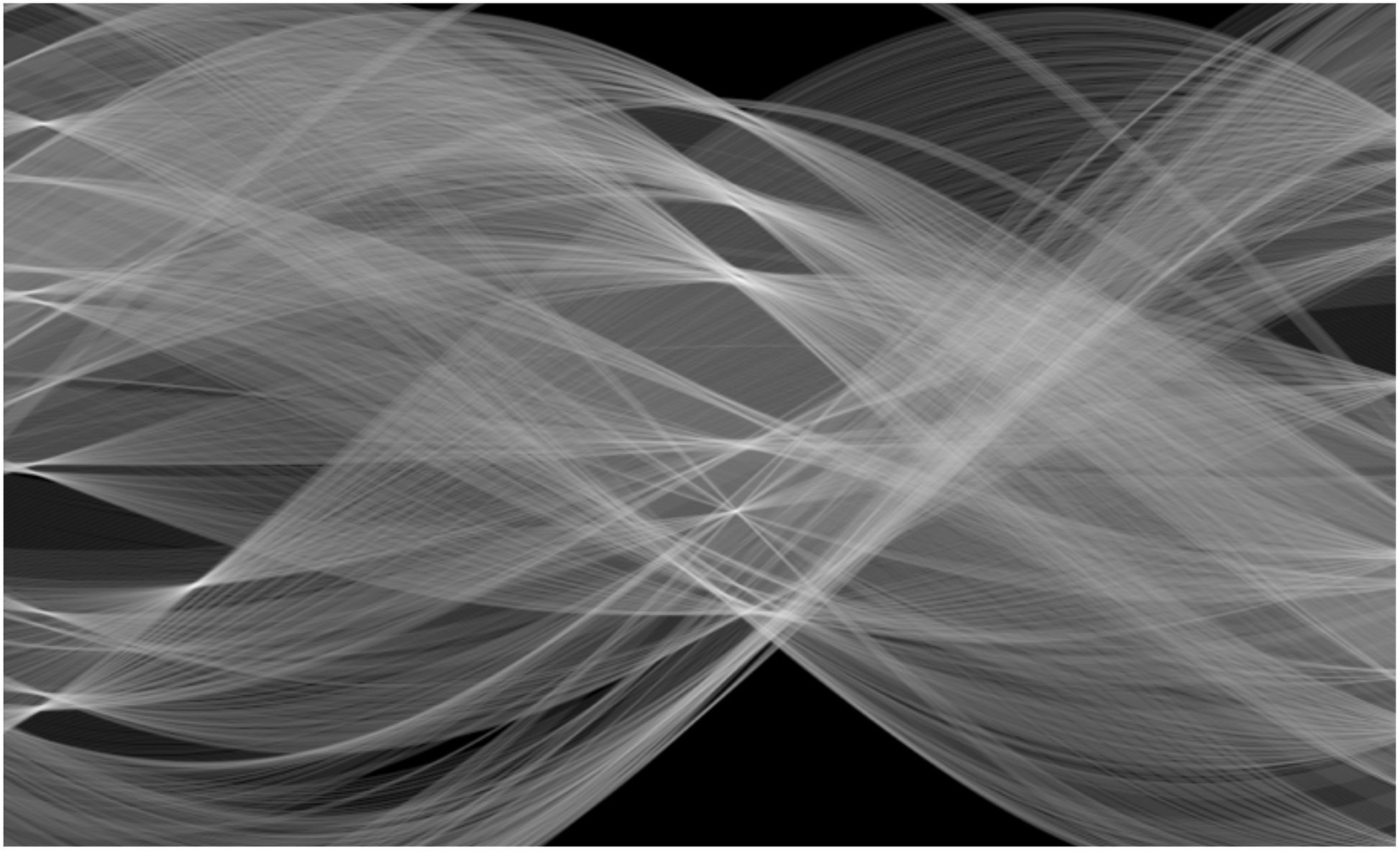
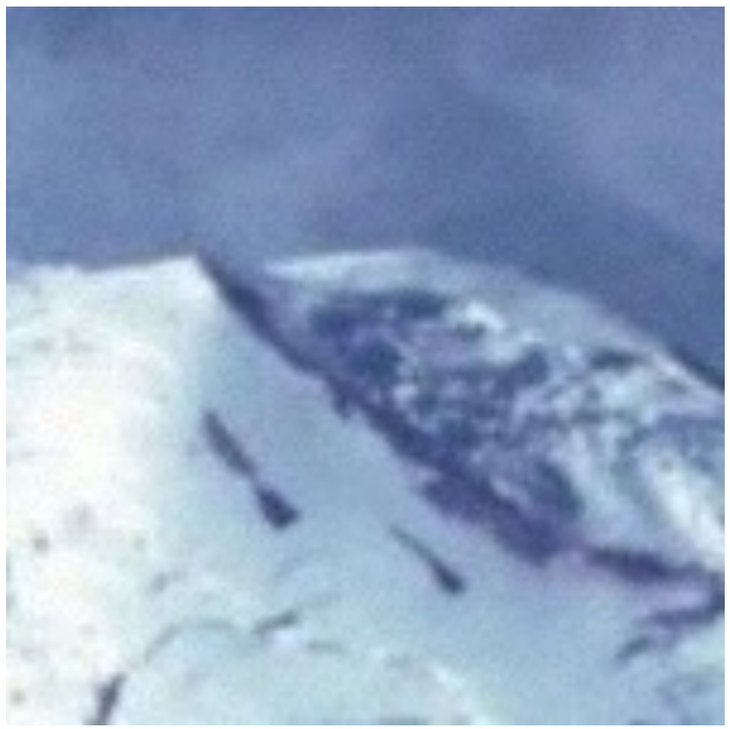


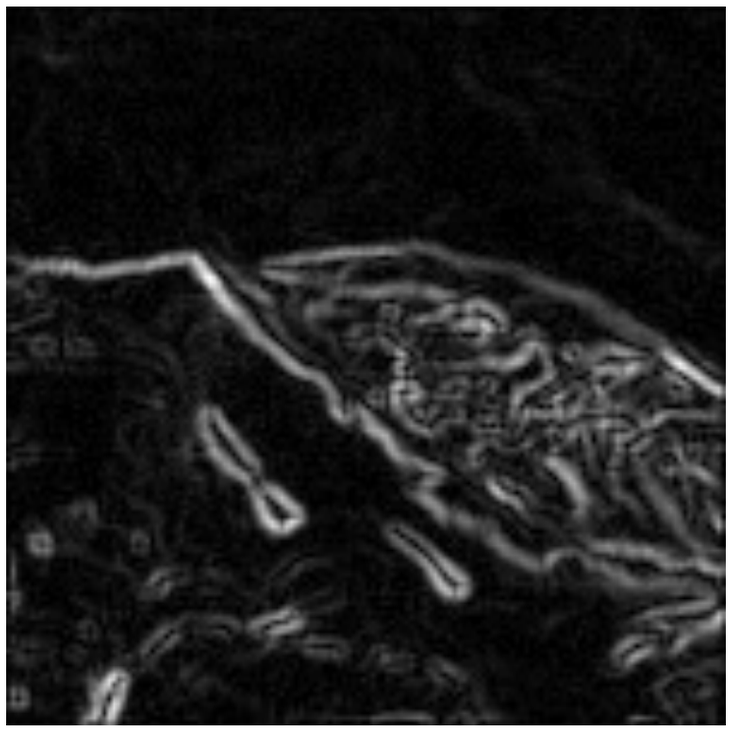
Hough Transform

16-385 Computer Vision

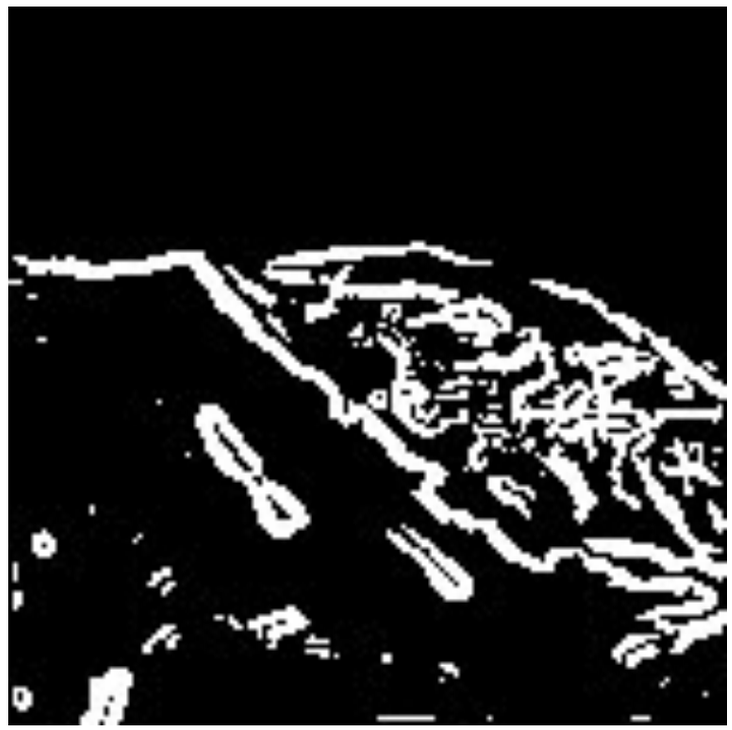
motivation



Original image



Edge detection



Thresholding

How do we find image boundaries (lines)?

Hough Transform

- Generic framework for detecting an object
- Edges don’t have to be connected
- Lines can be occluded
- Key idea: edges **vote** for the possible models

Voting in parameter space

Example: ’line detection’

Each point votes for the line it belongs to

Image and parameter space

variables

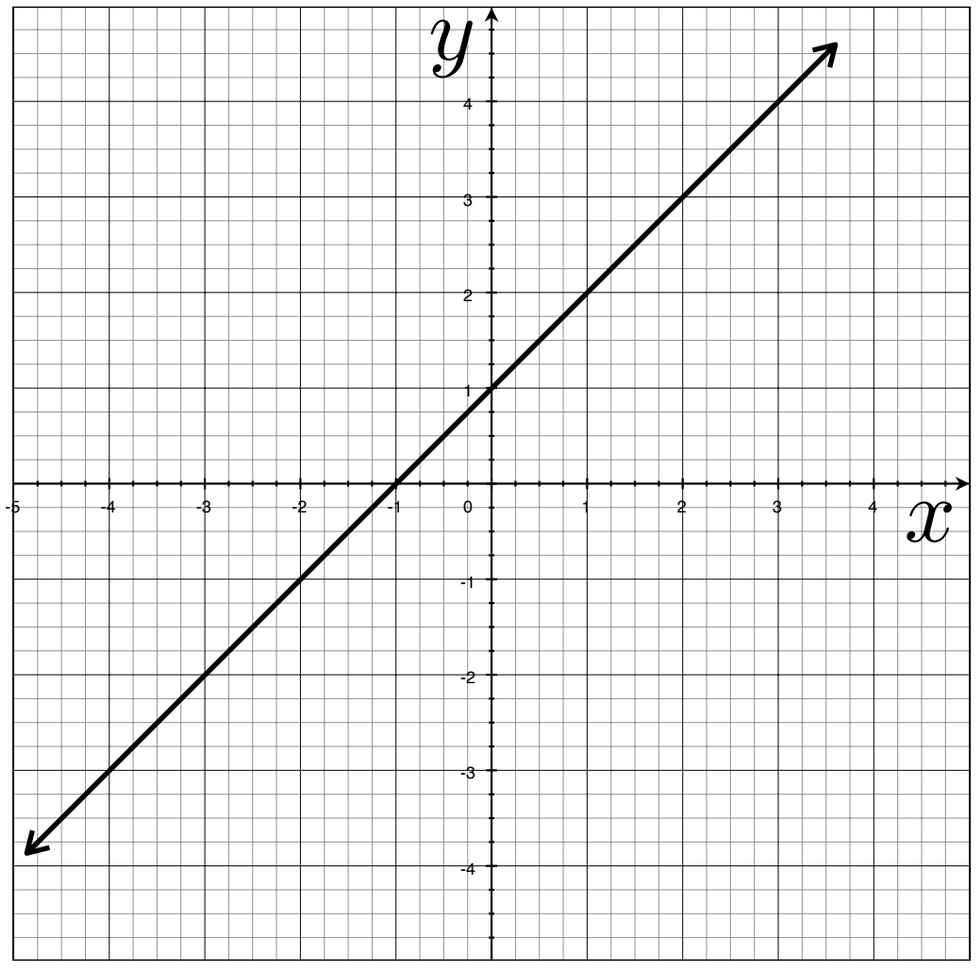
$$y = mx + b$$

parameters

variables

$$y - mx = b$$

parameters



a line becomes a point

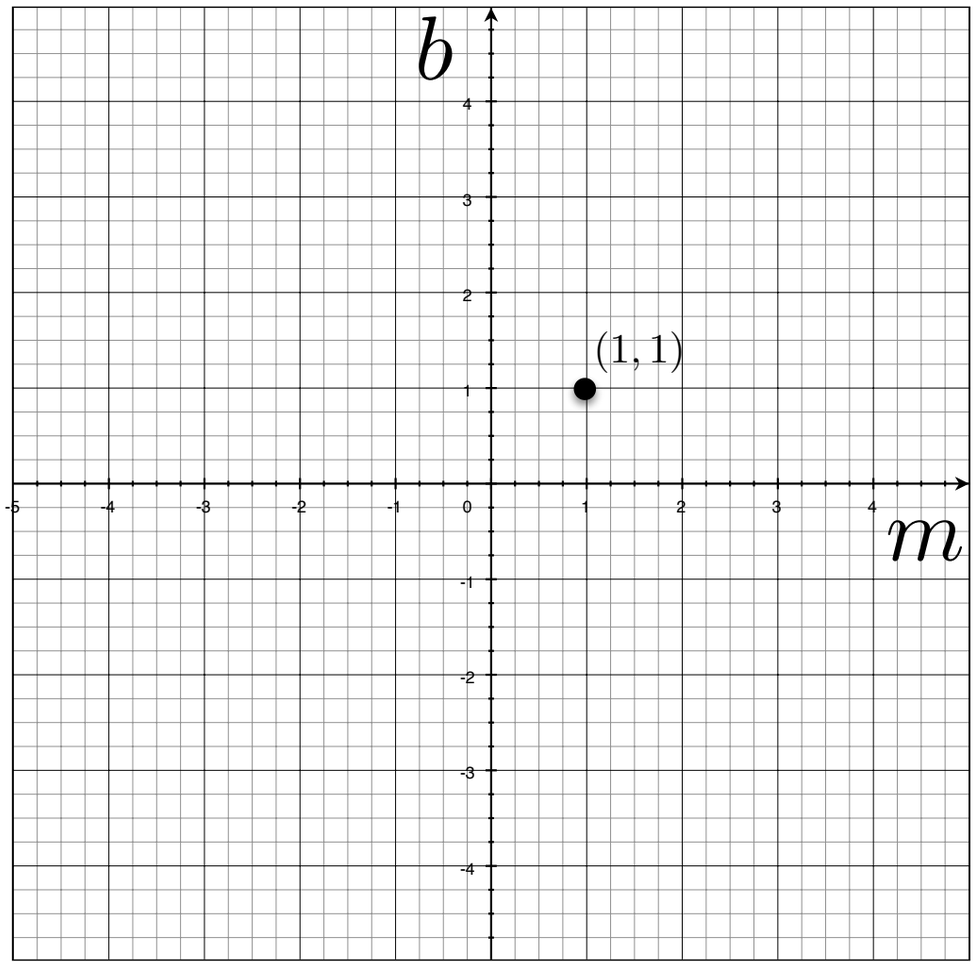


Image space

Parameter space

Image and parameter space

variables

$$y = mx + b$$

parameters

variables

$$y - mx = b$$

parameters

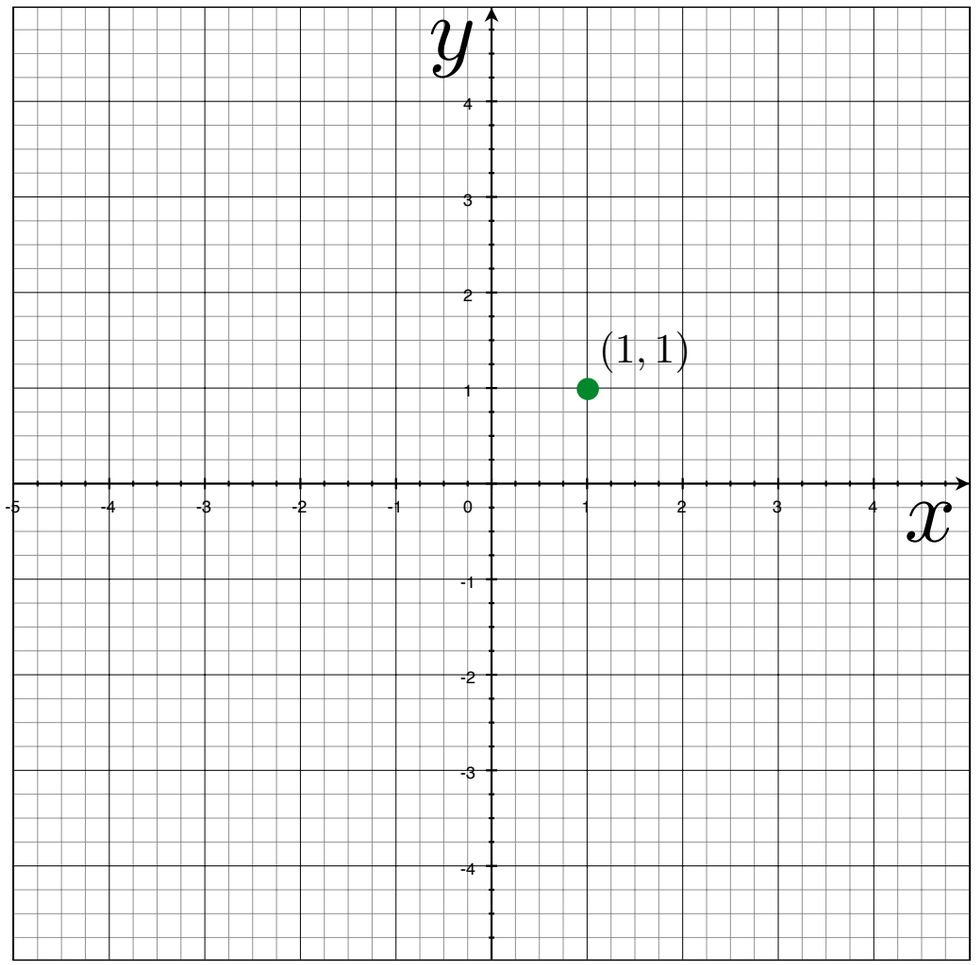
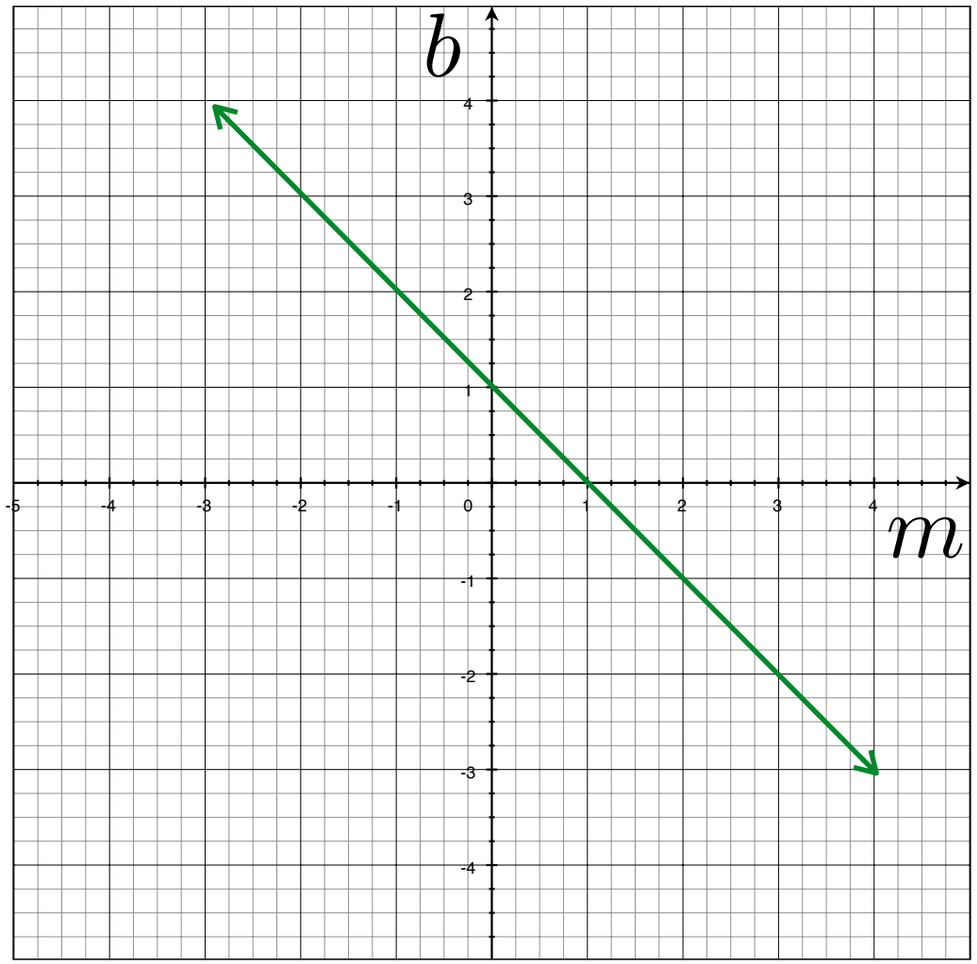


