CPSC 425: Computer Vision

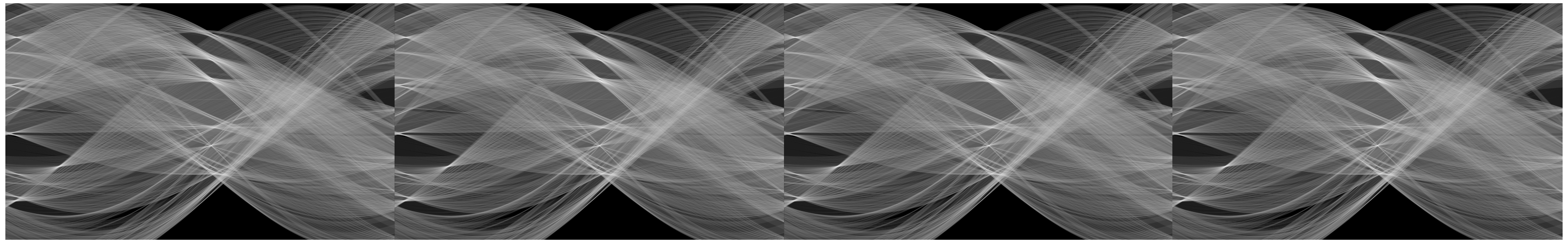


Image Credit: Ioannis (Yannis) Gkioulekas (CMU)

Lecture 17: RANSAC cont., Hough Transform

Menu for Today (**March 7, 2019**)

Topics:

— RANSCA continued

— Hough Transform

Redings:

— **Today's** Lecture: Forsyth & Ponce (2nd ed.) 10.1, 10.2

— **Next** Lecture: Forsyth & Ponce (2nd ed.) 7.1.1, 7.2.1, 7.4, 7.6

Reminders:

— **Assignment 4**: Texture Syntheis is **out,** due on **March 19th**

— Midterms **graded** (will post on Piazza when available)

(**Average**: 63%; there will be a curve ~10% but is not fixed yet)

Today's "**fun**" Example: Everybody Dance Now

Lecture 16: Re-cap

RANSAC is a technique to fit data to a model

— divide data into inliers and outliers

— estimate model from minimal set of inliers

— improve model estimate using all inliers

— alternate fitting with re-classification as inlier/outlier

RANSAC is a general method suited for a wide range of model fitting problems

— easy to implement

— easy to estimate/control failure rate

RANSAC only handles a moderate percentage of outliers without cost blowing up

RANSAC: *k* Samples Chosen (p = 0.99)

Sample size	Proportion of outliers						
n	**5%**	**10%**	**20%**	**25%**	**30%**	**40%**	**50%**
2	2	3	5	6	7	11	17
3	3	4	7	9	11	19	35
4	3	5	9	13	17	34	72
5	4	6	12	17	26	57	146
6	4	7	16	24	37	97	293
7	4	8	20	33	54	163	588
8	5	9	26	44	78	272	1177

Figure Credit: Hartley & Zisserman

After RANSAC

RANSAC divides data into inliers and outliers and yields estimate computed from minimal set of inliers

Improve this initial estimate with estimation over all inliers (e.g., with standard least-squares minimization)

But this may change inliers, so alternate fitting with re-classification as inlier/outlier

