

# Hough Transform

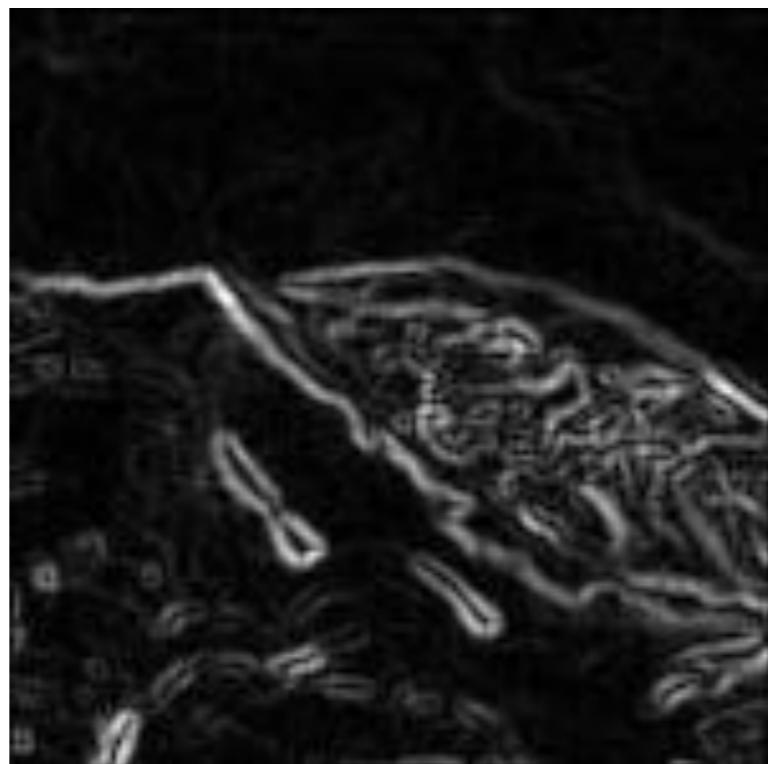
Computer Vision

**Carnegie Mellon University (Kris Kitani)**

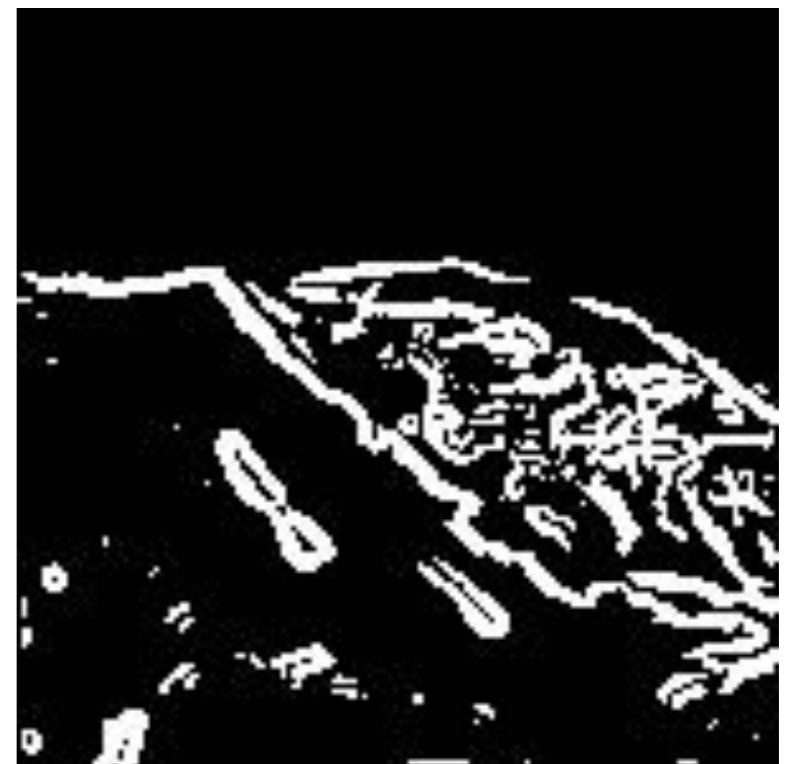
# motivation



Original image



Edge detection



Thresholding

How do we find image boundaries (lines)?

# Hough Transform

- Generic framework for detecting a shape/object
- Edges don't have to be connected
- Lines can be occluded

**Key idea:** ‘Democratic’ Detector

Each image edge votes for the possible line model

# Image and parameter space

$$y = mx + b$$

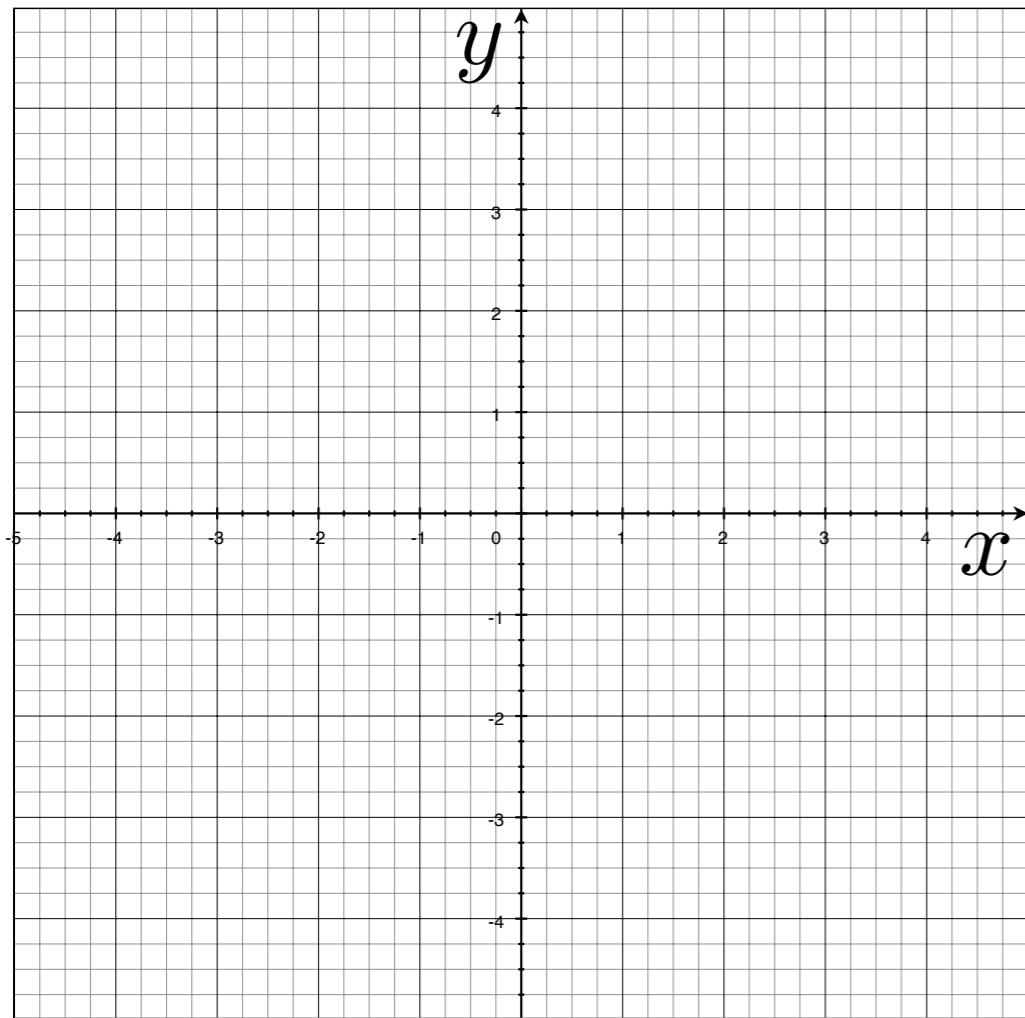
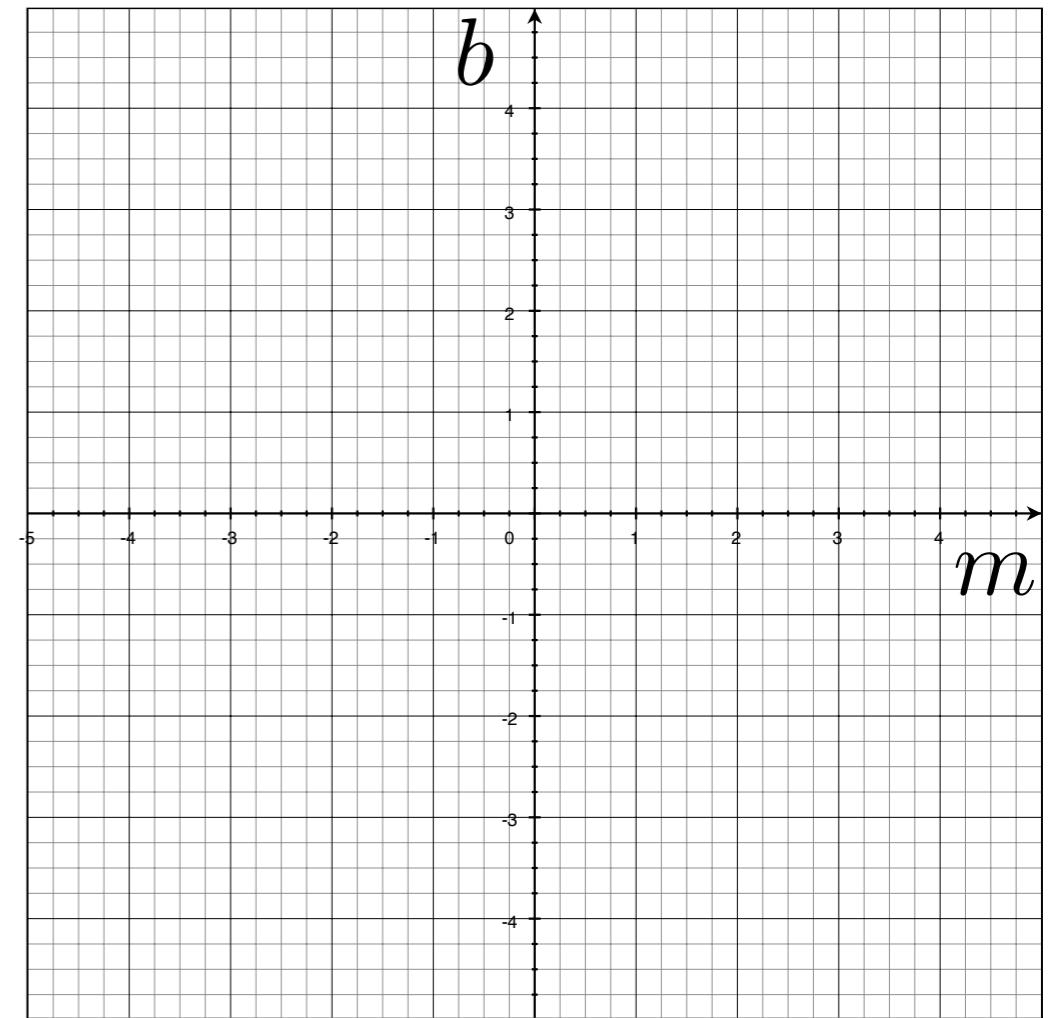


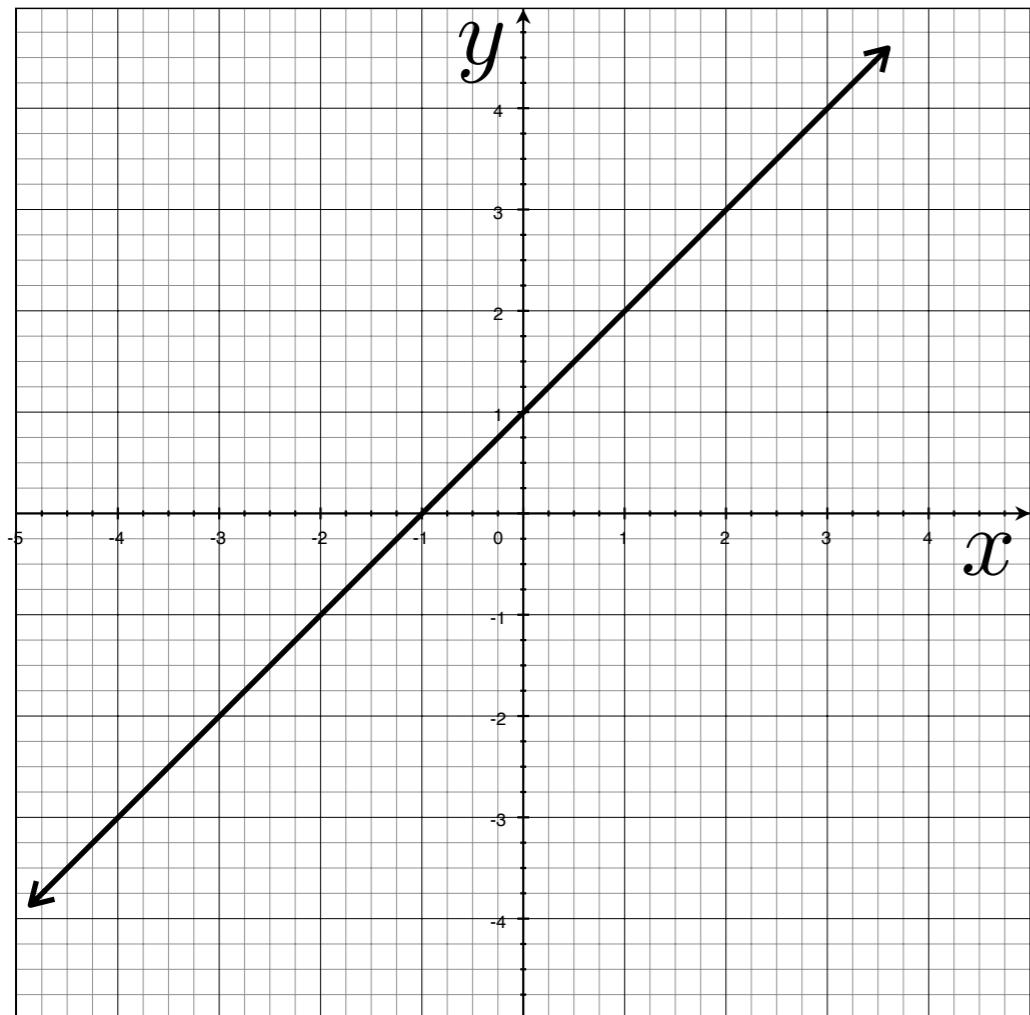
Image (variable) space



Parameter space

# Image and parameter space

variables  
 $y = mx + b$   
parameters



a line  
becomes  
a point

variables  
 $y - mx = b$   
parameters

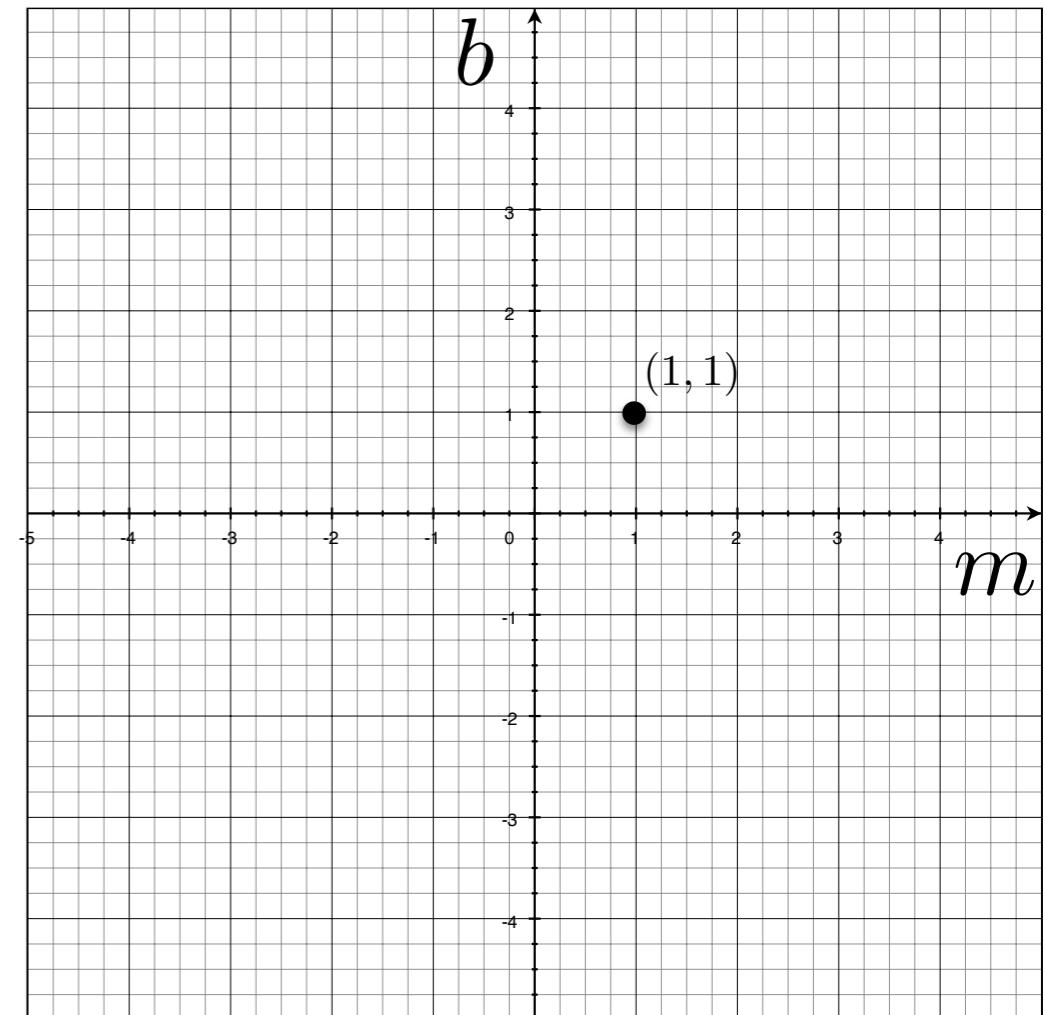
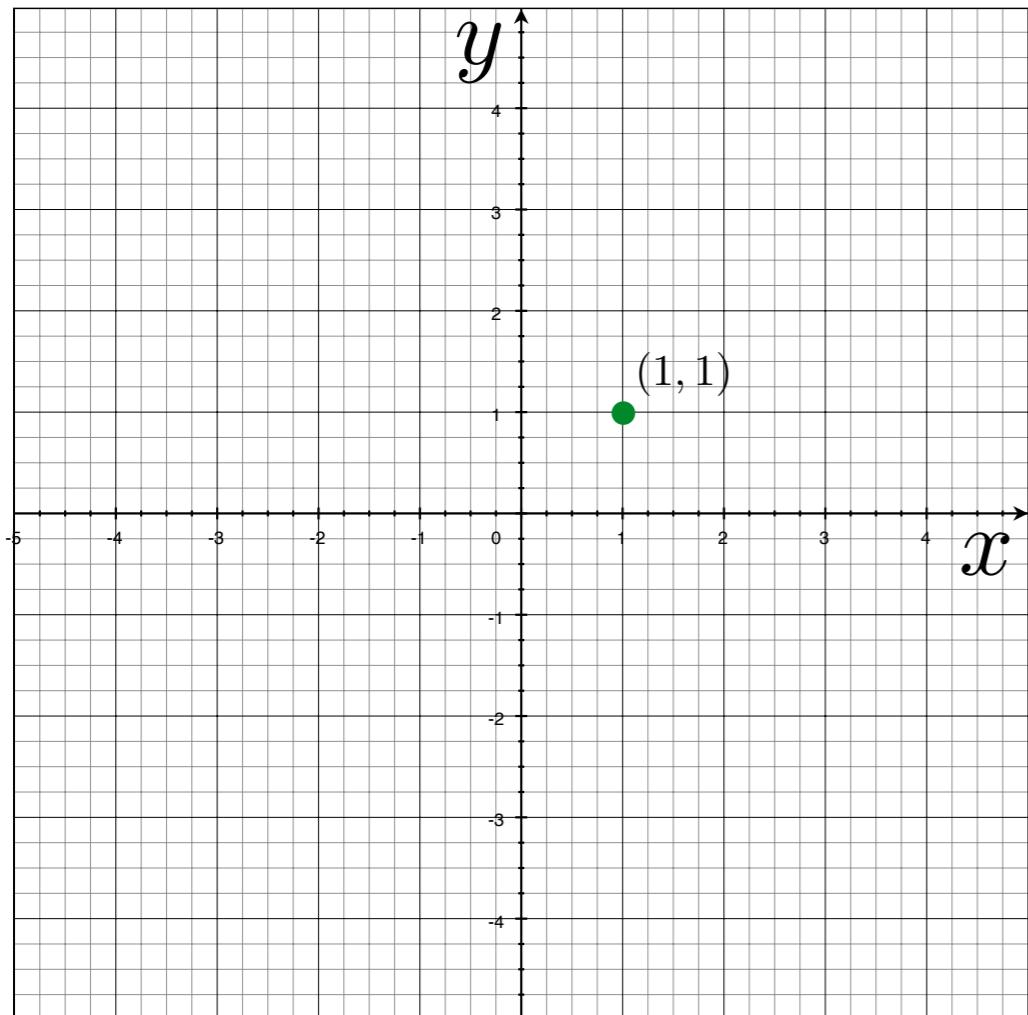


Image space

Parameter space

# Image and parameter space

variables  
 $y = mx + b$   
parameters



a point becomes  
a line

variables  
 $y - mx = b$   
parameters

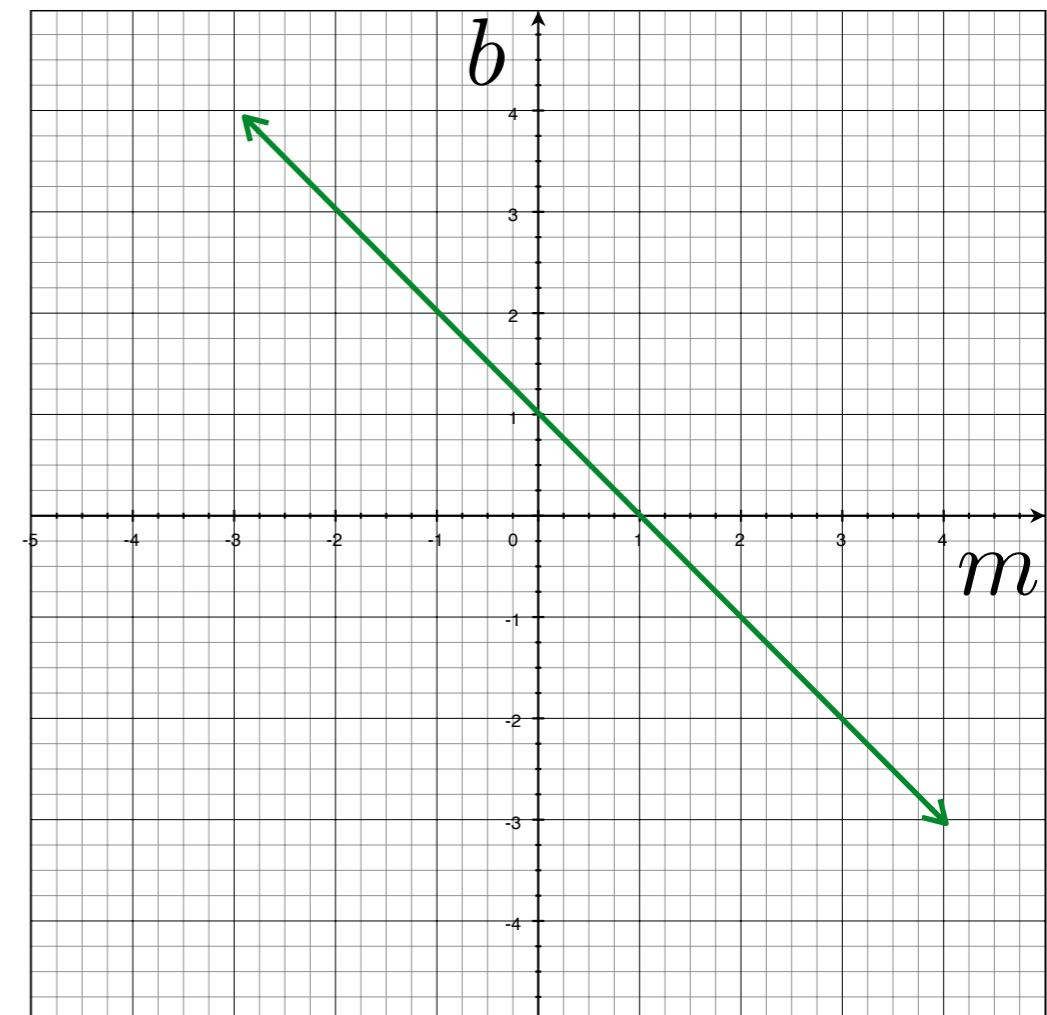
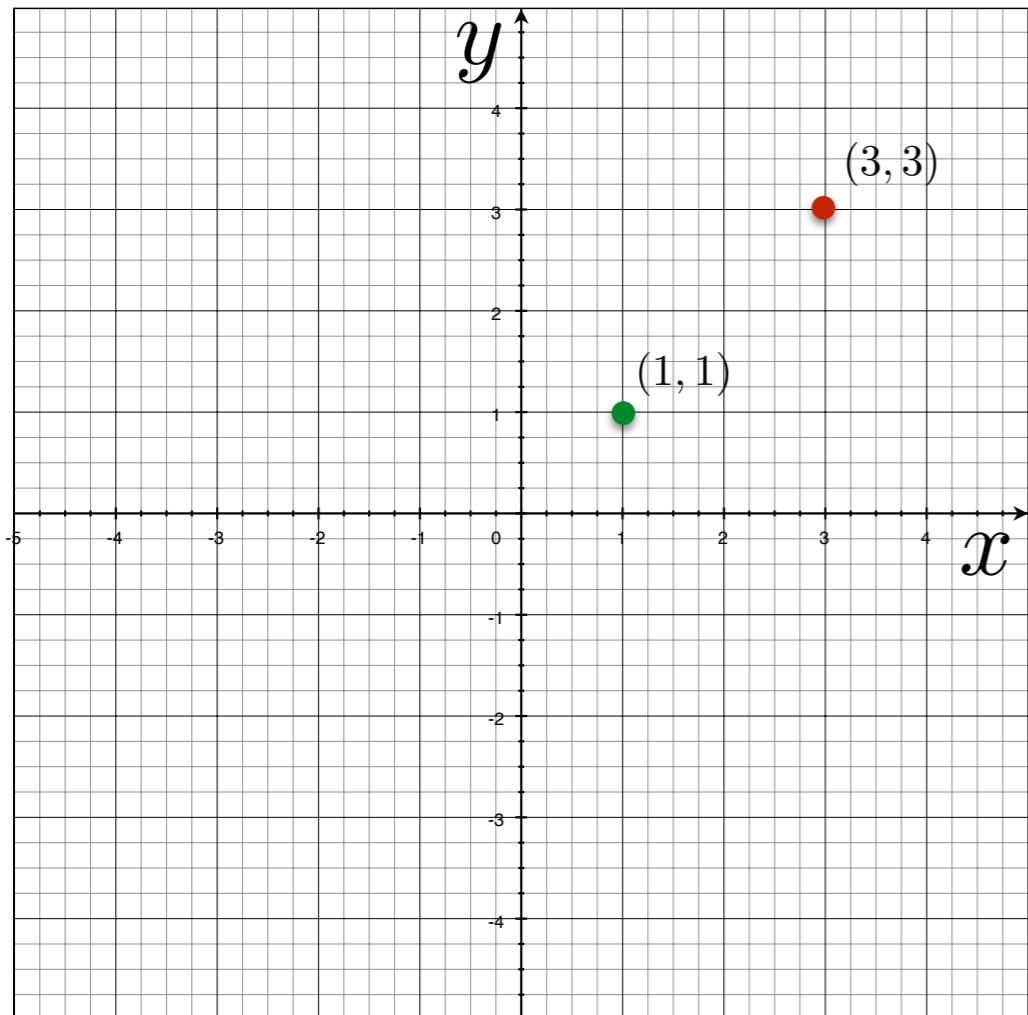


Image space

Parameter space

# Image and parameter space

variables  
 $y = mx + b$   
parameters



two points  
become  
?