Image space

a point becomes a line



Parameter space

Image and parameter space

variables

$$y = mx + b$$

parameters

variables

$$y - mx = b$$

parameters

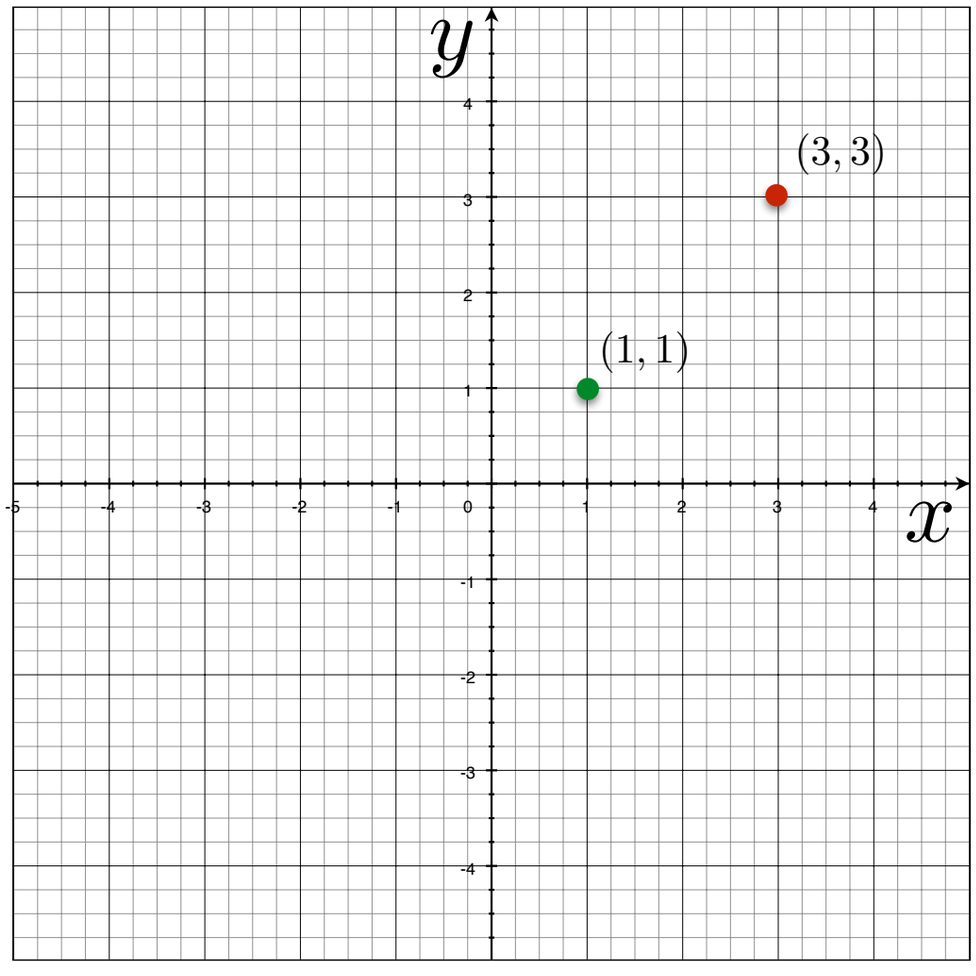
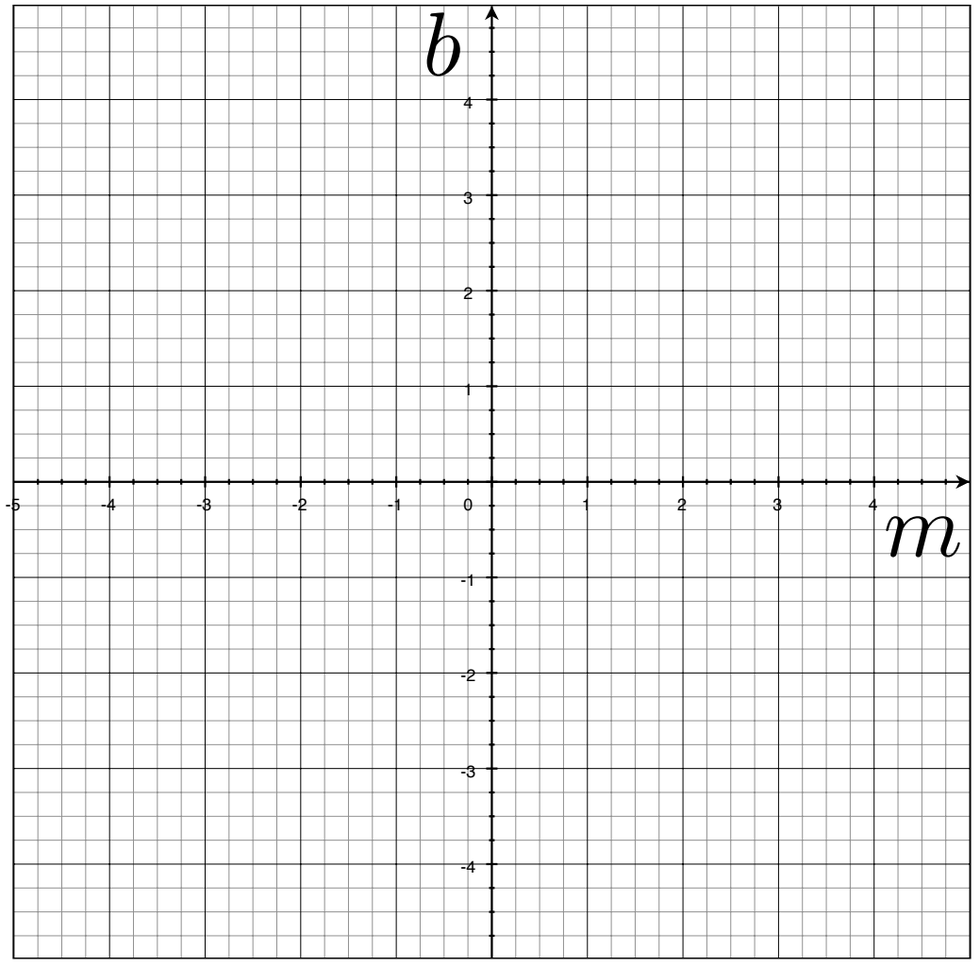


Image space

two points become ?



Parameter space

Image and parameter space

variables

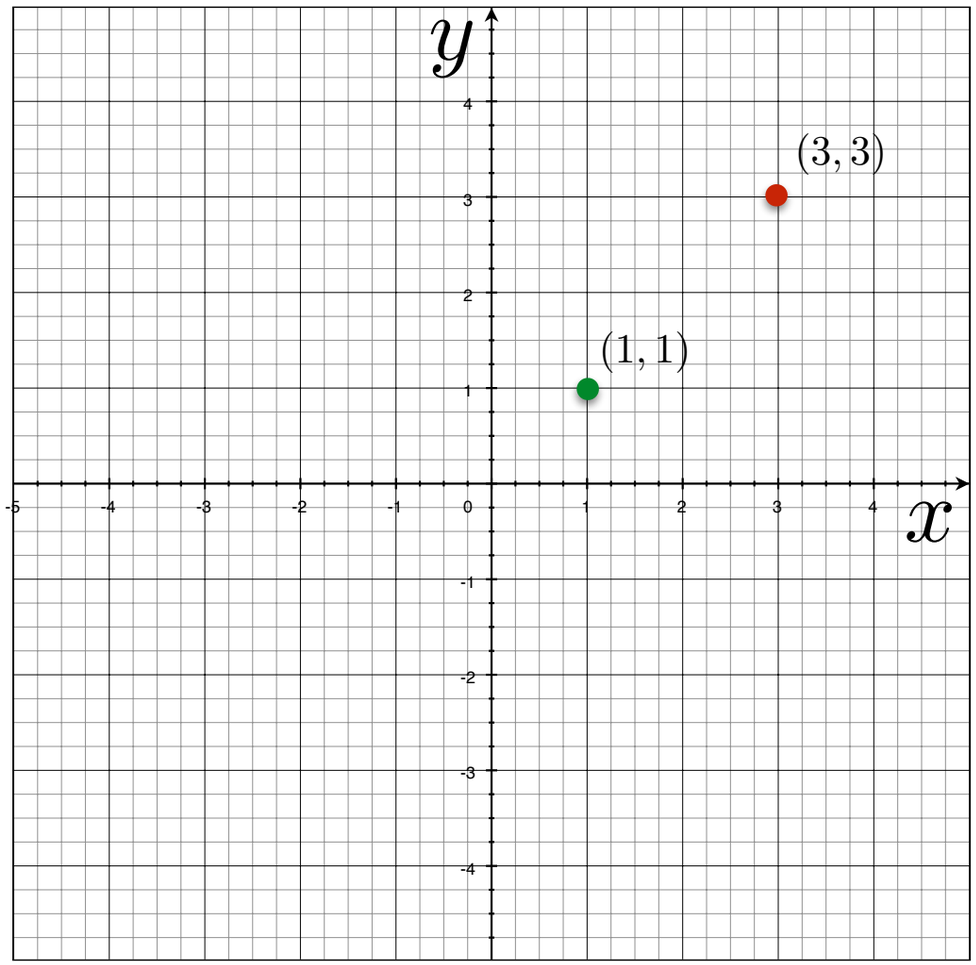
$$y = mx + b$$

parameters

variables

$$y - mx = b$$

parameters



two points become ?

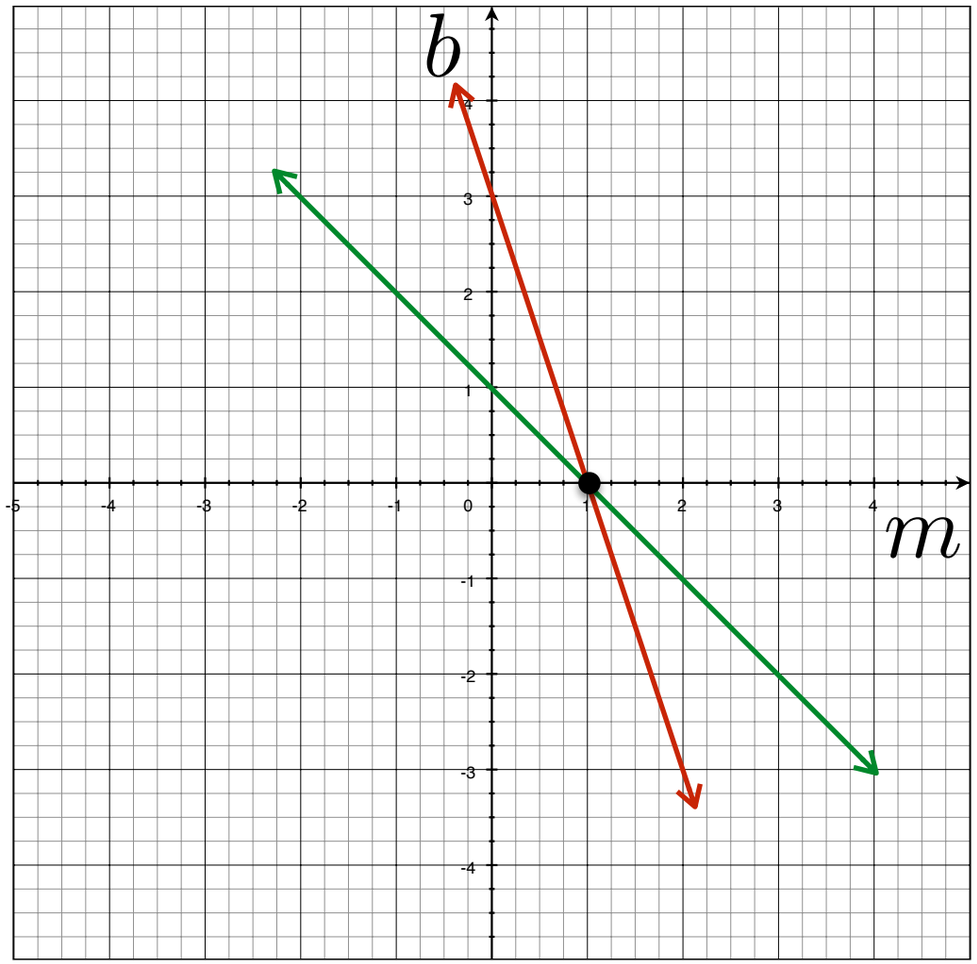


Image space

Parameter space

Image and parameter space

variables

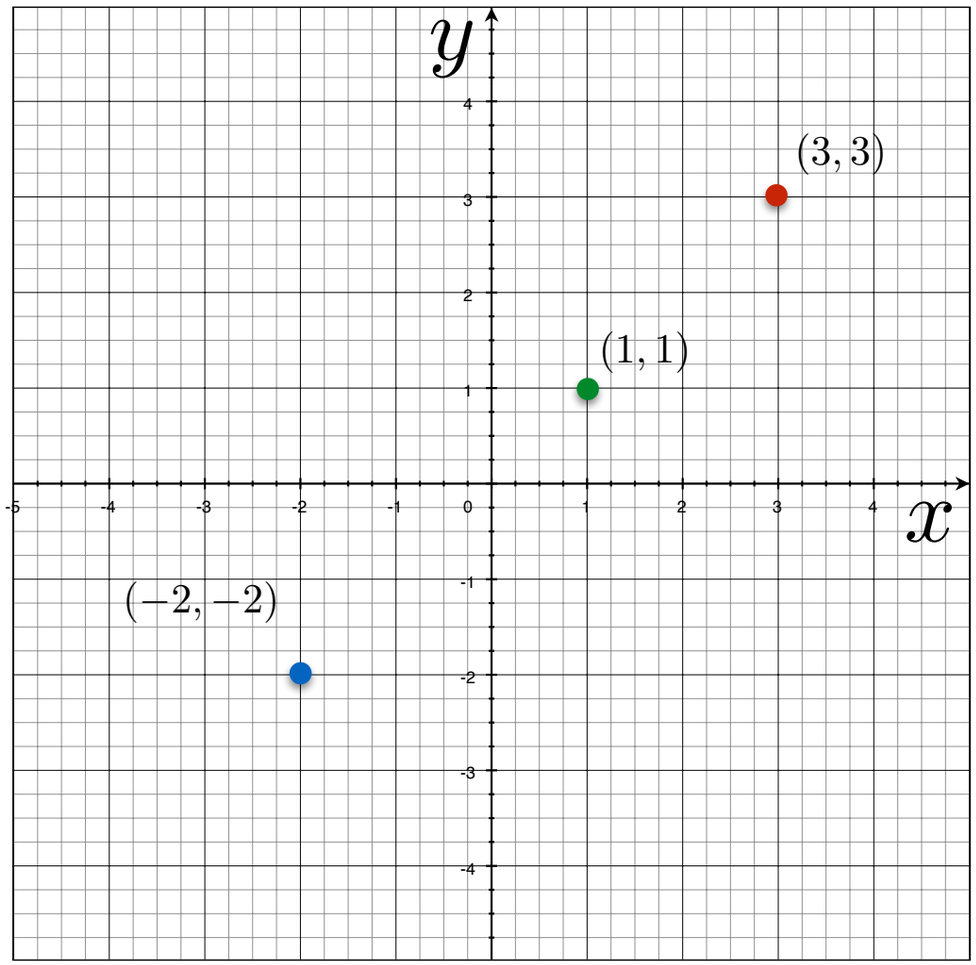
$$y = mx + b$$

parameters

variables

$$y - mx = b$$

parameters



three points become ?

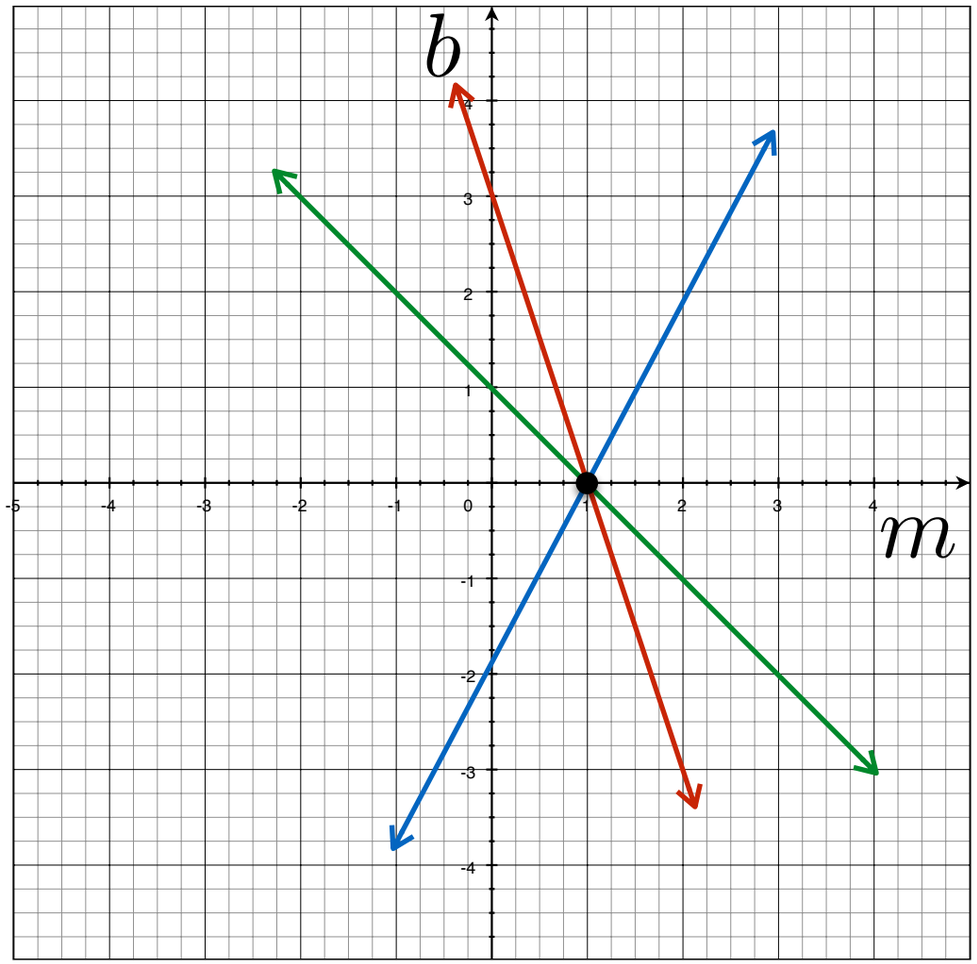


Image space

Parameter space

Image and parameter space

variables

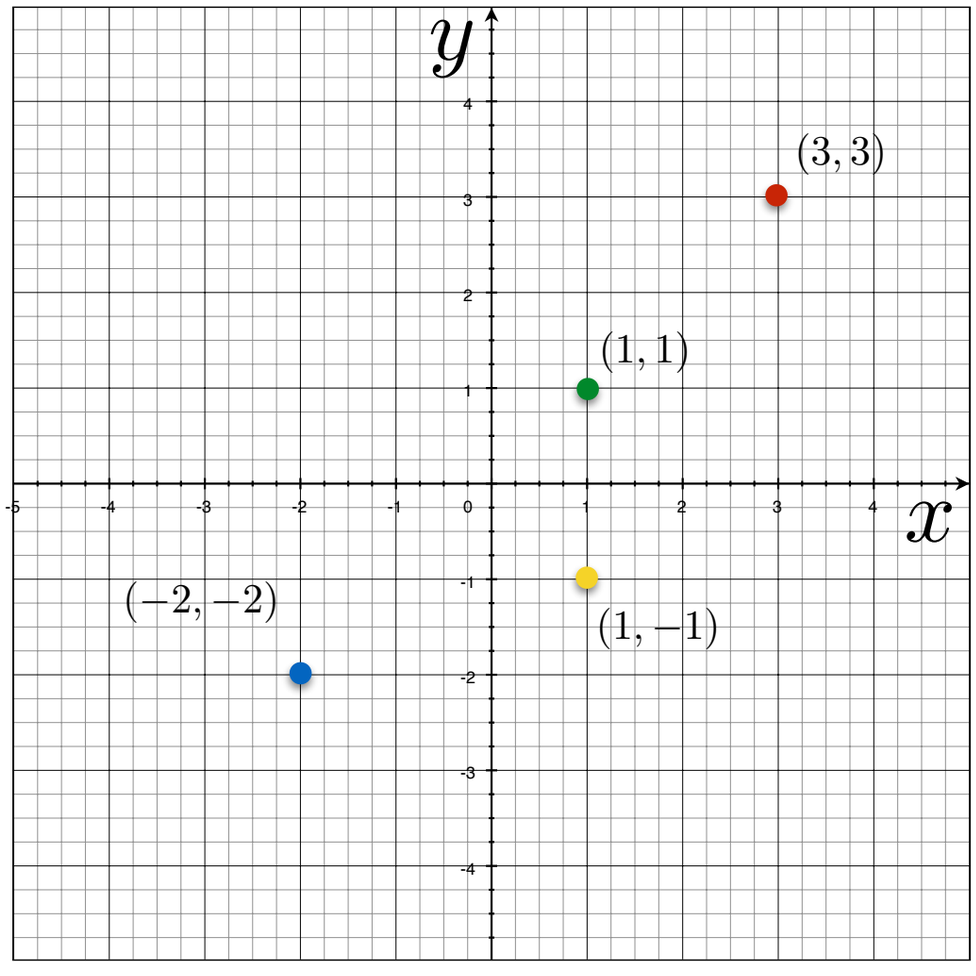
$$y = mx + b$$

parameters

variables

$$y - mx = b$$

parameters



four points become ?

