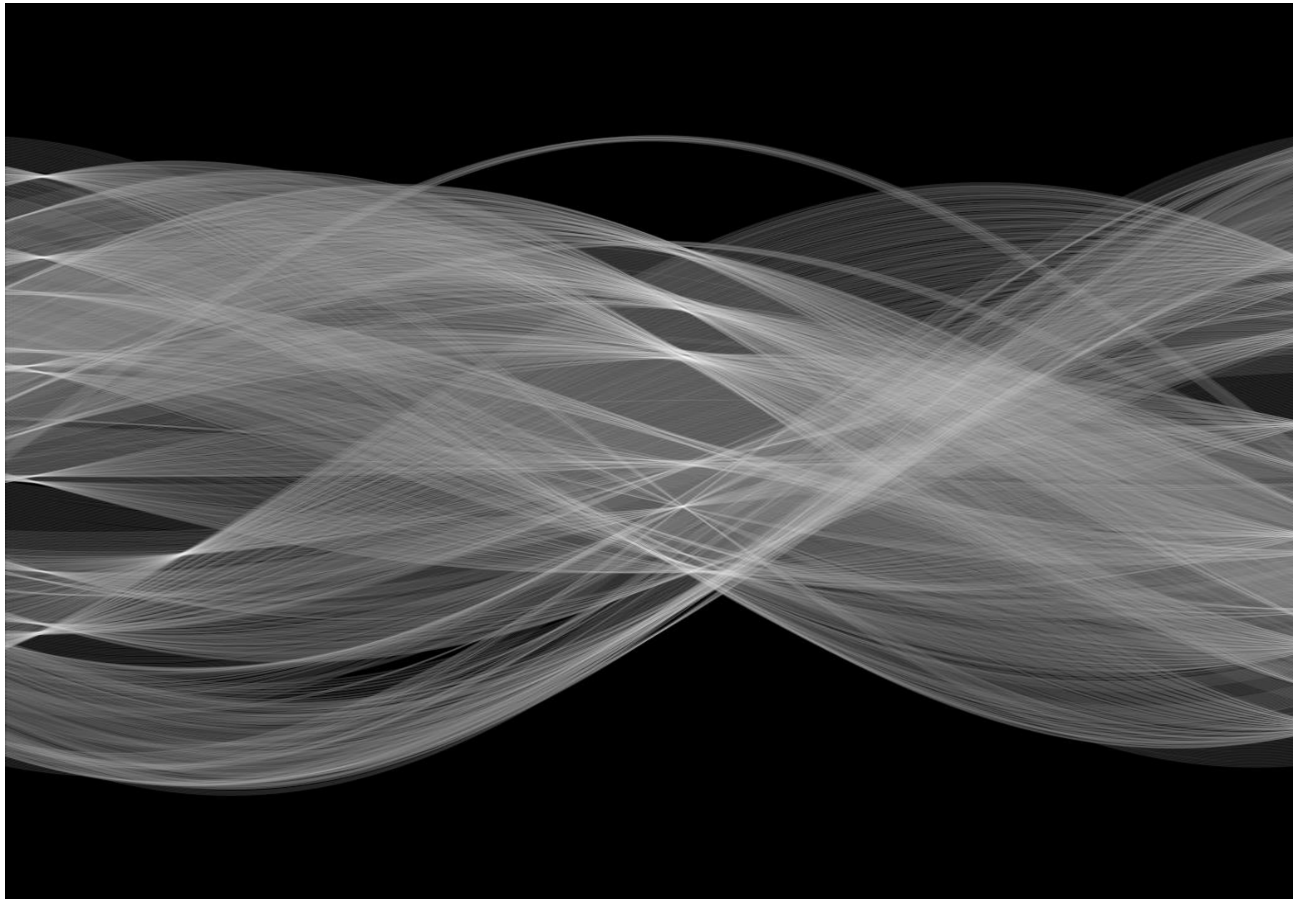
Hough transform

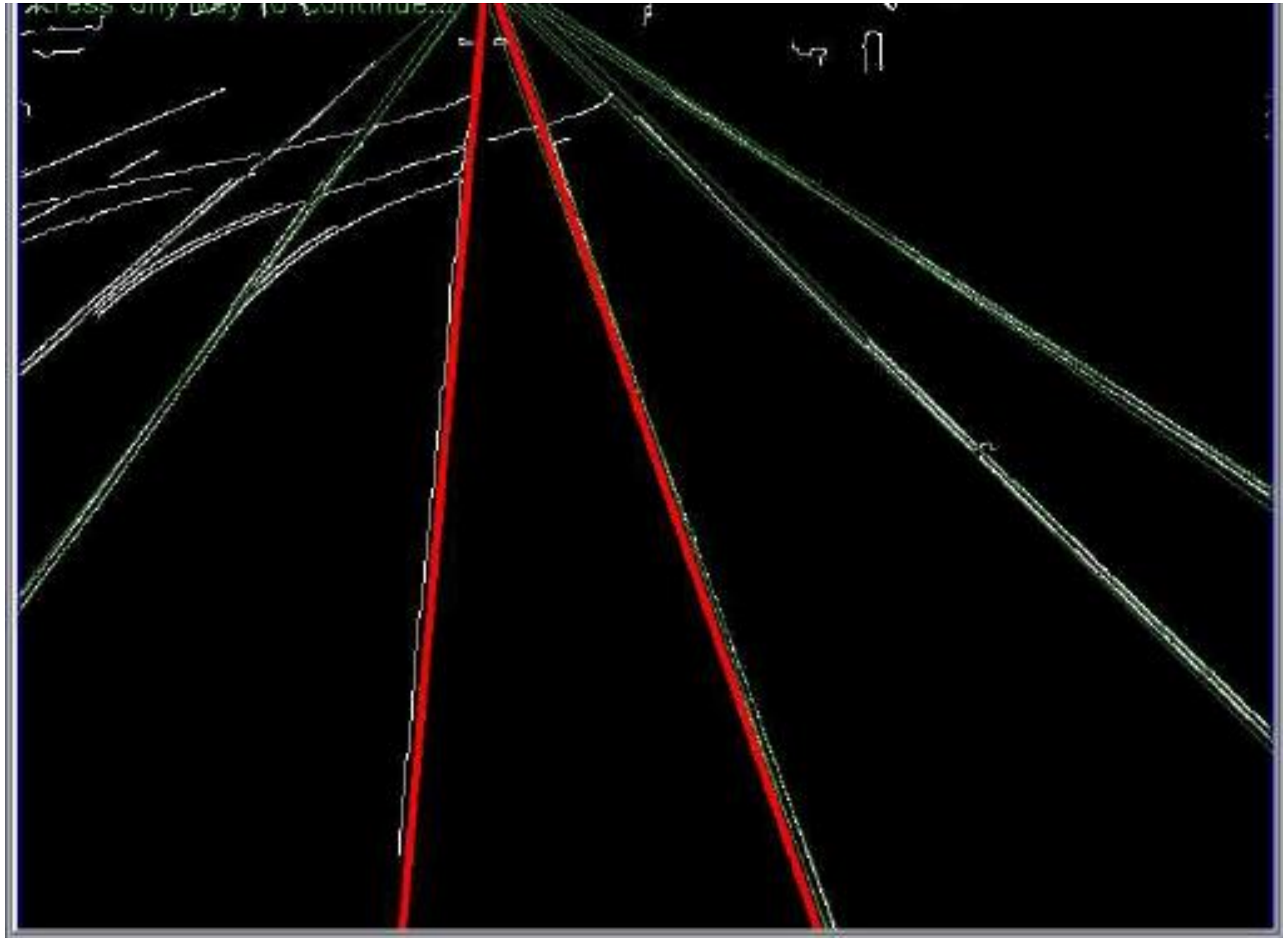


Yoni Chechik

www.AlisMath.com

Problem 3: fitting multiple lines

- (This class is a follow-up of the last curve fitting class).
- What if the initial data has multiple lines needed to detect. What can we do?



TOC

- Hough transform
  - $(\boldsymbol{m}, \boldsymbol{b})$ **parameter space**
  - (ρ, θ) parameter space

$(\boldsymbol{m}, \boldsymbol{b})$ Parameter space

- Line function: $y = mx + b$
  - Usually we are given (m, b) constants, and the variables are (x, y).
- In a regression problem we are given (x, y), and the unknowns we wish to find are the best fit for (m, b).
  - **Let's look at $(\boldsymbol{m}, \boldsymbol{b})$ as our variables.**

(m, b) Parameter space

variables

$$y = mx + b$$

parameters

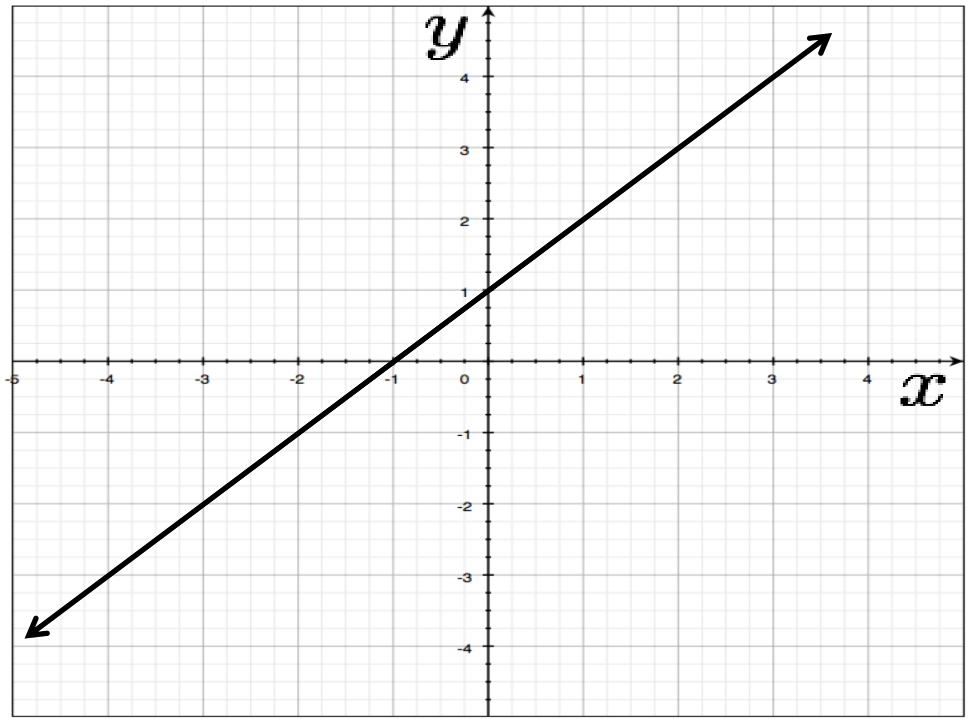


Image space

Let's look at (m,b) as our variables

Plots are from: http://www.cs.cmu.edu/~16385/

(m, b) Parameter space

variables

$$y = mx + b$$

parameters

variables

$$y - mx = b$$

parameters

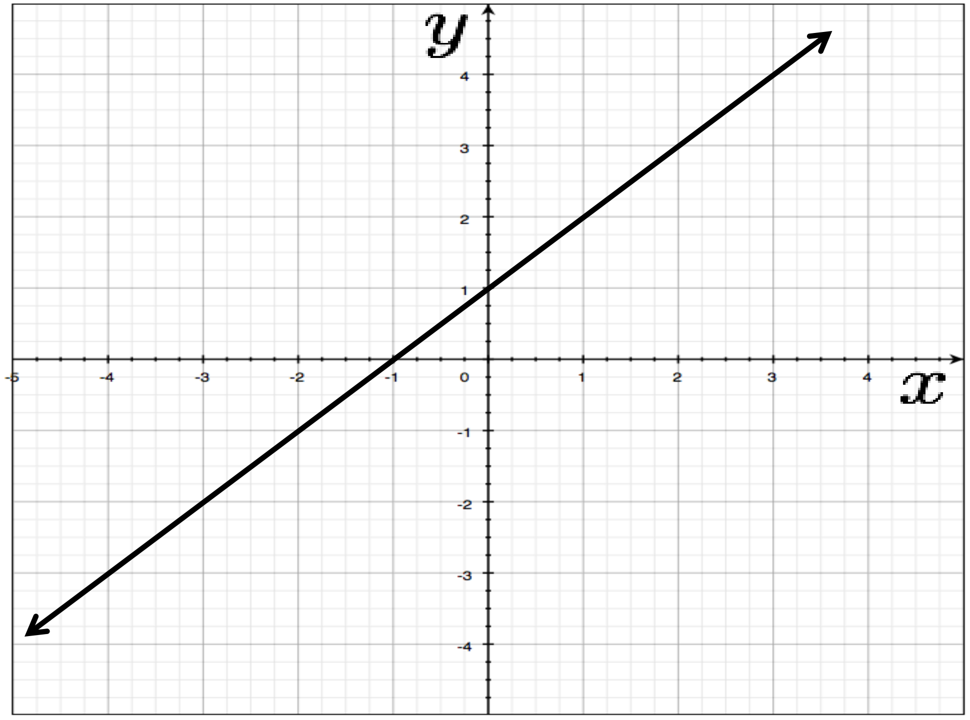
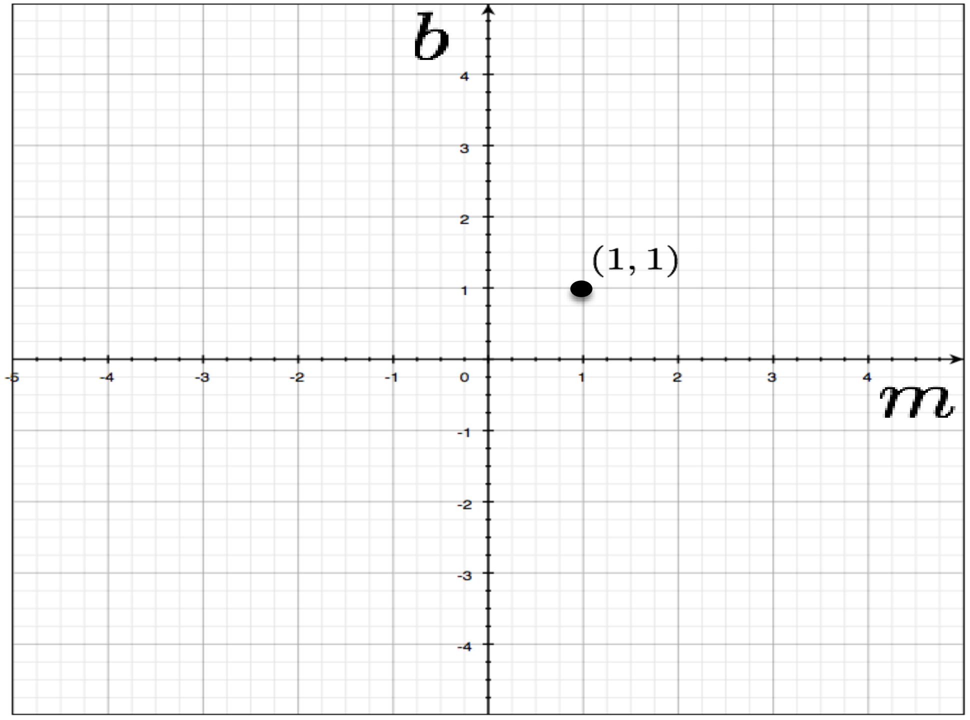


Image space

a line becomes a point



Parameter space

(m, b) Parameter space

variables

$$y = mx + b$$

parameters

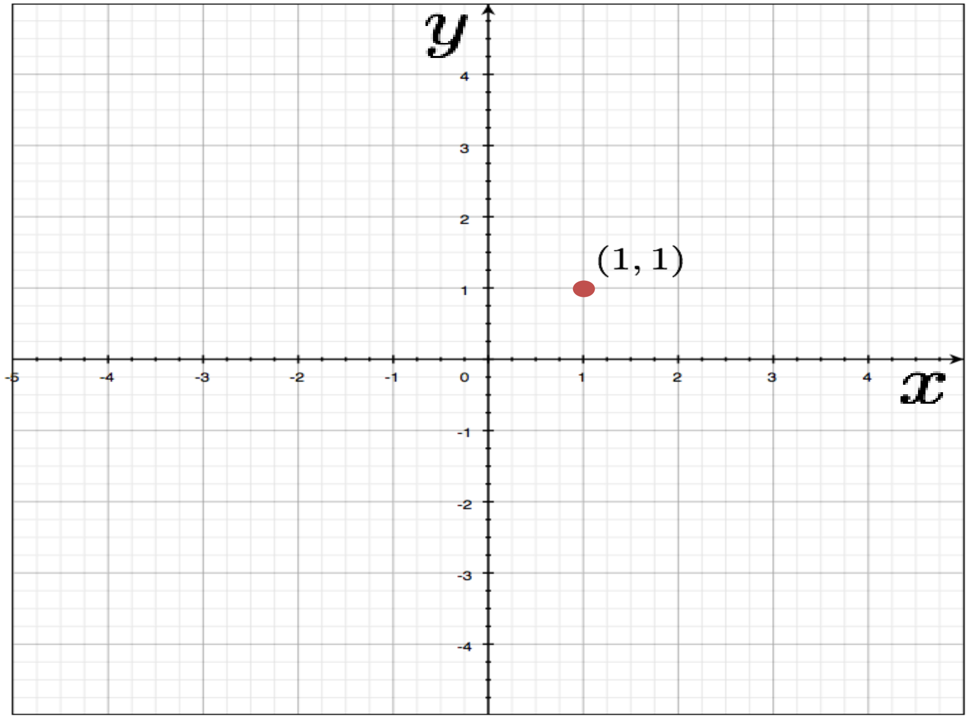


Image space

a point becomes a ?