Example 2: Fitting a Line

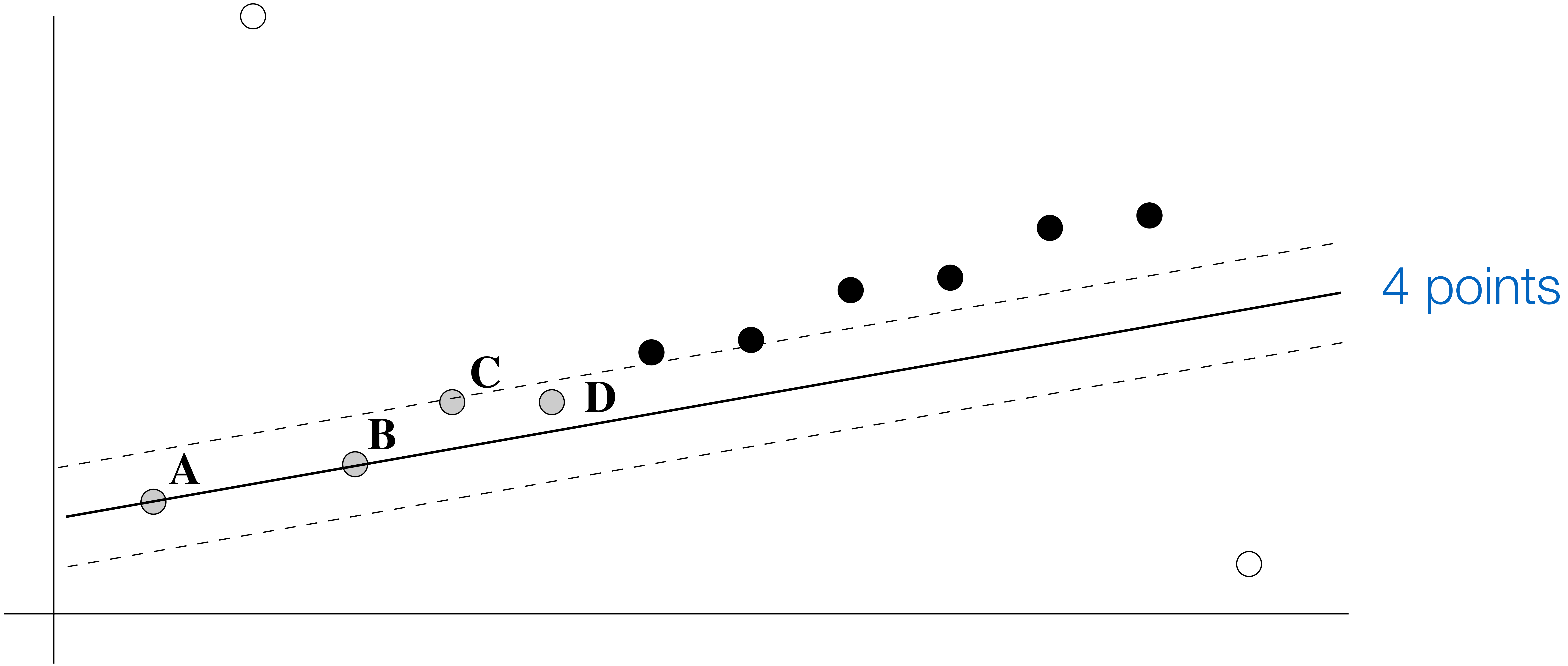


Figure Credit: Hartley & Zisserman

Example 2: Fitting a Line

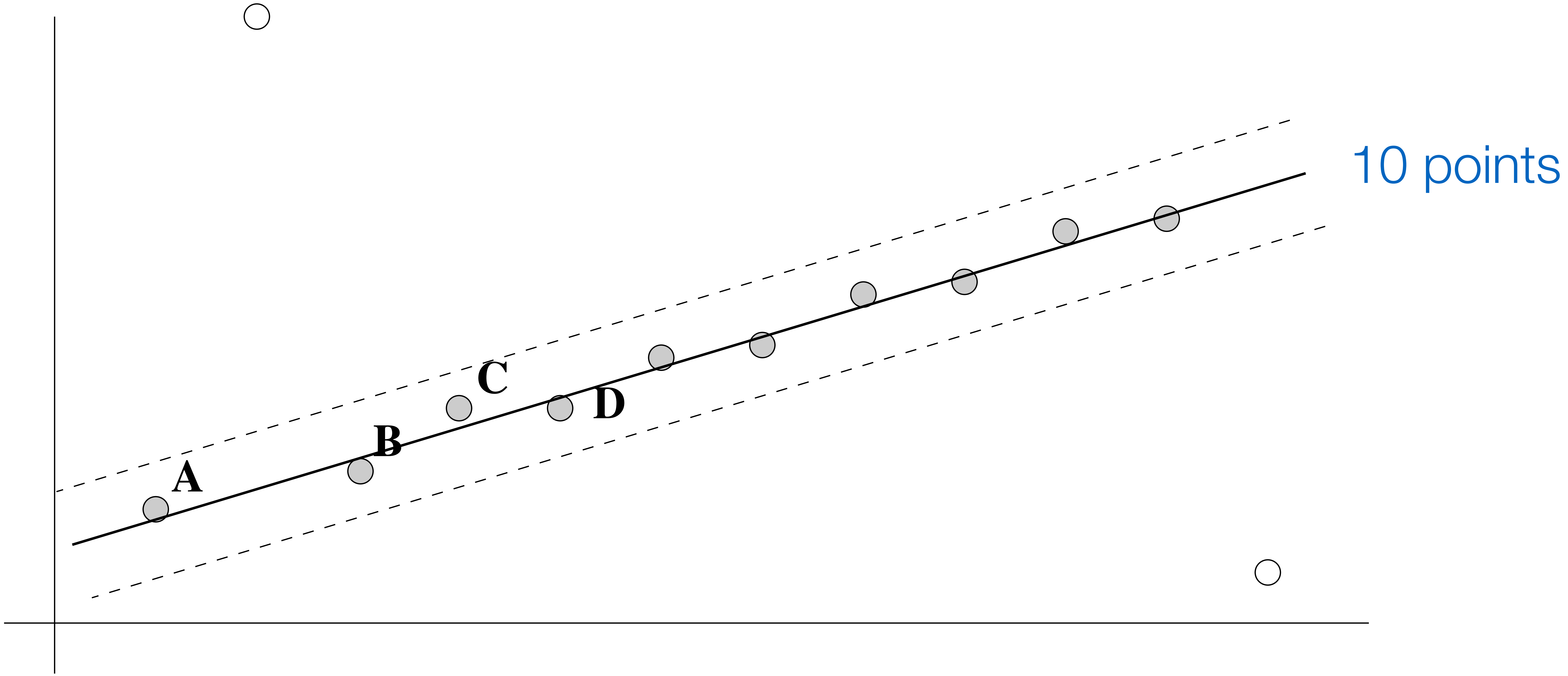


Figure Credit: Hartley & Zisserman

Example 3: Automatic Matching of Images

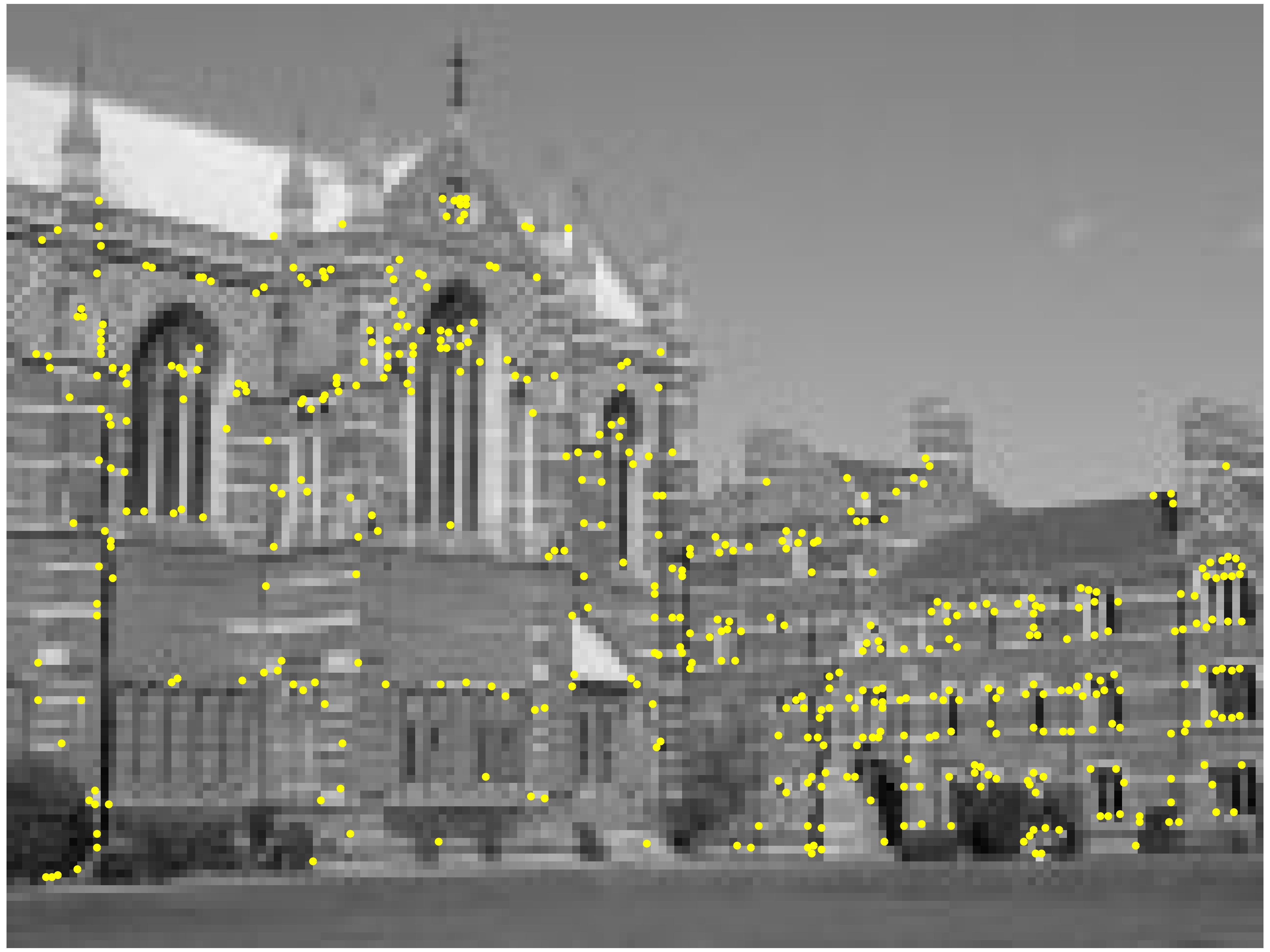
- How to get correct correspondences without human intervention?
- Can be used for image stitching or automatic determination of epipolar geometry