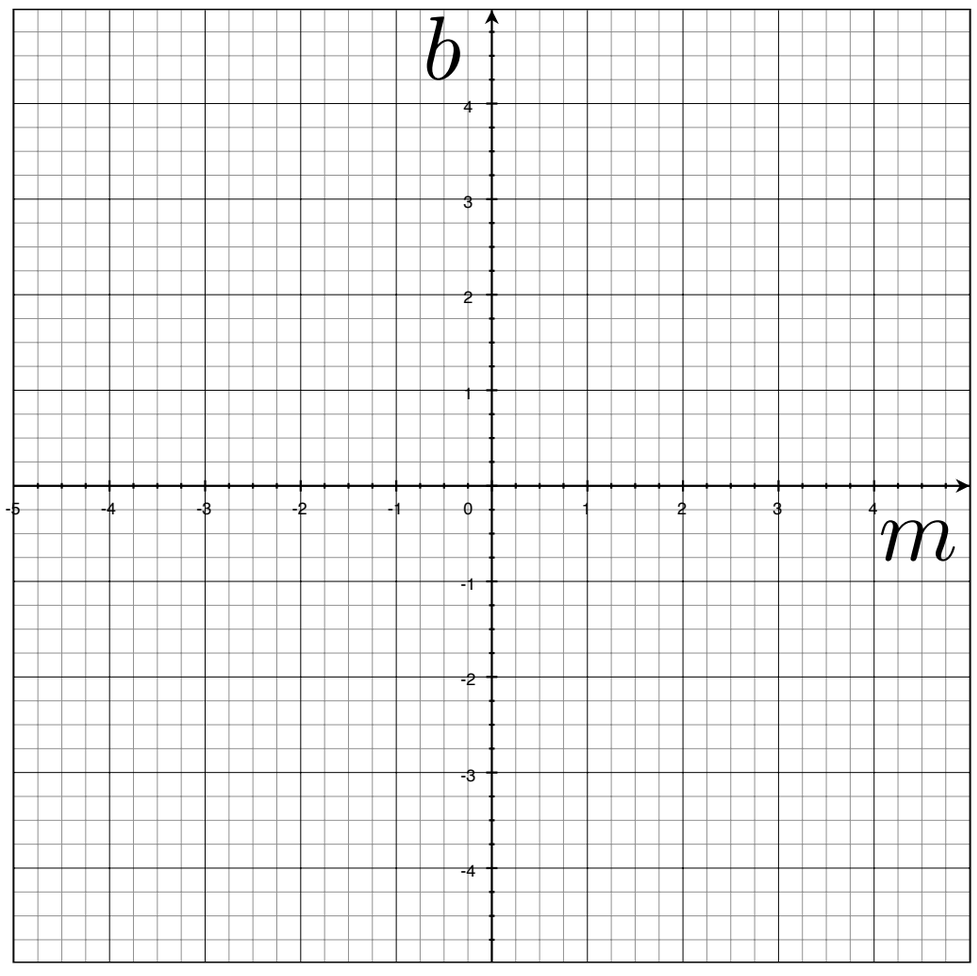


Image space

Parameter space

Image and parameter space

variables

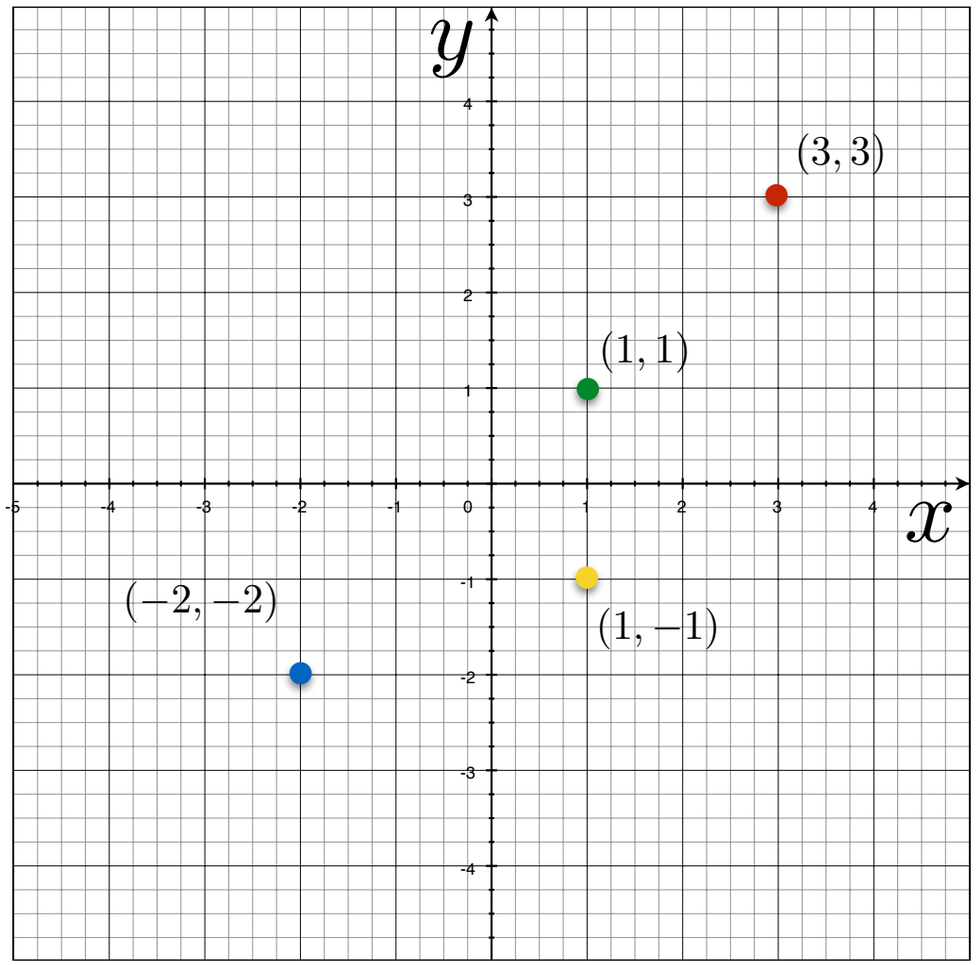
$$y = mx + b$$

parameters

variables

$$y - mx = b$$

parameters



four points become ?

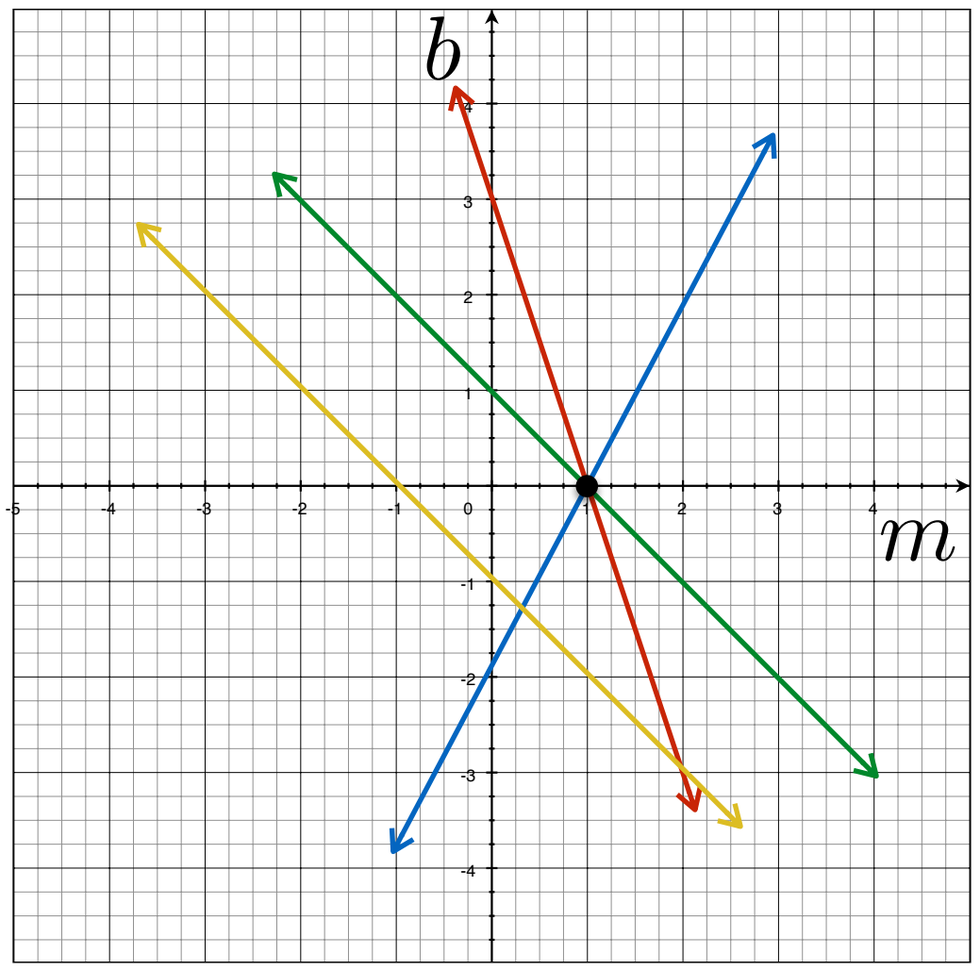


Image space

Parameter space

How would you find the best fitting line?

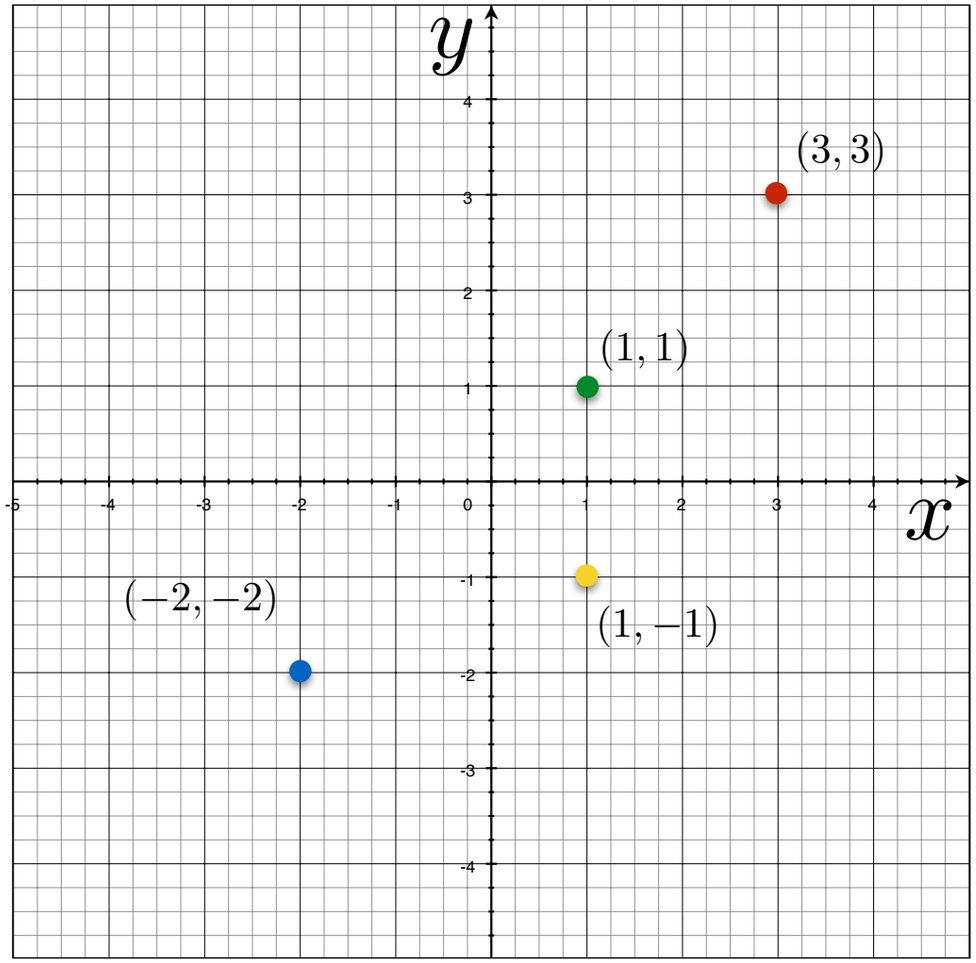
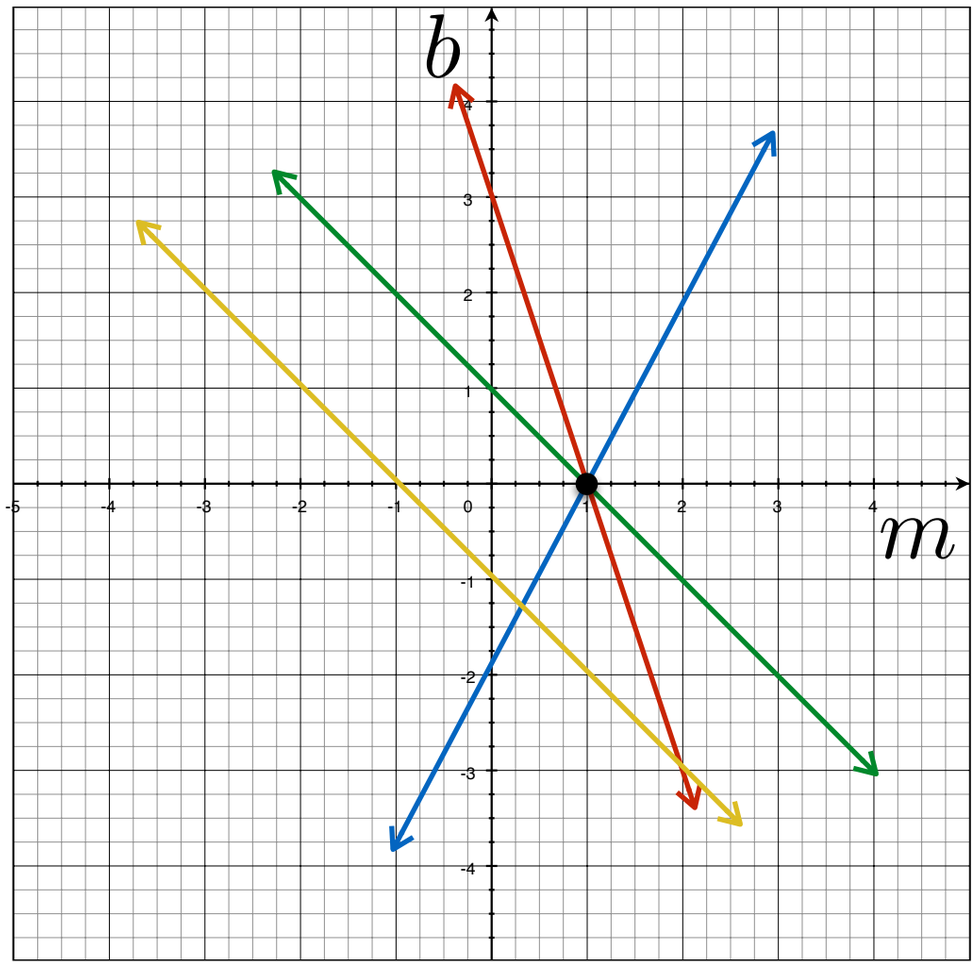


Image space



Parameter space

Is your method robust to measurement noise?

Is your method robust to outliers?

Line Detection by Hough Transform

Algorithm:

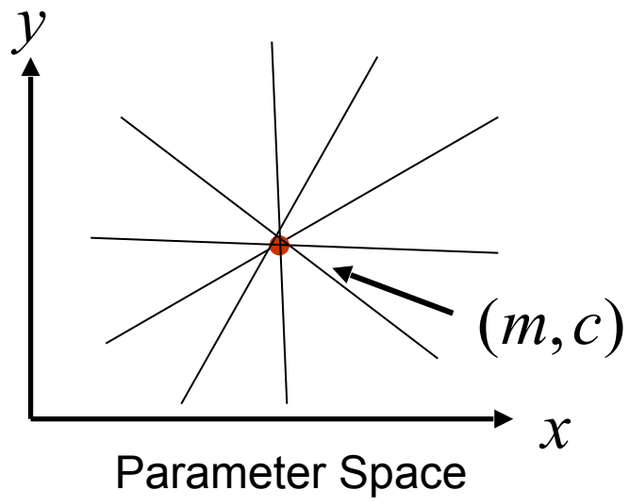
1. Quantize Parameter Space (m,c)

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 $A(m,c)$



$A(m,c)$

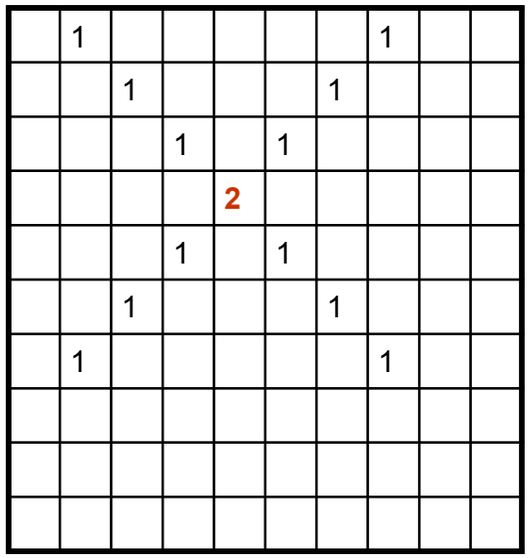
	1						1		
		1				1			
			1		1				
				2					
			1		1				
		1				1			
	1						1		

Problems with parameterization

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

How big does the accumulator need to be?

$A(m, c)$

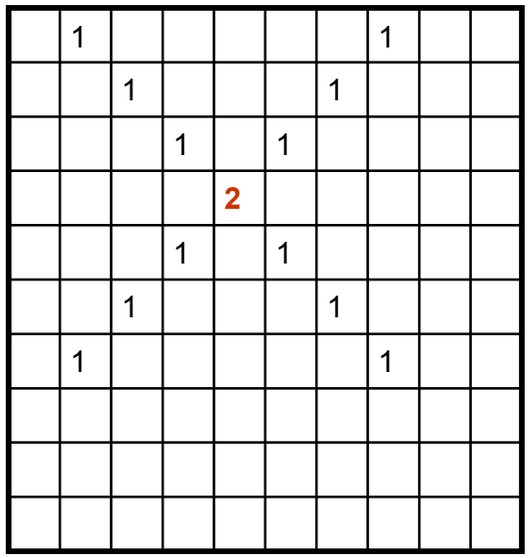


Problems with parameterization

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

How big does the accumulator need to be?

$A(m, c)$



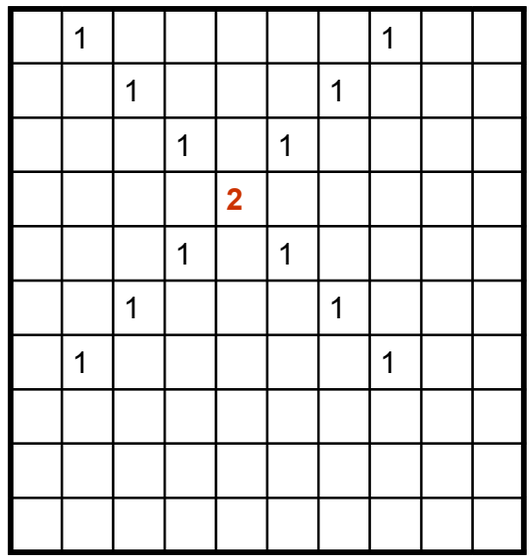
The space of m is huge! The space of c is huge!

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!