(m, b) Parameter space

variables

$$y = mx + b$$

parameters

variables

$$y - mx = b$$

parameters

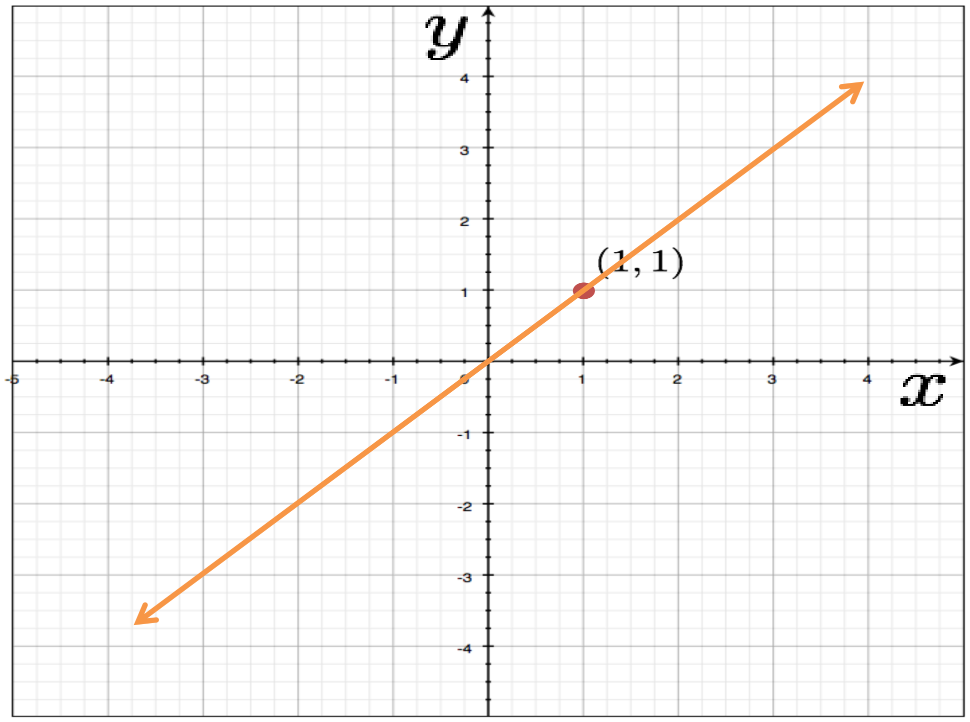
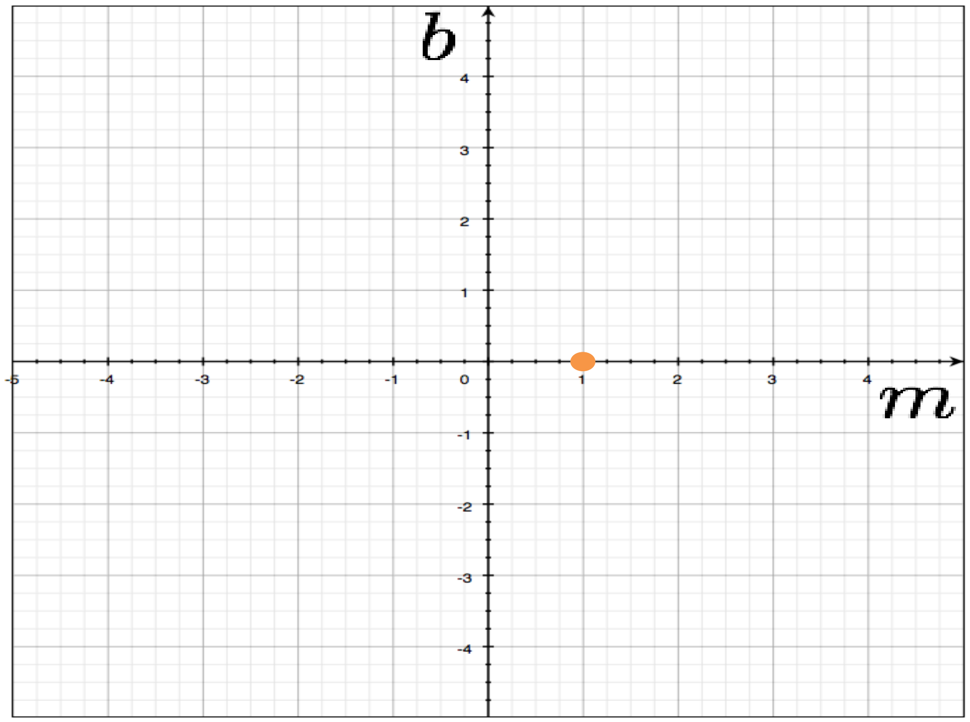


Image space

a point becomes a ?



Parameter space

(m, b) Parameter space

variables

$$y = mx + b$$

parameters

variables

$$y - mx = b$$

parameters

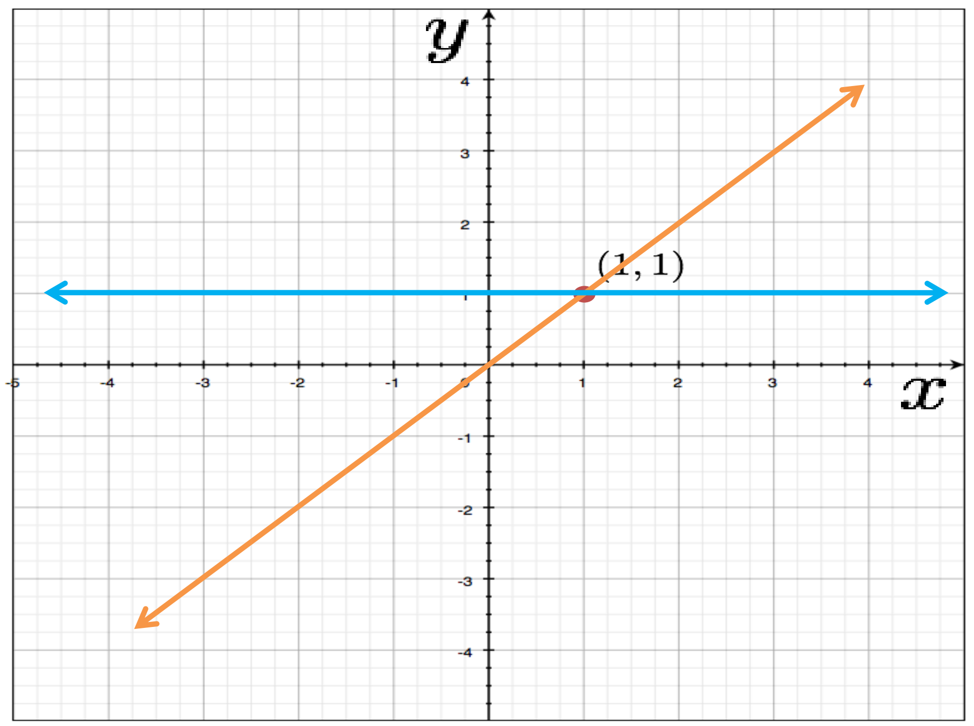
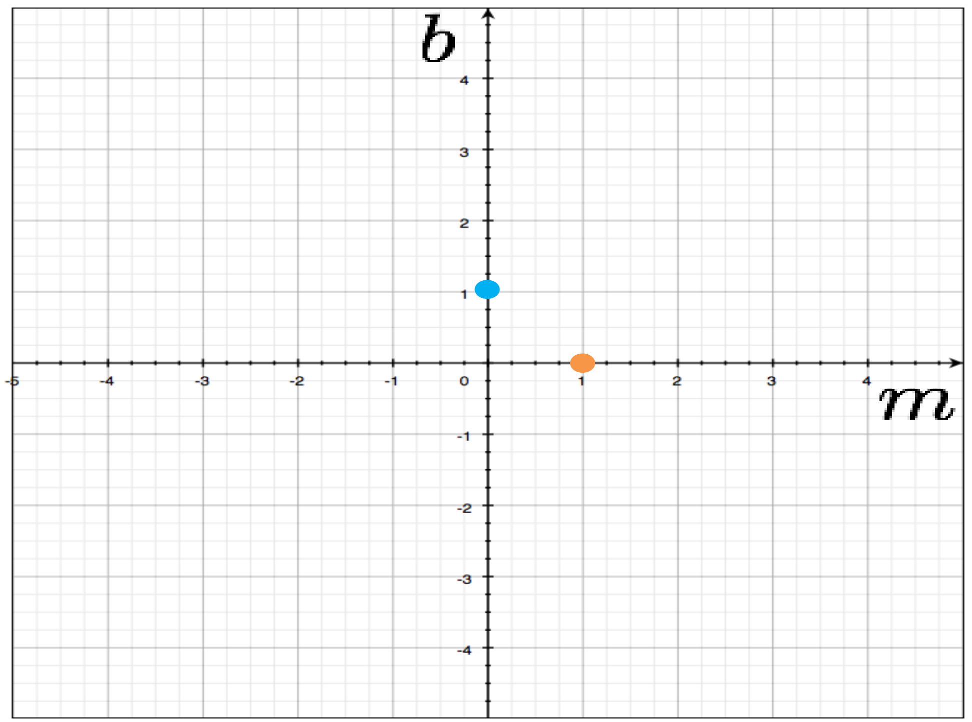


Image space

a point becomes a ?



Parameter space

(m, b) Parameter space

variables

$$y = mx + b$$

parameters

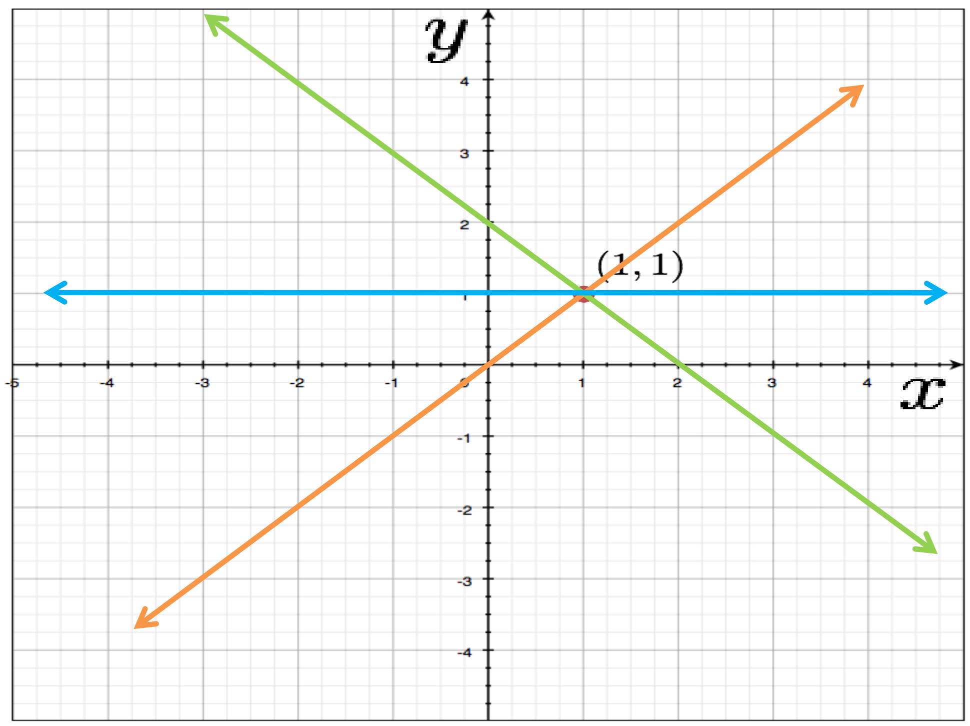


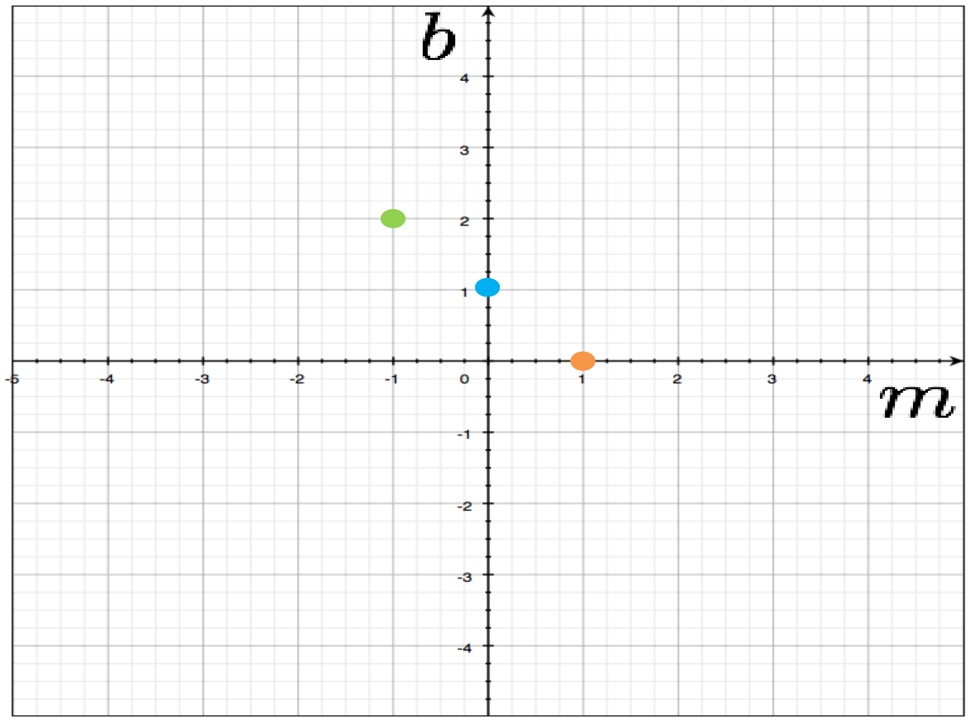
Image space

a point becomes a ?

variables

$$y - mx = b$$

parameters



Parameter space

(m, b) Parameter space

variables

$$y = mx + b$$

parameters

variables

$$y - mx = b$$

parameters

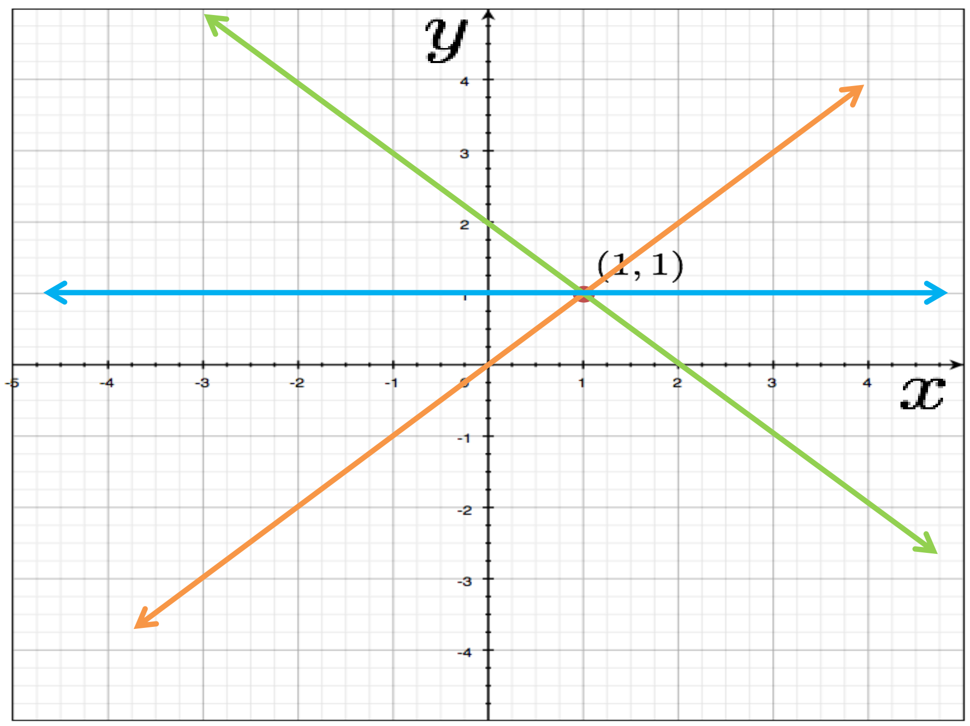
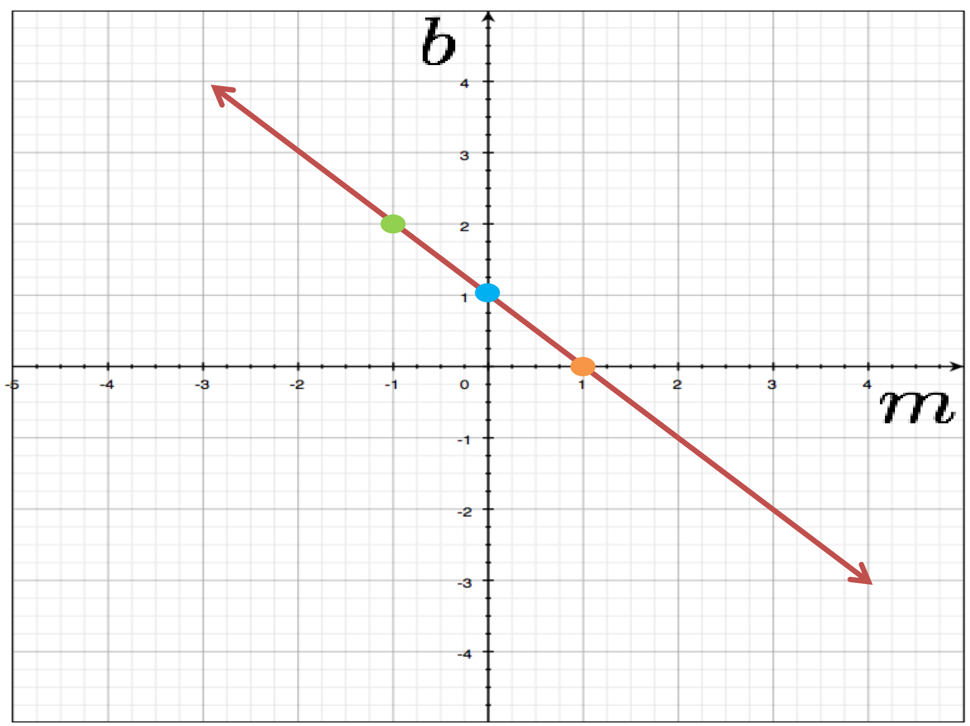


Image space

a point becomes a **line**



Parameter space

(m, b) Parameter space

variables

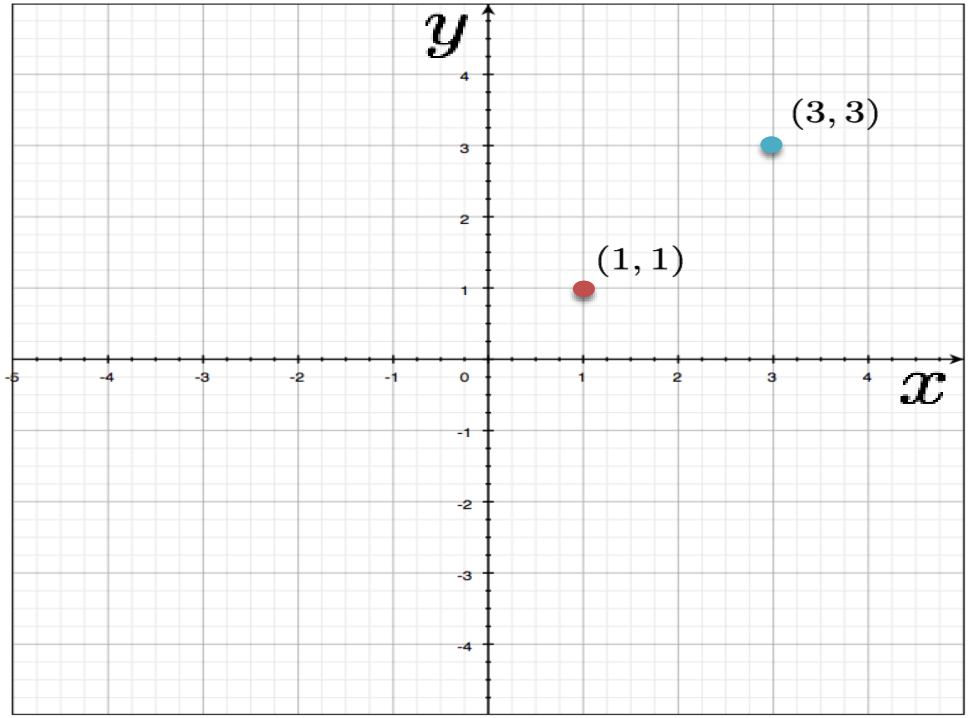
$$y = mx + b$$

parameters

variables

$$y - mx = b$$

parameters



two points become ?

