Slide credits

Matthew P. O'Toole, Carnegie Mellon University

Most of these slides were adapted from:

- Kris Kitani

Some slides were inspired or taken from:

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

Hough transform

Difficulties in Fitting Approach

- Extraneous Data: Which points to fit to
- Incomplete Data: Only part of the model is visible.
- Noise

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

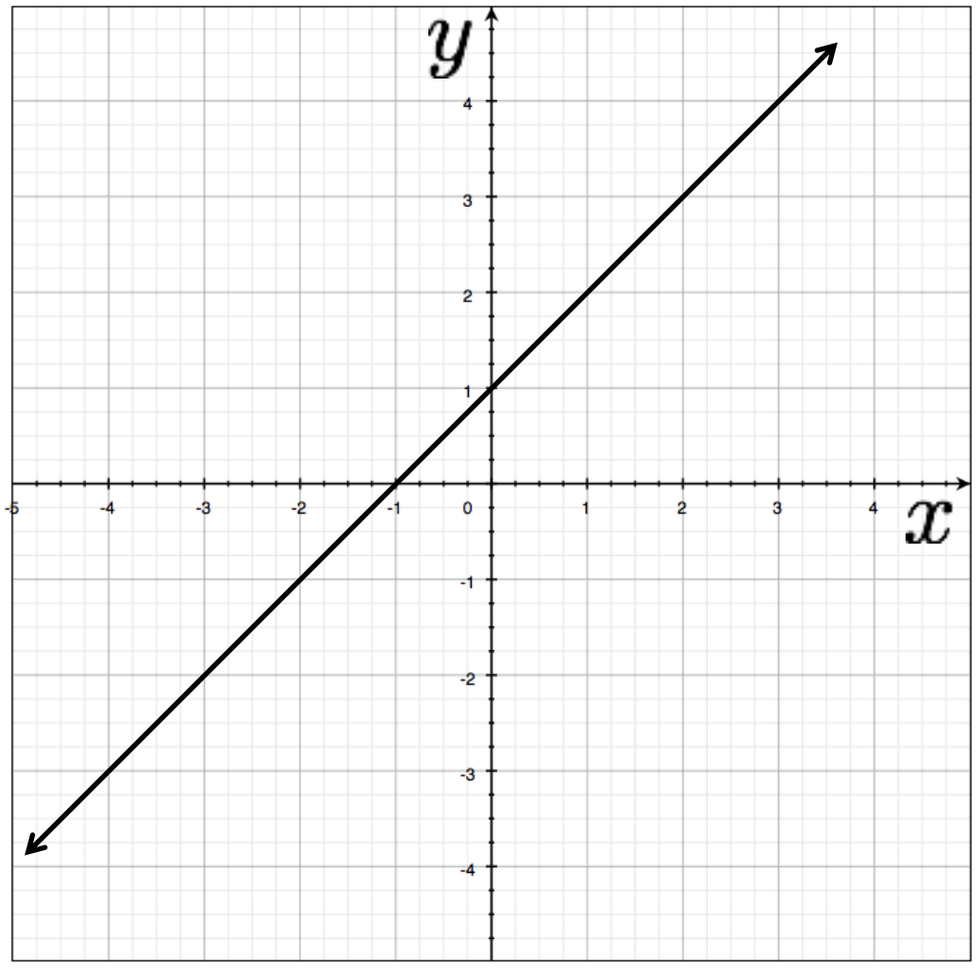


Image space

Image and parameter space

variables

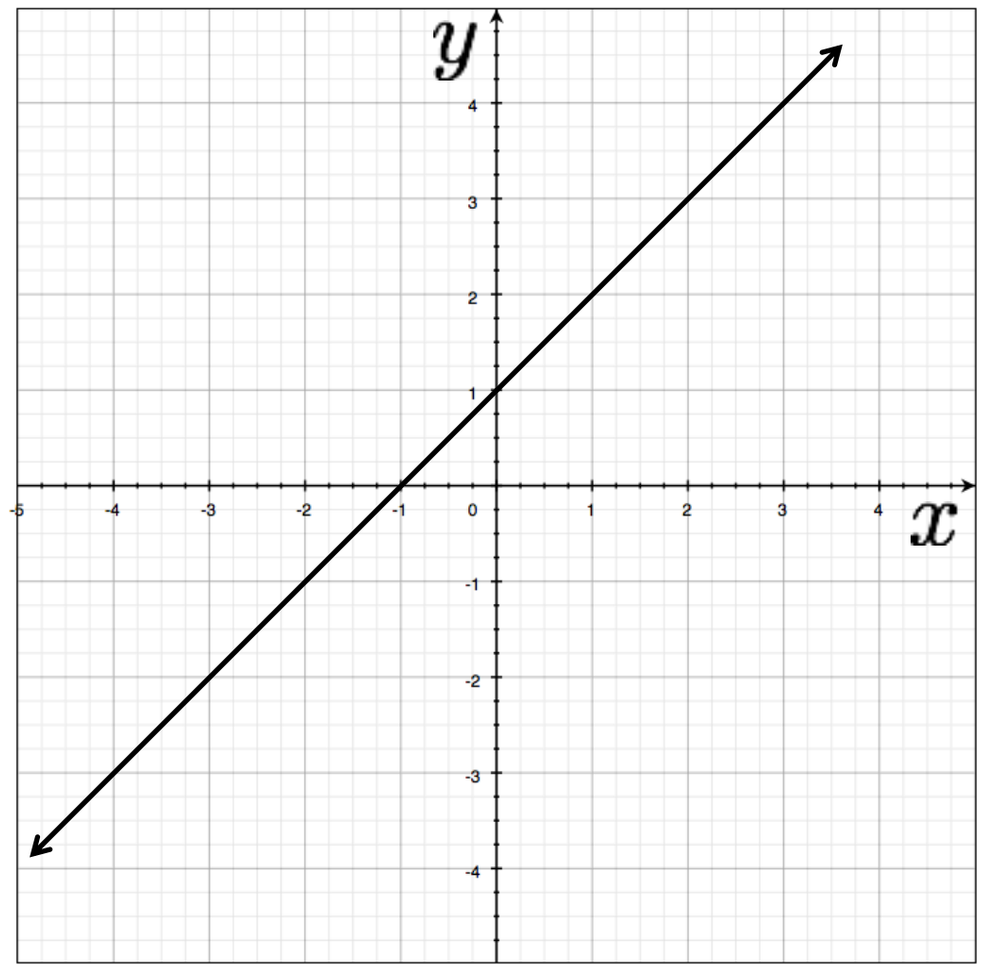
$$y = mx + b$$

parameters

variables

$$y - mx = b$$

parameters



a line becomes a point

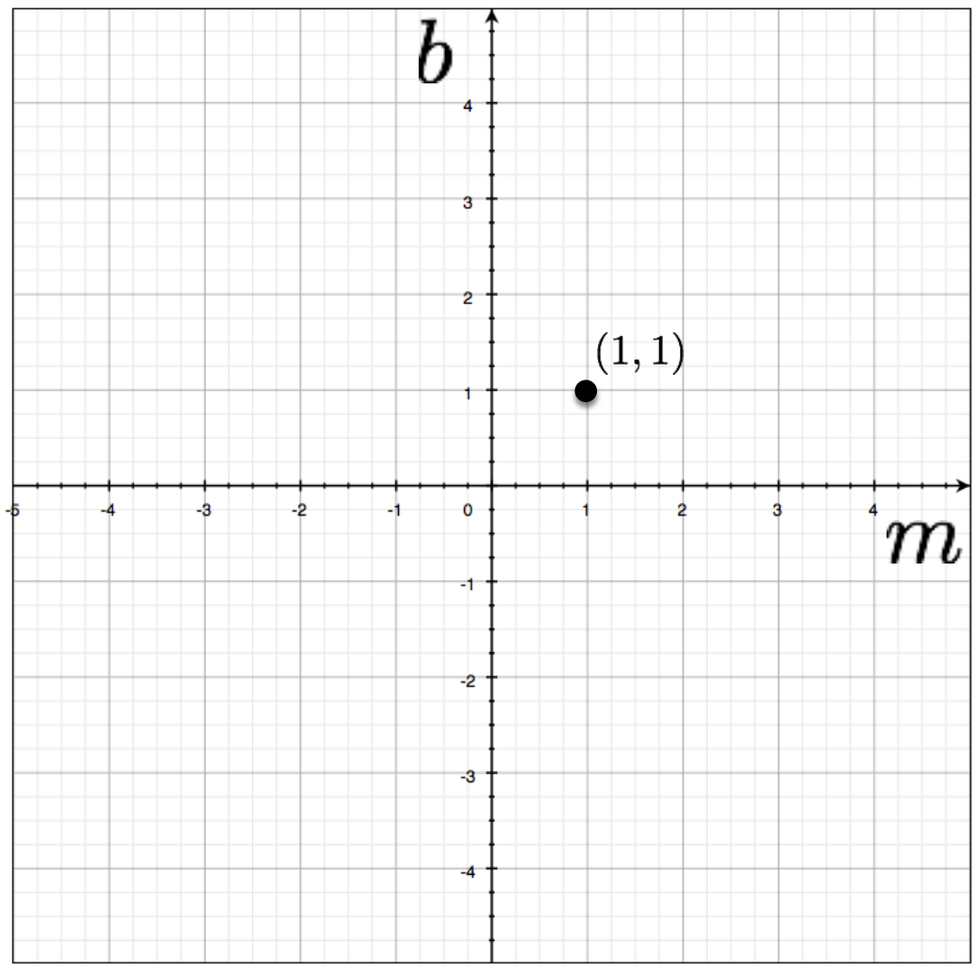


Image space

Parameter space

Image and parameter space

variables

$$y = mx + b$$

parameters

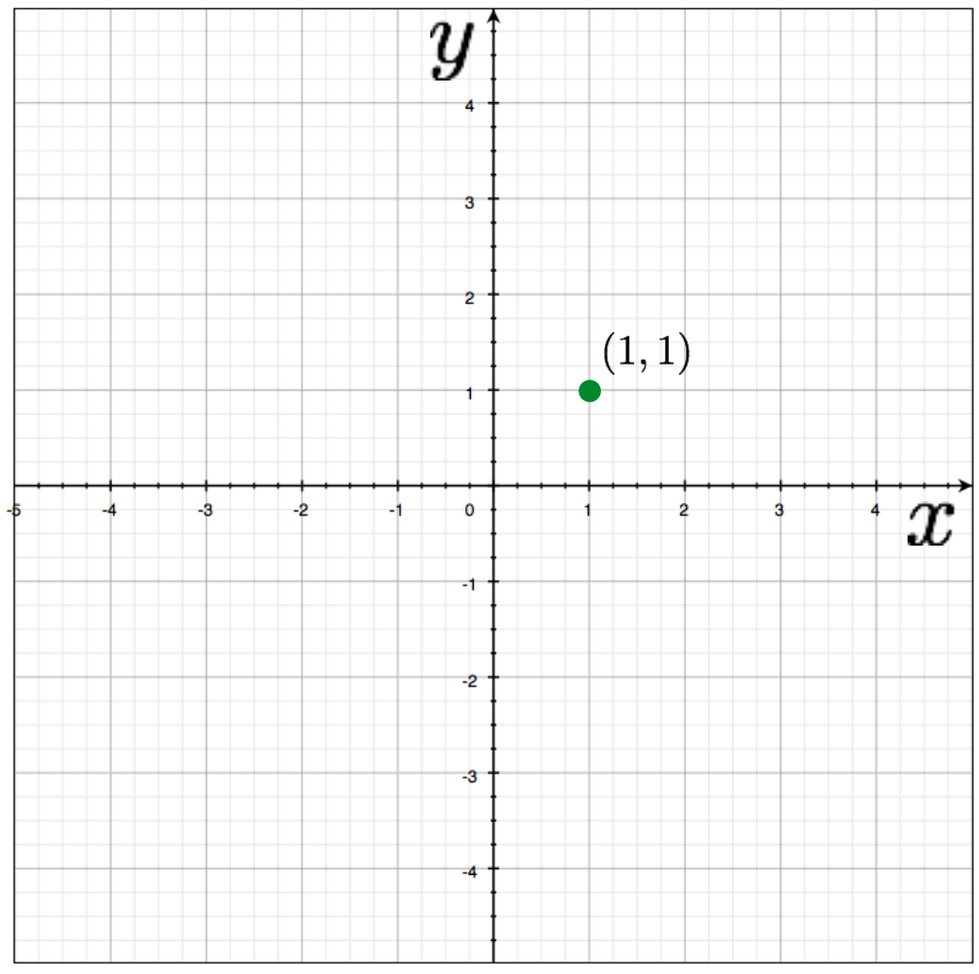


Image space

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

Image and parameter space

variables

$$y = mx + b$$

parameters

variables

$$y - mx = b$$

parameters

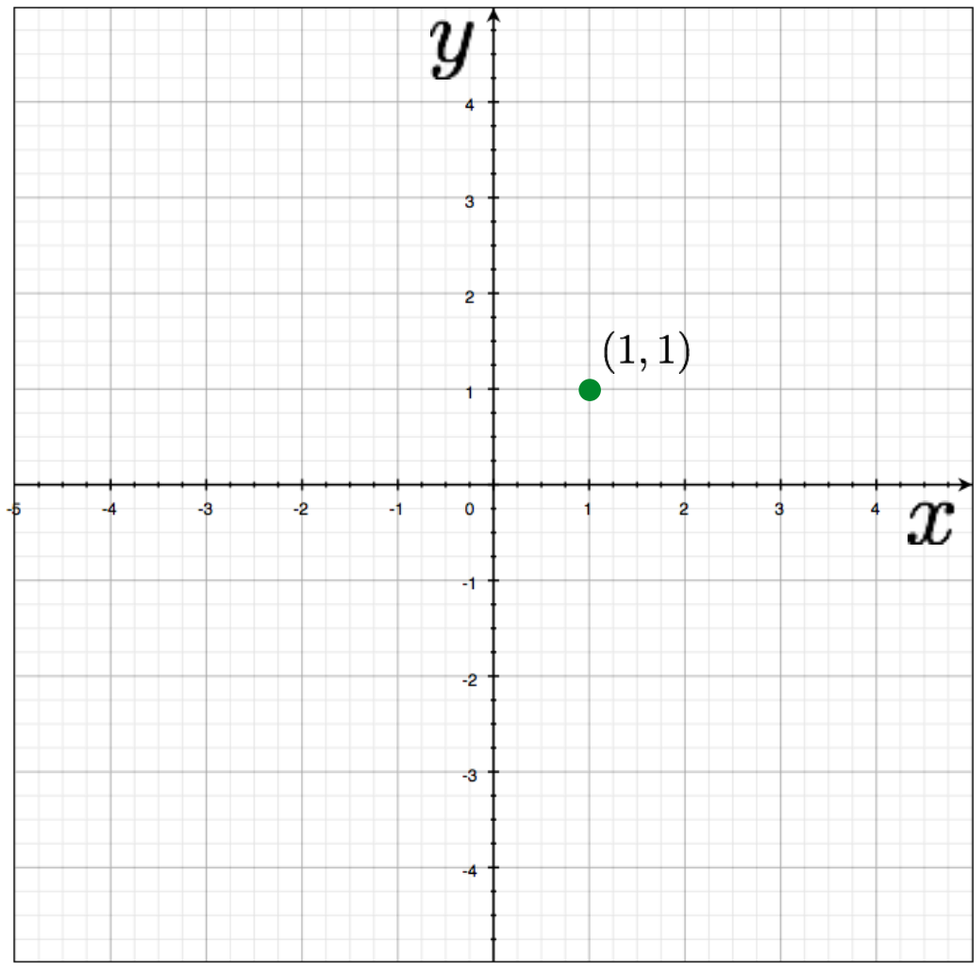
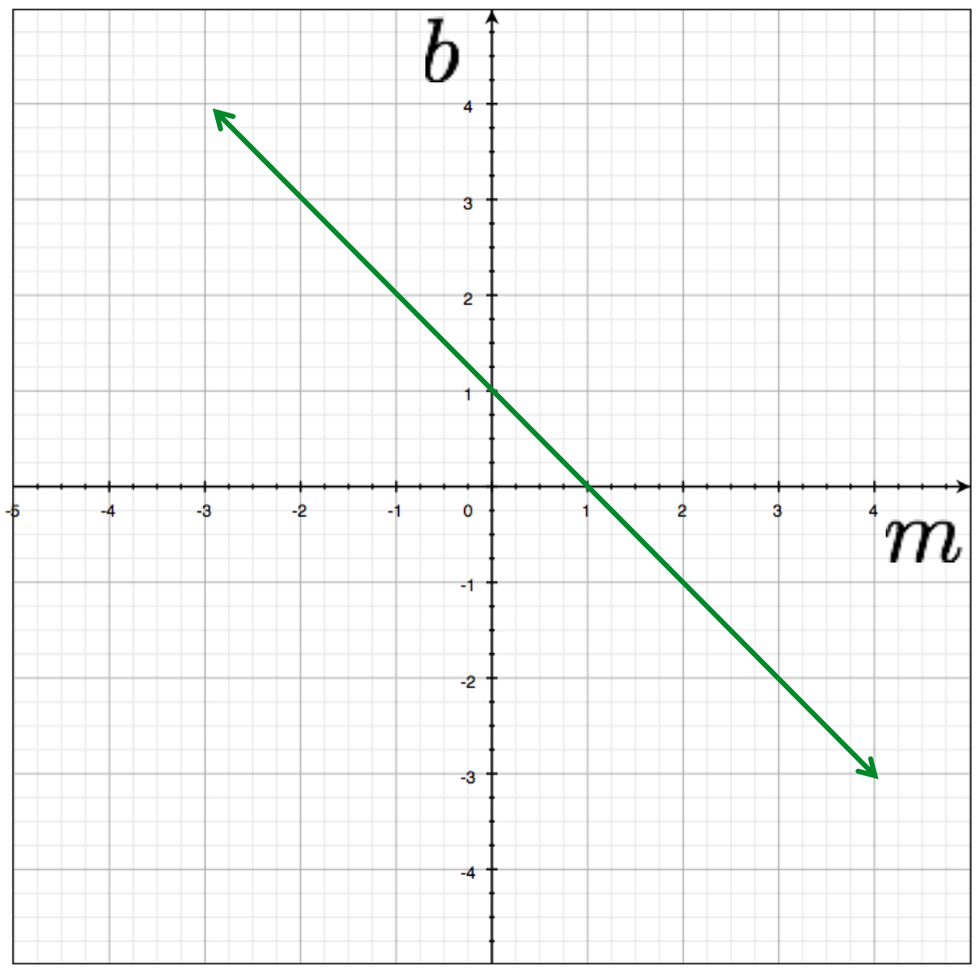


Image space

a point becomes a line



Parameter space

Image and parameter space

variables

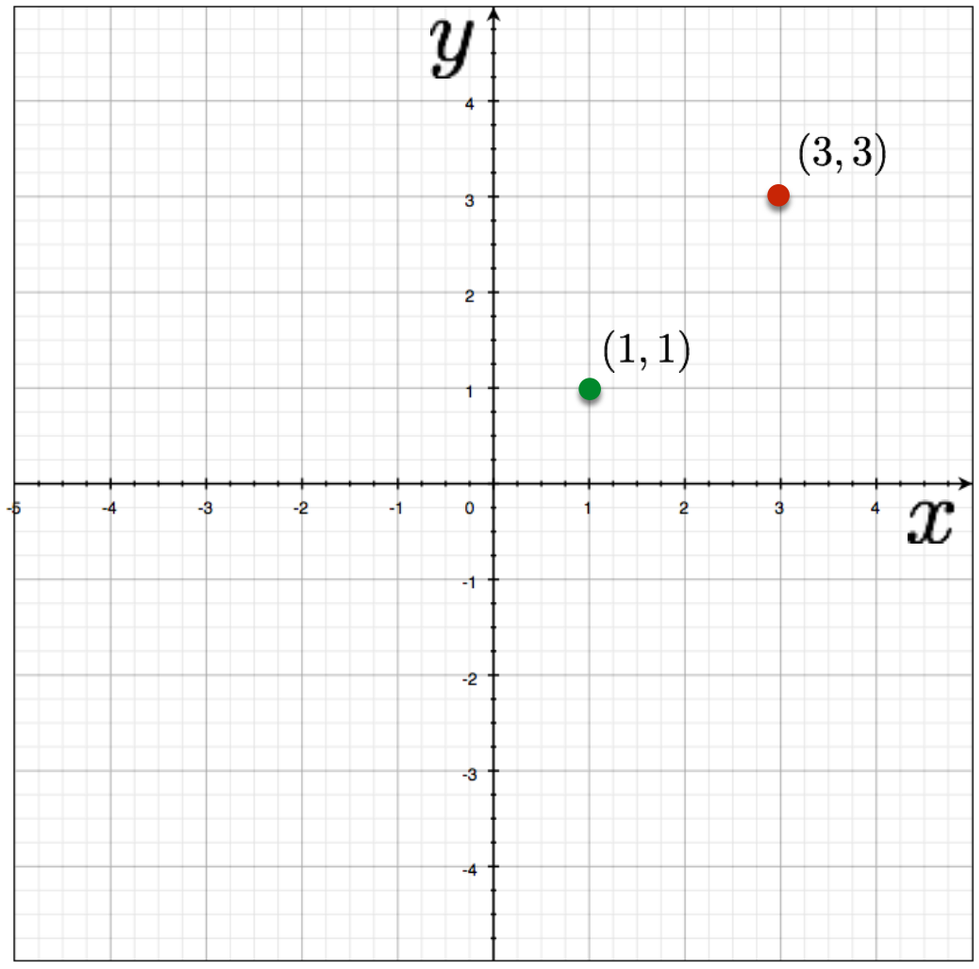
$$y = mx + b$$

parameters

variables

$$y - mx = b$$

parameters



two points become ?

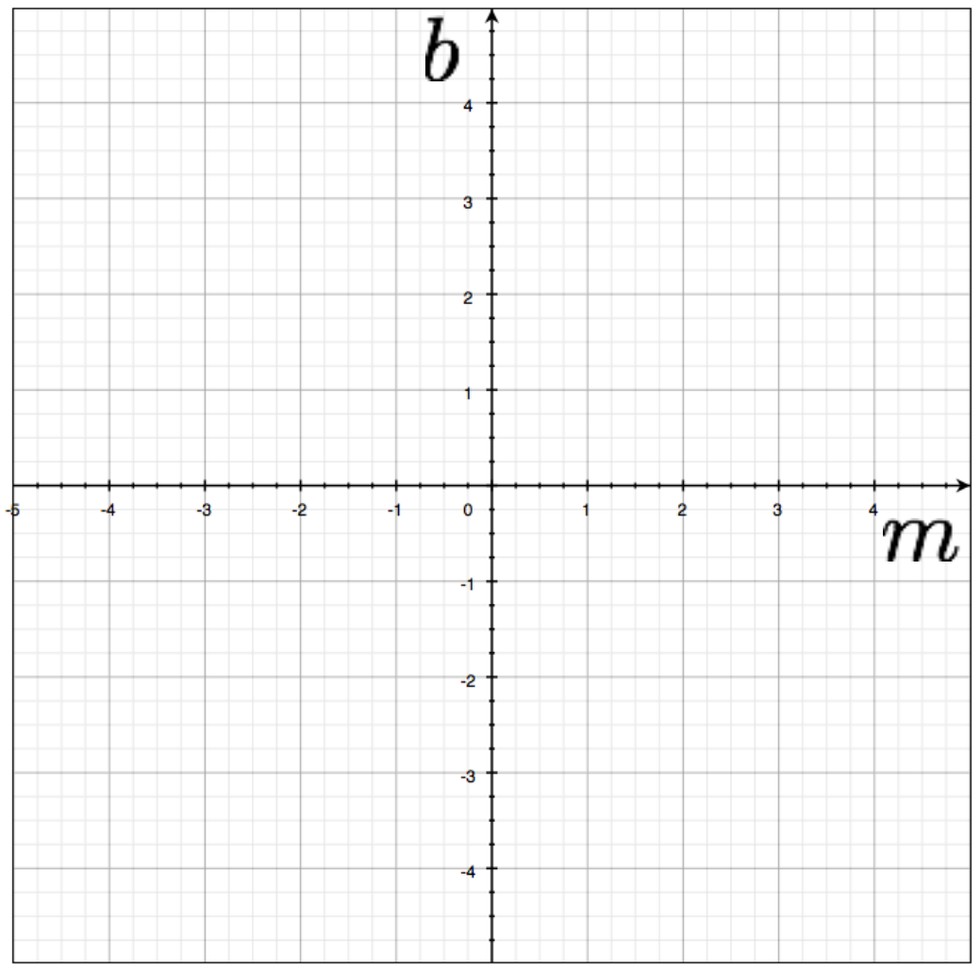


Image space

Parameter space

Image and parameter space

variables

$$y = mx + b$$

parameters

variables

$$y - mx = b$$

parameters

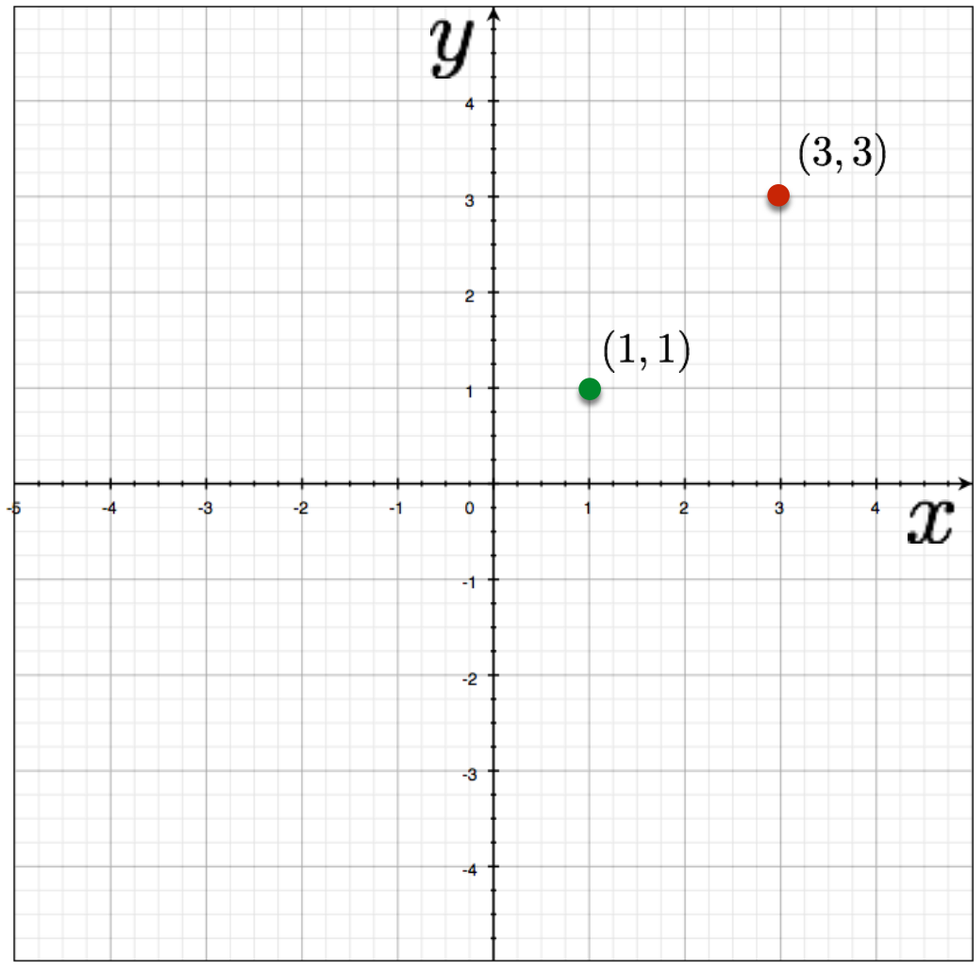
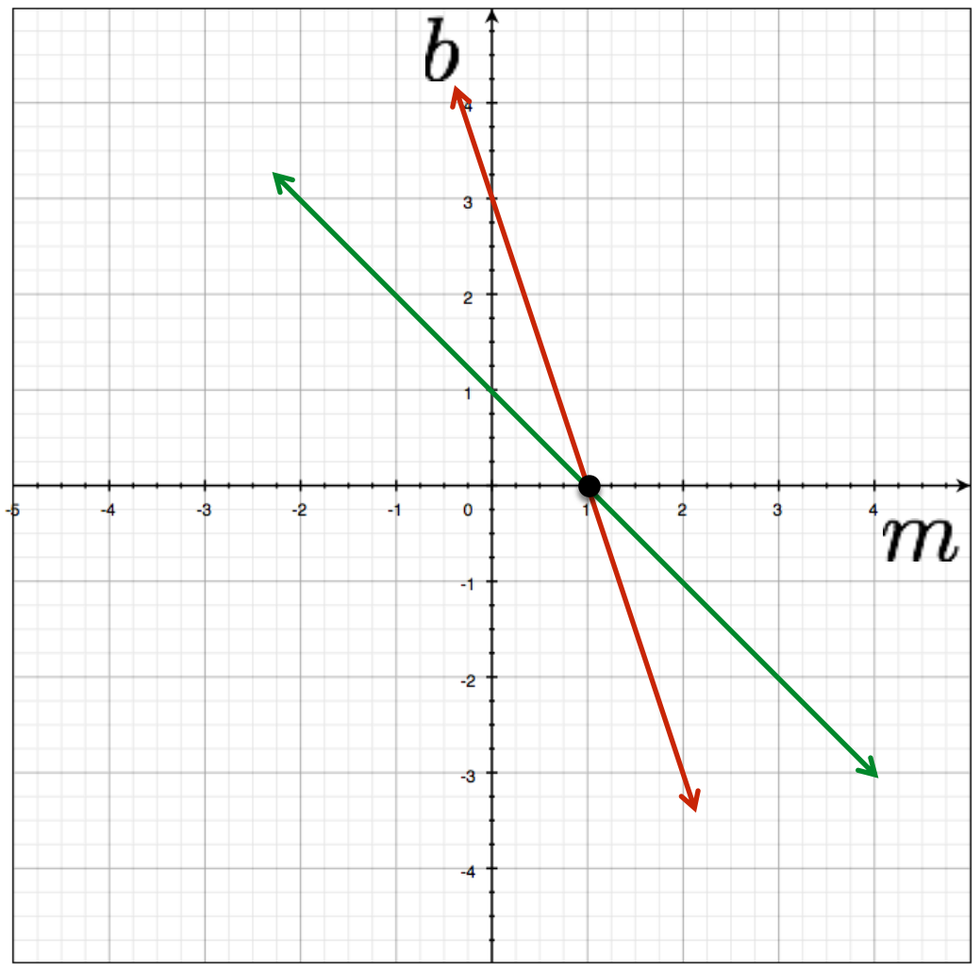


Image space

two points become ?



Parameter space

Image and parameter space

variables

$$y = mx + b$$

parameters

variables

$$y - mx = b$$

parameters

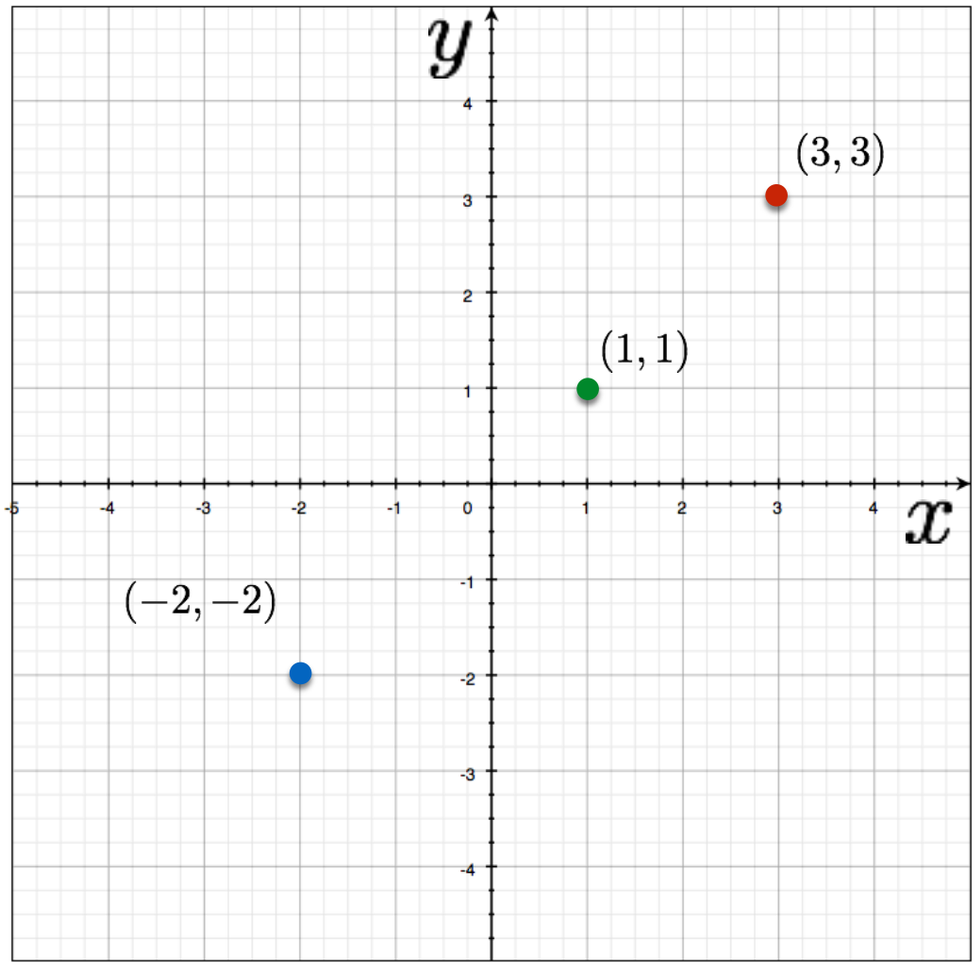
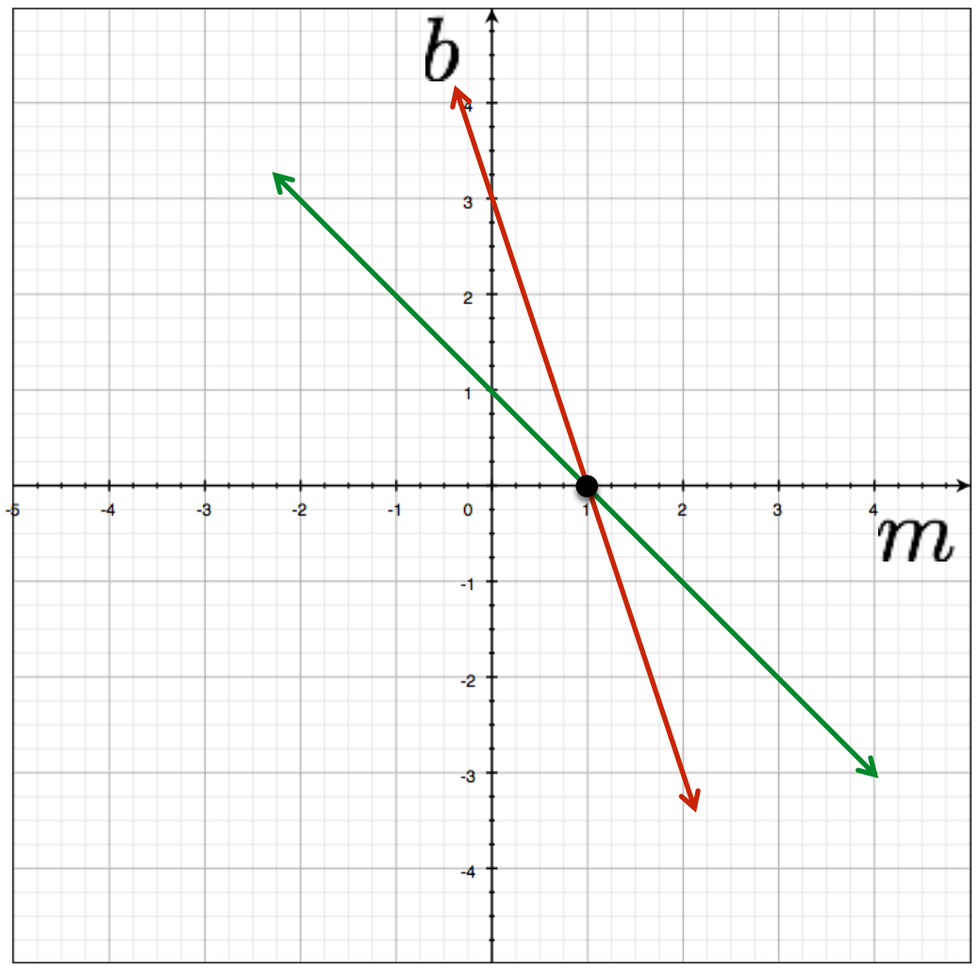


Image space

three points become ?