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

Better Parameterization

Use normal form:

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

Given points (x_i, y_i) find (ρ,θ)

Hough Space Sinusoid

$$0 \le \theta \le 2\pi$$

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

(Finite Accumulator Array Size)

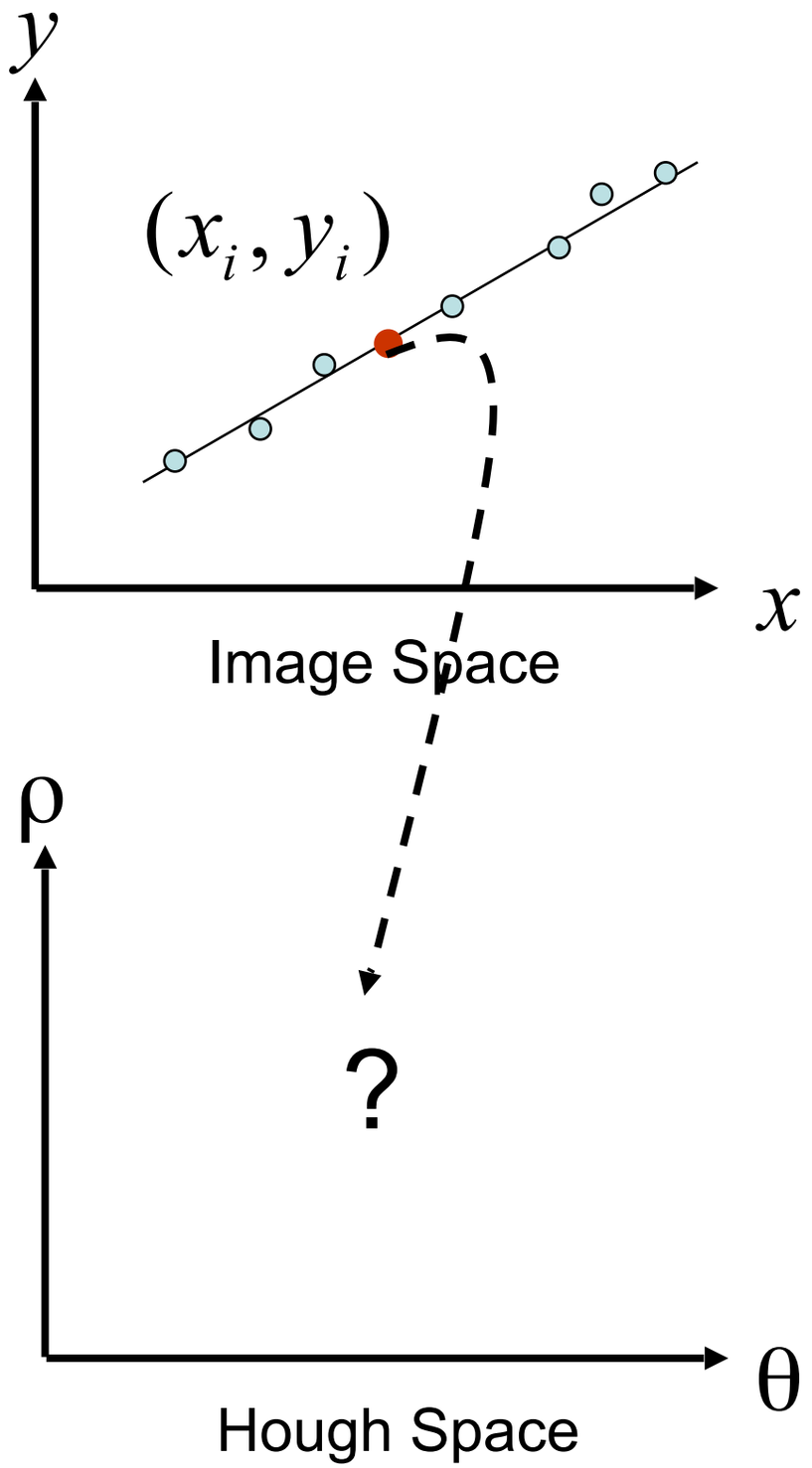


Image and parameter space

variables

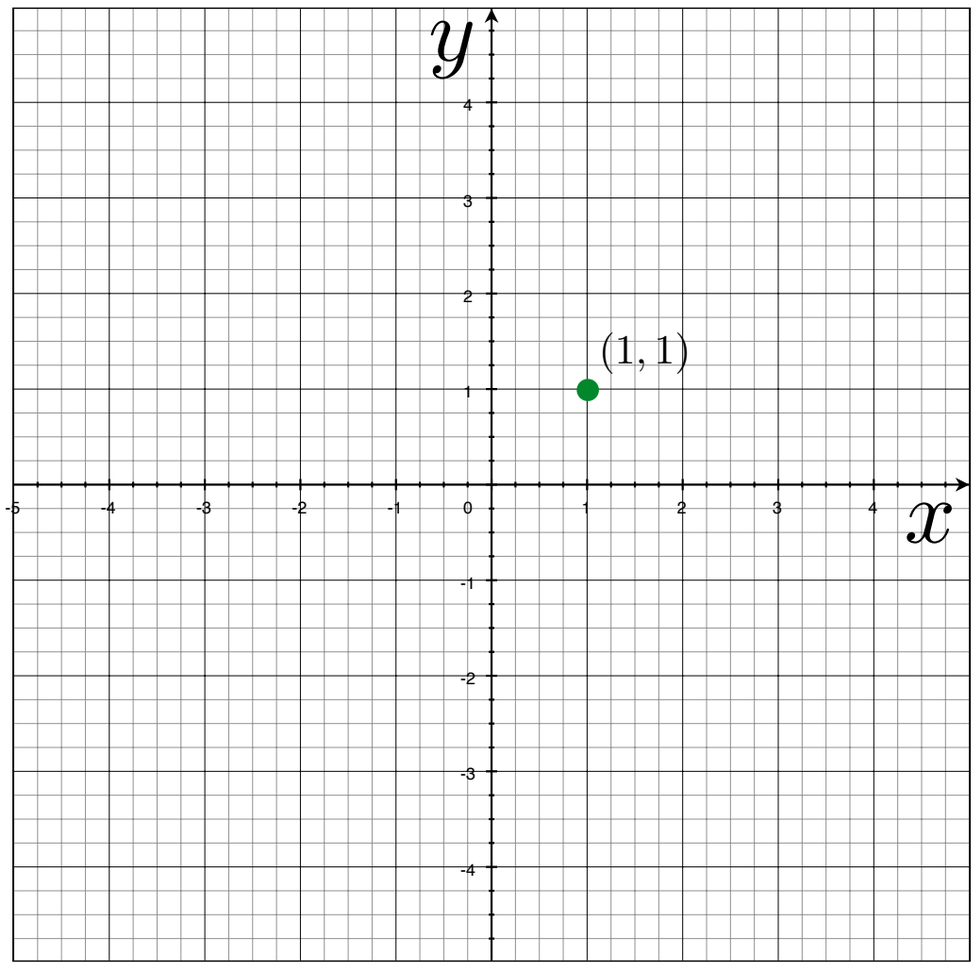
$$y = mx + b$$

parameters

parameters

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

variables



a point becomes a wave

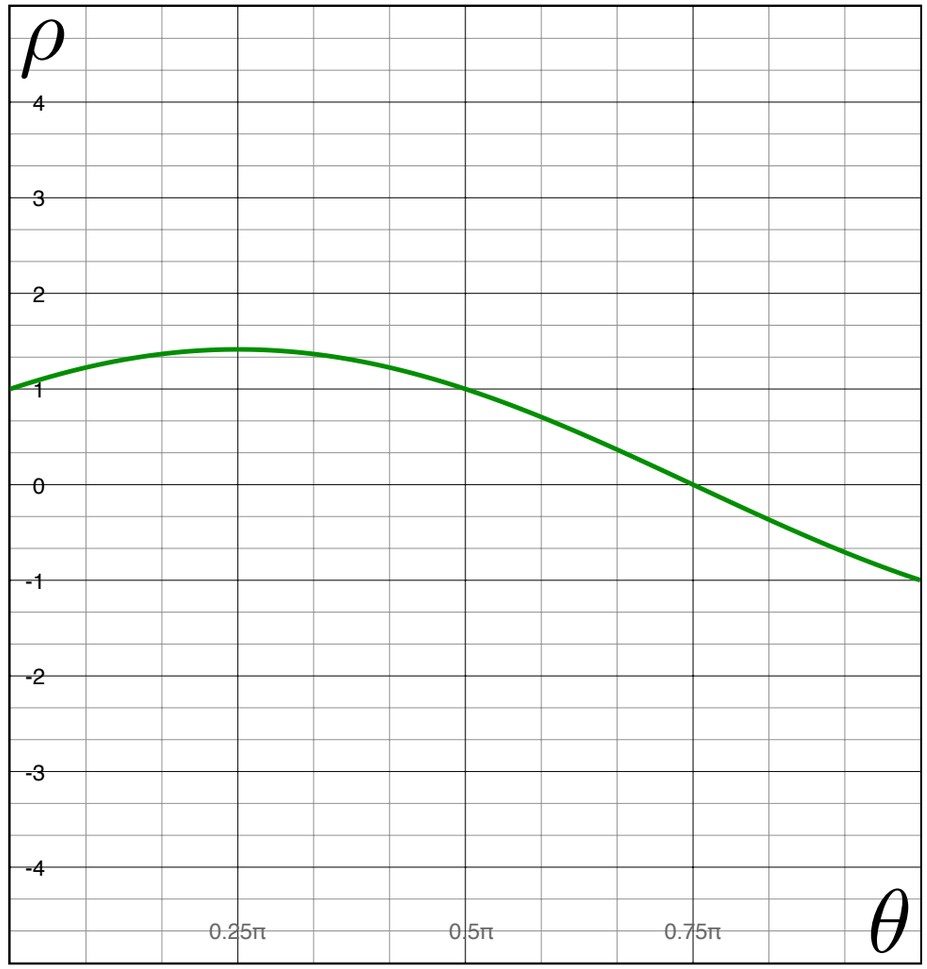


Image space

Parameter space

Image and parameter space

variables

$$y = mx + b$$

parameters

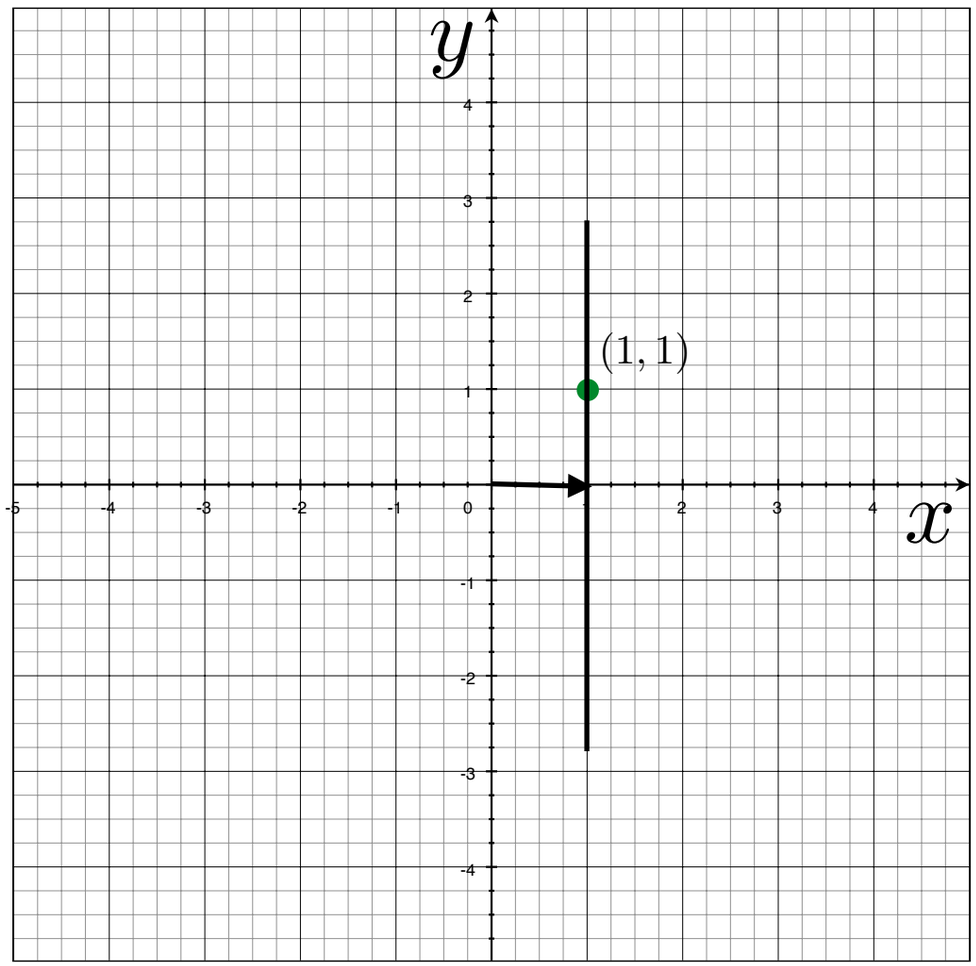
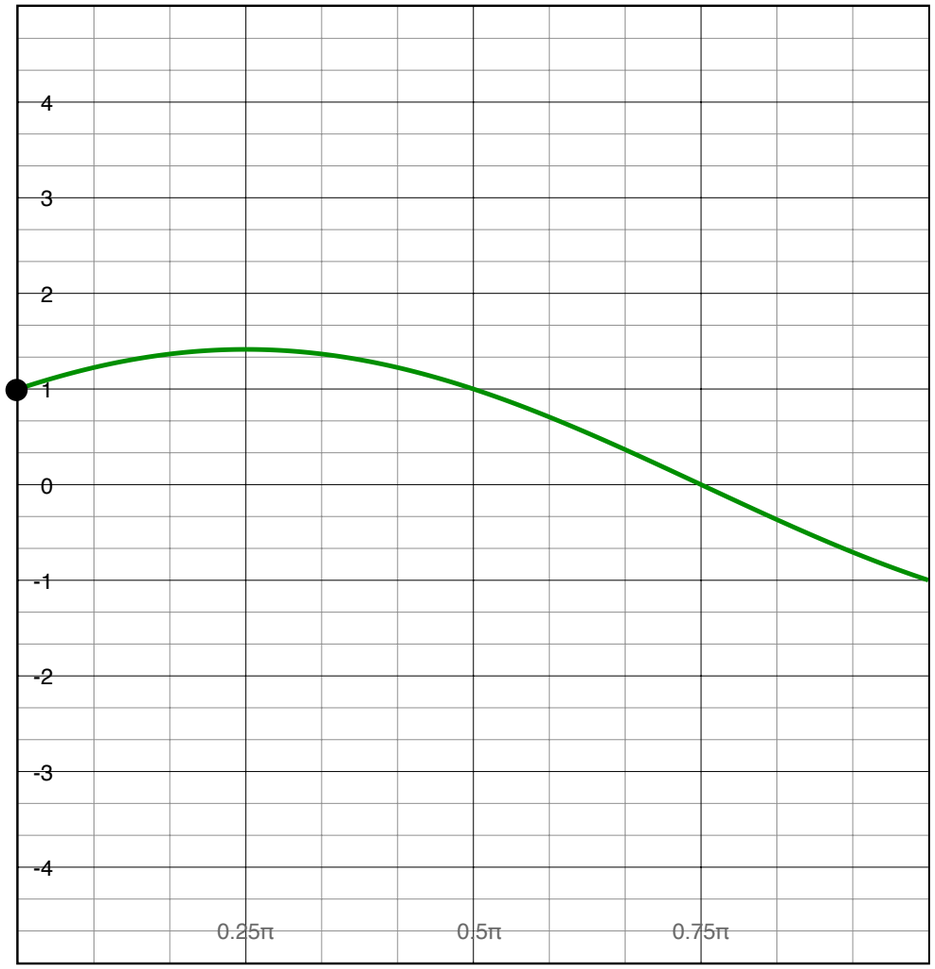


Image space

a line becomes a point



Parameter space

Image and parameter space

variables

$$y = mx + b$$

parameters

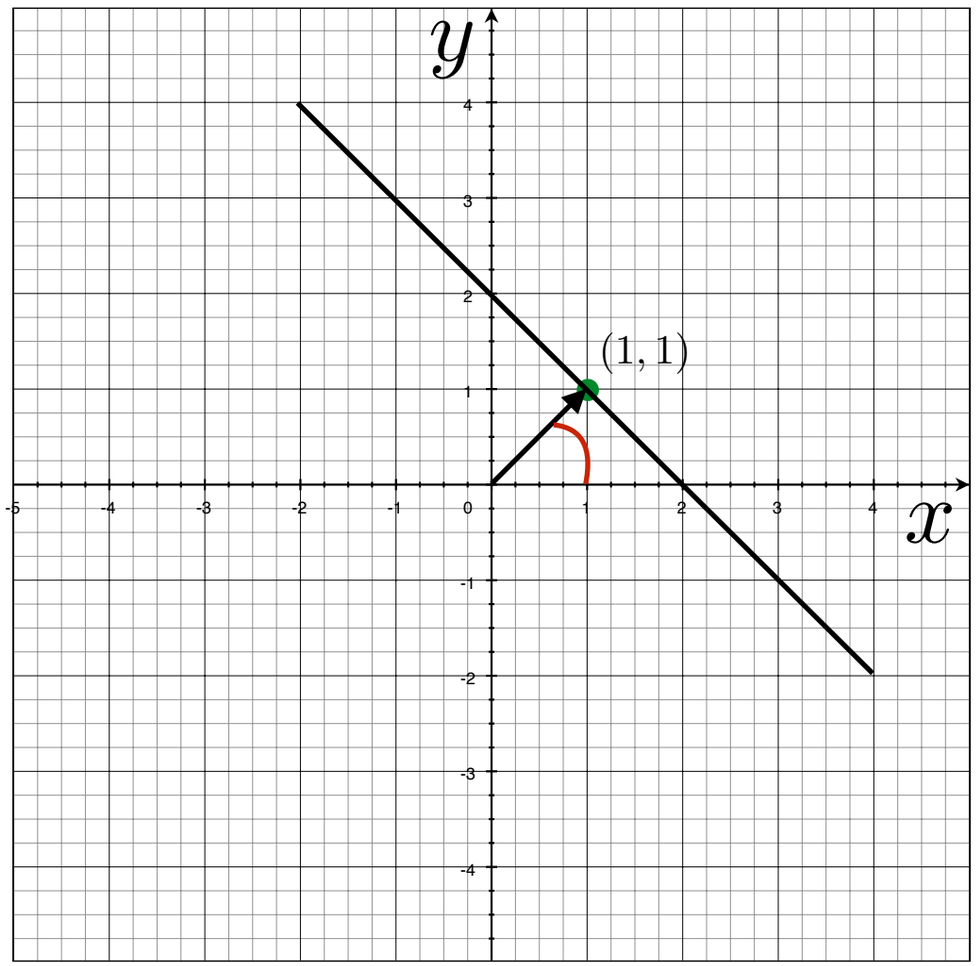


Image space

a line becomes a point

Parameter space

Image and parameter space

variables

$$y = mx + b$$

parameters

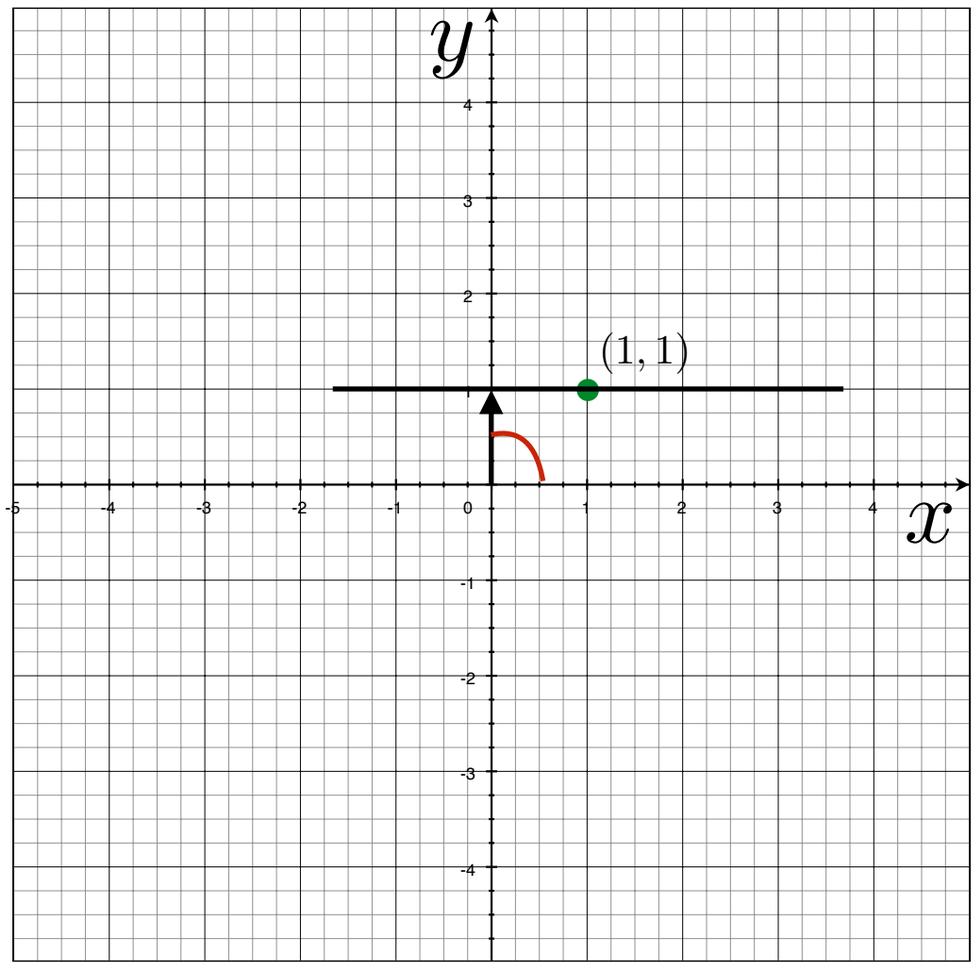
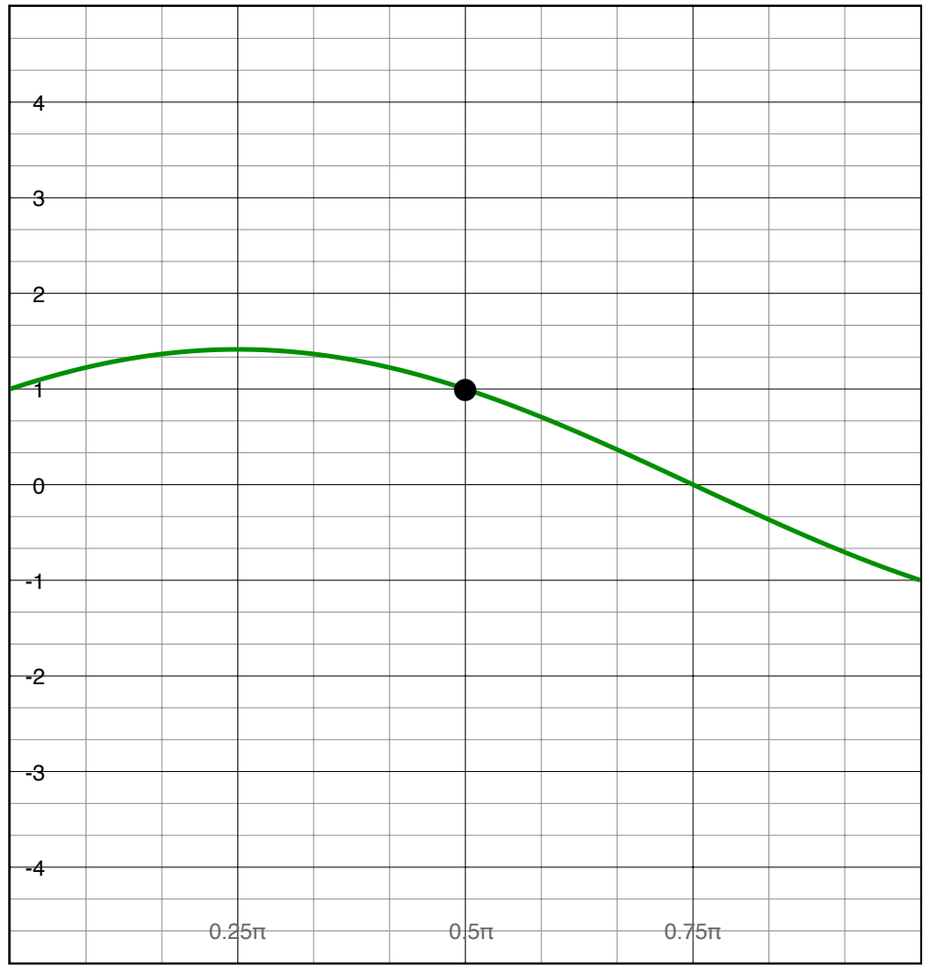