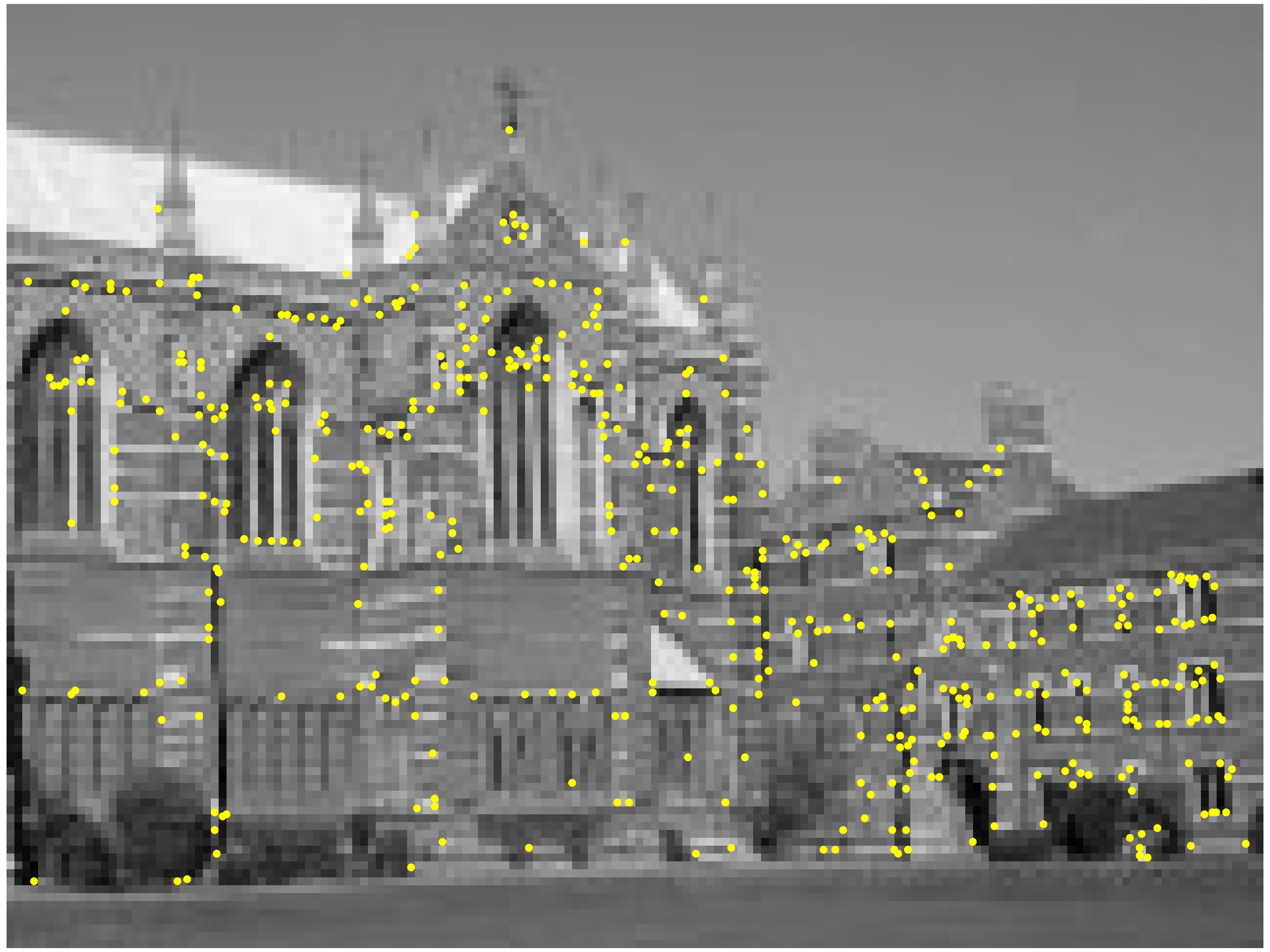
Figure Credit: Hartley & Zisserman

Example 3: Feature Extraction

— Find features in pair of images using Harris corner detector

— Assumes images are roughly the same scale



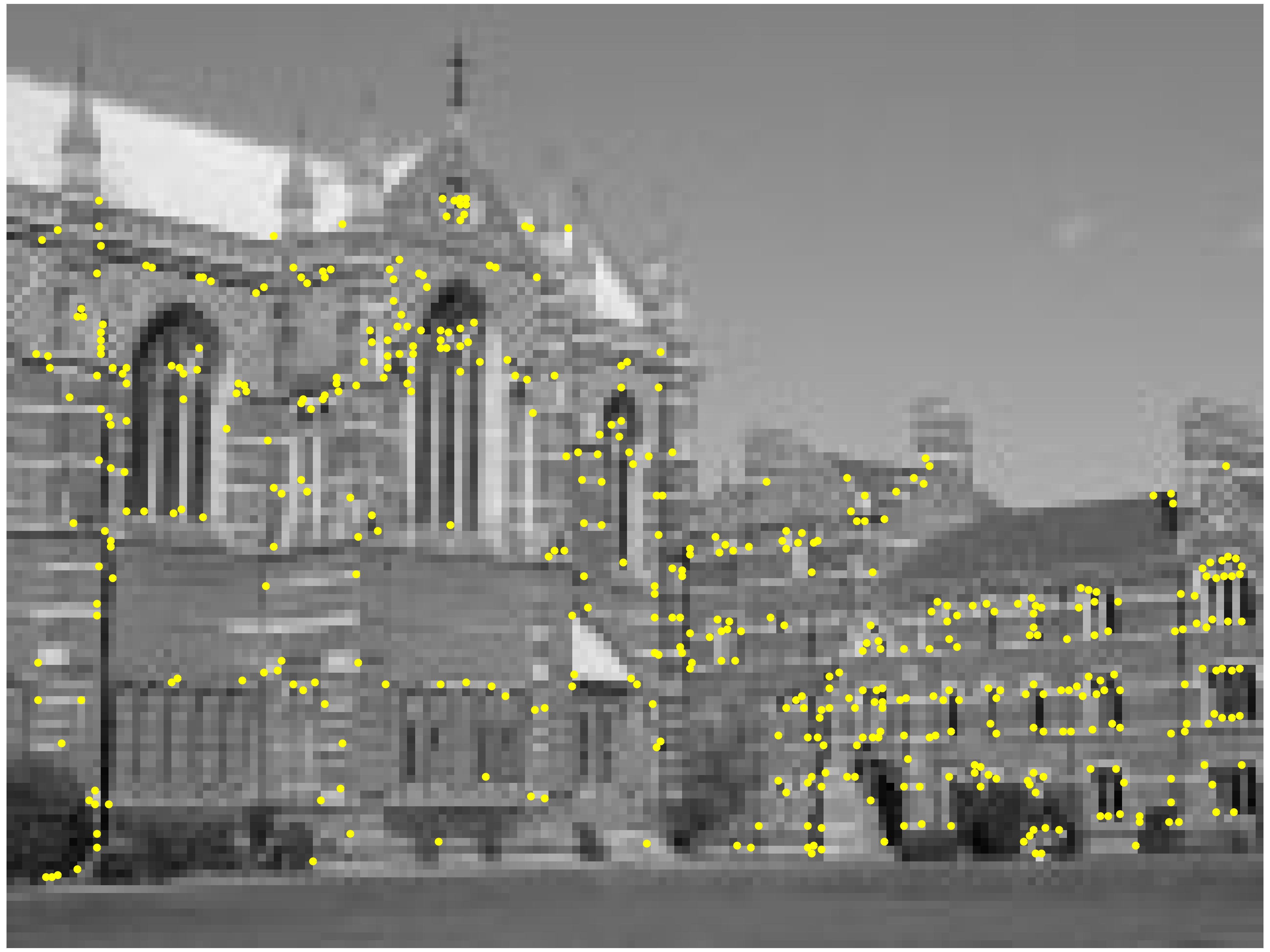


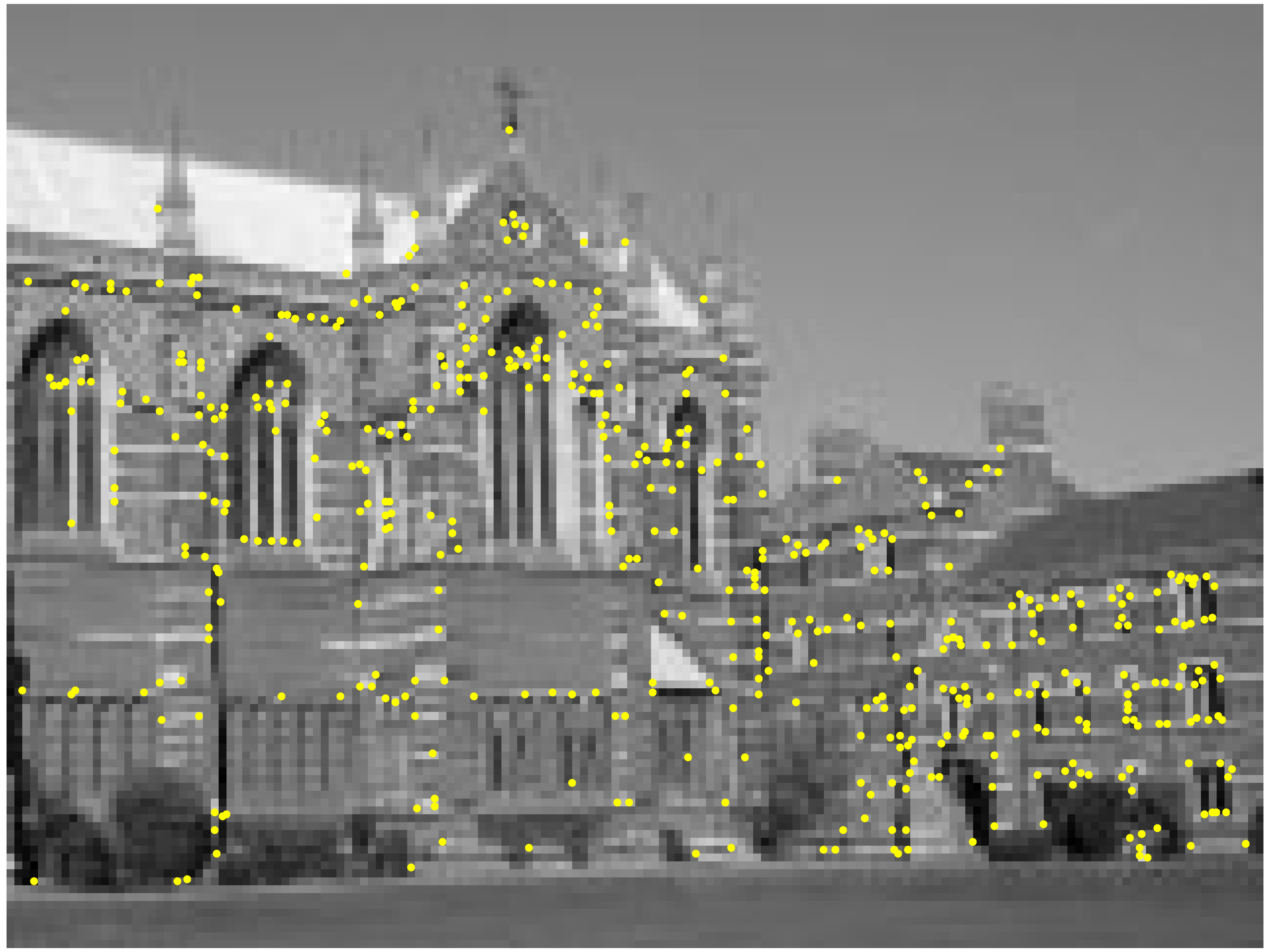
≈ 500 corner features found in each image

Figure Credit: Hartley & Zisserman

Example 3: Finding Feature Matches

Select best match over threshold within a square search window (here ±320 pixels) using SSD or (normalized) cross-correlation for small patch around the corner





≈ 500 corner features found in each image

Figure Credit: Hartley & Zisserman

Example 3: Initial Match Hypothesis

268 matched features (over SSD threshold) superimposed on left image (pointing to locations of corresponding feature in right image)