Parameter space

Image and parameter space

variables

$$y = mx + b$$

parameters

variables

$$y - mx = b$$

parameters

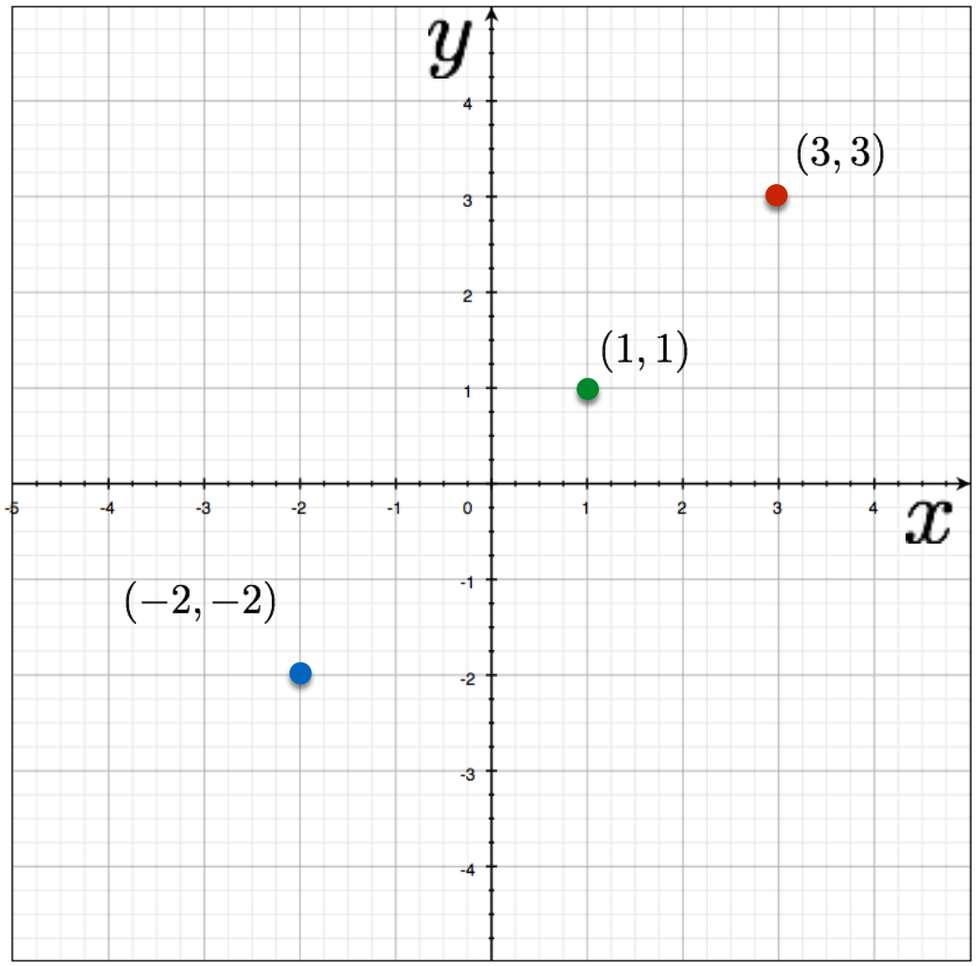
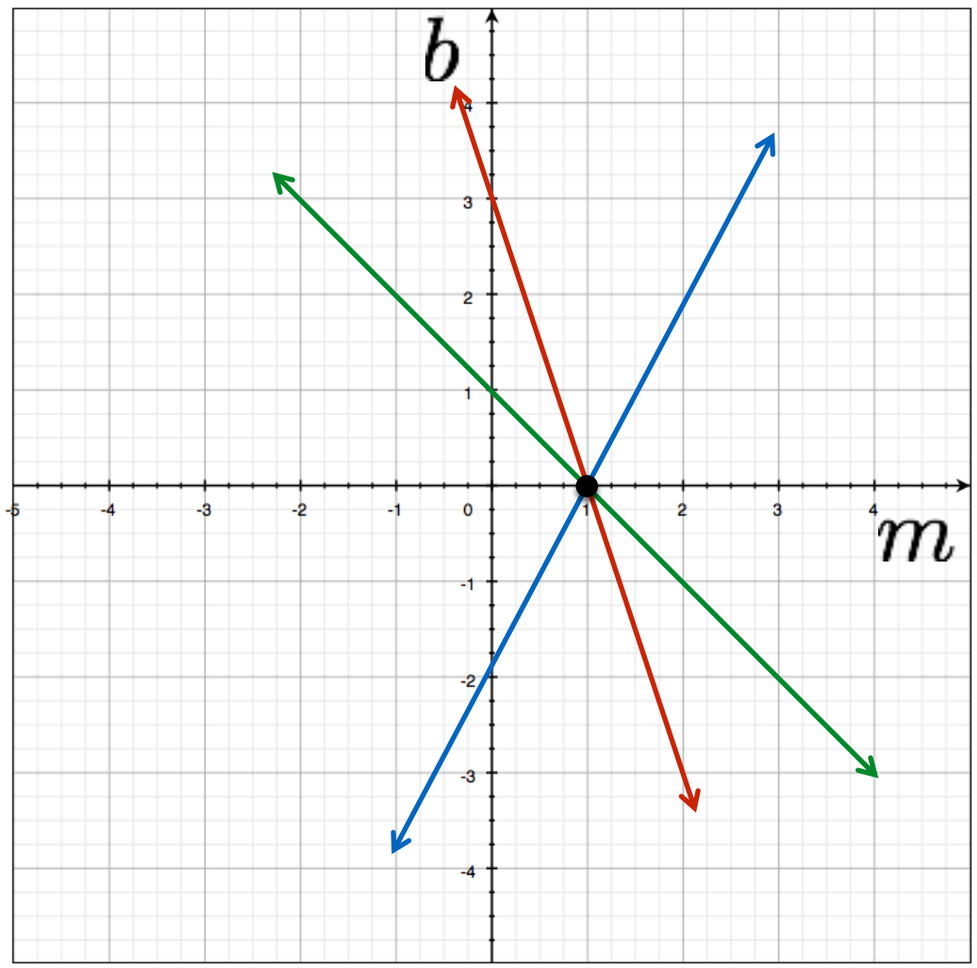


Image space

three points become ?



Parameter space

Image and parameter space

variables

$$y = mx + b$$

parameters

variables

$$y - mx = b$$

parameters

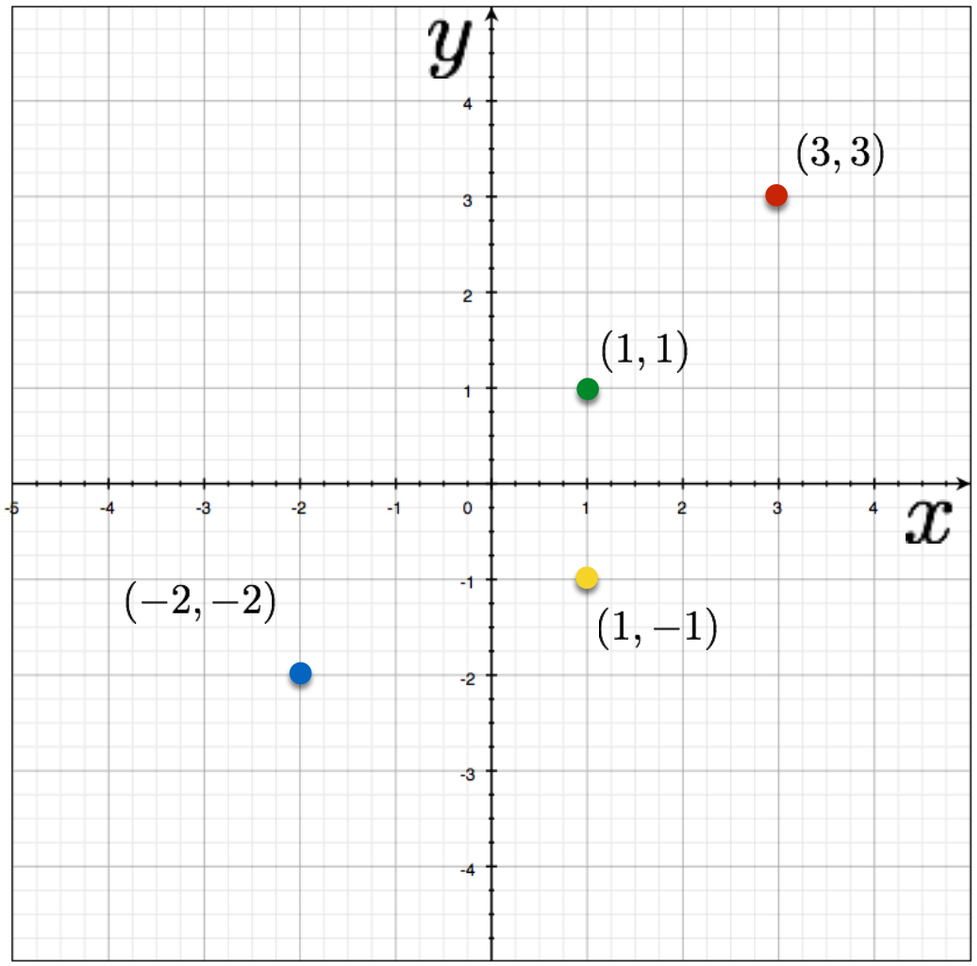
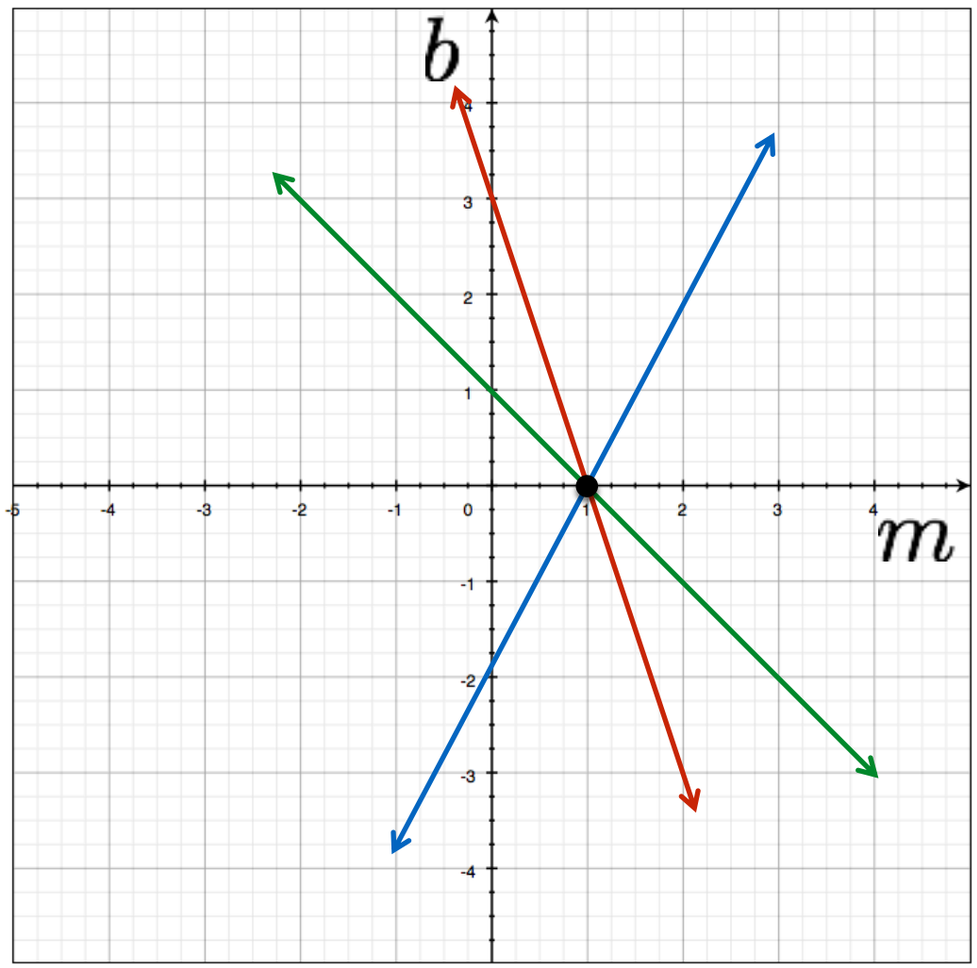


Image space

four points become ?



Parameter space

Image and parameter space

variables

$$y = mx + b$$

parameters

variables

$$y - mx = b$$

parameters

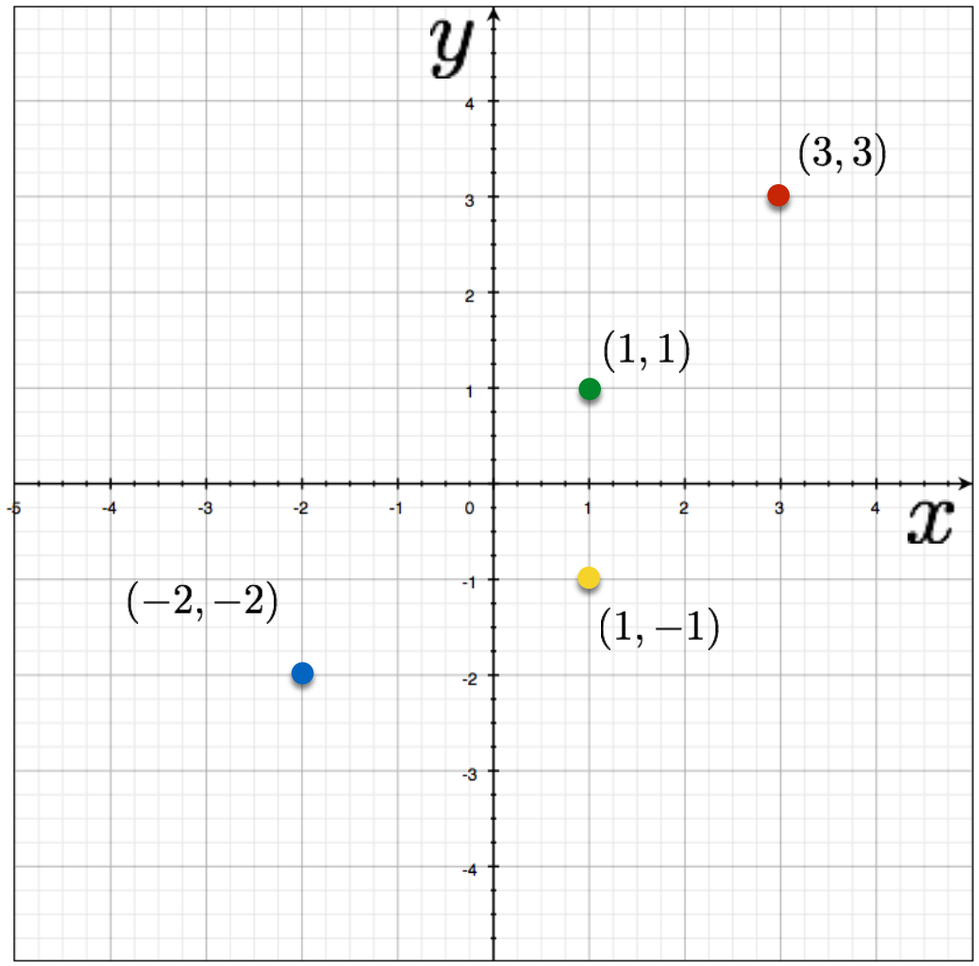
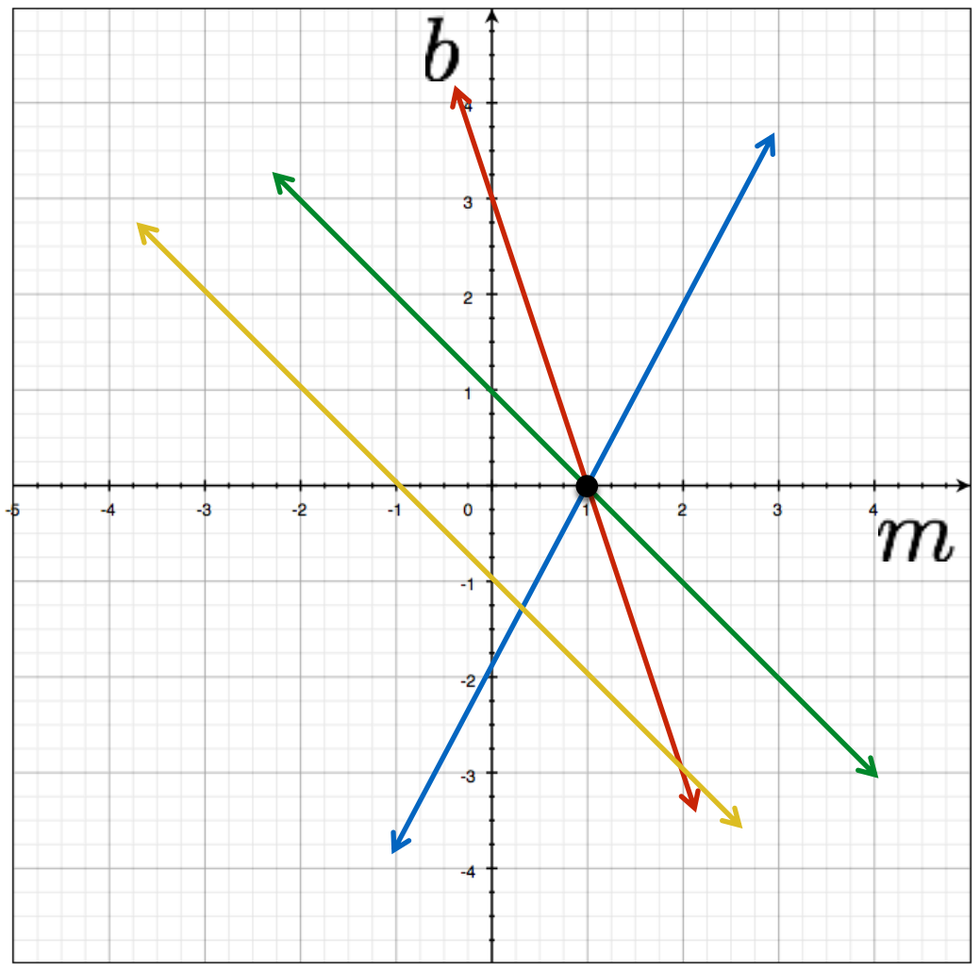


Image space

four points become ?



Parameter space

How would you find the best fitting line?

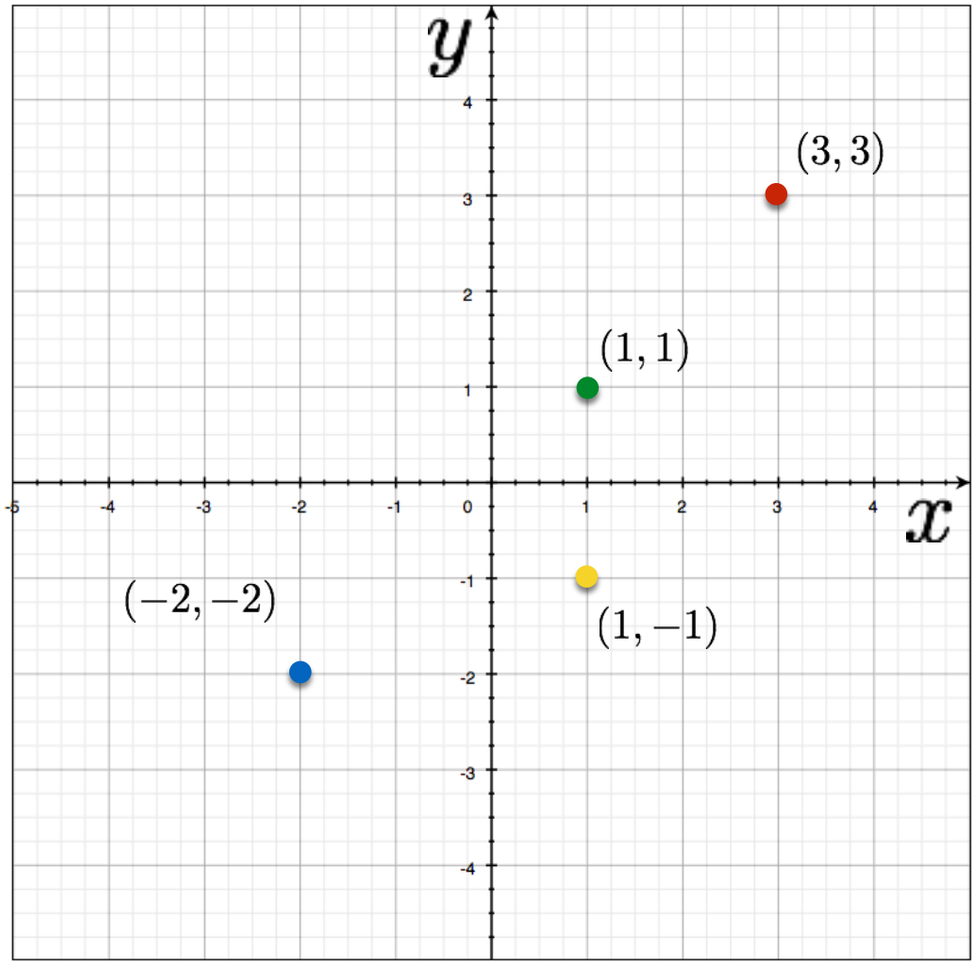
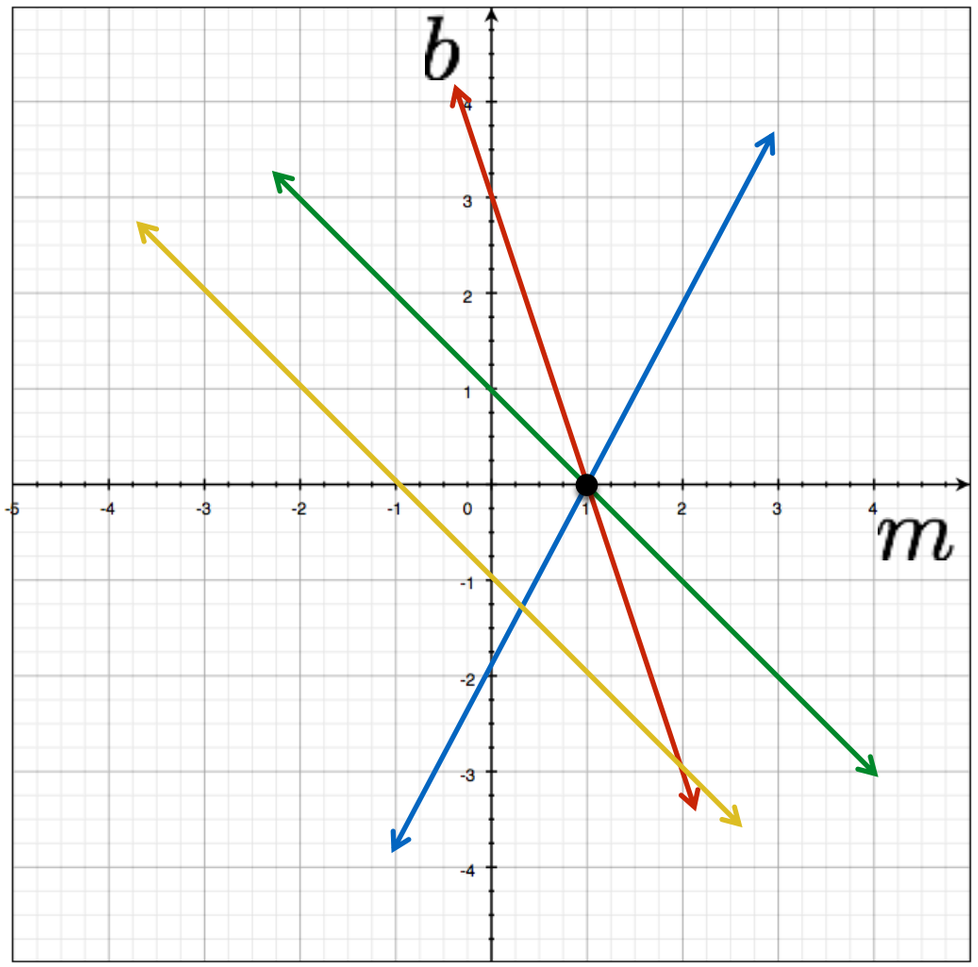


Image space



Parameter space

Line Detection by Hough Transform

Algorithm:

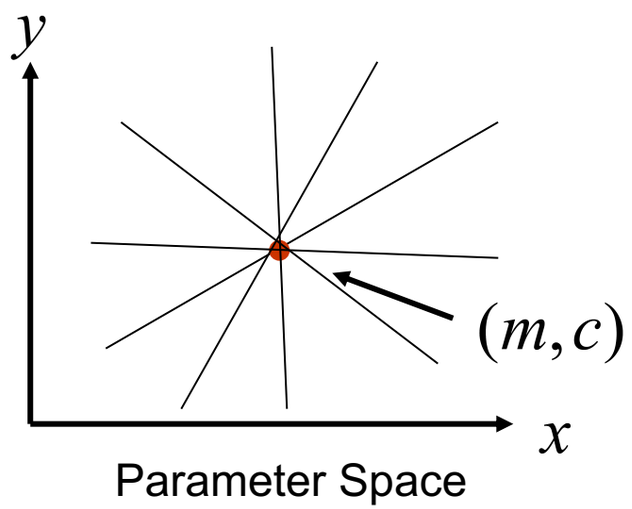
1. Quantize Parameter Space (m,c)

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 $A(m,c)$



Parameter Space

$A(m,c)$

	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)*?*

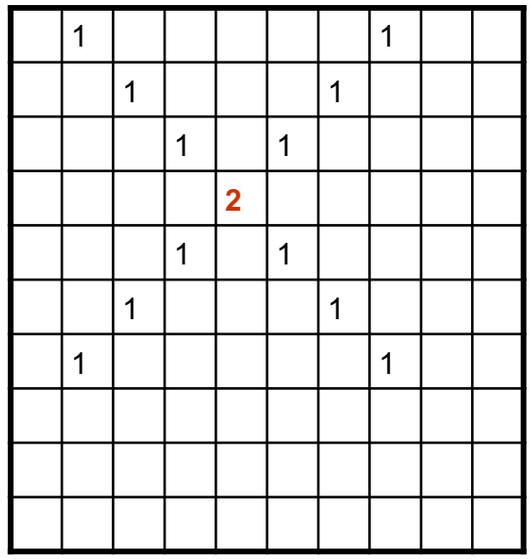
$A(m, c)$

	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)*?*

$A(m,c)$



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 (ρ,θ)

Hough Space Sinusoid

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

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

(Finite Accumulator Array Size)

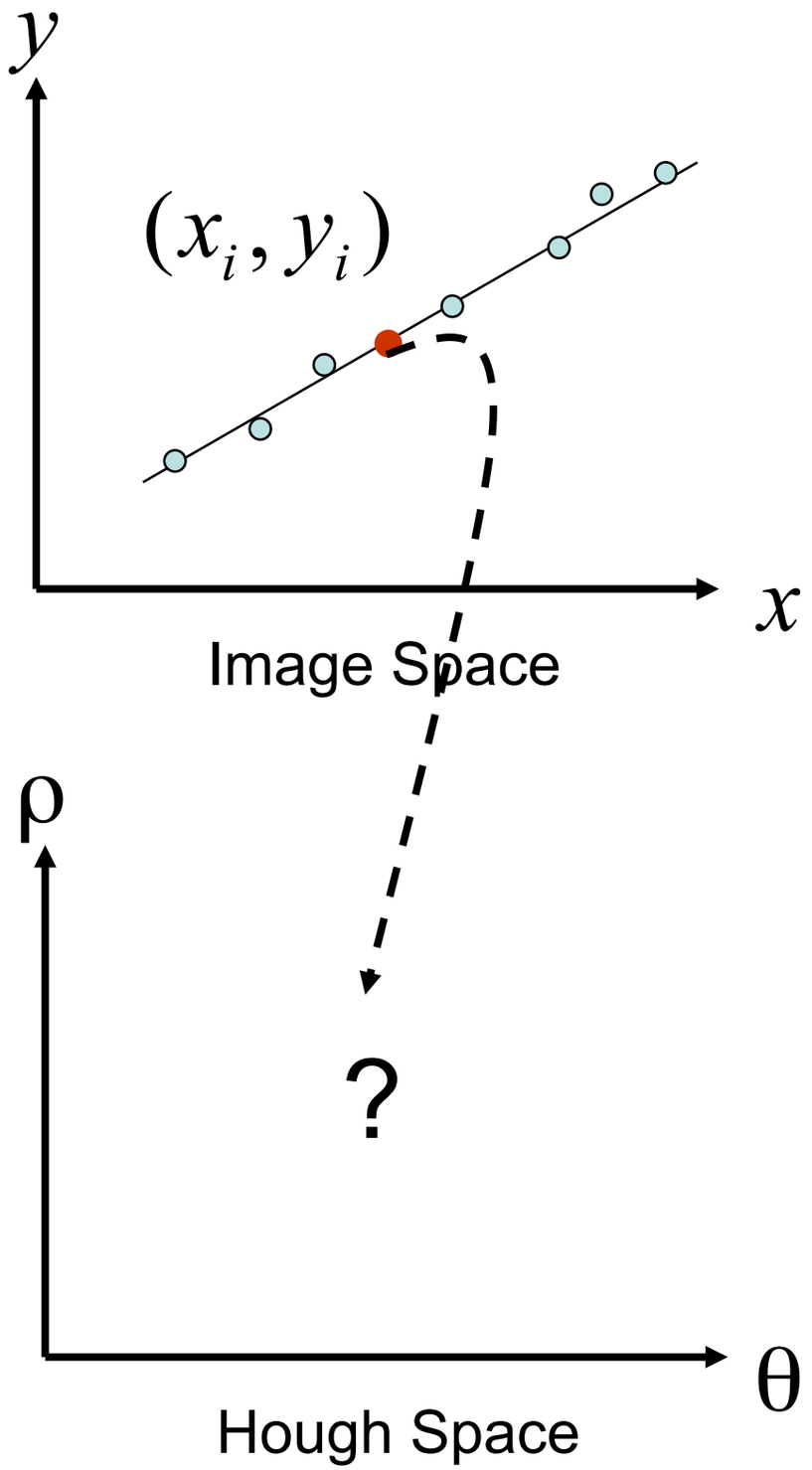


Image and parameter space

variables

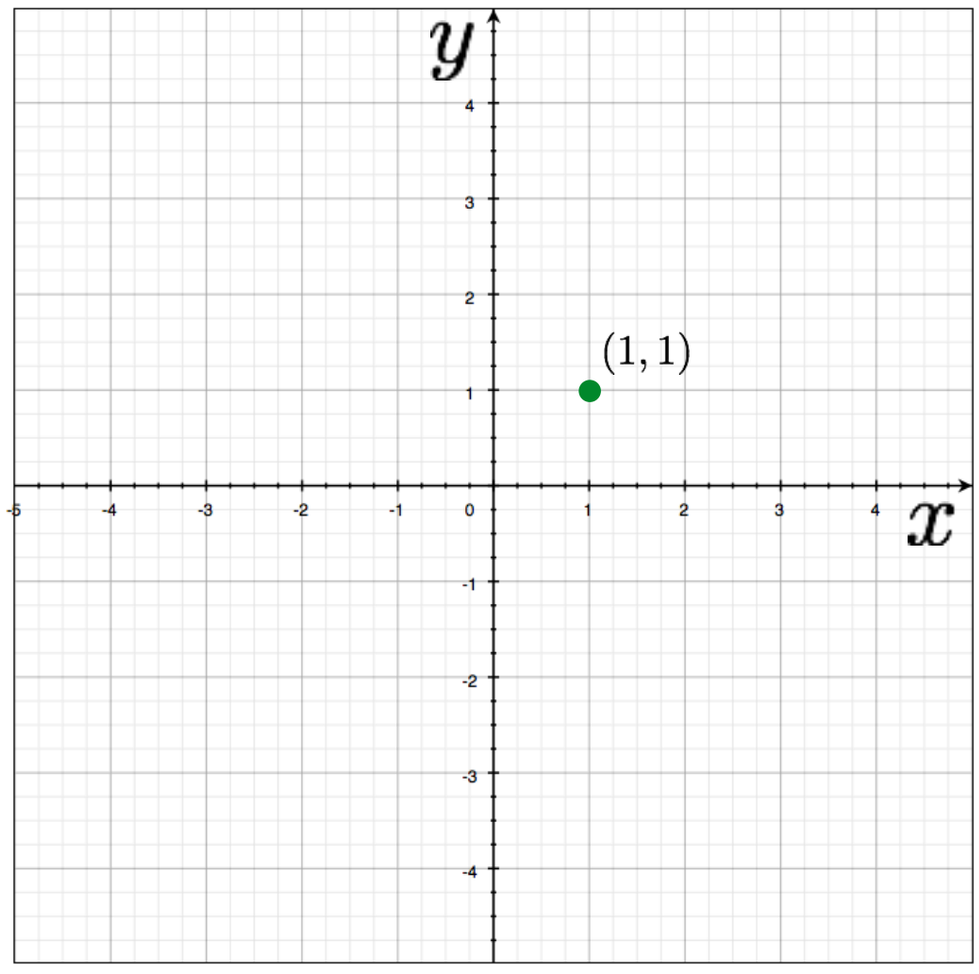
$$y = mx + b$$

parameters

parameters

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

variables



a point becomes?