Image space

a line becomes a point



Parameter space

Image and parameter space

variables

$$y = mx + b$$

parameters

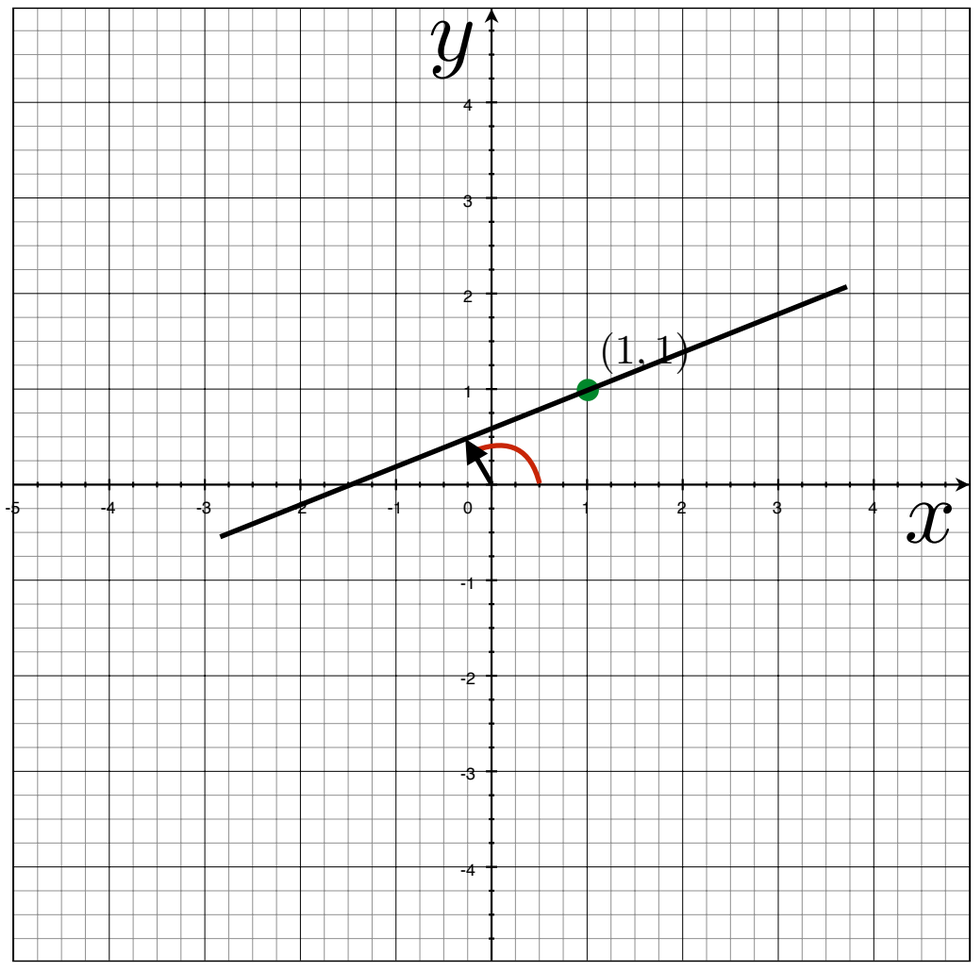
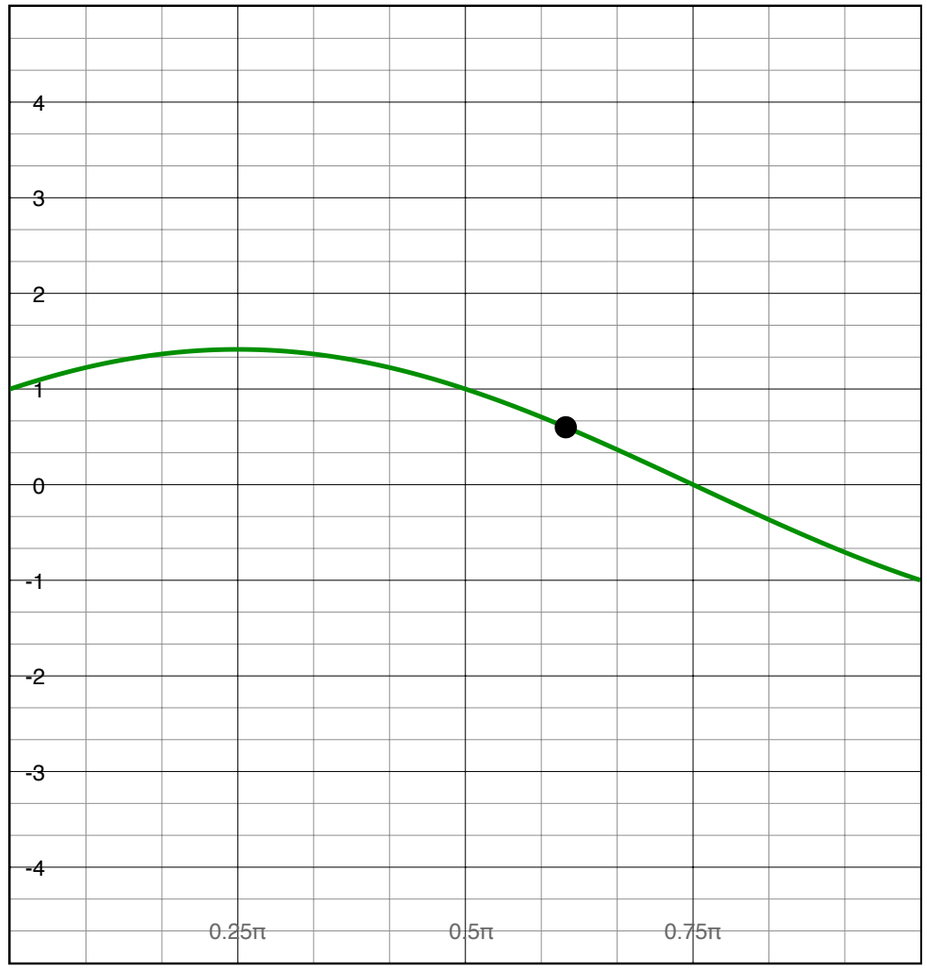


Image space

a line becomes a point



Parameter space

Image and parameter space

variables

$$y = mx + b$$

parameters

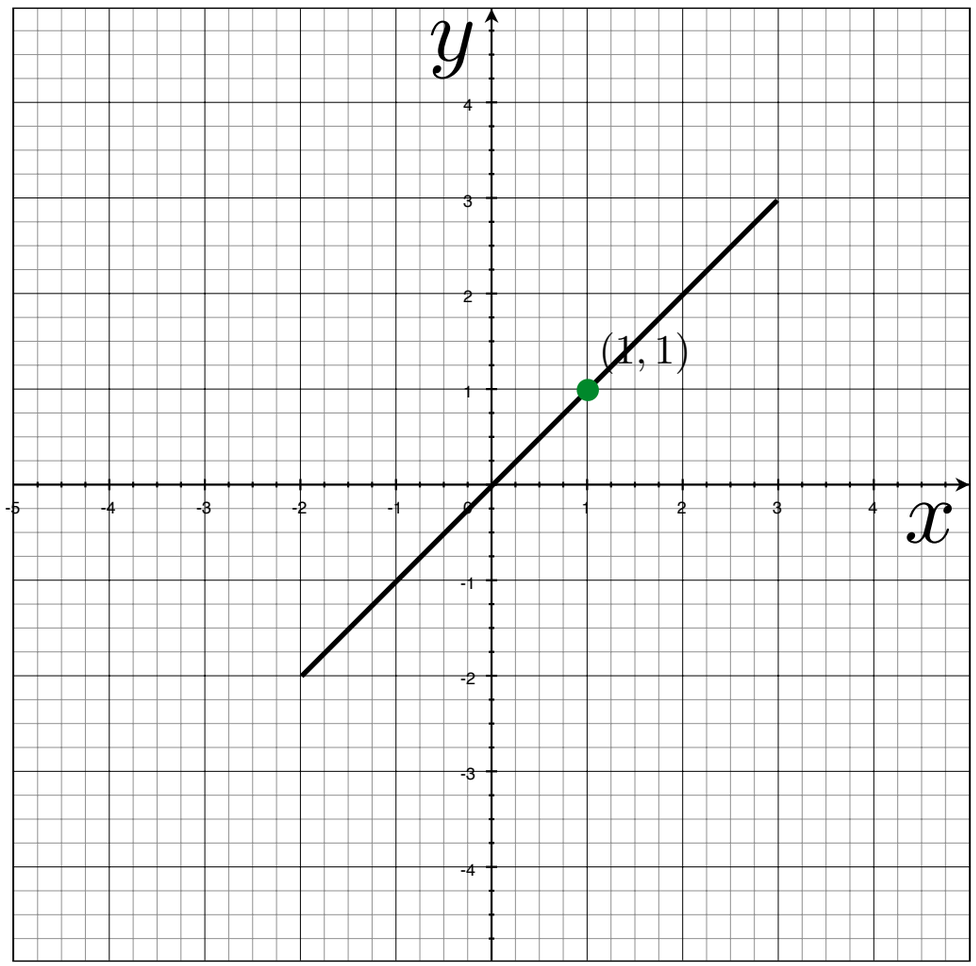
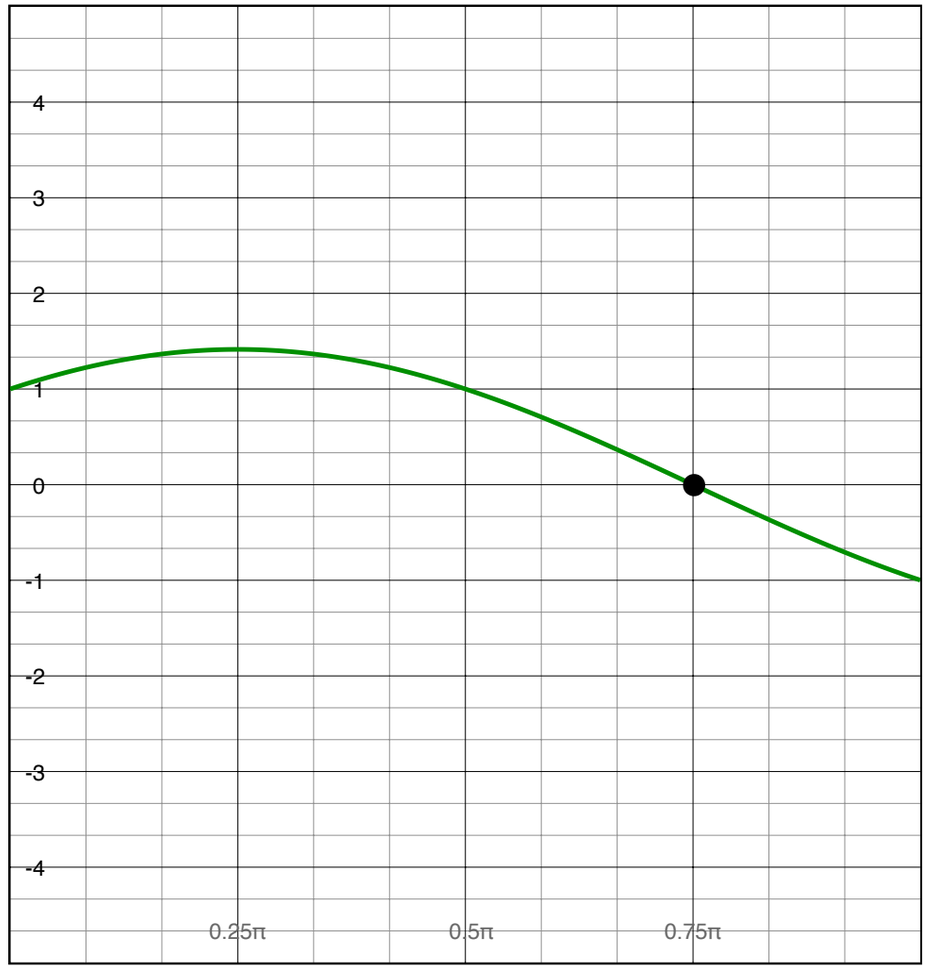


Image space

a line becomes a point



Parameter space

Image and parameter space

variables

$$y = mx + b$$

parameters

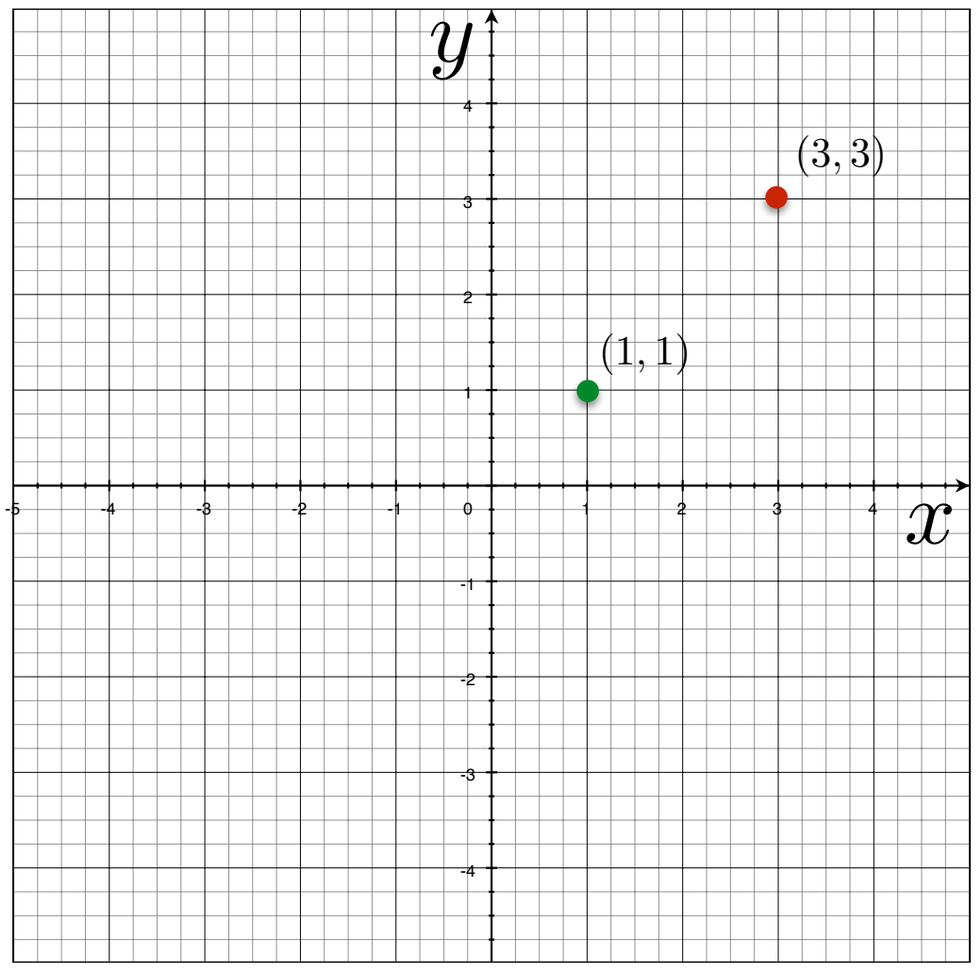
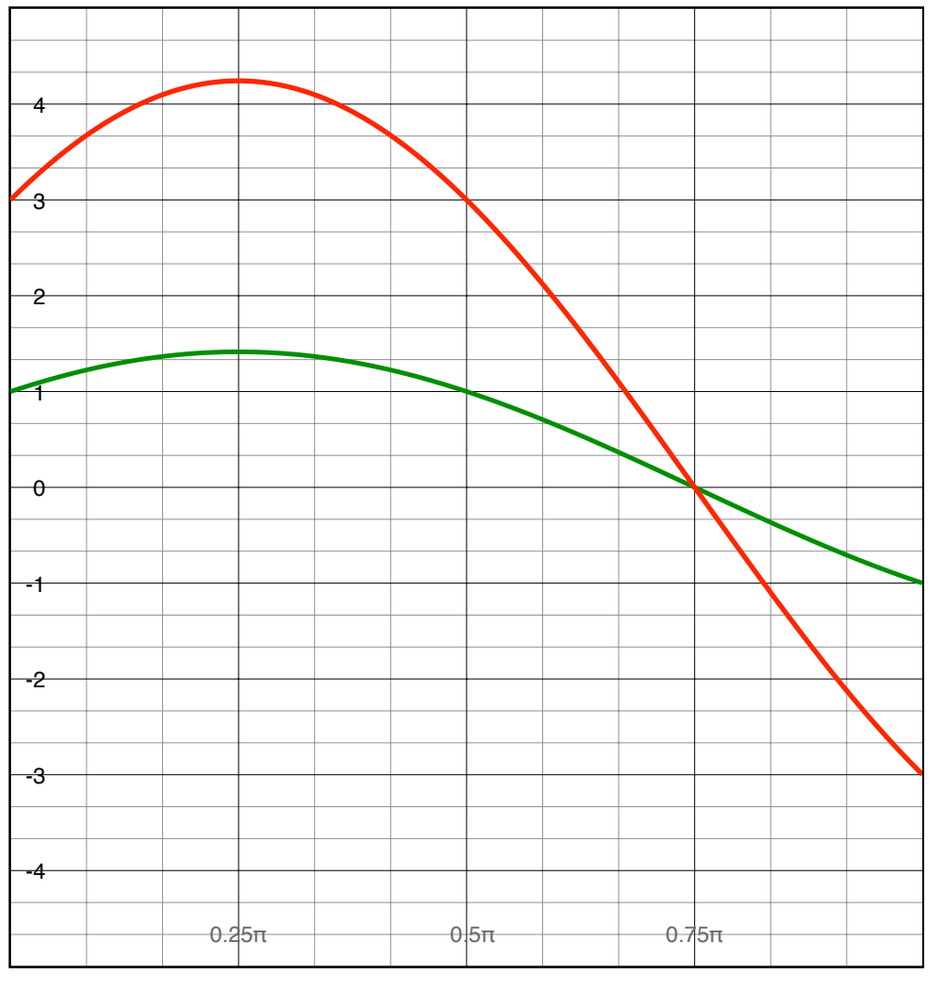


Image space

two points become ?