Figure Credit: Hartley & Zisserman

Example 3: Outliers & Inliers after RANSAC

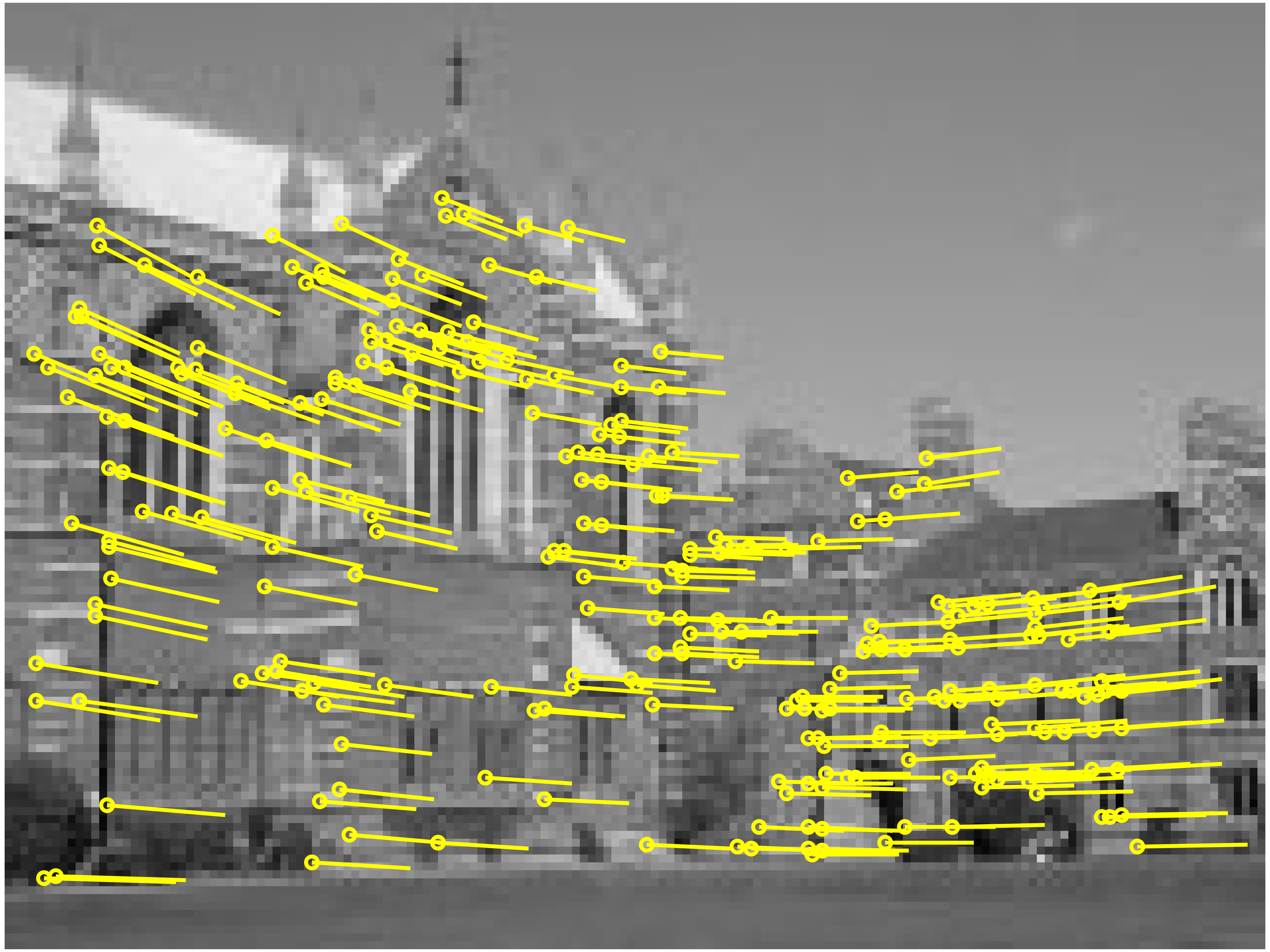
- *n* is 4 for this problem (a homography relating 2 images)
- Assume up to 50% outliers
- 43 samples used with t = 1.25 pixels

117 **outliers**

151 **inliers**

Figure Credit: Hartley & Zisserman

Example 3: Final Matches



final set of 262 matches

Figure Credit: Hartley & Zisserman

Discussion of RANSAC

Advantages:

— General method suited for a wide range of model fitting problems

— Easy to implement and easy to calculate its failure rate

Disadvantages:

— Only handles a moderate percentage of outliers without cost blowing up

— Many real problems have high rate of outliers (but sometimes selective choice of random subsets can help)

The Hough transform can handle high percentage of outliers

Example: Photo Tourism

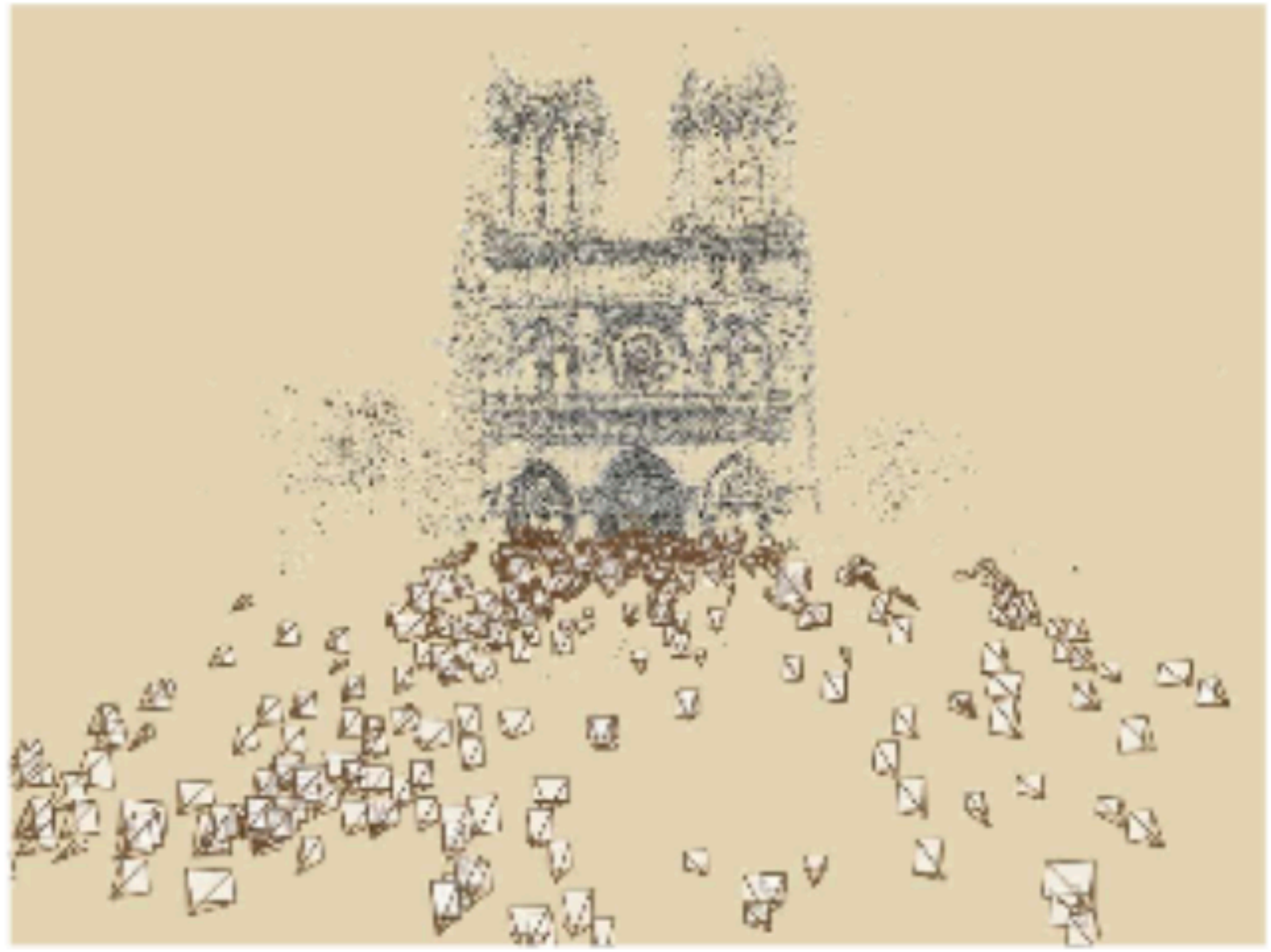


Figure credit: Snavely et al. 2006

Takes as input unstructured collections of photographs and reconstructs each photo's viewpoint and a sparse 3D model of the scene

Uses both SIFT and RANSAC

Fitting a Model

Suppose we want to fit a model to a set of tokens

— e.g. A line fits well to a set of points. This is unlikely to be due to chance, so we represent the points as a line.

— e.g. A 3D model can be scaled, rotated and translated to closely fit a set of points or line segments. If it fits well, the object is recognized.

Fitting a Model is Difficult

Difficulties arise owing to:

Extraneous data: clutter or multiple models

— We do not know what is part of the model

— Can we fit models with a few parts when there is significant background clutter?

Missing data: only some parts of model are present Noise

Computational cost:

— Not feasible to check all combinations of features by fitting a model to each possible subset

Hough Transform

Idea of **Hough transform**:

— For each token vote for all models to which the token could belong

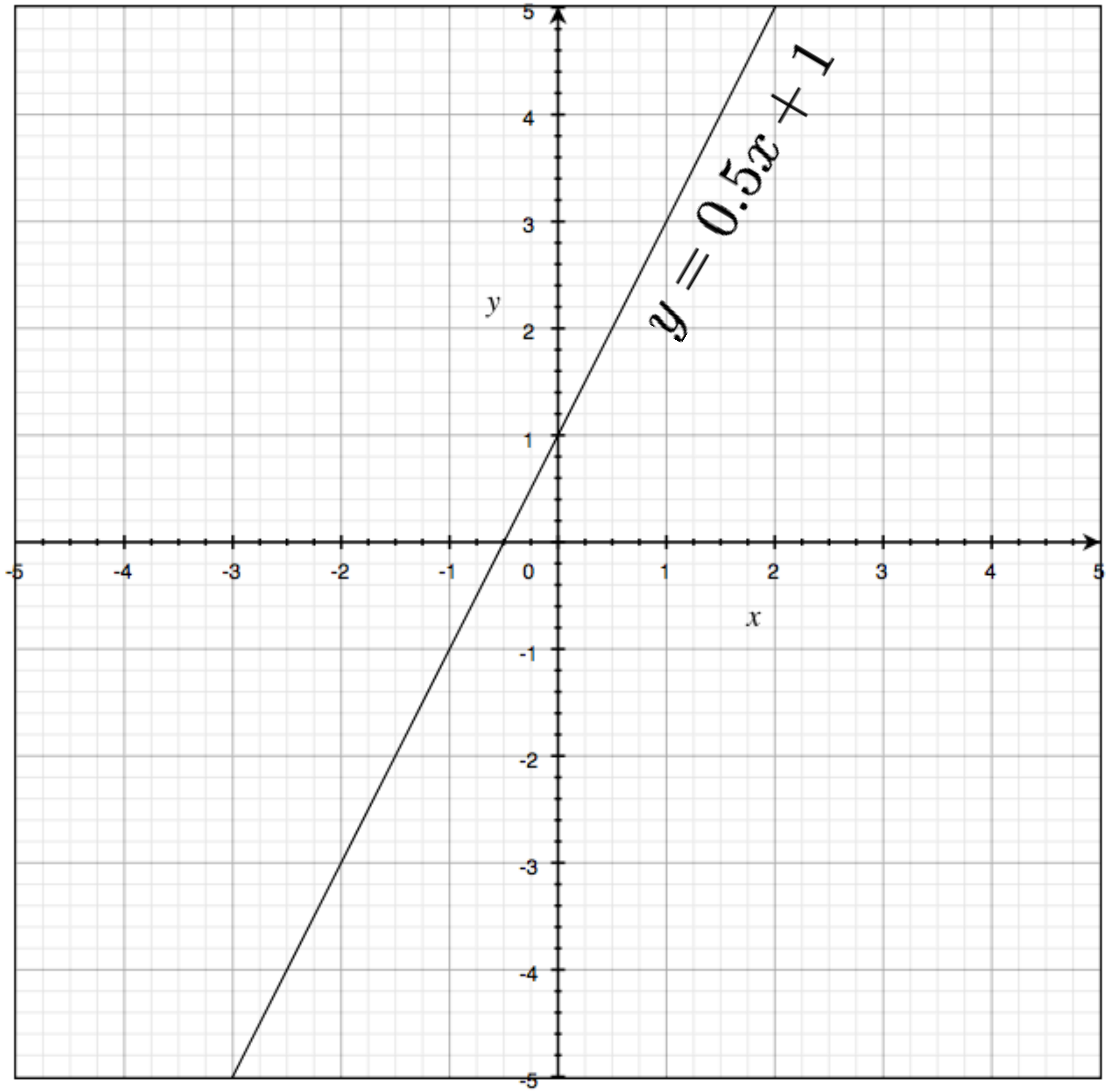
— Return models that get many votes

Example: For each point, vote for all lines that could pass through it; the true lines will pass through many points and so receive many votes

Lines: Slope intercept form

$$y = mx + b$$

slope y-intercept



Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Hough Transform: Image and Parameter Space

variables

$$y = mx + b$$

parameters

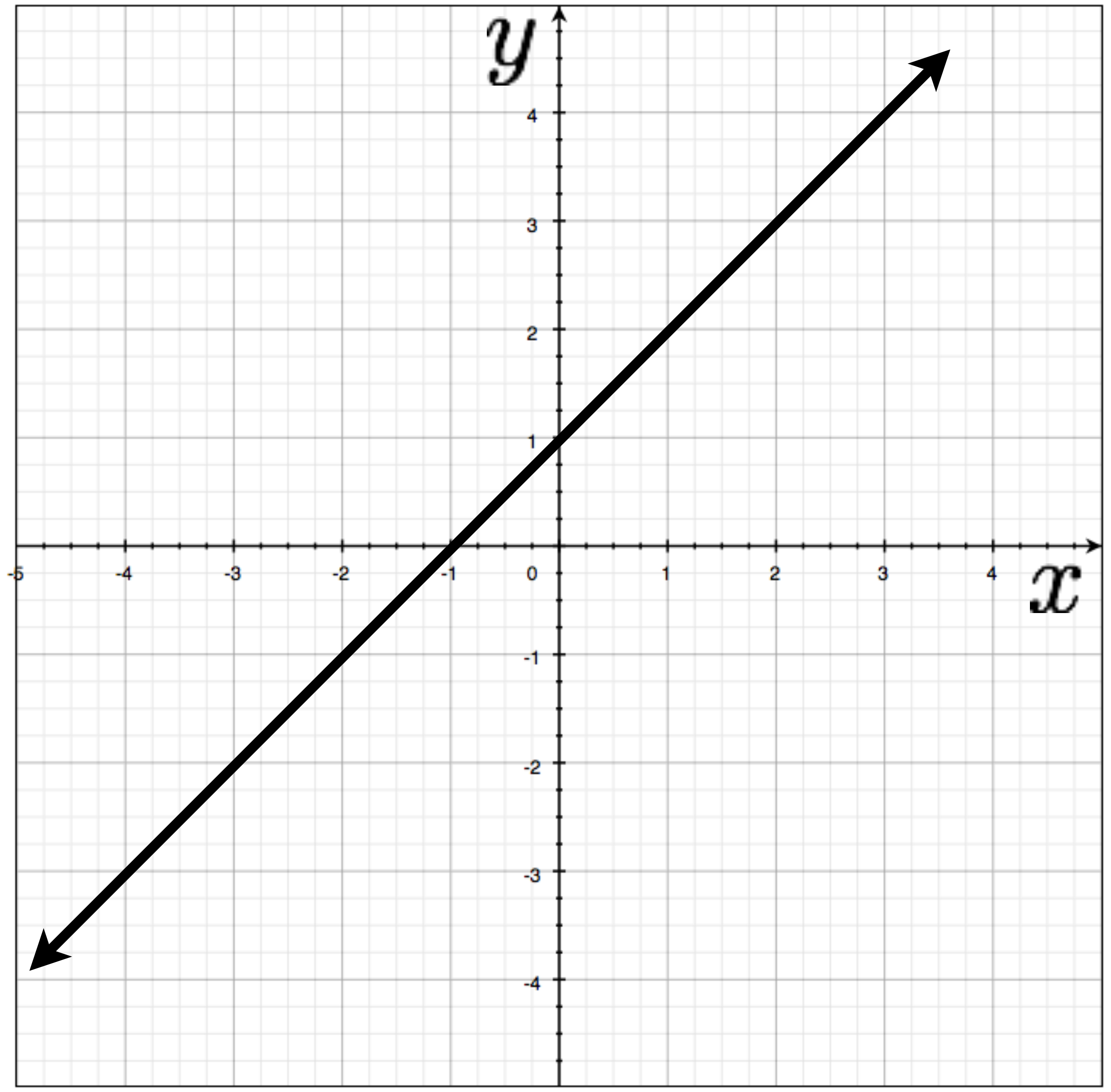


Image space

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Hough Transform: Image and Parameter Space