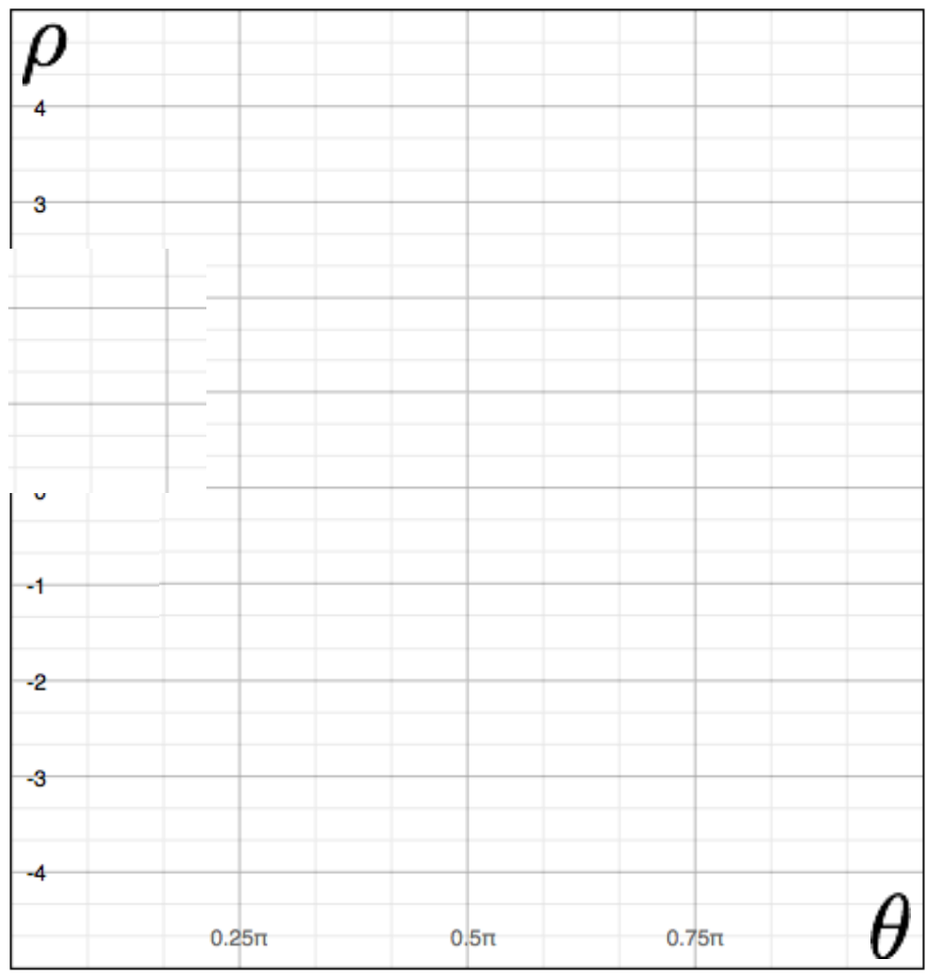


Image space

Parameter space

Image and parameter space

variables

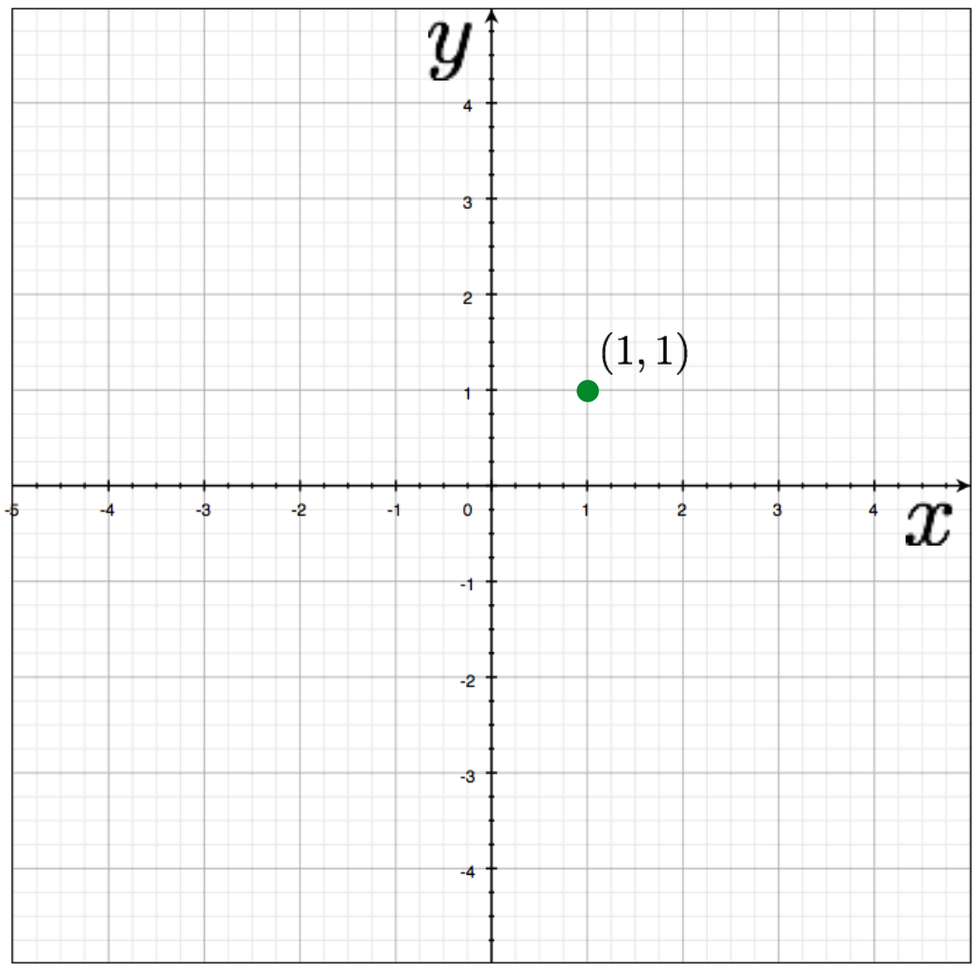
$$y = mx + b$$

parameters

parameters

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

variables



a point becomes a wave

Image space

Parameter space

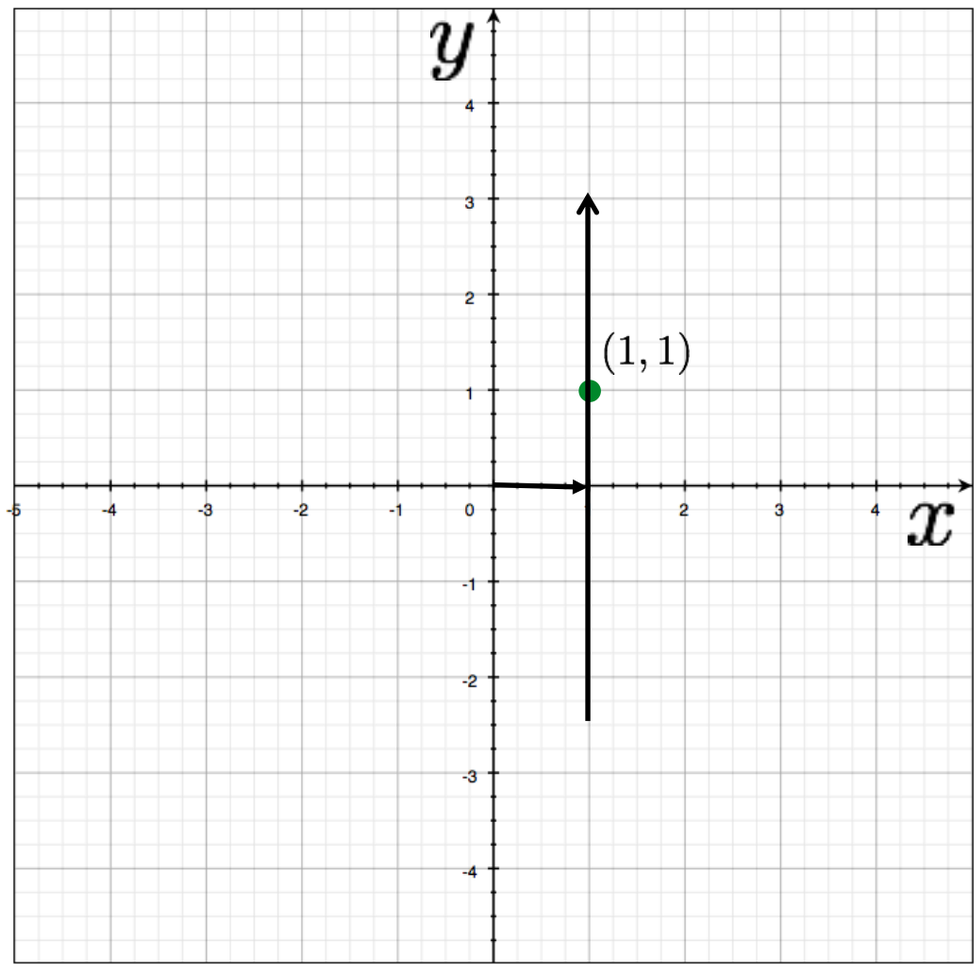
Image and parameter space

variables

$$y = mx + b$$

parameters

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



a line becomes?

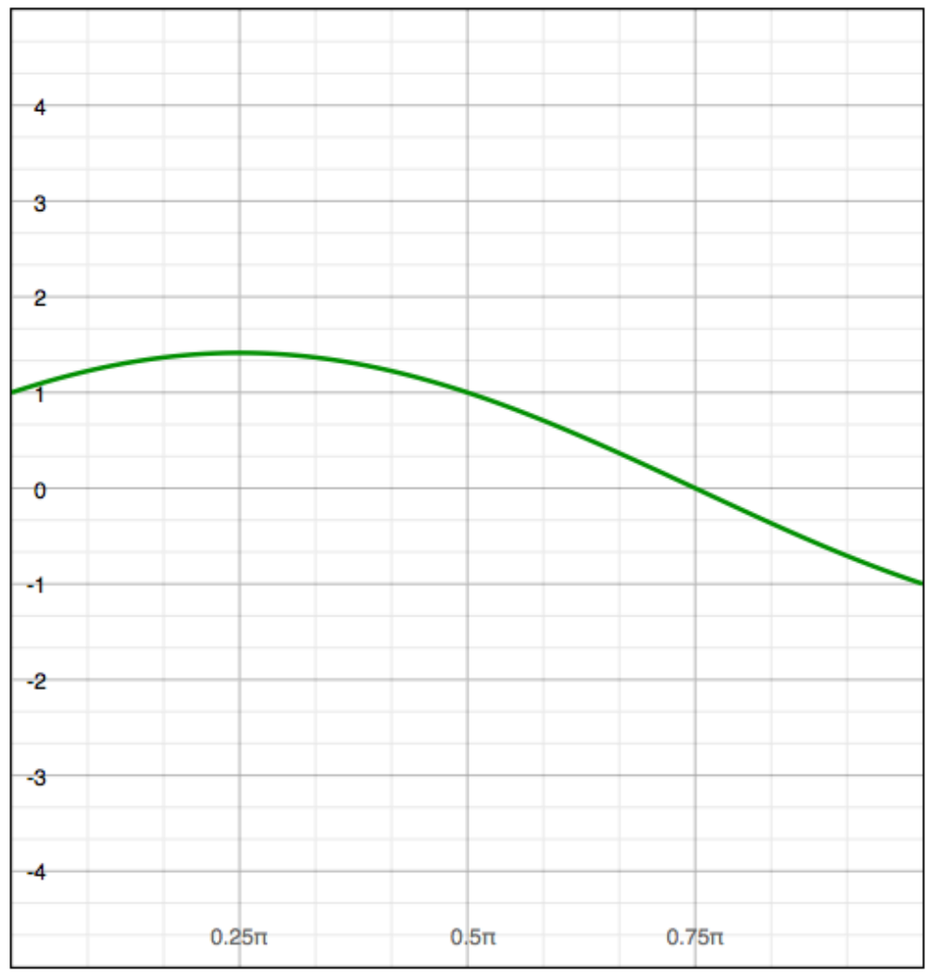


Image space

Parameter space

Image and parameter space

variables

$$y = mx + b$$

parameters

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

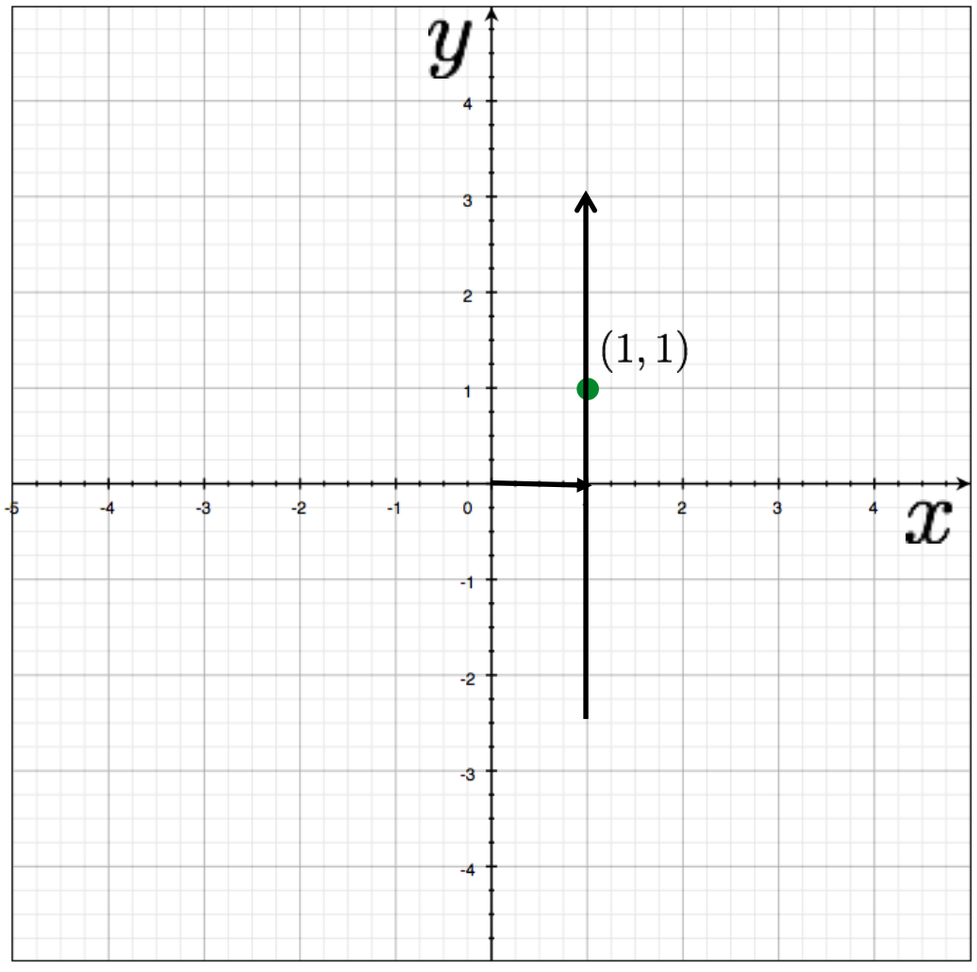
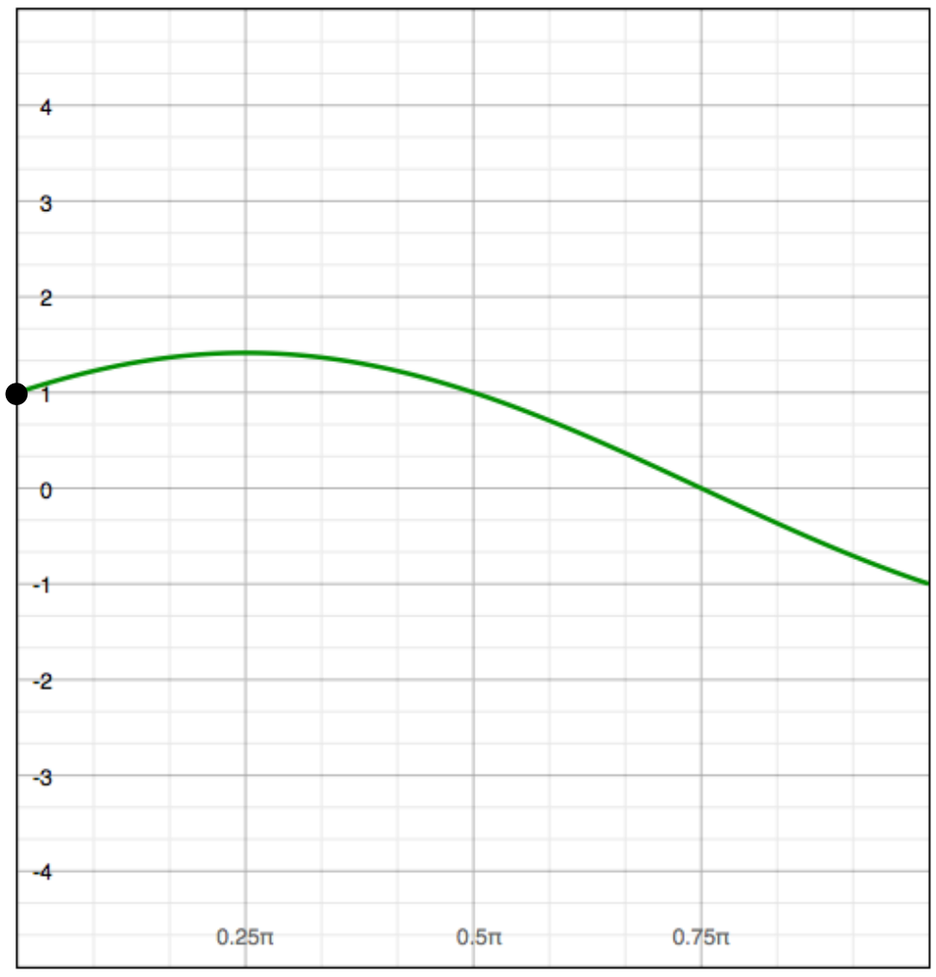


Image space

a line becomes a point



Parameter space

Image and parameter space

variables

$$y = mx + b$$

parameters

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

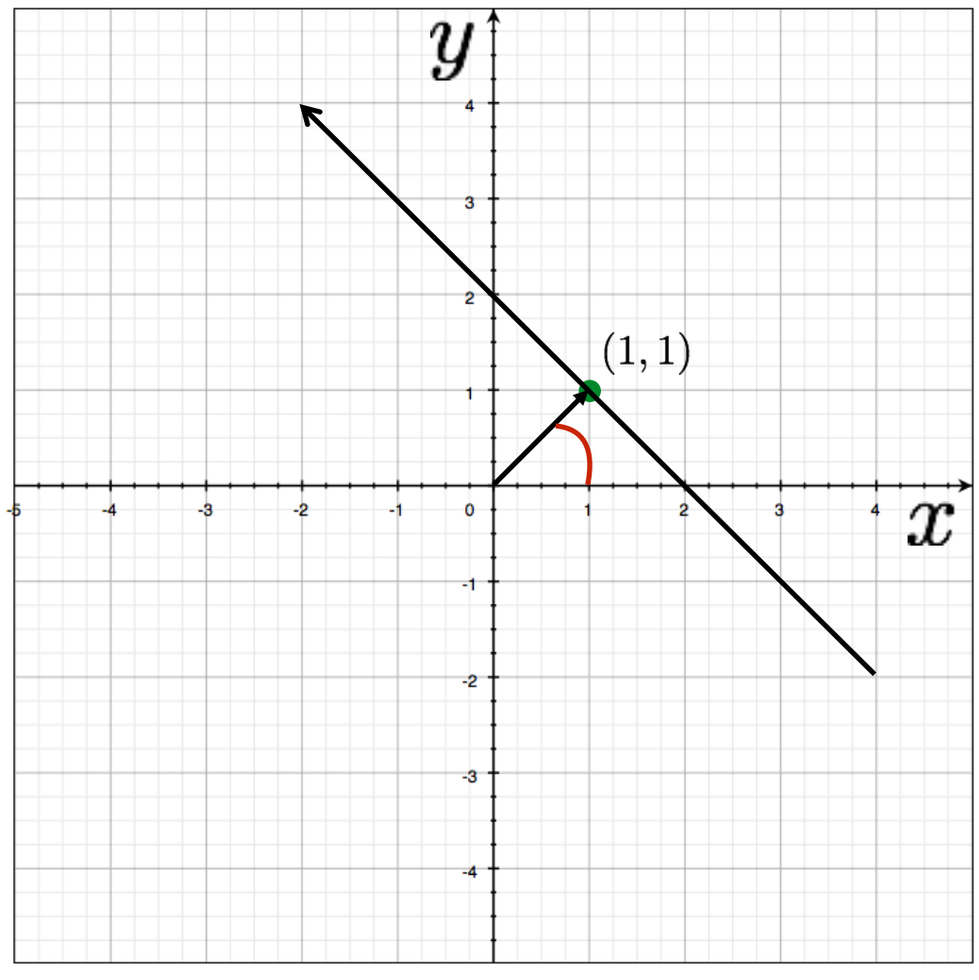
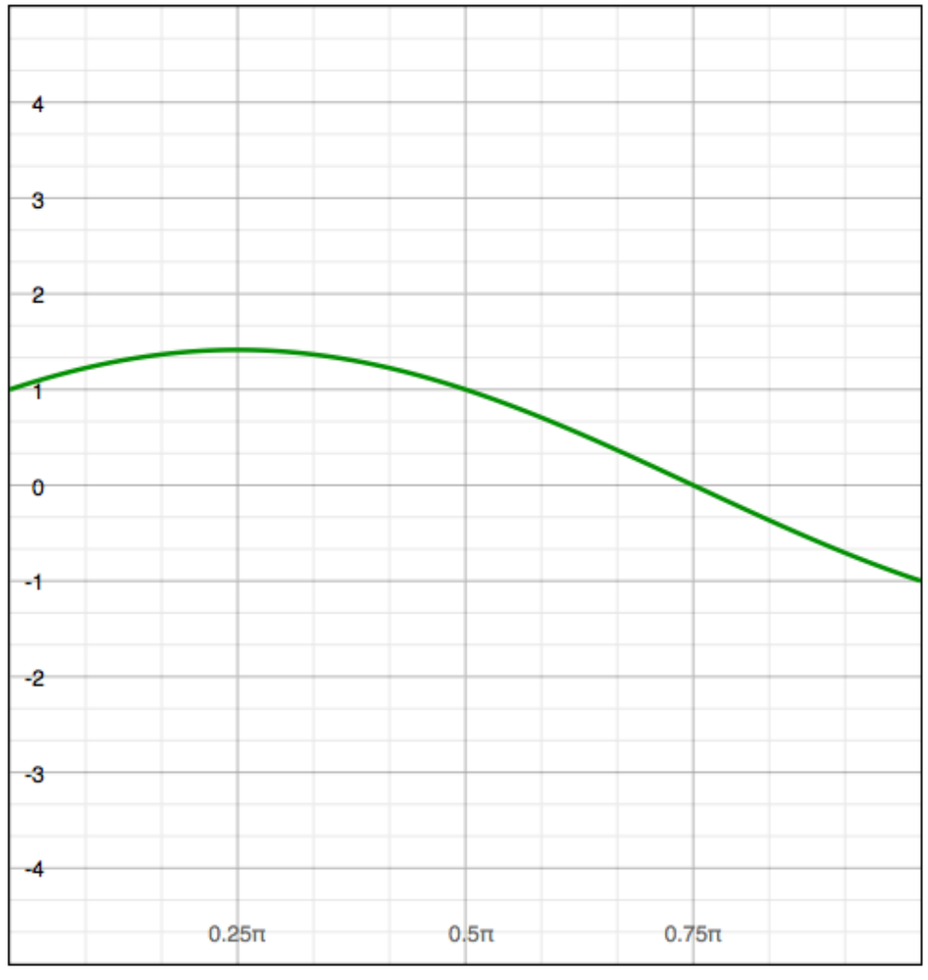


Image space

a line becomes?



Parameter space

Image and parameter space

variables

$$y = mx + b$$

parameters

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

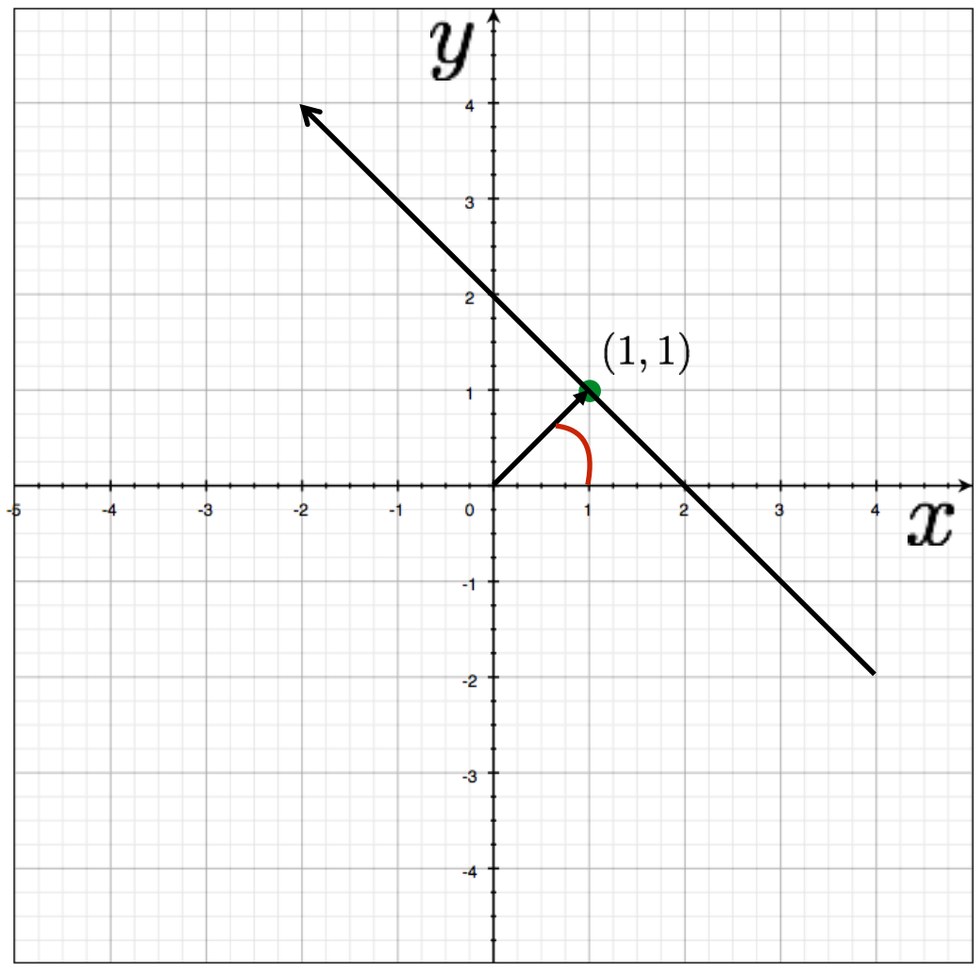
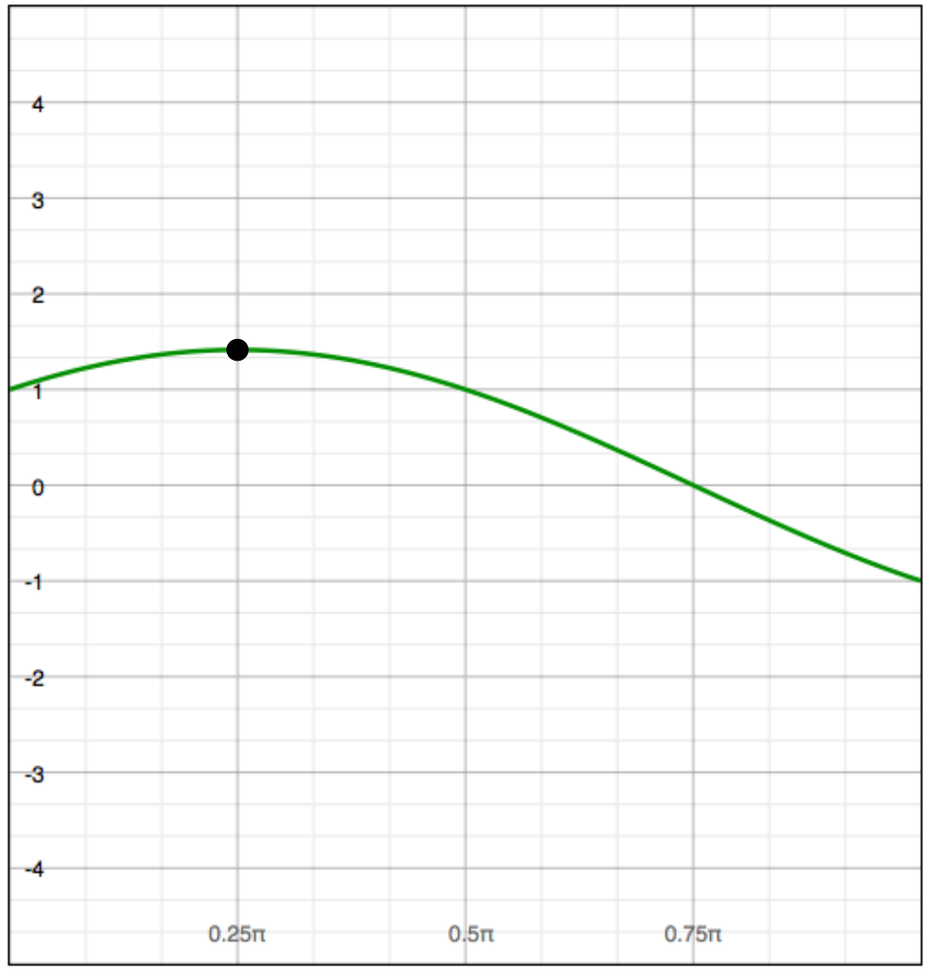


Image space

a line becomes a point



Parameter space

Image and parameter space

variables

$$y = mx + b$$

parameters

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

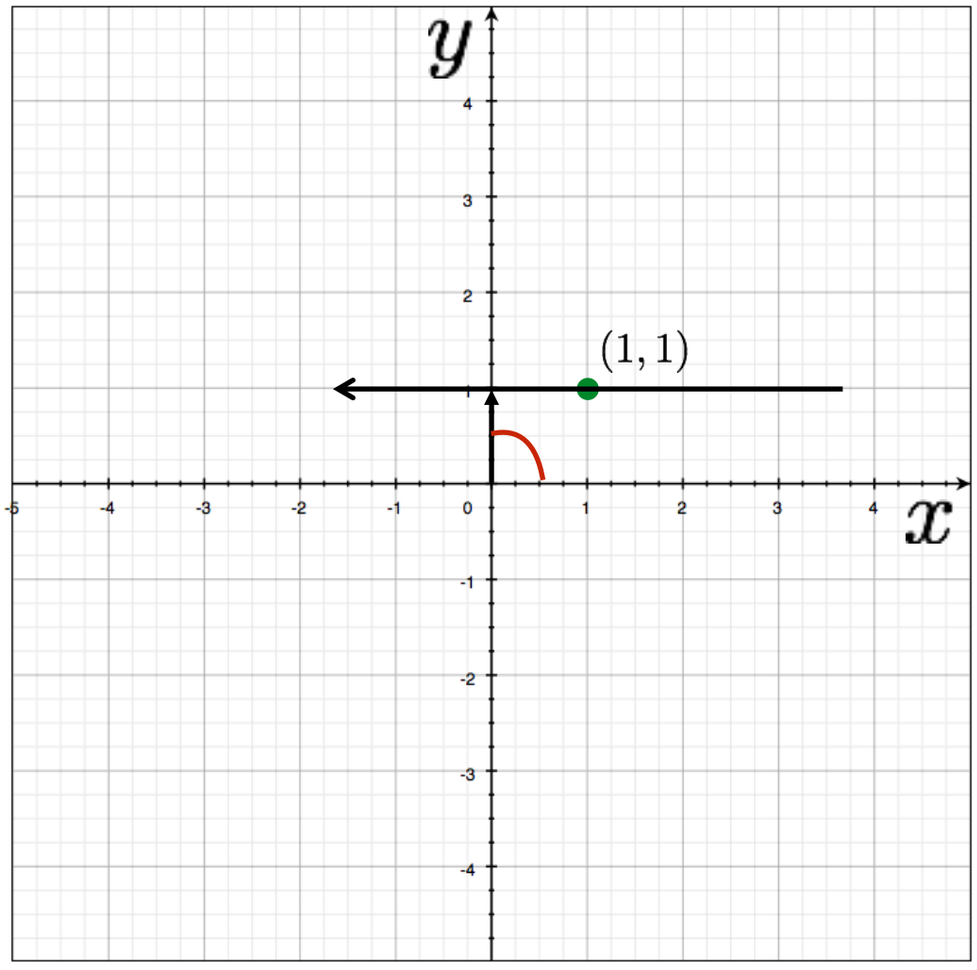
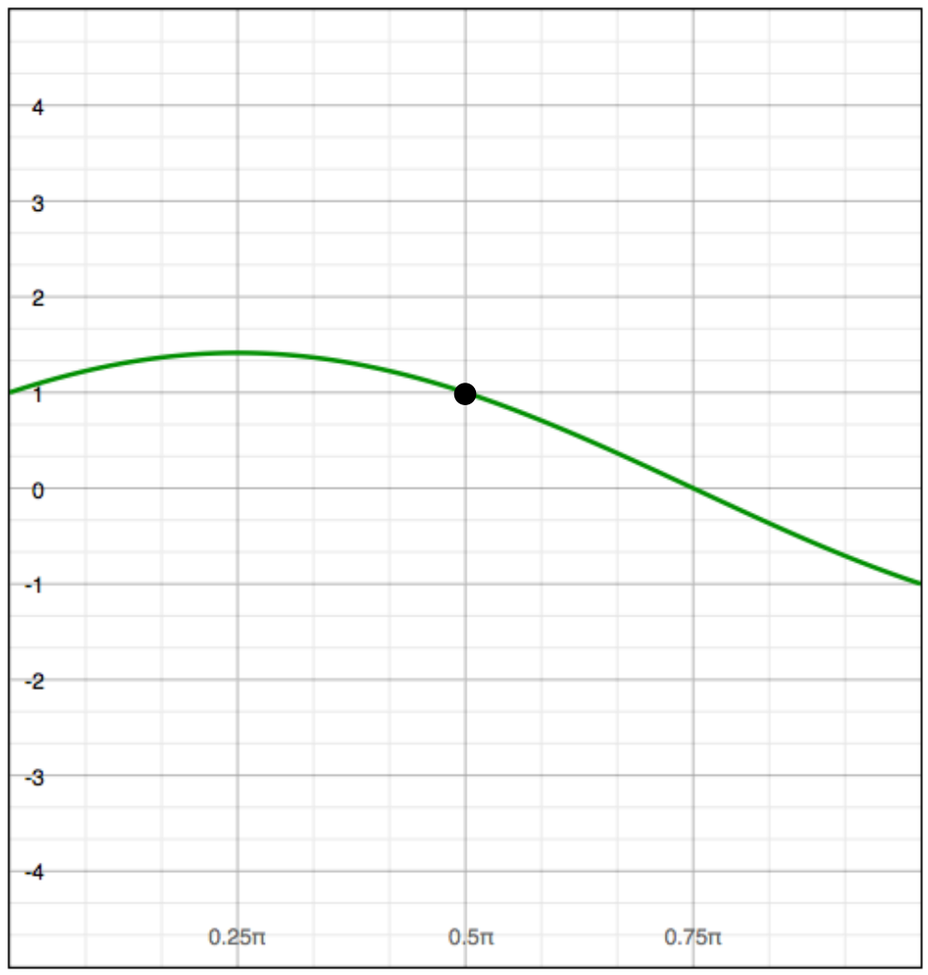