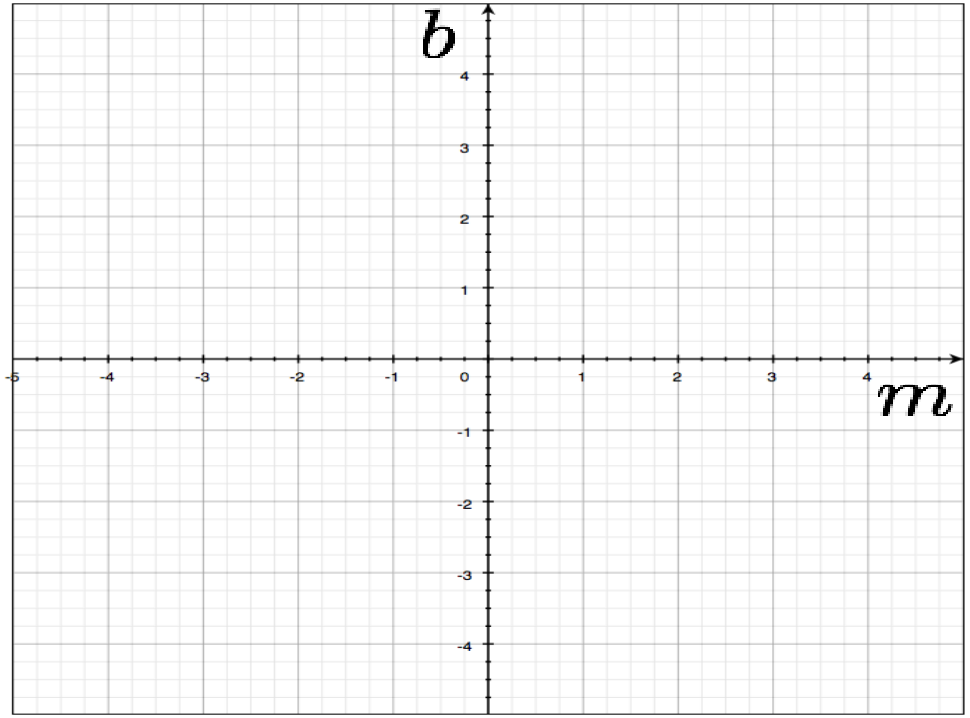


Image space

Parameter space

(m, b) Parameter space

variables

$$y = mx + b$$

parameters

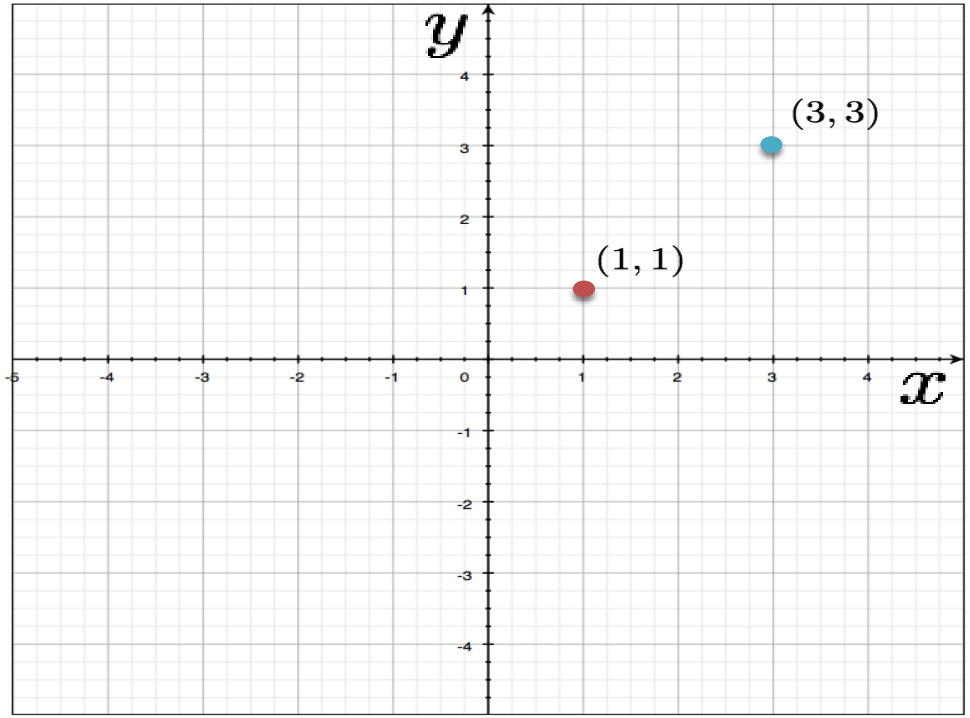


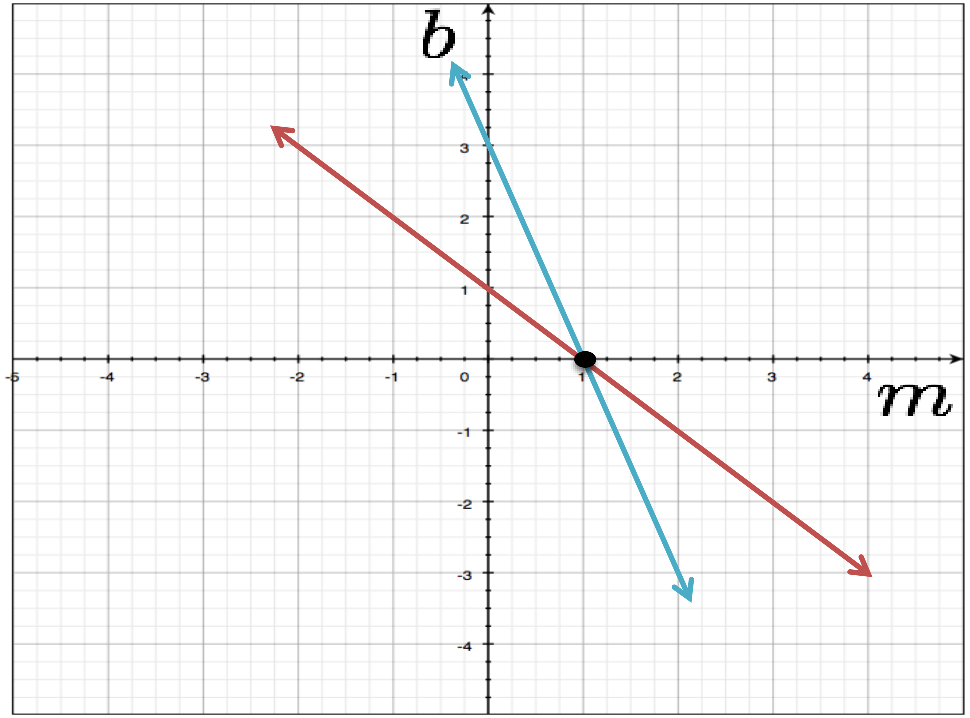
Image space

two points
become
Two lines

variables

$$y - mx = b$$

parameters



Parameter space

(m, b) Parameter space

variables

$$y = mx + b$$

parameters

variables

$$y - mx = b$$

parameters

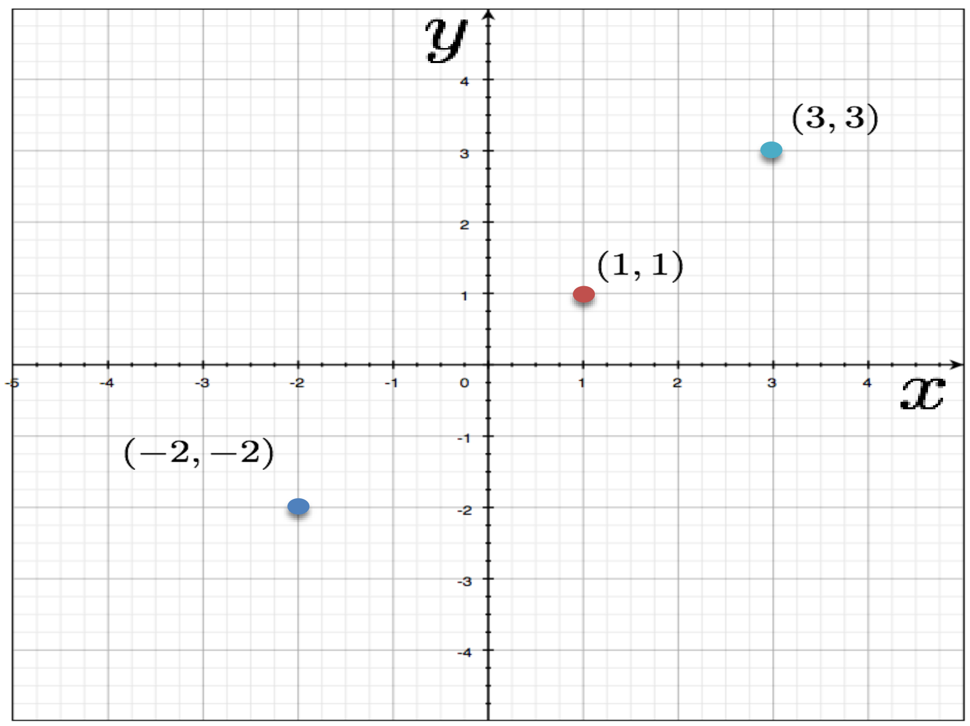
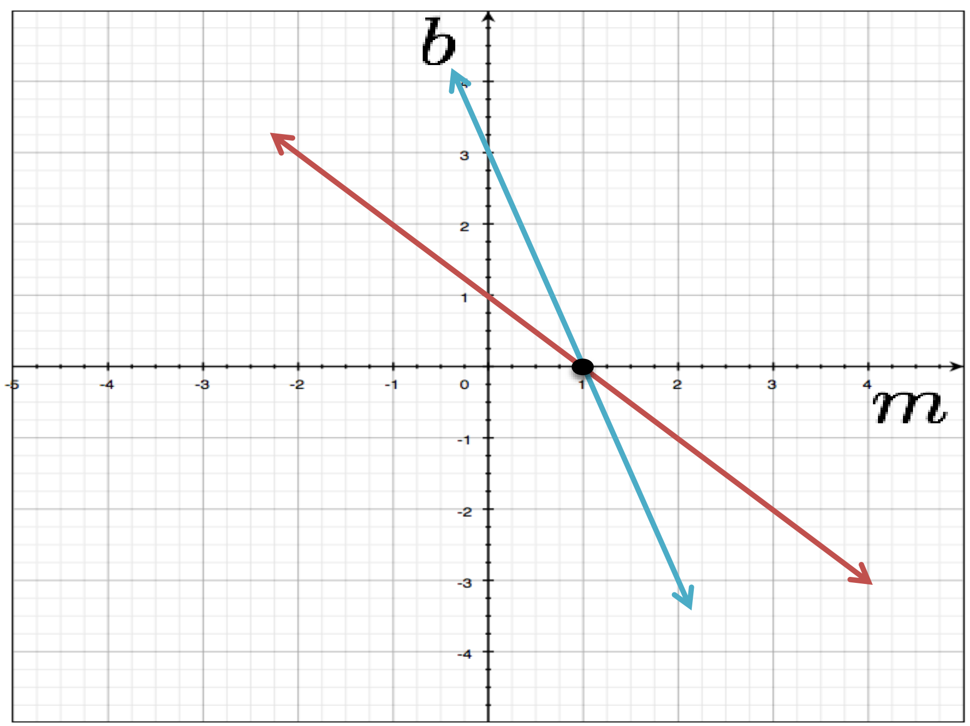


Image space

three points become ?



Parameter space

(m, b) Parameter space

variables

$$y = mx + b$$

parameters

variables

$$y - mx = b$$

parameters

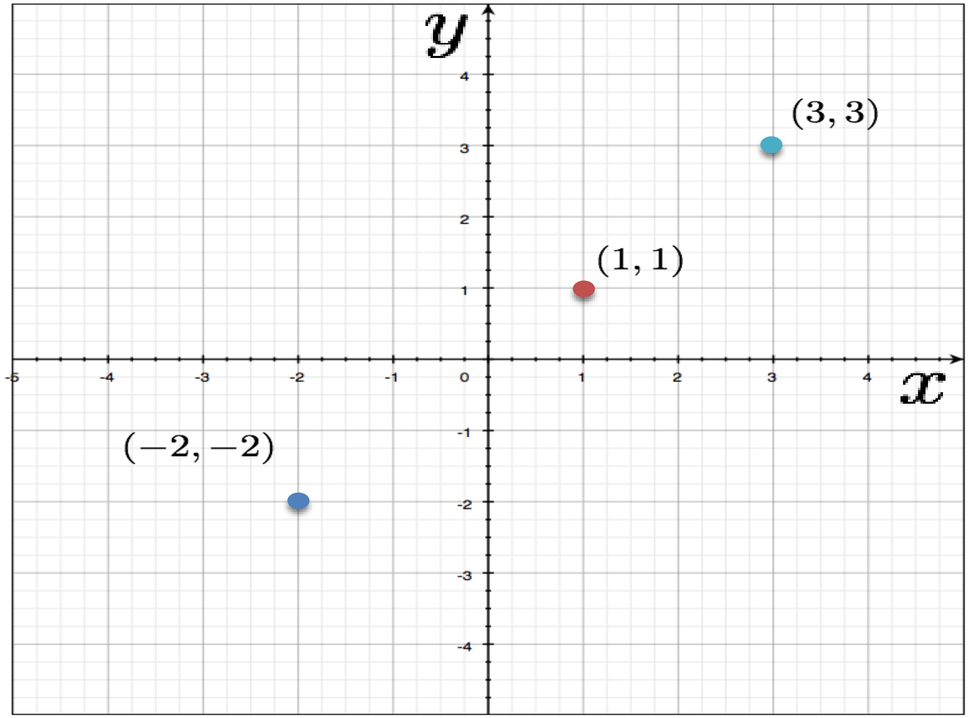
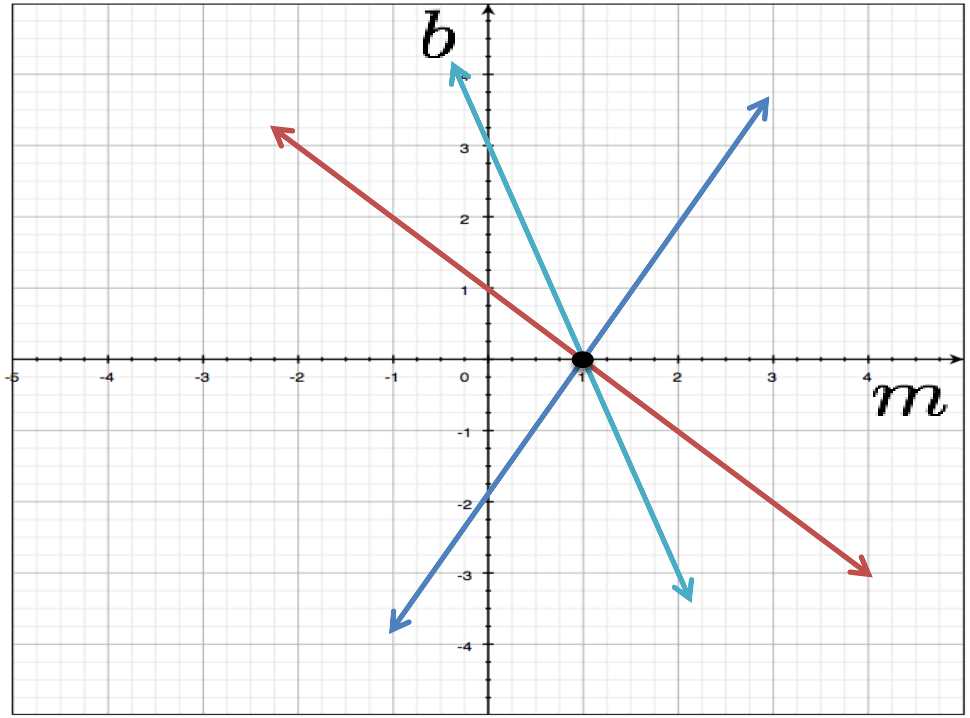


Image space

three points become Three lines



Parameter space

(m, b) Parameter space

variables

$$y = mx + b$$

parameters

variables

$$y - mx = b$$

parameters

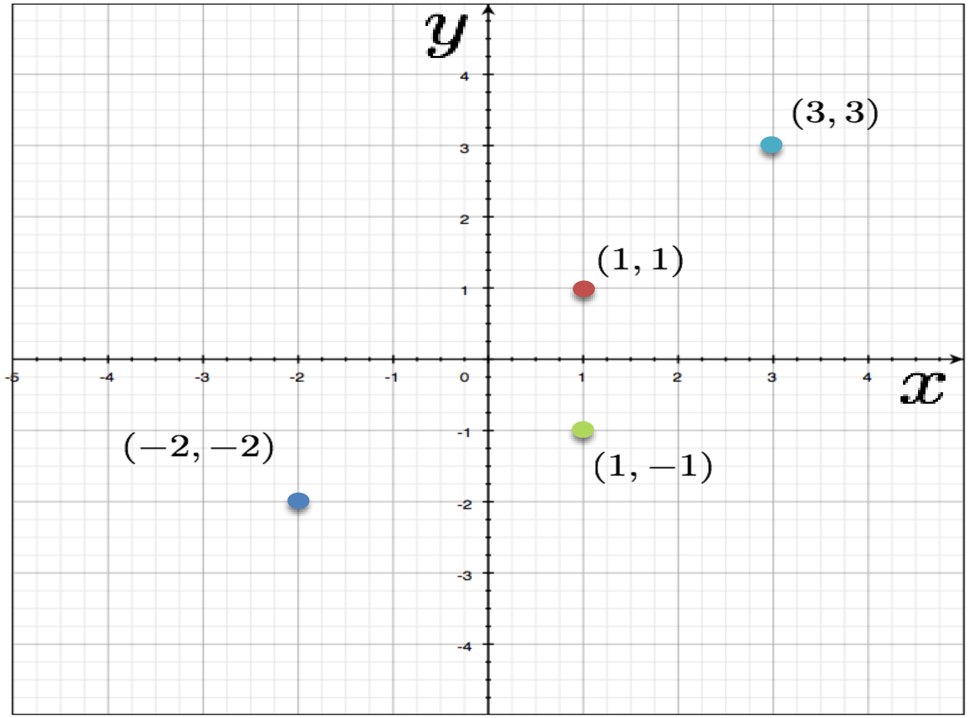
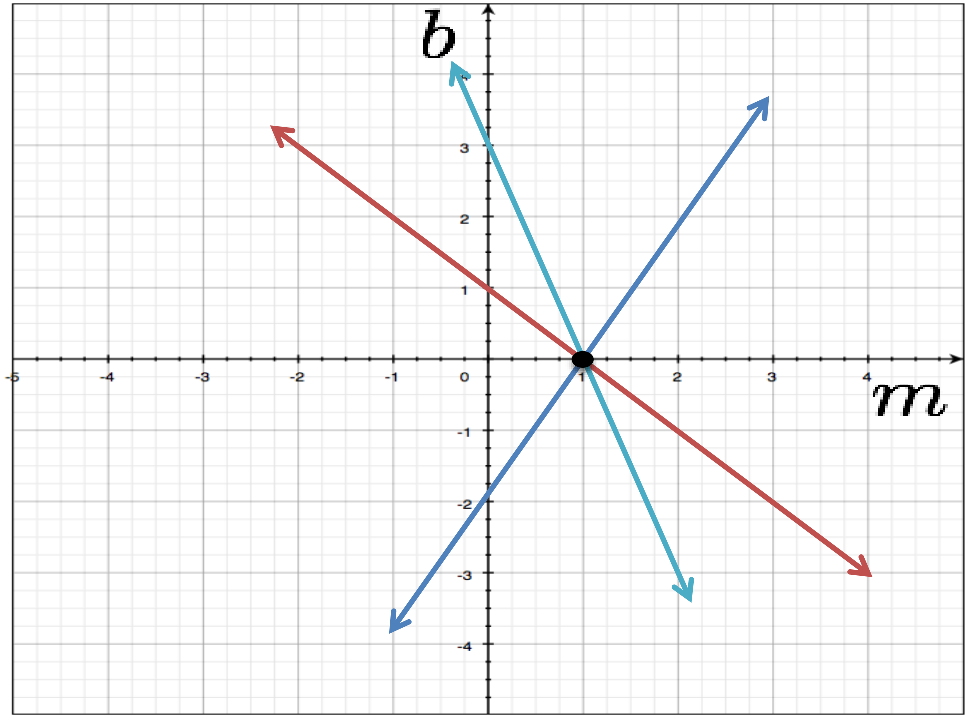


Image space

four points become ?