variables

$$y = mx + b$$

parameters

variables

$$y - mx = b$$

parameters

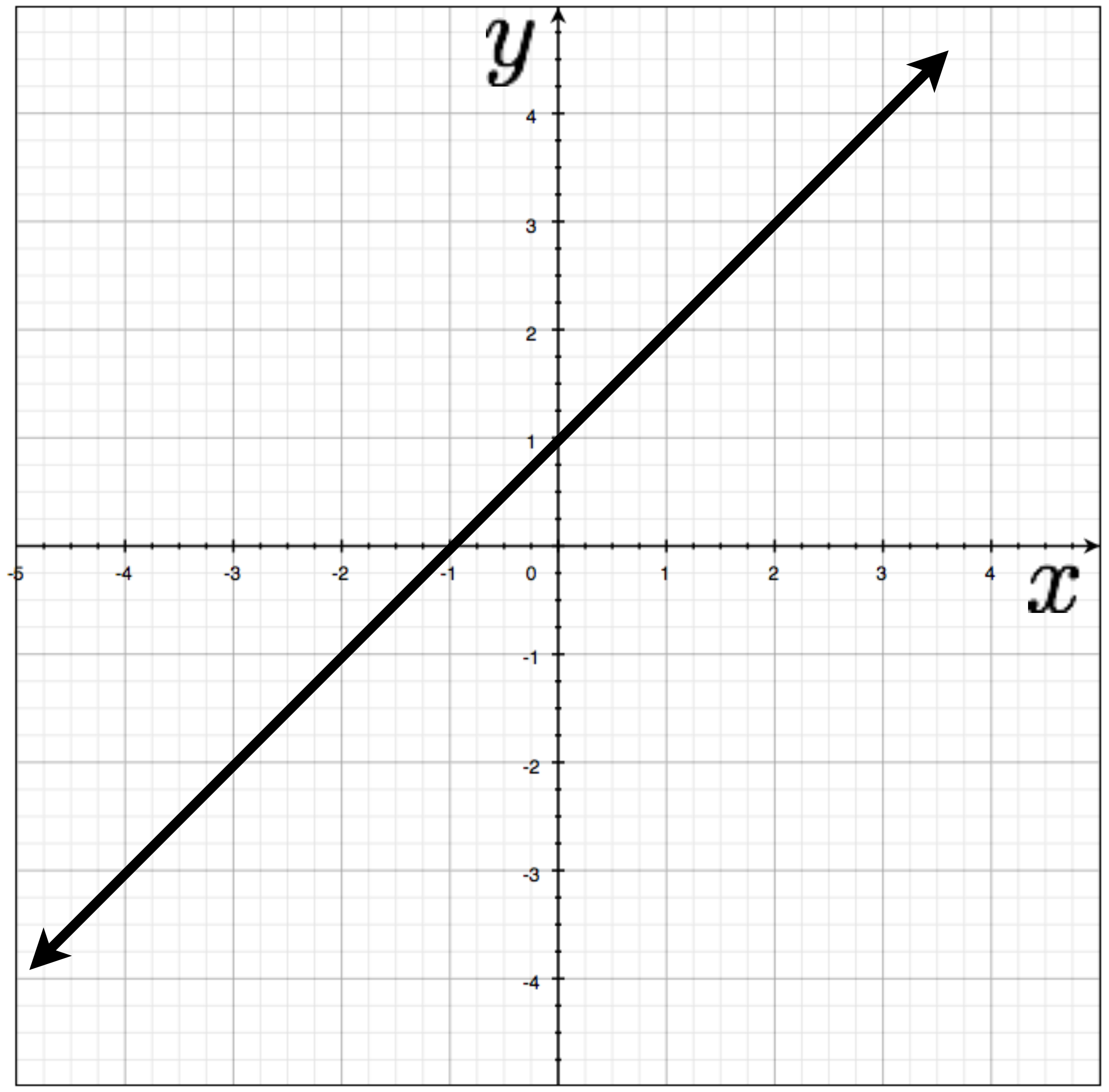
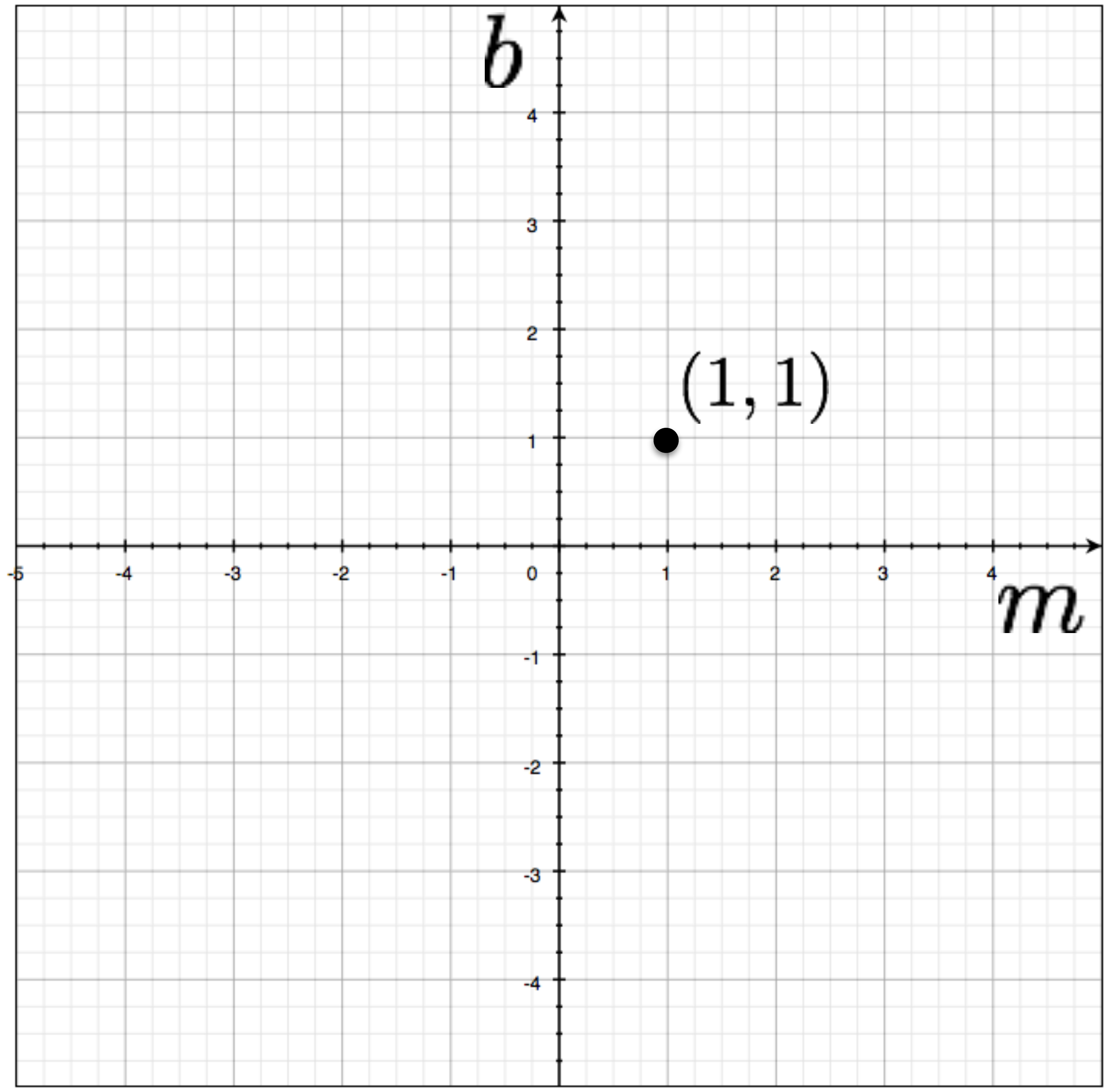


Image space

a line
becomes a
point



Parameter space

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Hough Transform: Image and Parameter Space