Image space

a line becomes a point



Parameter space

Image and parameter space

variables

$$y = mx + b$$

parameters

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

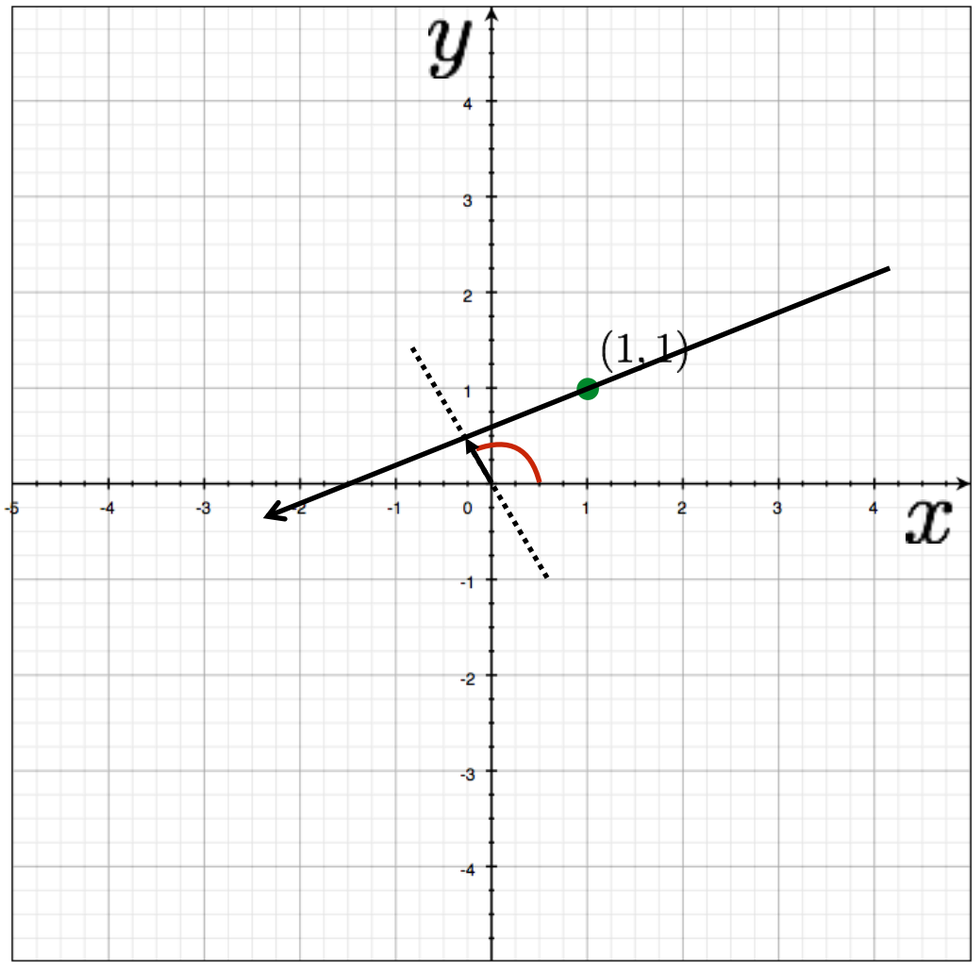
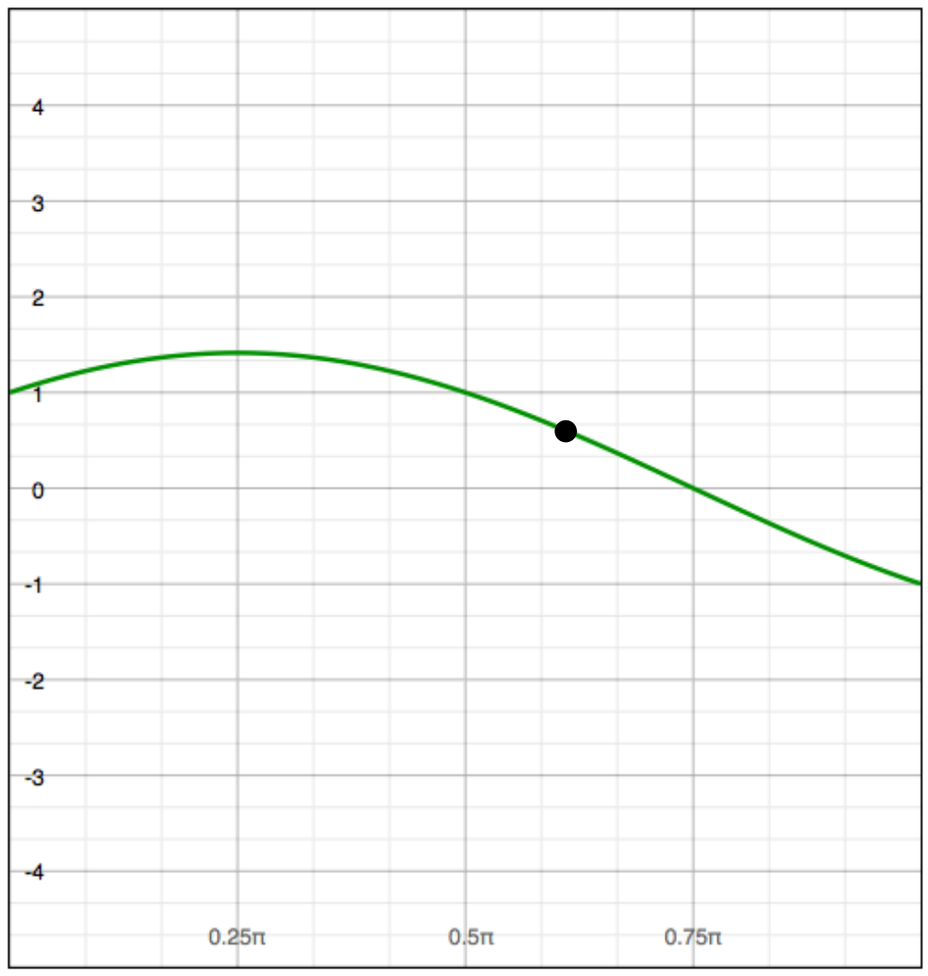


Image space

a line becomes a point



Parameter space

Image and parameter space

variables

$$y = mx + b$$

parameters

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

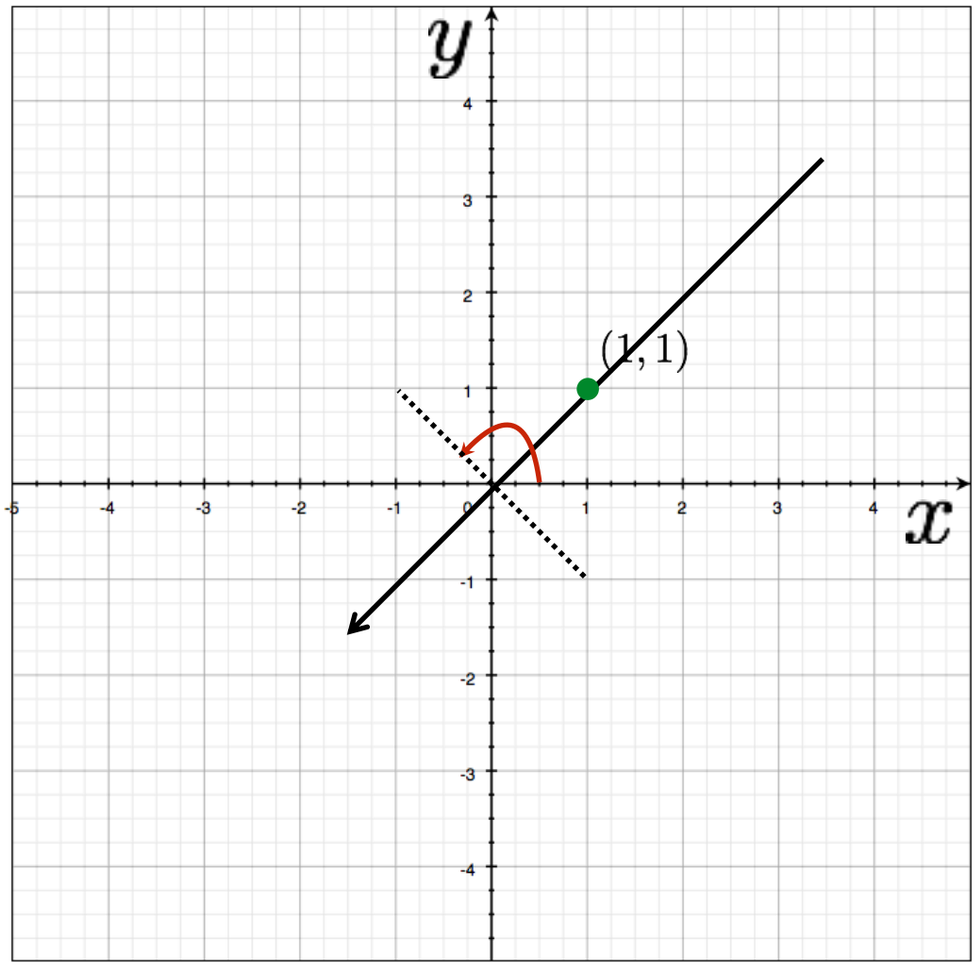
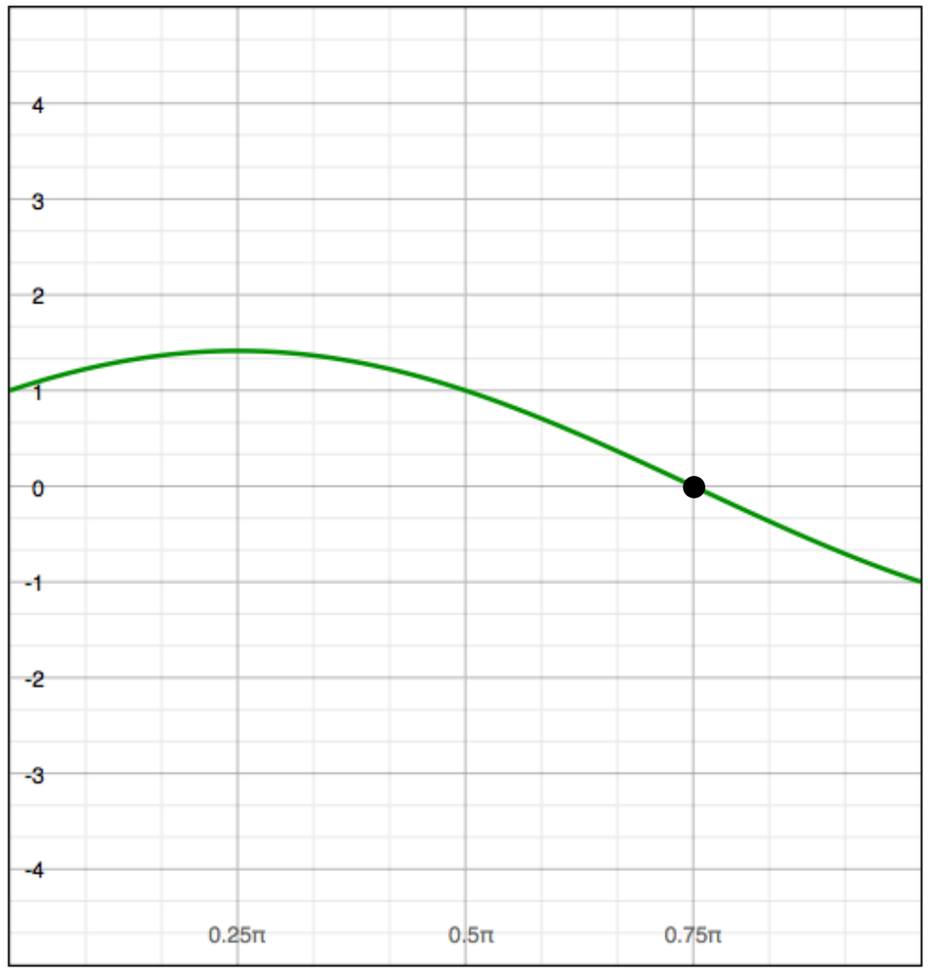


Image space

a line becomes a point



Parameter space

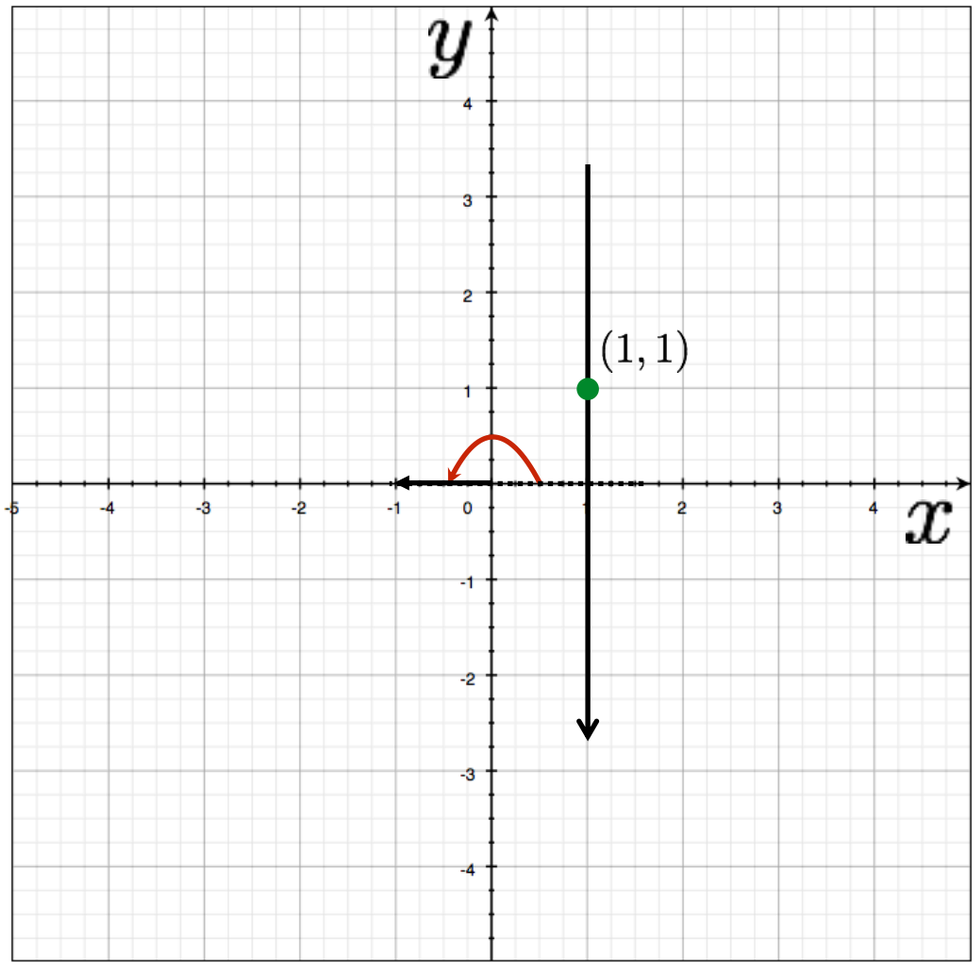
Image and parameter space

variables

$$y = mx + b$$

parameters

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



a line becomes a point

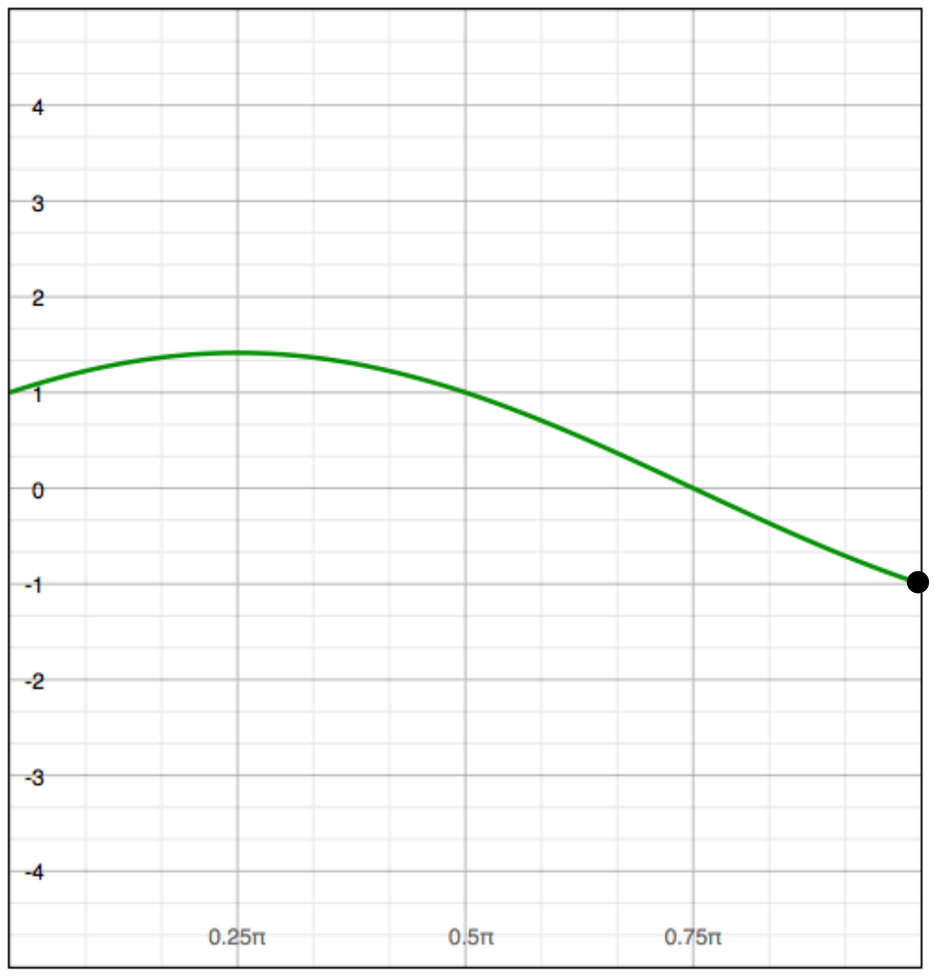


Image space

Parameter space

Image and parameter space

variables

$$y = mx + b$$

parameters

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

Wait …why is rho negative?

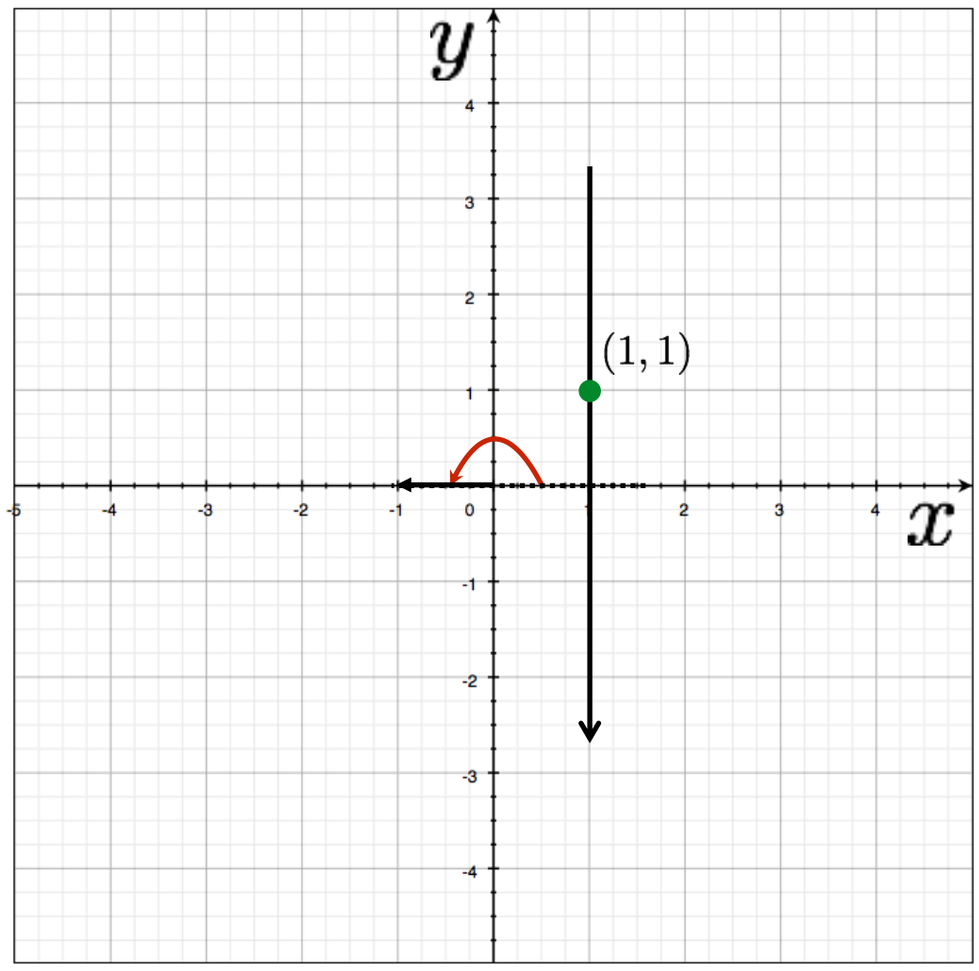
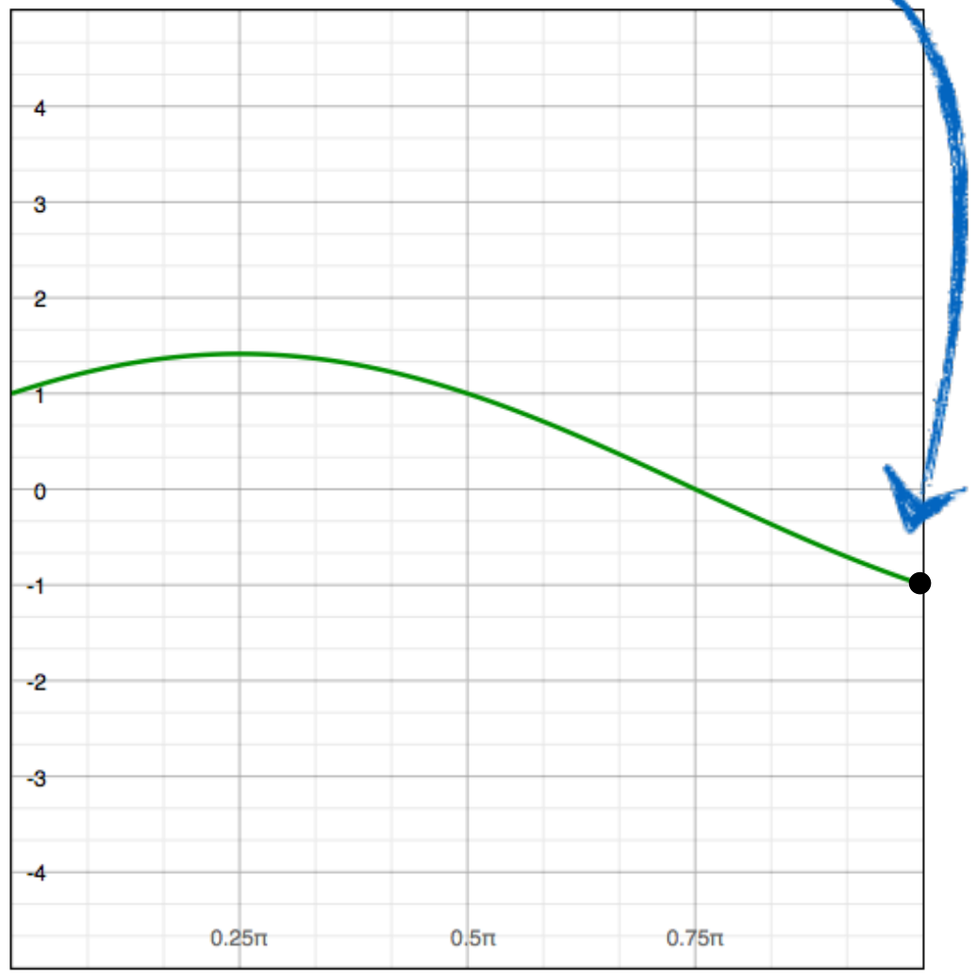


Image space

a line becomes a point



Parameter space

Image and parameter space

variables

$$y = mx + b$$

parameters

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

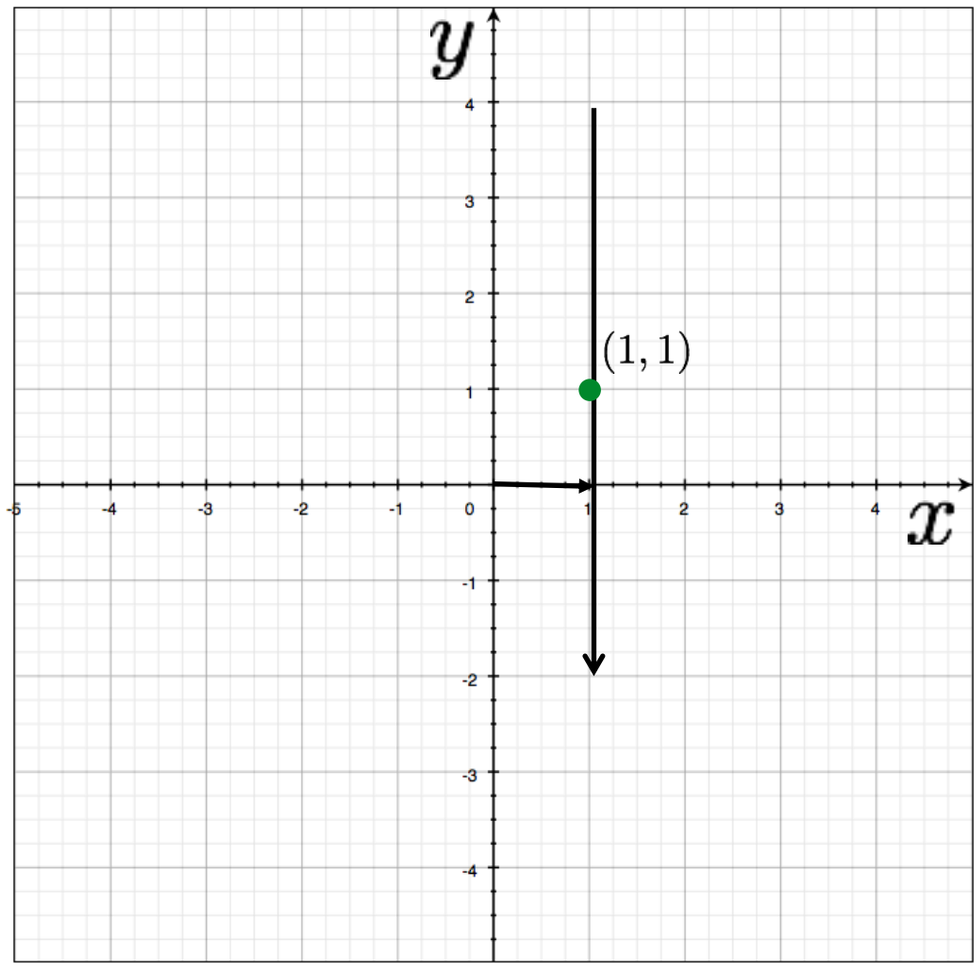
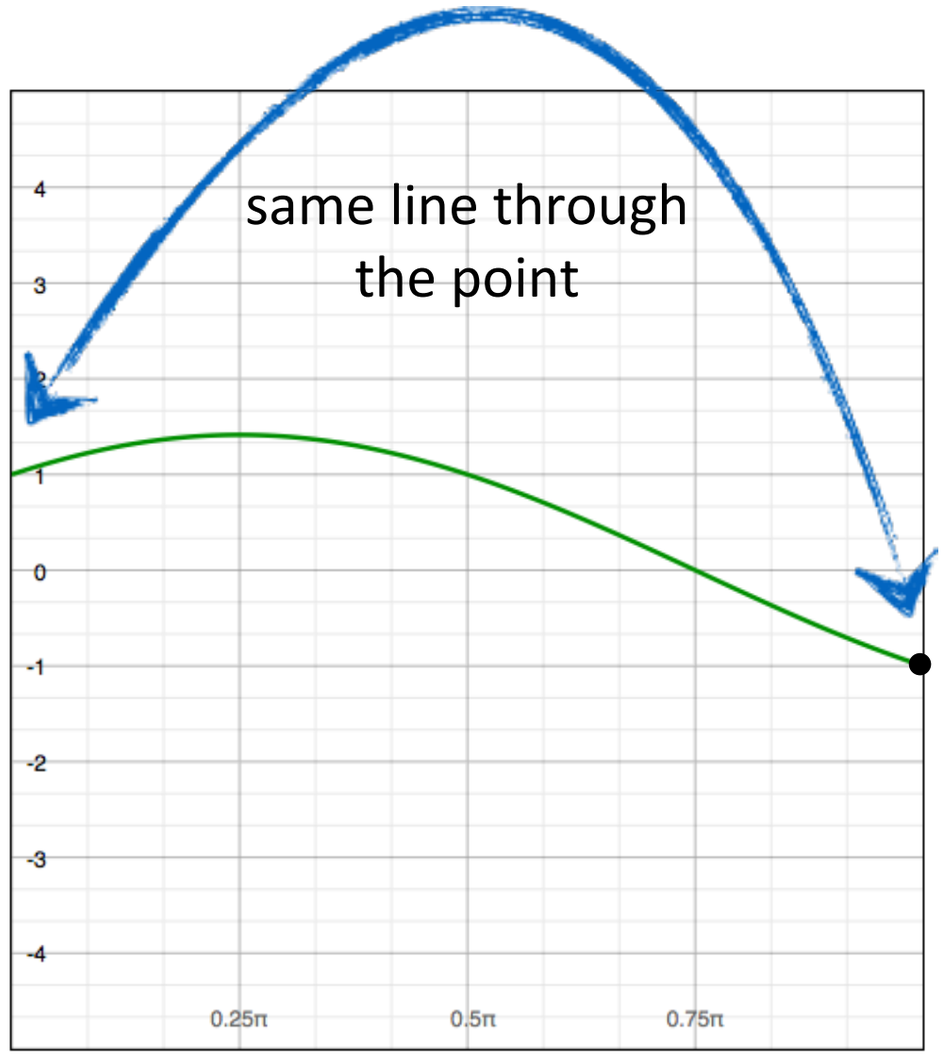


Image space

a line becomes a point

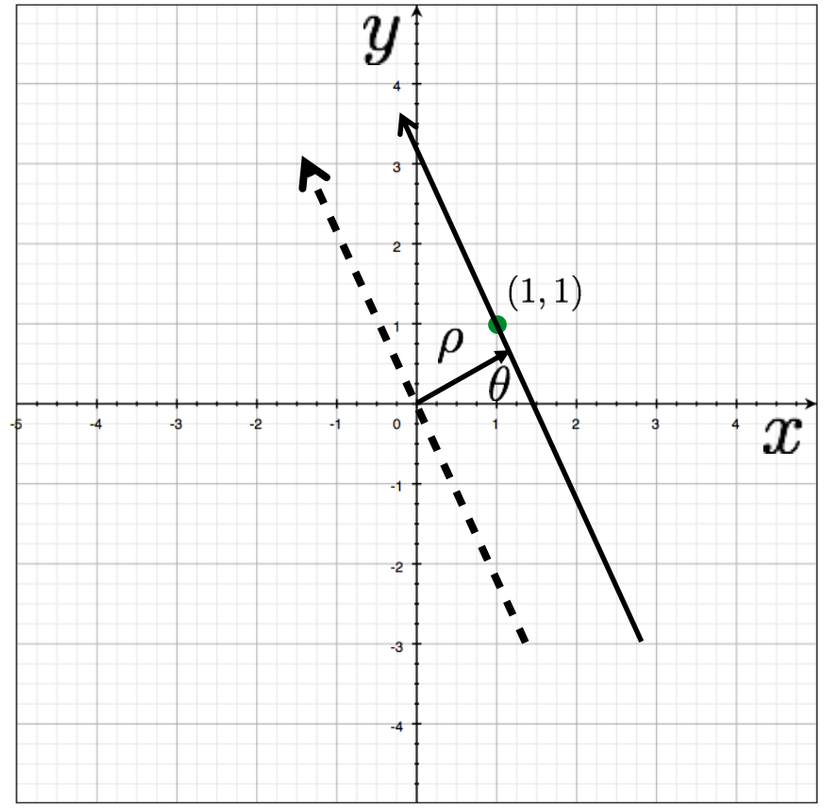


Parameter space

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:

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

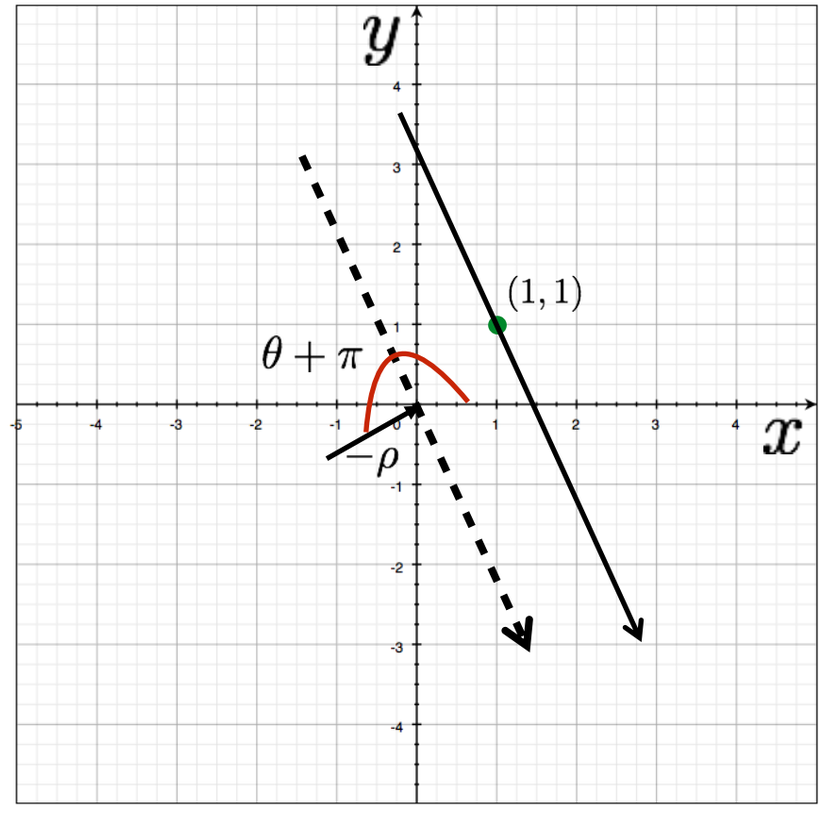


Image and parameter space

variables

$$y = mx + b$$

parameters

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

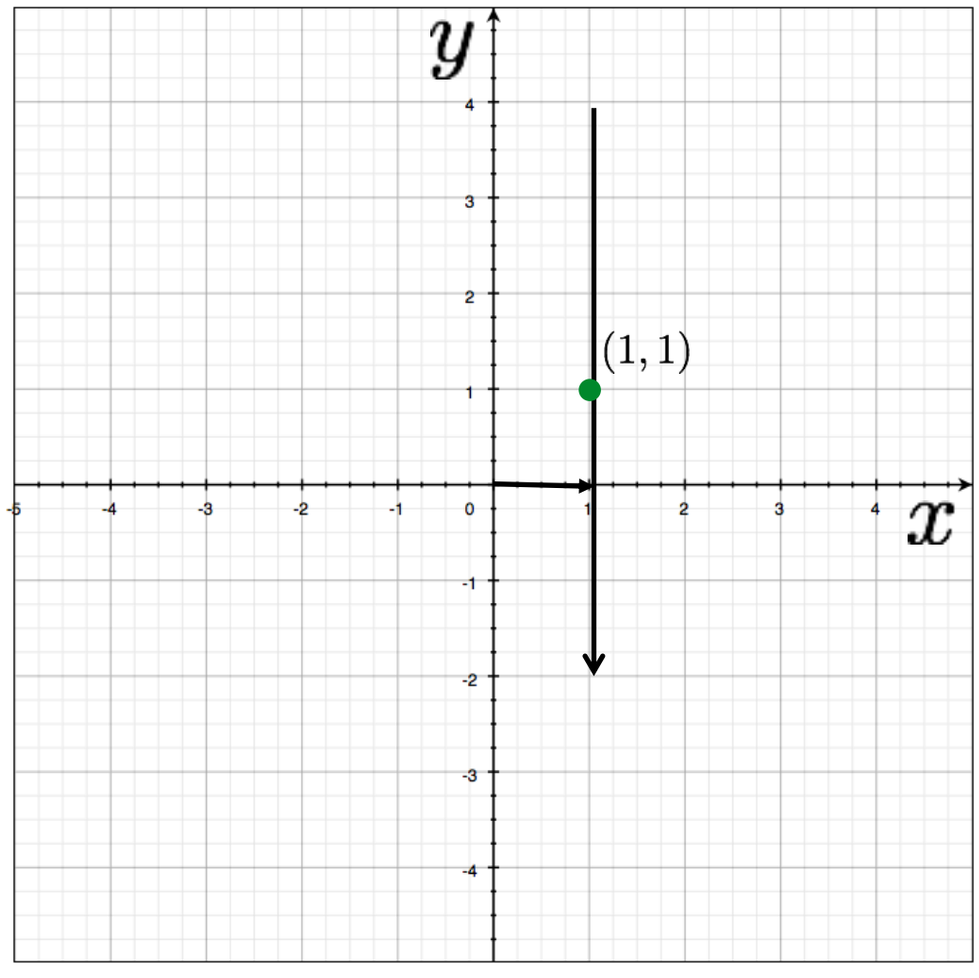
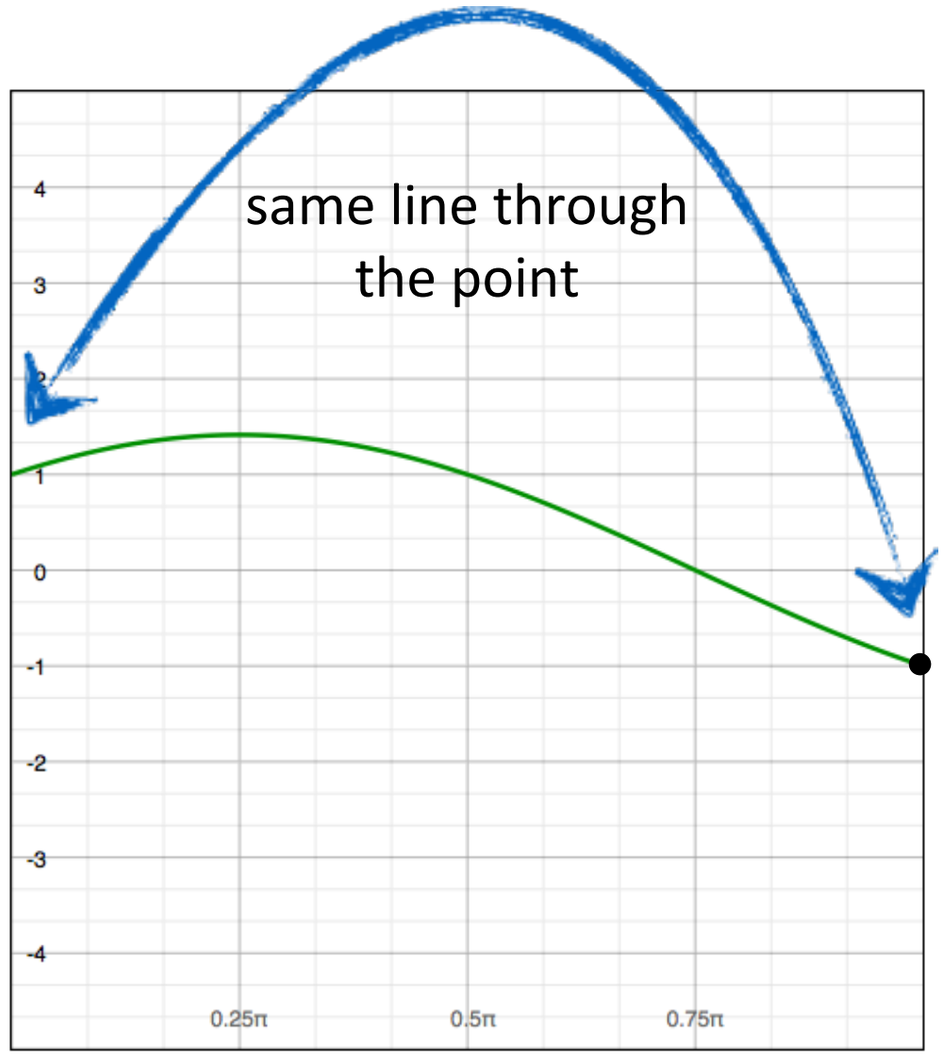


Image space

a line becomes a point



Parameter space

Image and parameter space

variables

$$y = mx + b$$

parameters

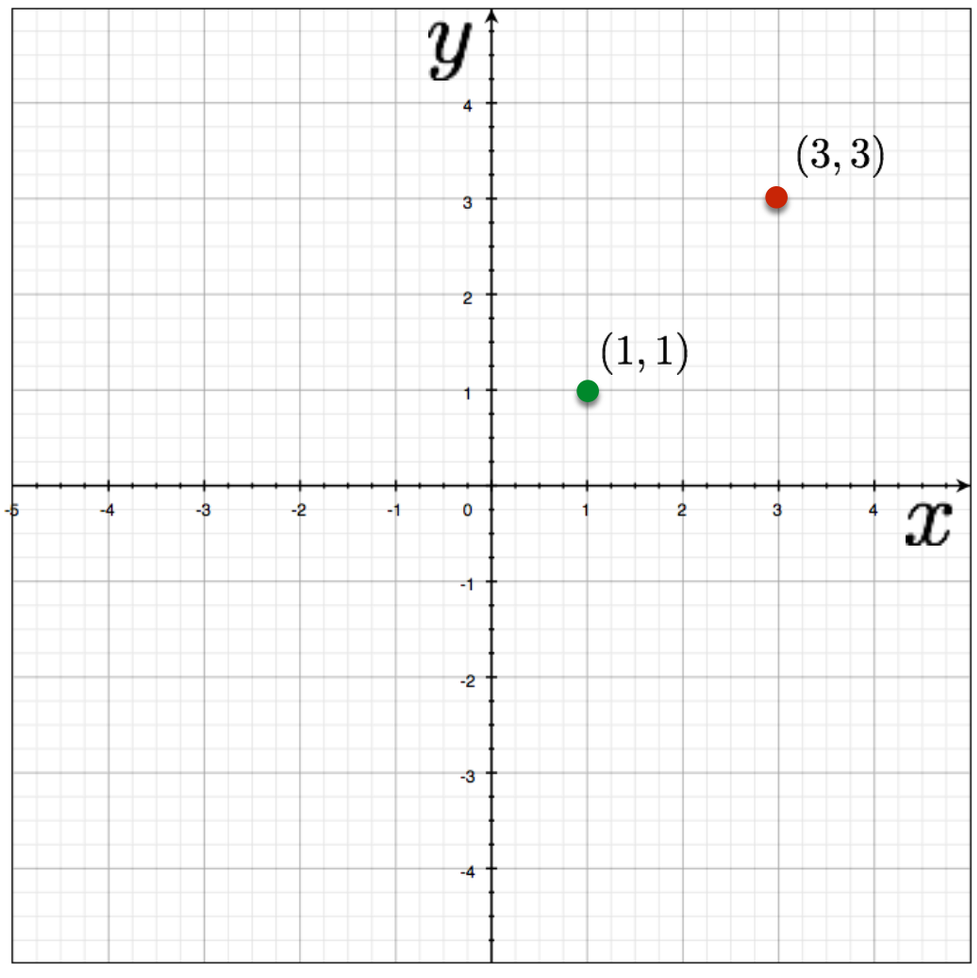
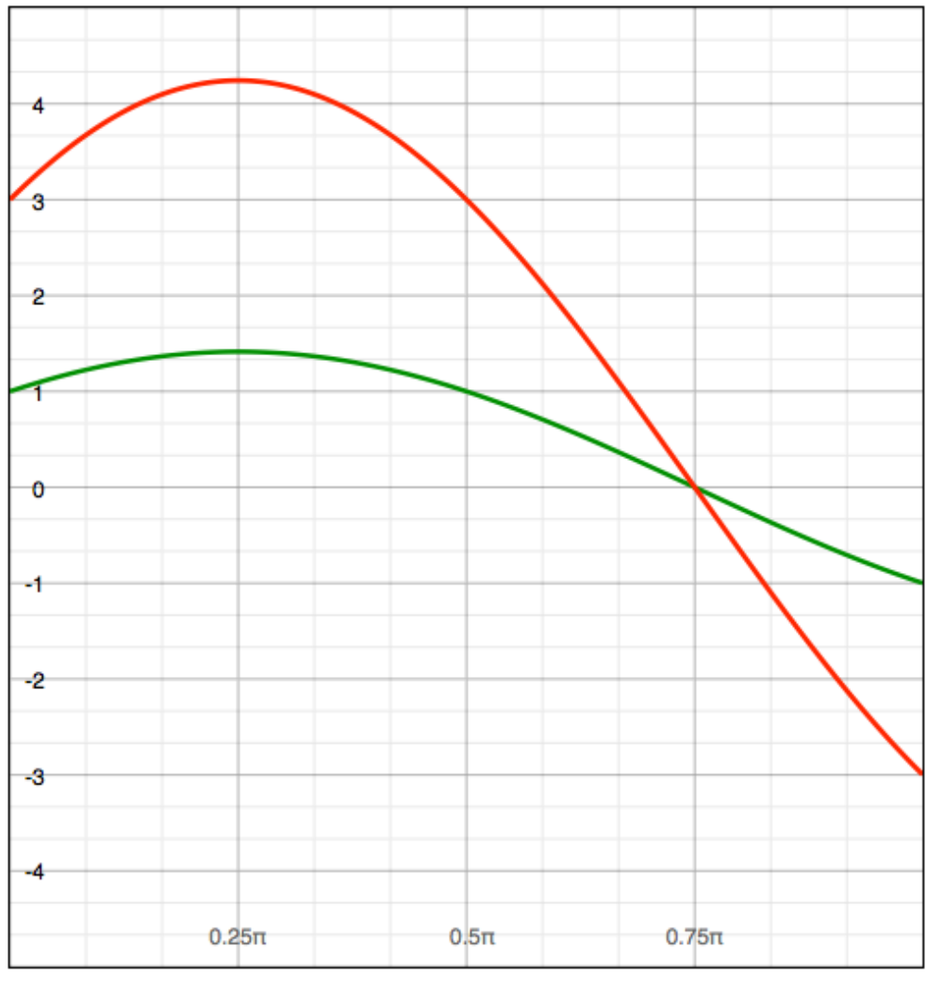


Image space

two points become ?



Parameter space

Image and parameter space

variables

$$y = mx + b$$

parameters

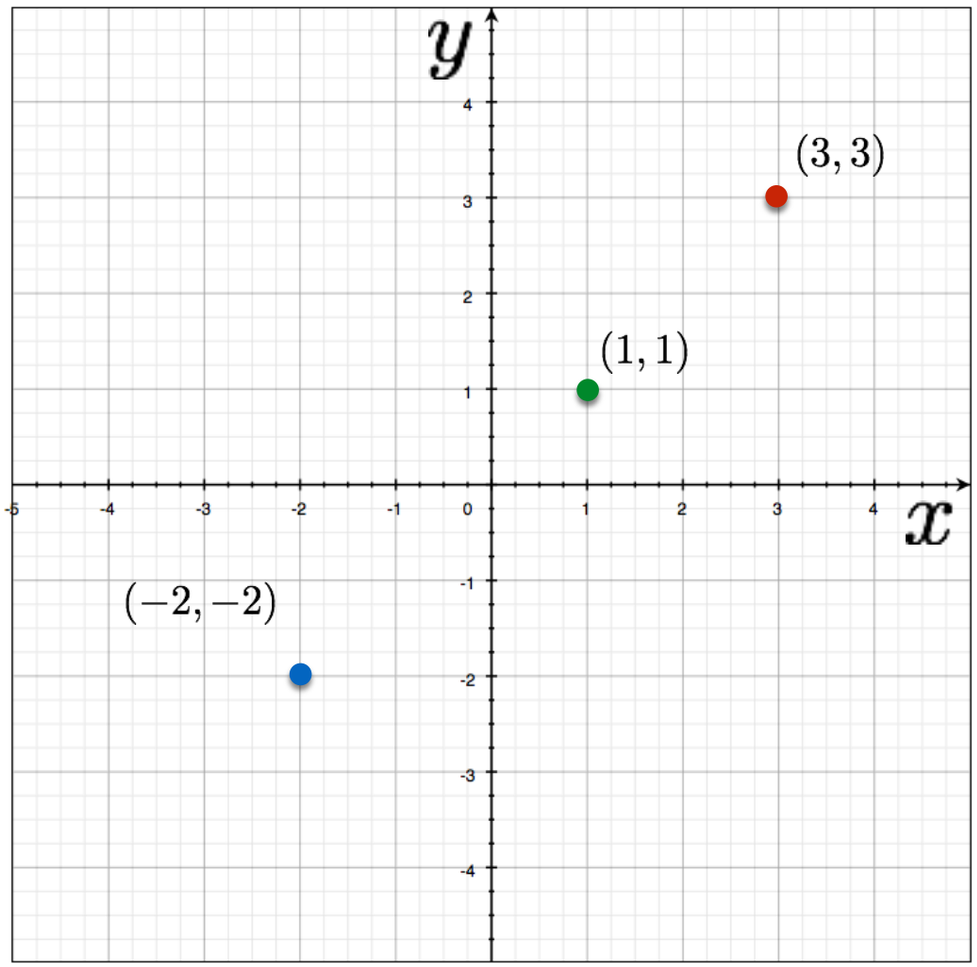
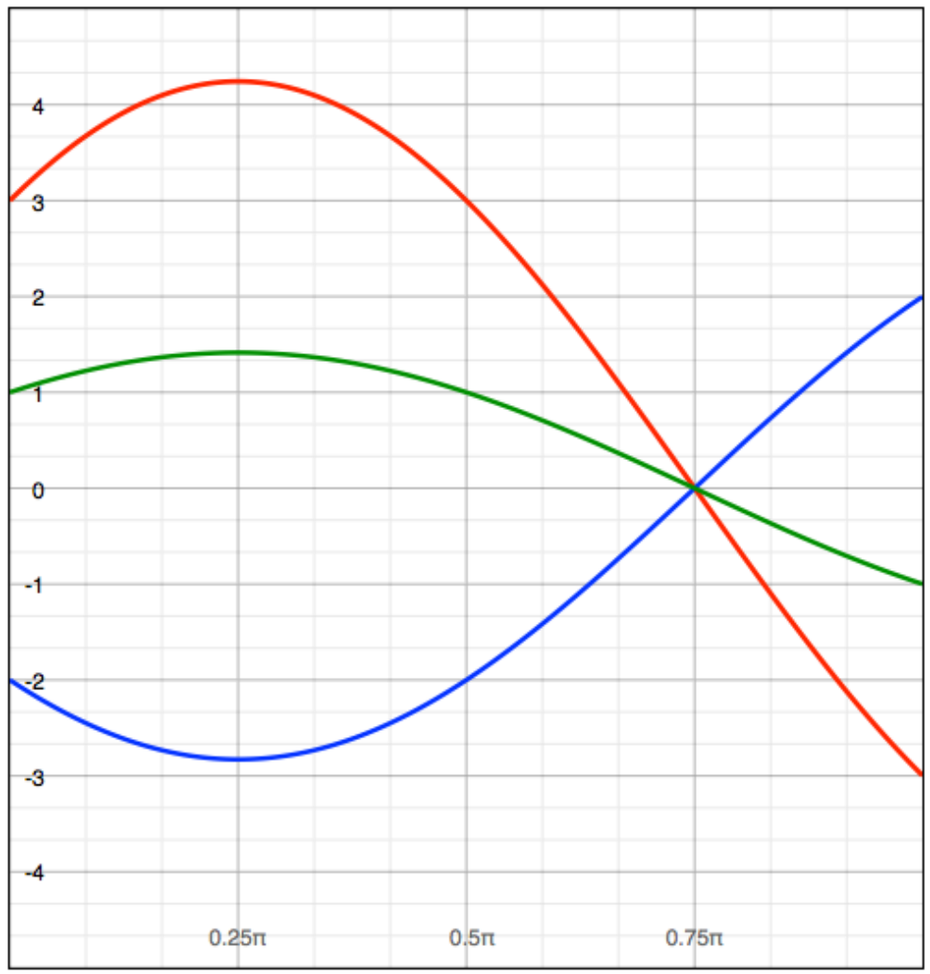


Image space

three points become ?



Parameter space

Image and parameter space

variables

$$y = mx + b$$

parameters

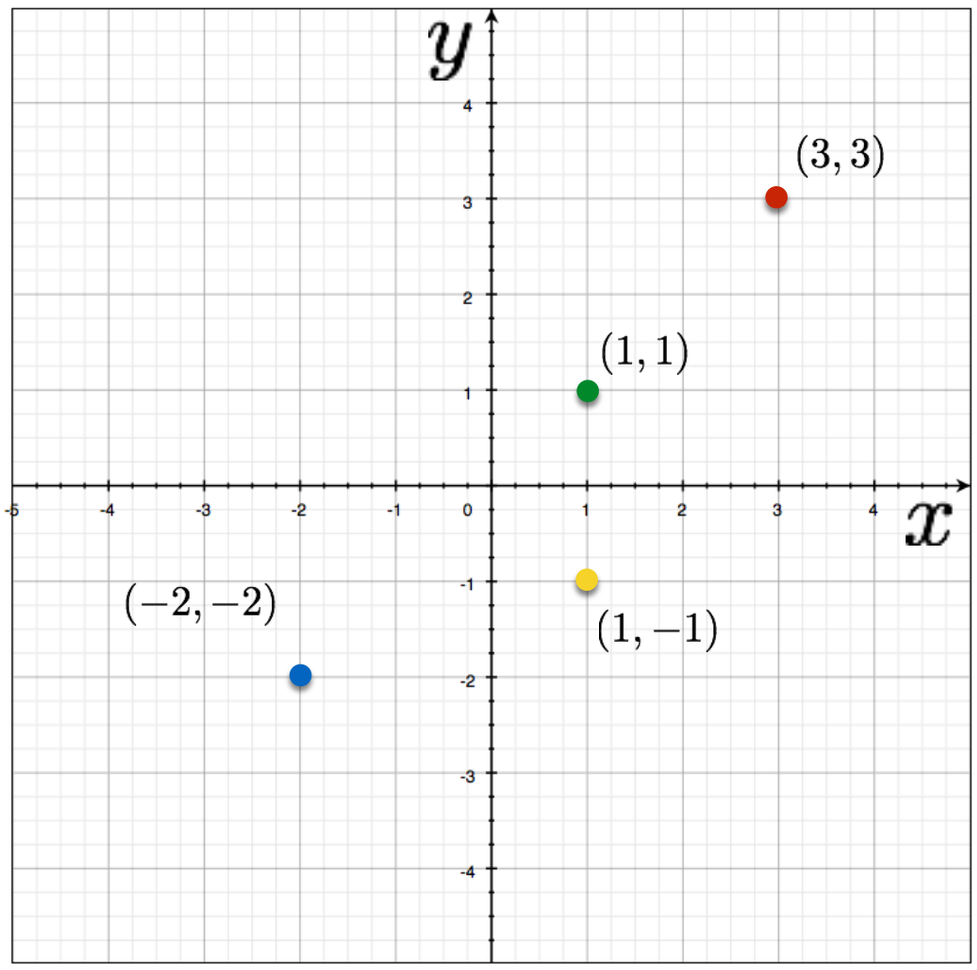
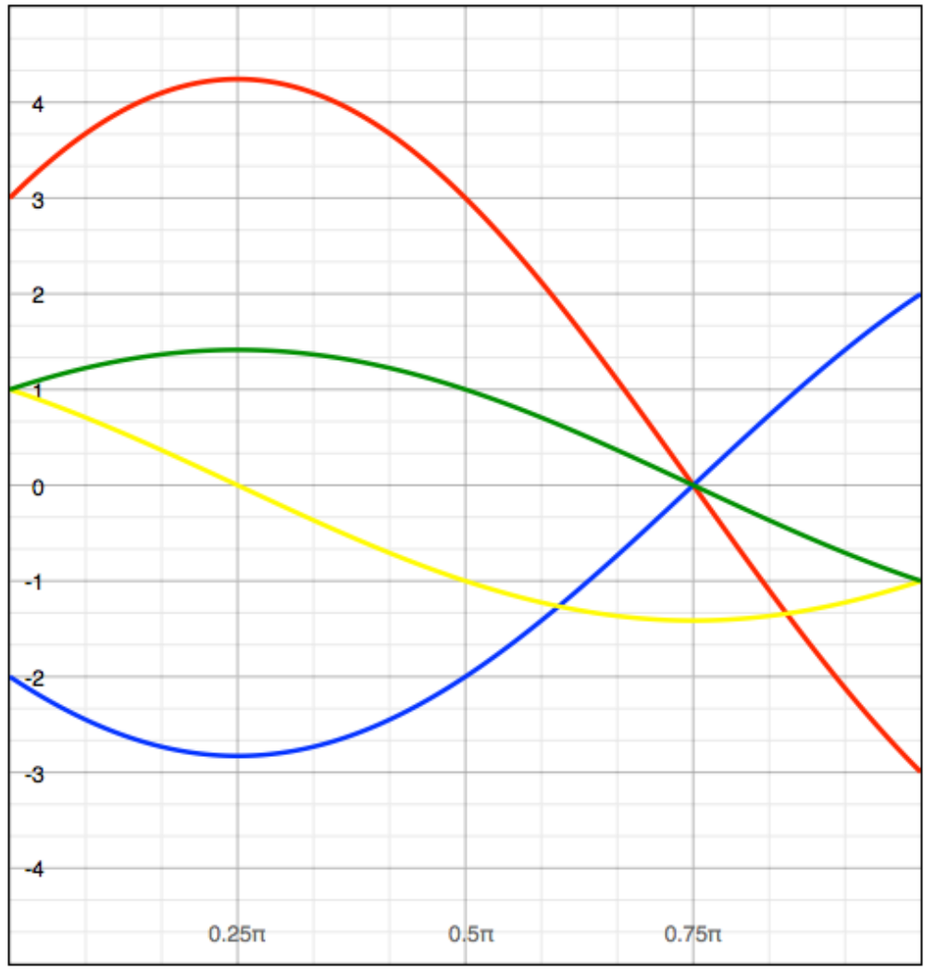


Image space

four points become ?



Parameter space

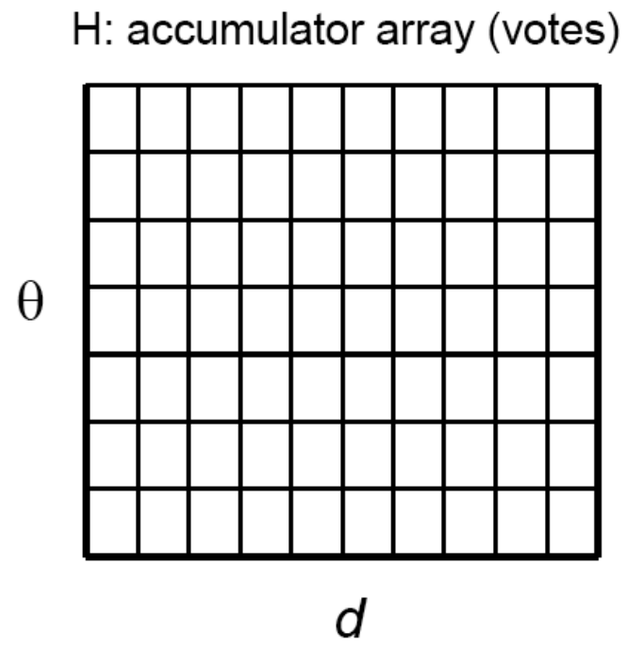
Implementation

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



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

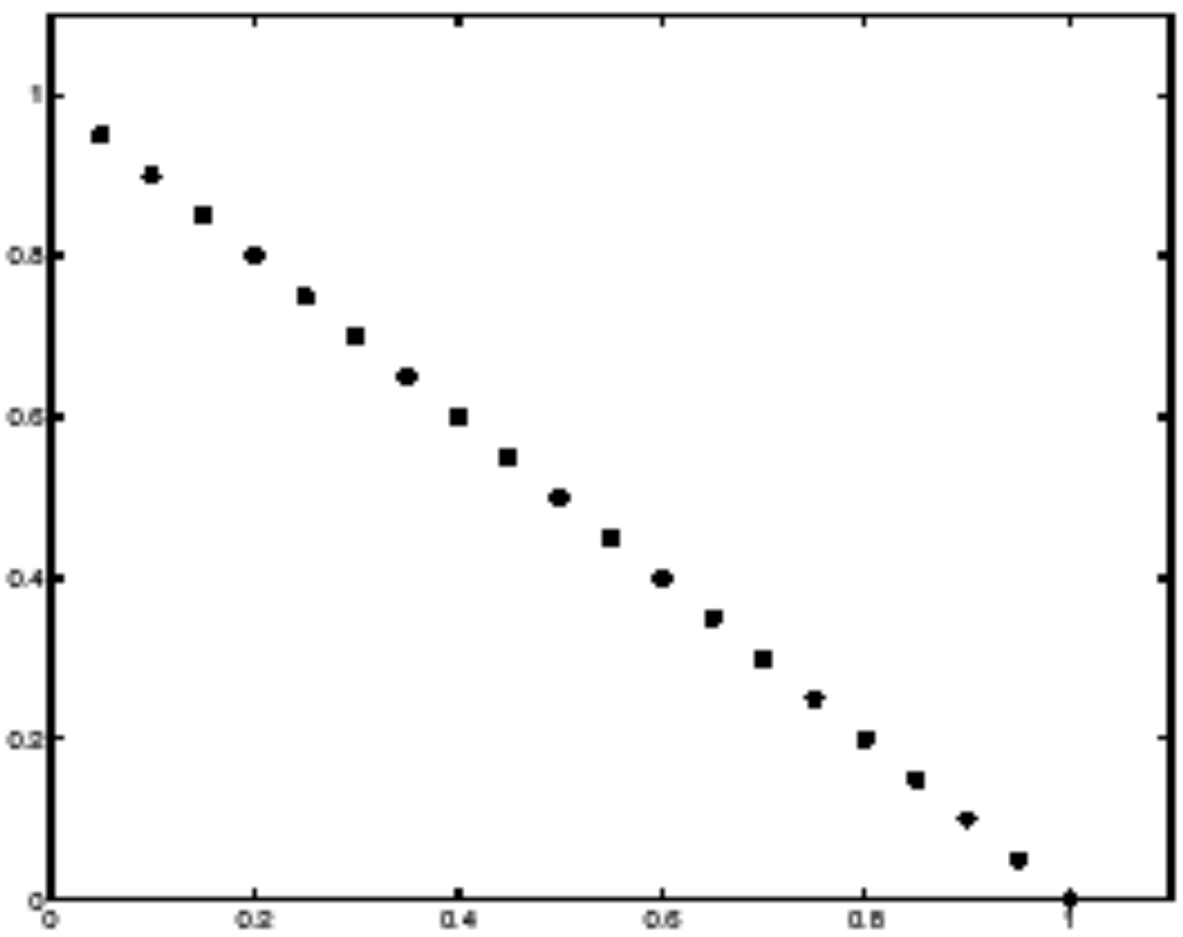
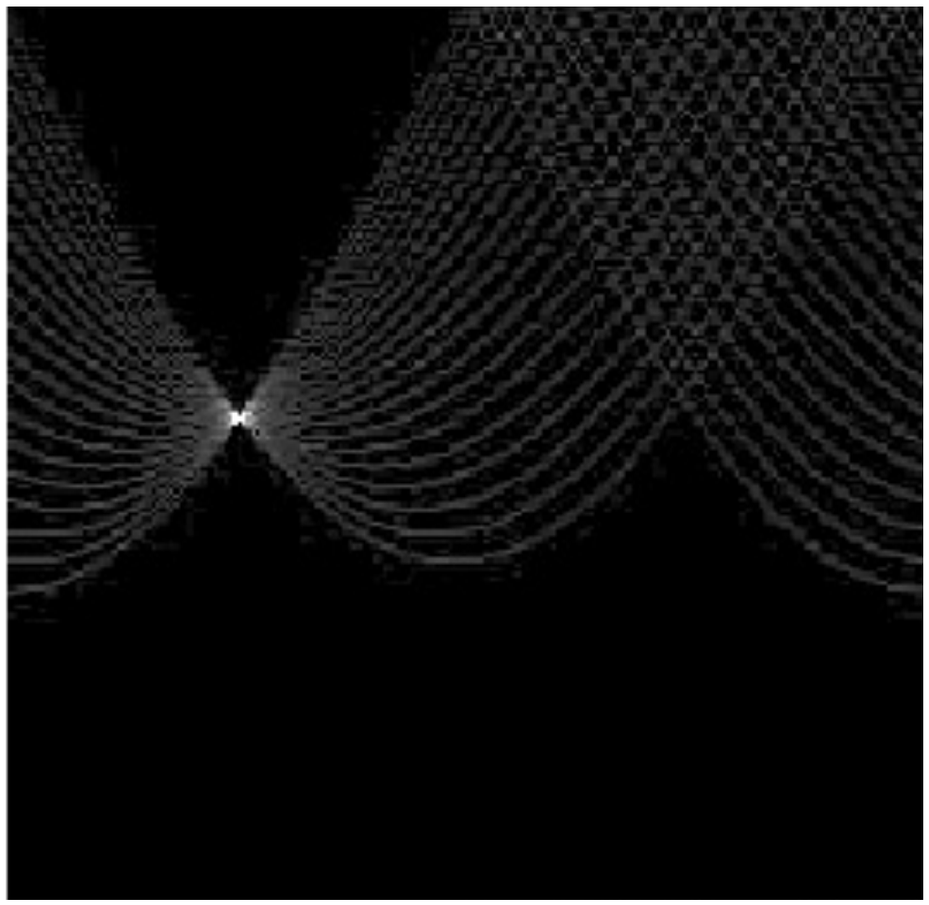


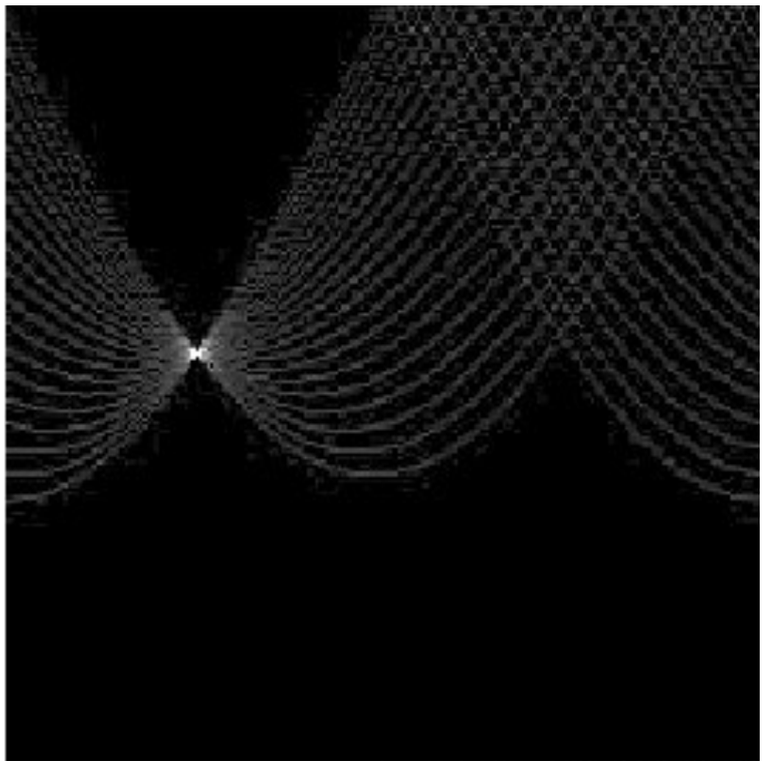
Image space

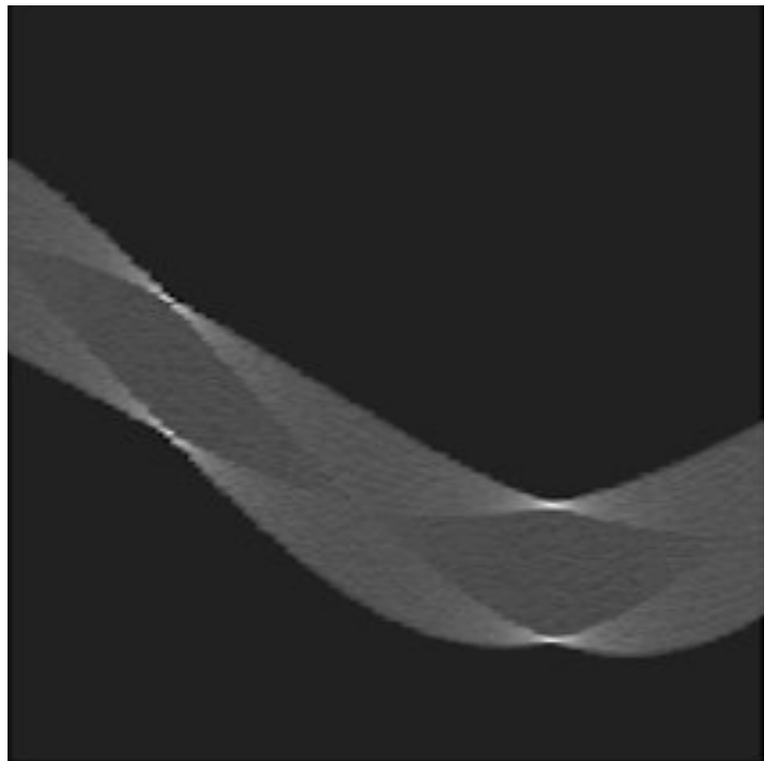


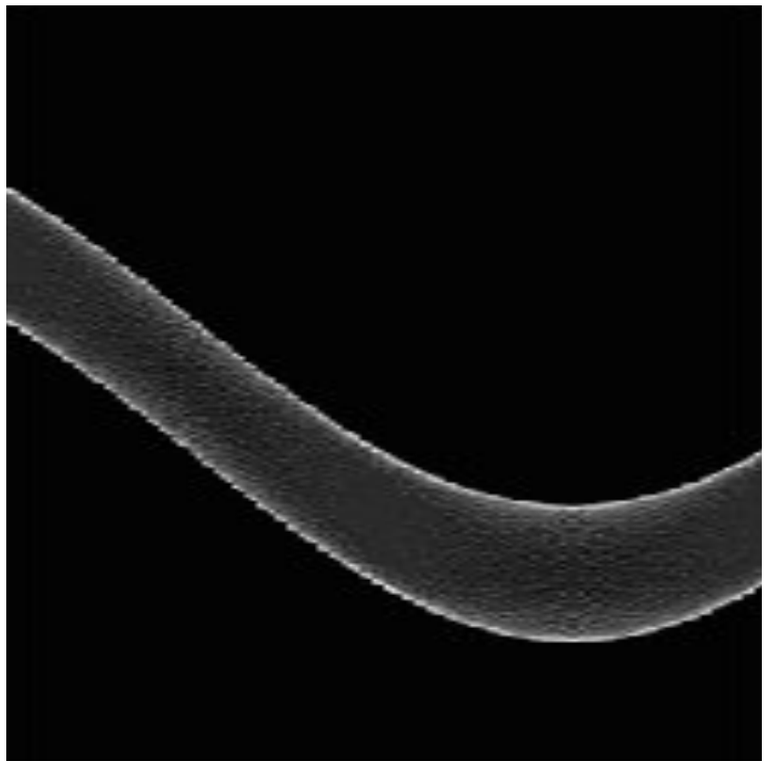
Votes

Basic shapes

(in parameter space)