variables  
 $y - mx = b$   
parameters

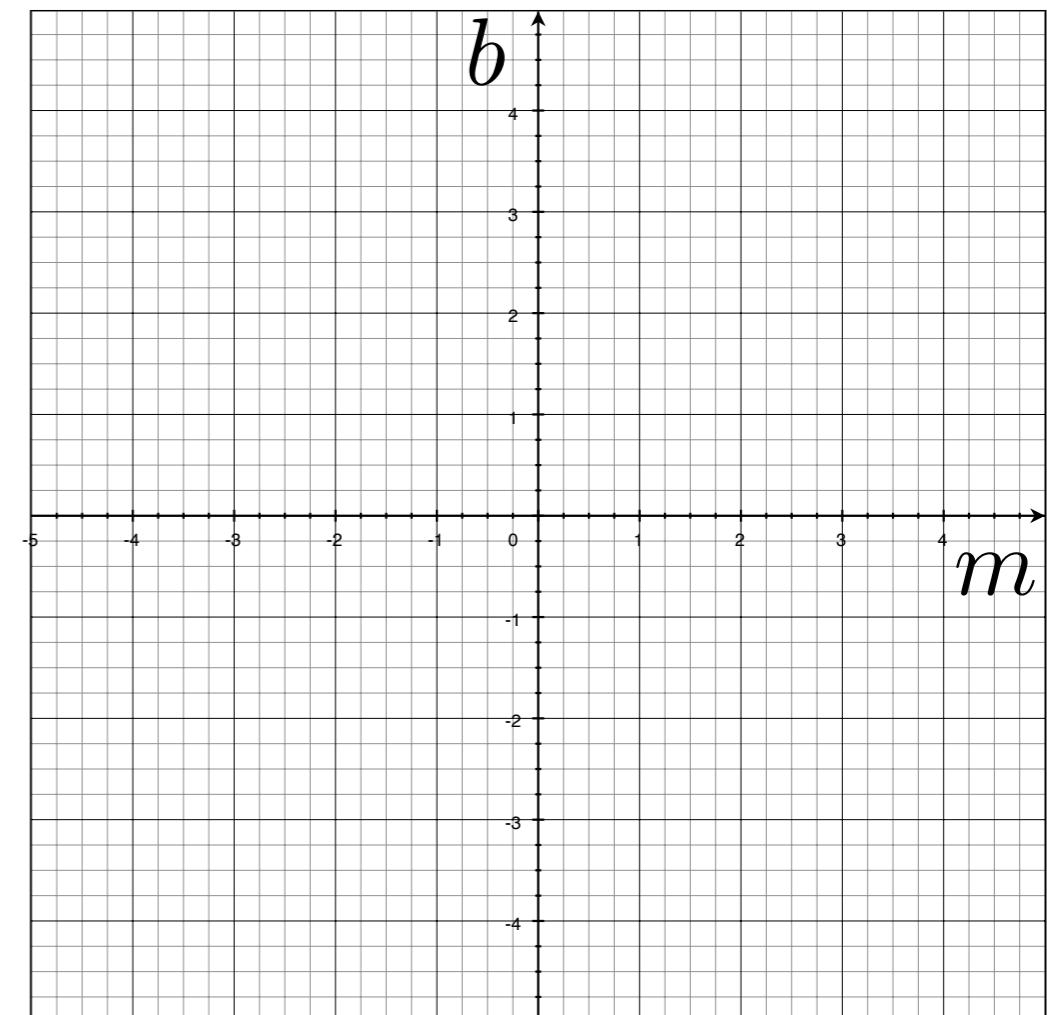
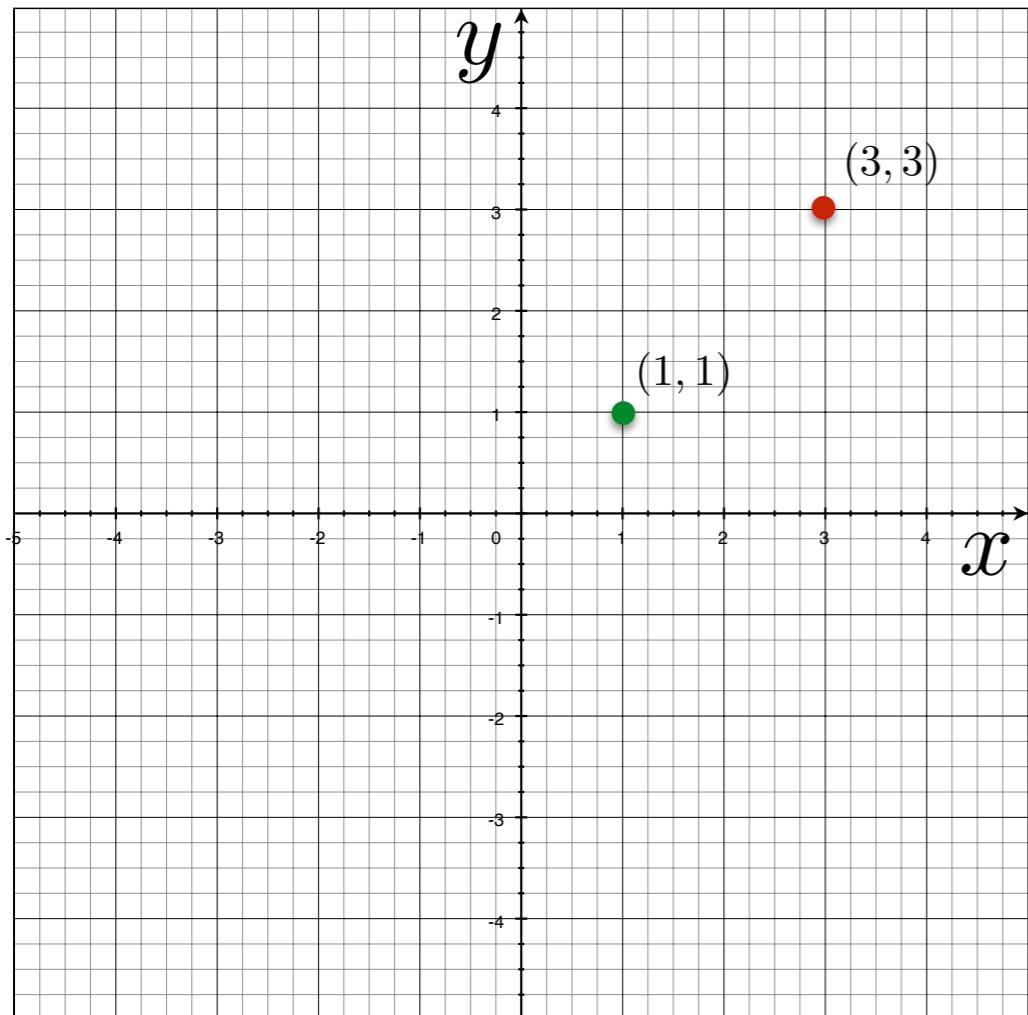


Image space

Parameter space

# Image and parameter space

variables  
 $y = mx + b$   
parameters



two points  
become  
?

variables  
 $y - mx = b$   
parameters

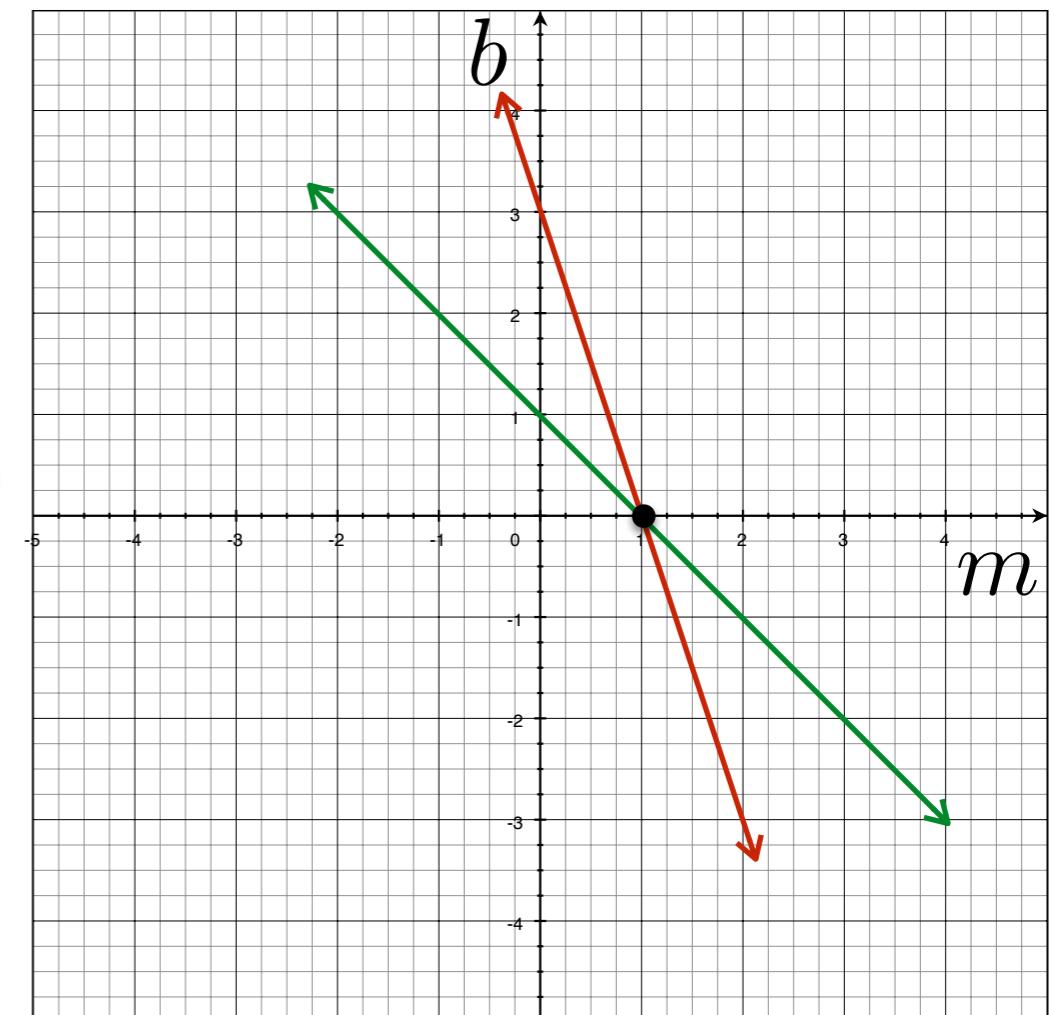
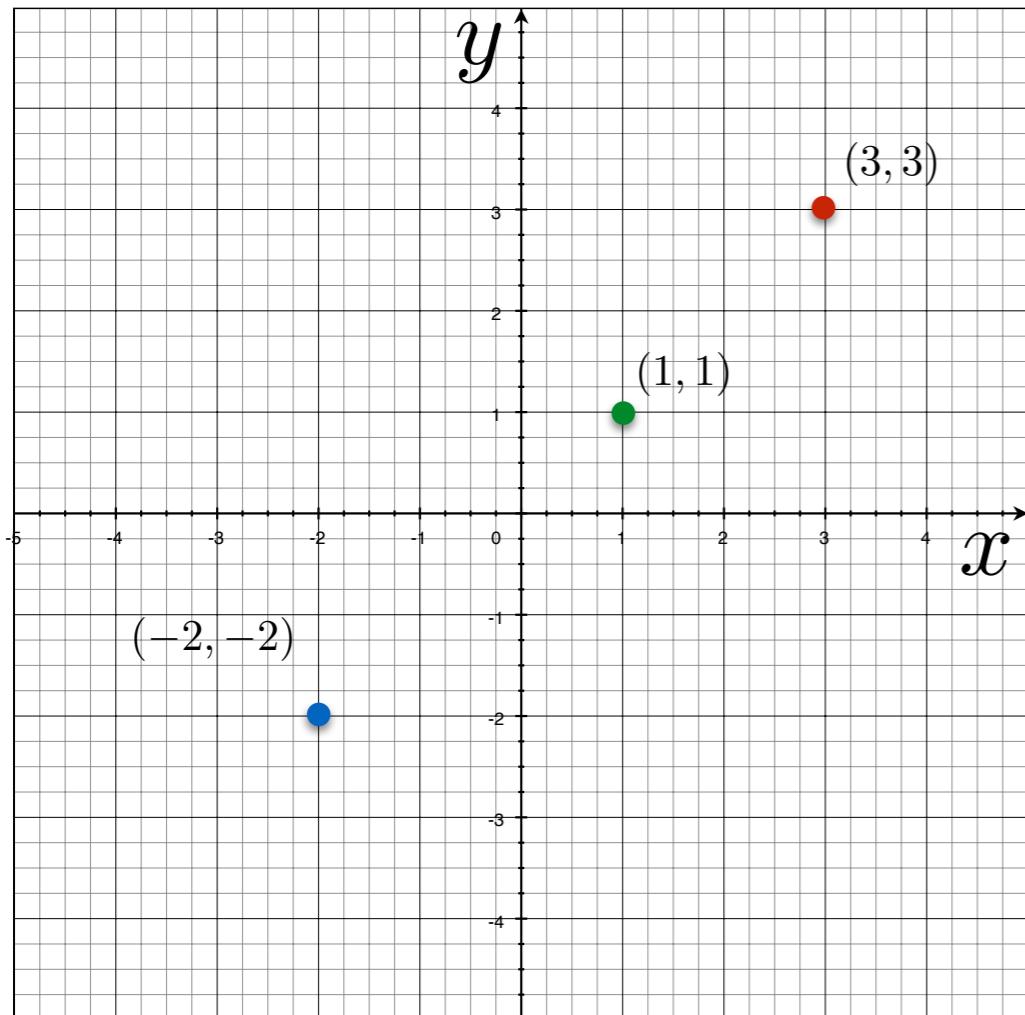


Image space

Parameter space

# Image and parameter space

variables  
 $y = mx + b$   
parameters



three points  
become  
?

variables  
 $y - mx = b$   
parameters

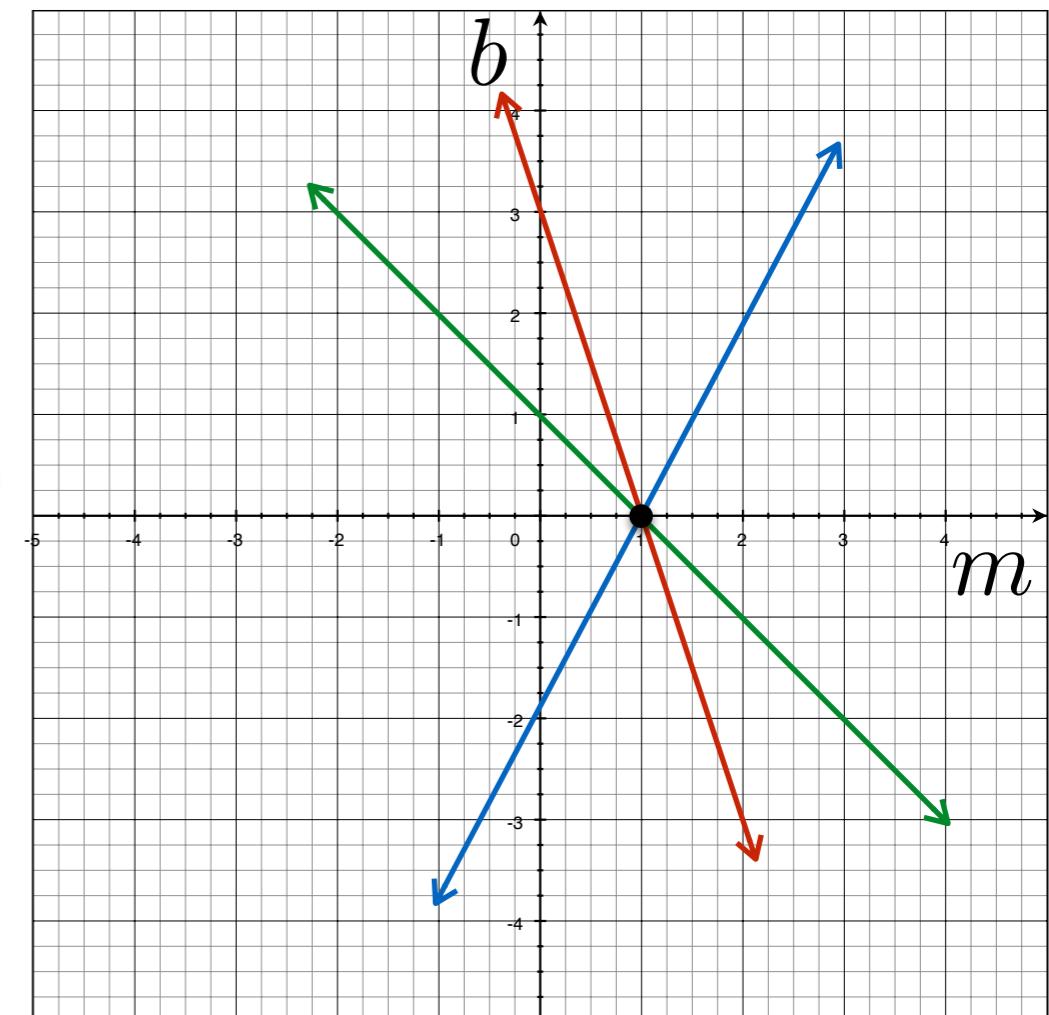
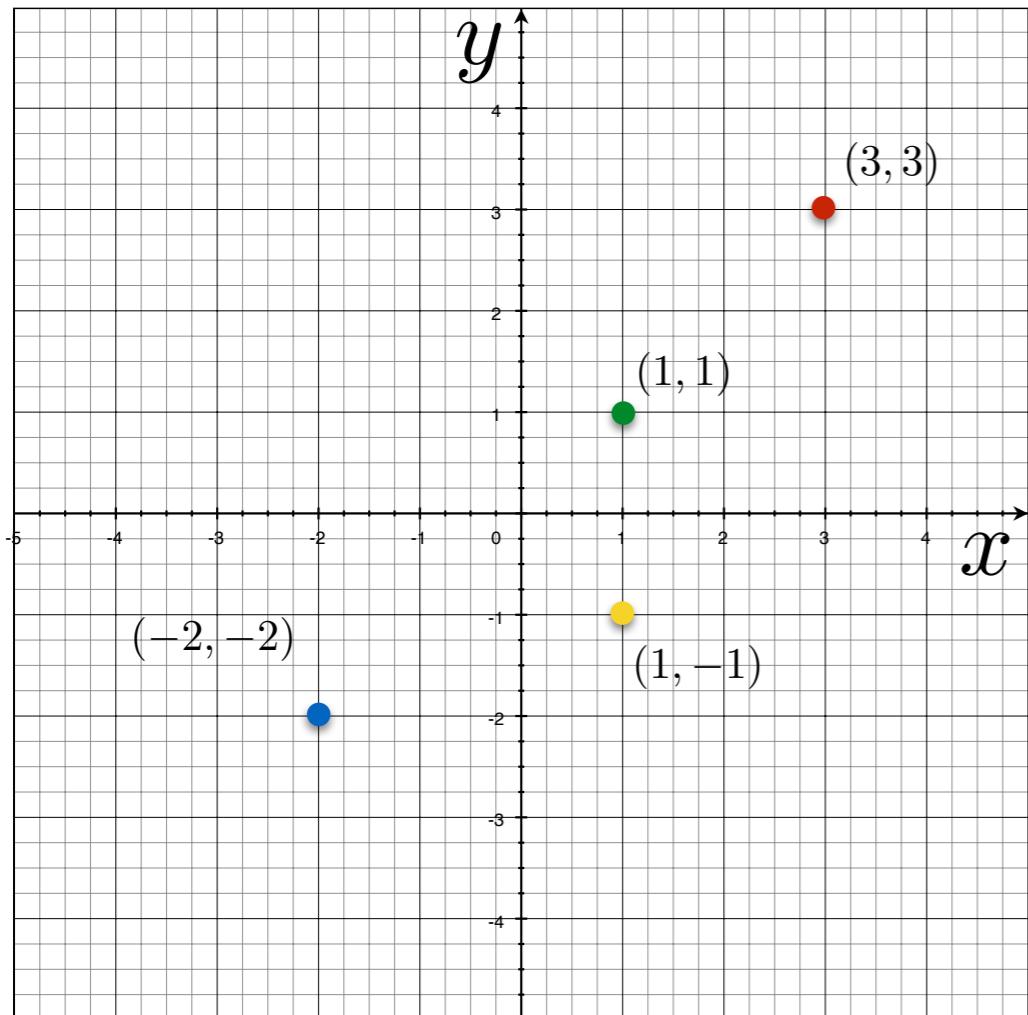


Image space

Parameter space

# Image and parameter space

variables  
 $y = mx + b$   
parameters



four points  
become  
?

variables  
 $y - mx = b$   
parameters

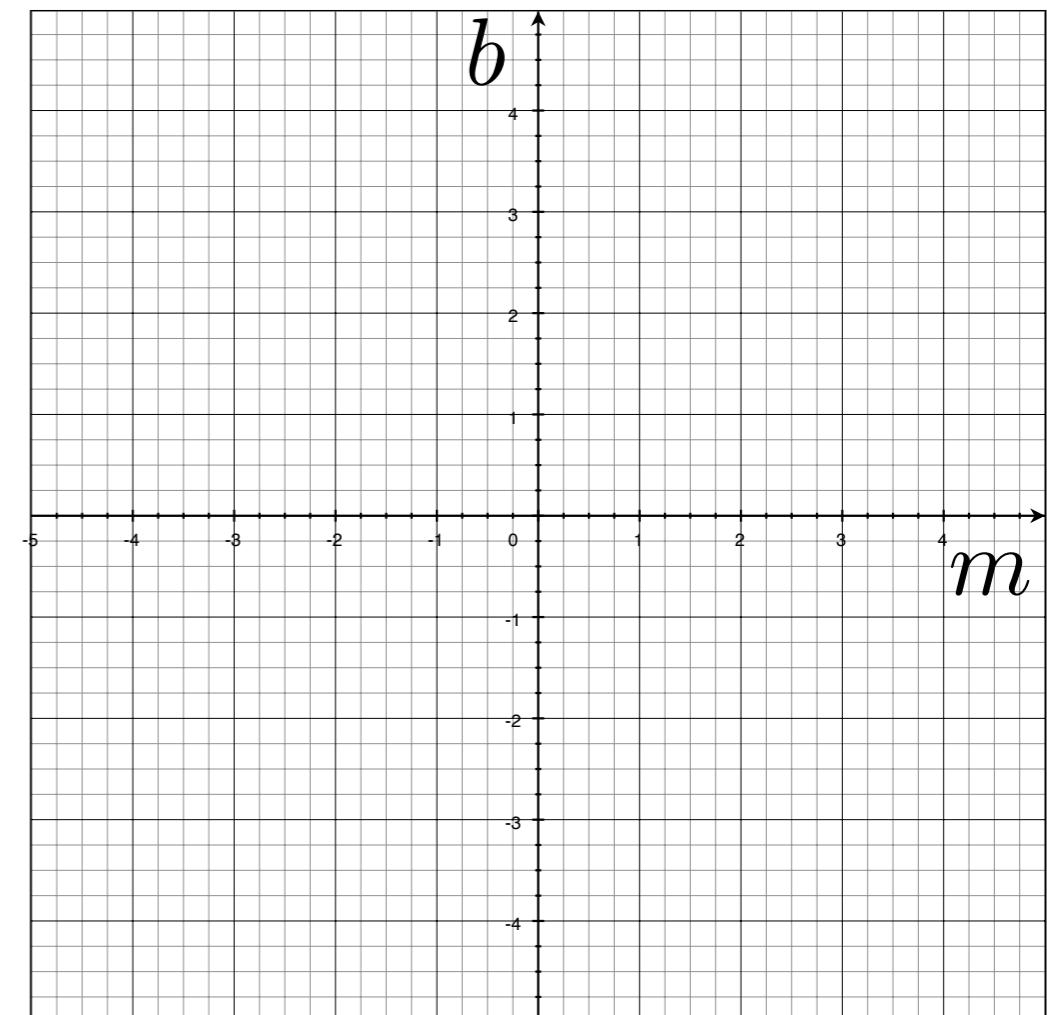
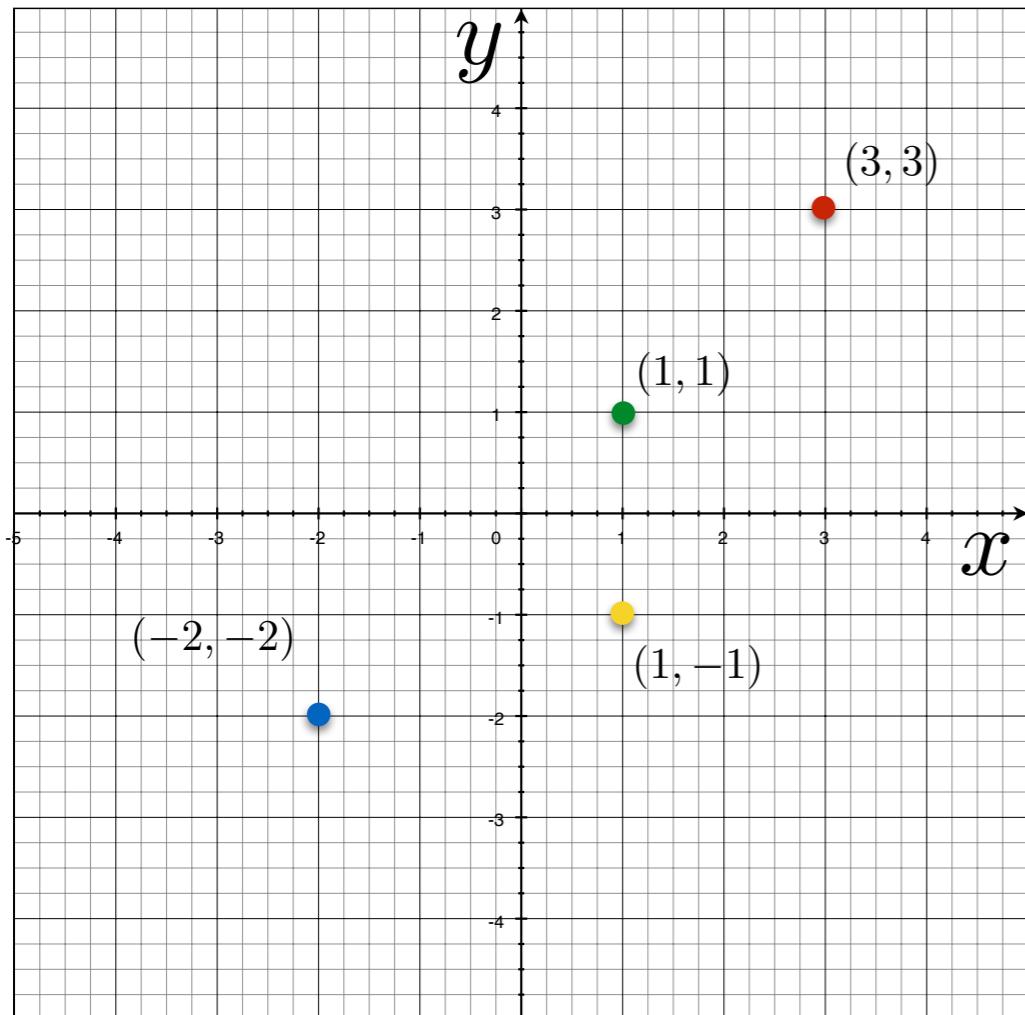


Image space

Parameter space

# Image and parameter space

variables  
 $y = mx + b$   
parameters



four points  
become  
?

