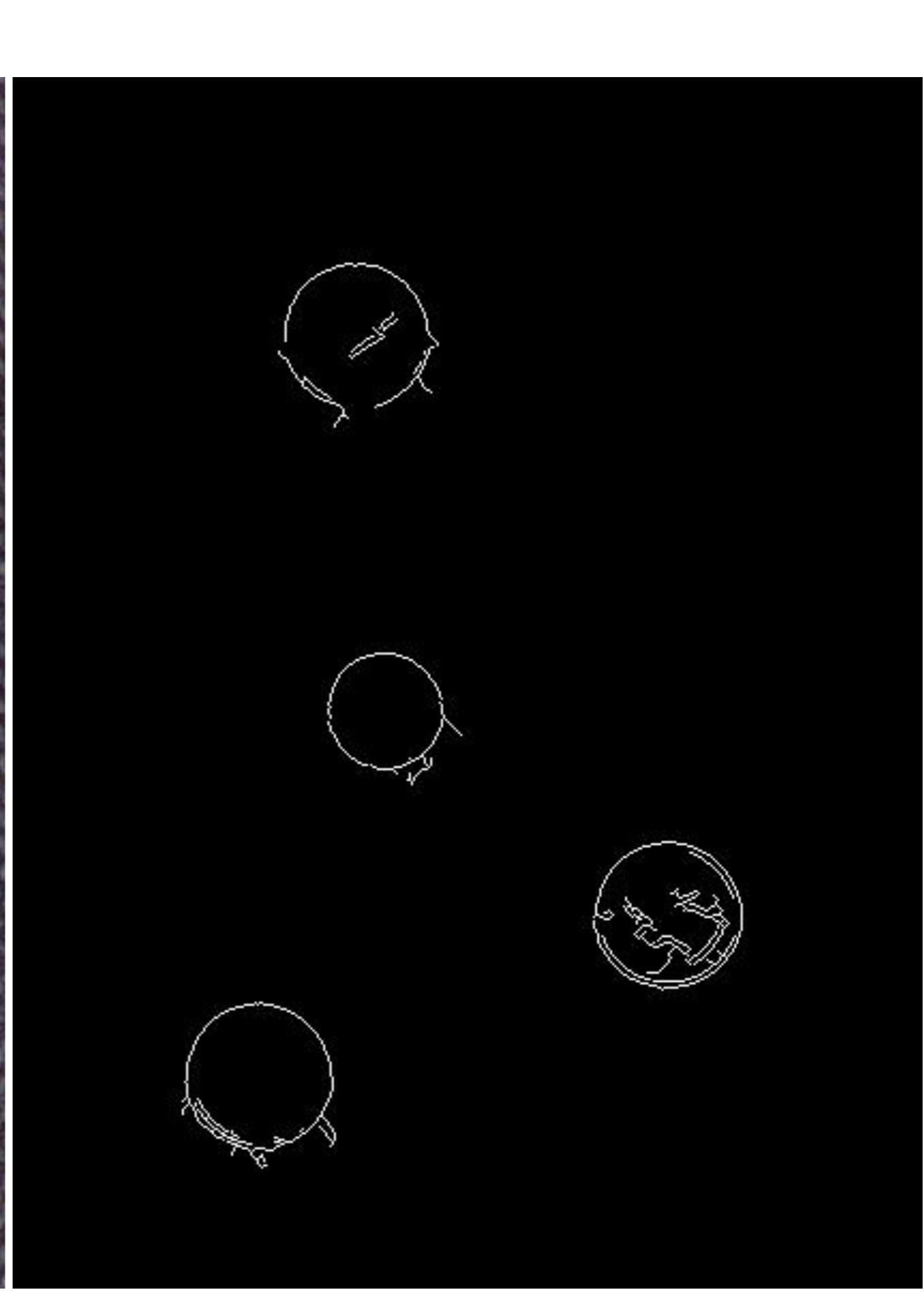


parameters

$$(x - a)^2 + (y - b)^2 = r^2$$

variables





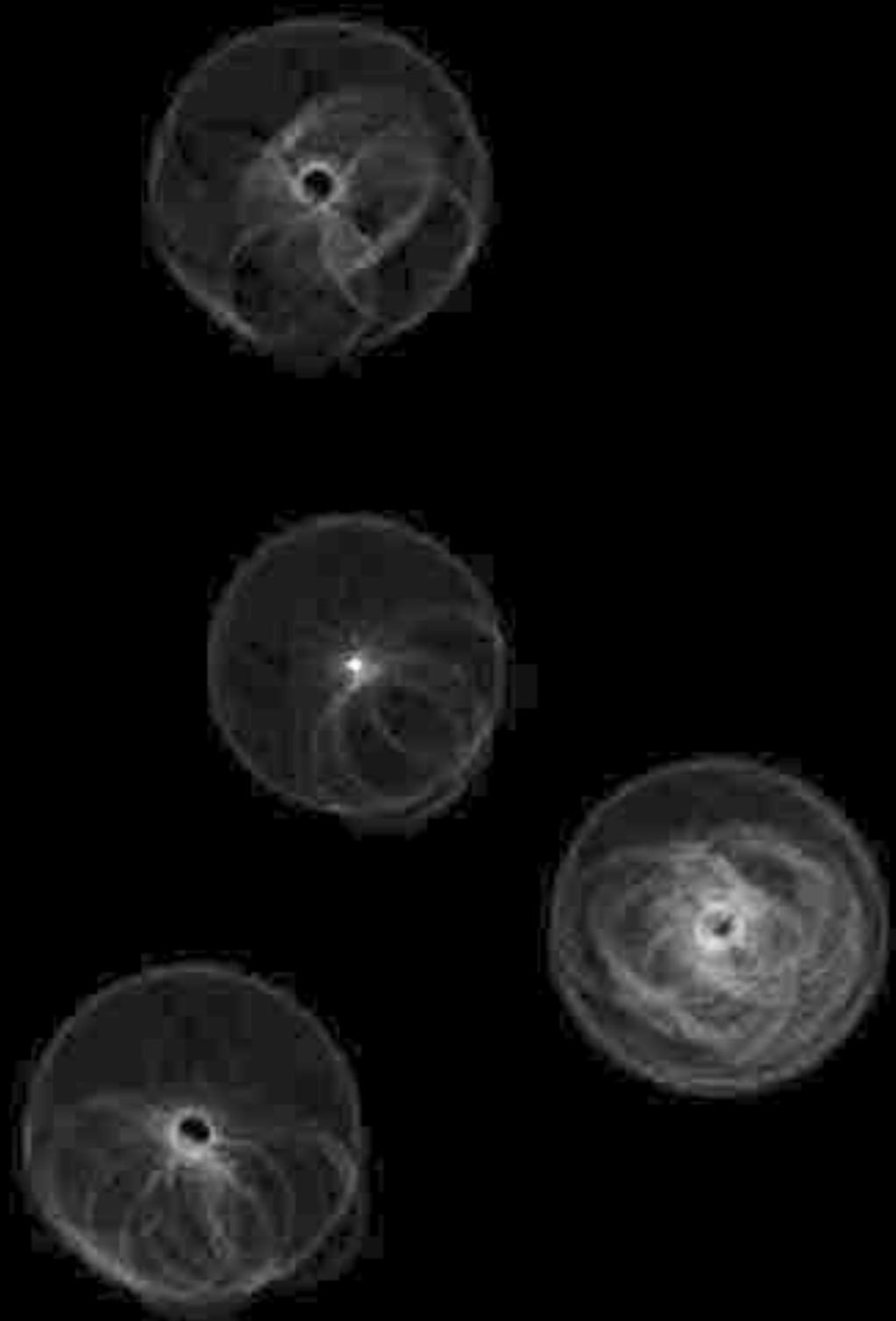