Parameter space

(m, b) Parameter space

variables

$$y = mx + b$$

parameters

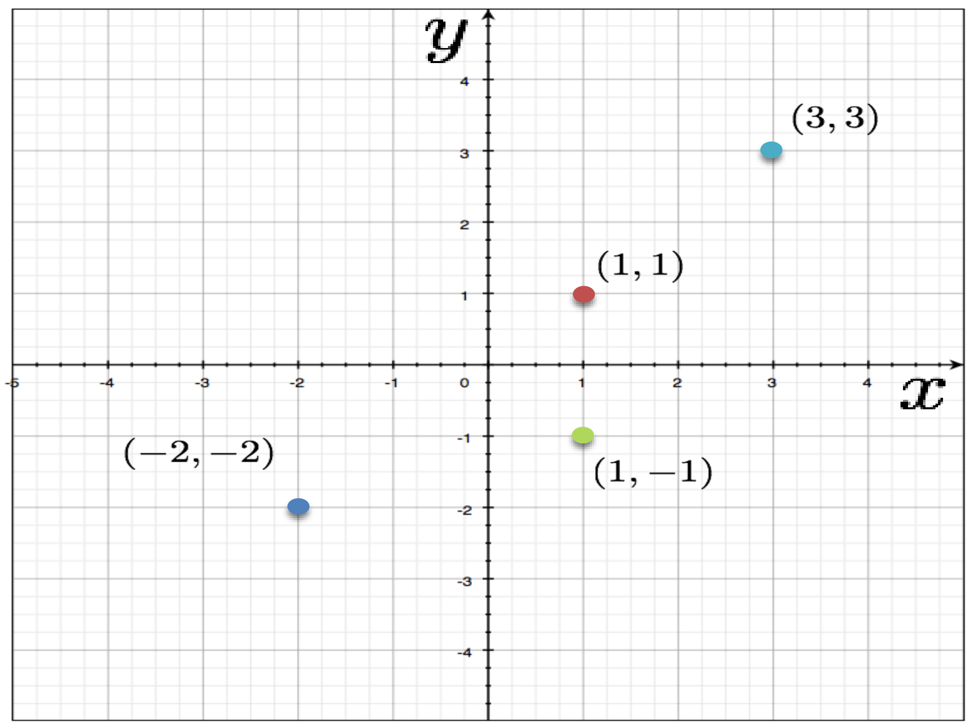


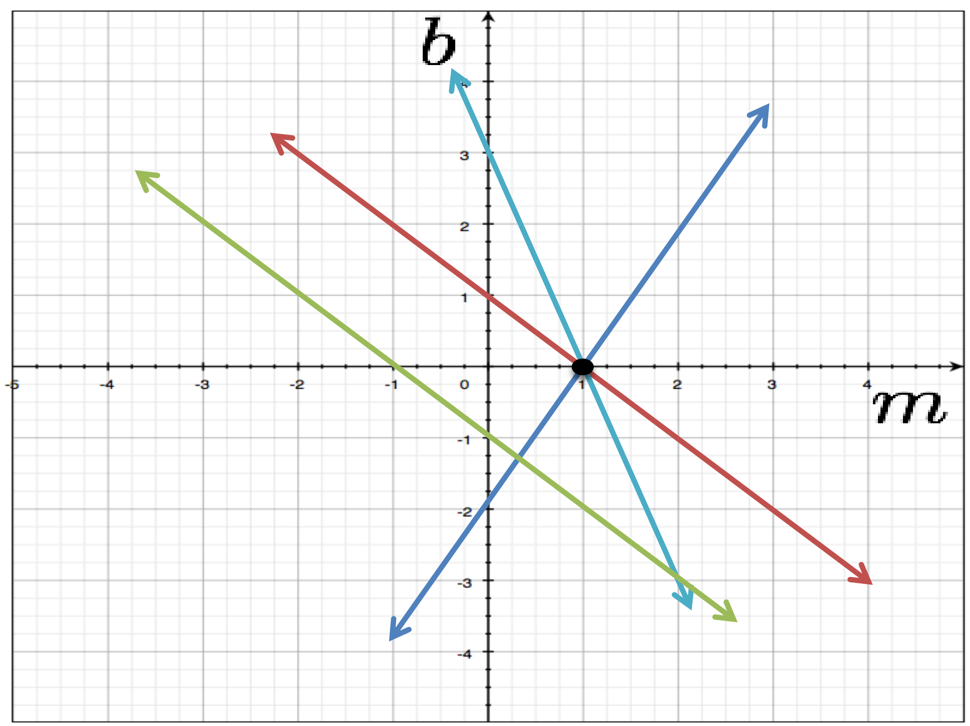
Image space

four points
become
Four lines

variables

$$y - mx = b$$

parameters



Parameter space

Parameter space- what for?

- How can we find lines in dataset using the parameter space?

Parameter space- what for?

- How can we find lines in dataset using the parameter space?

1. Quantize the output parameter space to user defined bins- we will call this table the **accumulator table.**

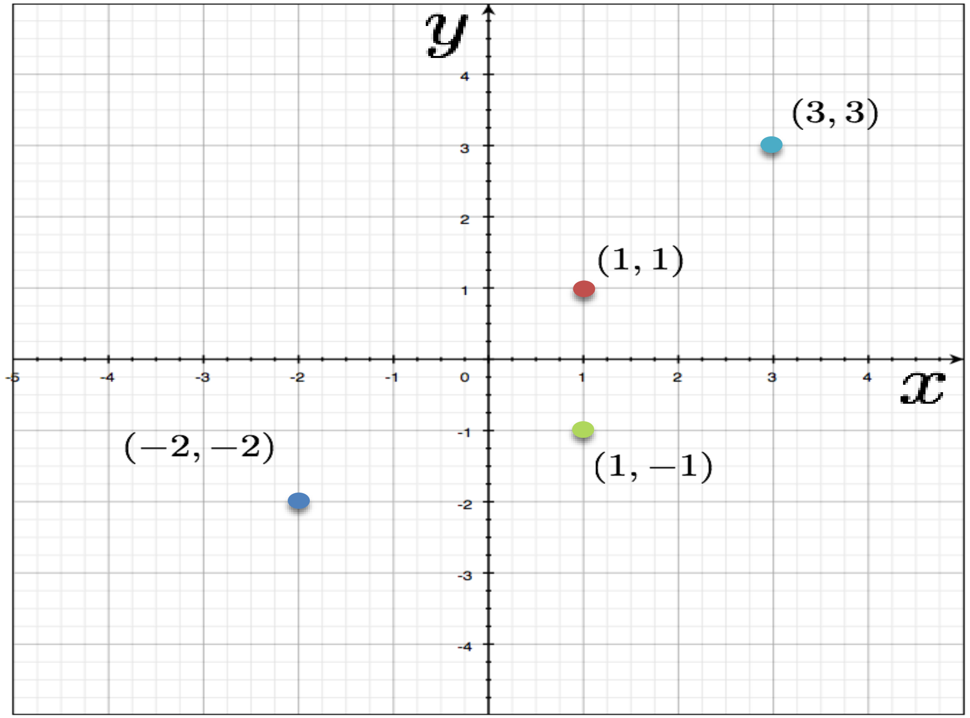
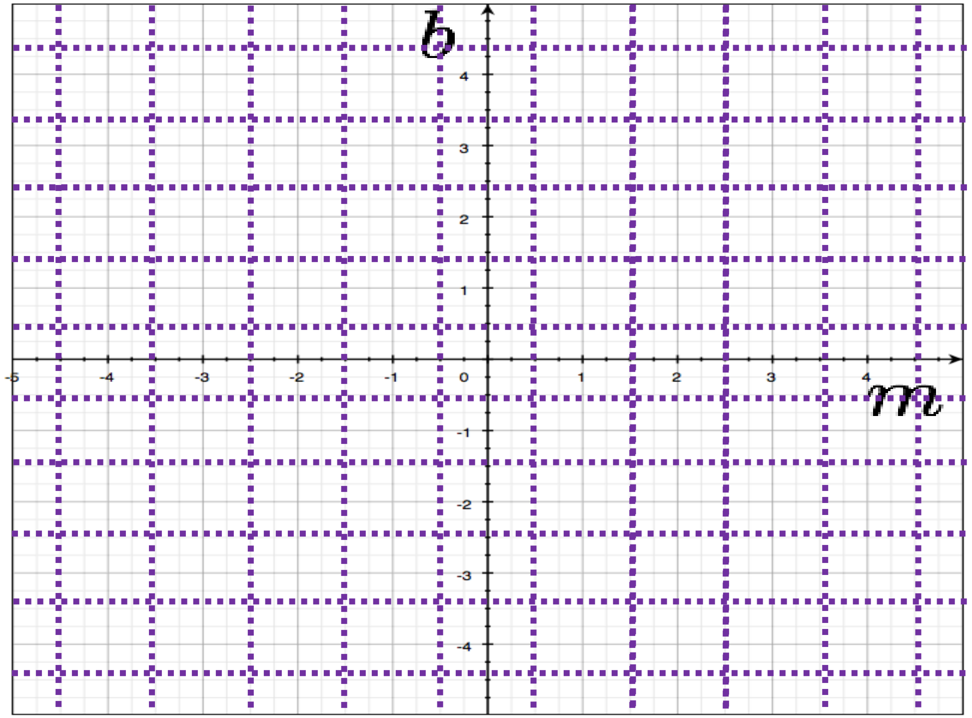


Image space



Parameter space

Parameter space- what for?

- How can we find lines in dataset using the parameter space?

2. For each point in image space- find corresponding line in parameter space and increment +1 the intersecting bins.

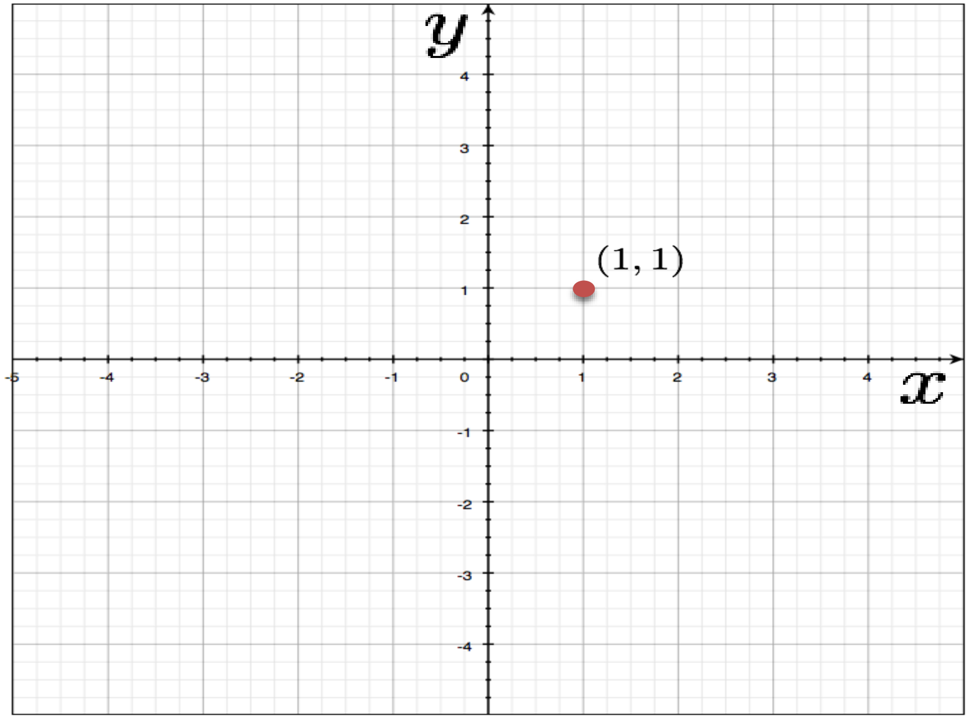
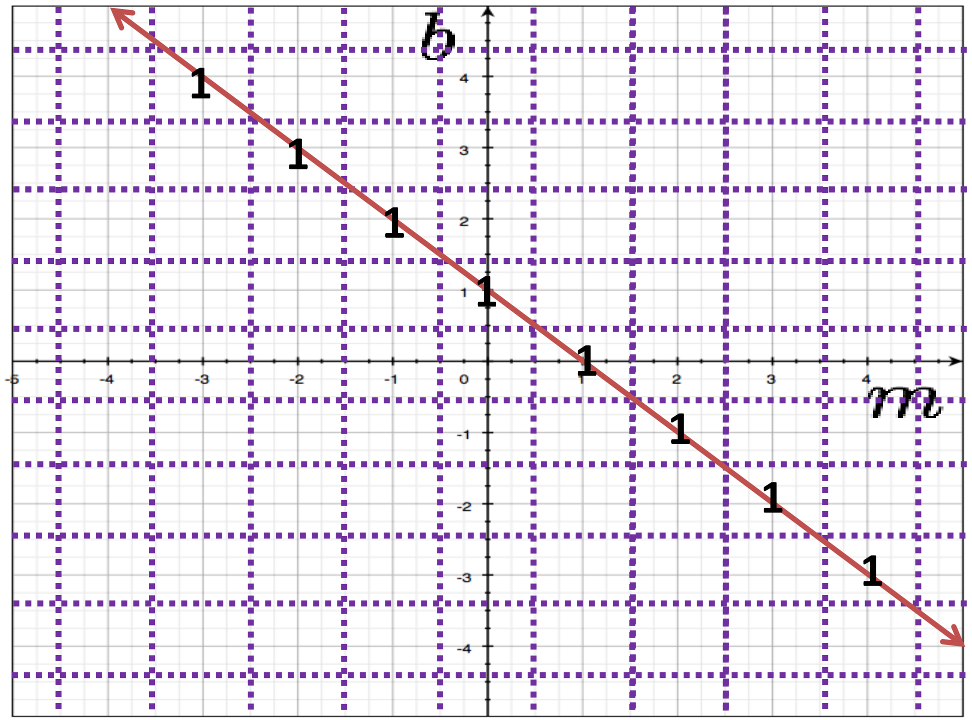


Image space



Parameter space

Parameter space- what for?

- How can we find lines in dataset using the parameter space?

2. For each point in image space- find corresponding line in parameter space and increment +1 the intersecting bins.

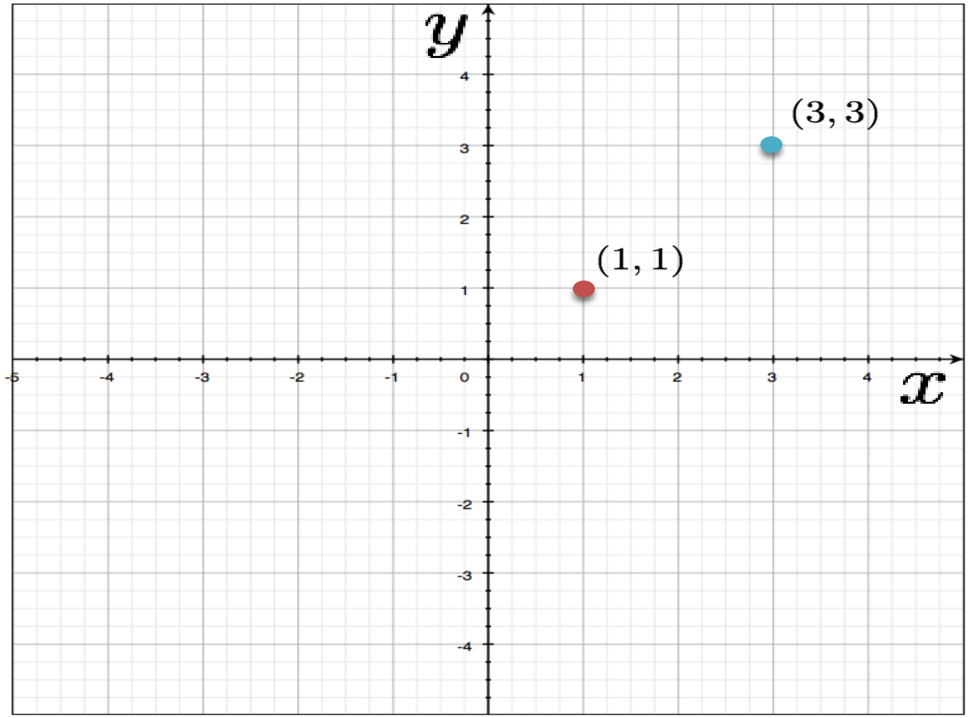
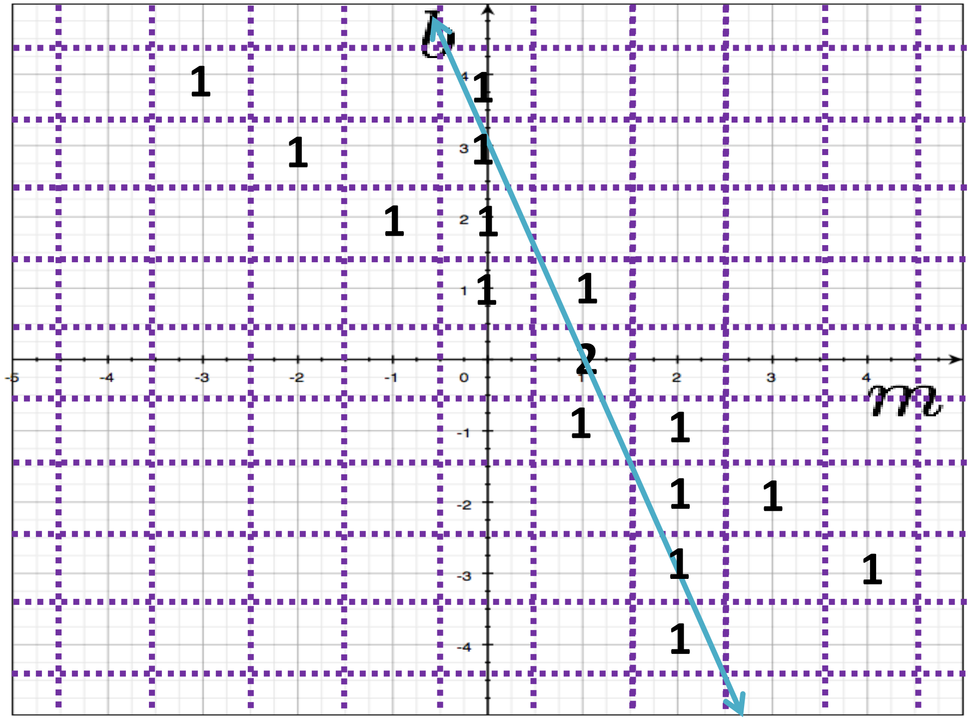


Image space



Parameter space

Parameter space- what for?