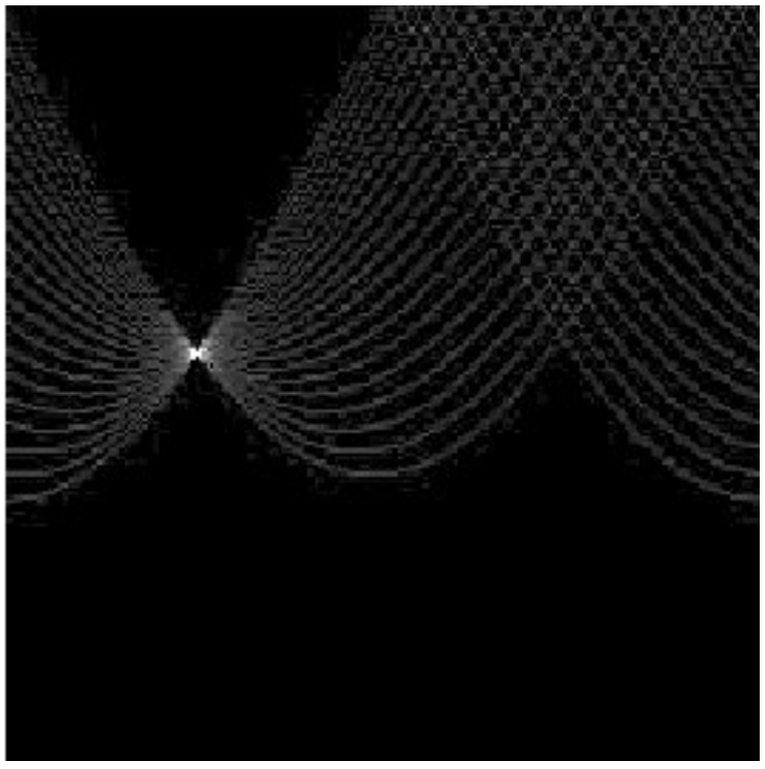


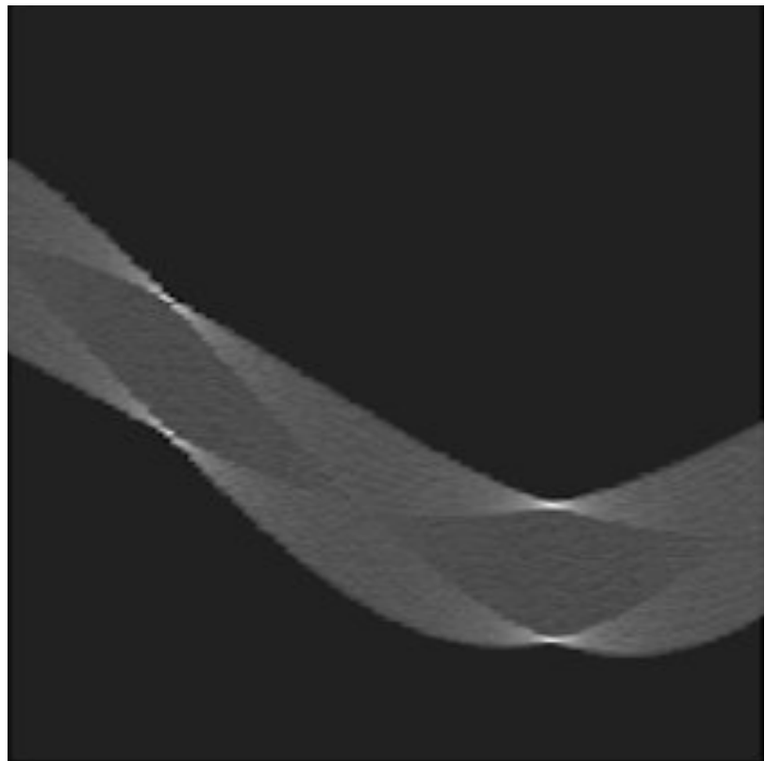


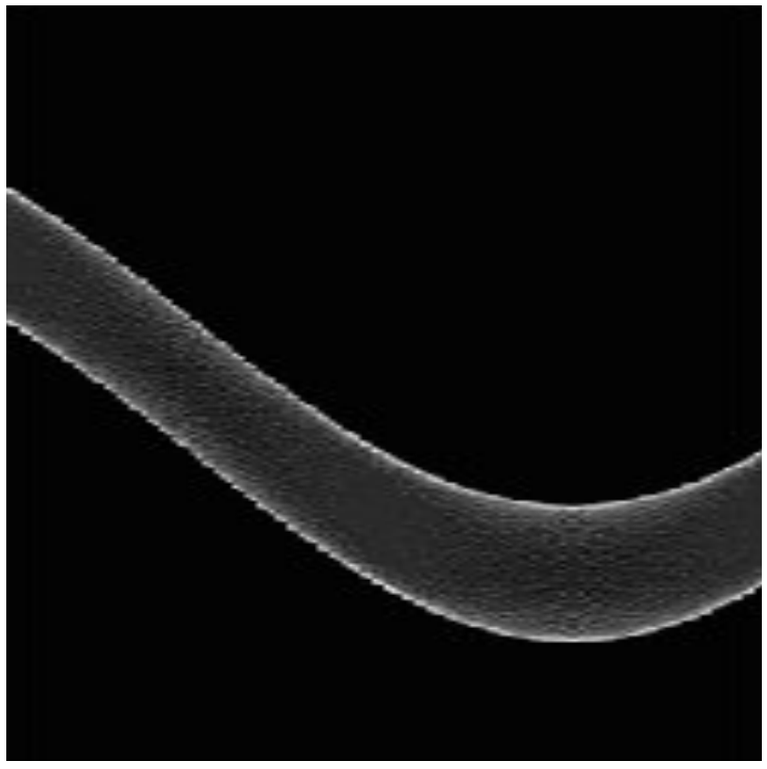
can you guess the shape?

Basic shapes

(in parameter space)



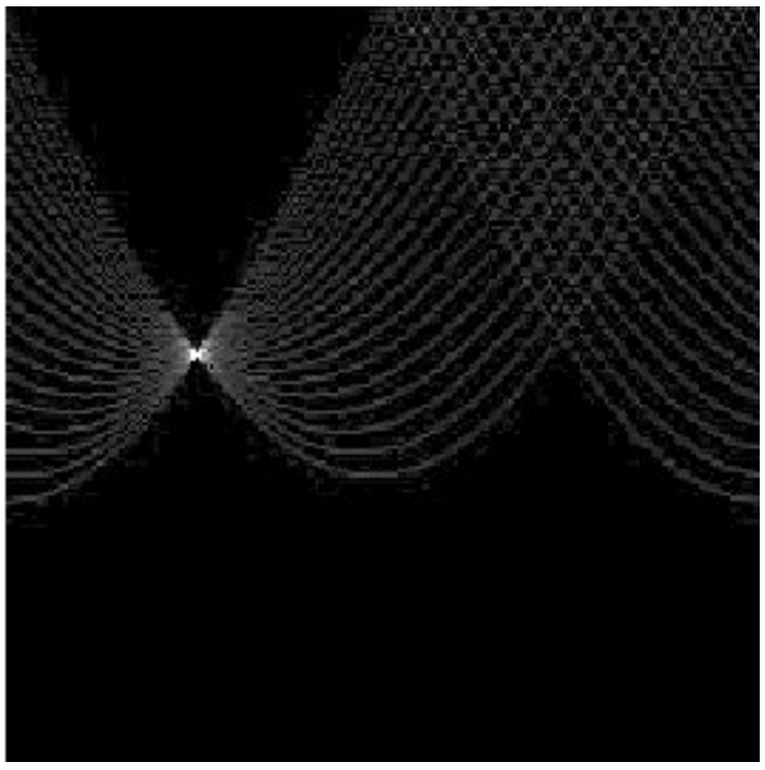




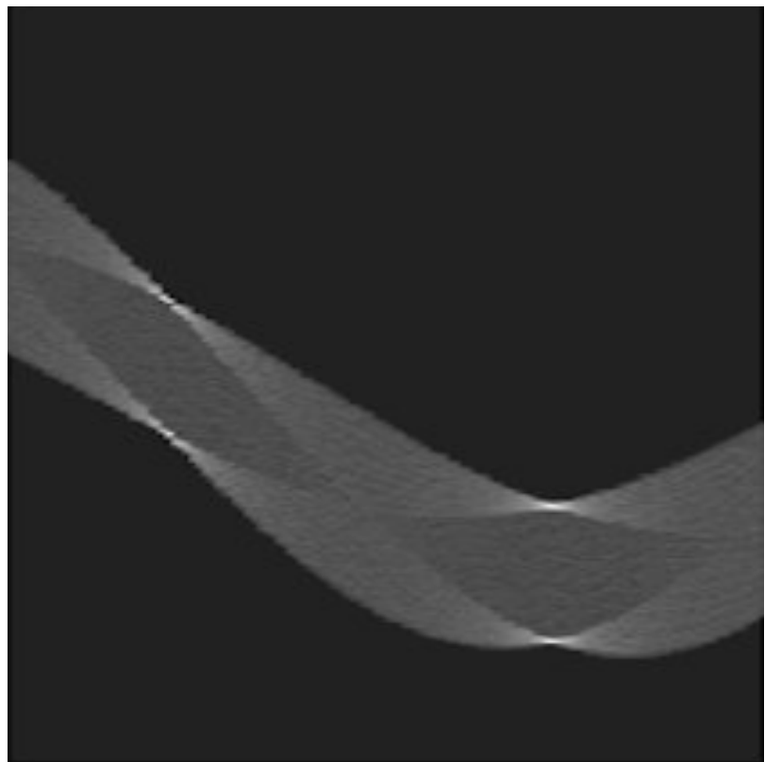
line

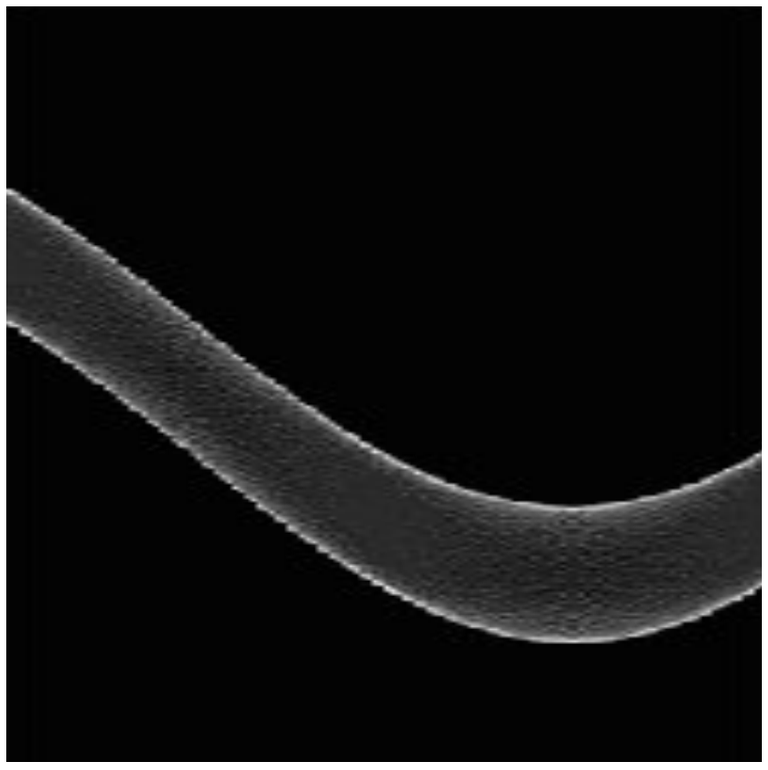
Basic shapes

(in parameter space)



line

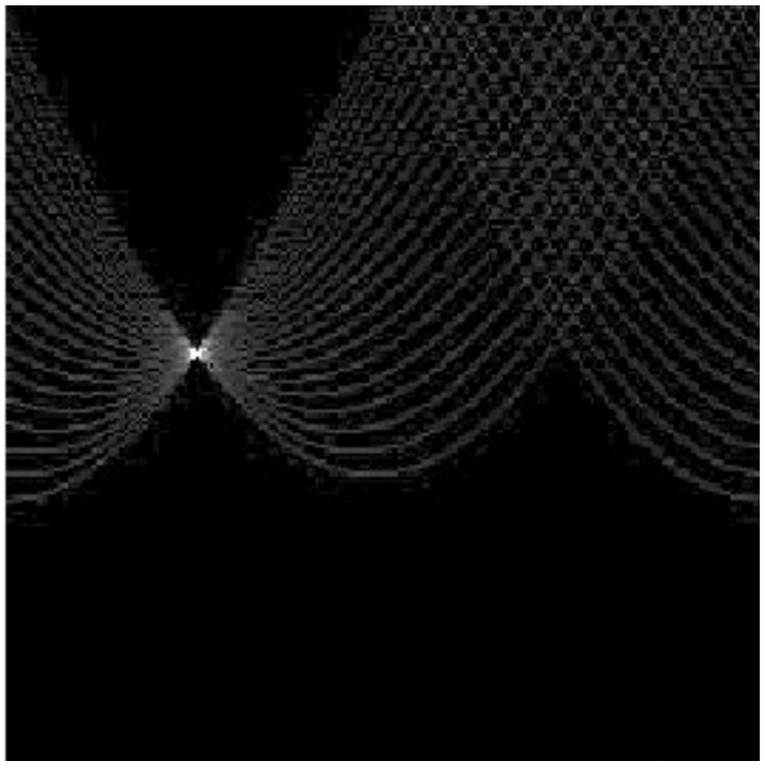




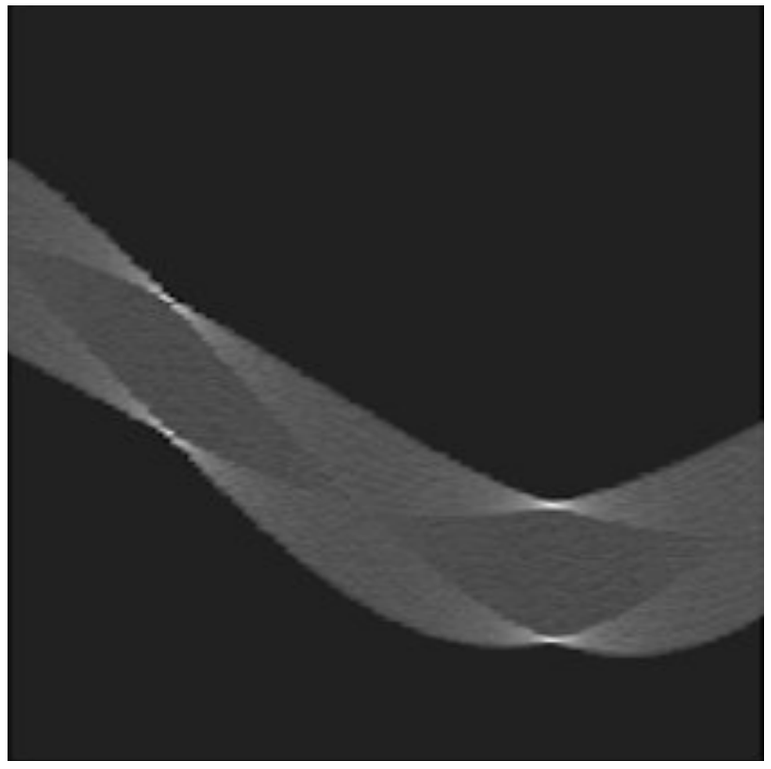
rectangle

Basic shapes

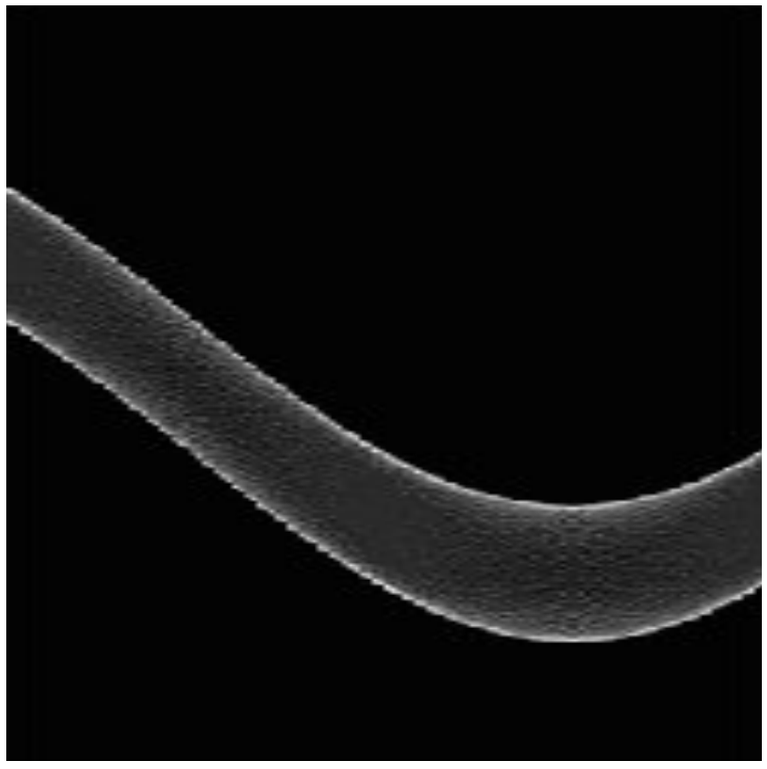
(in parameter space)



line



rectangle



circle

In practice, measurements are noisy…

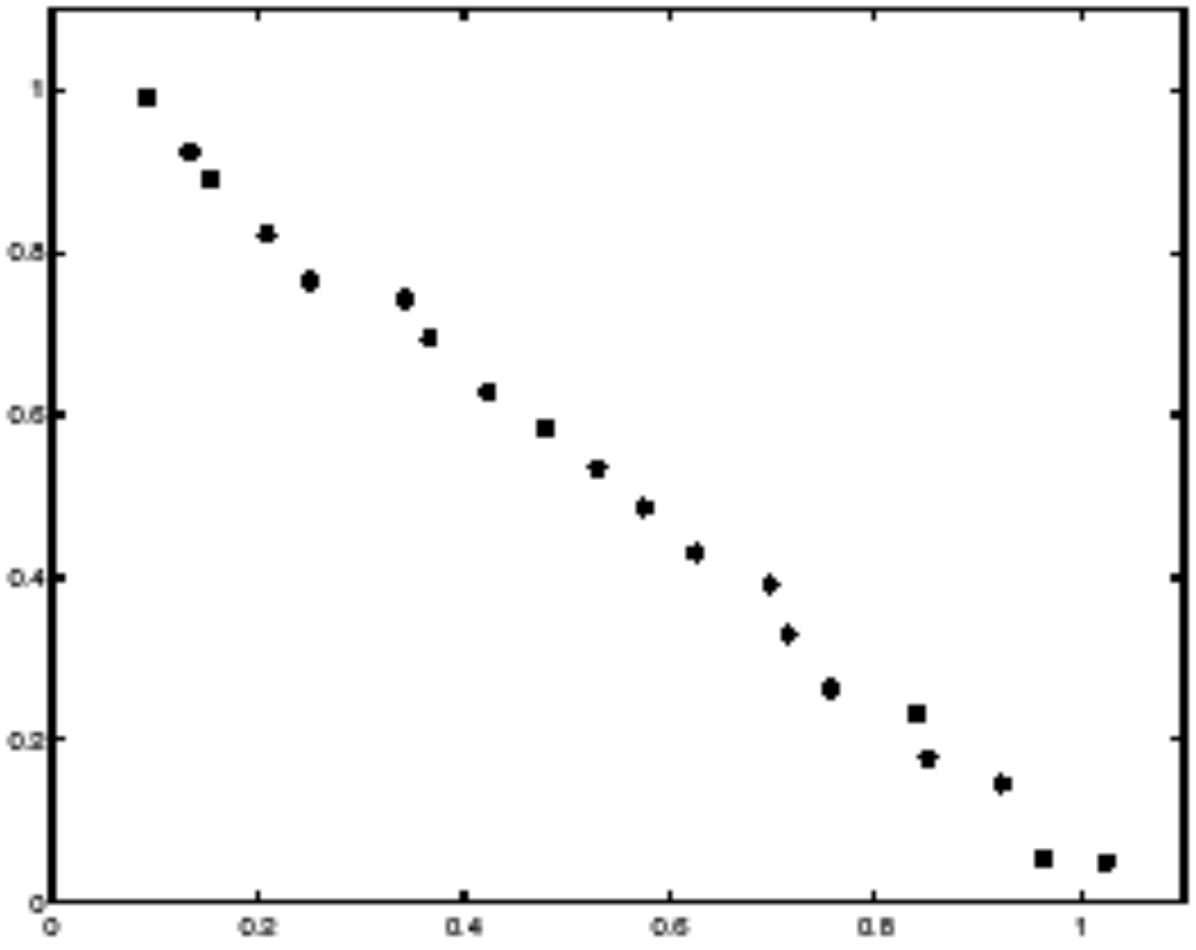
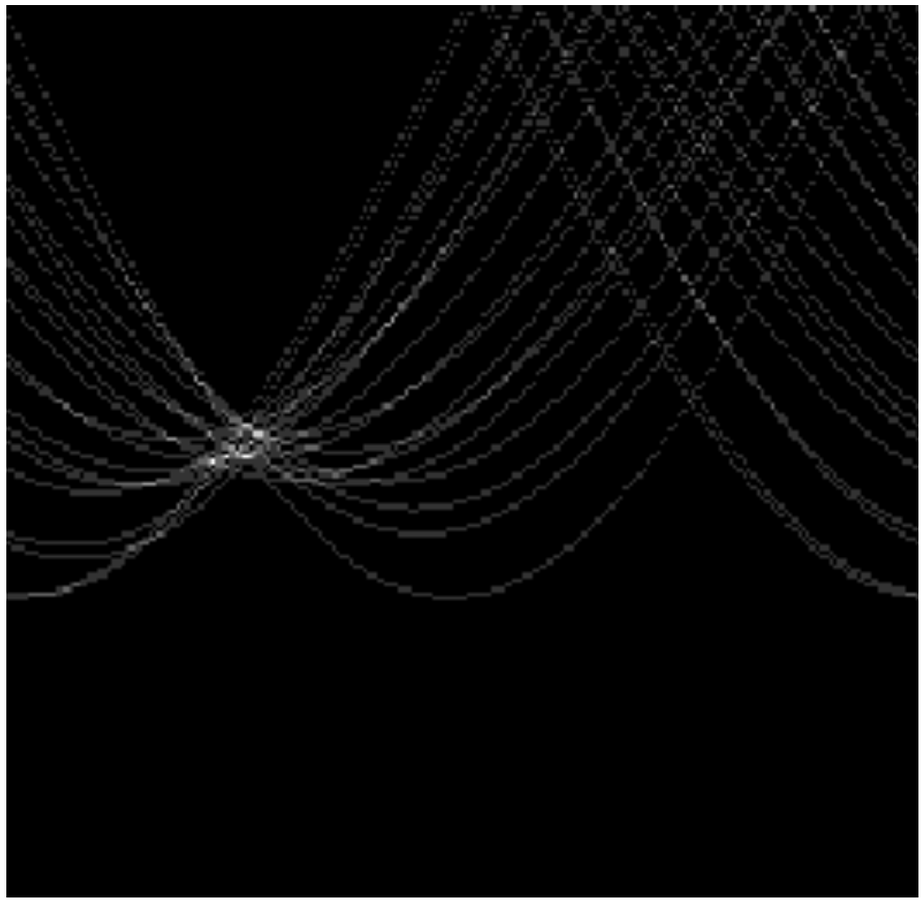


Image space



Votes

Too much noise …

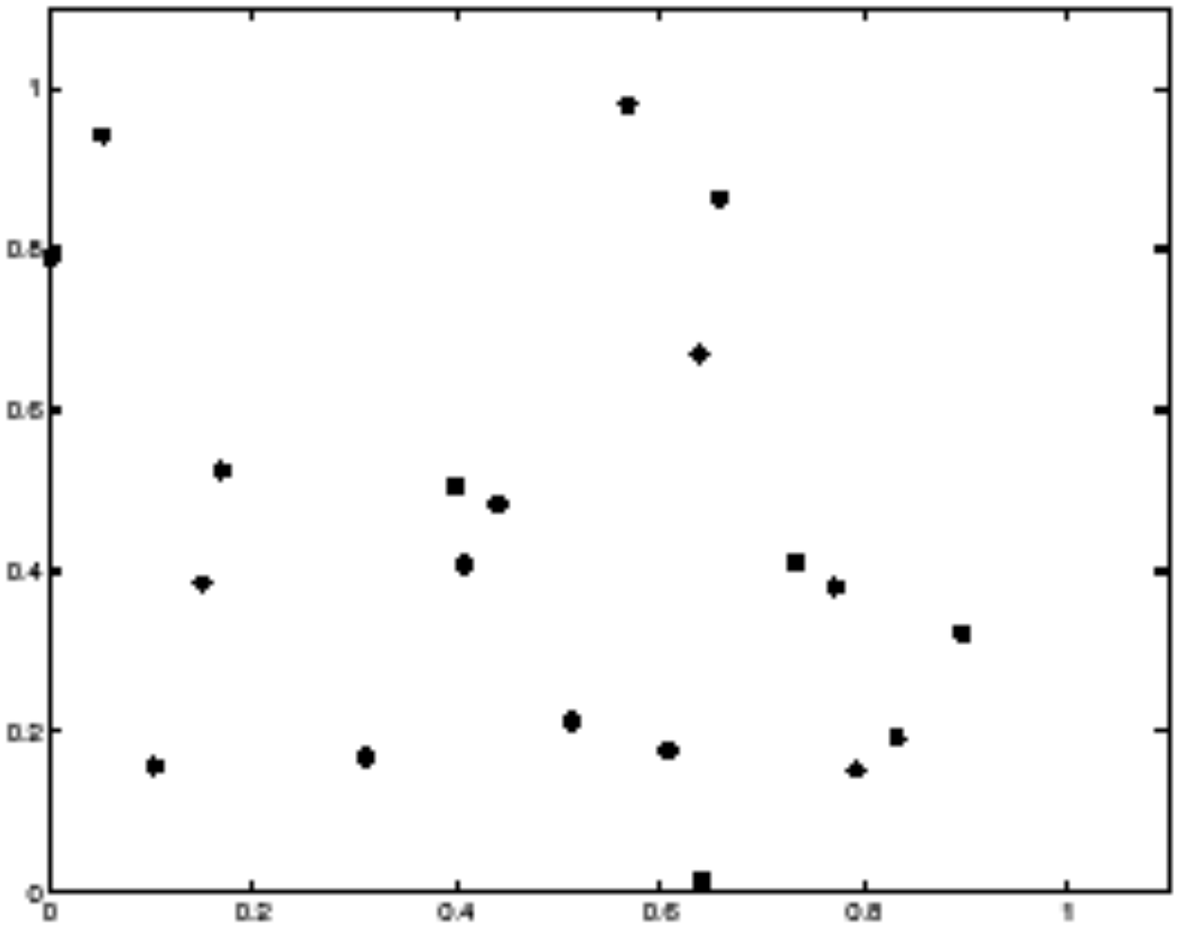
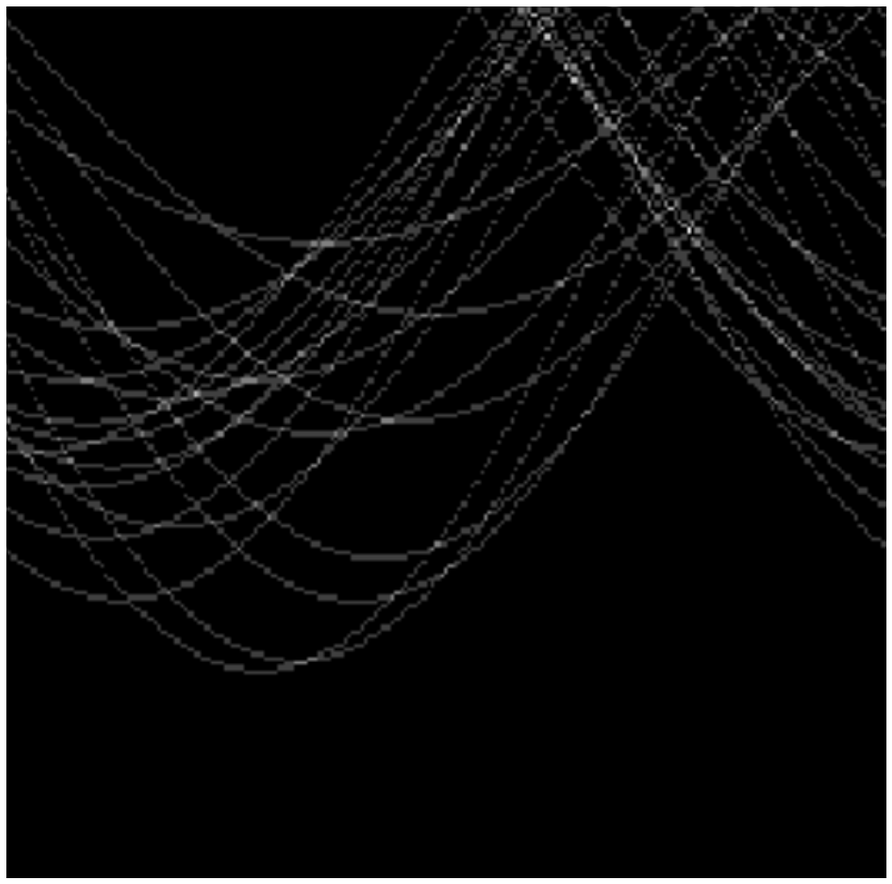
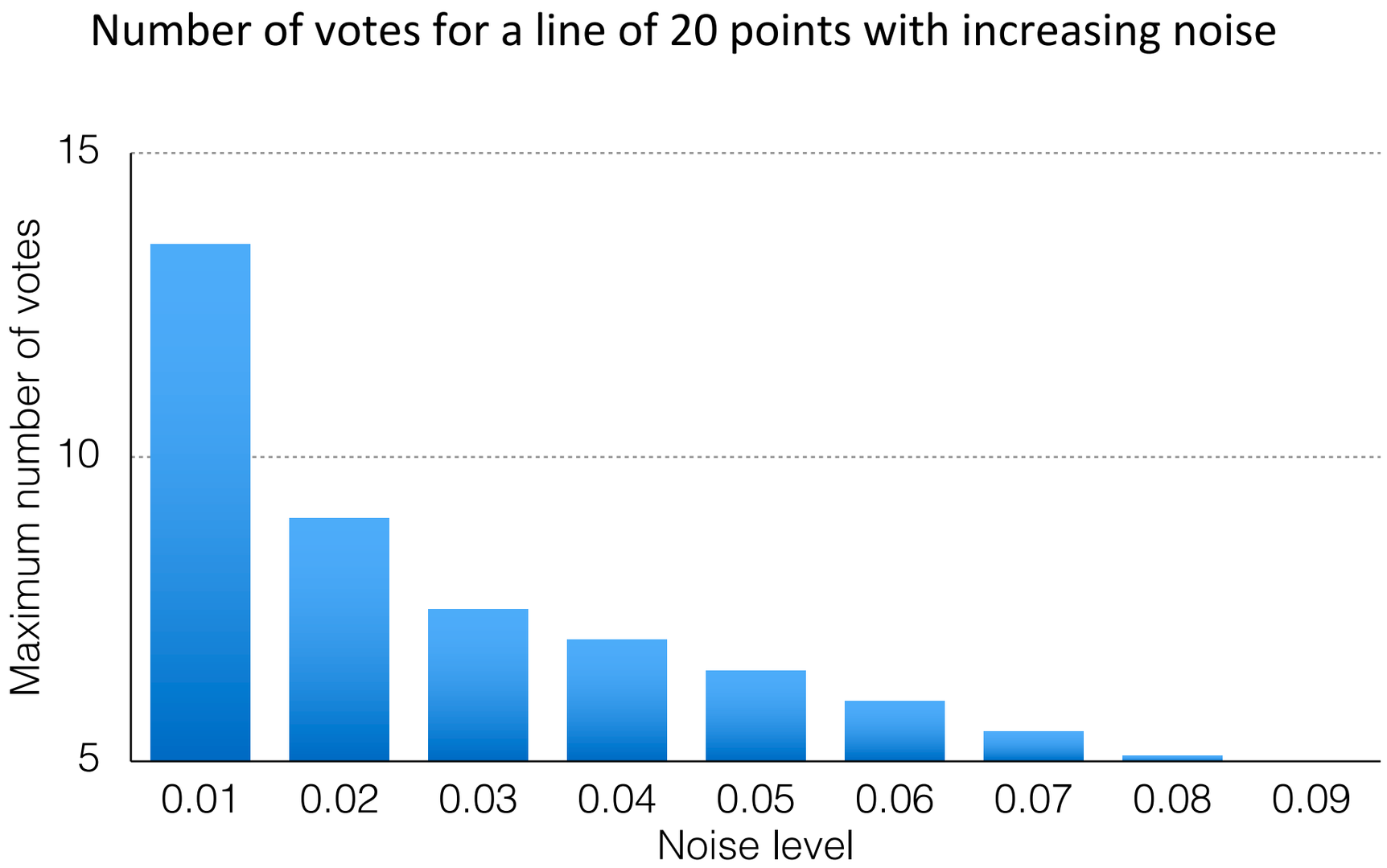