variables  
 $y - mx = b$   
parameters

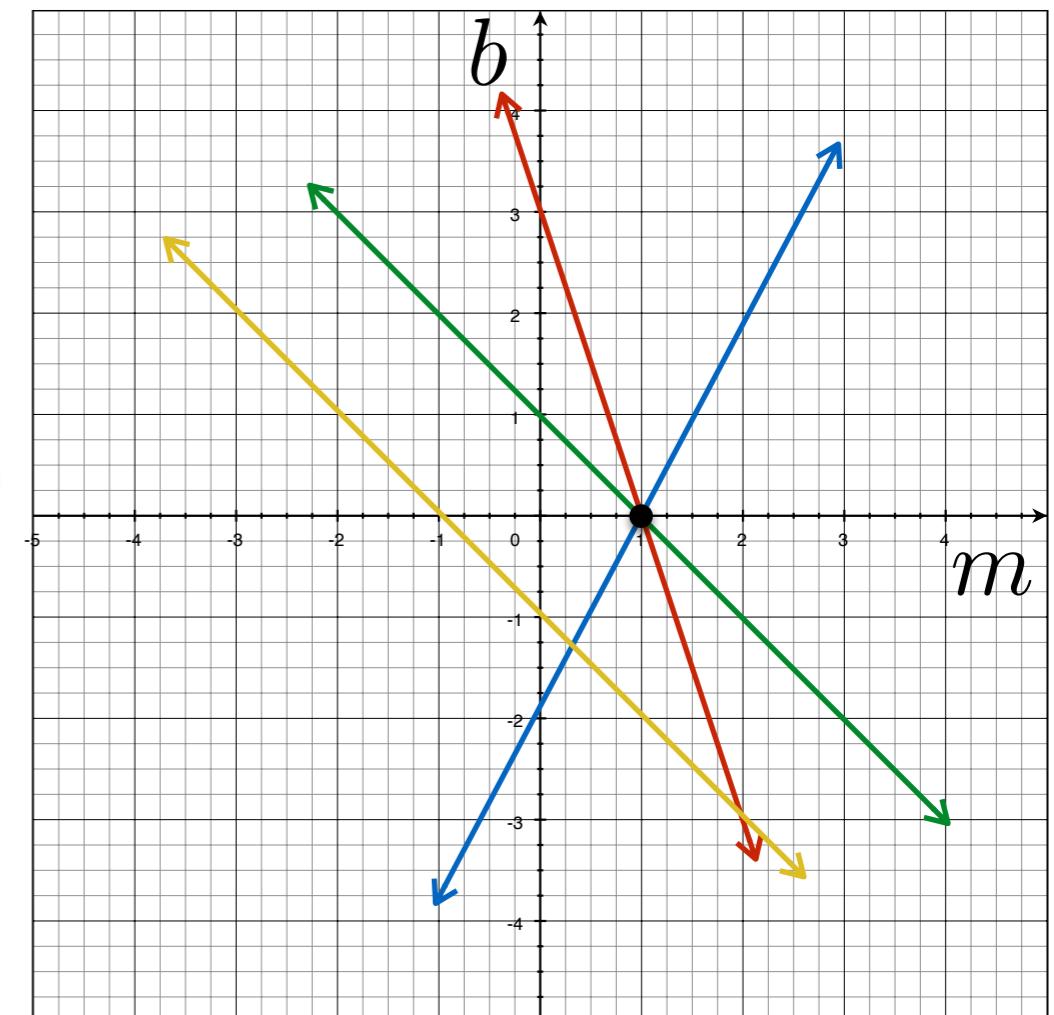


Image space

Parameter space

# How would you find the best fitting line?

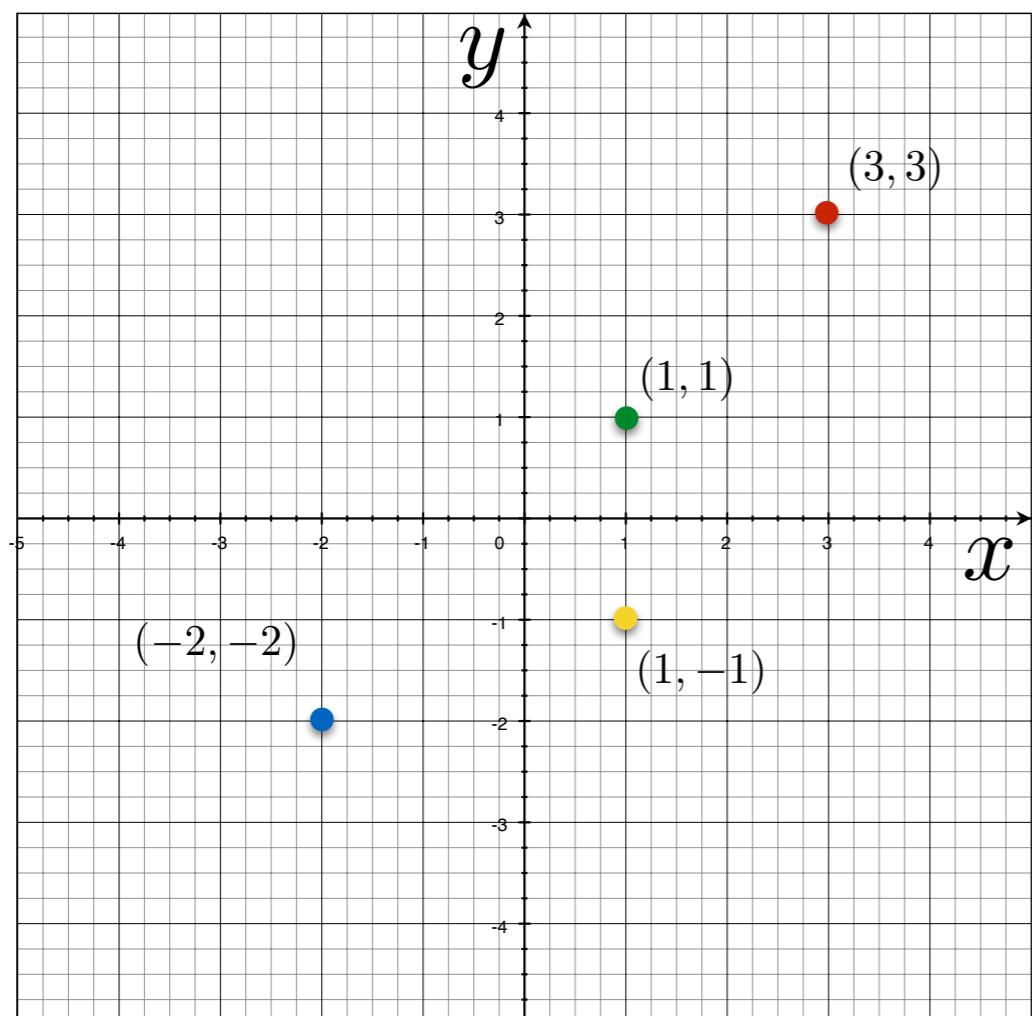
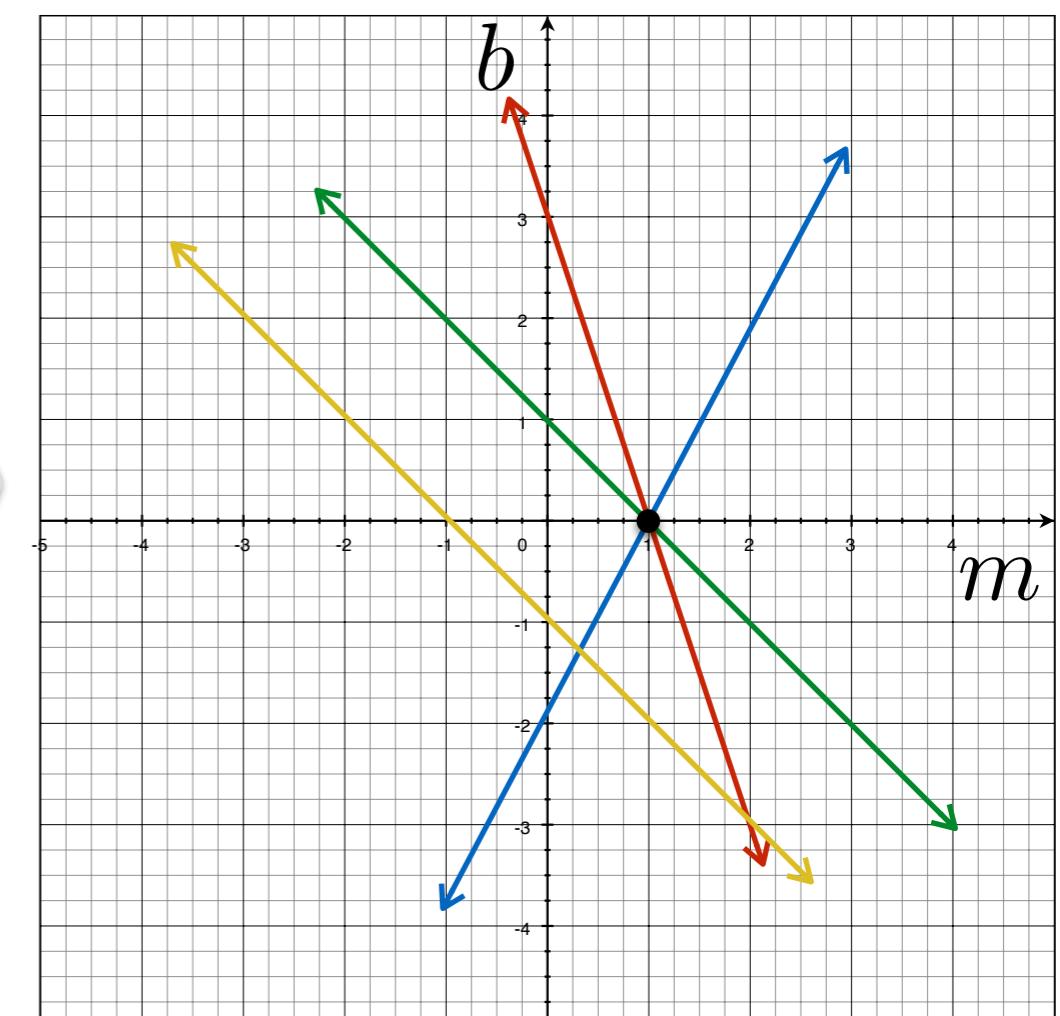


Image space



Parameter space

*Is your method robust to measurement noise?*

*Is your 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  $A(m, c) = 0 \quad \forall m, c$

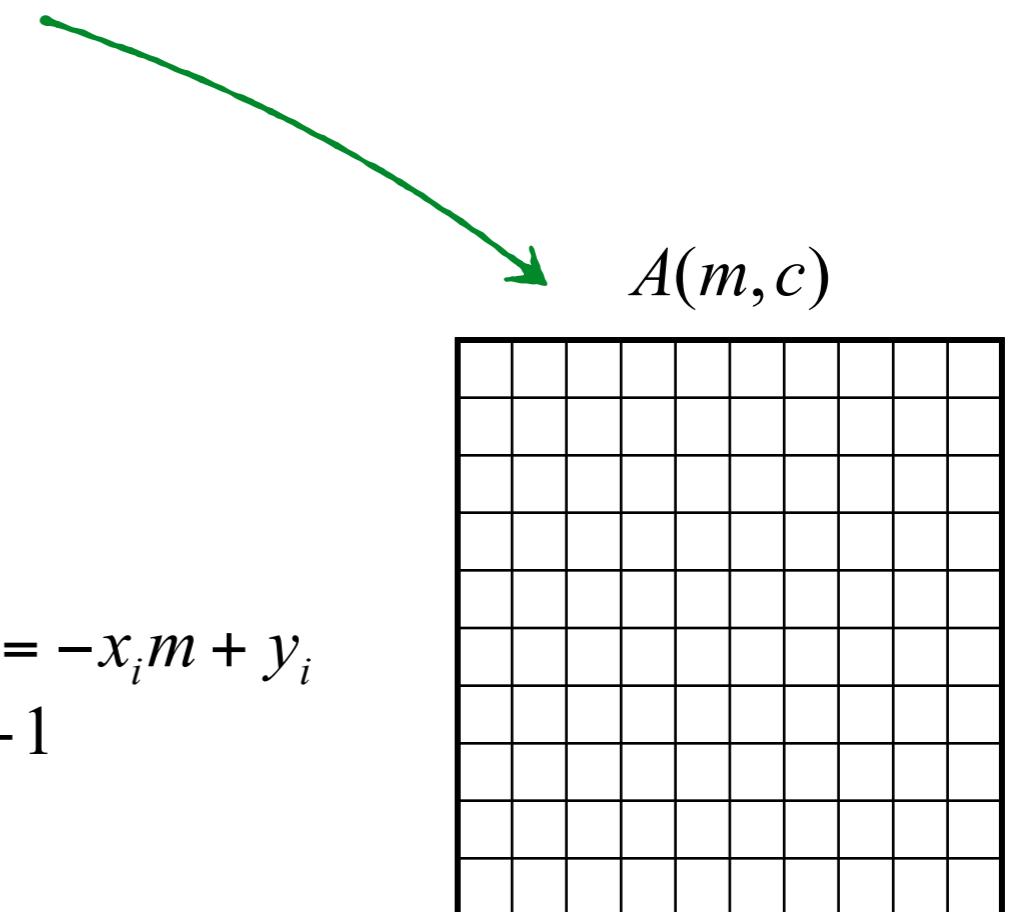
4. For each image edge  $(x_i, y_i)$

    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  $A(m, c)$



# Line Detection by Hough Transform

## Algorithm:

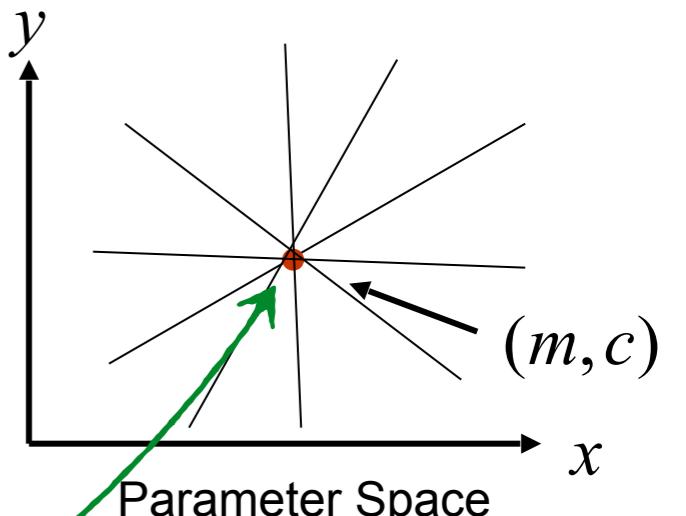
1. Quantize Parameter Space  $(m, c)$

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  $A(m, c)$

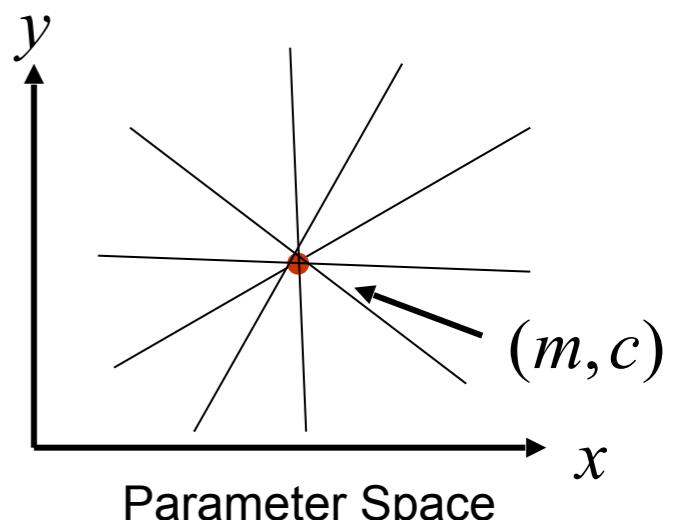


1				1	
	1			1	
		1	1		
			2		
		1	1		
	1			1	
1				1	

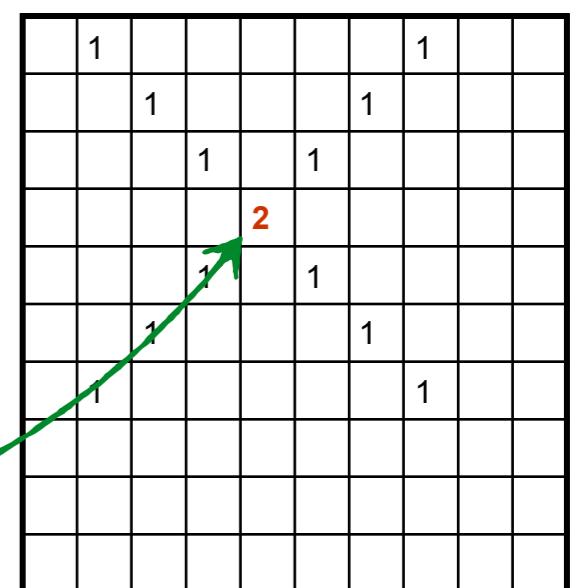
# Line Detection by Hough Transform

## 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  $A(m, c)$



# Problems with parameterization

*What's wrong with the parameterization  $(m, c)$  ?*

## *How big does the accumulator need to be?*

$$A(m,c)$$

# Problems with parameterization

*What's wrong with the parameterization  $(m, c)$  ?*

*How big does the accumulator need to be?*

$$A(m,c)$$

The space of  $m$  is huge!      The space of  $c$  is huge!

# Problems with parameterization

*What's wrong with the parameterization  $(m, c)$  ?*

*How big does the accumulator need to be?*

$$A(m,c)$$

The space of  $m$  is huge!    The space of  $c$  is huge!

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

# A better parameterization...

Use normal form:

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

# A better parameterization...

Use normal form:

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

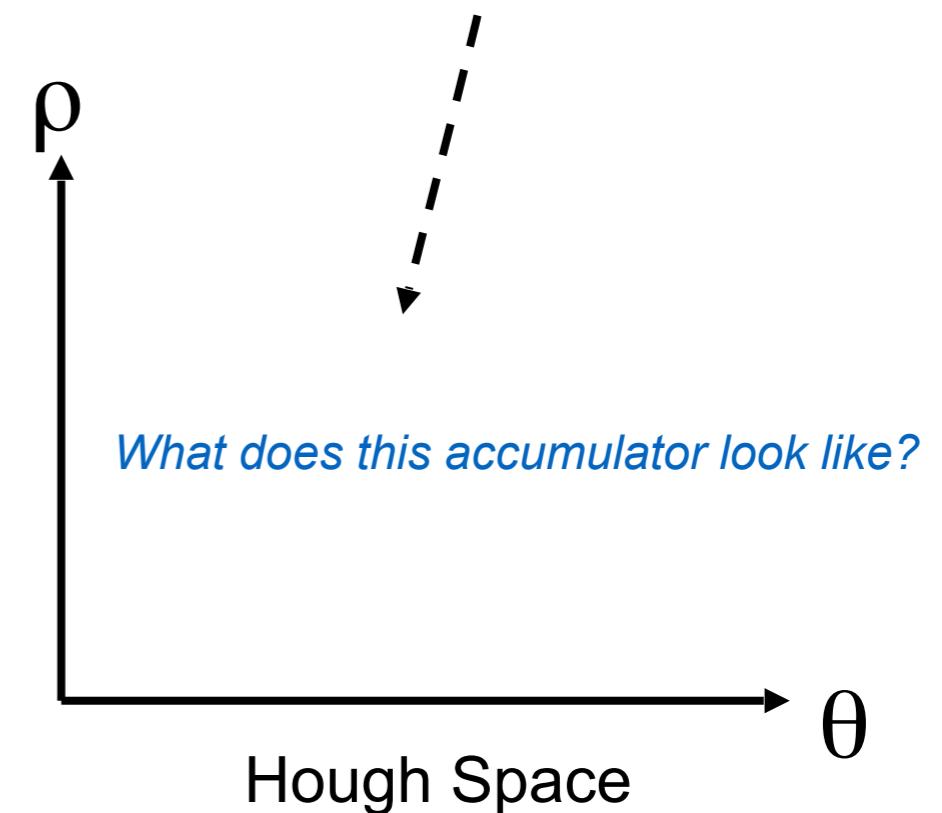
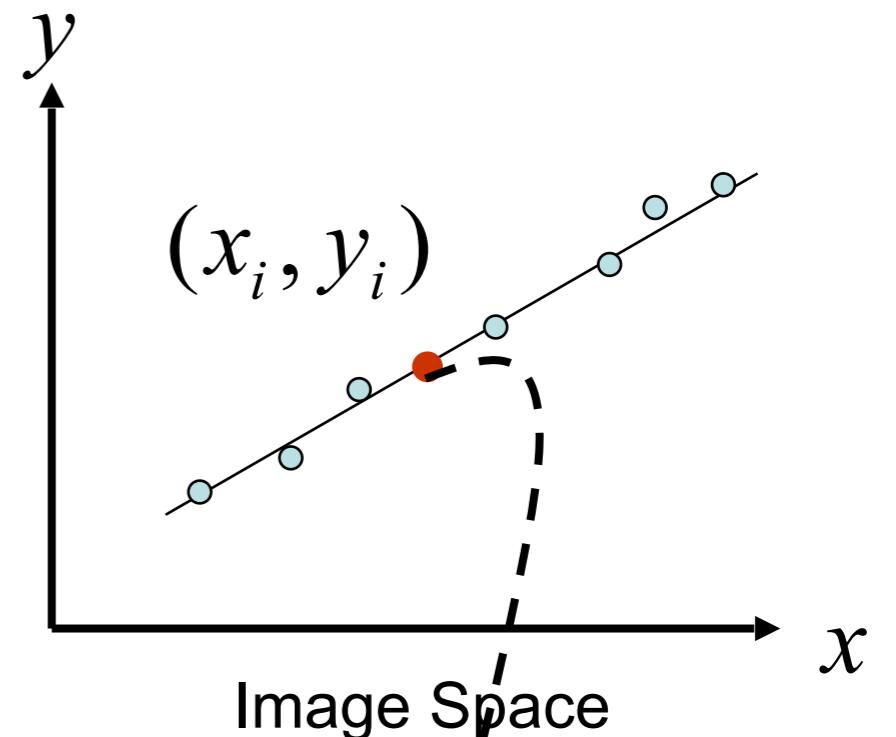
Given points  $(x_i, y_i)$  find  $(\rho, \theta)$

Hough Space Sinusoid

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

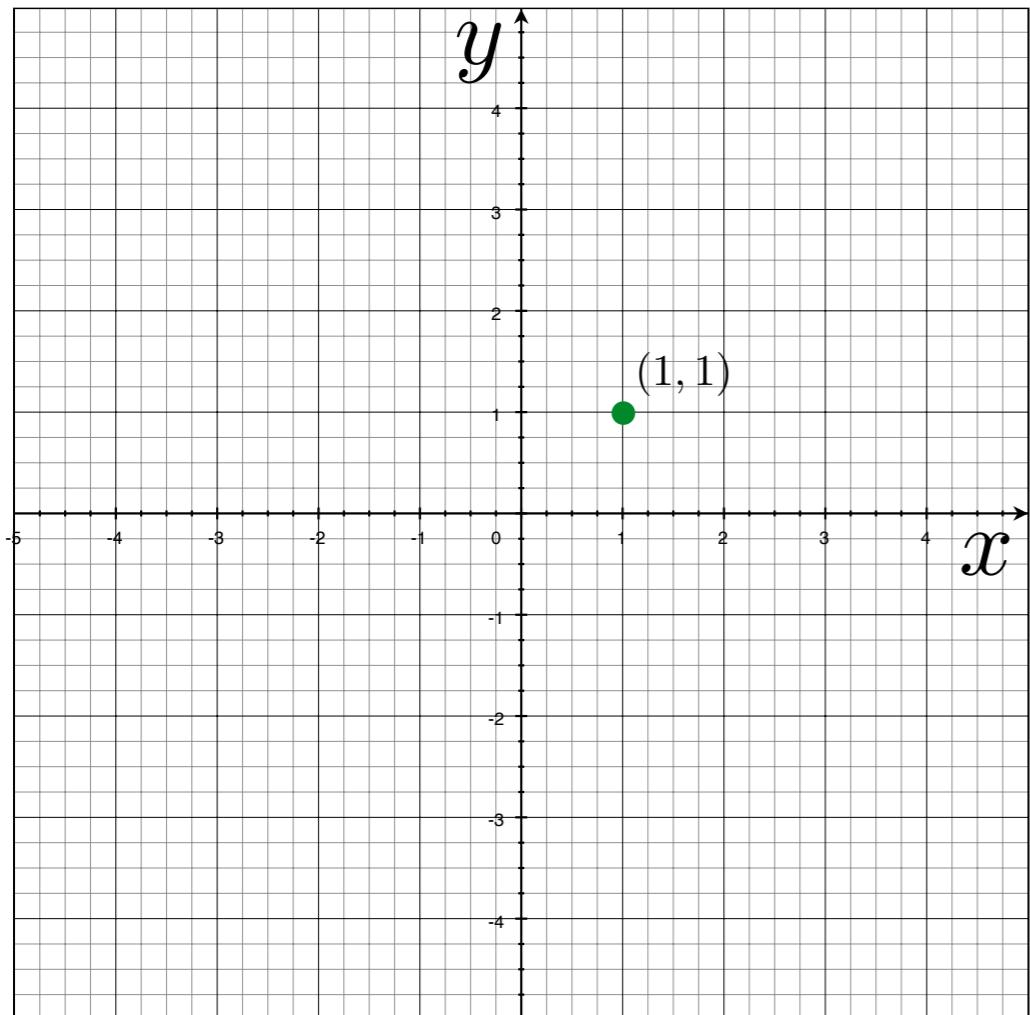
$$0 \leq \rho \leq \rho_{\max}$$

(Finite Accumulator Array Size)