- How can we find lines in dataset using the parameter space?

2. For each point in image space- find corresponding line in parameter space and increment +1 the intersecting bins.

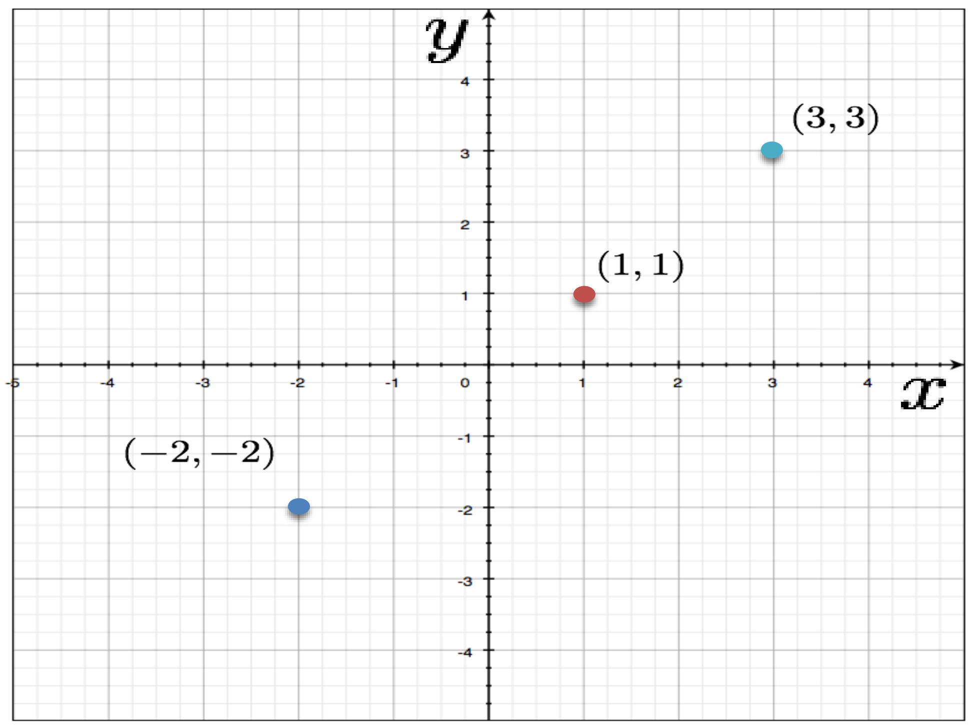
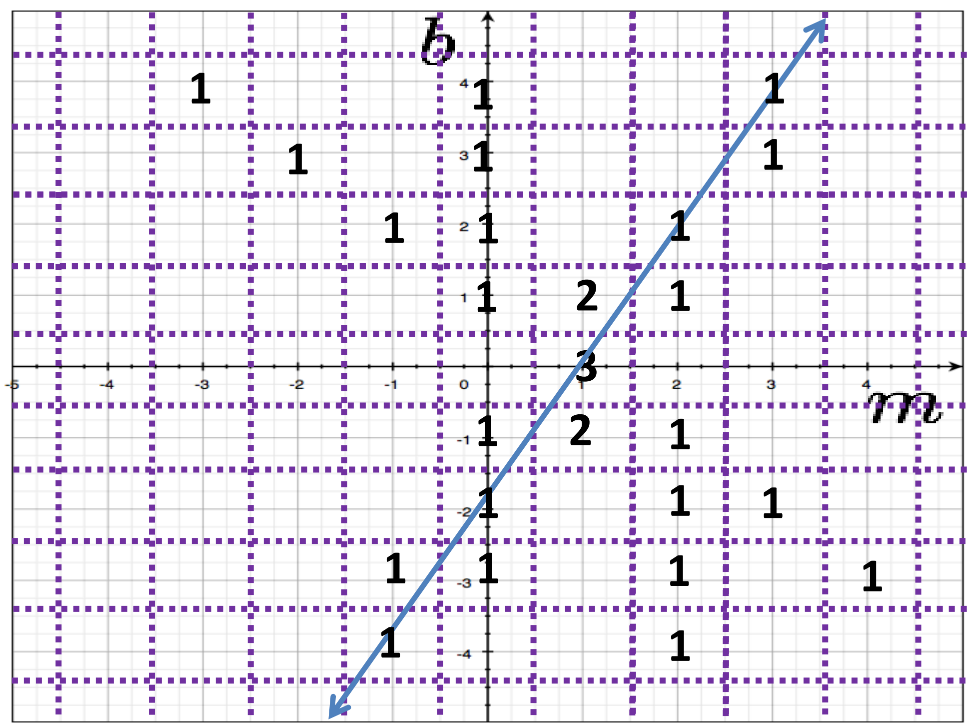


Image space



Parameter space

Parameter space- what for?

- How can we find lines in dataset using the parameter space?

2. For each point in image space- find corresponding line in parameter space and increment +1 the intersecting bins.

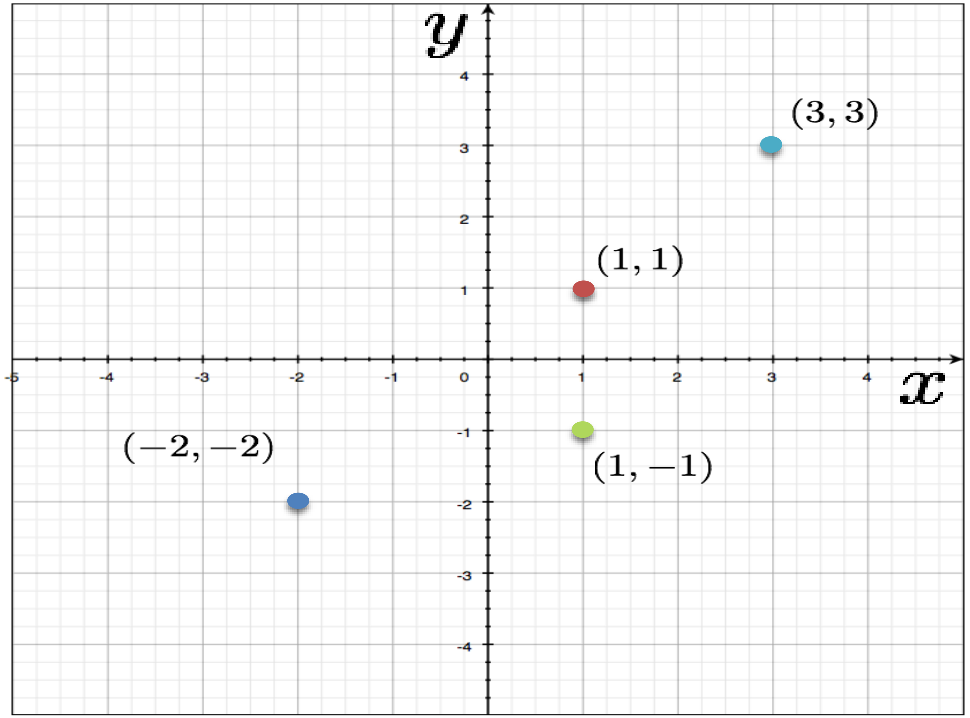
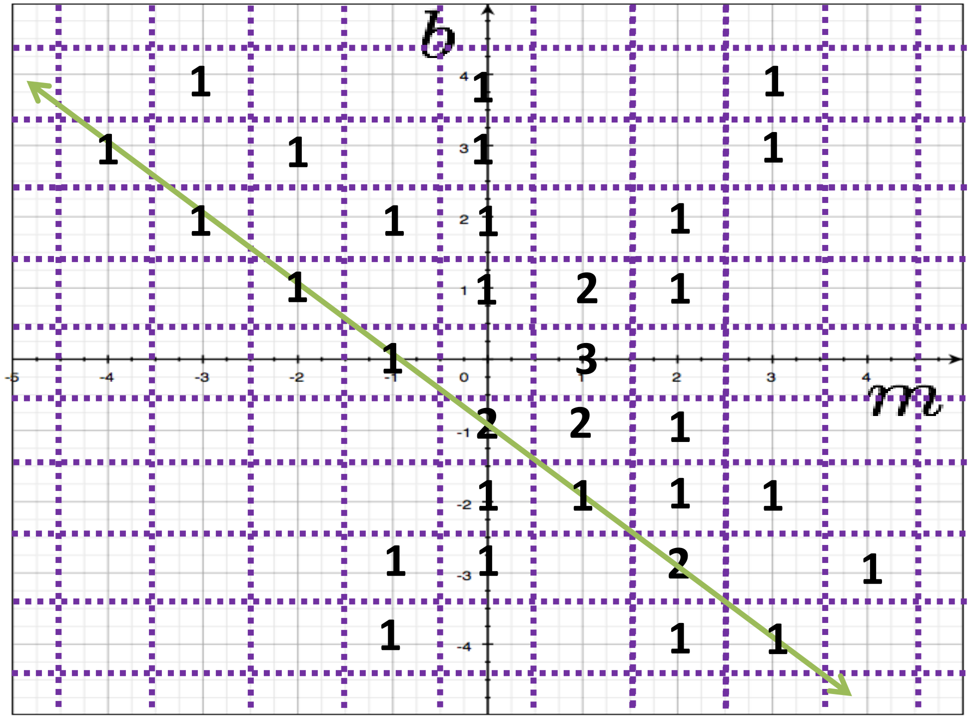


Image space



Parameter space

Parameter space- what for?

- How can we find lines in dataset using the parameter space?

3. Threshold the accumulator table result by some TH and get the corresponding line parameters.

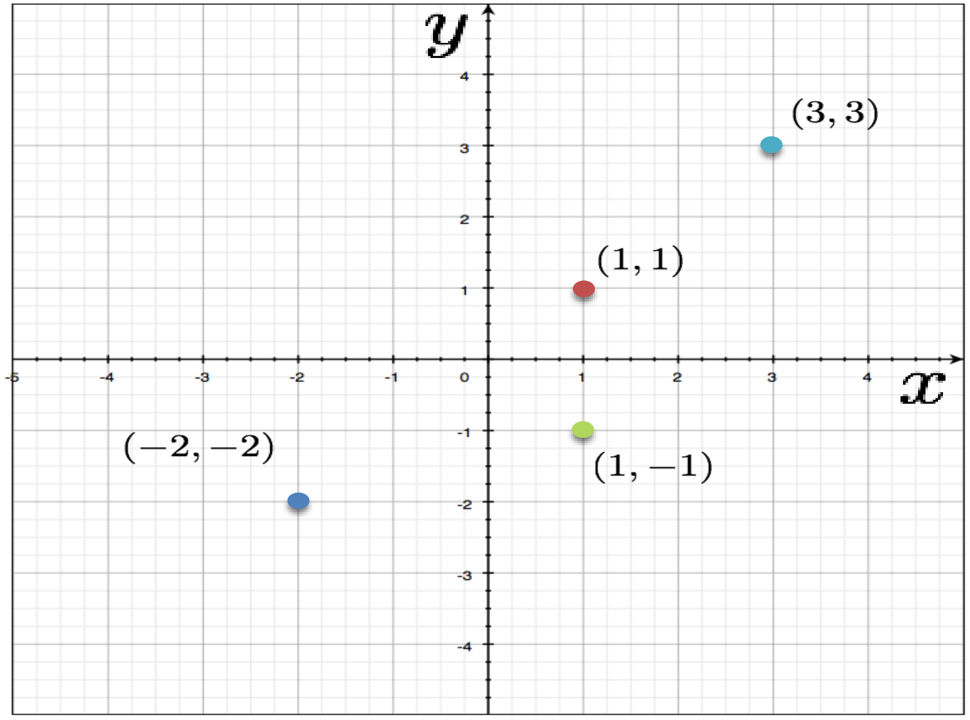
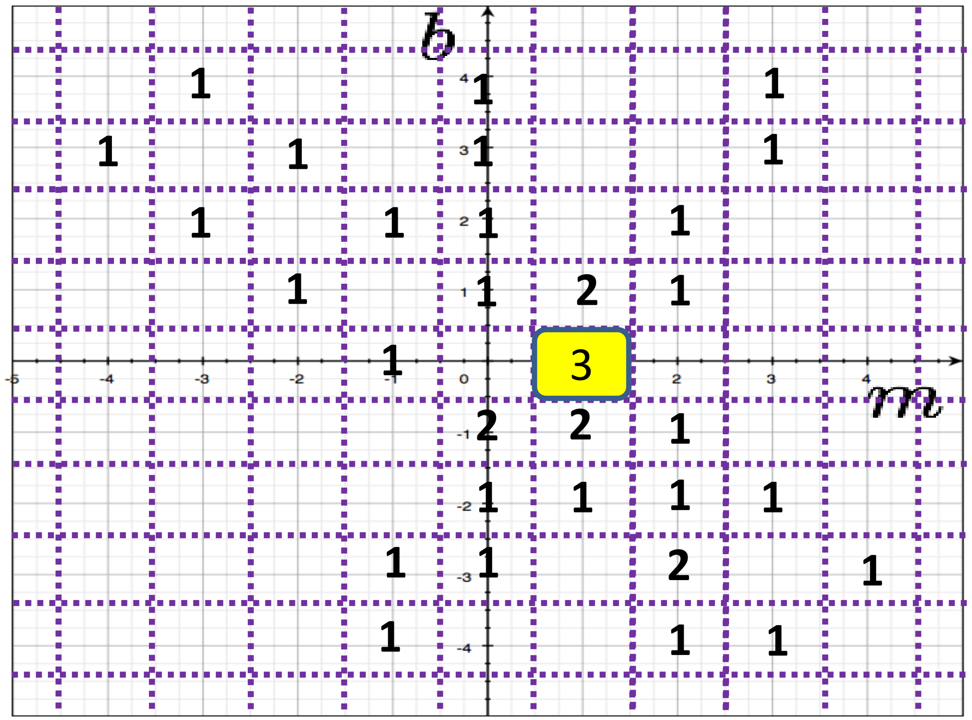


Image space



Parameter space

Hough transform

```
Build accumulator table.
For each point in image space:
   find corresponding line in parameter space and
   increment +1 the intersecting bins.
Threshold the accumulator table result by some TH and get
the corresponding line parameters.
```

Hough transform- pros & cons

- Pros:
  - Can detect multiple lines in image space.
  - can be extended to detect different parameterized curves (e.g.: circles, ellipsoids), and even un-parameterized curves (**generalized hough transform** [out of scope]- similar to template matching).
- Cons:
  - For the shown (m, b) parameter space, can't detect vertical lines. **<u>Why?</u>**
  - Susceptive to noise. **<u>Why?</u>**
  - Computationally costly.