Parameter space

Image and parameter space

variables

$$y = mx + b$$

parameters

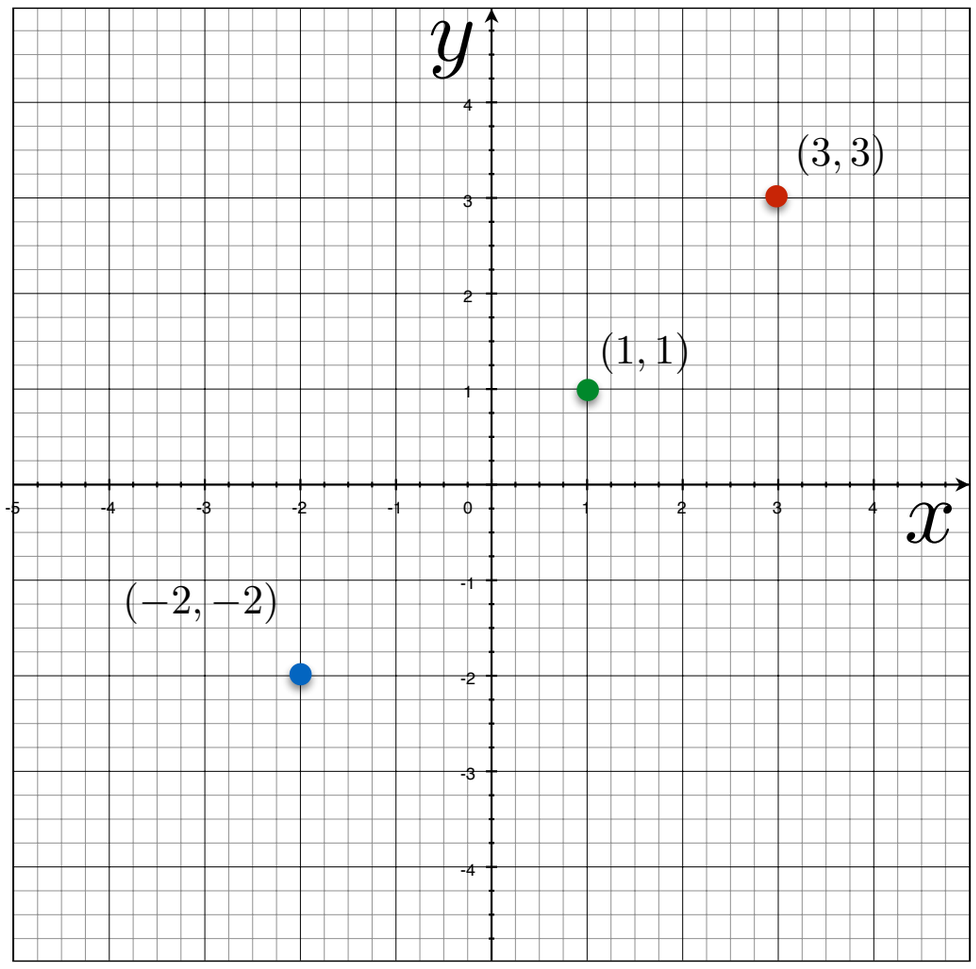
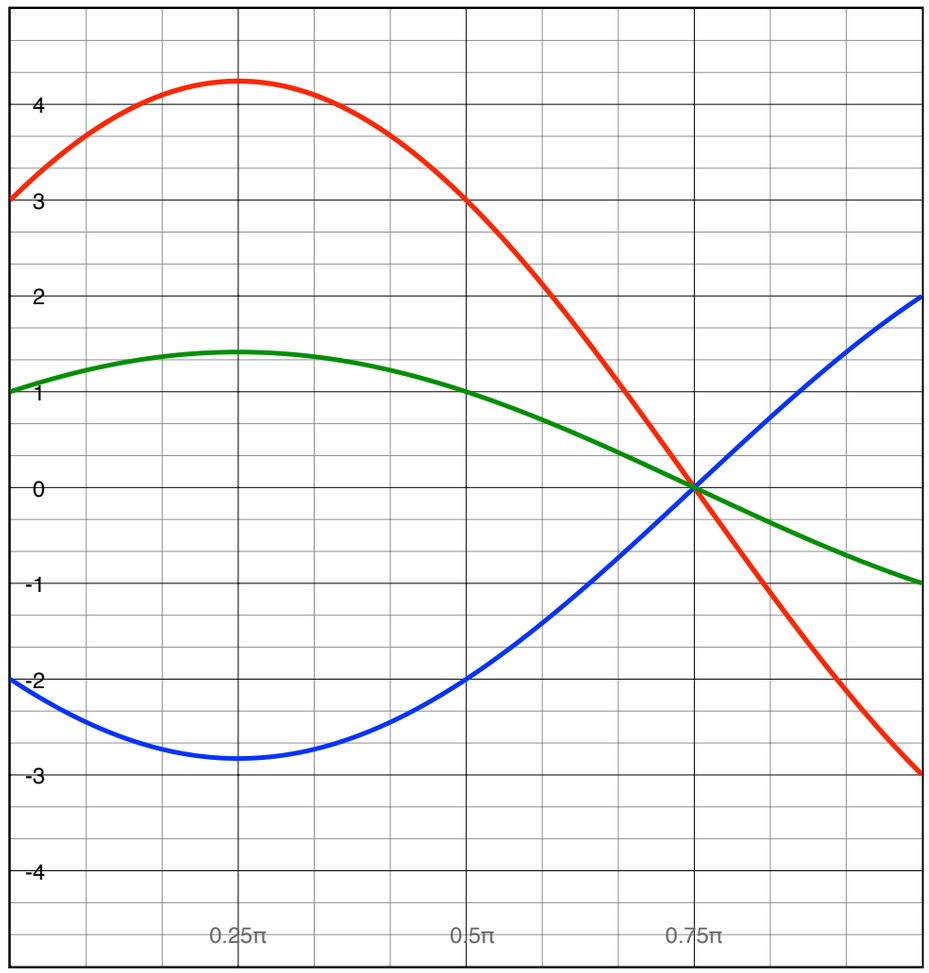


Image space

three points become ?



Parameter space

Image and parameter space

variables

$$y = mx + b$$

parameters

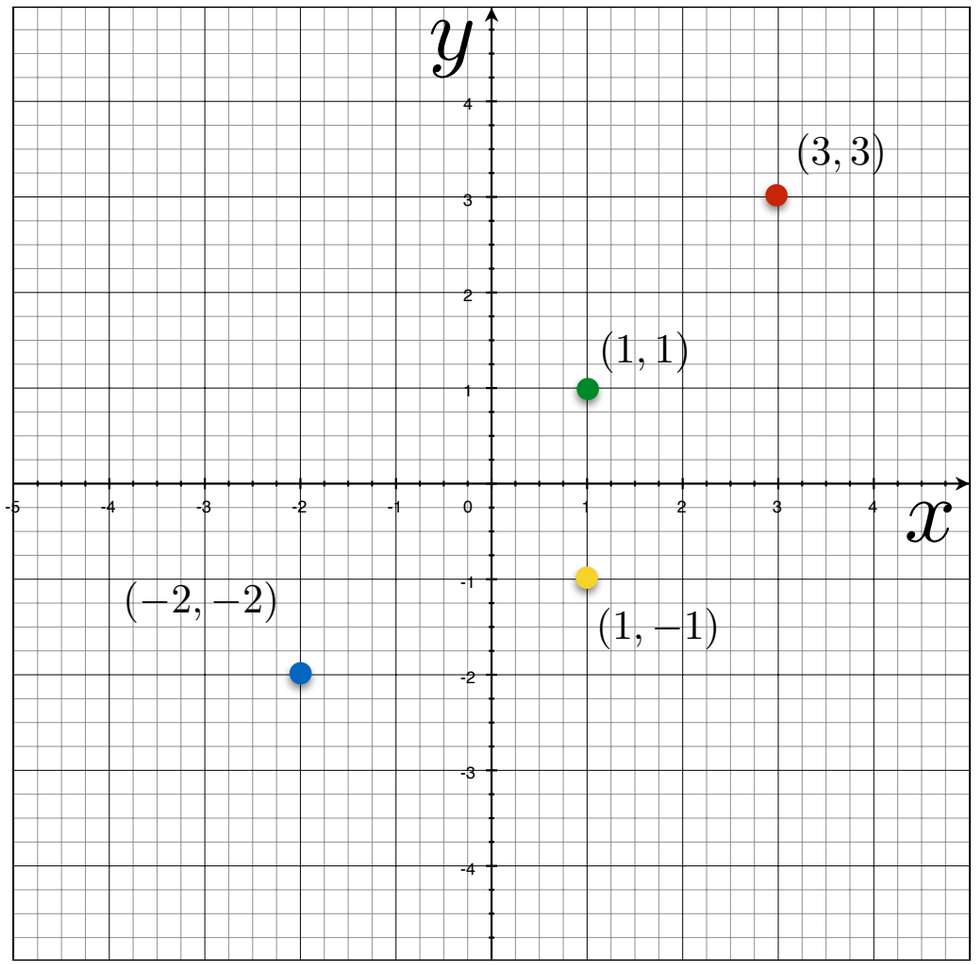
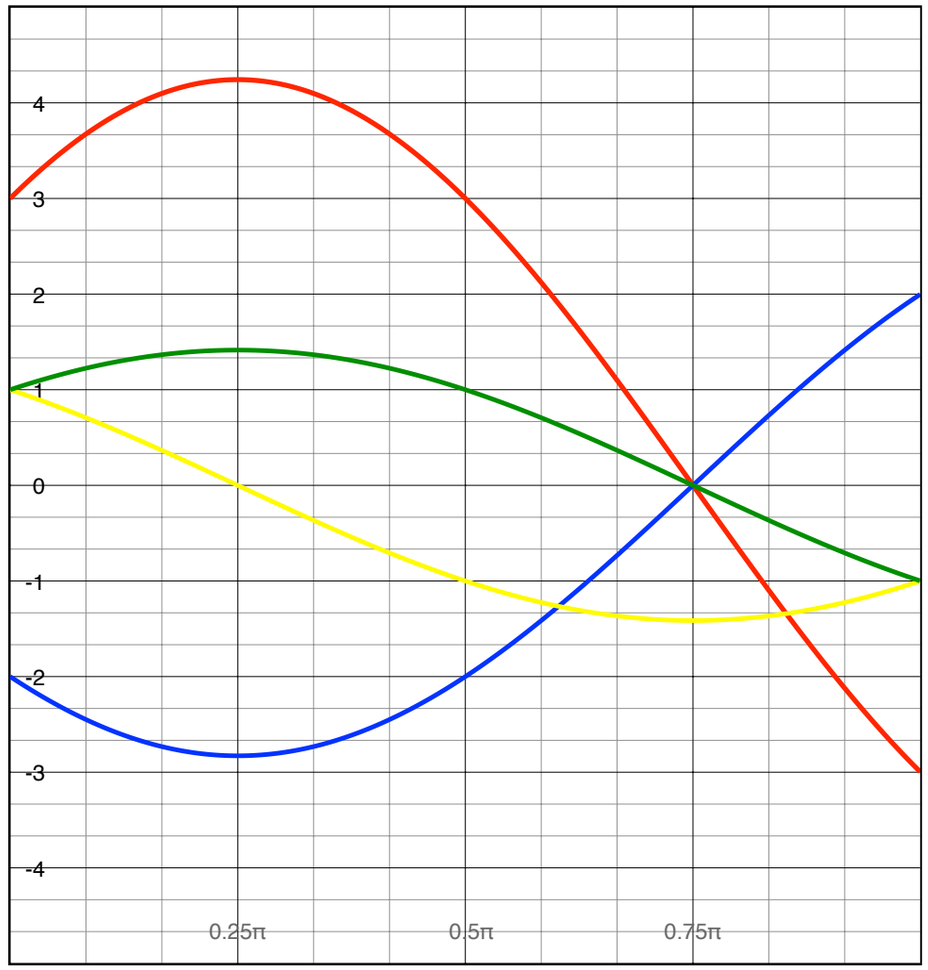


Image space

four points become ?



Parameter space

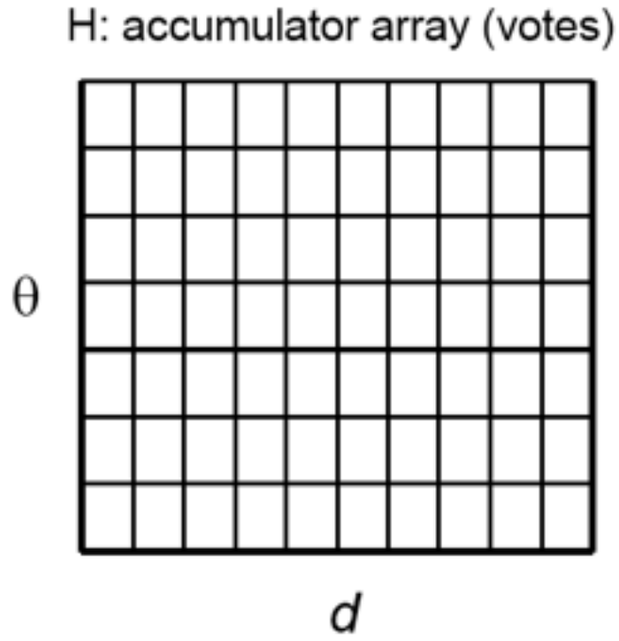
implementation

1. Initialize accumulator H to all zeros

2. For each edge point (x,y) in the image
   For θ = 0 to 180
      ρ = x cos θ + y sin θ
      H(θ, ρ) = H(θ, ρ) + 1
   end
end

3. Find the value(s) of (θ, ρ) where H(θ, ρ) is a local maximum

4. The detected line in the image is given by
   ρ = x cos θ + y sin θ



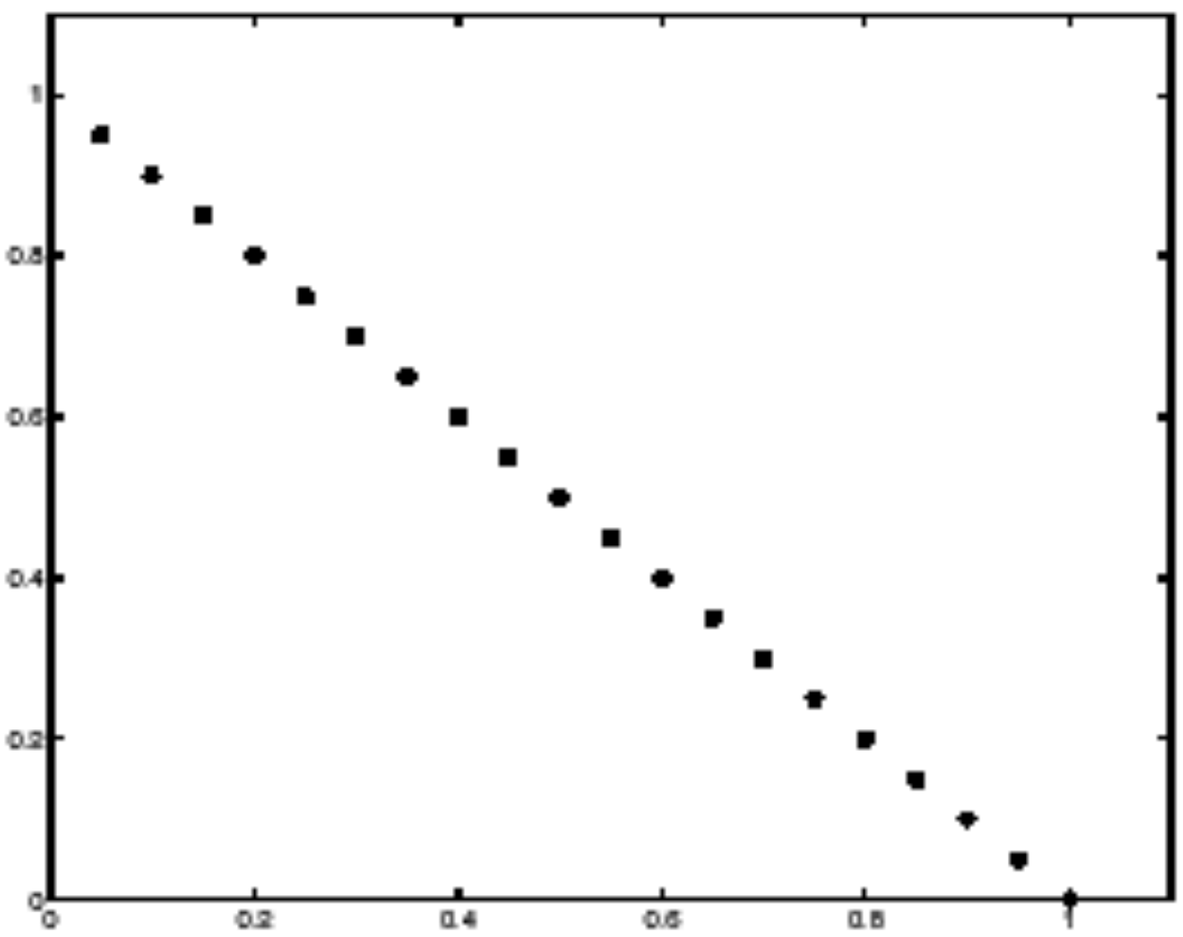
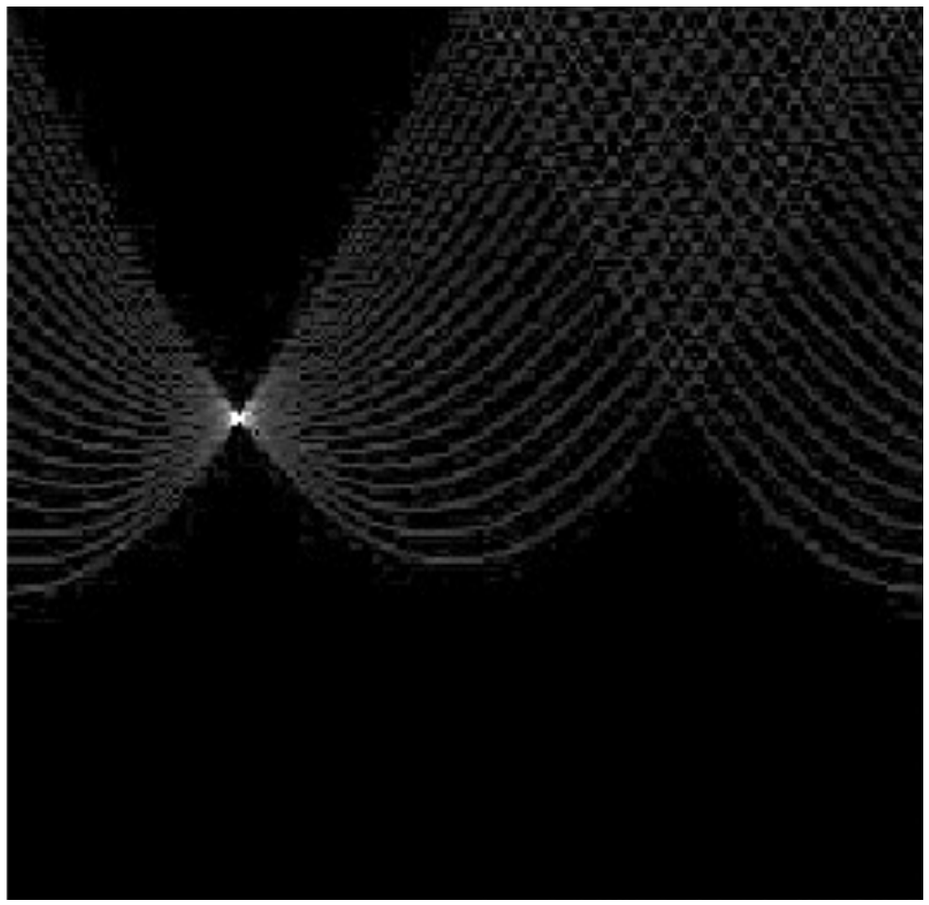