# Image and parameter space

variables  
 $y = mx + b$   
parameters



a point becomes a wave

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

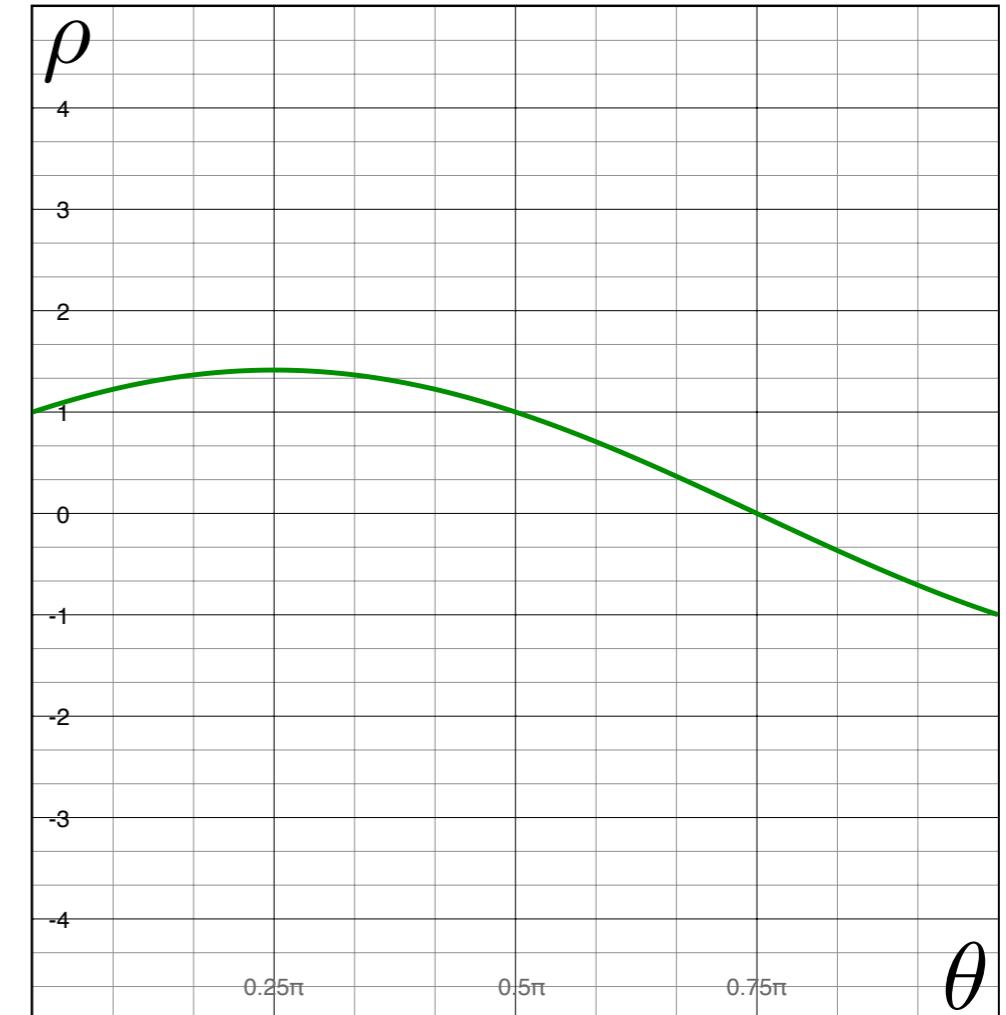


Image space

Parameter space

# Image and parameter space

variables  
 $y = mx + b$   
parameters

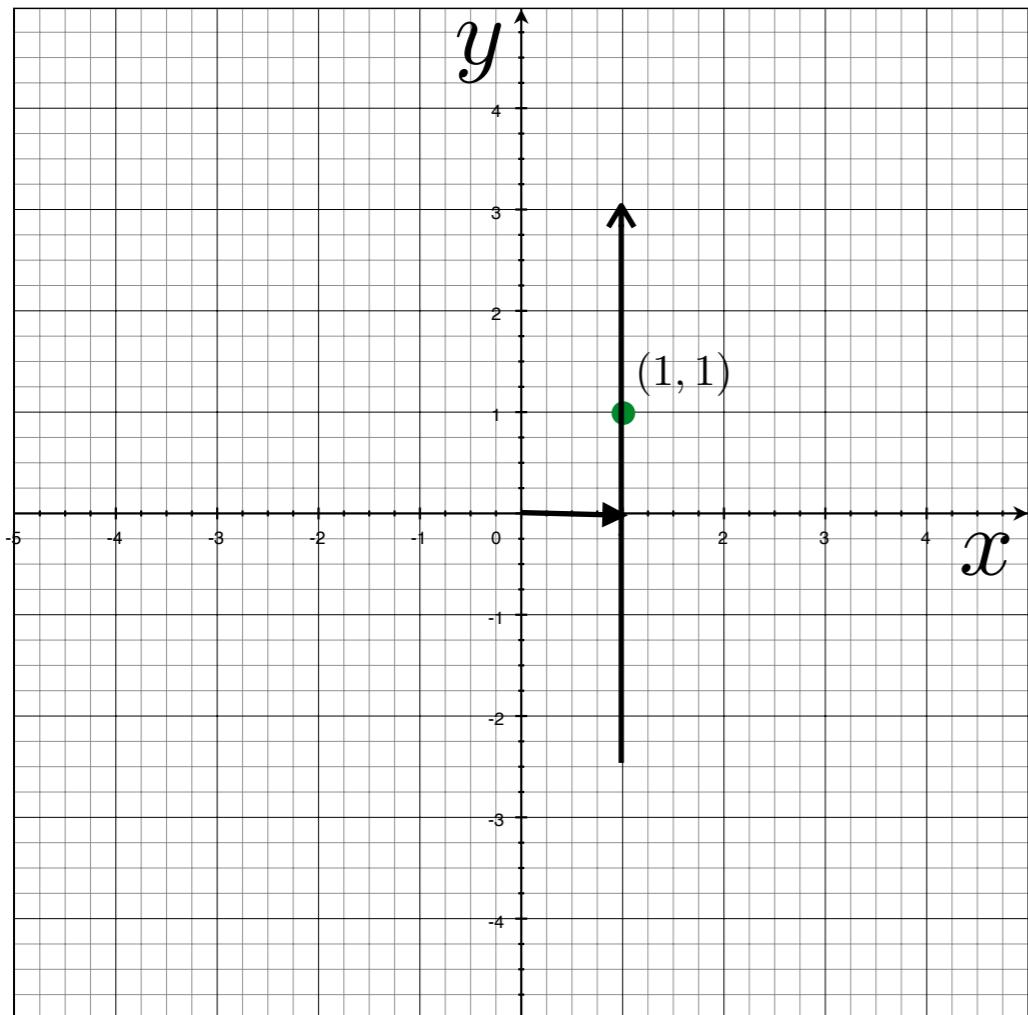
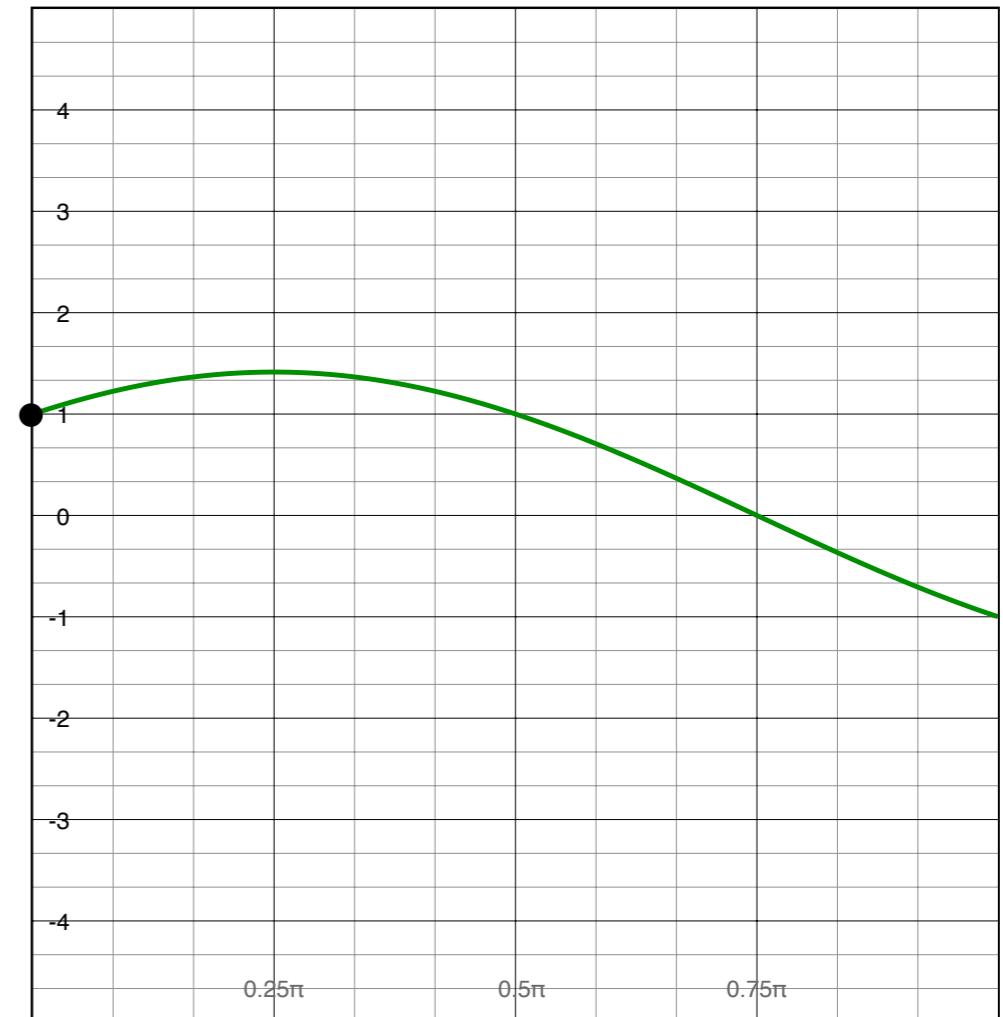


Image space

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

a line becomes a point



Parameter space

# Image and parameter space

variables  
 $y = mx + b$   
parameters

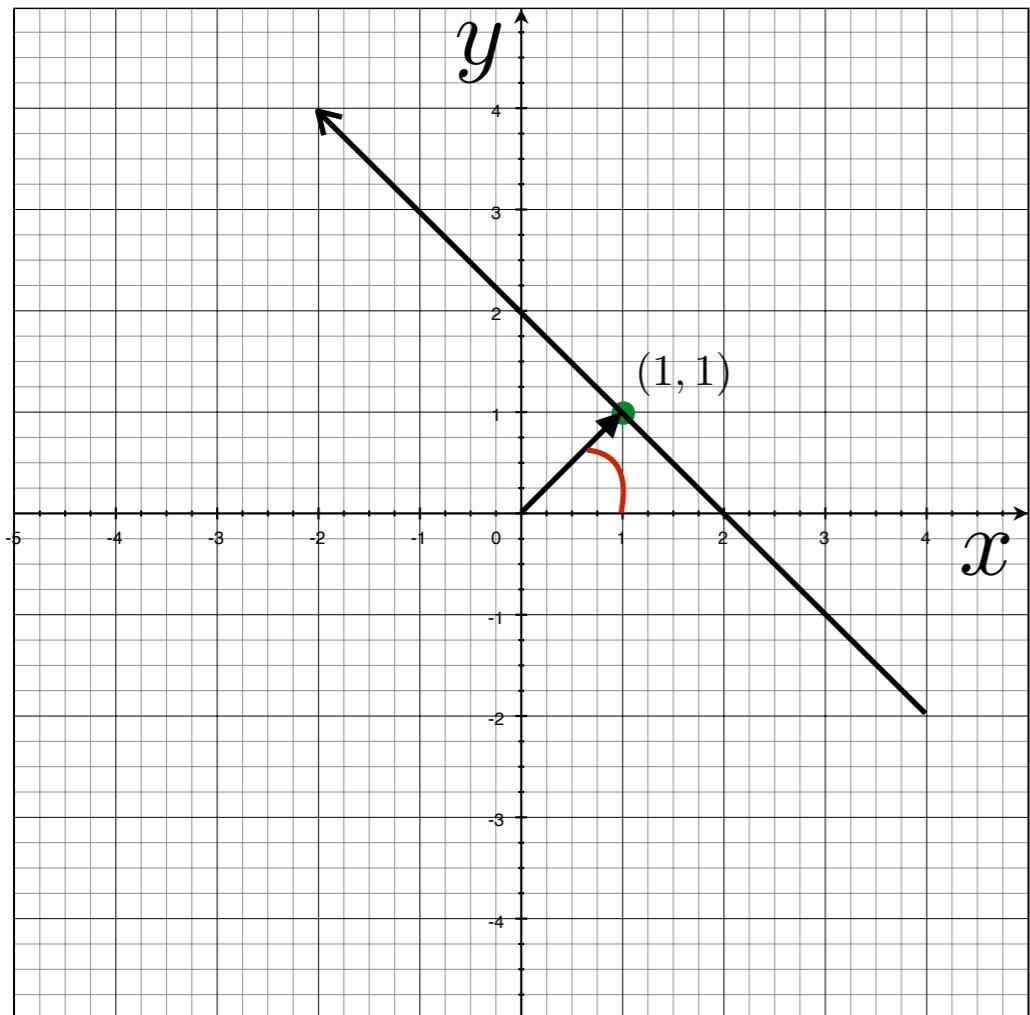
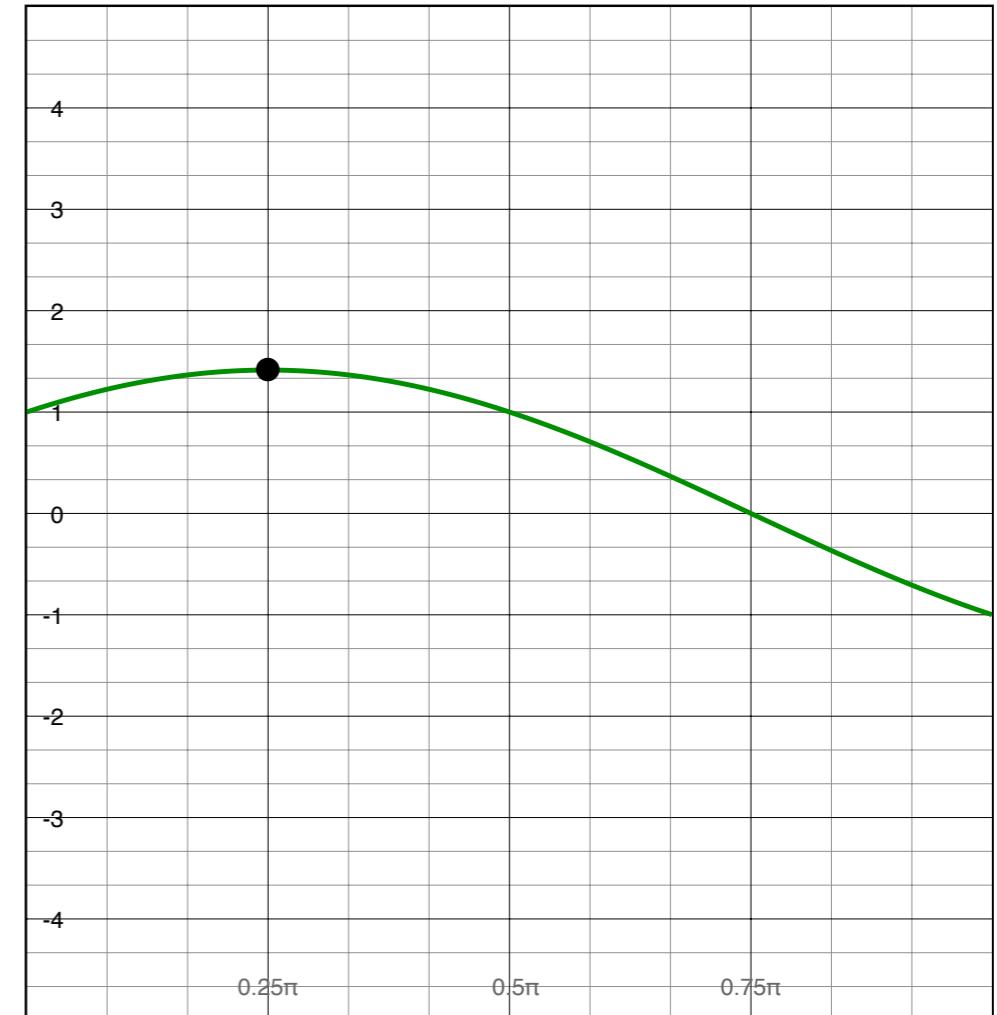


Image space

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

a line becomes a point



Parameter space

# Image and parameter space

variables  
 $y = mx + b$   
parameters

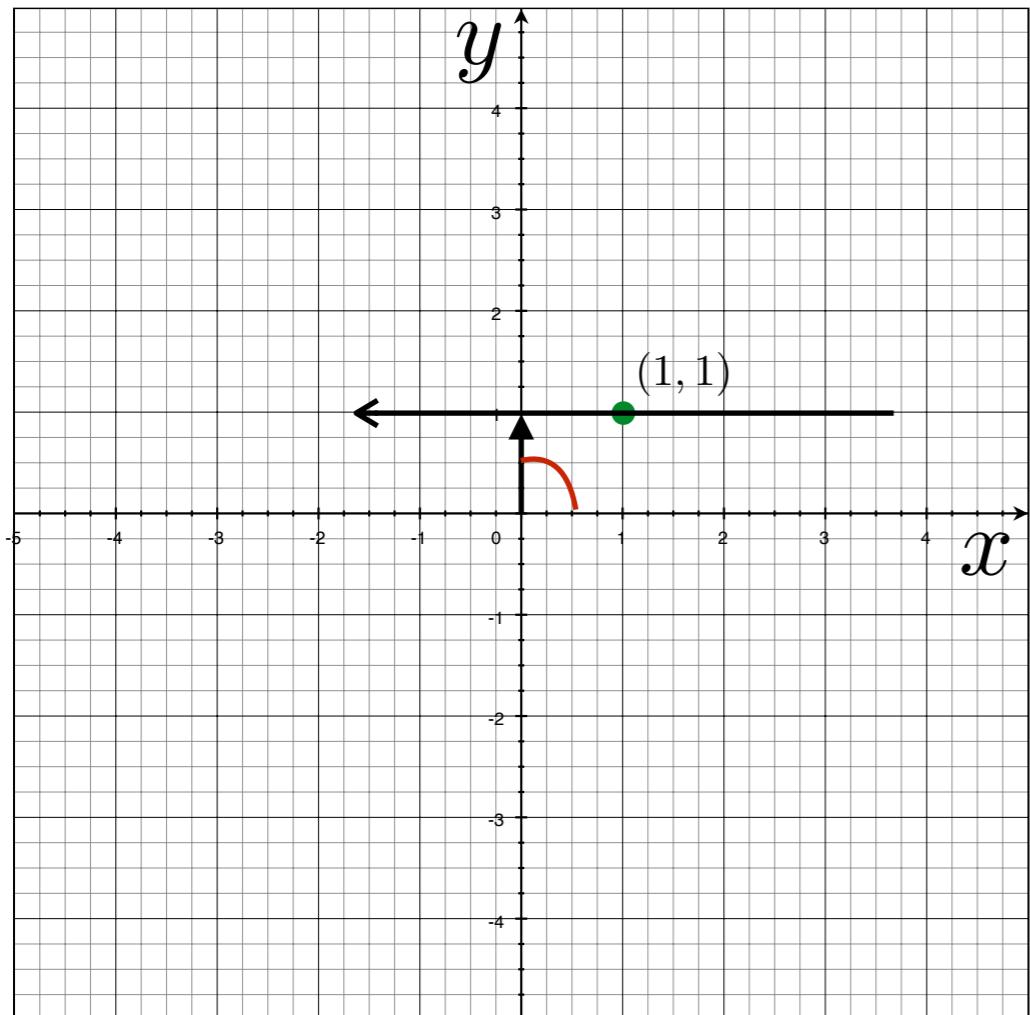
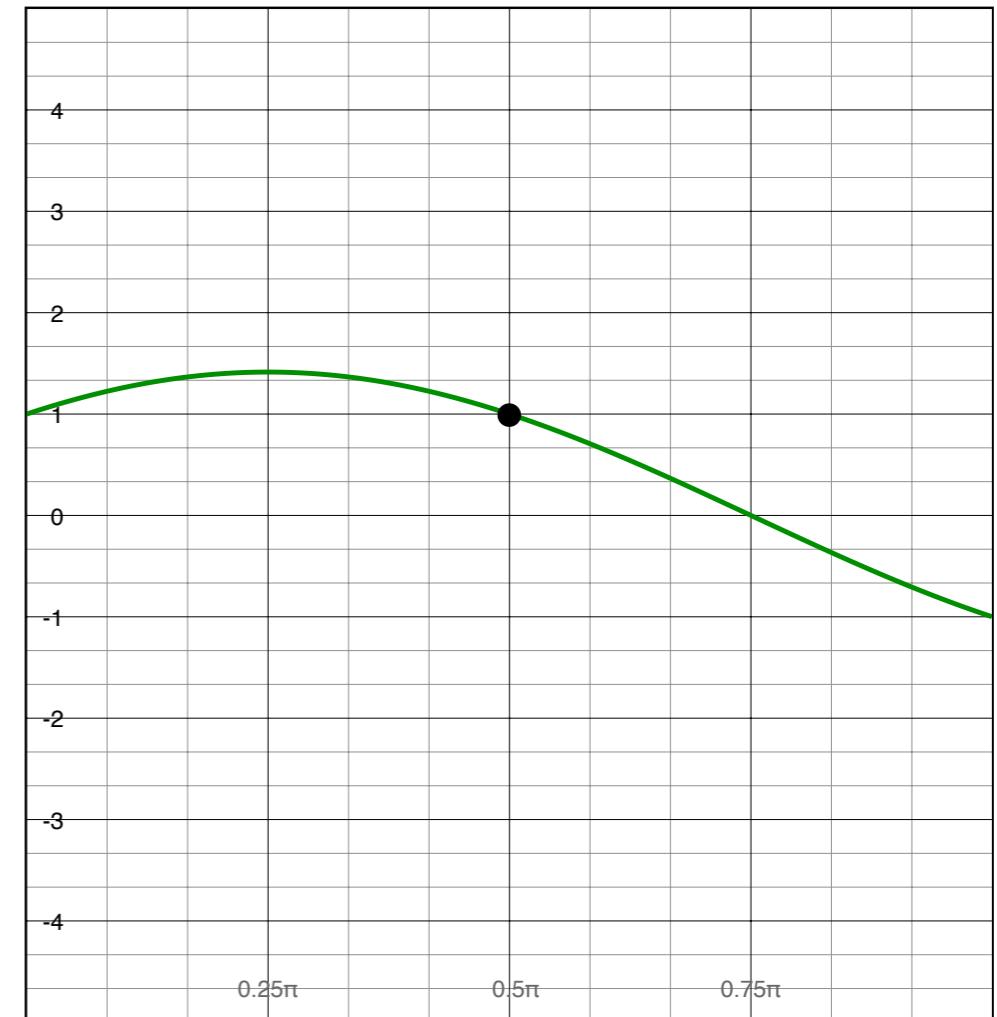


Image space

a line becomes a point

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



Parameter space

# Image and parameter space

variables  
 $y = mx + b$   
parameters

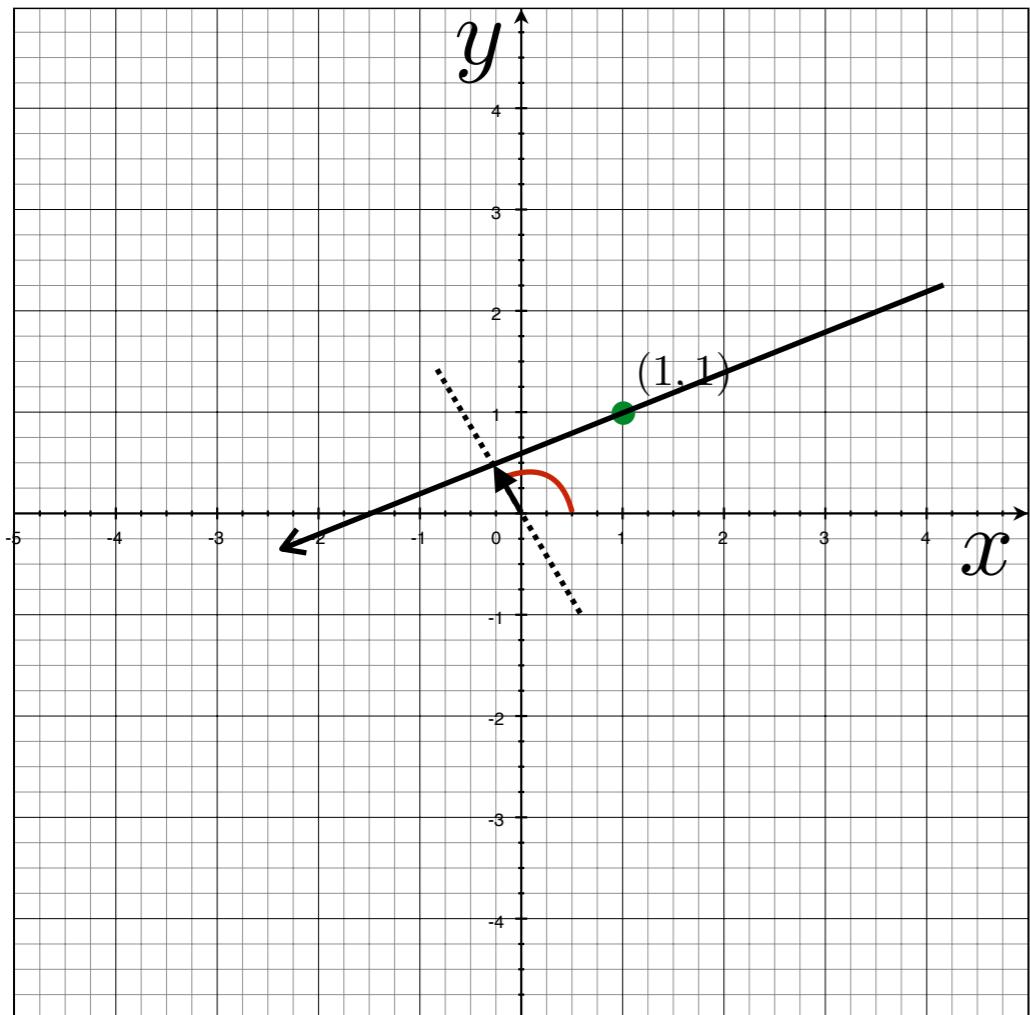
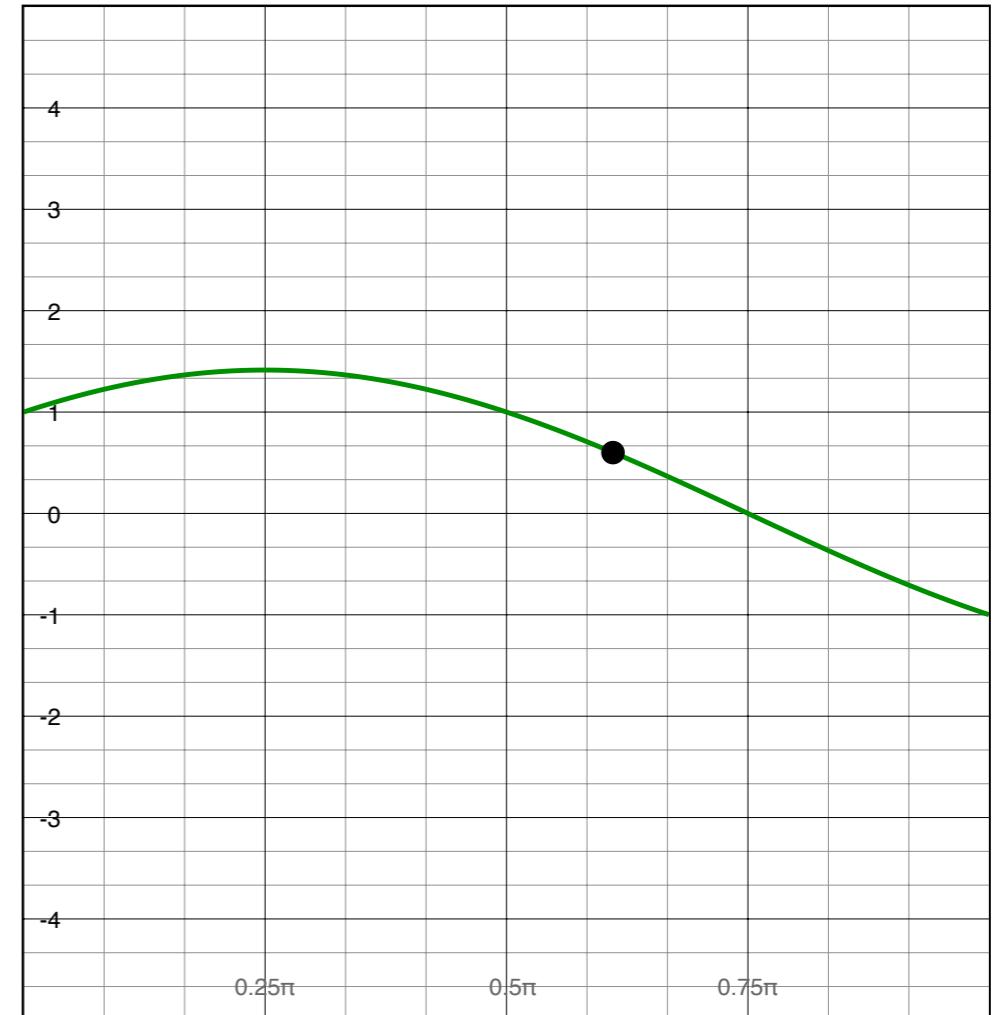