Problem 1: vertical line detection

- Vertical (or near vertical) lines have a big slope: $m \to \infty$. This causes the accumulator table to be very big in m direction.
- A solution is to give a **different parameterization to lines**:

$$y = mx + b$$

Problem 1: vertical line detection

- Vertical (or near vertical) lines have a big slope: $m \to \infty$. This causes the accumulator table to be very big in m direction.
- A solution is to give a **different parameterization to lines**:

$$y = mx + b$$

$$\xrightarrow{x=0} y = b$$

Problem 1: vertical line detection

- Vertical (or near vertical) lines have a big slope: $m \to \infty$. This causes the accumulator table to be very big in m direction.
- A solution is to give a **different parameterization to lines**:

$$y = mx + b$$
$$\xrightarrow{x=0} y = b$$
$$\xrightarrow{y=0} x = -\frac{b}{m}$$

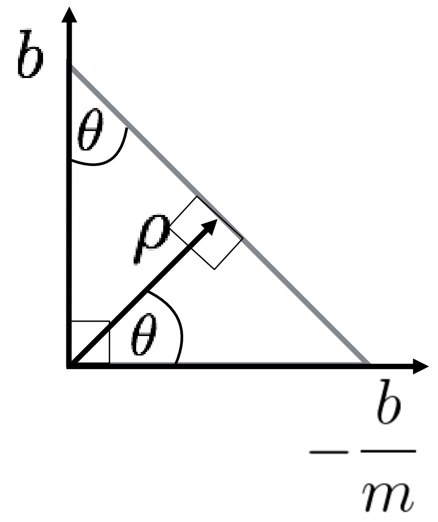
Problem 1: vertical line detection

- Vertical (or near vertical) lines have a big slope: $m \to \infty$. This causes the accumulator table to be very big in m direction.
- A solution is to give a **different parameterization to lines**:

$$y = mx + b$$

$$\xrightarrow{x=0} y = b$$

$$\xrightarrow{y=0} x = -\frac{b}{m}$$



Problem 1: vertical line detection

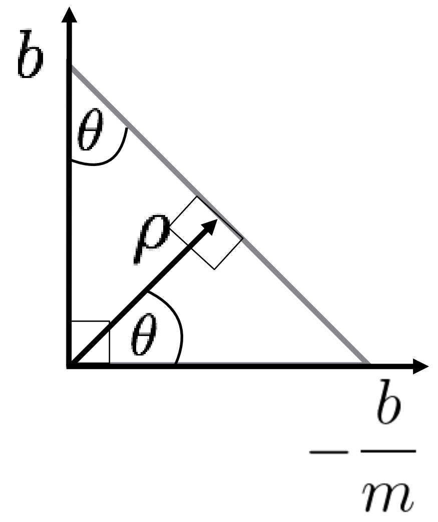
- Vertical (or near vertical) lines have a big slope: $m \to \infty$. This causes the accumulator table to be very big in m direction.
- A solution is to give a **different parameterization to lines**:

$$y = mx + b$$

$$\xrightarrow{x=0} y = b$$

$$\xrightarrow{y=0} x = -\frac{b}{m}$$

$$\frac{\frac{-b}{m}}{b} = -\frac{1}{m} = tg\theta$$



Problem 1: vertical line detection

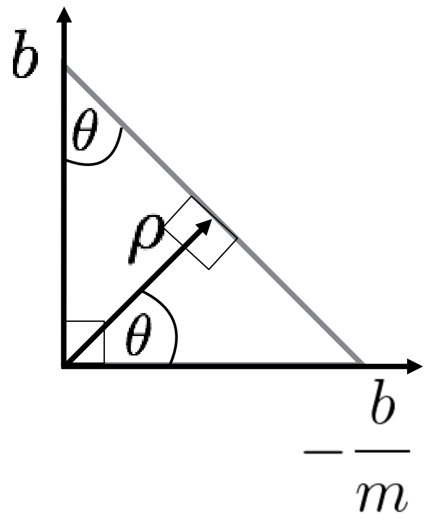
- Vertical (or near vertical) lines have a big slope: $m \to \infty$. This causes the accumulator table to be very big in m direction.
- A solution is to give a **different parameterization to lines**:

$$y = mx + b$$

$$\xrightarrow{x=0} y = b$$

$$\xrightarrow{y=0} x = -\frac{b}{m}$$

$$\frac{\frac{-b}{m}}{b} = -\frac{1}{m} = tg\theta \to m = -\frac{1}{tg\theta} = -\frac{cos\theta}{sin\theta}$$



Problem 1: vertical line detection

- Vertical (or near vertical) lines have a big slope: $m \to \infty$. This causes the accumulator table to be very big in m direction.
- A solution is to give a **different parameterization to lines**:

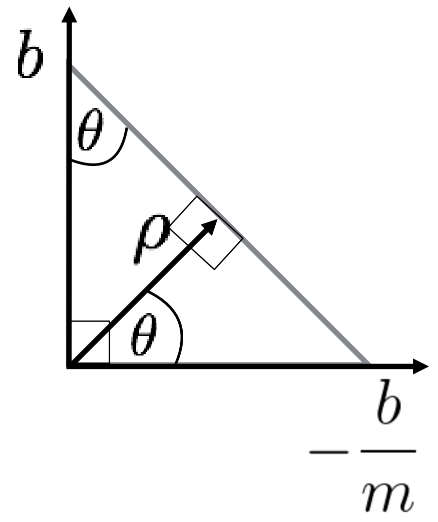
$$y = mx + b$$

$$\xrightarrow{x=0} y = b$$

$$\xrightarrow{y=0} x = -\frac{b}{m}$$

$$\frac{\frac{-b}{m}}{b} = -\frac{1}{m} = tg\theta \to m = -\frac{1}{tg\theta} = -\frac{cos\theta}{sin\theta}$$

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



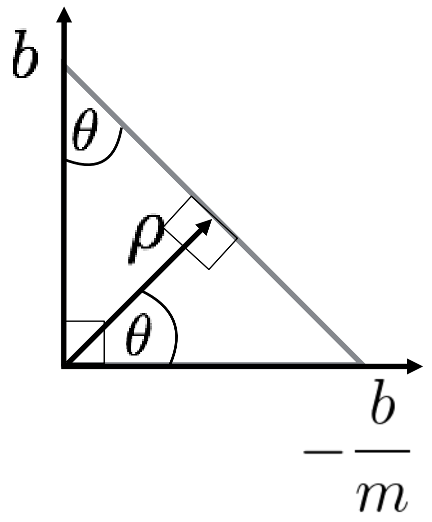
Problem 1: vertical line detection

- Vertical (or near vertical) lines have a big slope: $m \to \infty$. This causes the accumulator table to be very big in m direction.
- A solution is to give a **different parameterization to lines**:

$$y = mx + b$$

$$\xrightarrow{x=0} y = b$$

$$\xrightarrow{y=0} x = -\frac{b}{m}$$



$$\frac{\frac{-b}{m}}{b} = -\frac{1}{m} = tg\theta \to m = -\frac{1}{tg\theta} = -\frac{cos\theta}{sin\theta}$$

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

$$y = mx + b \to y = -\frac{cos\theta}{sin\theta}x + \frac{\rho}{sin\theta}$$

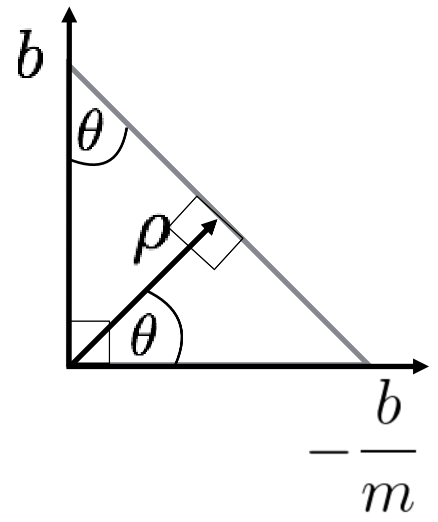
Problem 1: vertical line detection

- Vertical (or near vertical) lines have a big slope: $m \to \infty$. This causes the accumulator table to be very big in m direction.
- A solution is to give a **different parameterization to lines**:

$$y = mx + b$$

$$\xrightarrow{x=0} y = b$$

$$\xrightarrow{y=0} x = -\frac{b}{m}$$



$$\frac{\frac{-b}{m}}{b} = -\frac{1}{m} = tg\theta \to m = -\frac{1}{tg\theta} = -\frac{cos\theta}{sin\theta}$$

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

$$y = mx + b \to y = -\frac{cos\theta}{sin\theta}x + \frac{\rho}{sin\theta} \to \boxed{xcos\theta + ysin\theta = \rho}$$

TOC

- Hough transform
  - (m, b) parameter space
  - $(\boldsymbol{\rho}, \boldsymbol{\theta})$ **parameter space**

(ρ, θ) parameter space

variables

$$y = mx + b$$

parameters

parameters

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

variables

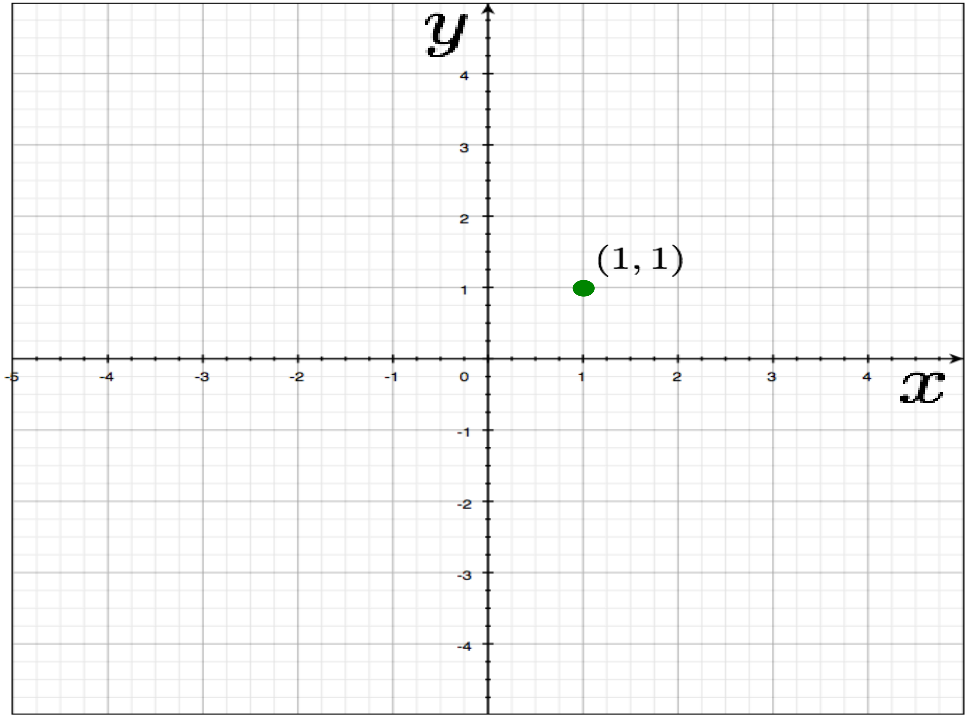
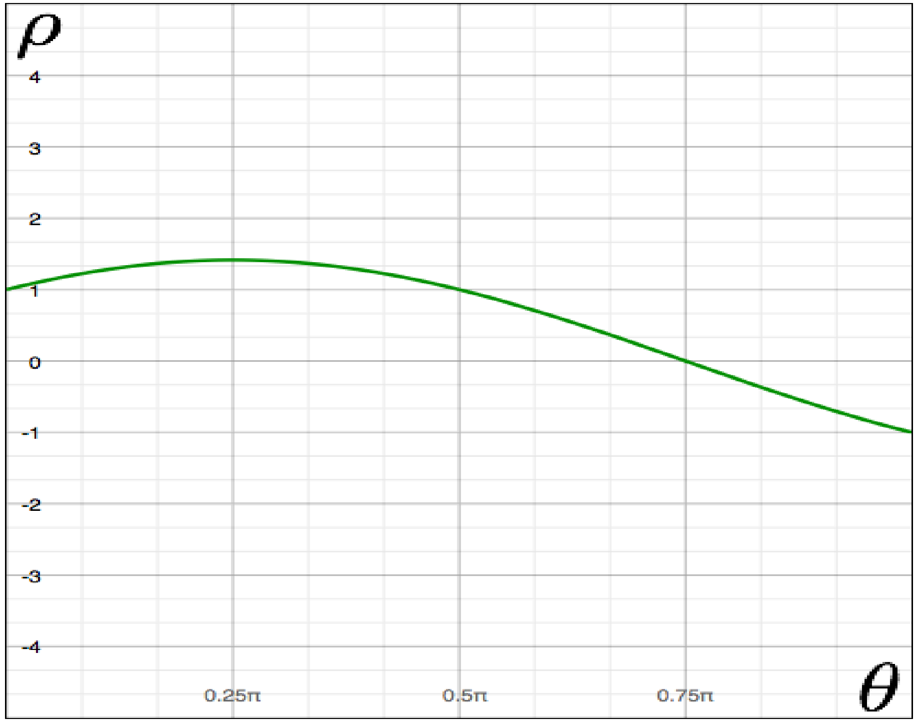


Image space

a point becomes a wave



Parameter space

(ρ, θ) parameter space

variables

$$y = mx + b$$

parameters

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

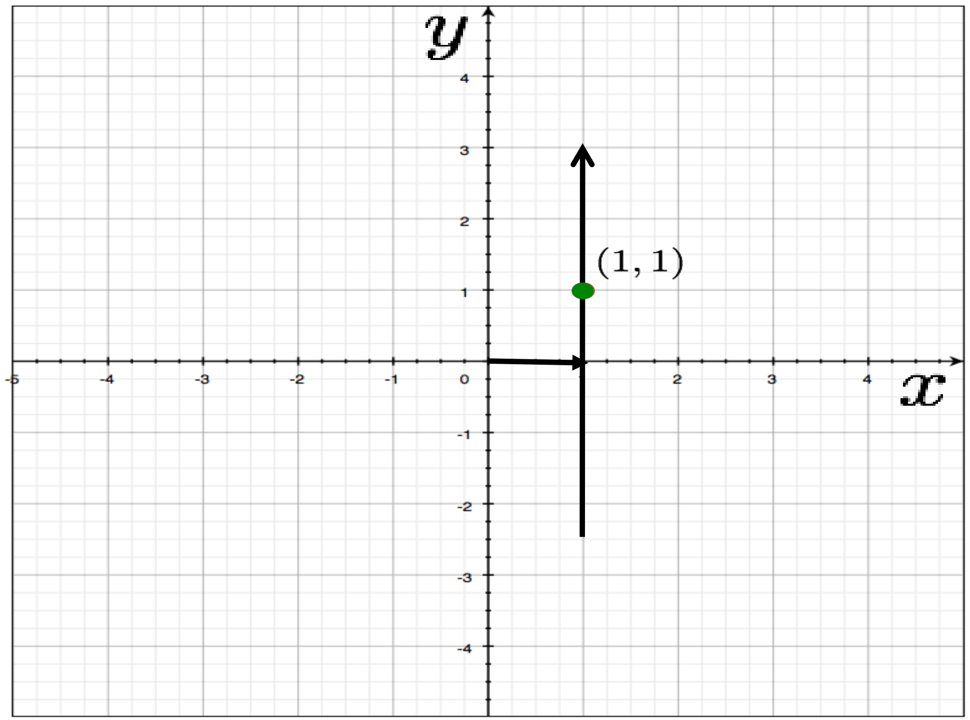
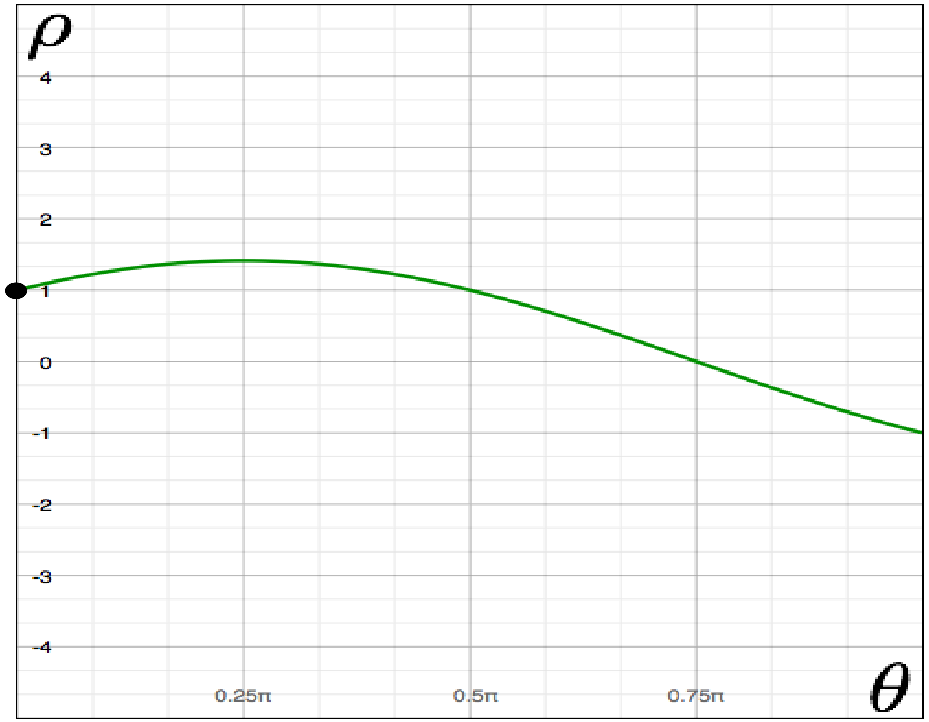


Image space

a line becomes a point



Parameter space

$(\boldsymbol{\rho}, \boldsymbol{\theta})$ parameter space

variables

$$y = mx + b$$

parameters

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

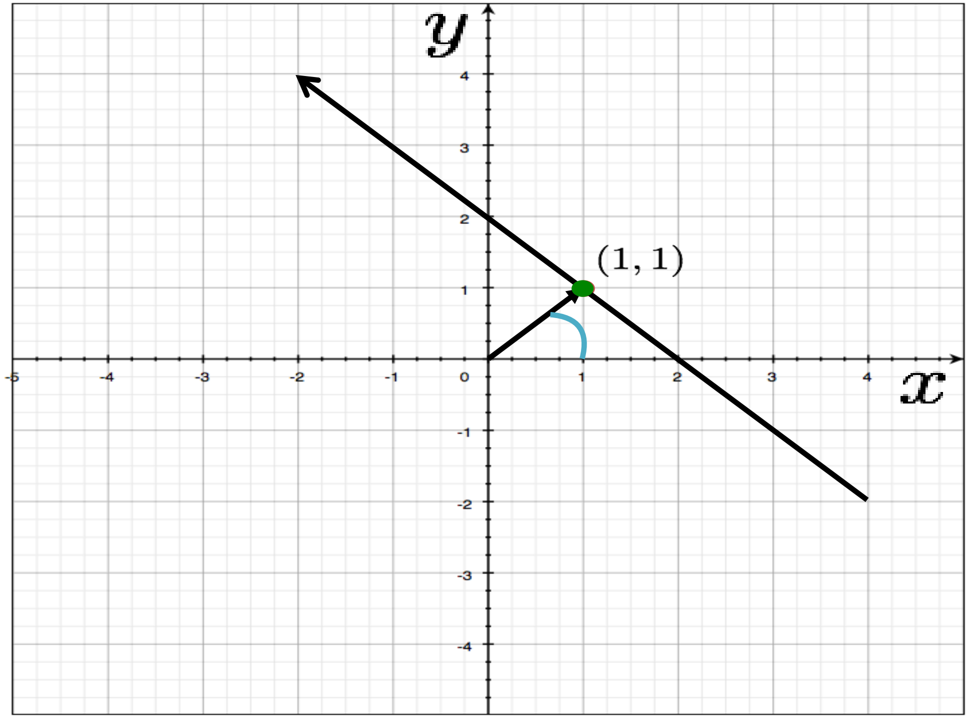
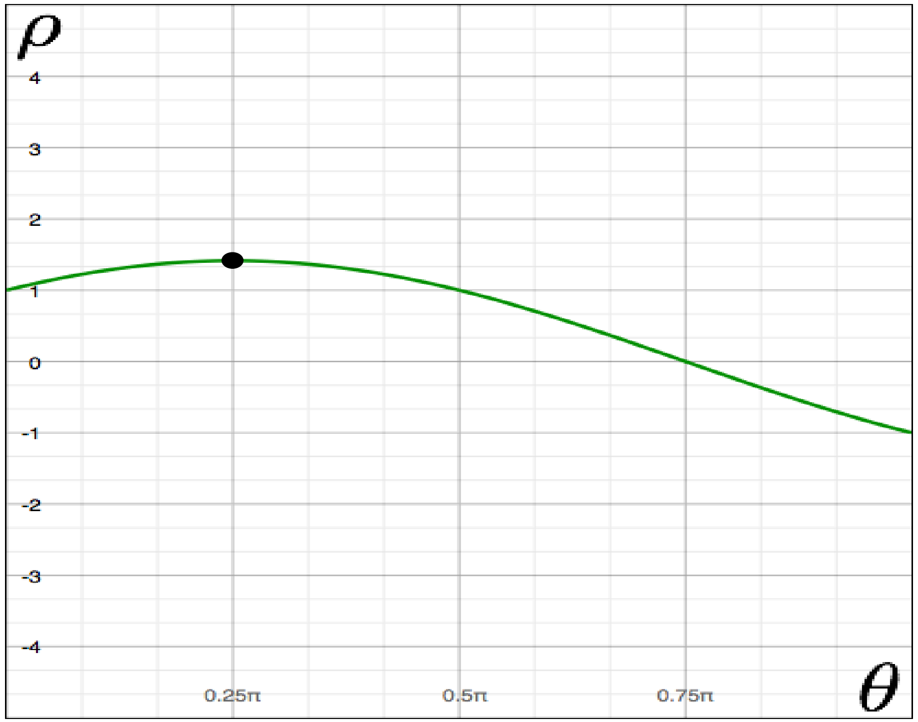