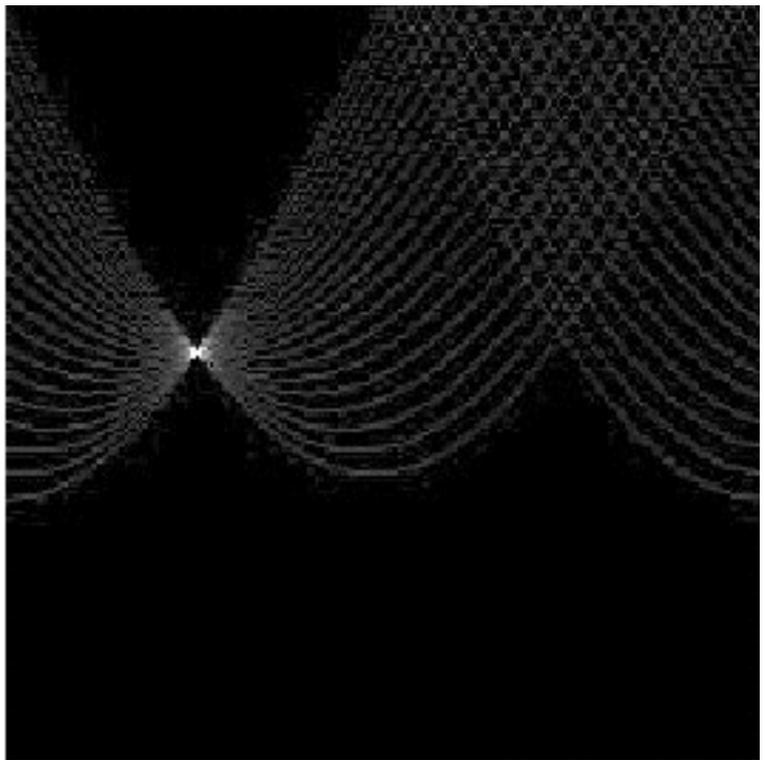
Image space

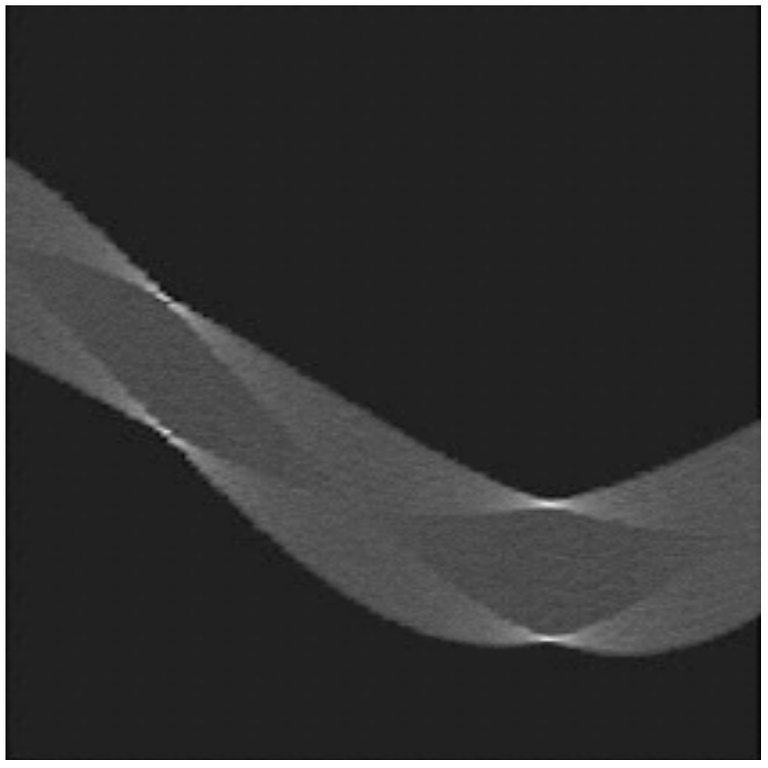


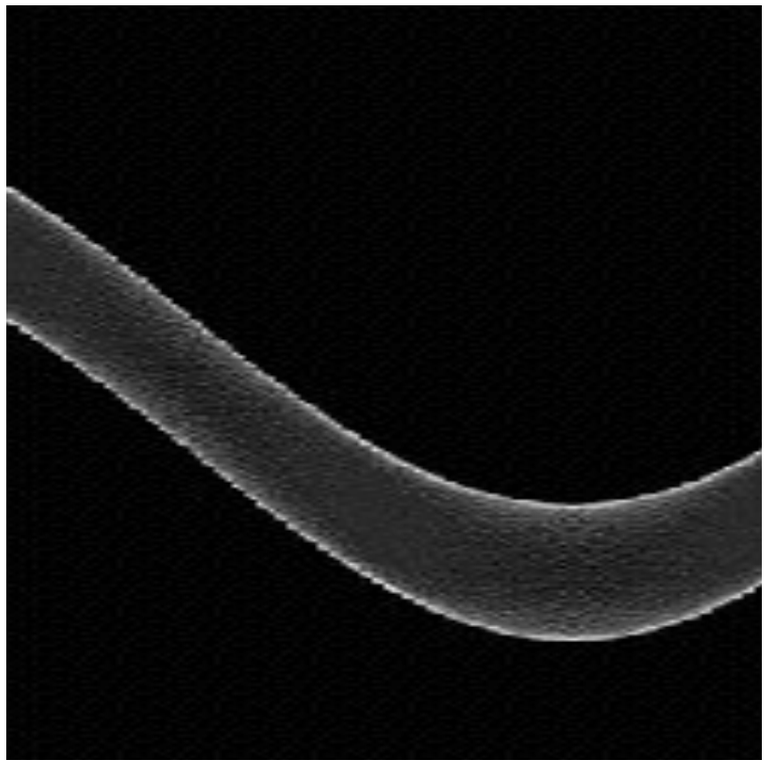
Votes

Basic shapes

(in polar parameter space)



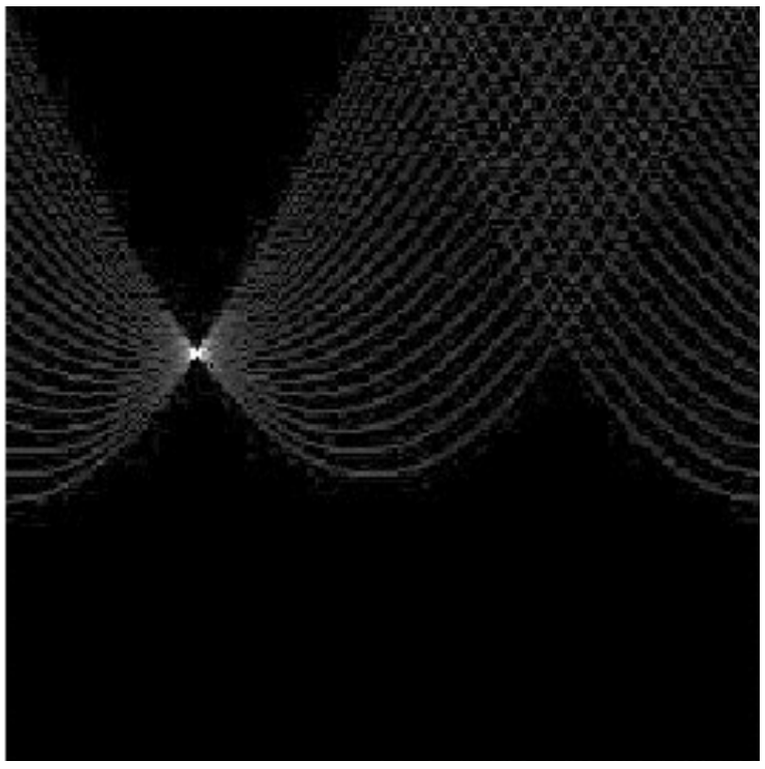




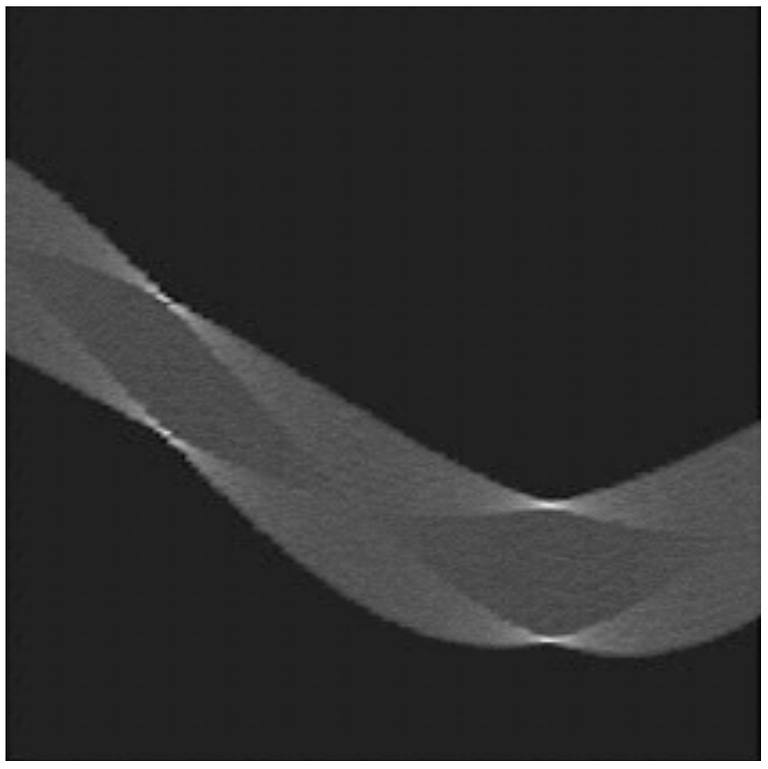
can you guess the shape?

Basic shapes

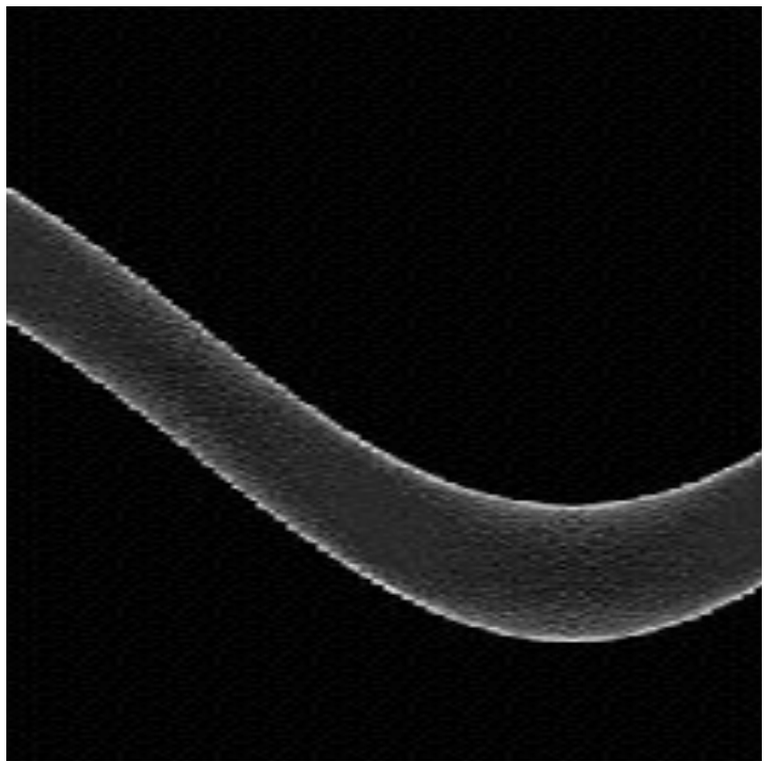
(in polar parameter space)