Image space

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

a line becomes a point



Parameter space

# Image and parameter space

variables  
 $y = mx + b$   
parameters

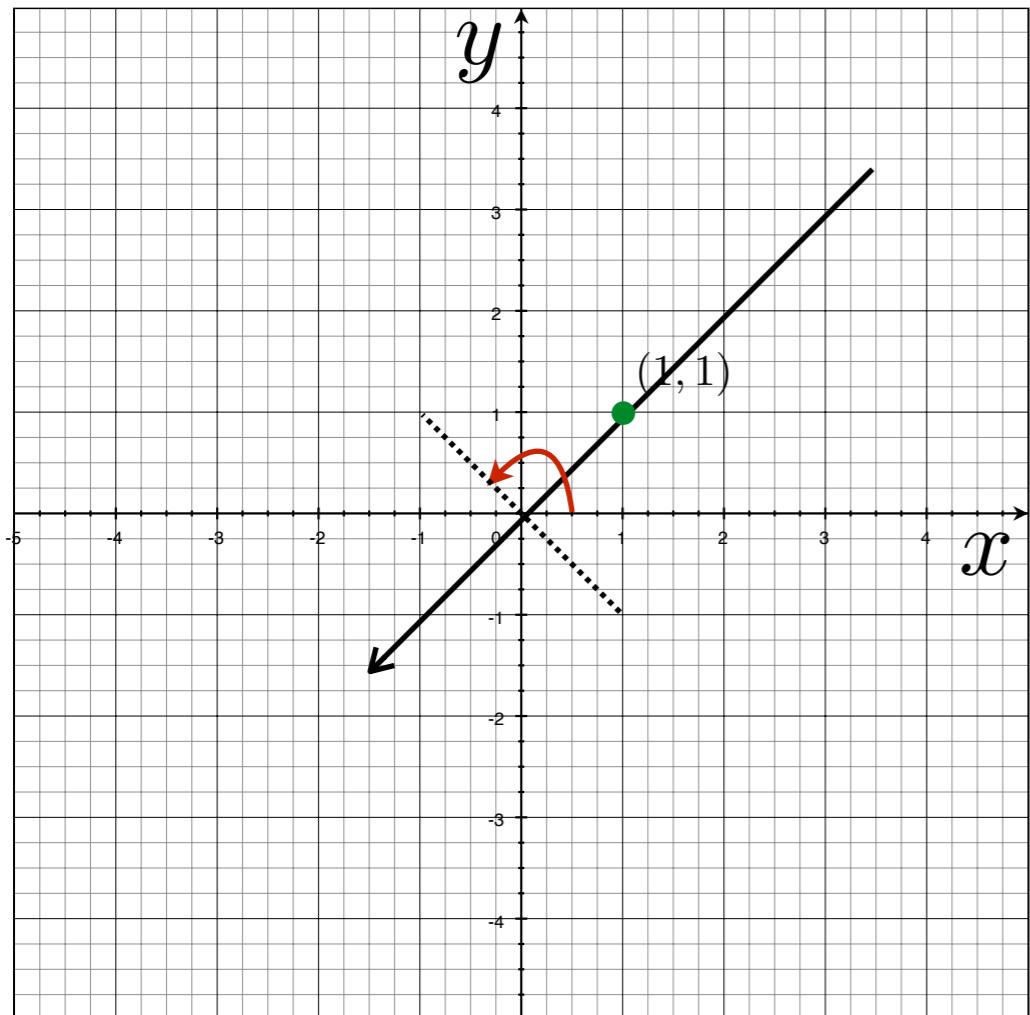
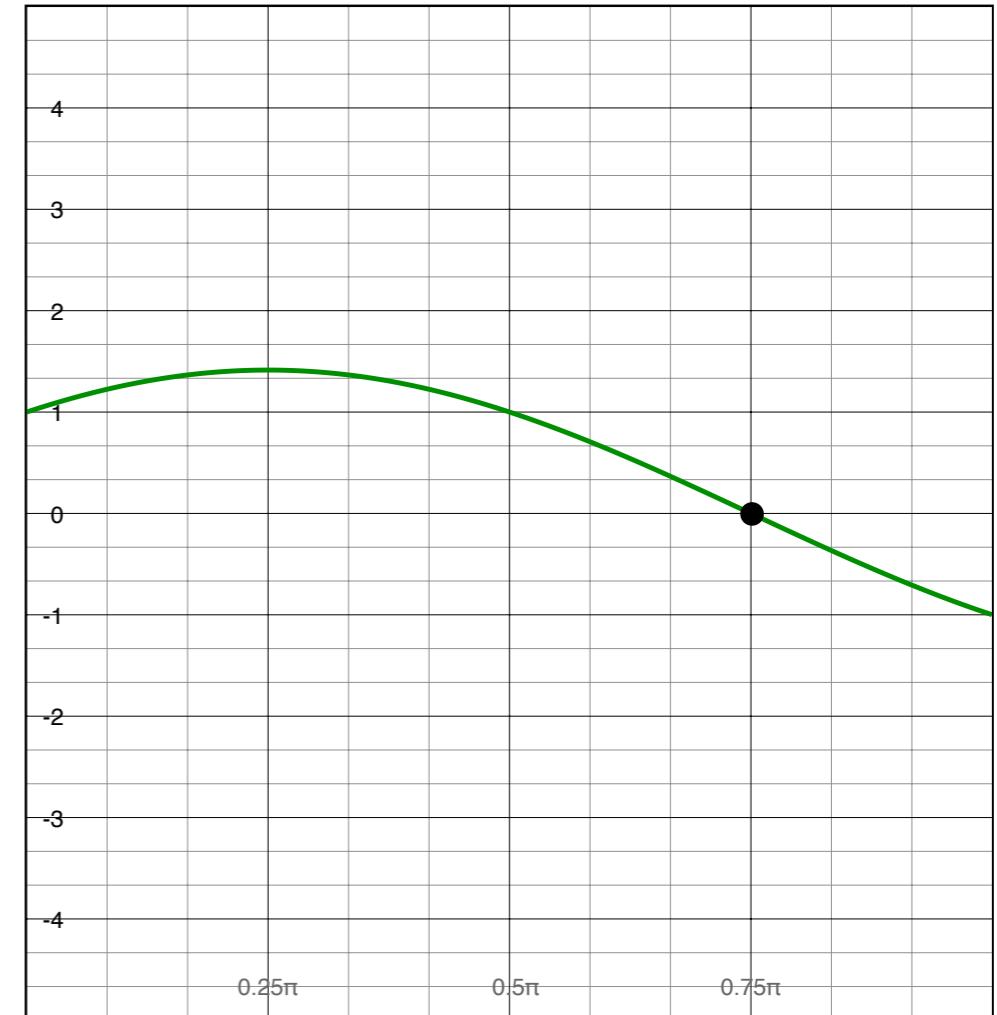


Image space

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

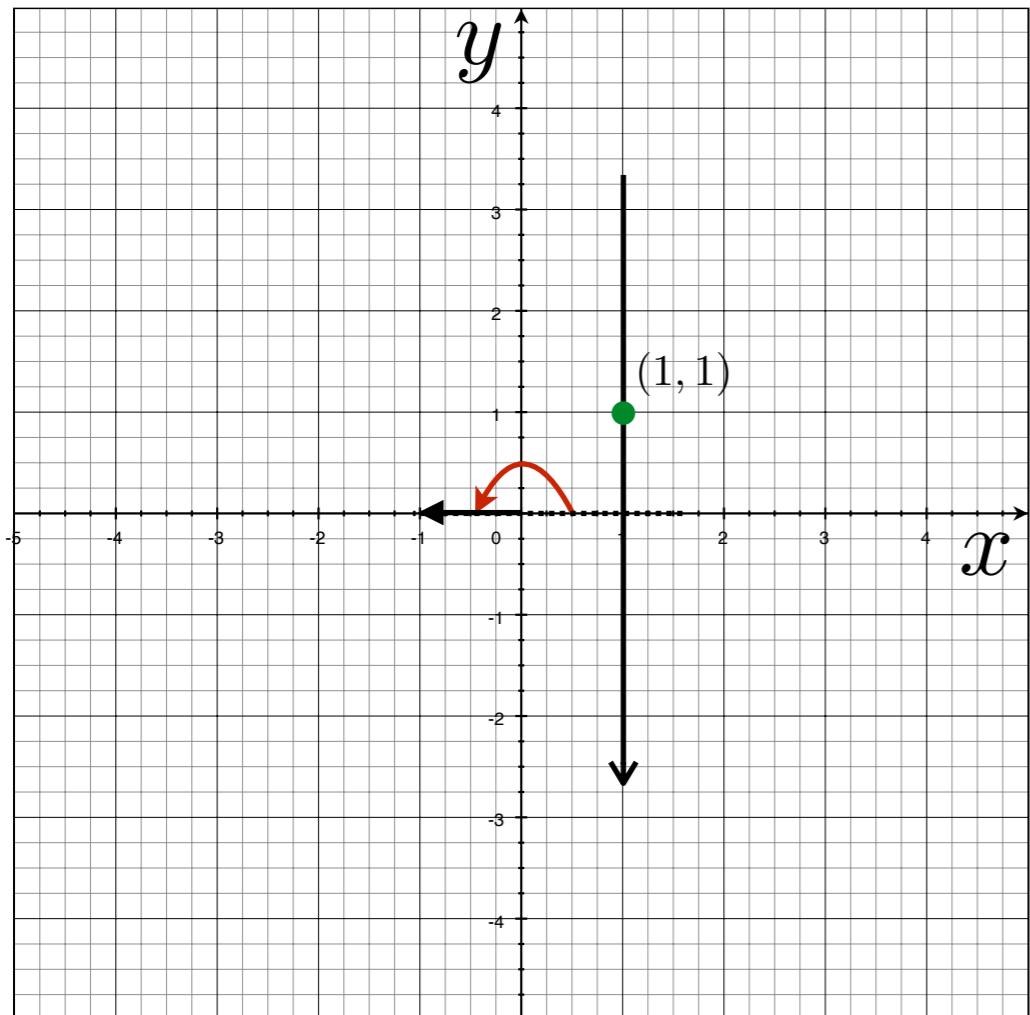
a line becomes a point



Parameter space

# Image and parameter space

variables  
 $y = mx + b$   
parameters



a line becomes a point

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

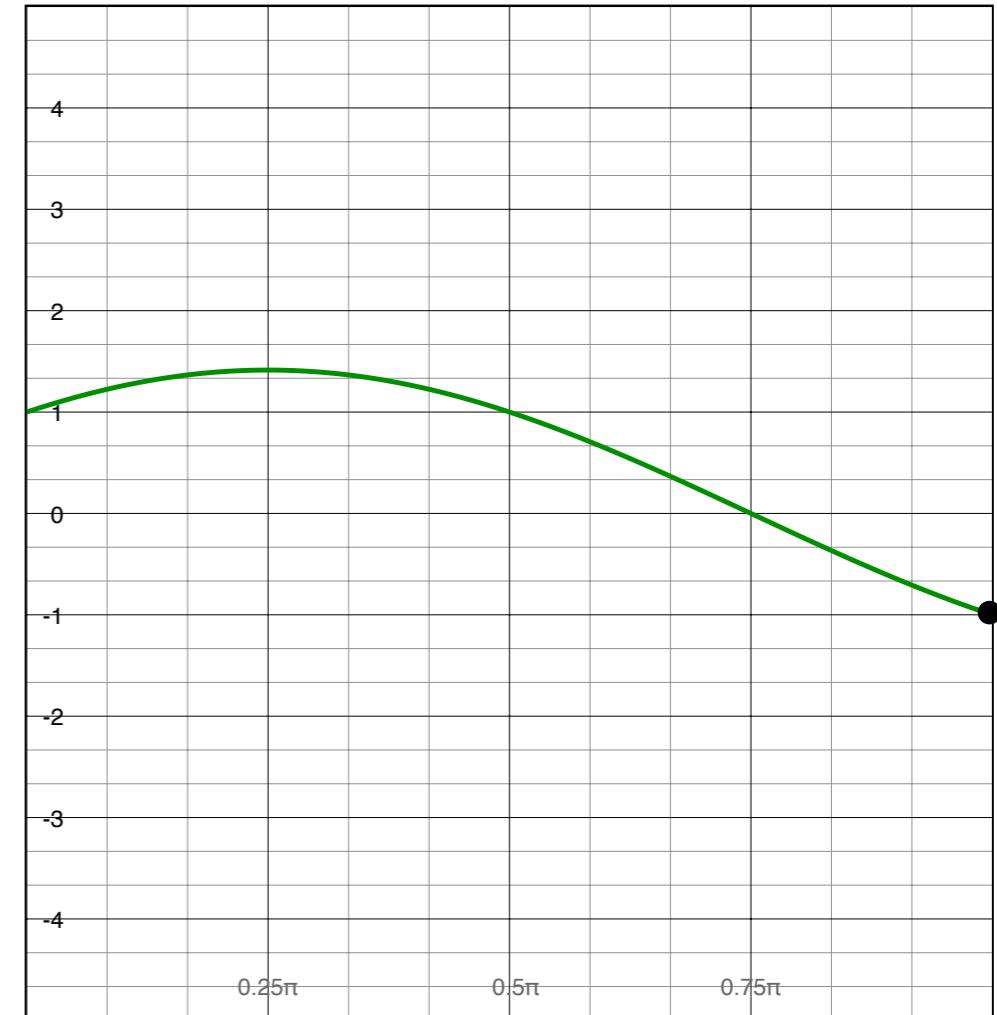
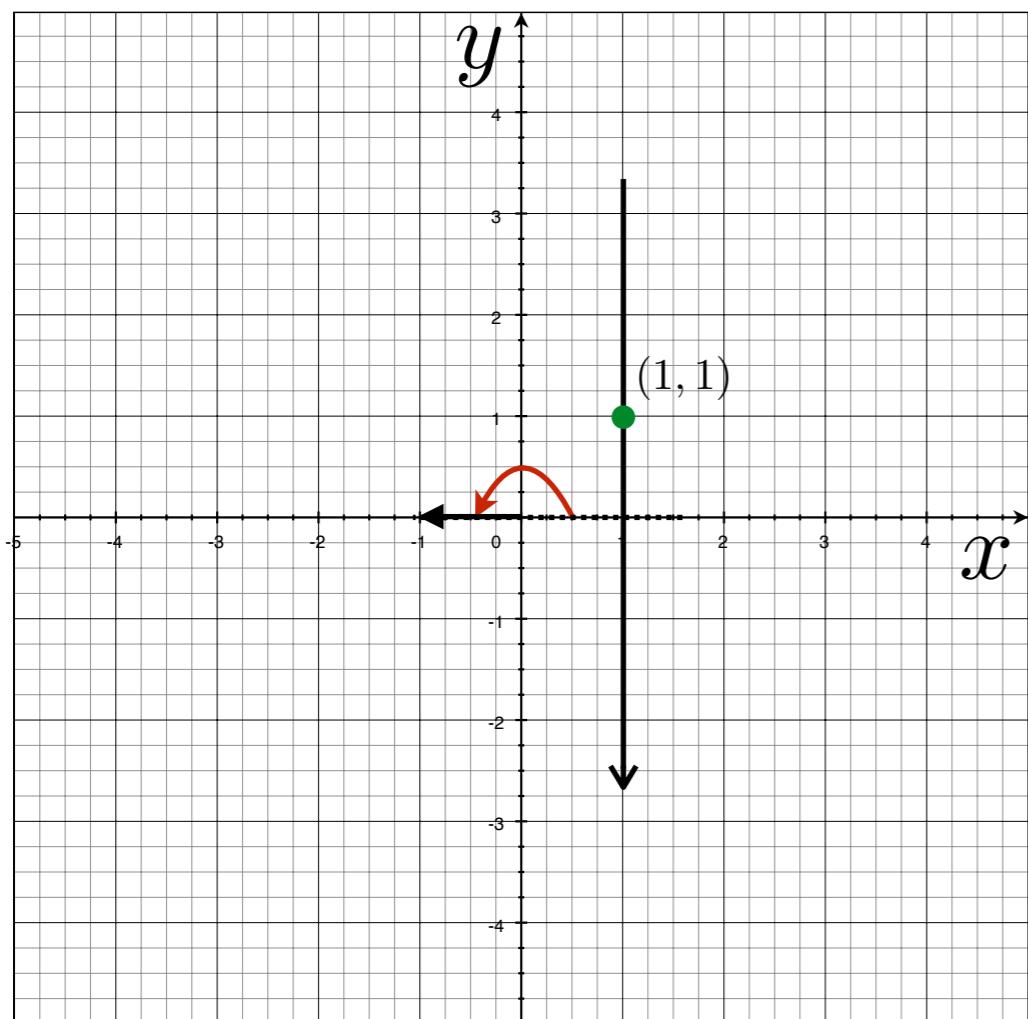


Image space

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?

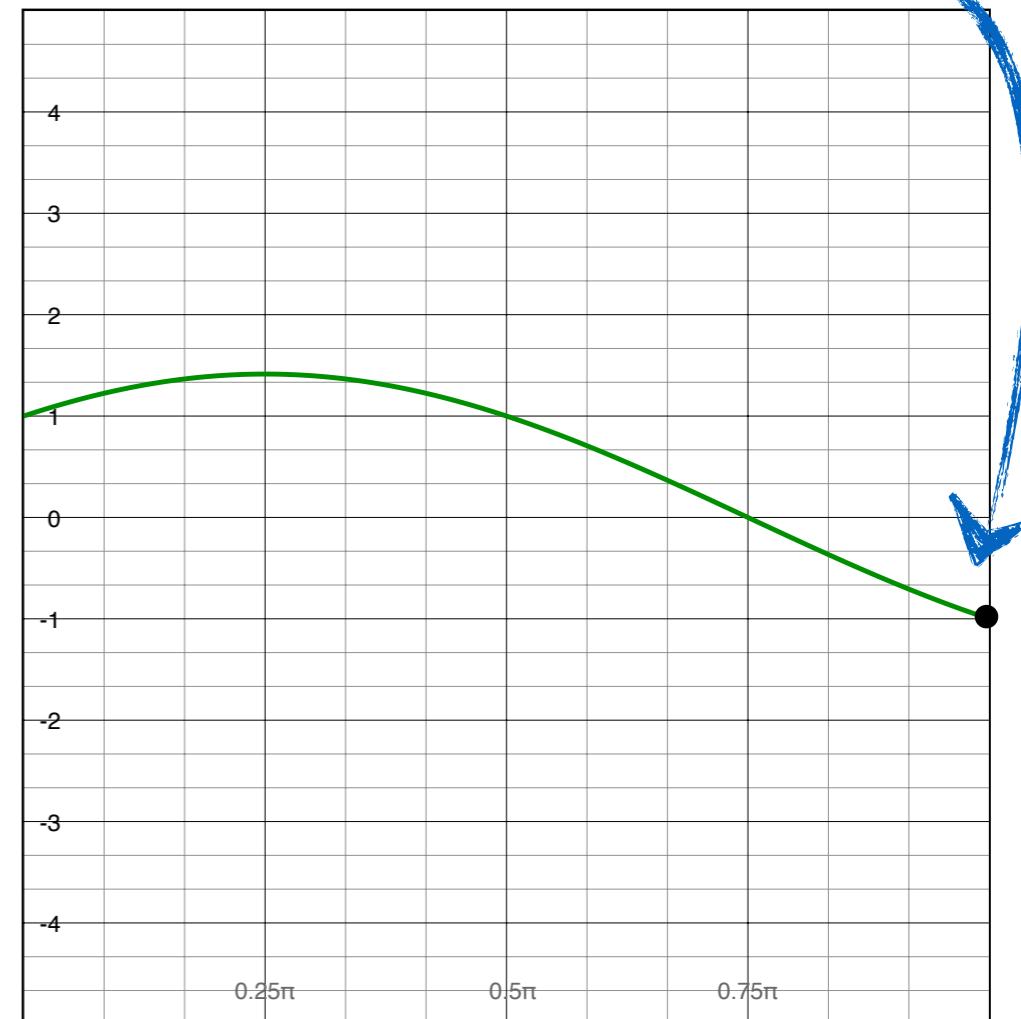


Image space

Parameter space

# Image and parameter space

variables  
 $y = mx + b$   
parameters

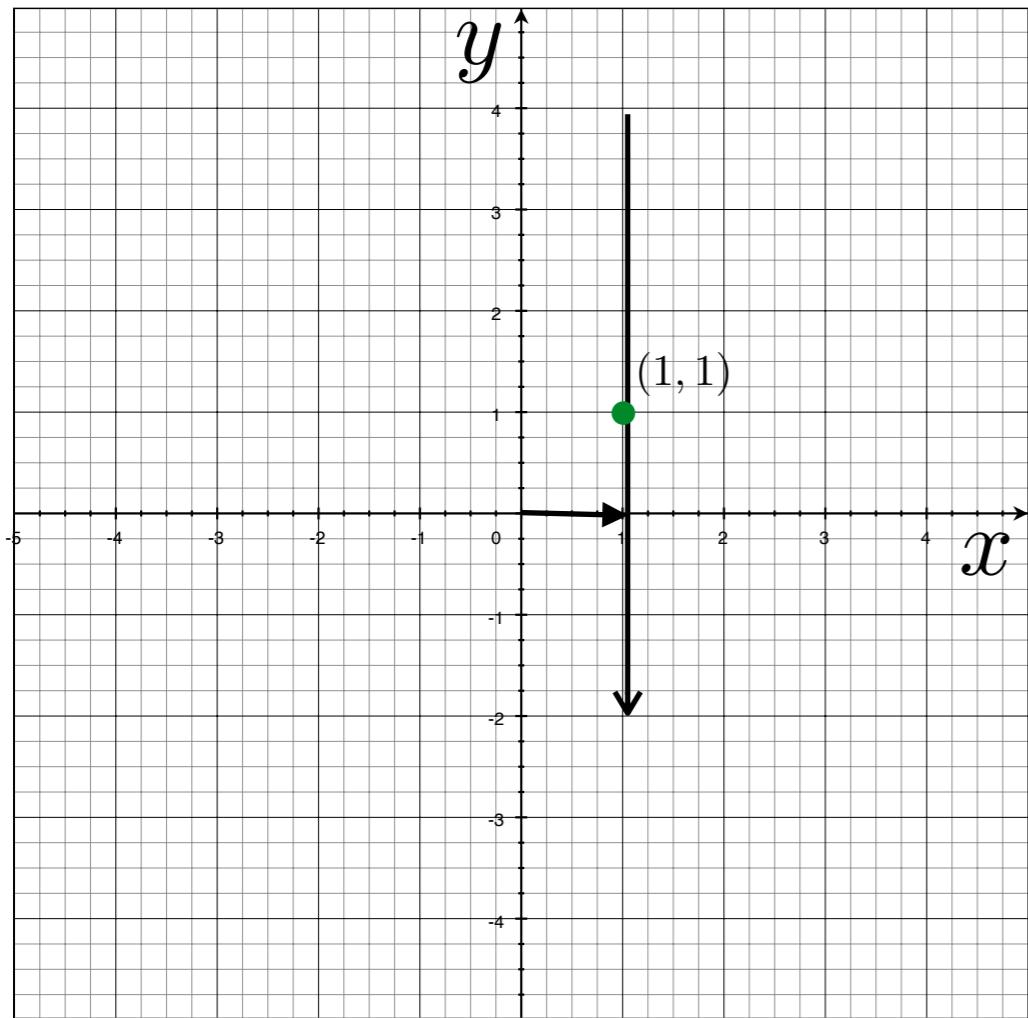
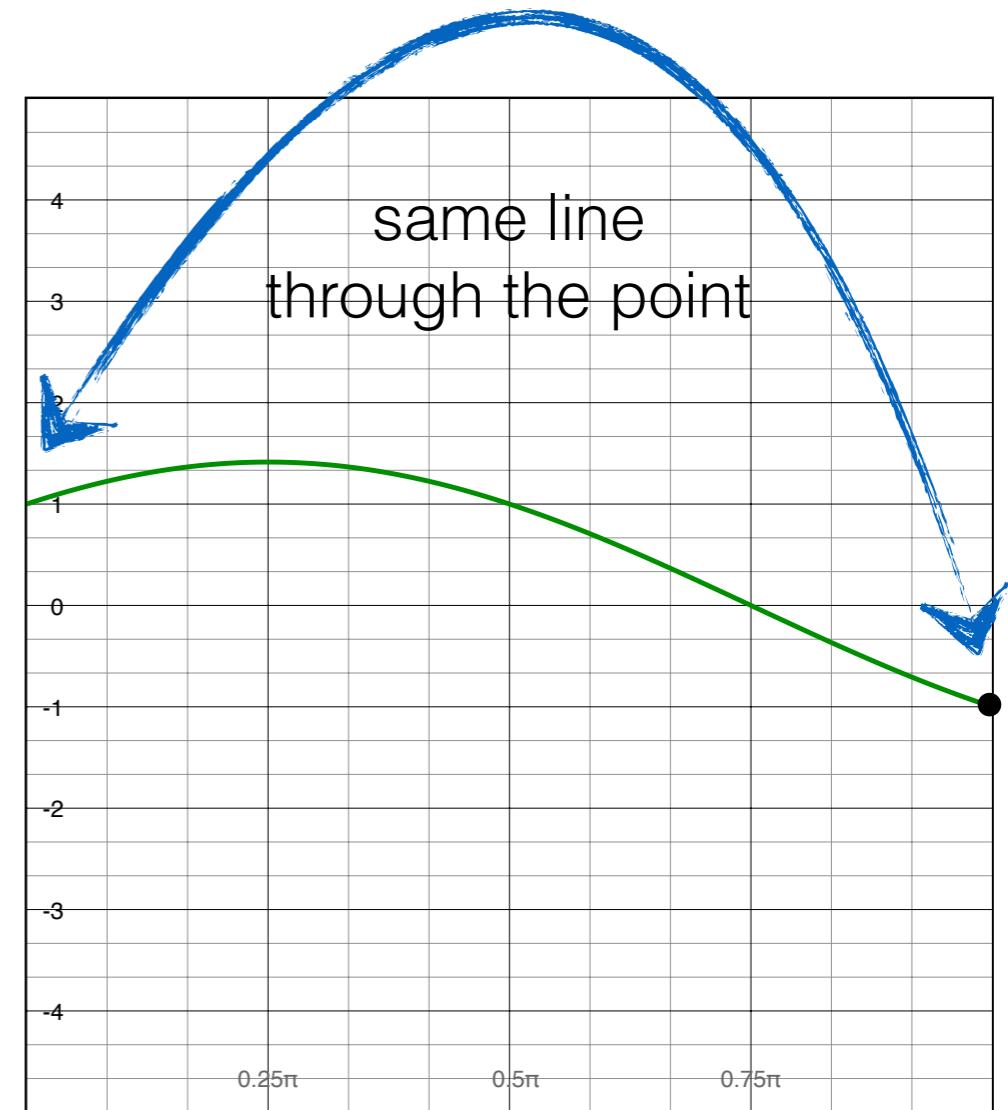


Image space

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



a line becomes a point

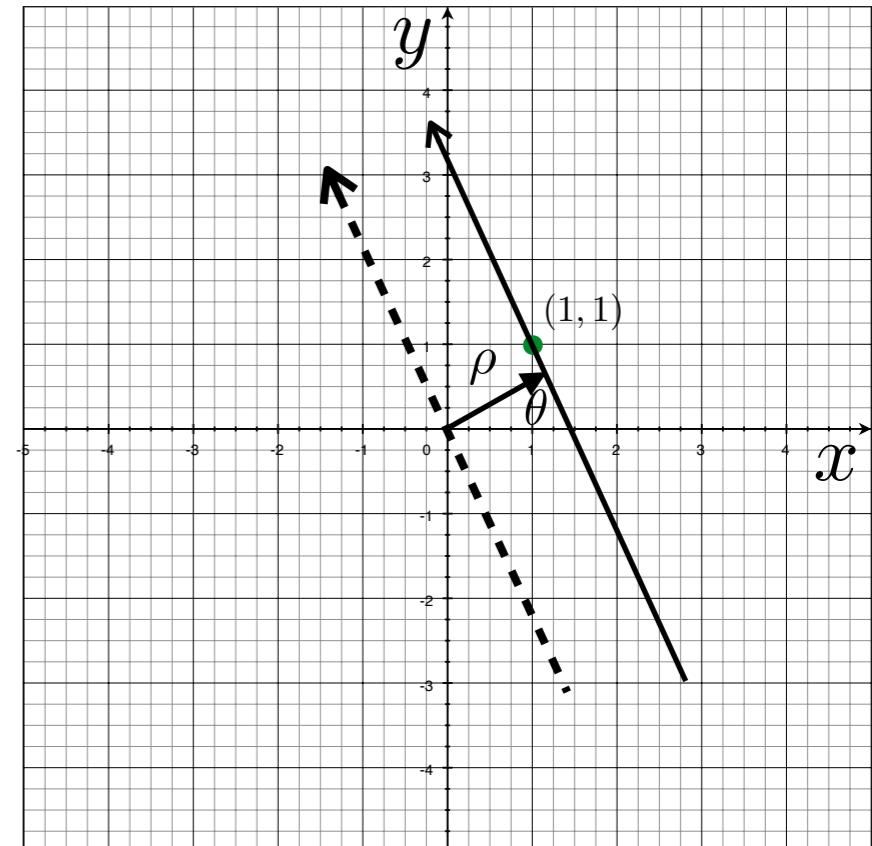
same line  
through the point

Parameter space

There are two ways to write the same line:

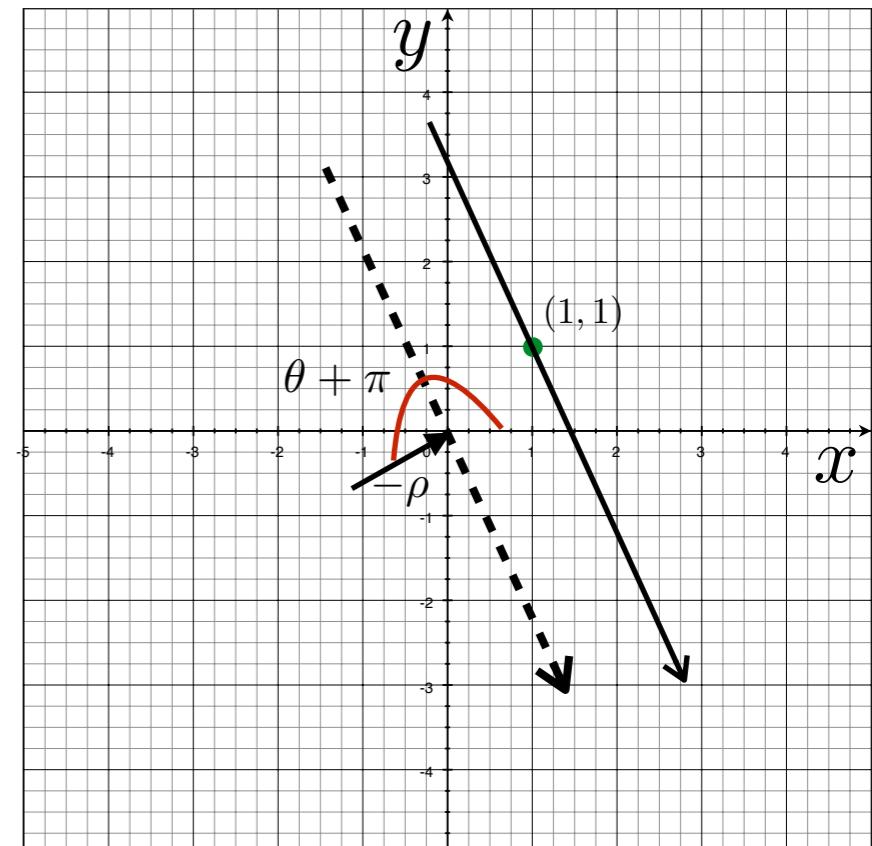
Positive rho version:

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



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

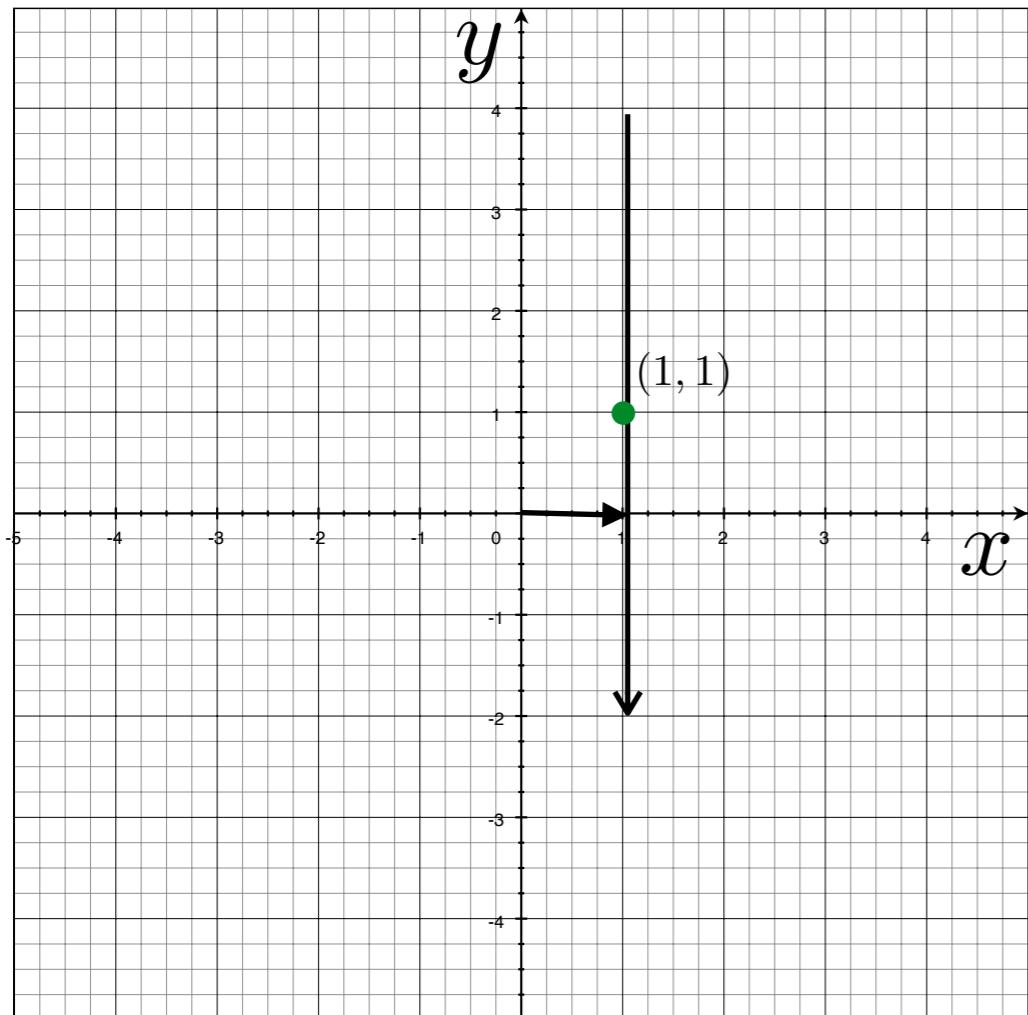
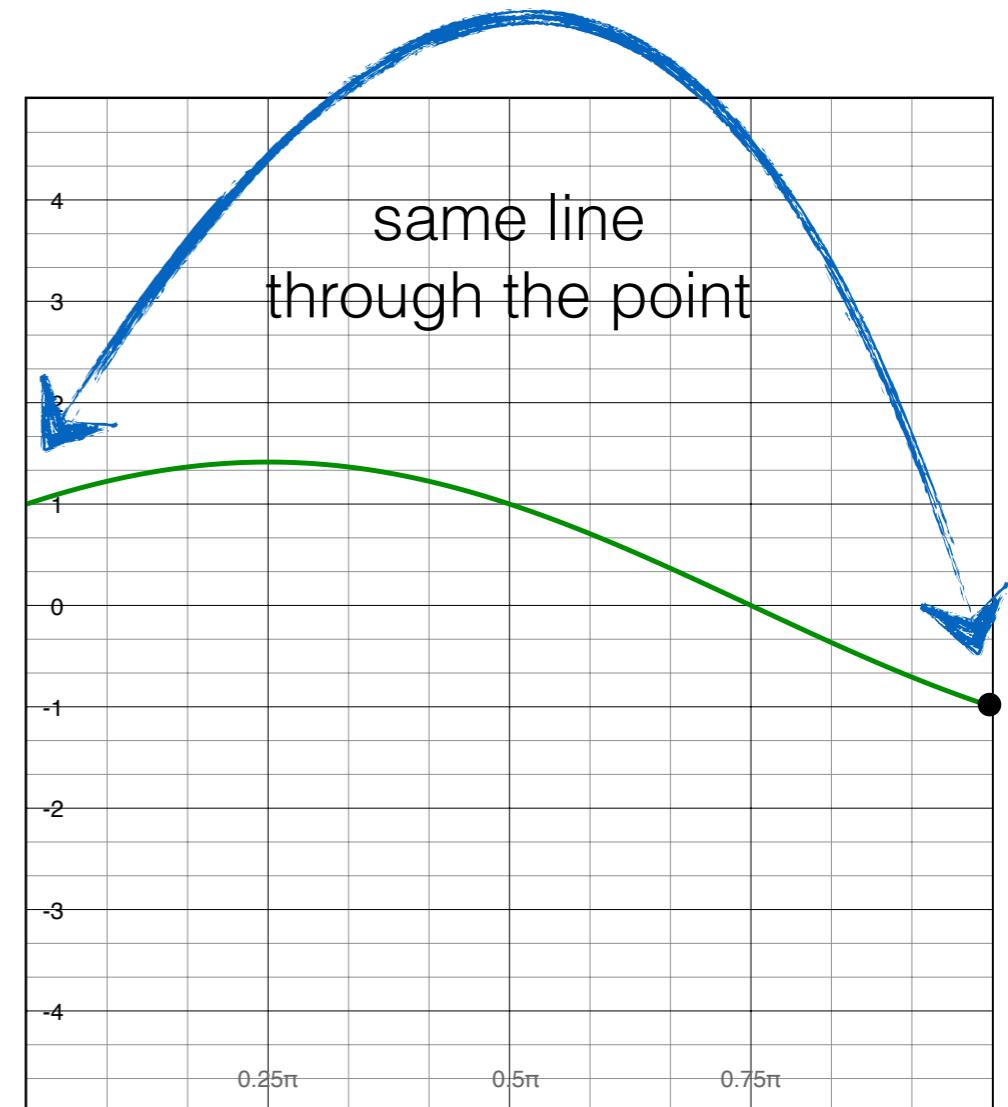


Image space

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



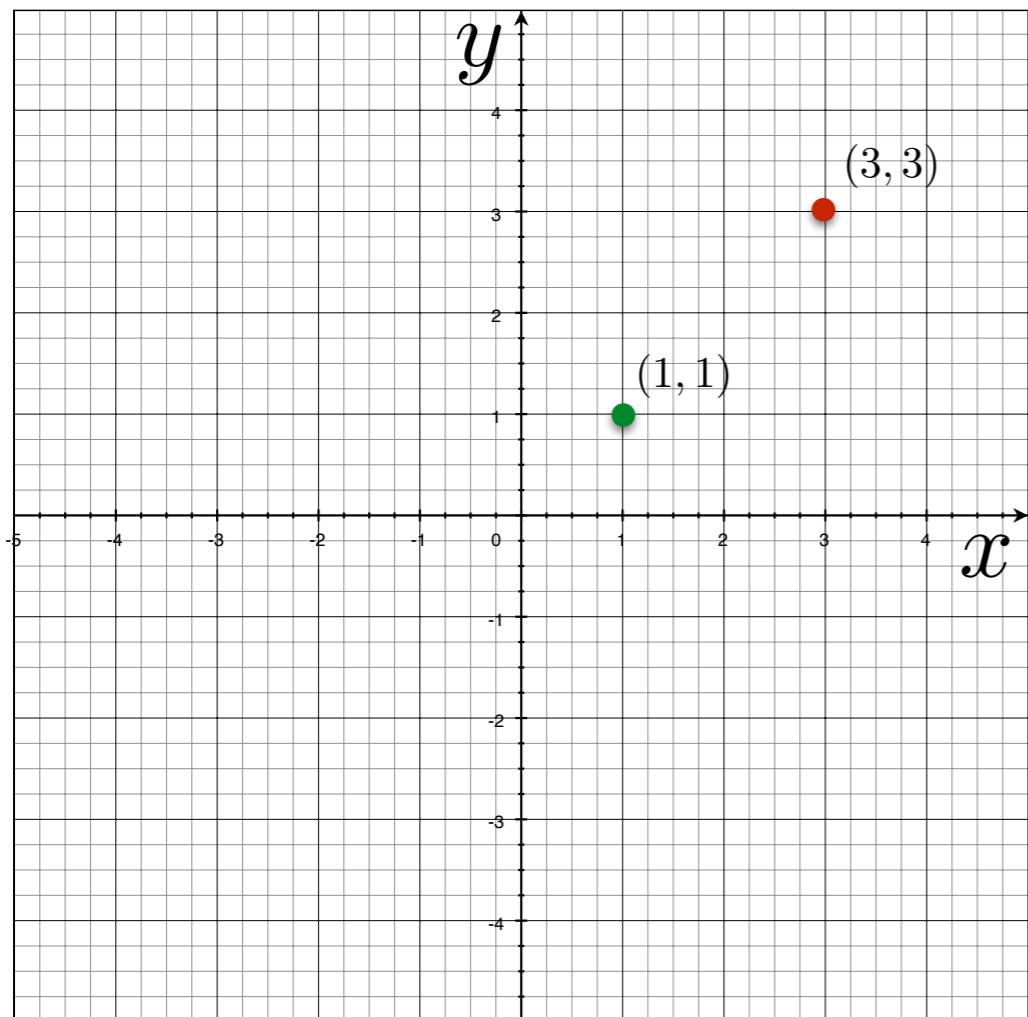
a line  
becomes  
a point

same line  
through the point

Parameter space

# Image and parameter space

variables  
 $y = mx + b$   
parameters



two points  
become  
?

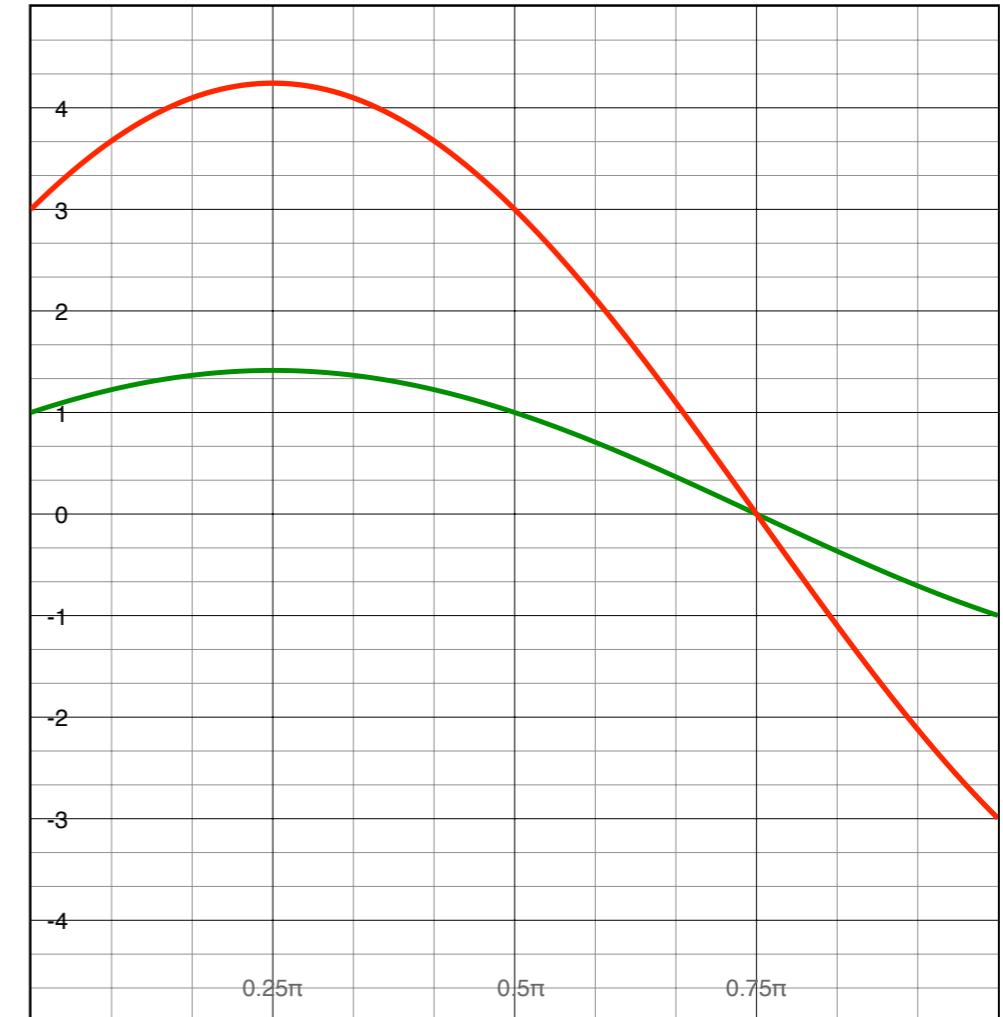
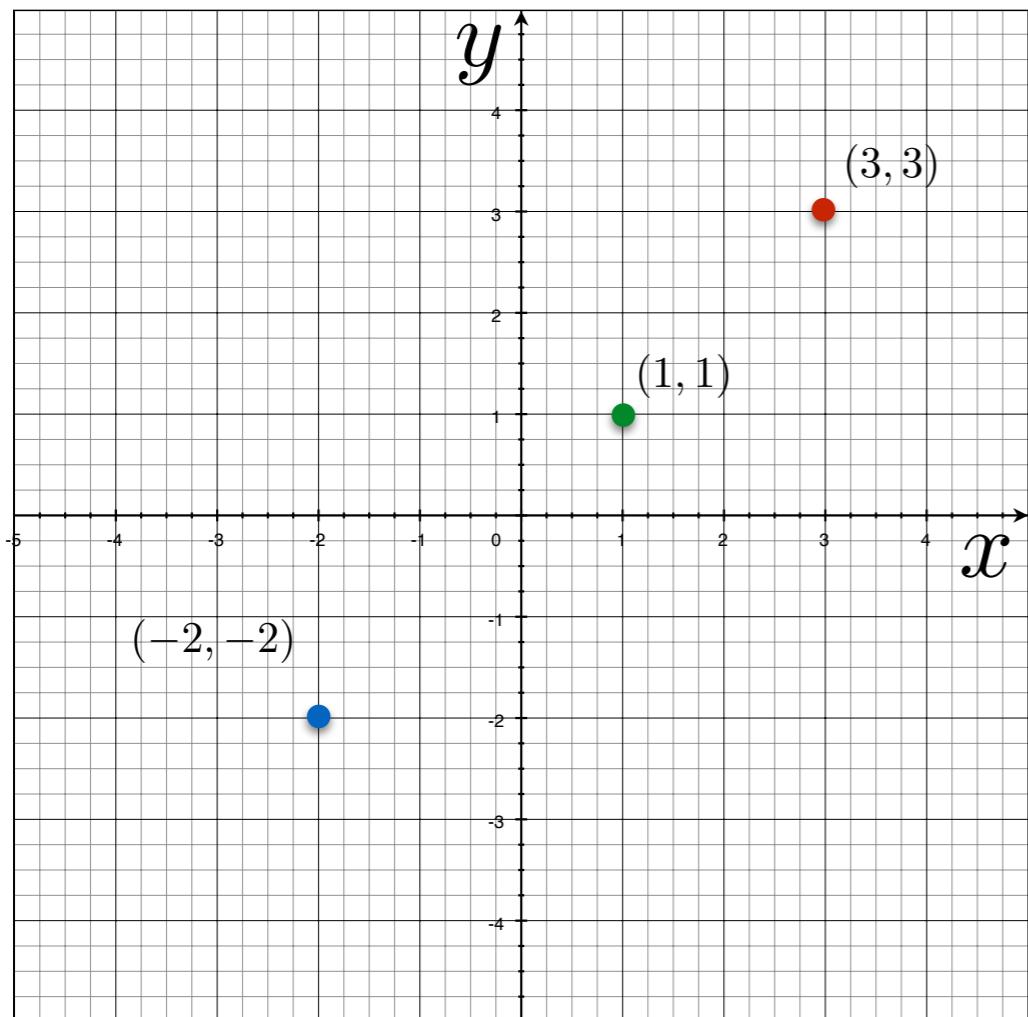


Image space

Parameter space

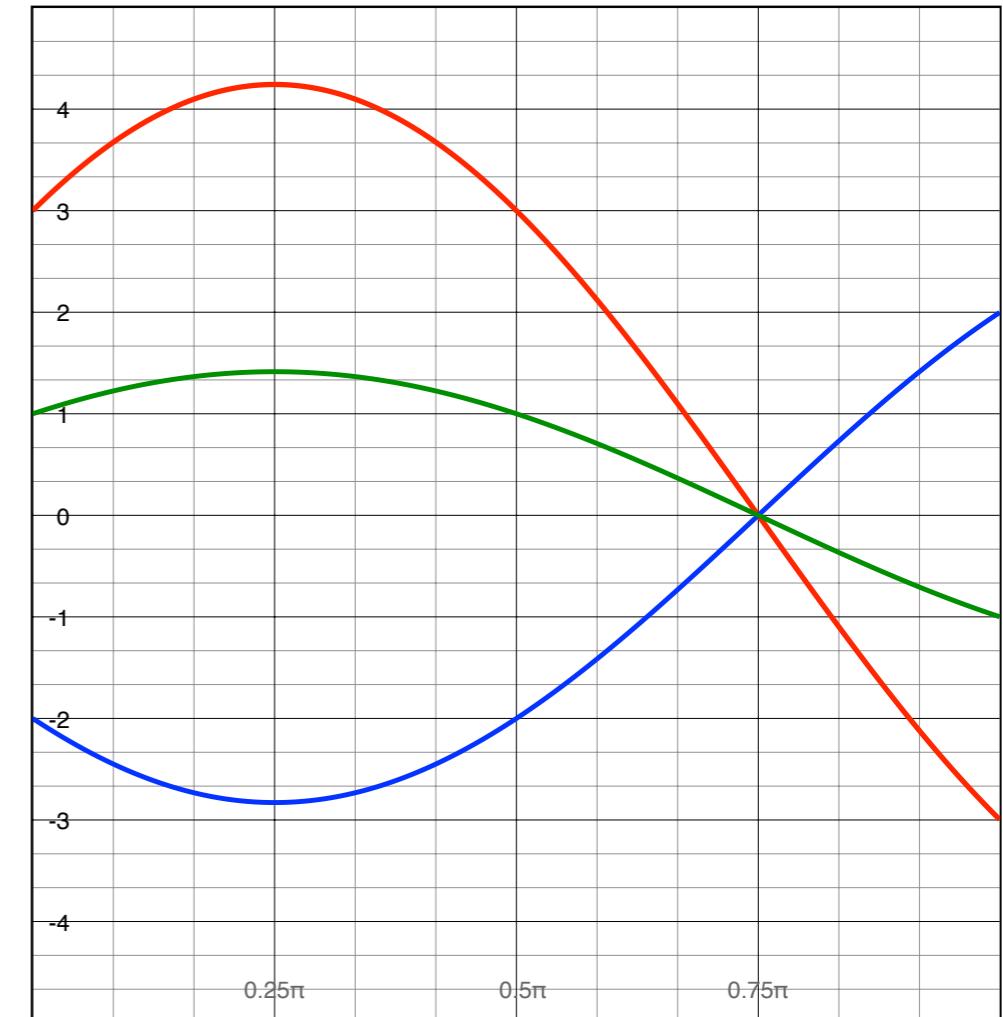
# Image and parameter space

variables  
 $y = mx + b$   
parameters



three points  
become  
?

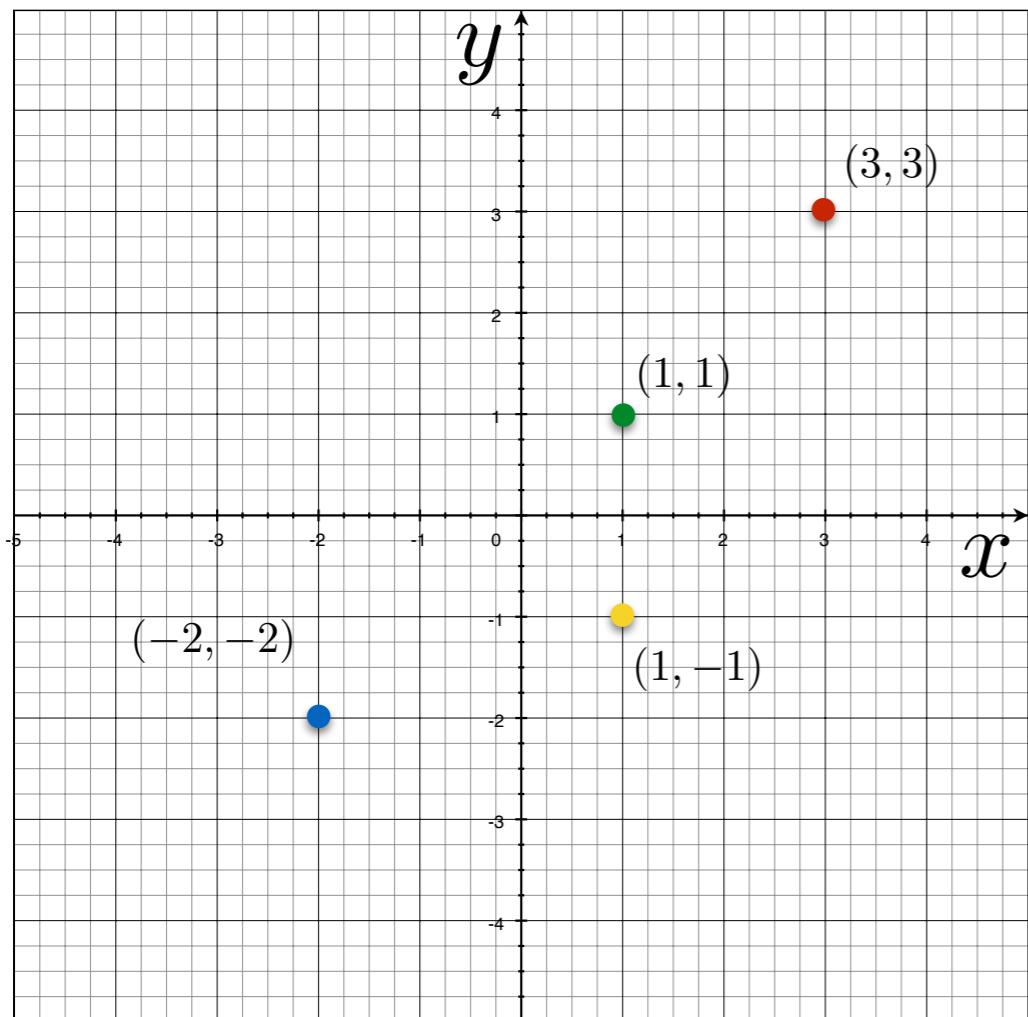
Image space



Parameter space

# Image and parameter space

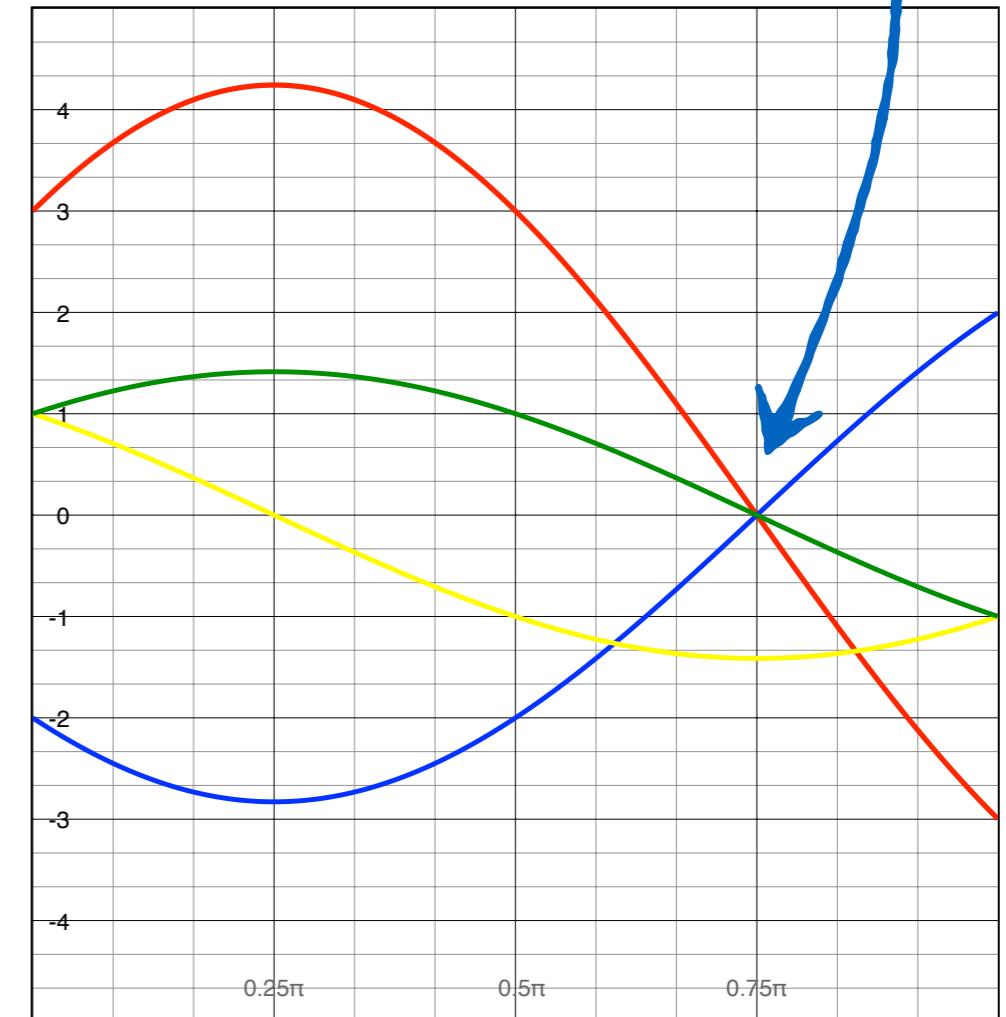
variables  
 $y = mx + b$   
parameters



four points become ?