variables

$$y = mx + b$$

parameters

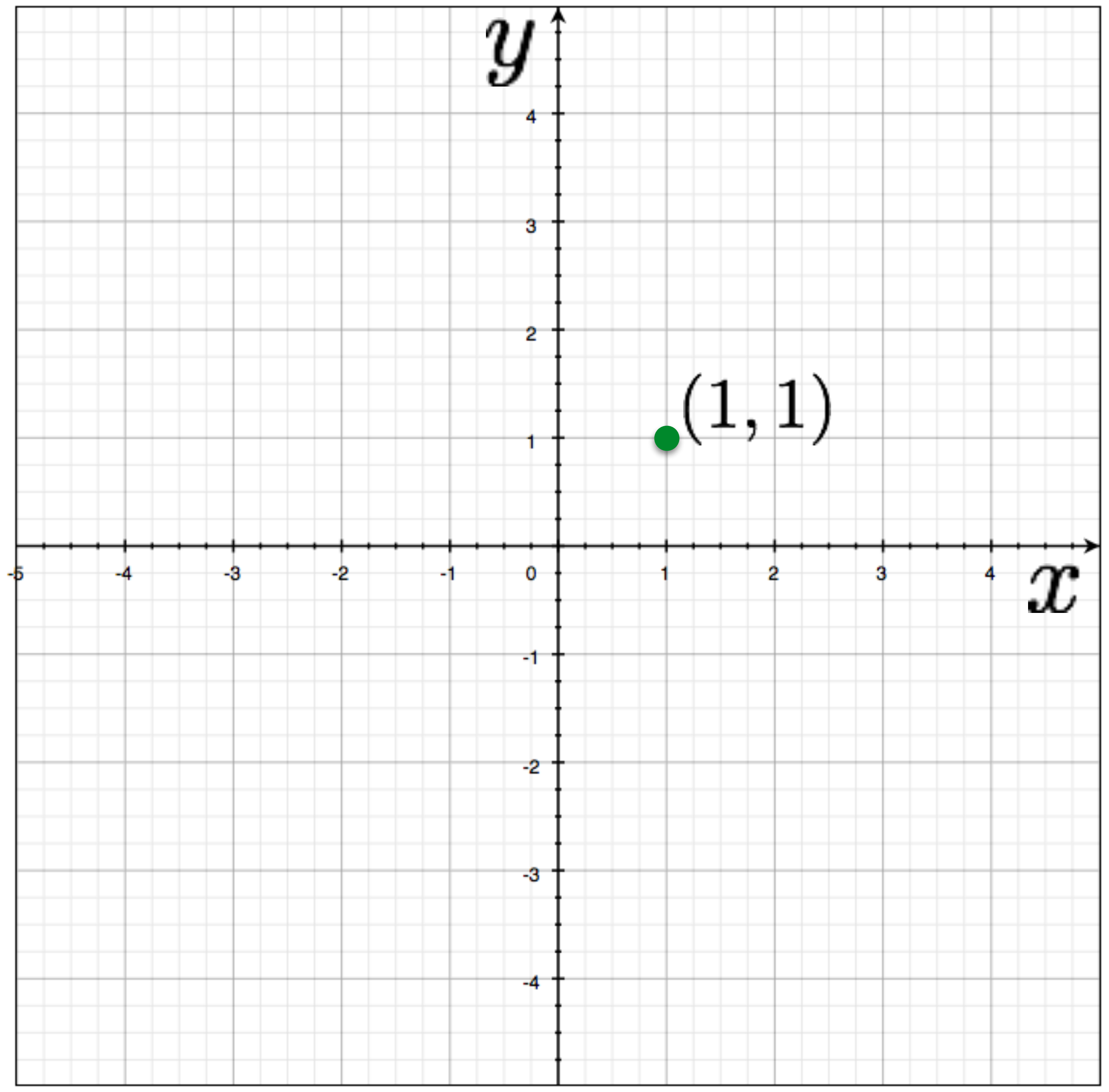


Image space

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

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Hough Transform: Lines

variables

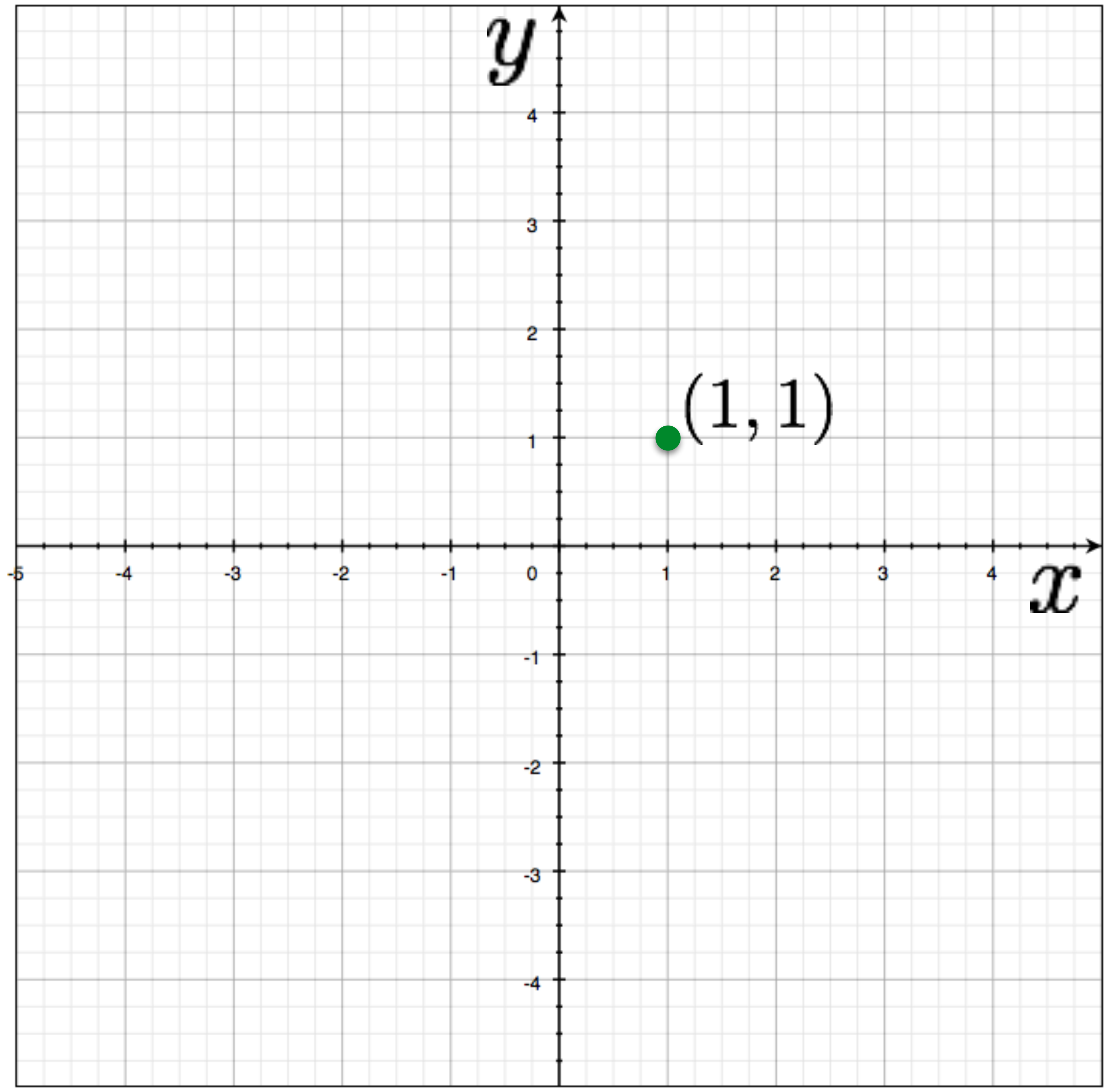
$$y = mx + b$$

parameters

variables

$$y - mx = b$$

parameters



a point becomes a line

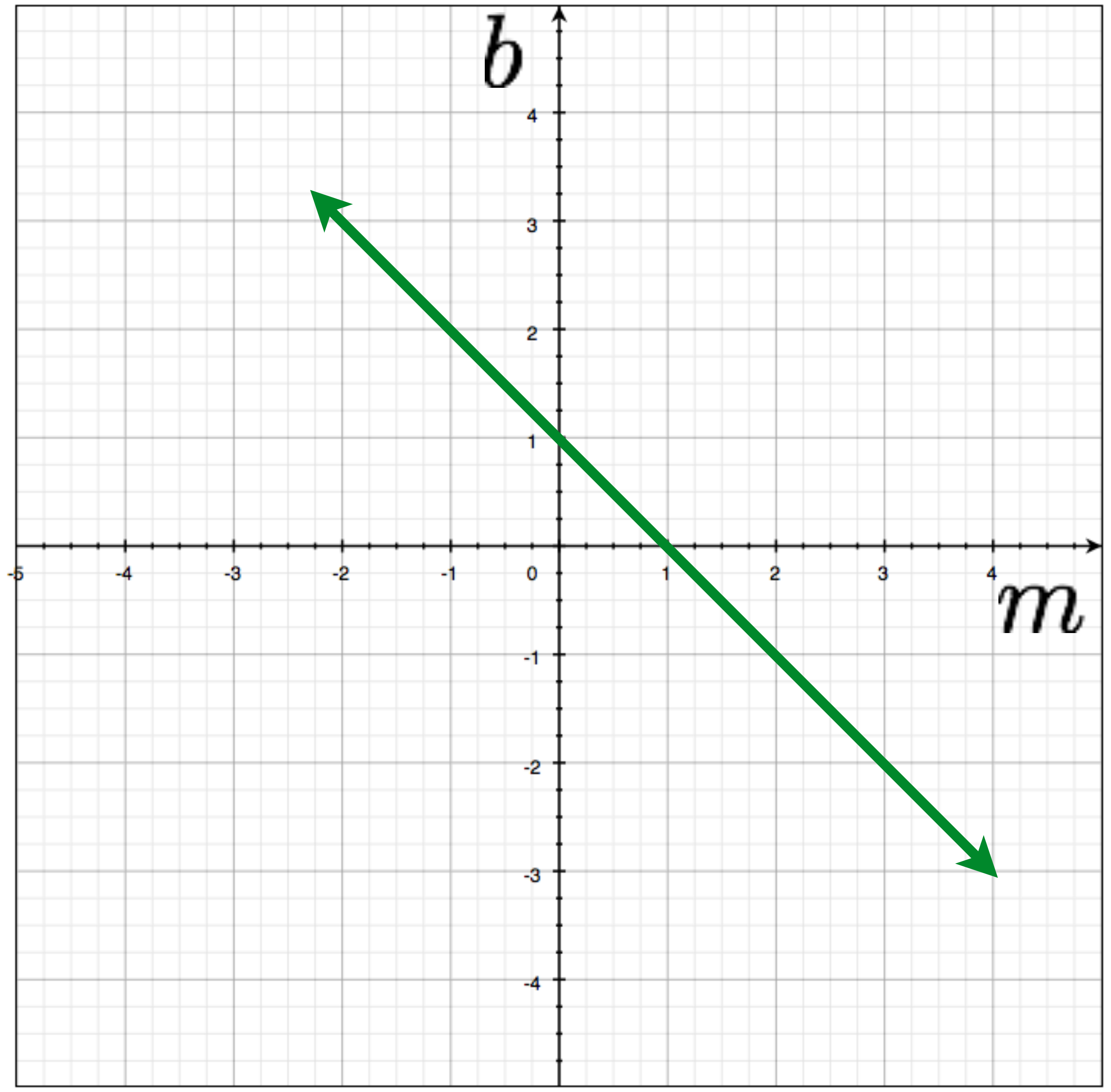


Image space