line

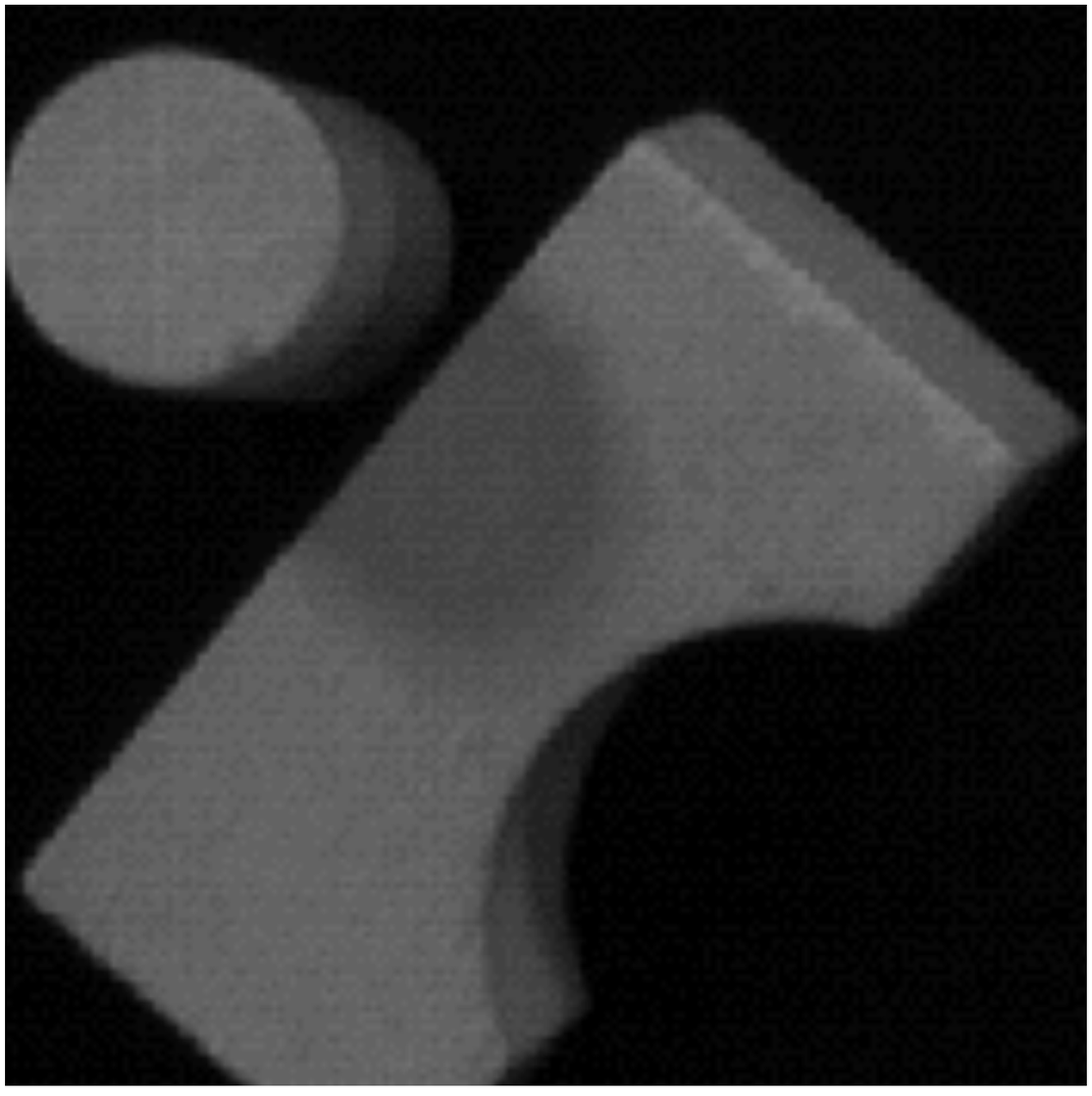


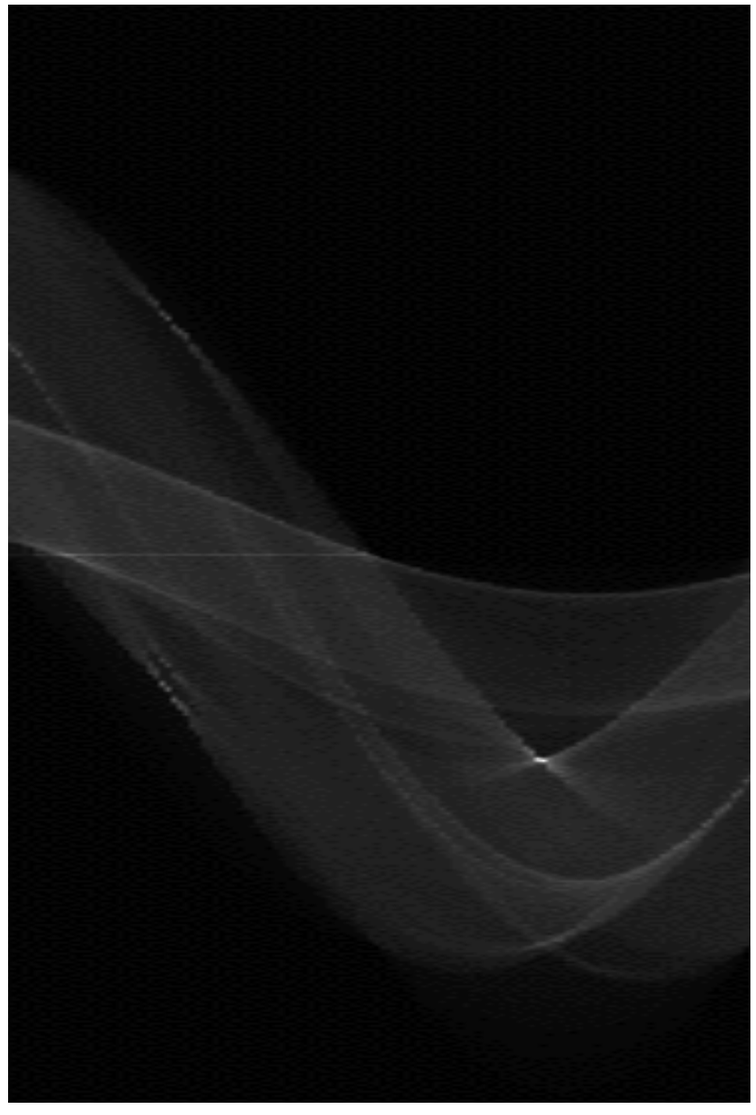
rectangle



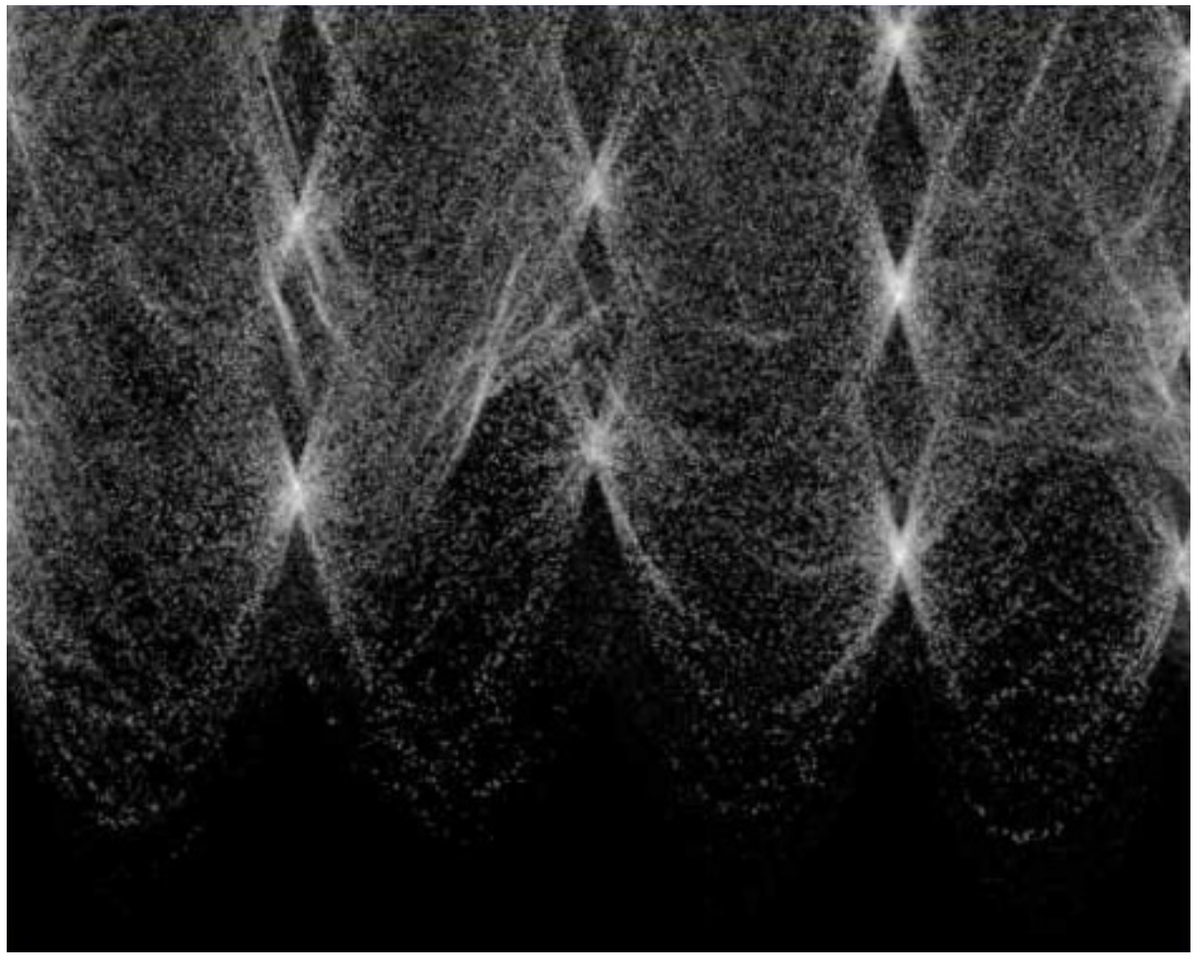
circle

Basic Shapes





More complex image



In practice, measurements are noisy…

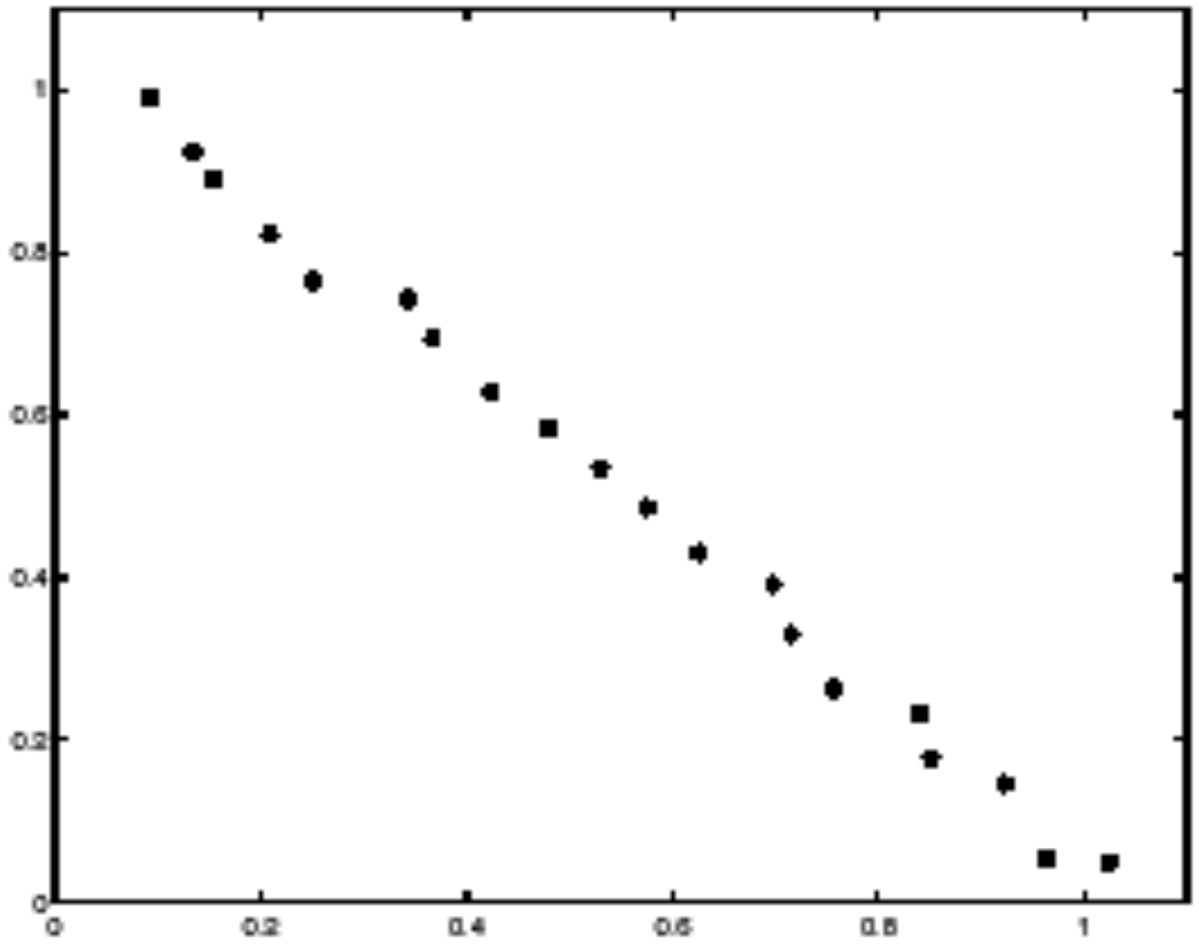
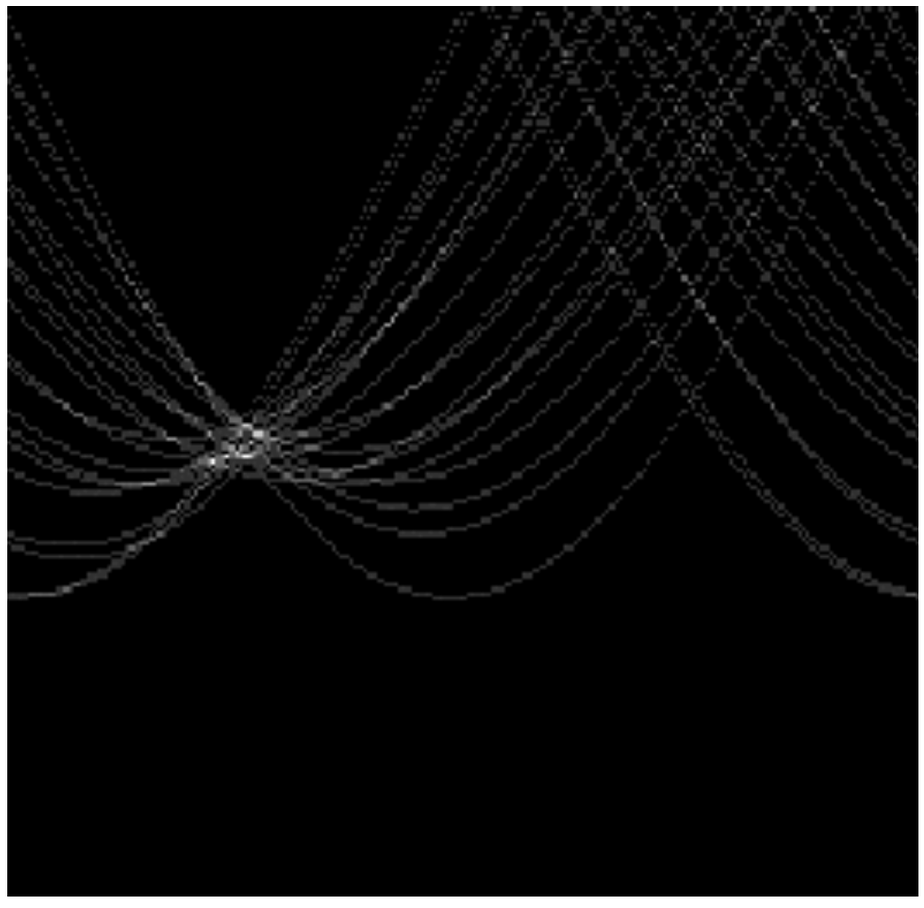


Image space



Votes

Too much noise …

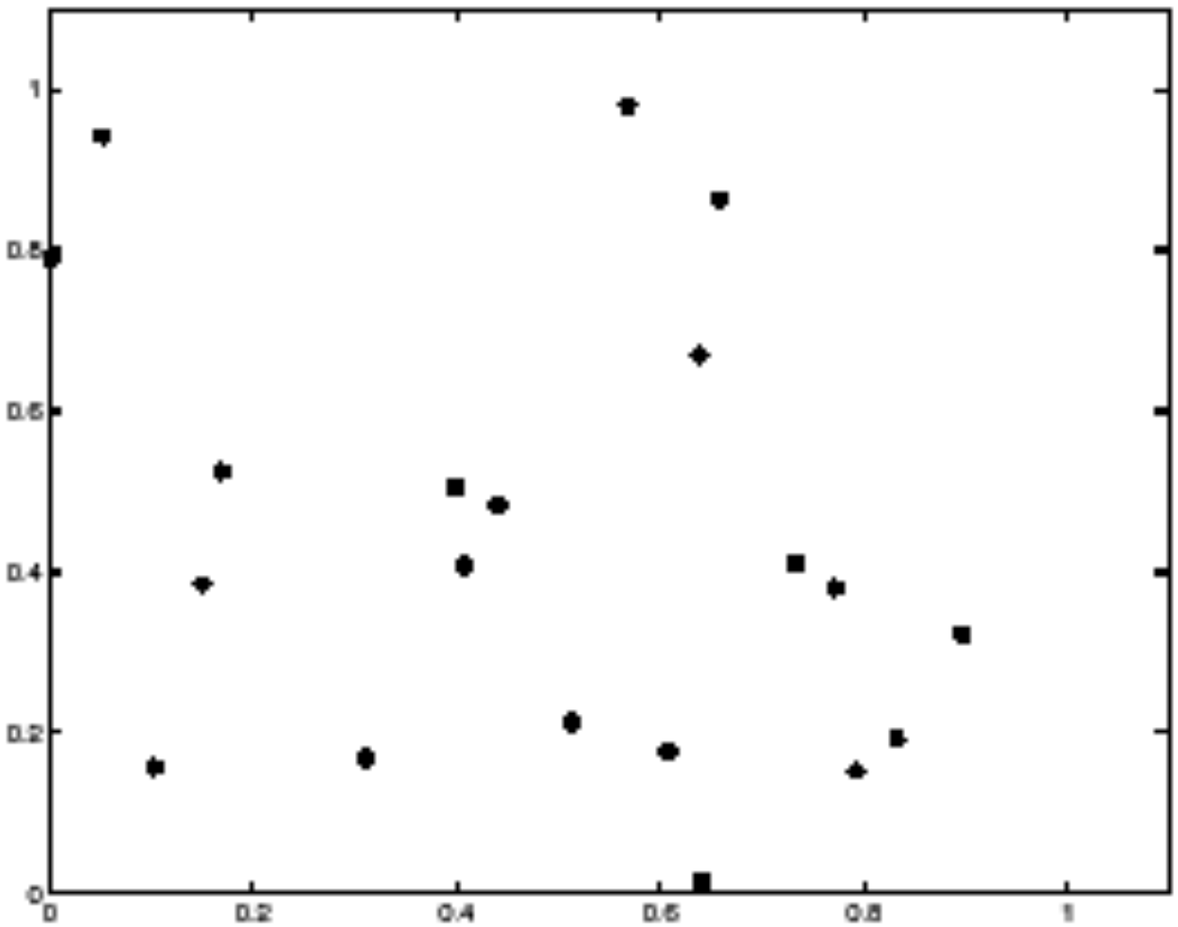
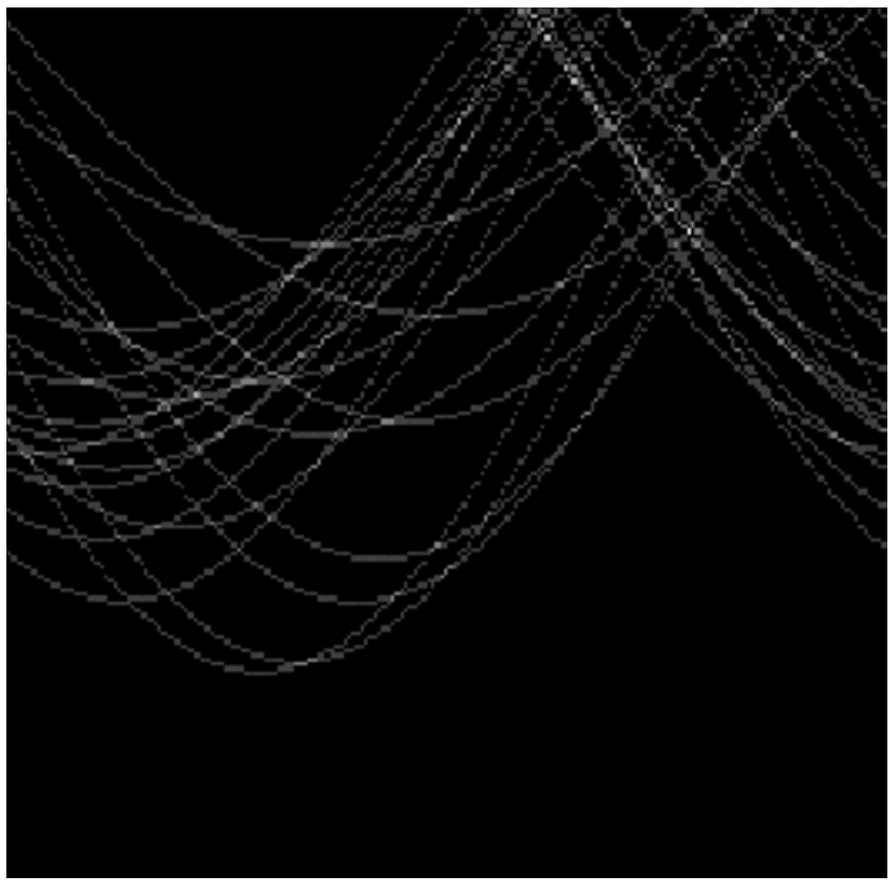
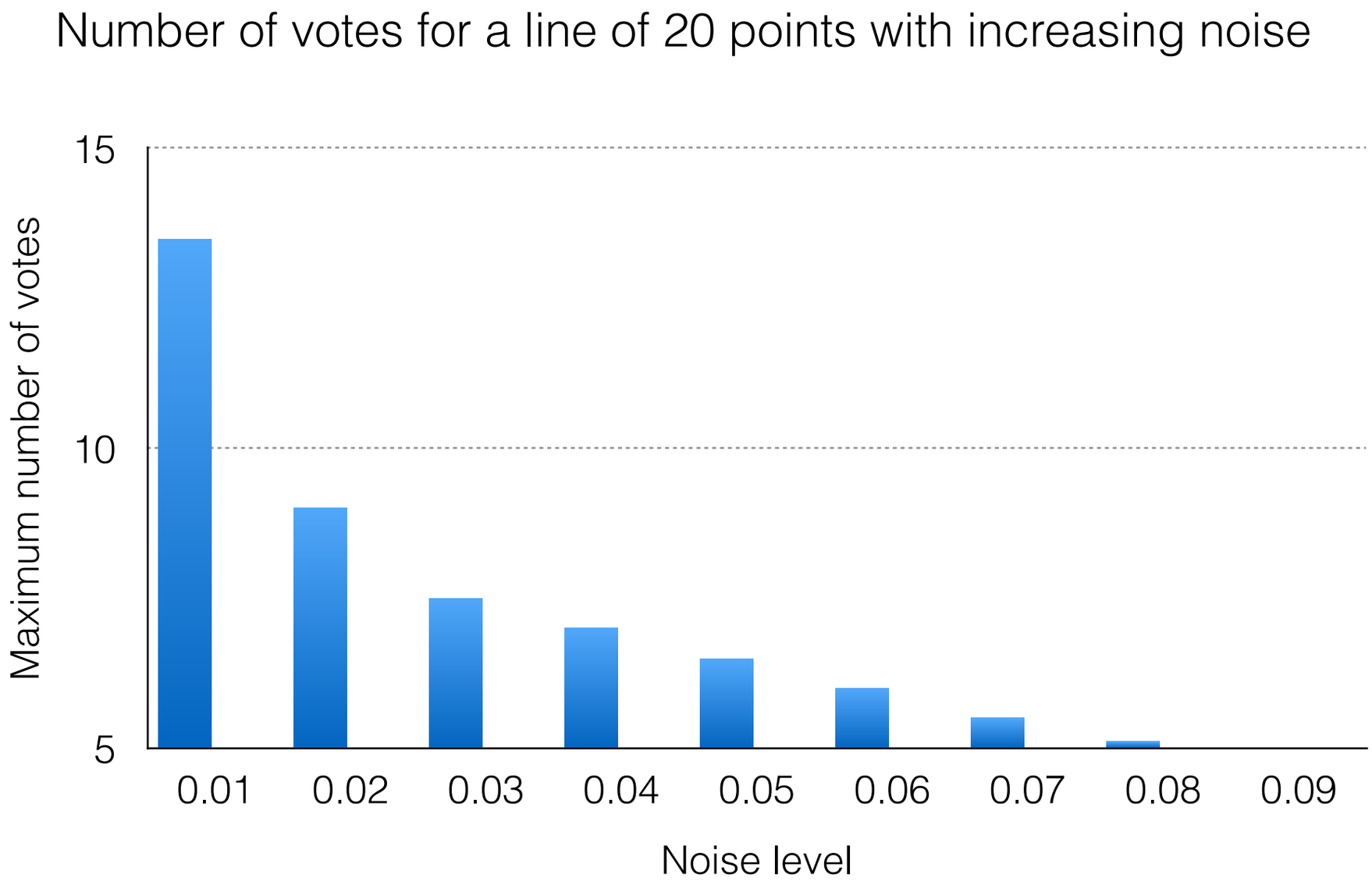


Image space



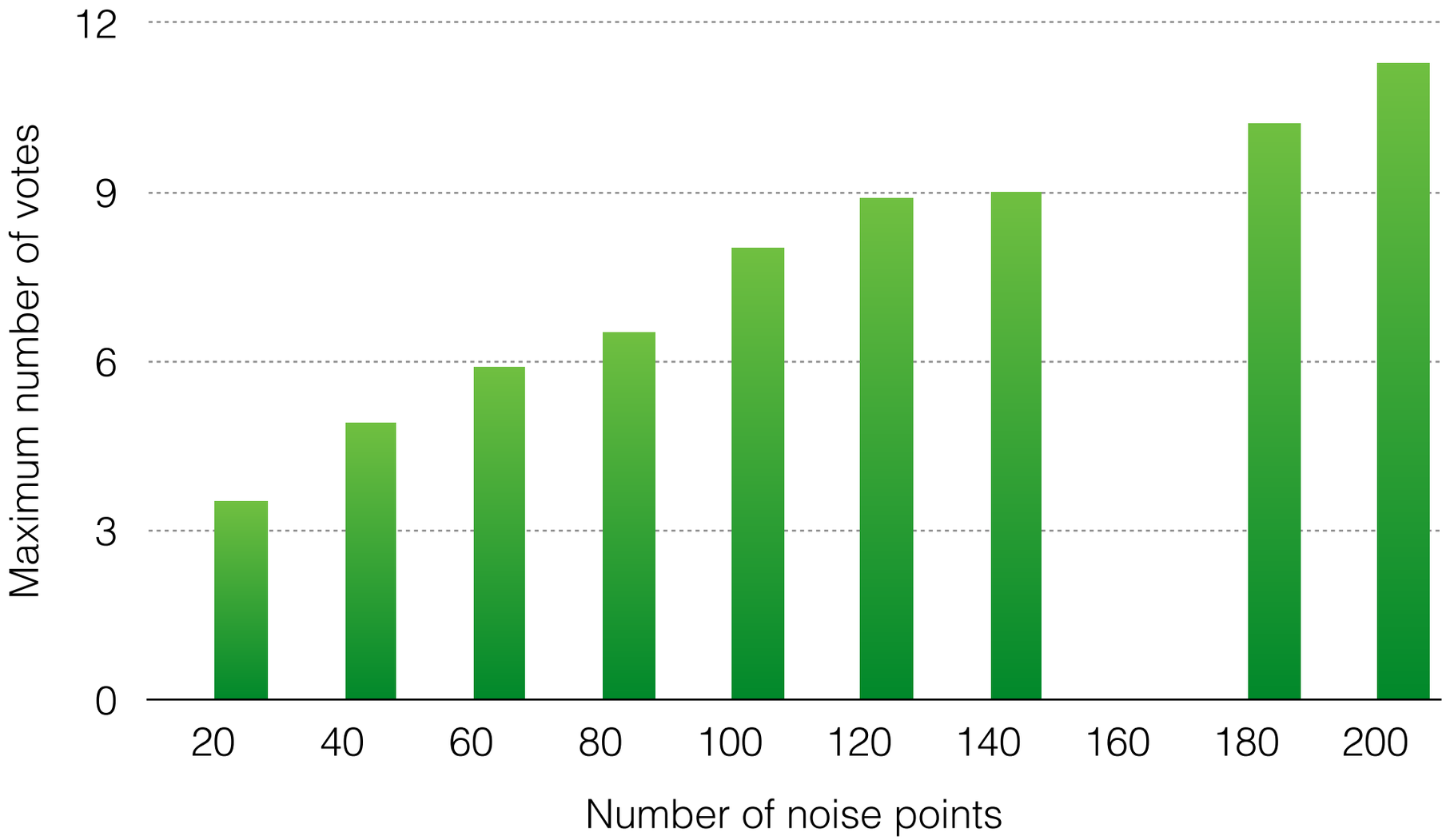
Votes

Effects of noise level



More noise, less votes (in the right bin)

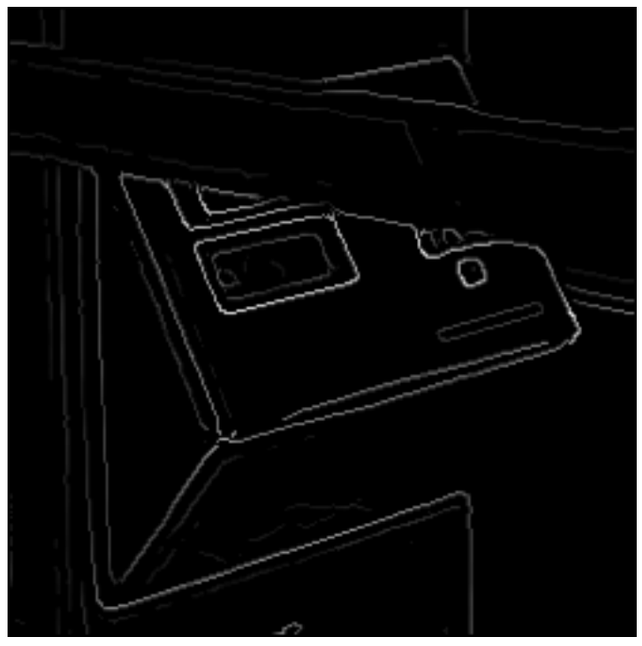
Effect of noise points



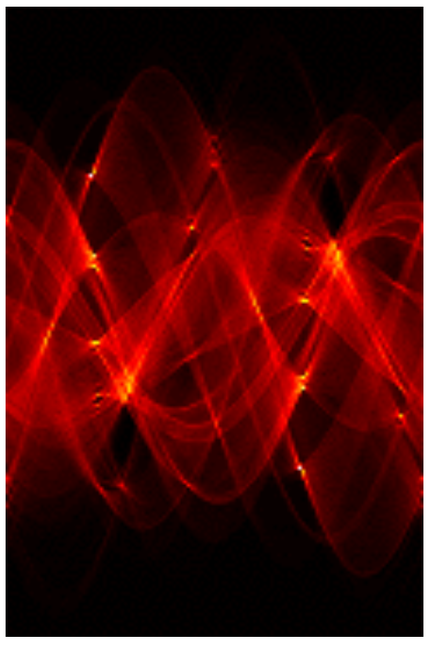
More noise, more votes (in the wrong bin)

Real-world example

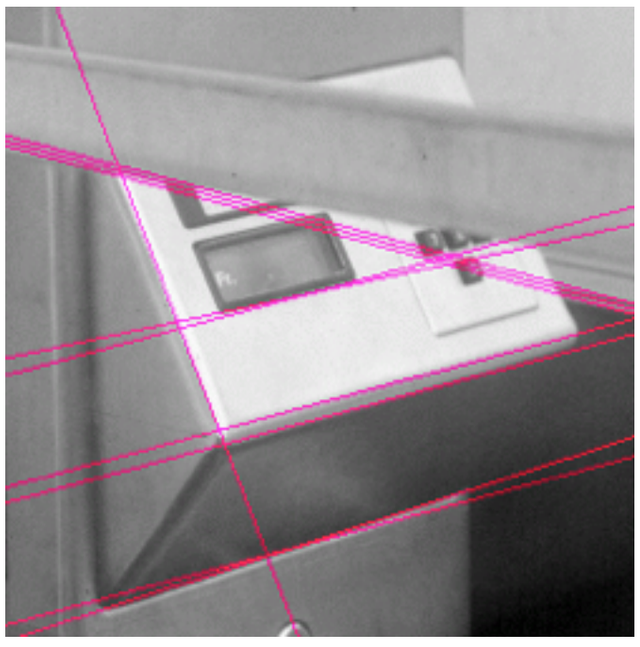
Original



Edges



parameter space



Hough Lines