Image space

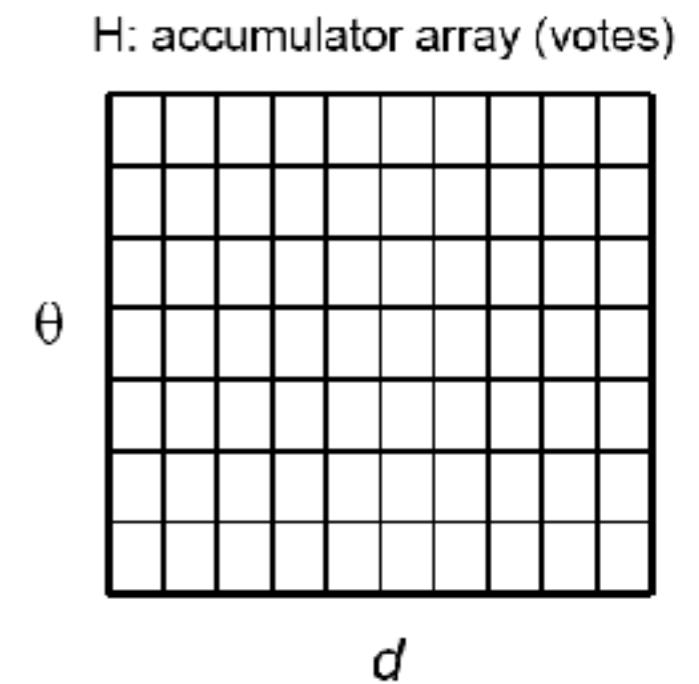
*What does this represent?*



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!

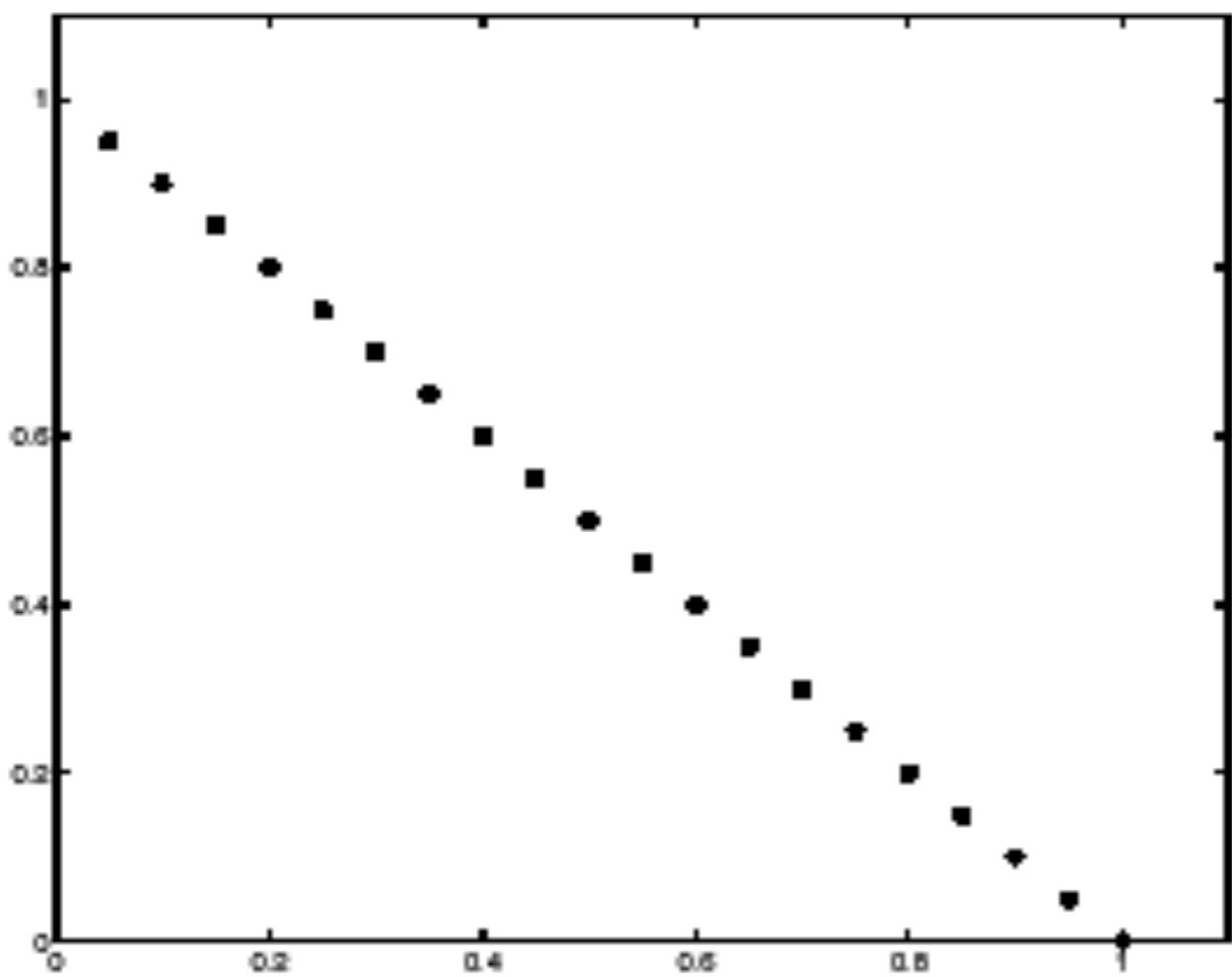
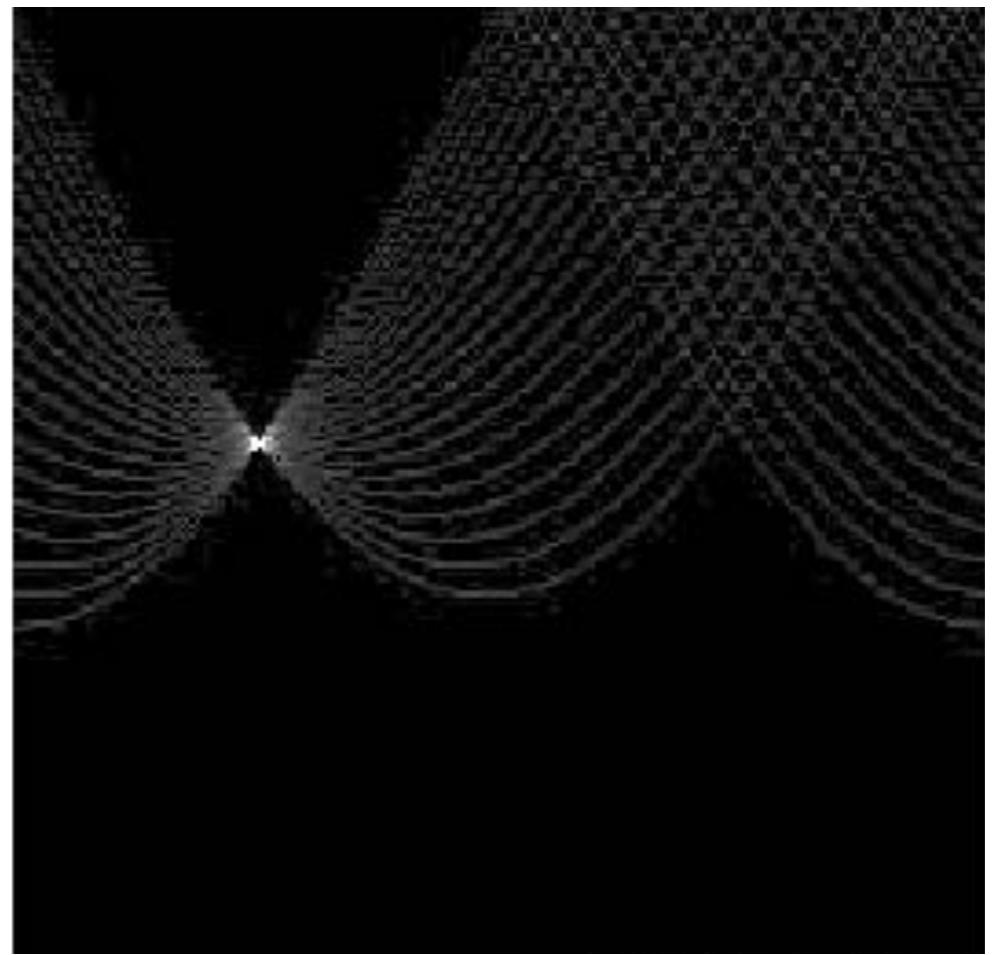


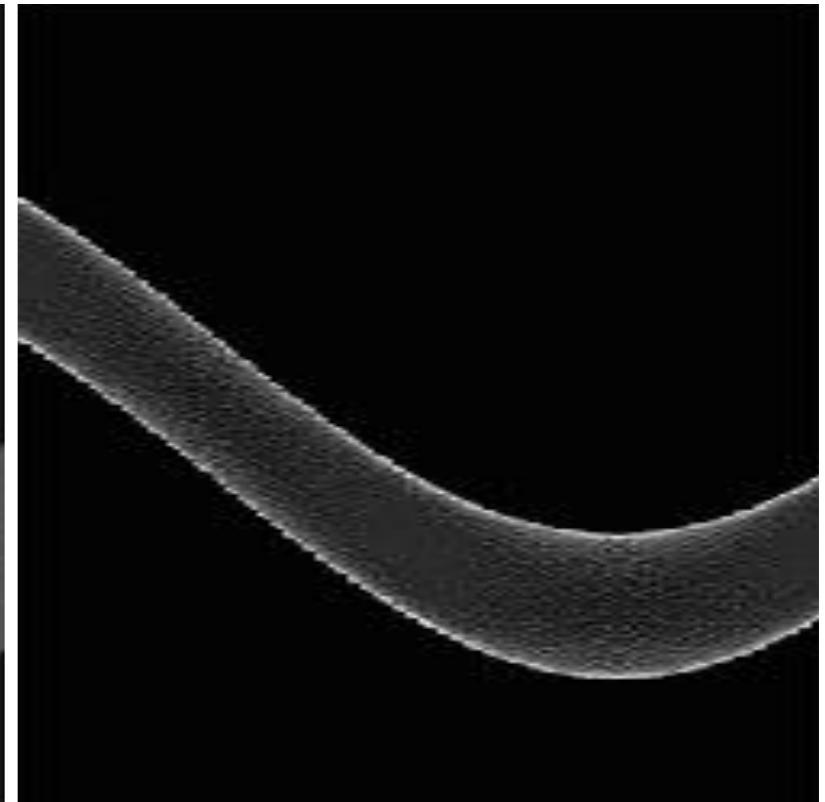
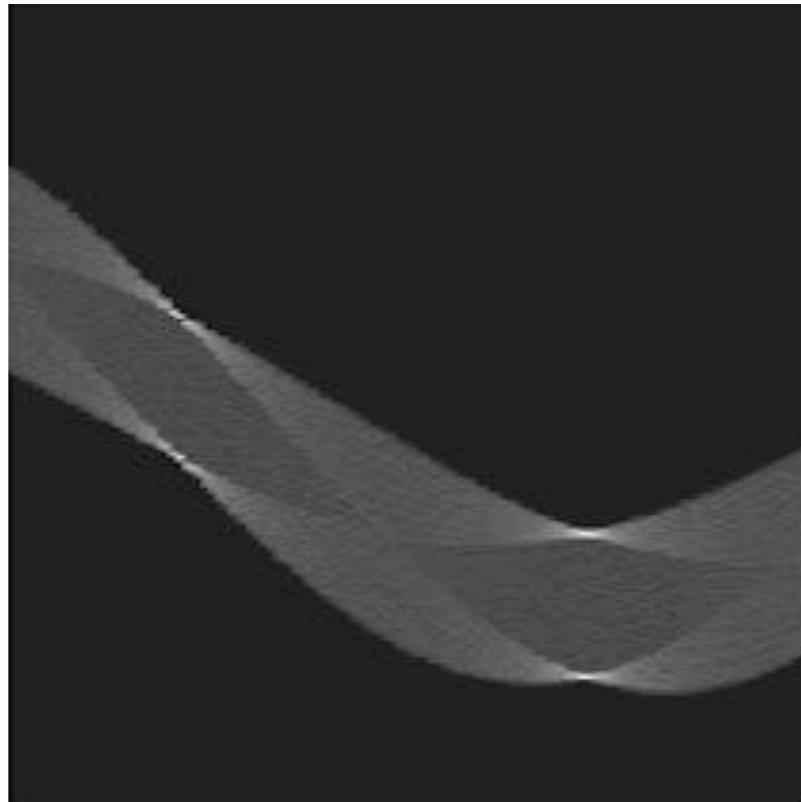
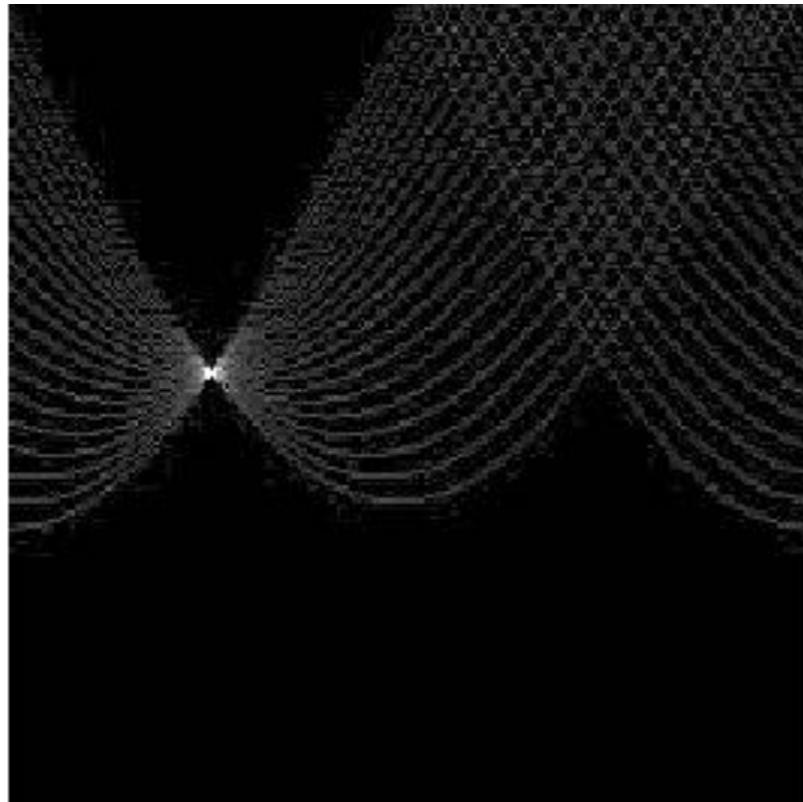
Image space



Votes

# Basic shapes

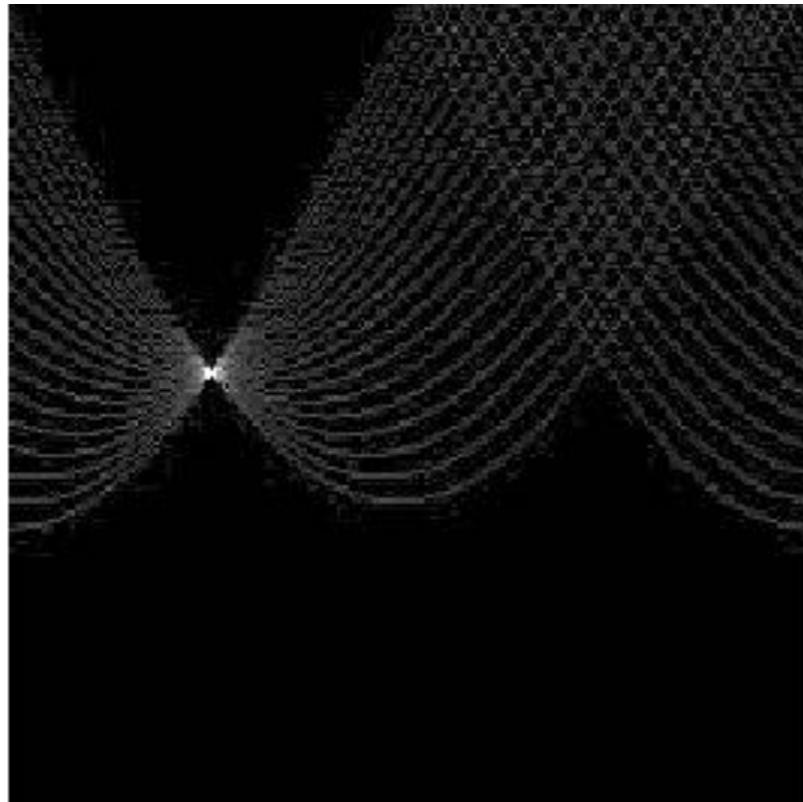
(in polar parameter space)



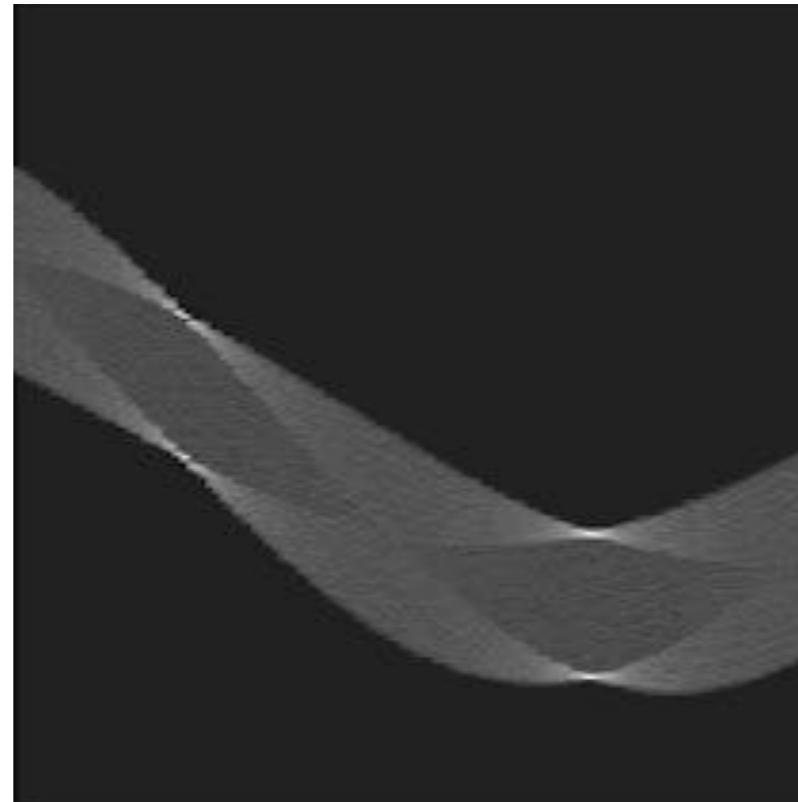
*can you guess the shape?*

# Basic shapes

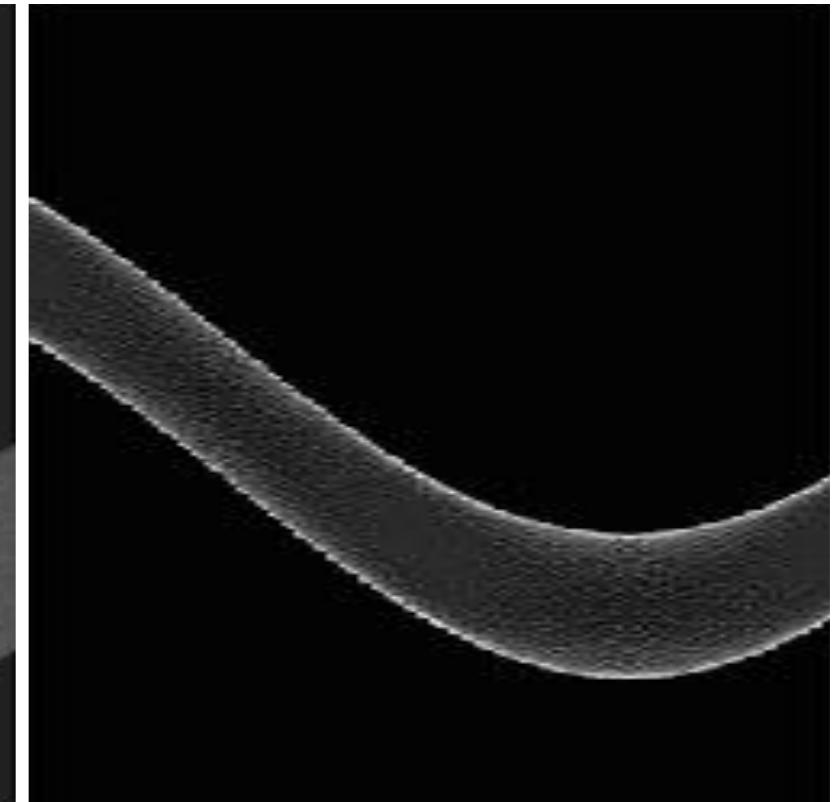
(in polar parameter space)



line

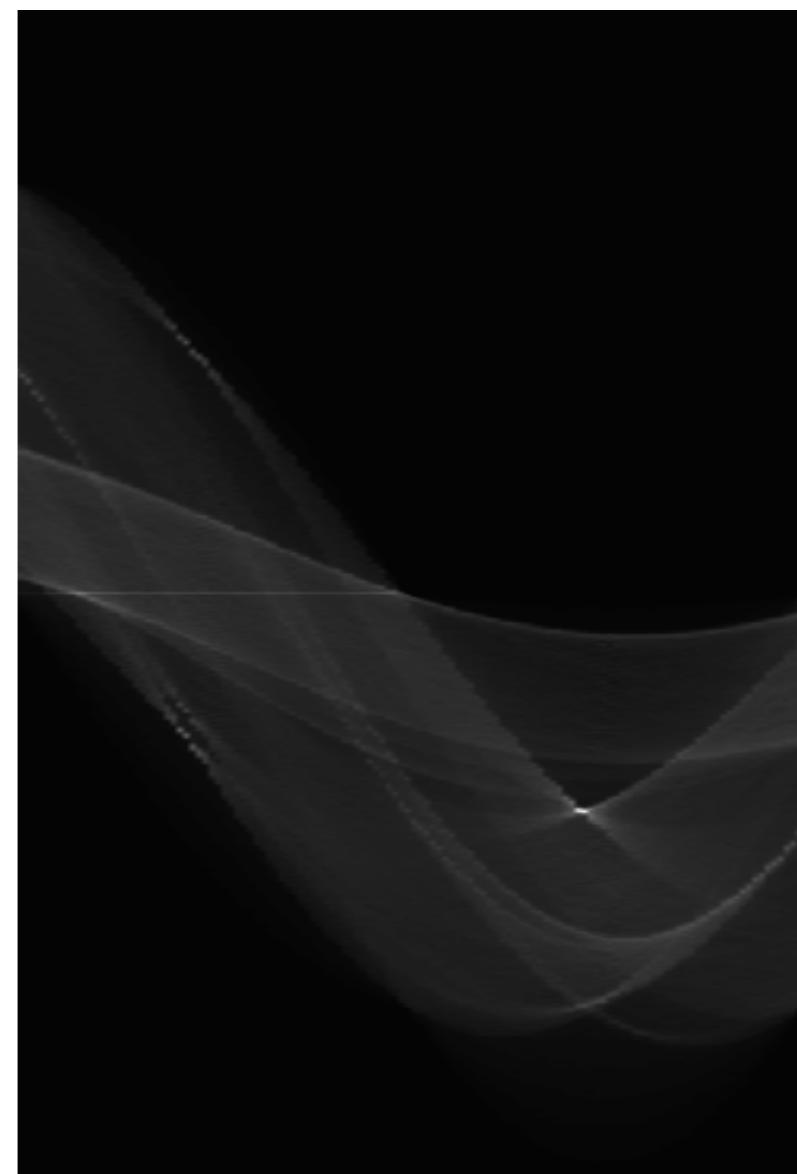
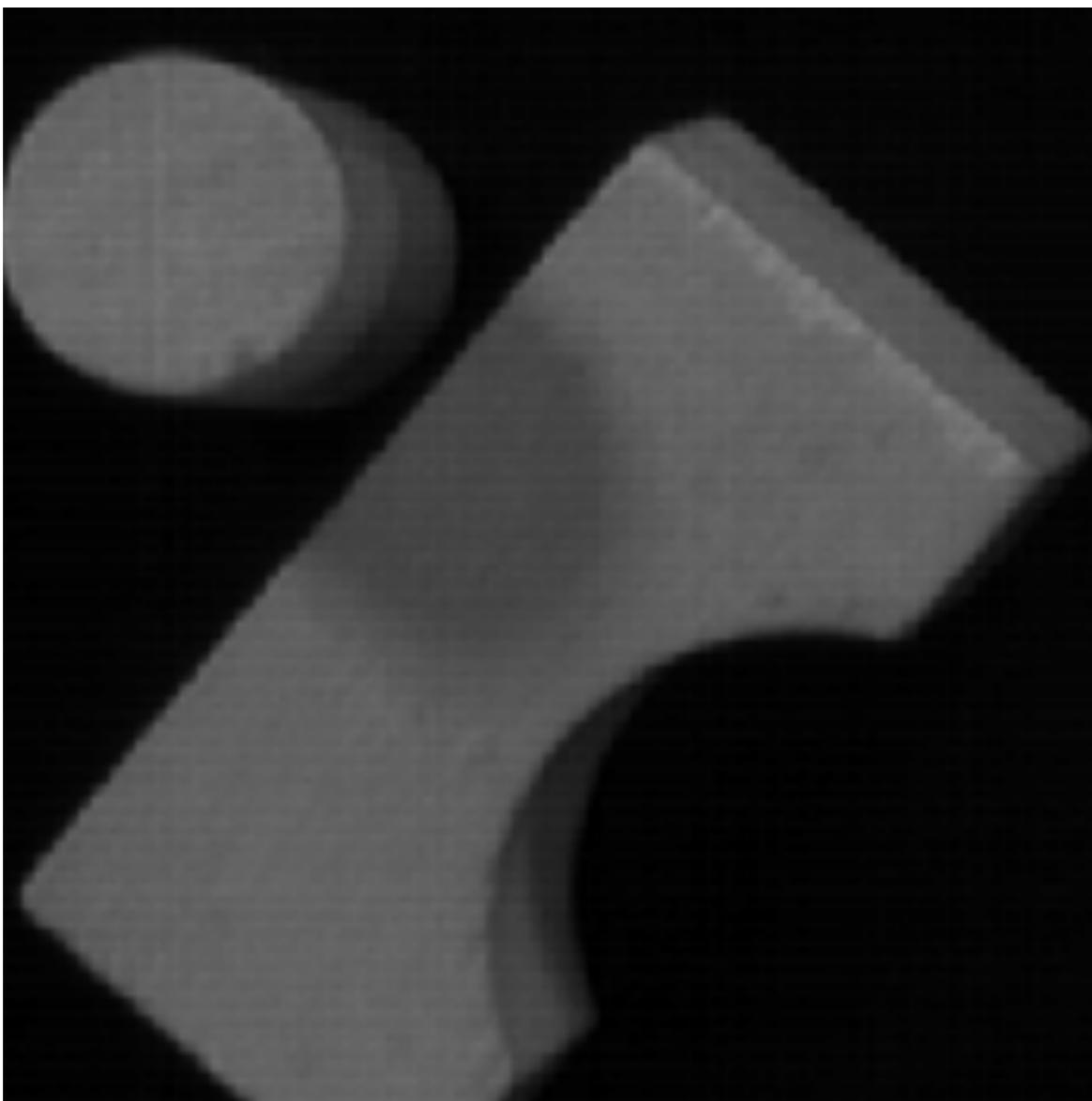