Image space



Votes

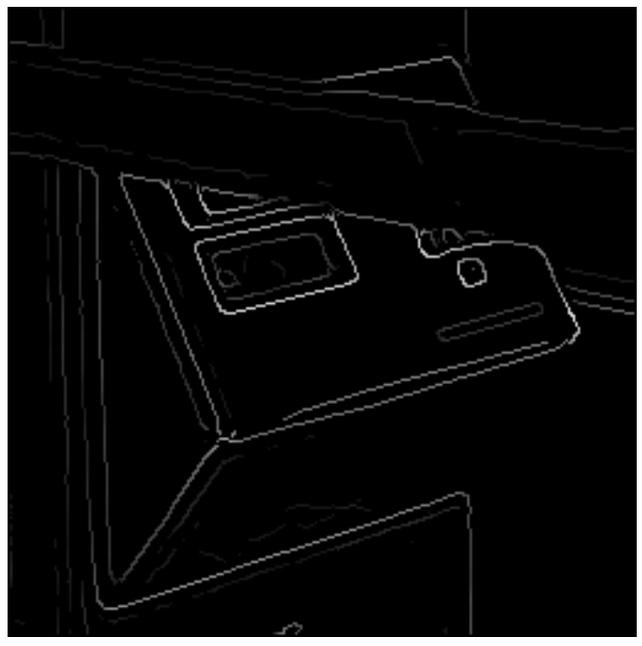
Effects of noise level



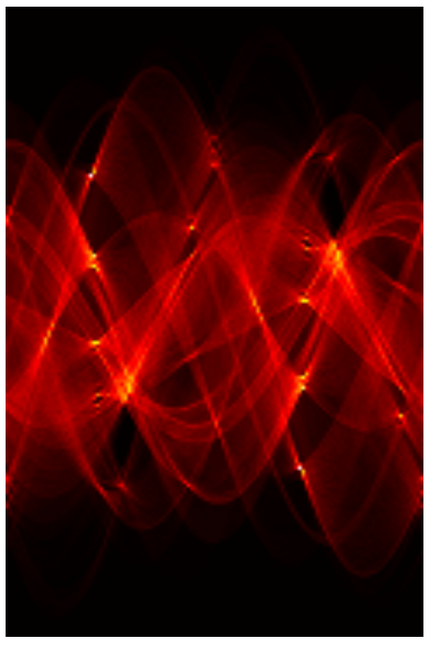
More noise, fewer votes (in the right bin)

Real-world example

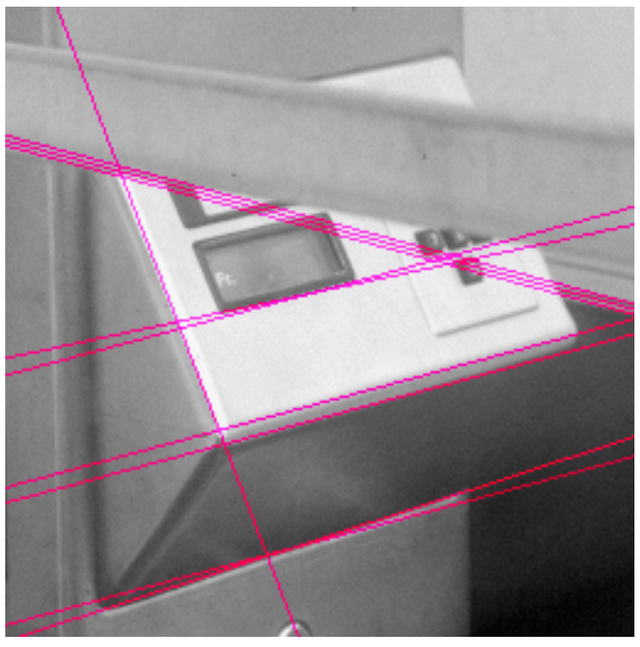
Original



Edges



parameter space



Hough Lines

Hough Circles

Let’s assume radius known

parameters

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

variables

parameters

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

variables

What is the dimension of the parameter space?

parameters

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

variables

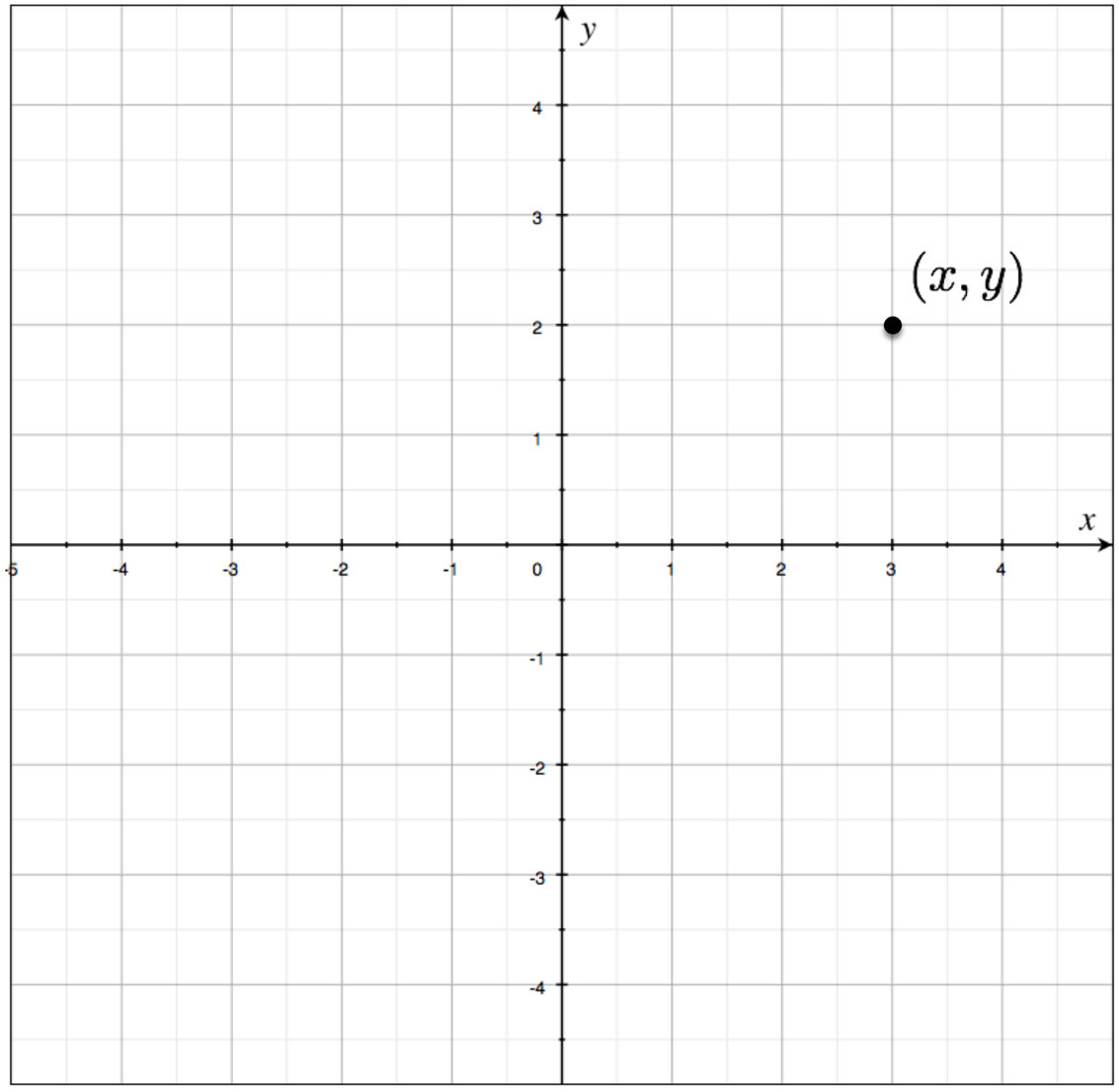
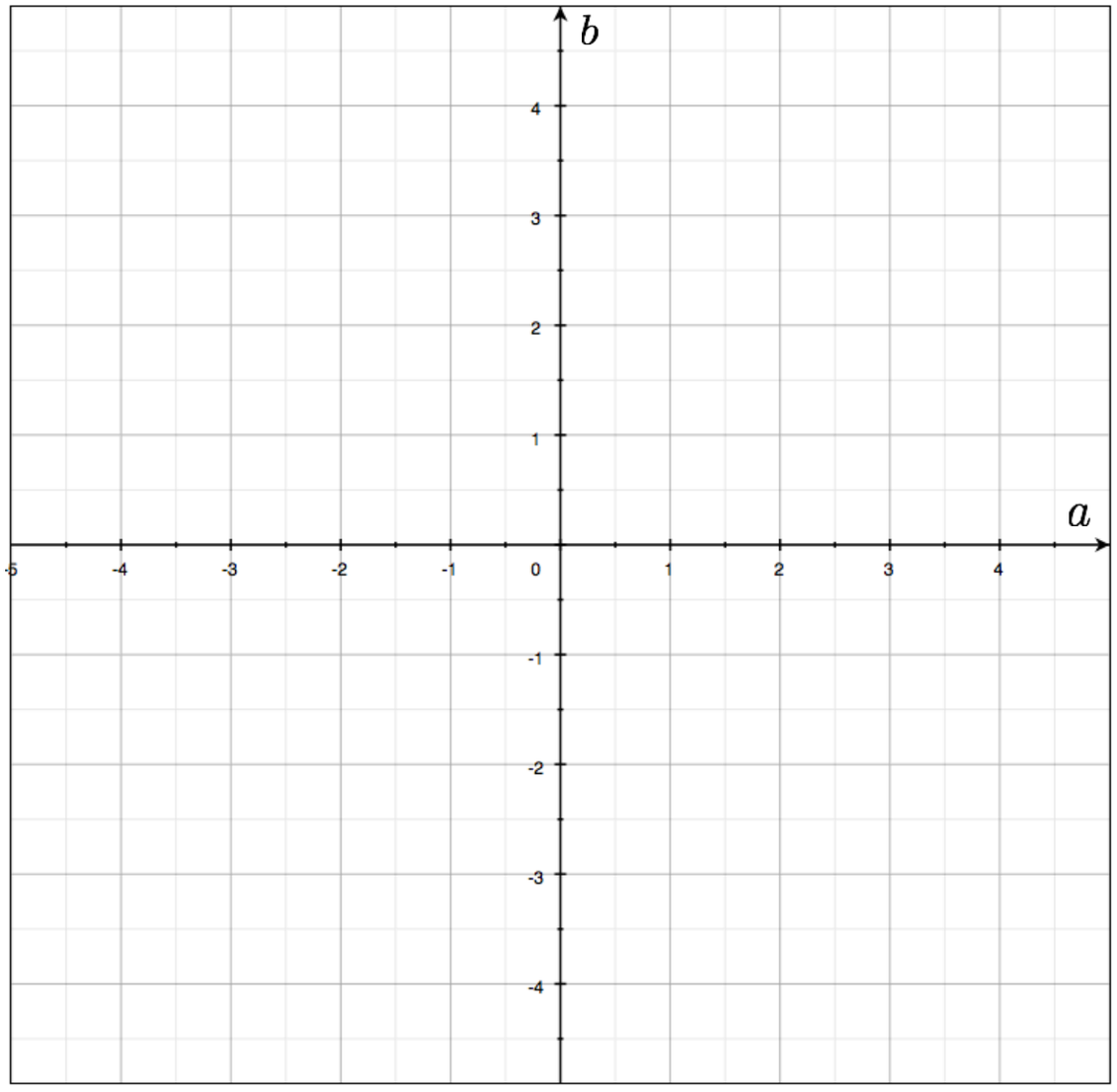