Image space

a line becomes a point



Parameter space

(ρ, θ) parameter space

variables

$$y = mx + b$$

parameters

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

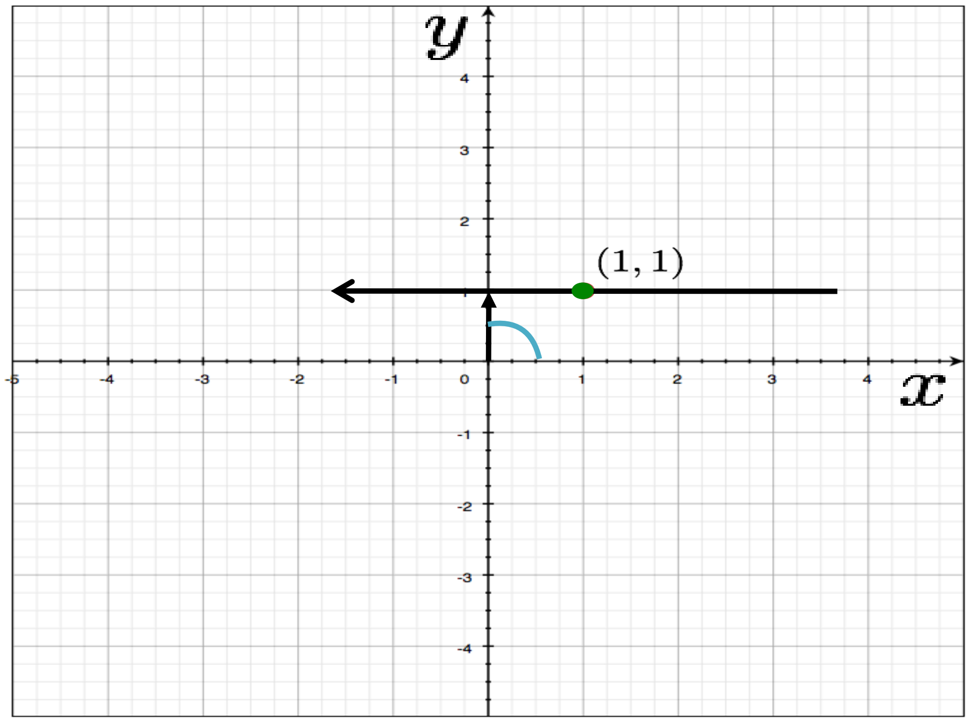
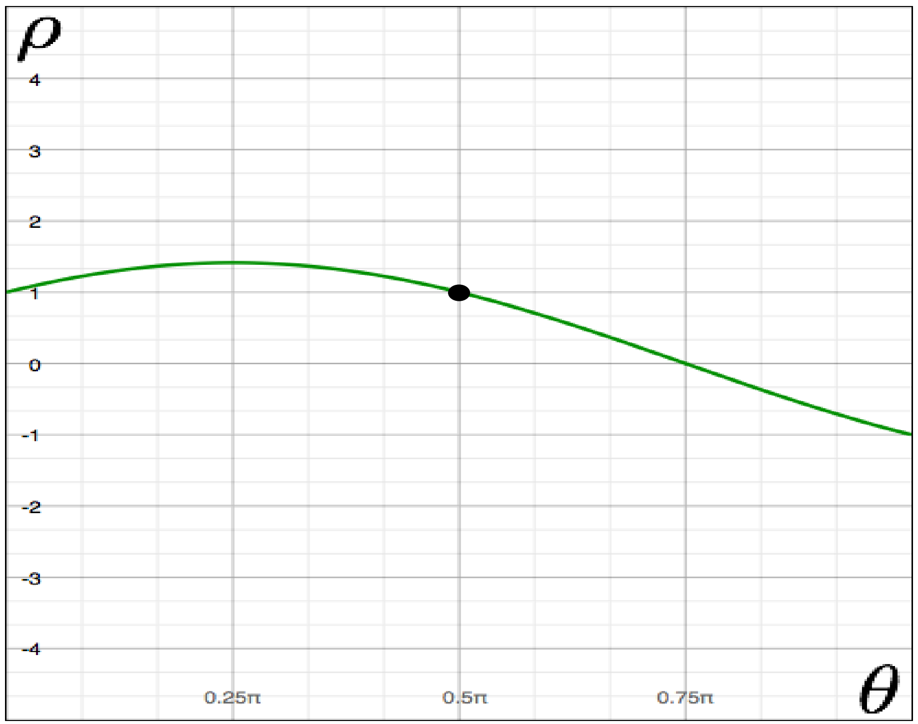


Image space

a line becomes a point



Parameter space

(ρ, θ) parameter space

variables

$$y = mx + b$$

parameters

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

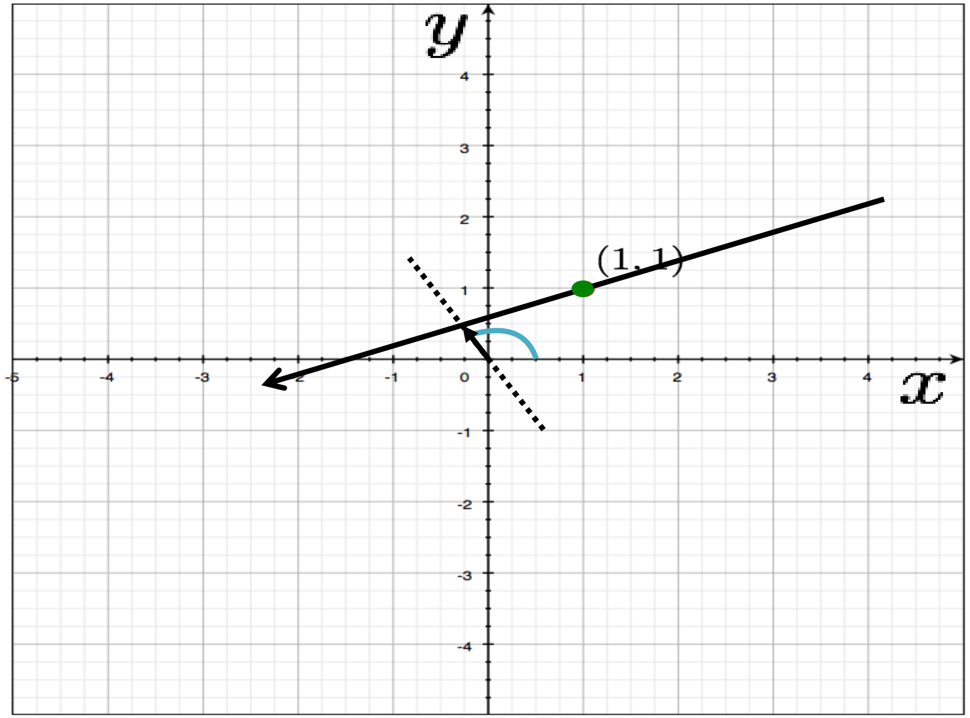
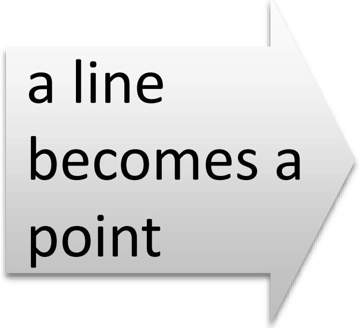
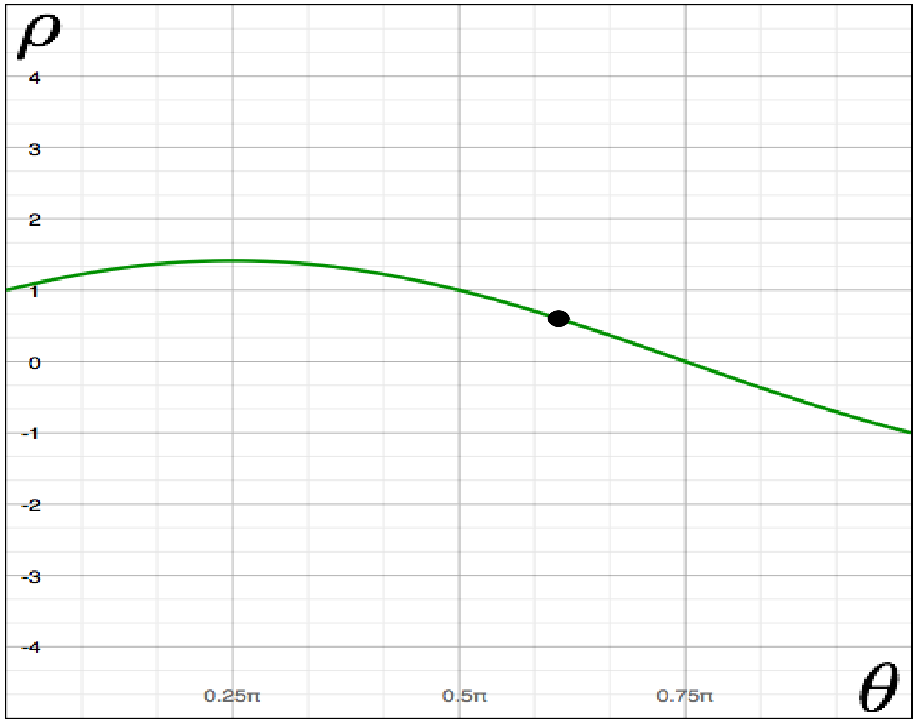


Image space





Parameter space

(ρ, θ) parameter space

variables

$$y = mx + b$$

parameters

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

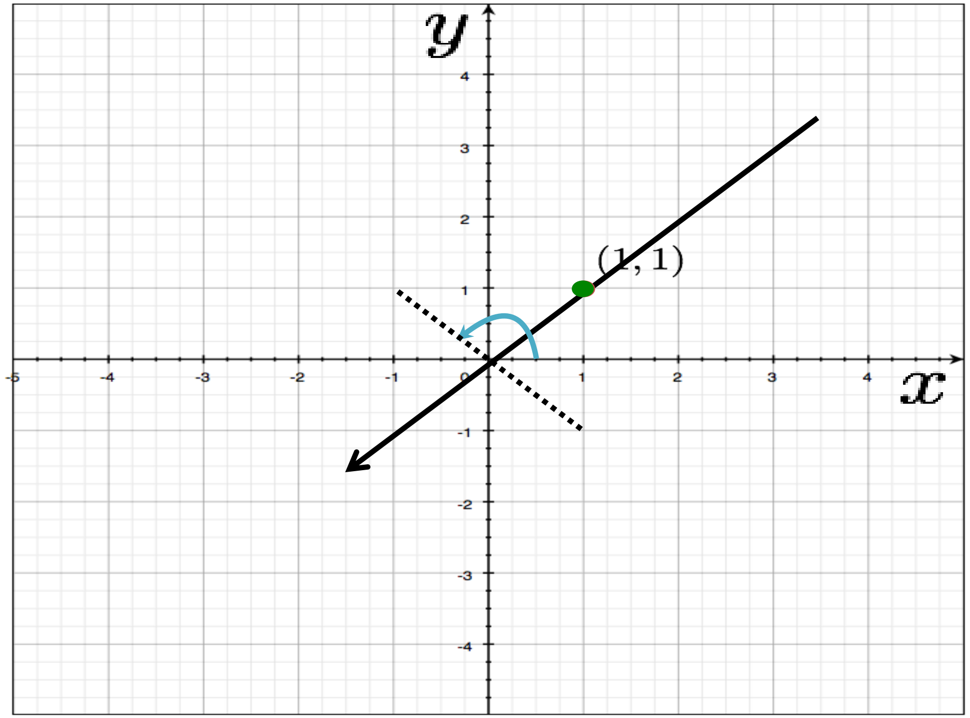
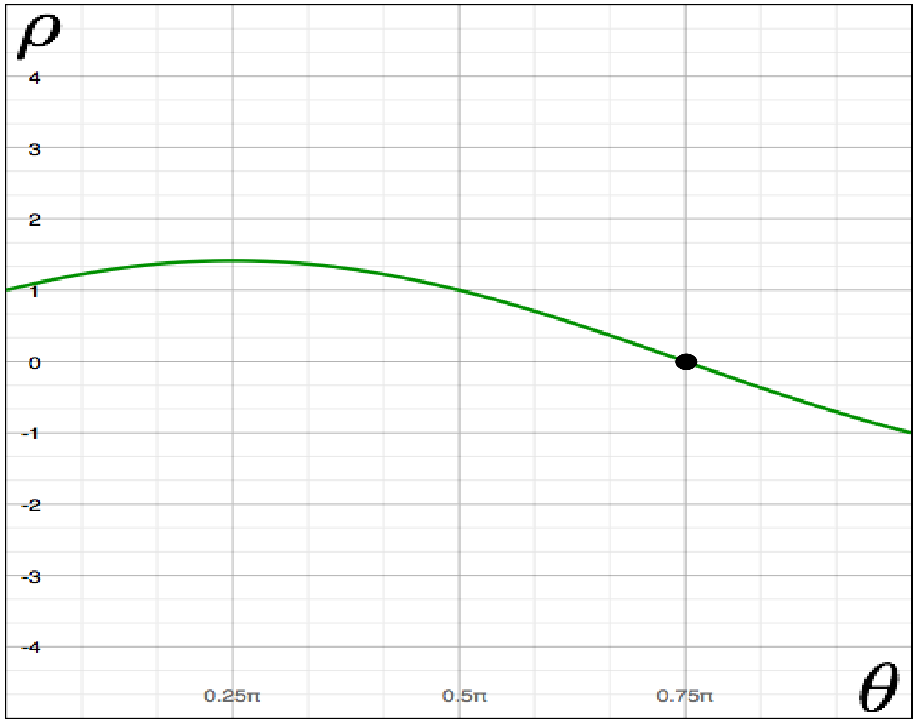


Image space

a line becomes a point



Parameter space

(ρ, θ) parameter space

variables

$$y = mx + b$$

parameters

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

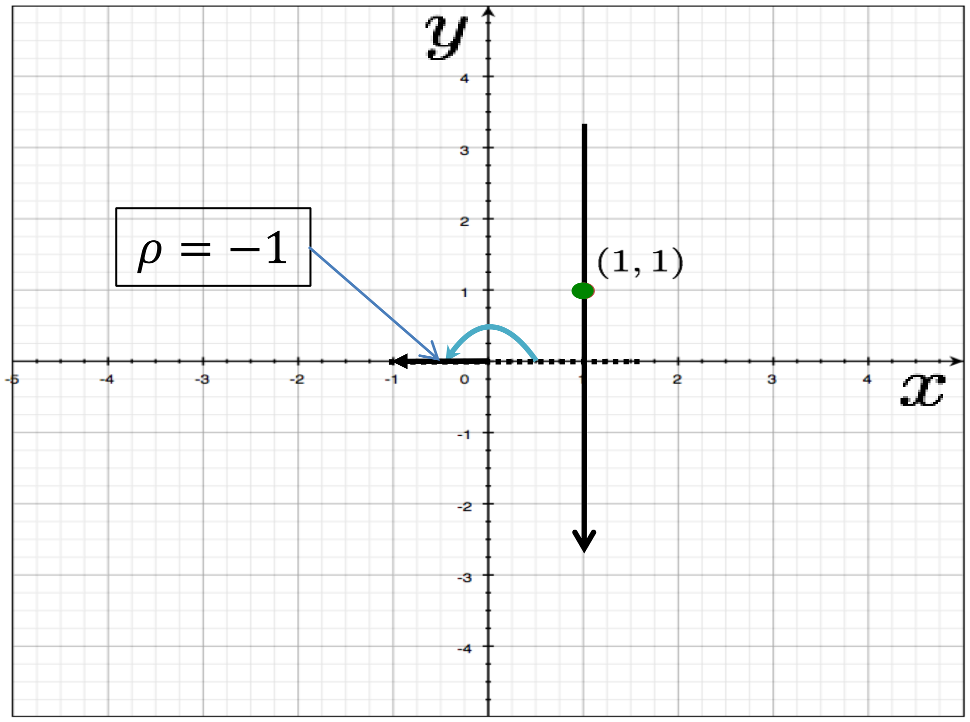
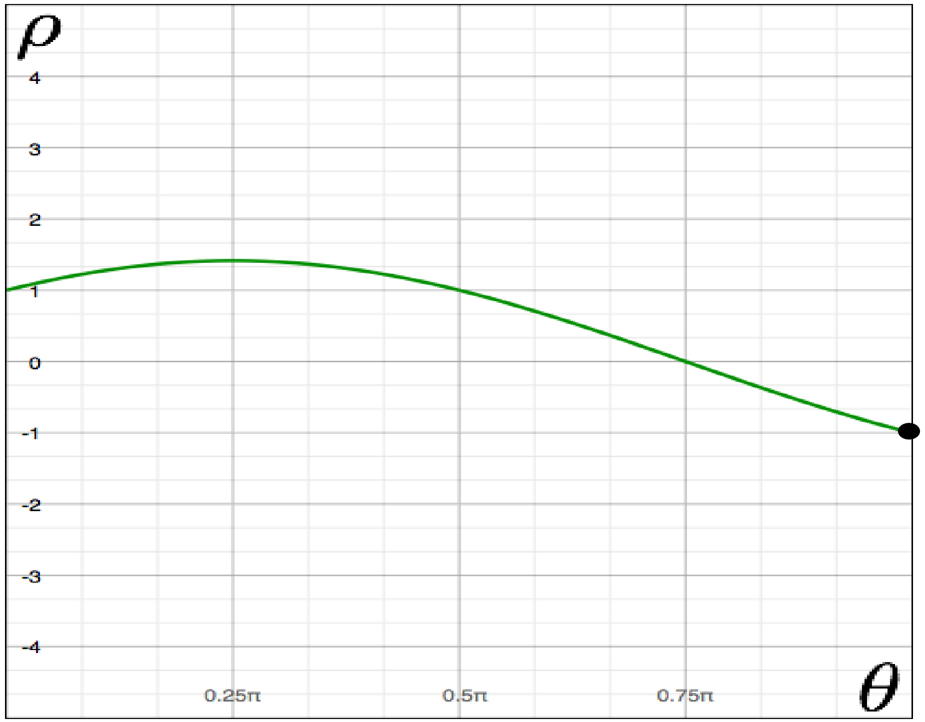


Image space

a line becomes a point



Parameter space

Notes on $(\boldsymbol{\rho}, \boldsymbol{\theta})$ parameter space

- We only care about $0 \leq \theta < \pi$, otherwise it's symmetric.
- As we so earlier, for some $\theta \rightarrow sign(\rho) = -1$. this is acceptable since the derivation earlier was right only for the first quadrant.