rectangle

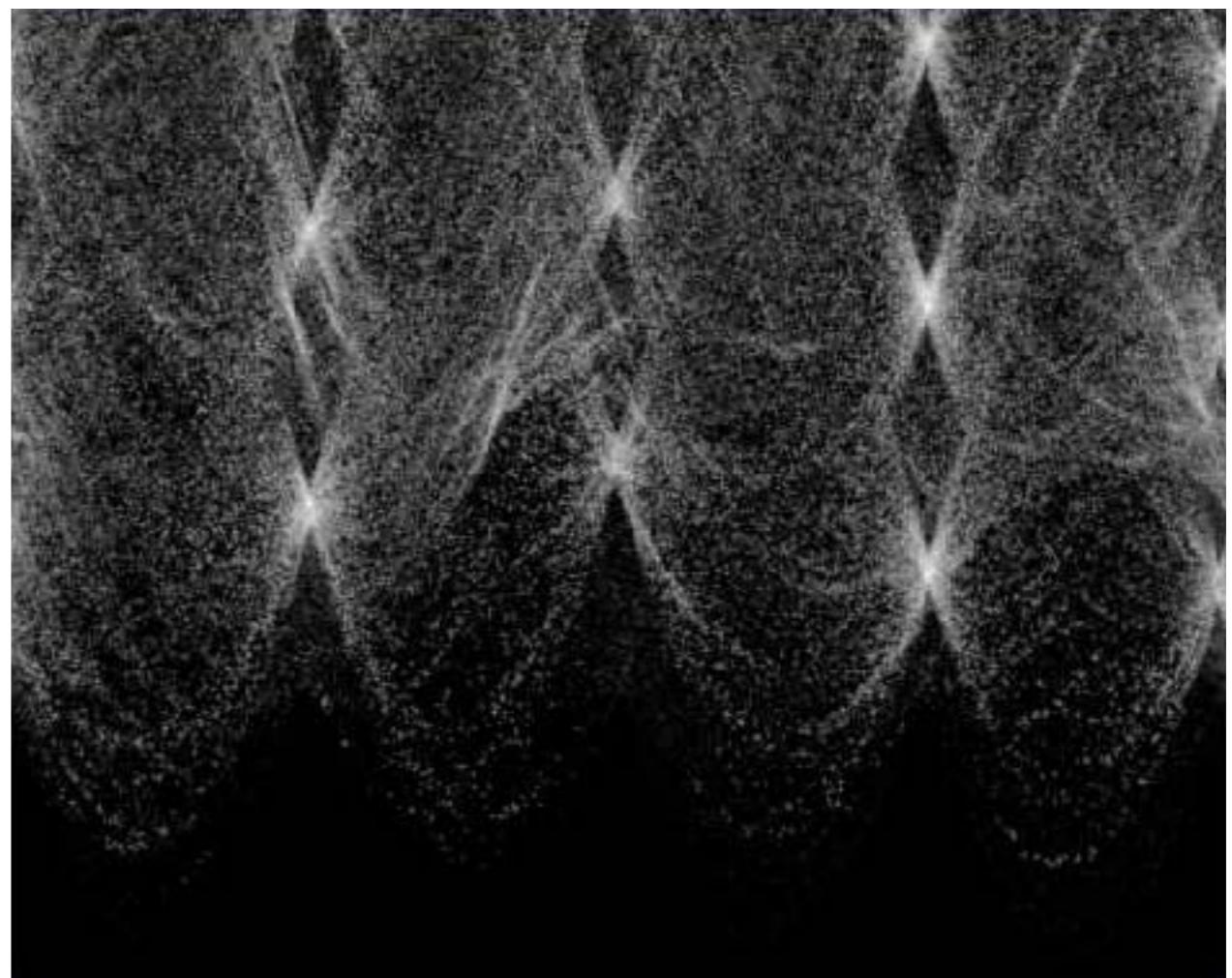


circle

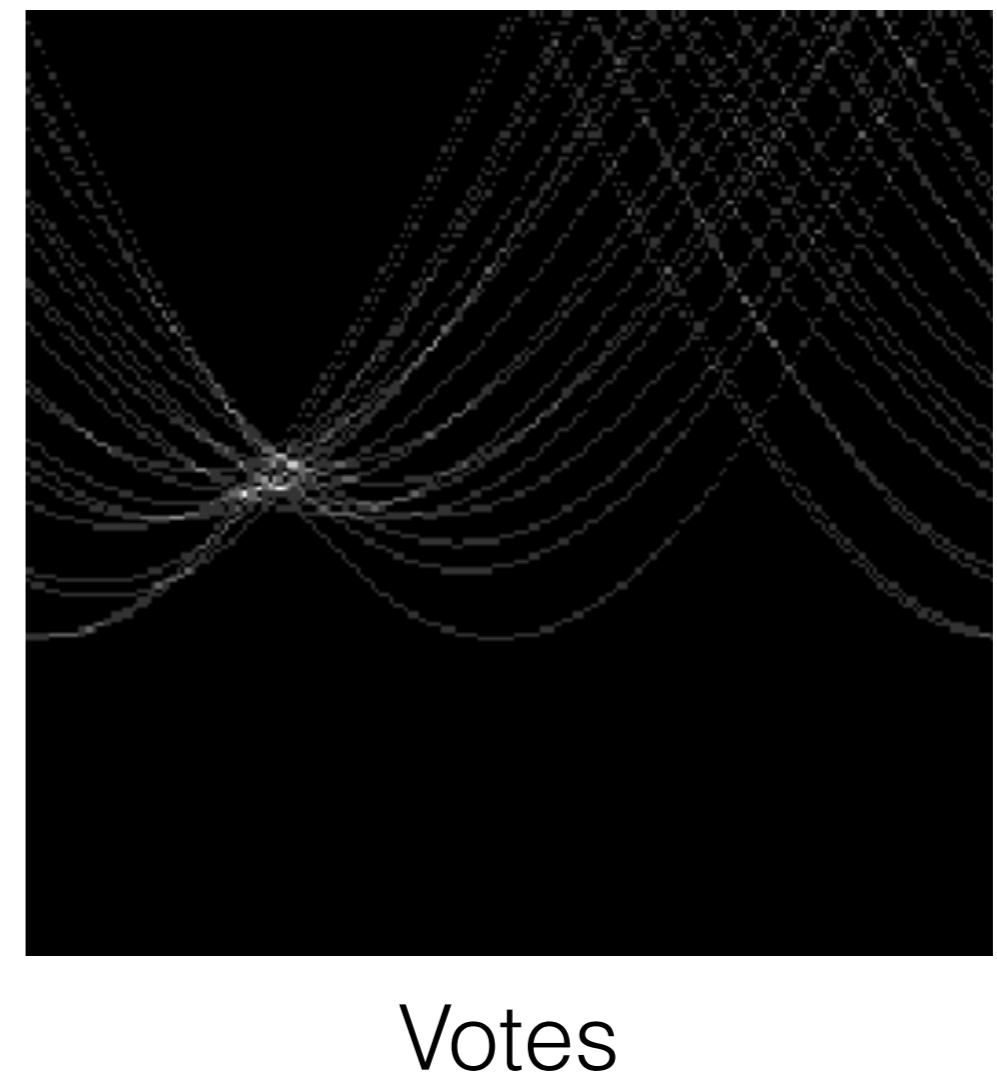
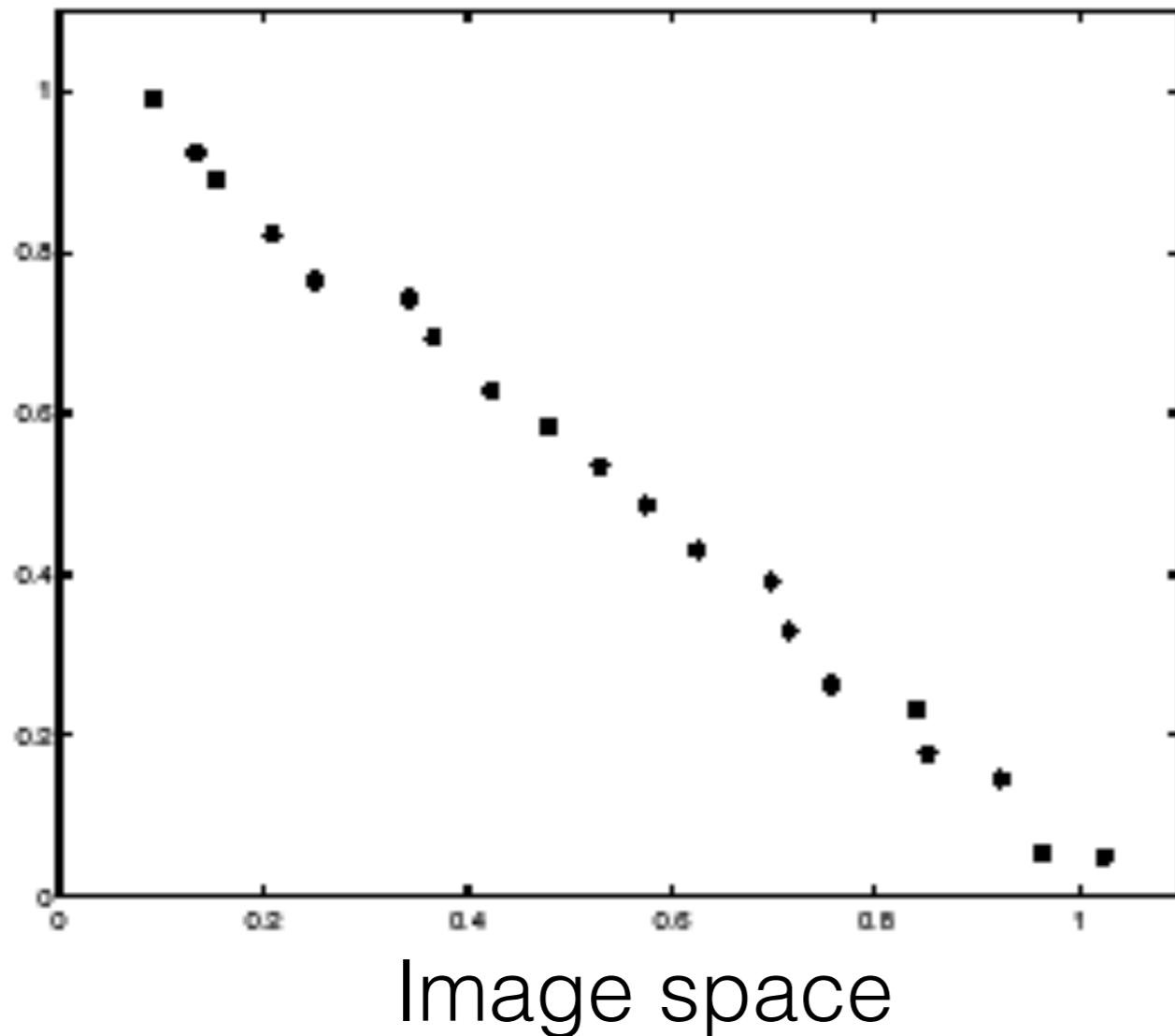
# Basic Shapes



# More complex image



In practice, measurements are noisy...



Too much noise ...

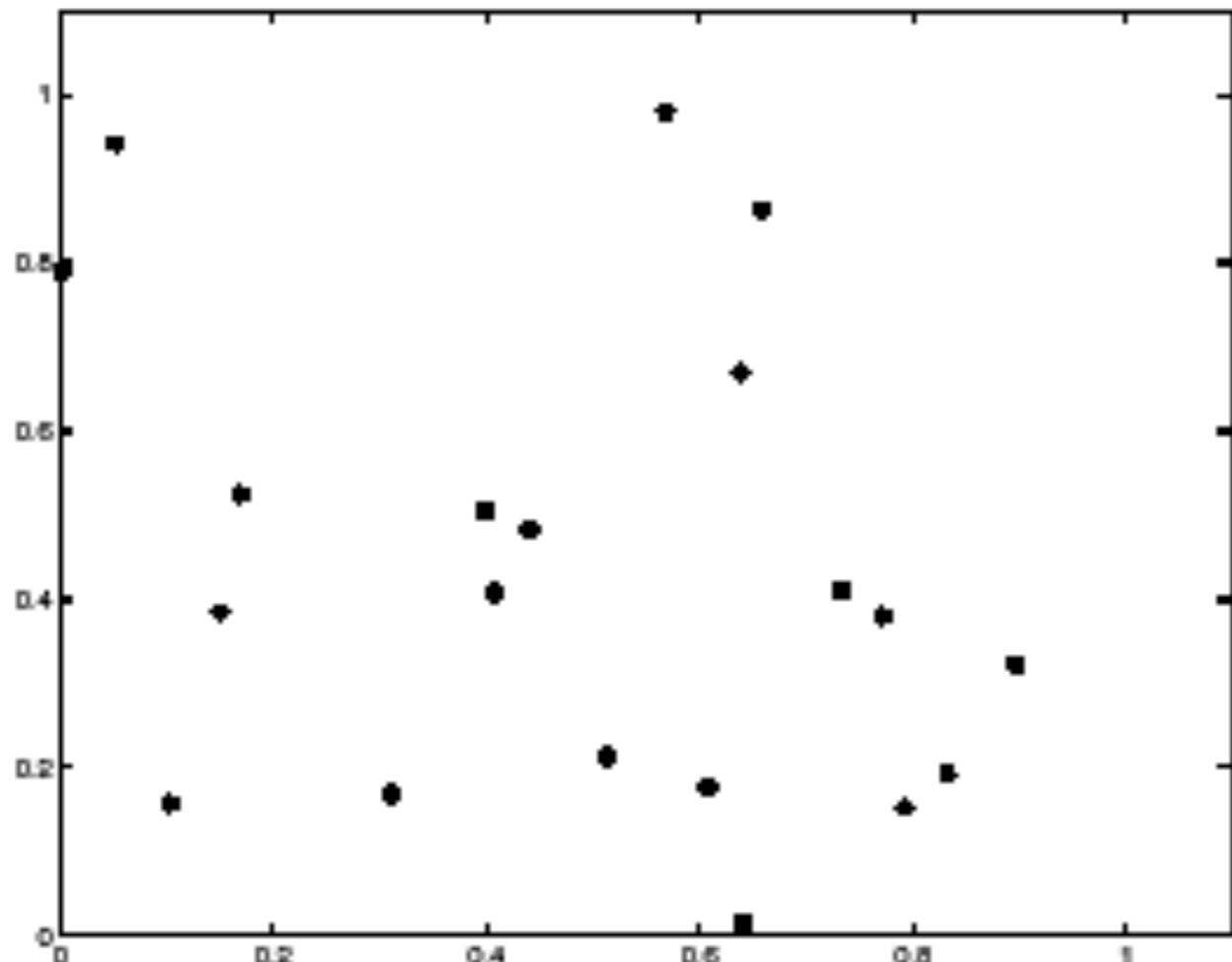
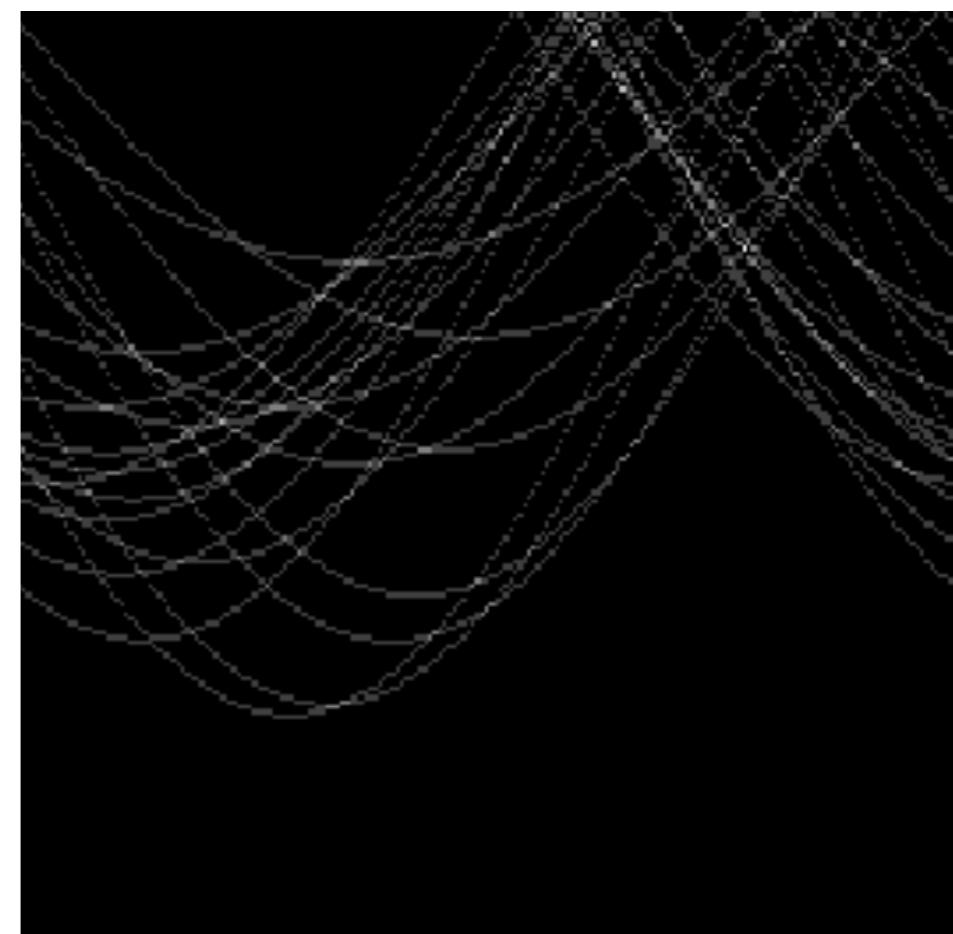


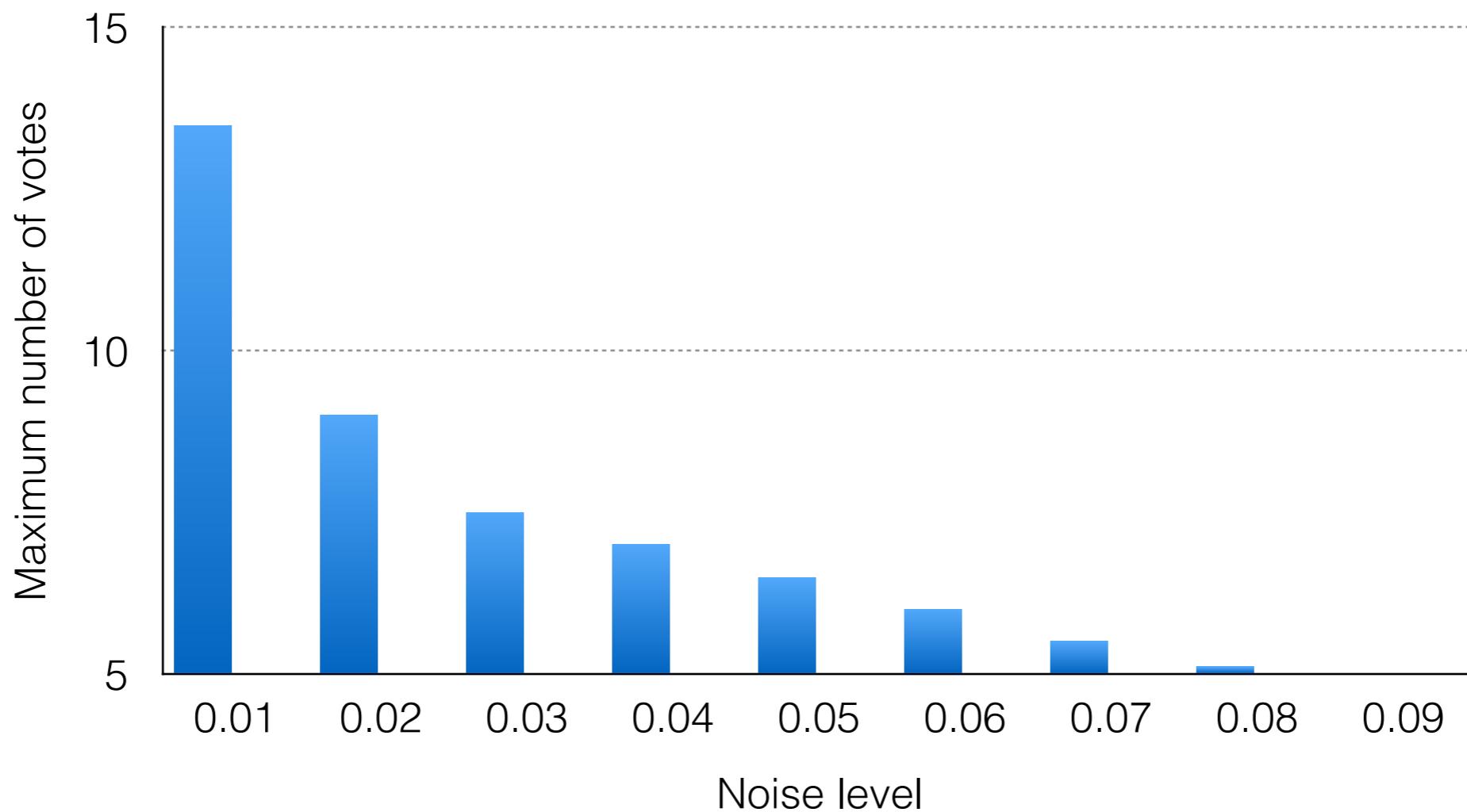
Image space



Votes

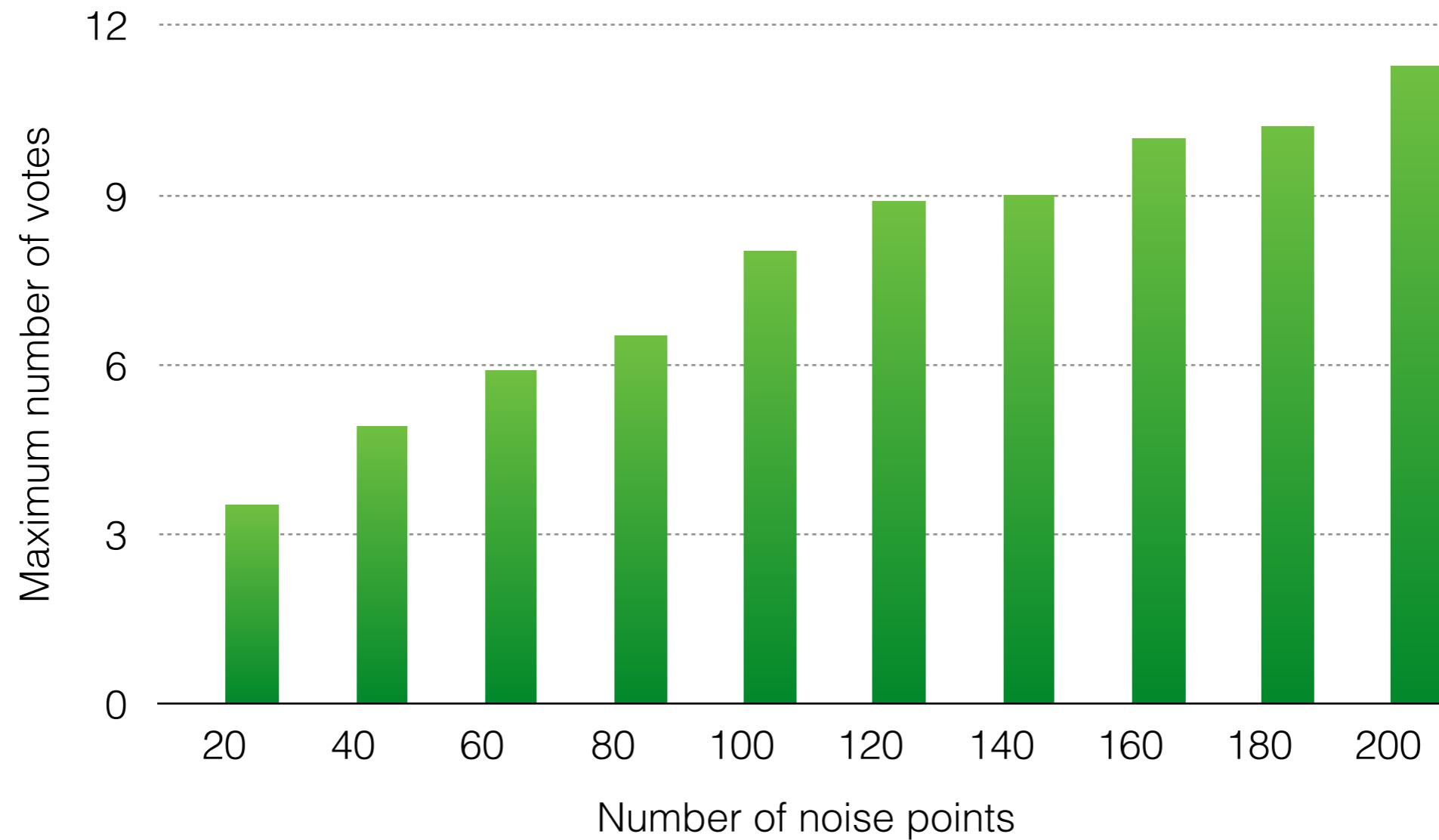
# Effects of noise level

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



More noise, less votes (in the right bin)

# Effect of noise points

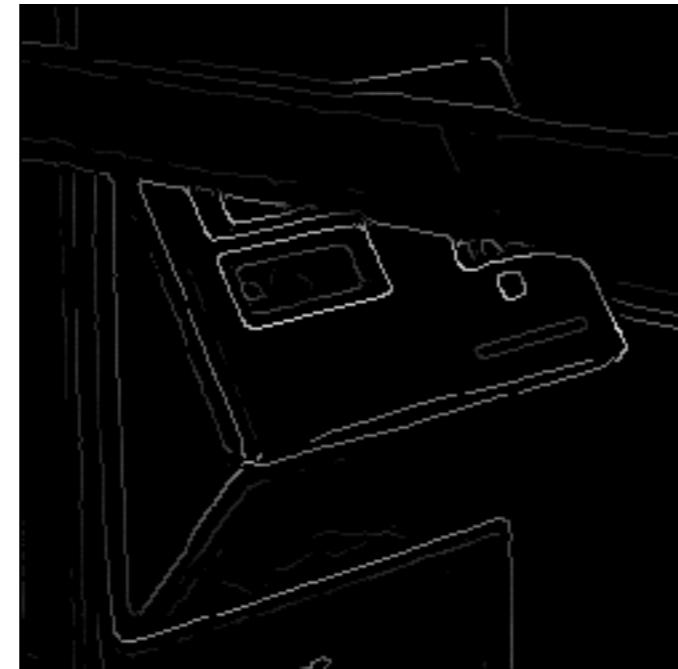


More noise, more votes (in the wrong bin)

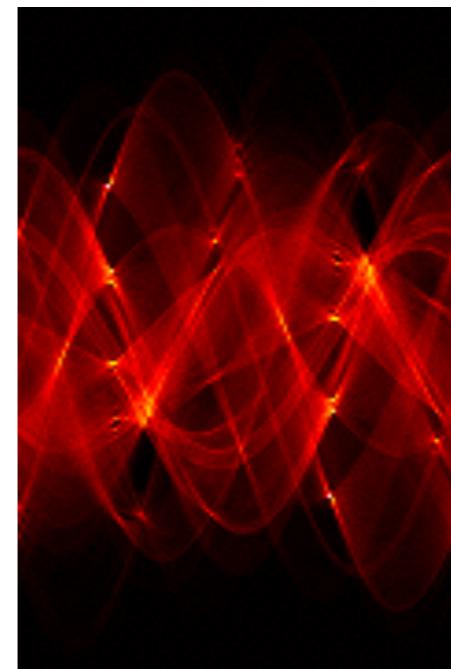
# Real-world example



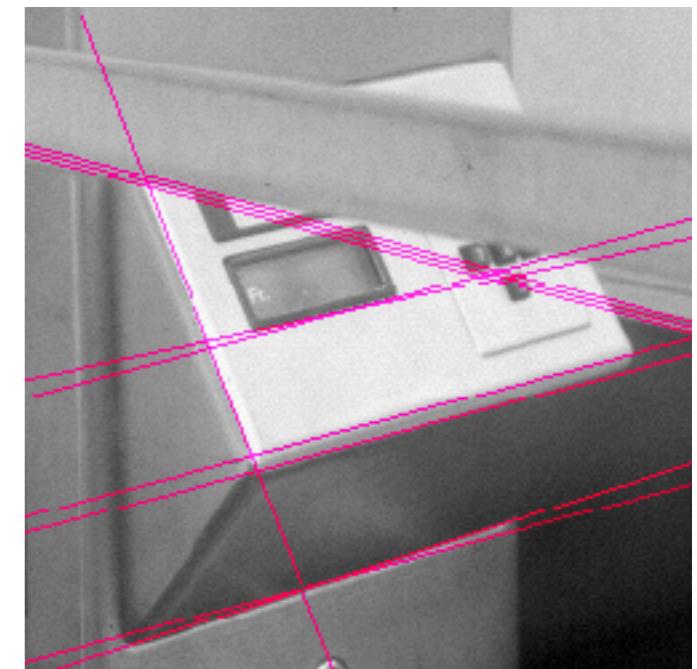
Original



Edges



parameter space



Hough Lines