Image space

parameters

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

variables



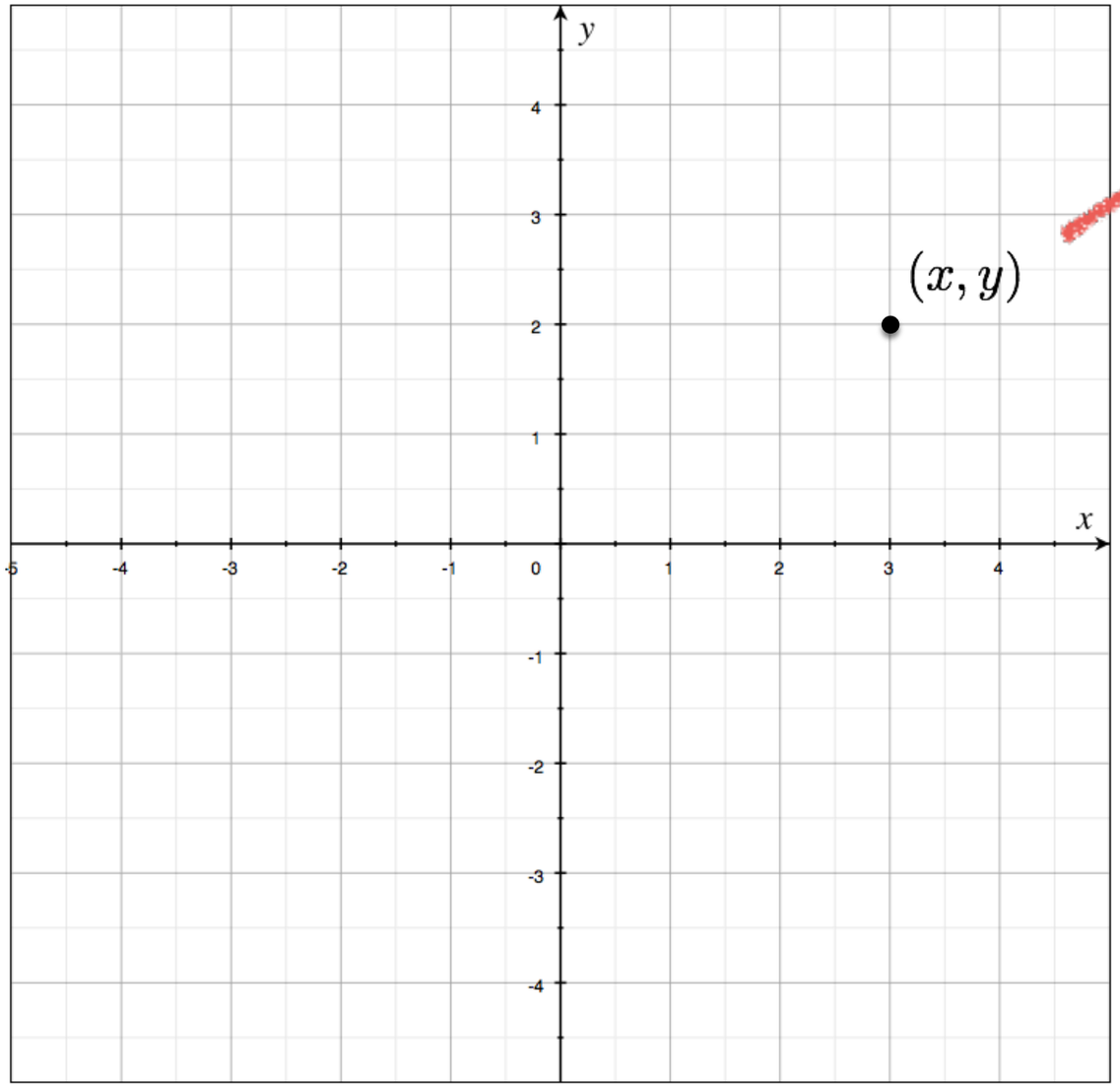
Parameter space

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

parameters

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

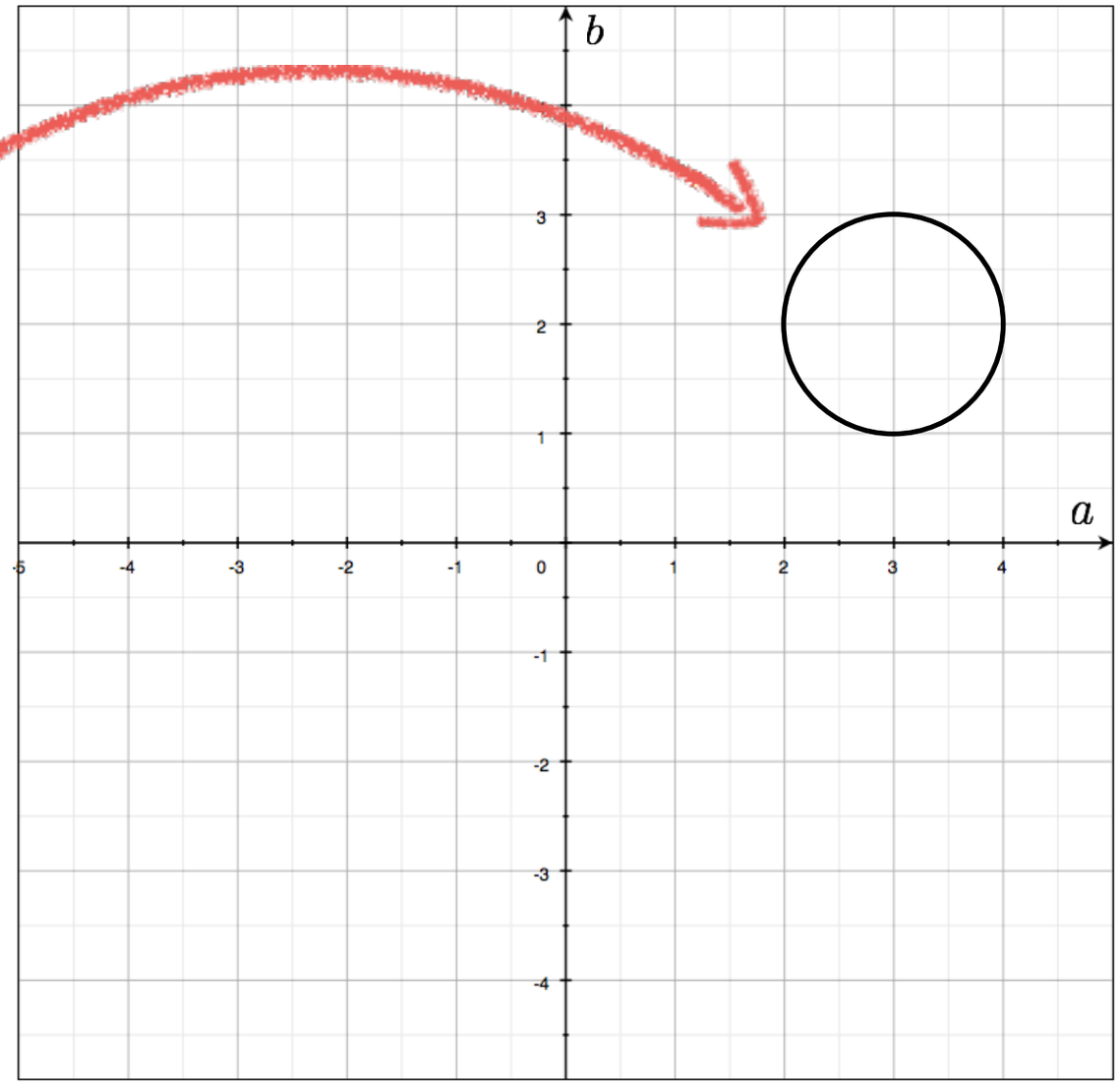
variables



parameters

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

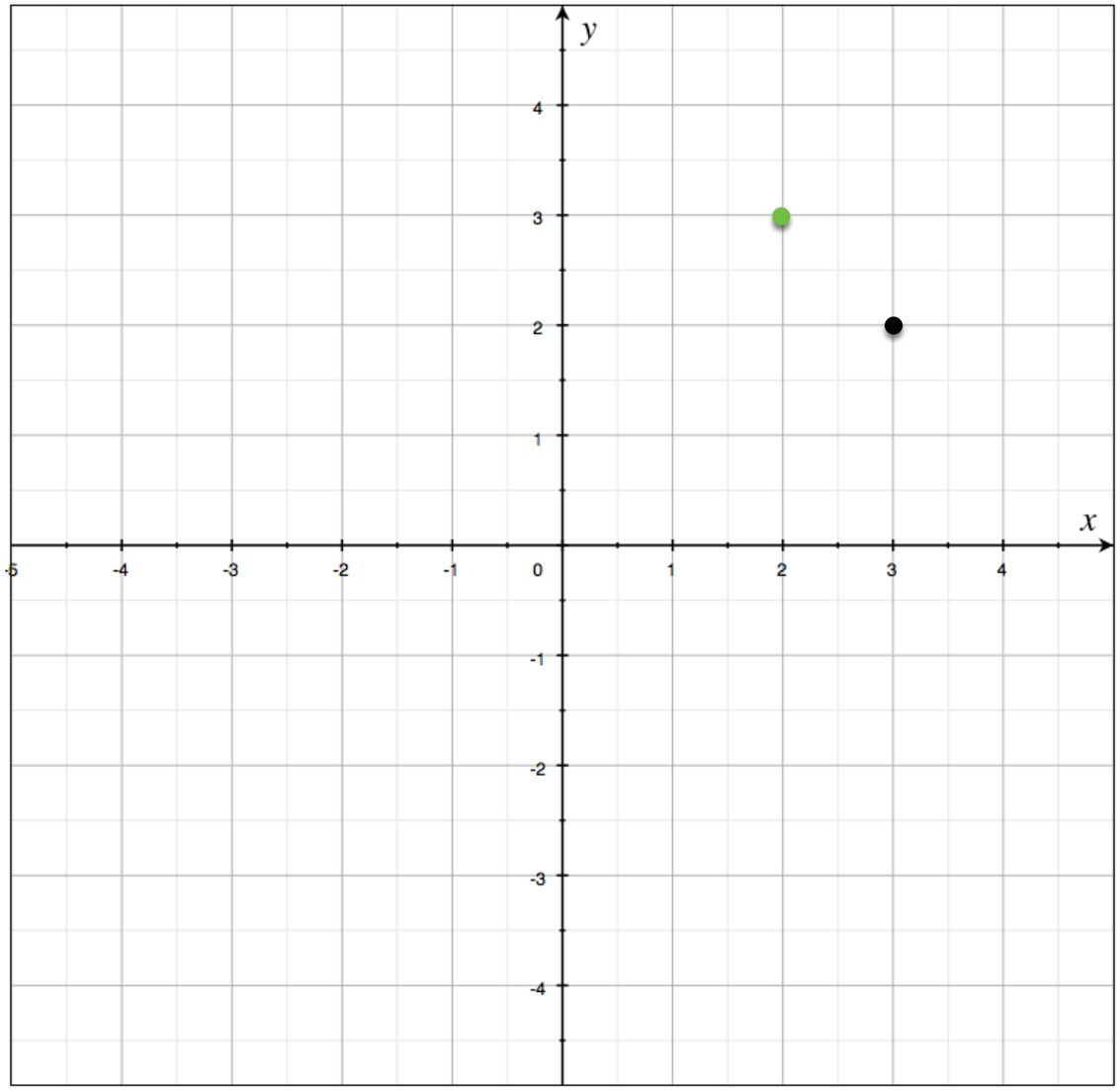
variables



parameters

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

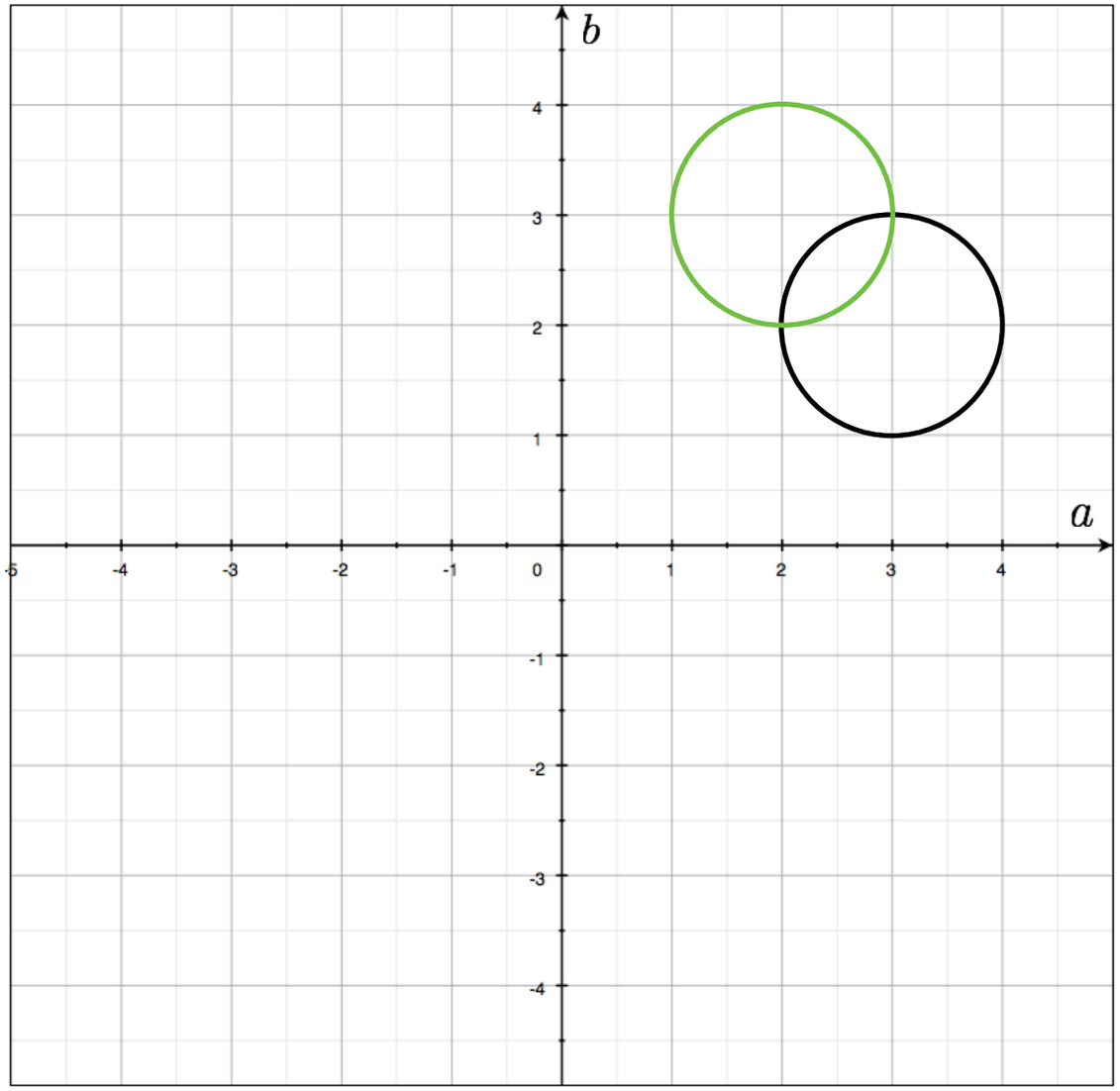
variables



parameters

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

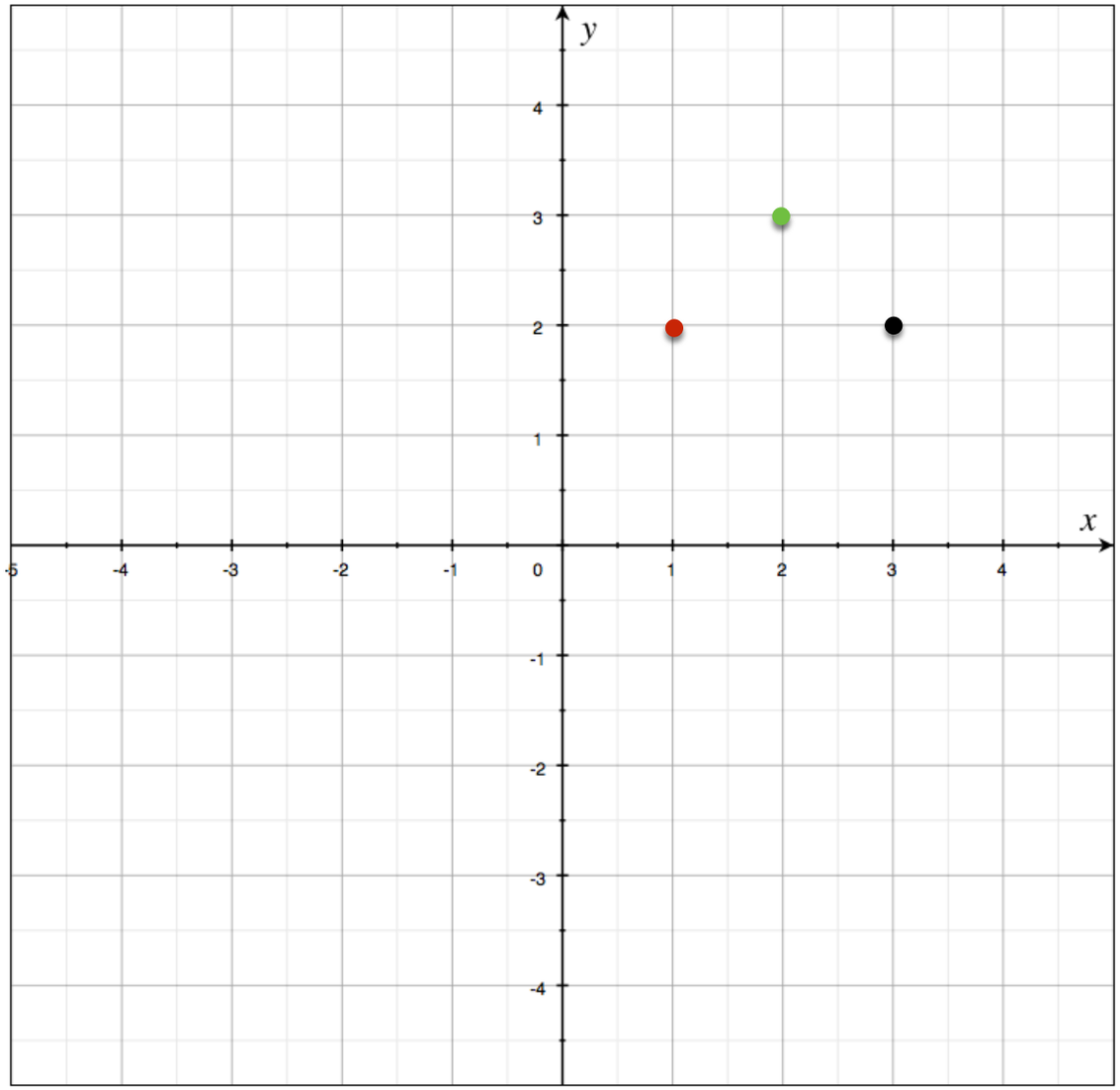
variables



parameters

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

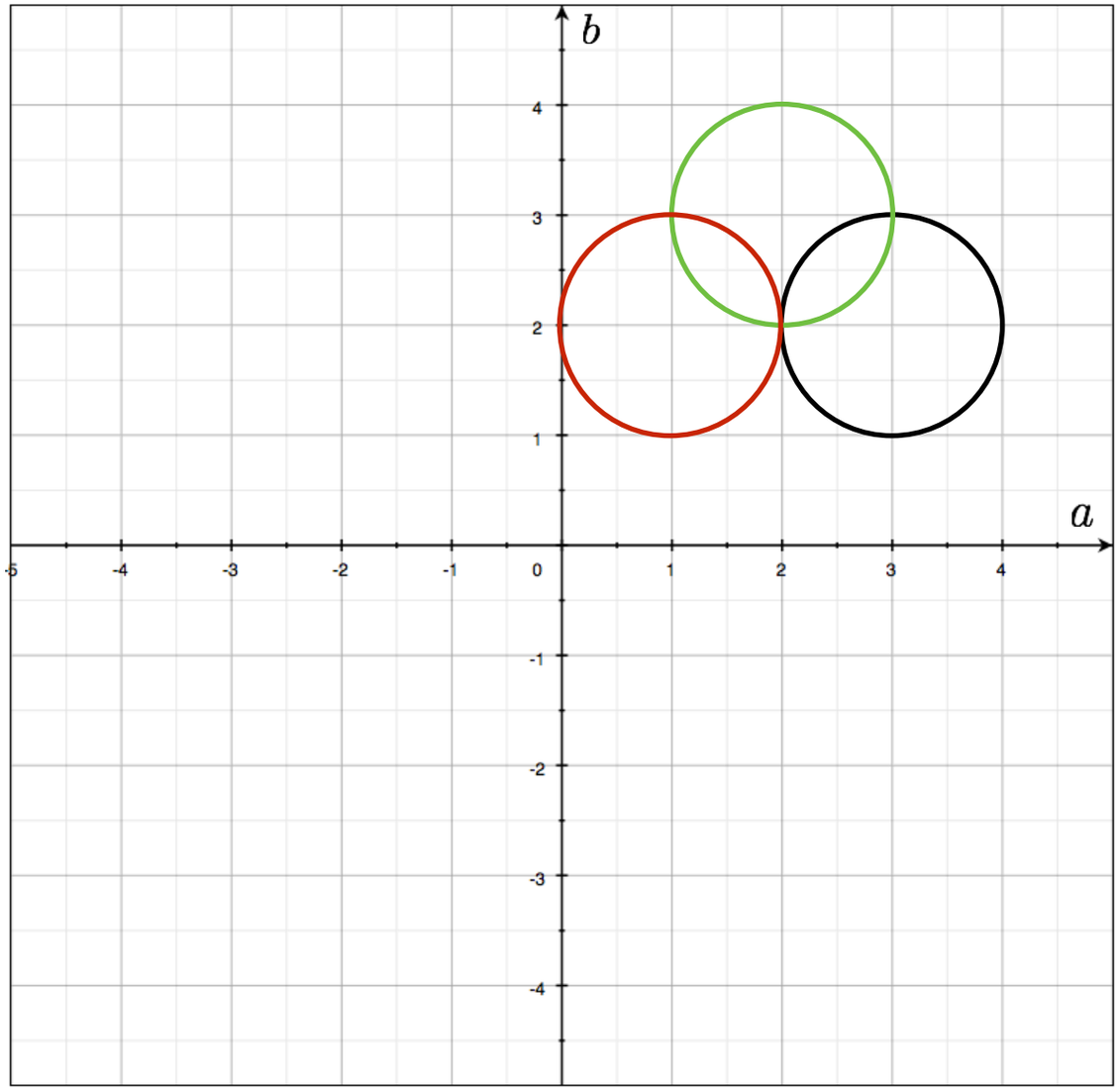
variables



parameters

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

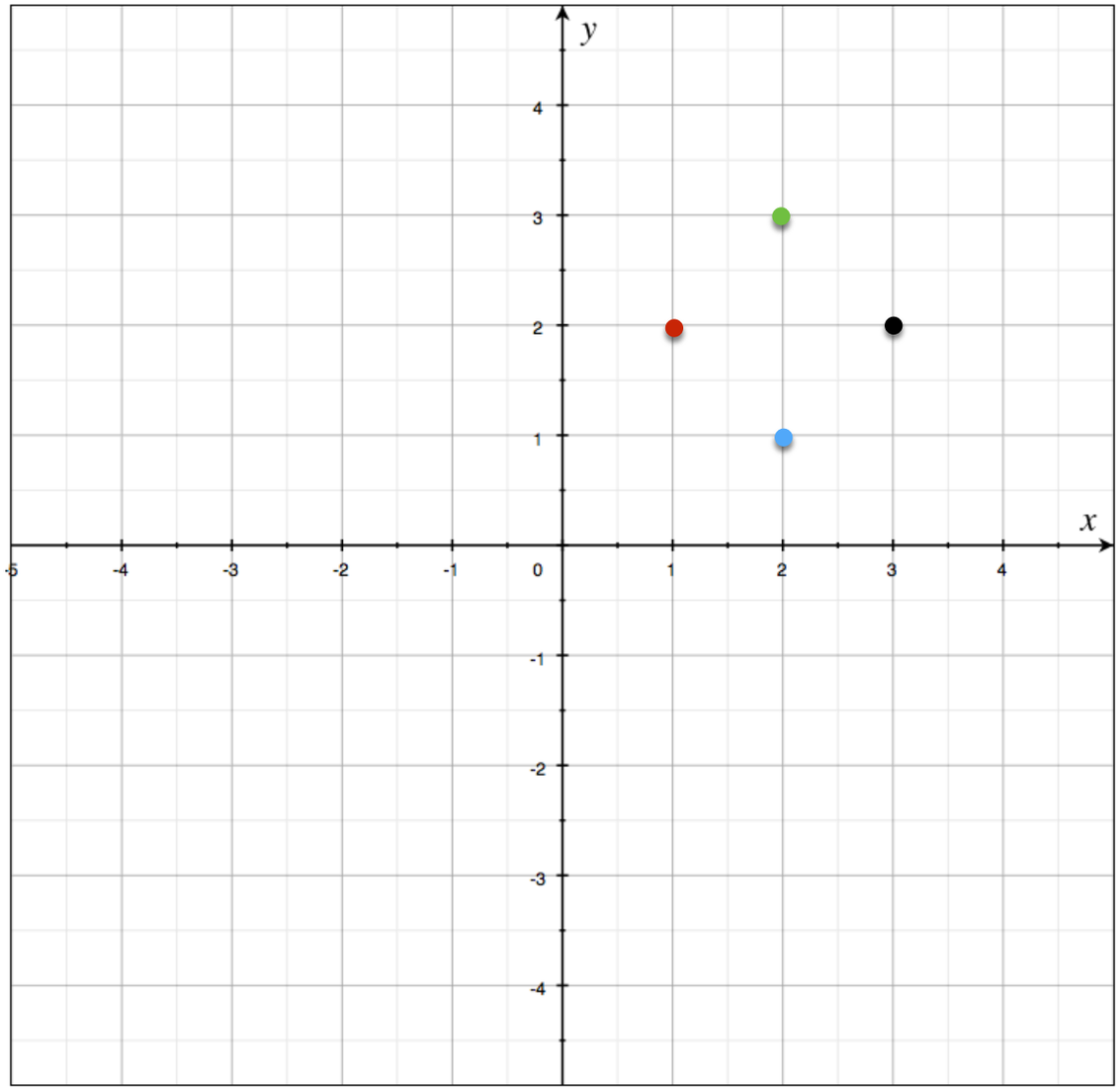
variables



parameters

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

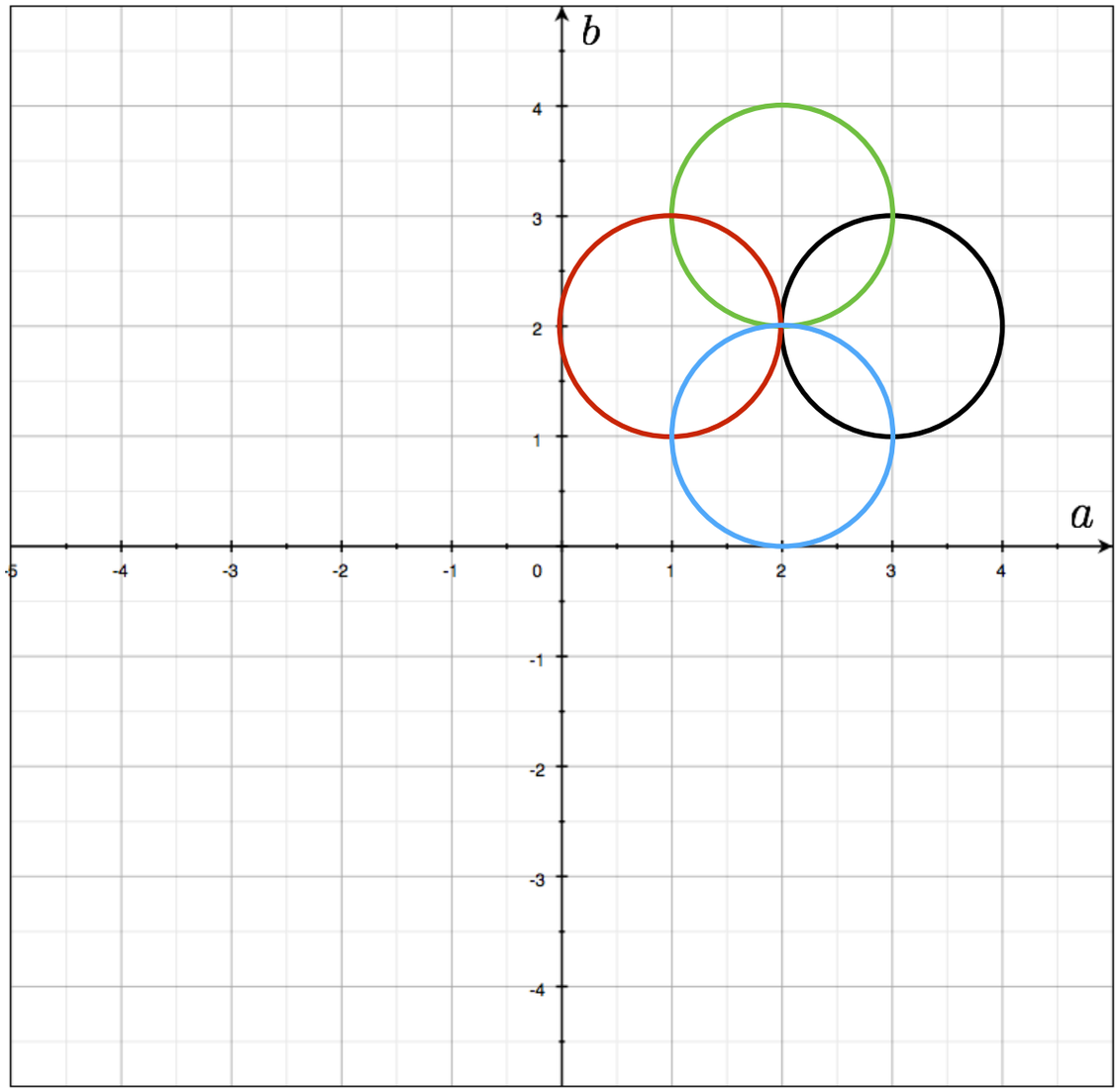
variables



parameters

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

variables



What if radius is unknown?

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?

parameters

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

variables

parameters

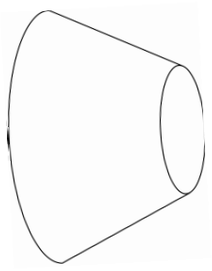
$$(x-a)^2+(y-b)^2=r^2$$

variables

If radius is not known: 3D Hough Space!

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

Surface shape in Hough space is complicated

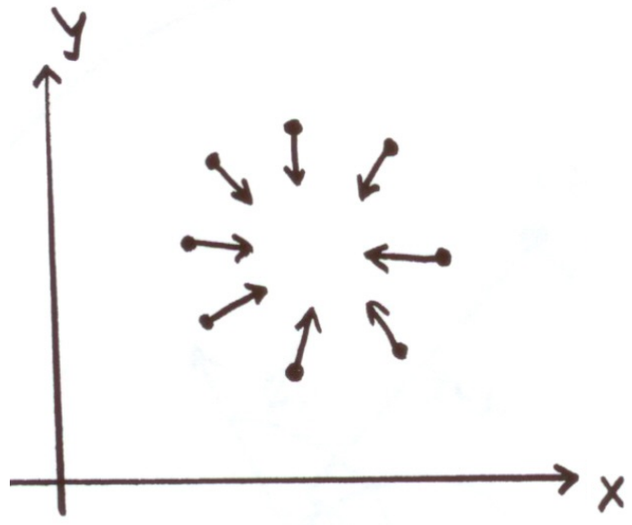


Using Gradient Information

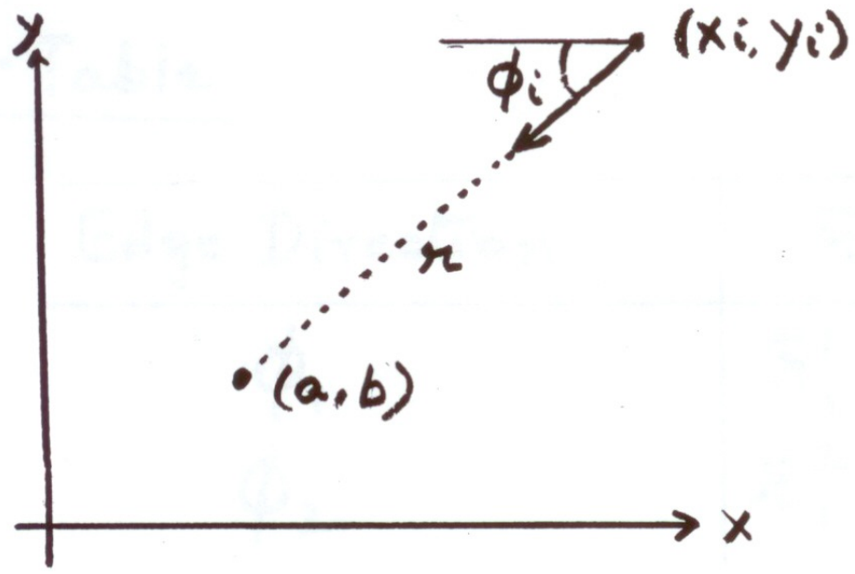
Gradient information can save lot of computation:

Edge Location (x_i, y_i)

Edge Direction ϕ_i



Assume radius is known:



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

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

Need to increment only one point in accumulator!

parameters

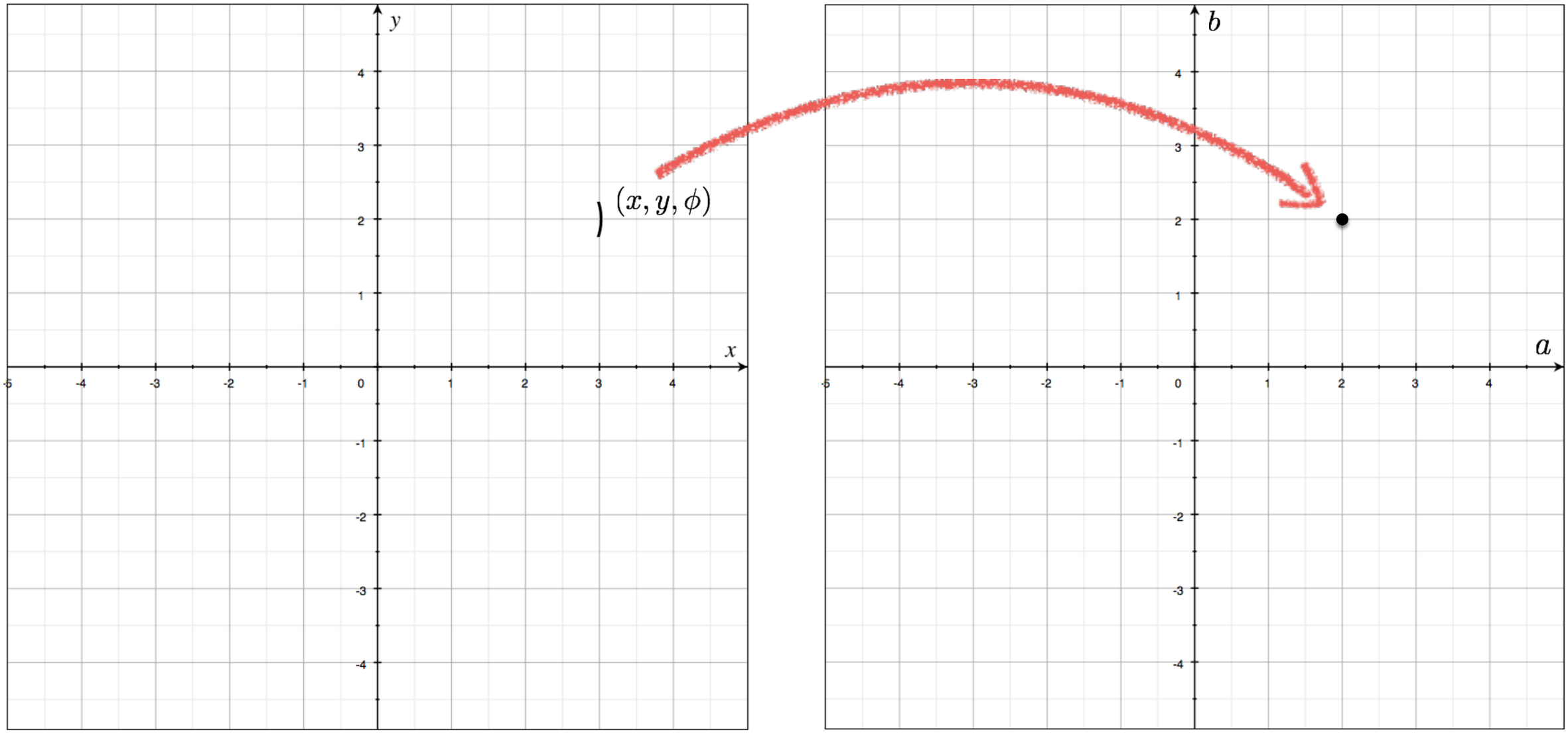
$$(x-a)^2+(y-b)^2=r^2$$

variables

parameters

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

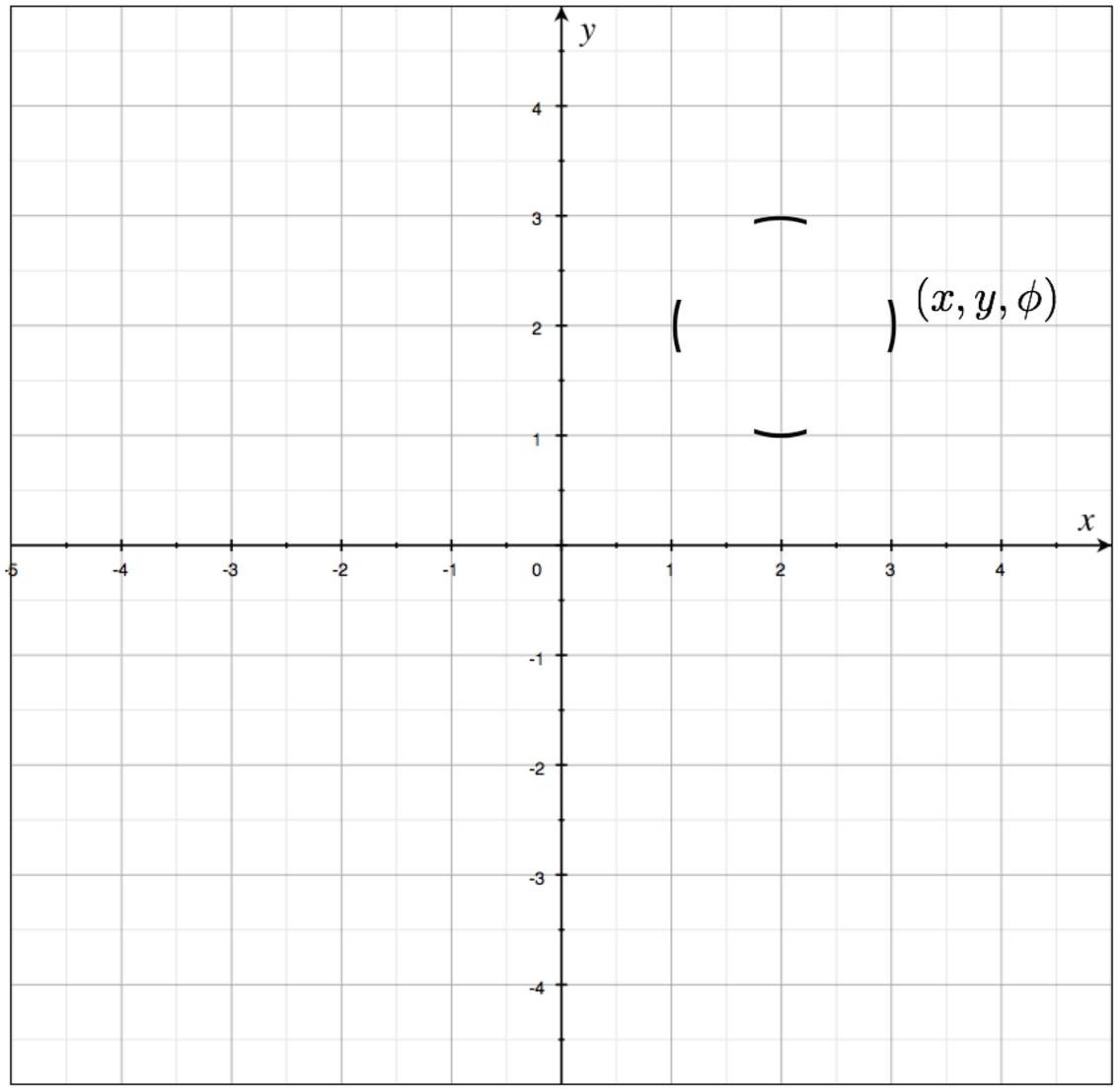
variables



parameters

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

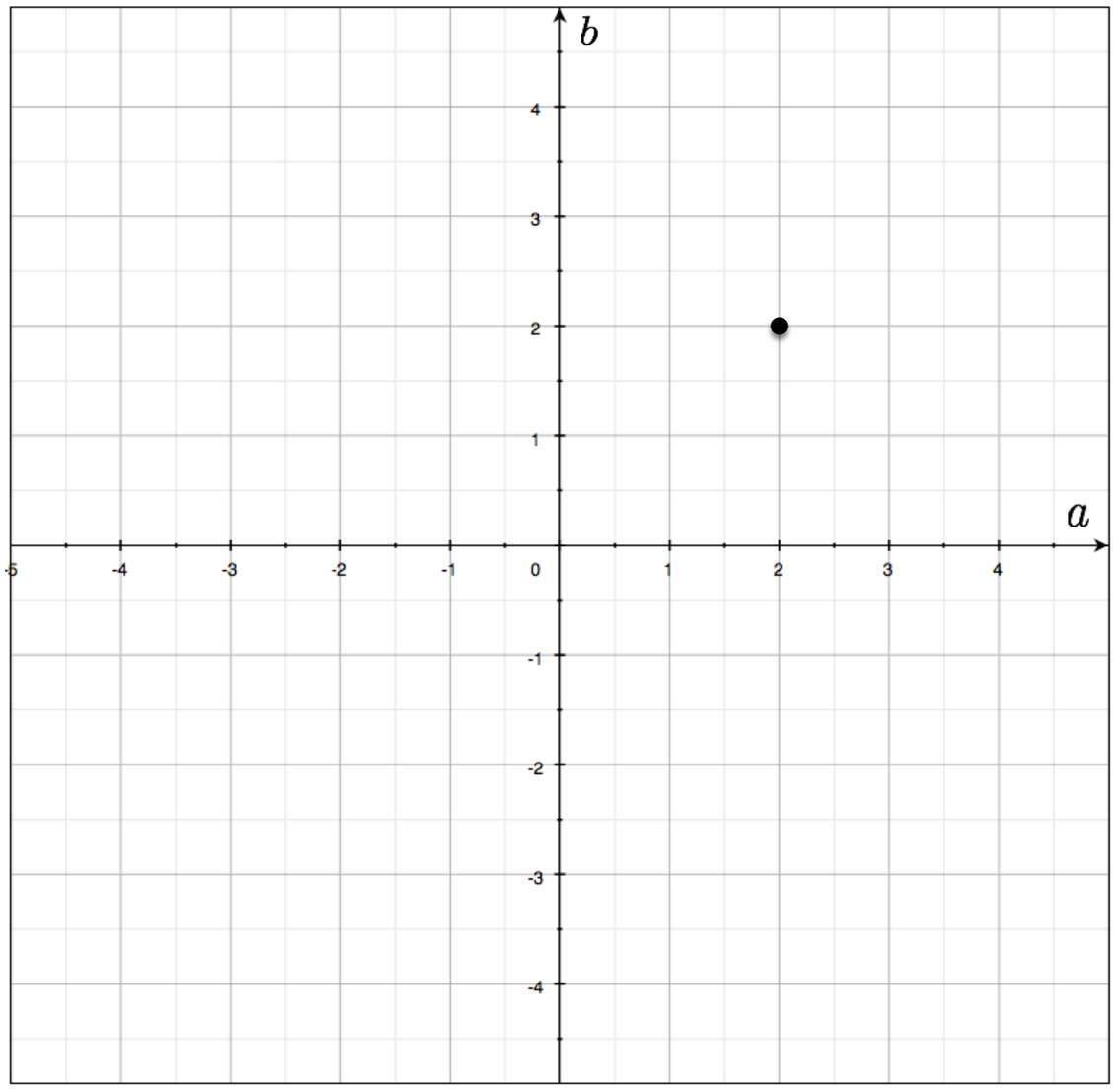
variables

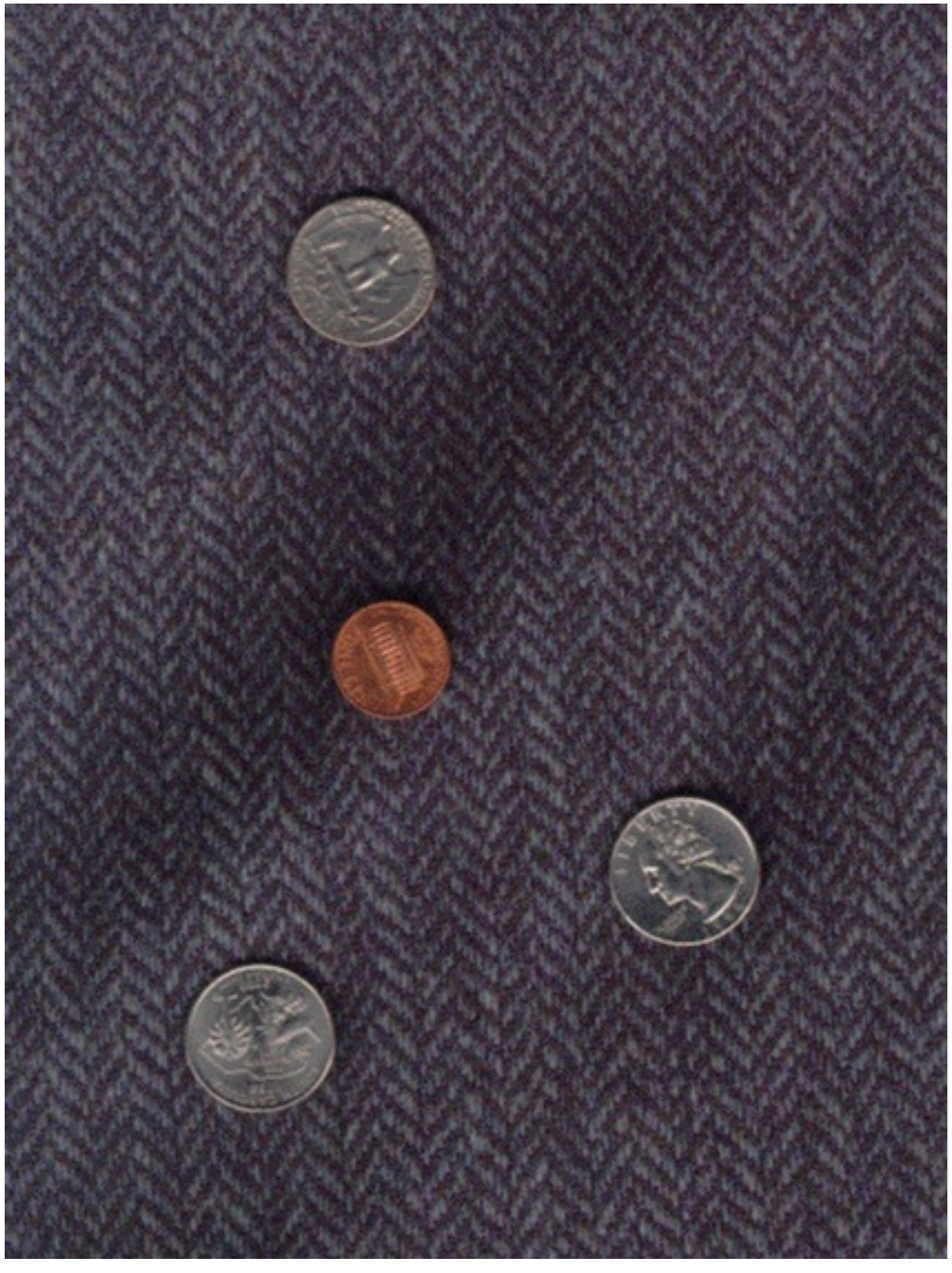


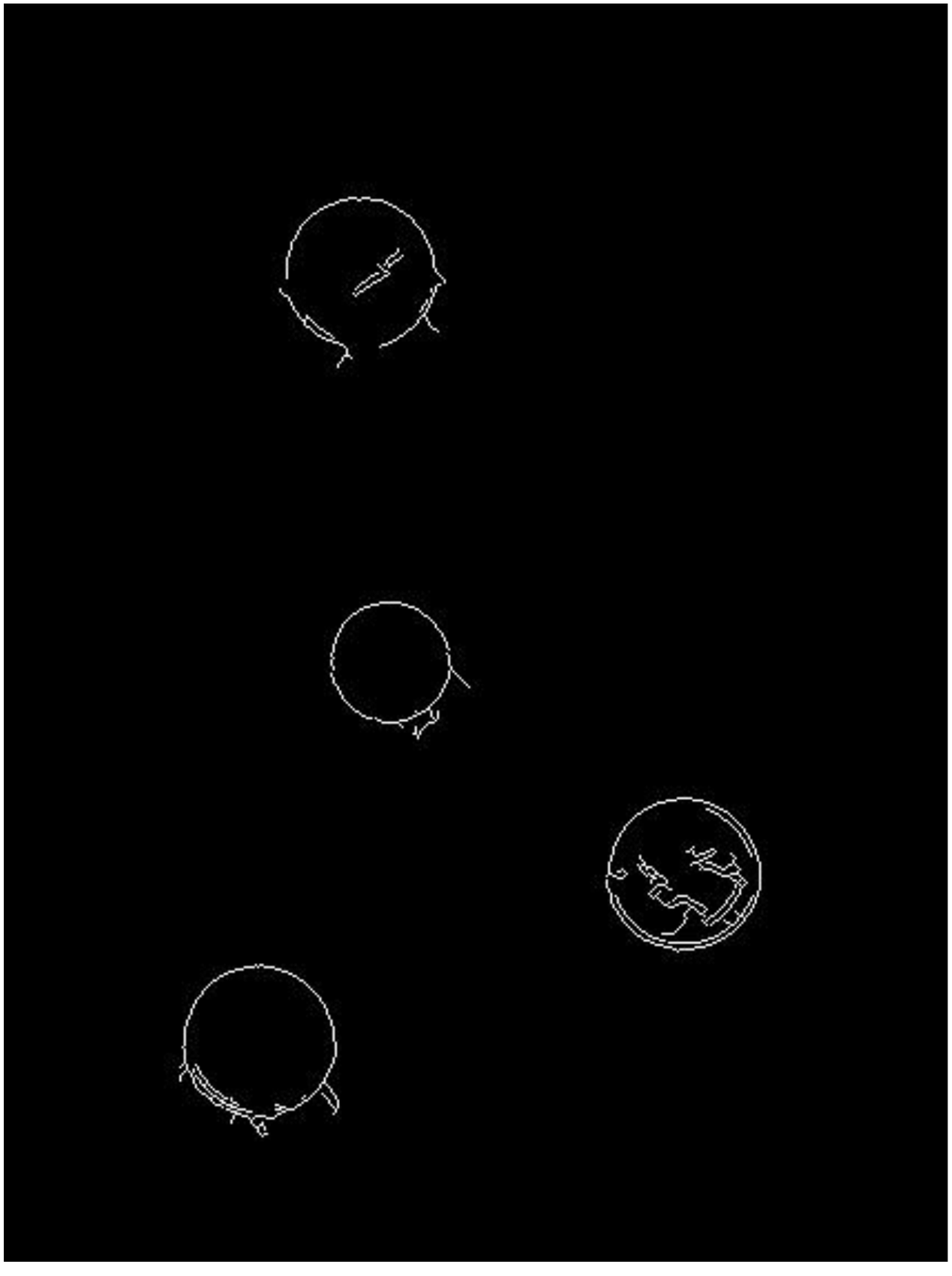
parameters

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

variables



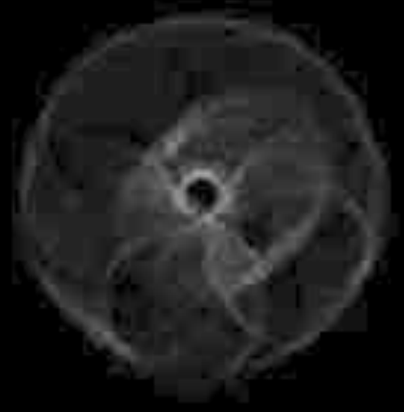


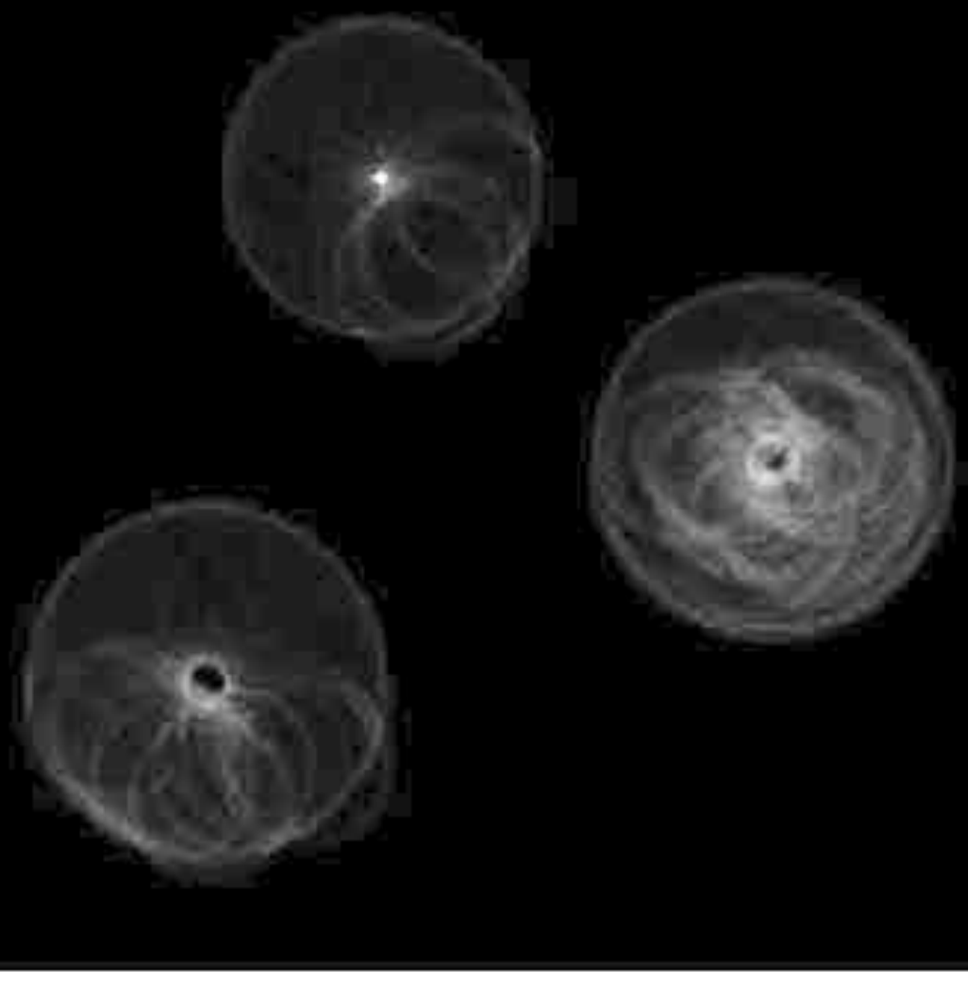


Pennie Hough detector

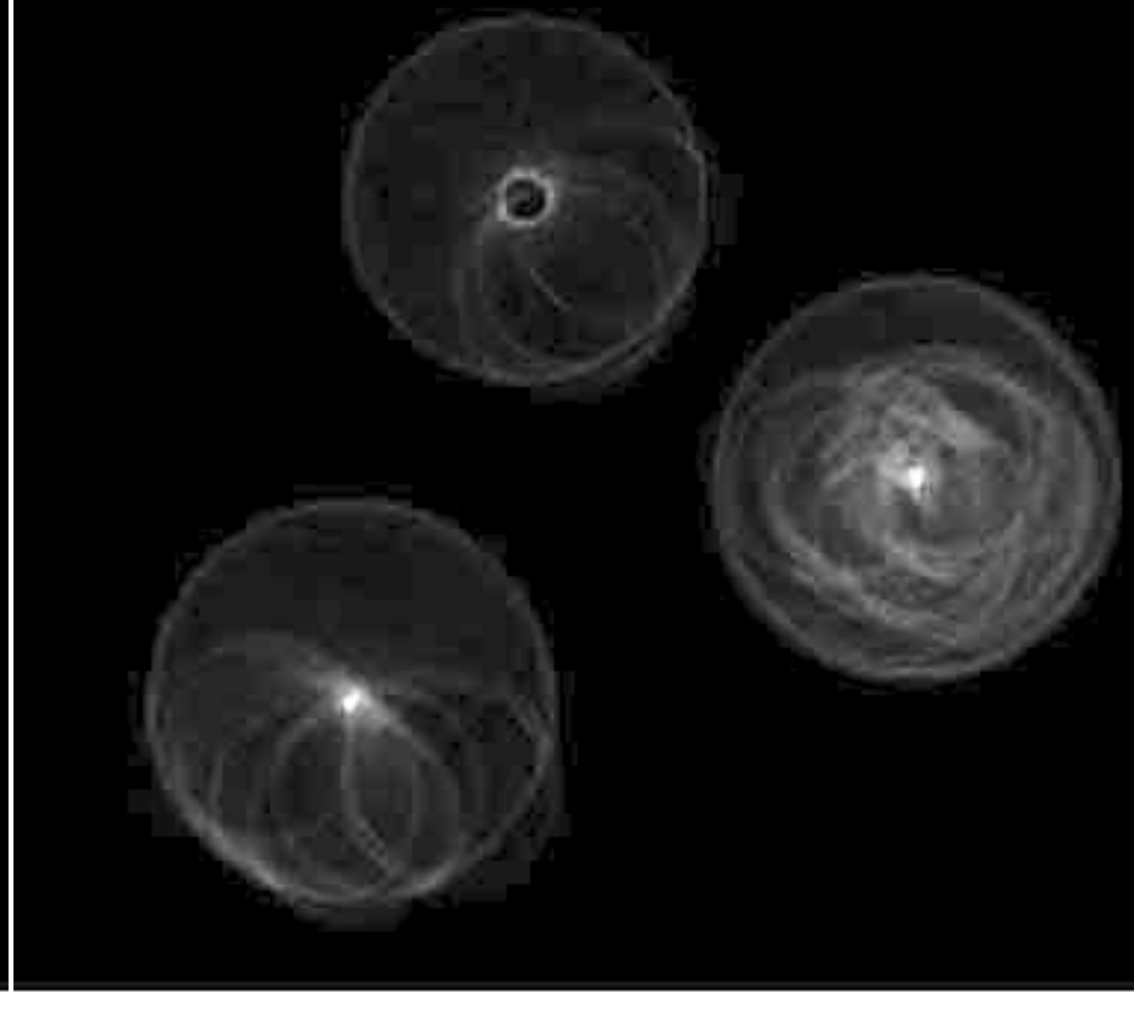
Quarter Hough detector

Pennie Hough detector





Quarter Hough detector



The Hough transform …

Deals with occlusion well?

Detects multiple instances?

Robust to noise?

Good computational complexity?

Easy to set parameters?