(ρ, θ) parameter space

variables

$$y = mx + b$$

parameters

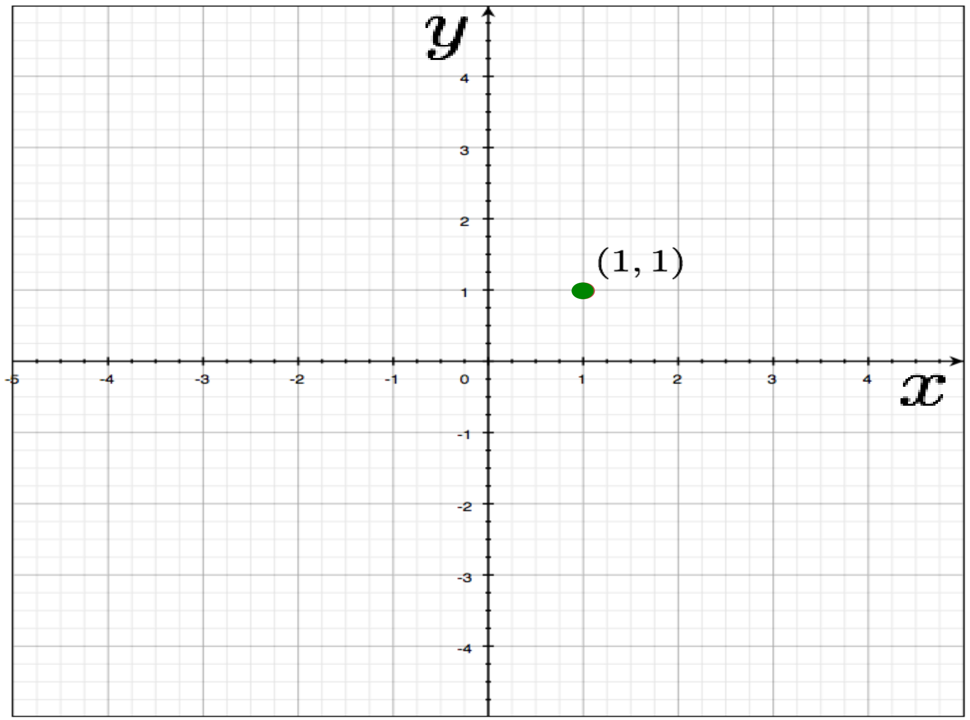
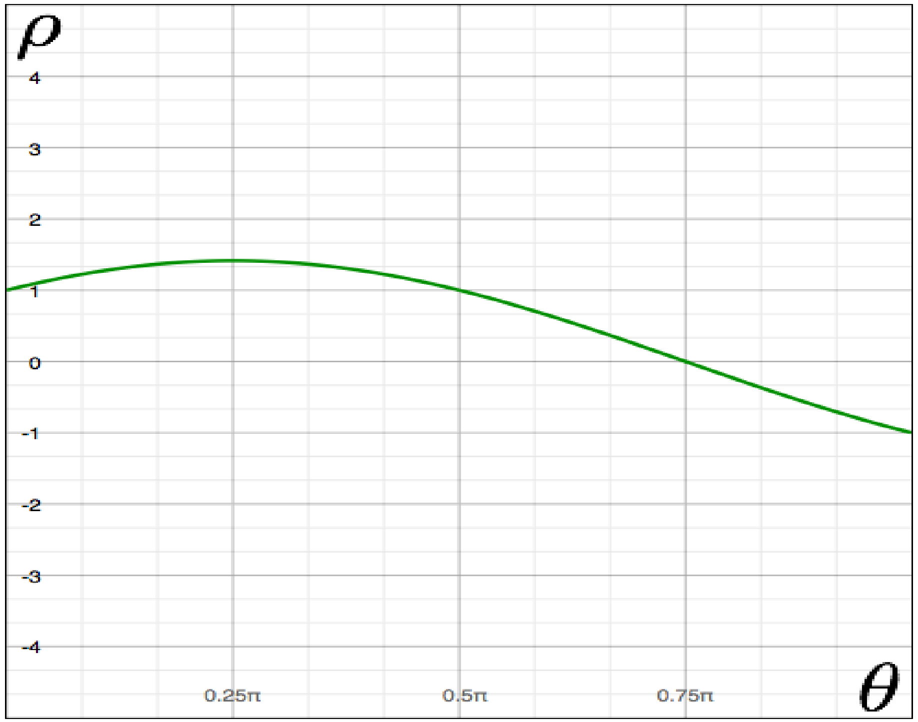


Image space

a point becomes a wave



Parameter space

(ρ, θ) parameter space

variables

$$y = mx + b$$

parameters

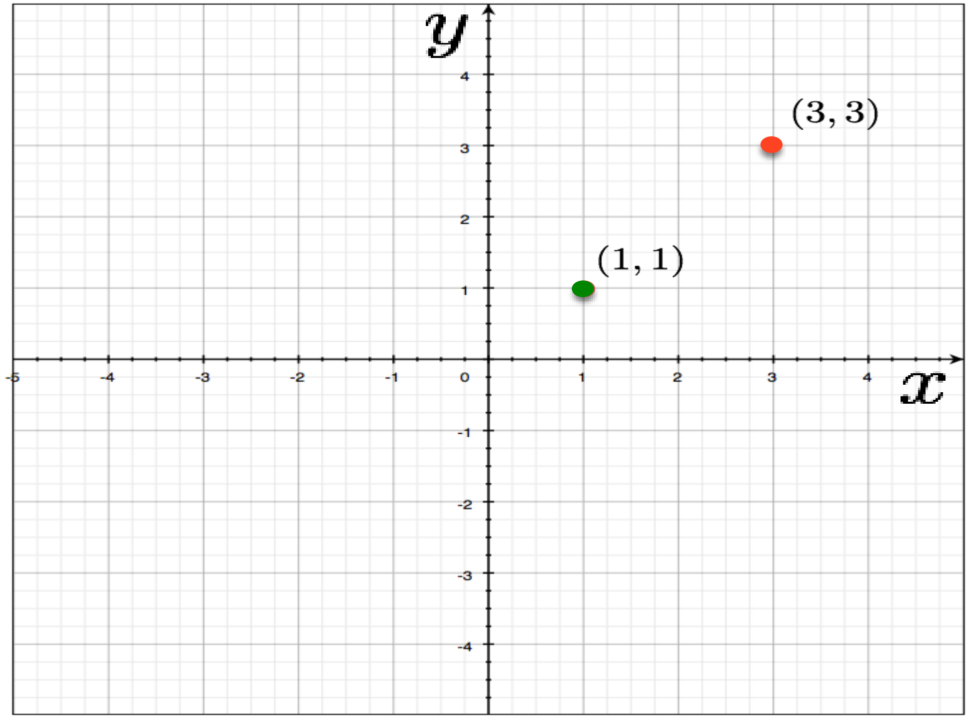
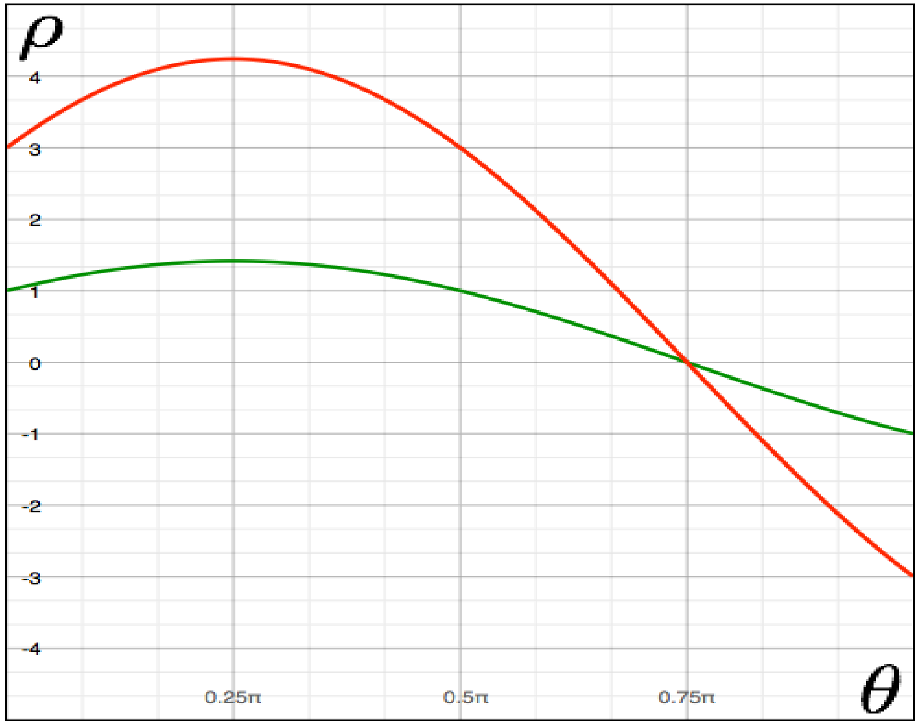


Image space



Parameter space

(ρ, θ) parameter space

variables

$$y = mx + b$$

parameters

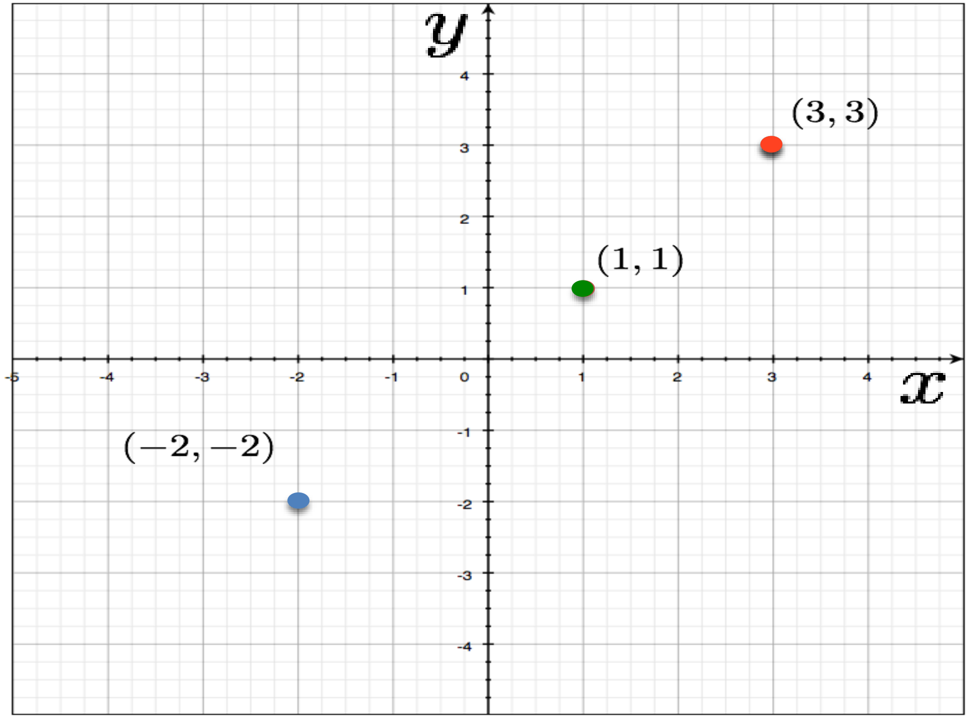
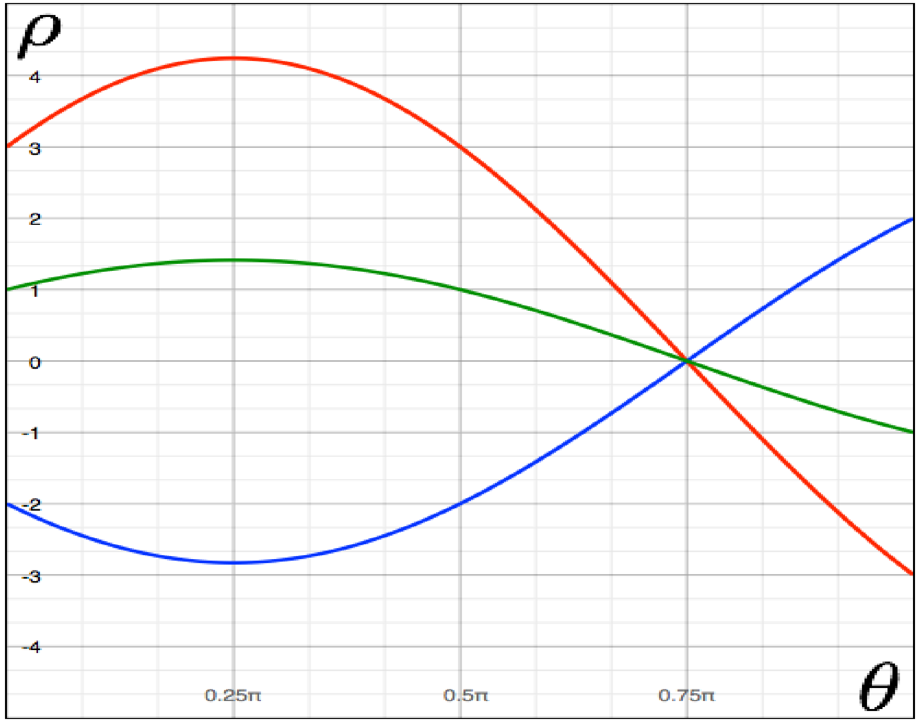


Image space



Parameter space

(ρ, θ) parameter space

variables

$$y = mx + b$$

parameters

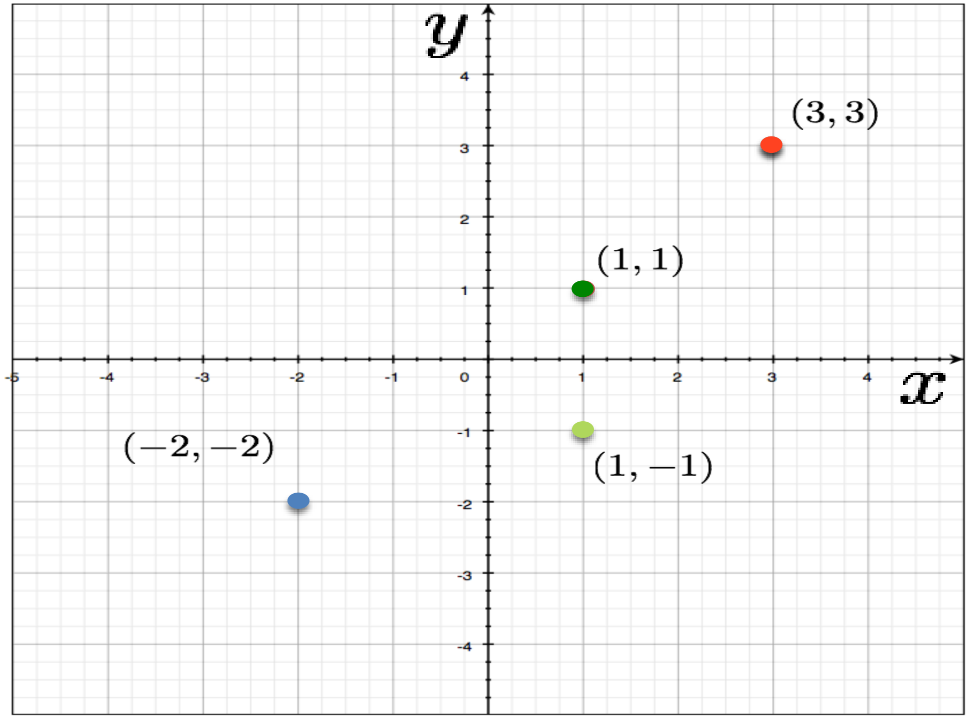
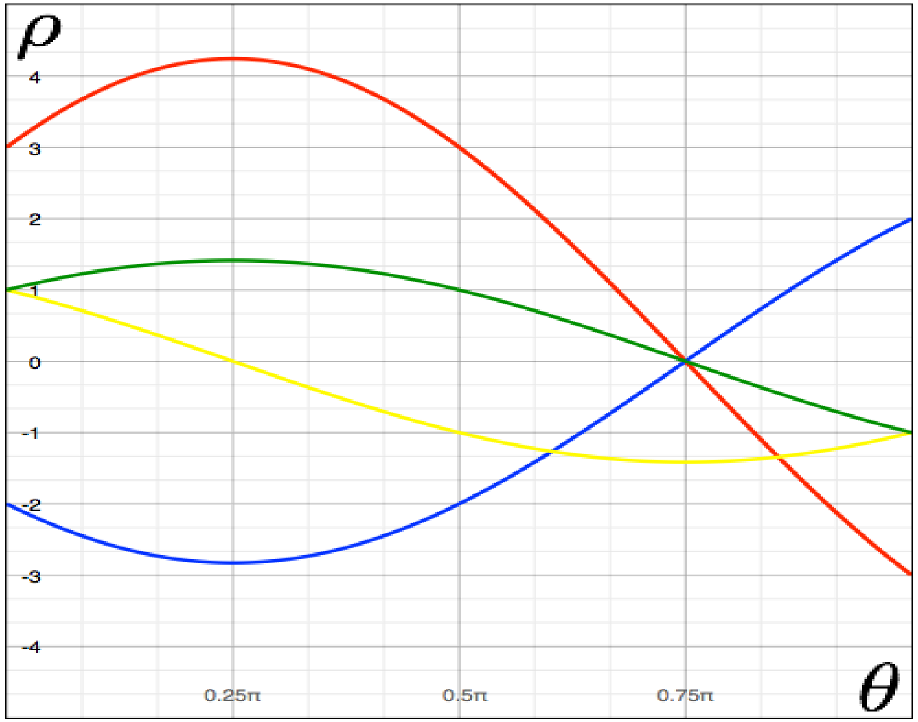


Image space



Parameter space

Hough transform algorithm stays the same!

```
Build accumulator table.
For each point in image space:
   find corresponding line in parameter space and
   increment +1 the intersecting bins.
Threshold the accumulator table result by some TH and get
the corresponding line parameters.
```

Hough transform- pros & cons

- Pros:
  - Can detect multiple lines in image space.
  - can be extended to detect different parameterized curves (e.g.: circles, ellipsoids), and even un-parameterized curves (**generalized hough transform** [out of scope]- similar to template matching that will be covered later in course).
- Cons:
  - ~~For the shown~~ $\cancel{(m, b)}$ ~~parameter space, can't detect vertical lines.~~ ~~**<u>Why?</u>**~~
  - Susceptive to noise. **<u>Why?</u>**
  - Computationally costly.

Hough transform noise

- In case that the discovered edge is noisy, the binning process in the accumulation matrix can be problematic.
- This is a known problem of Hough transform… some ways to get better results is to try:
  - Different bin size (different step size for R & θ).
  - Smooth the accumulation matrix before thresholding.