Pennie Hough detector



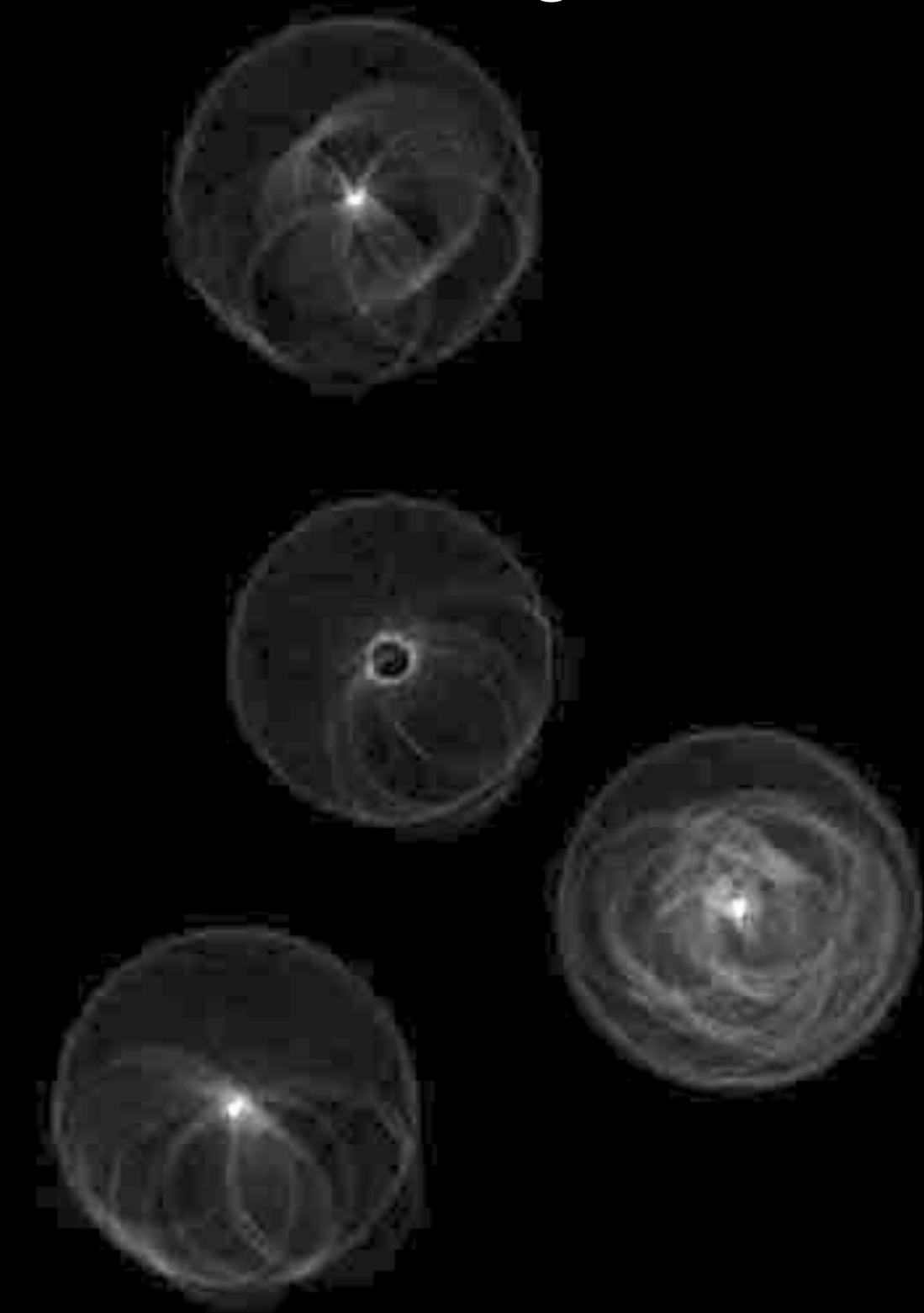
Quarter Hough detector



Pennie Hough detector



Quarter Hough detector



# The Hough transform ...

Deals with occlusion well?



Detects multiple instances?



Robust to noise?



Good computational complexity?