Parameter space

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Hough Transform: Lines

variables

$$y = mx + b$$

parameters

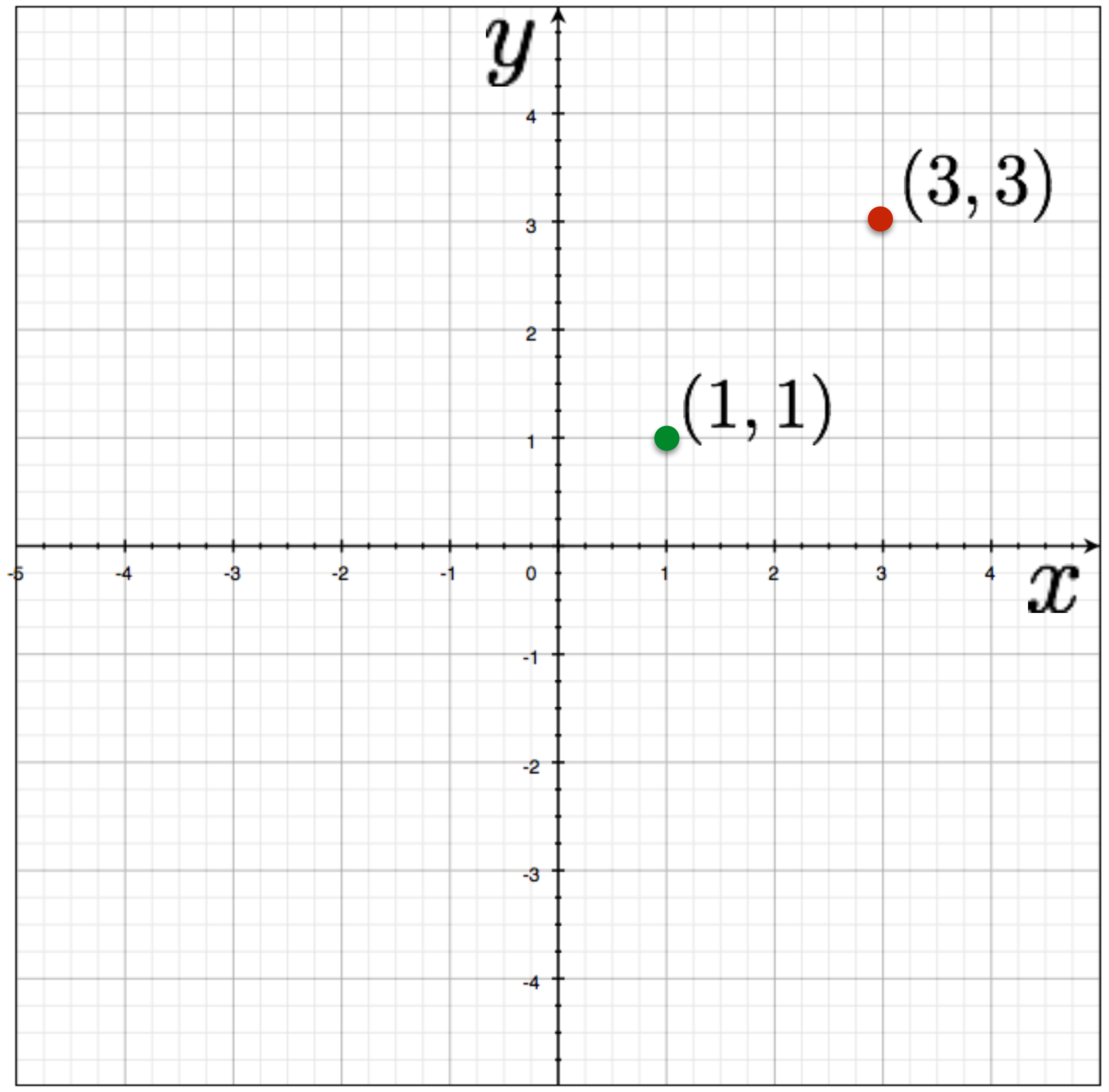


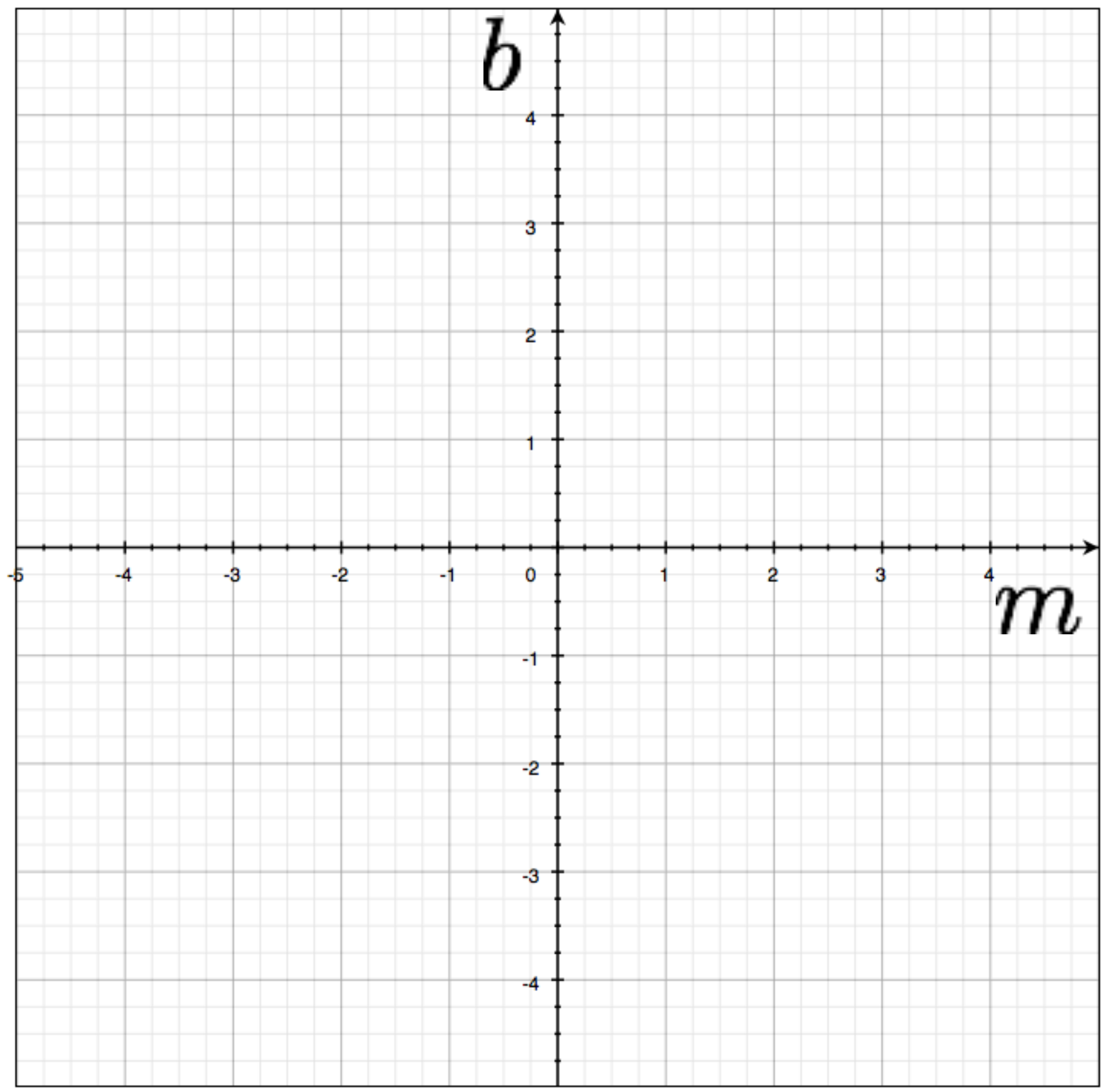
Image space

two points?

variables

$$y - mx = b$$

parameters



Parameter space

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Hough Transform: Lines

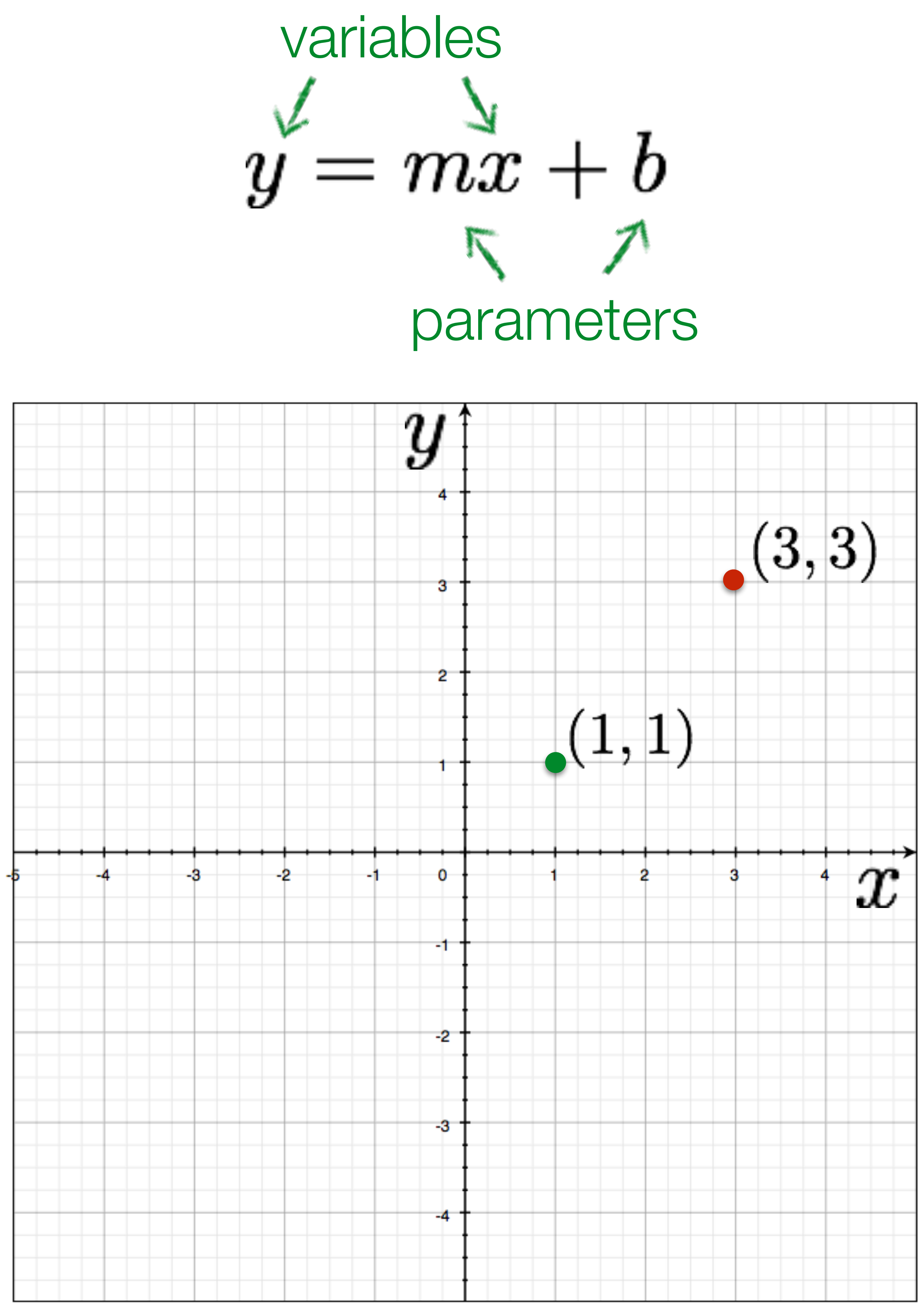
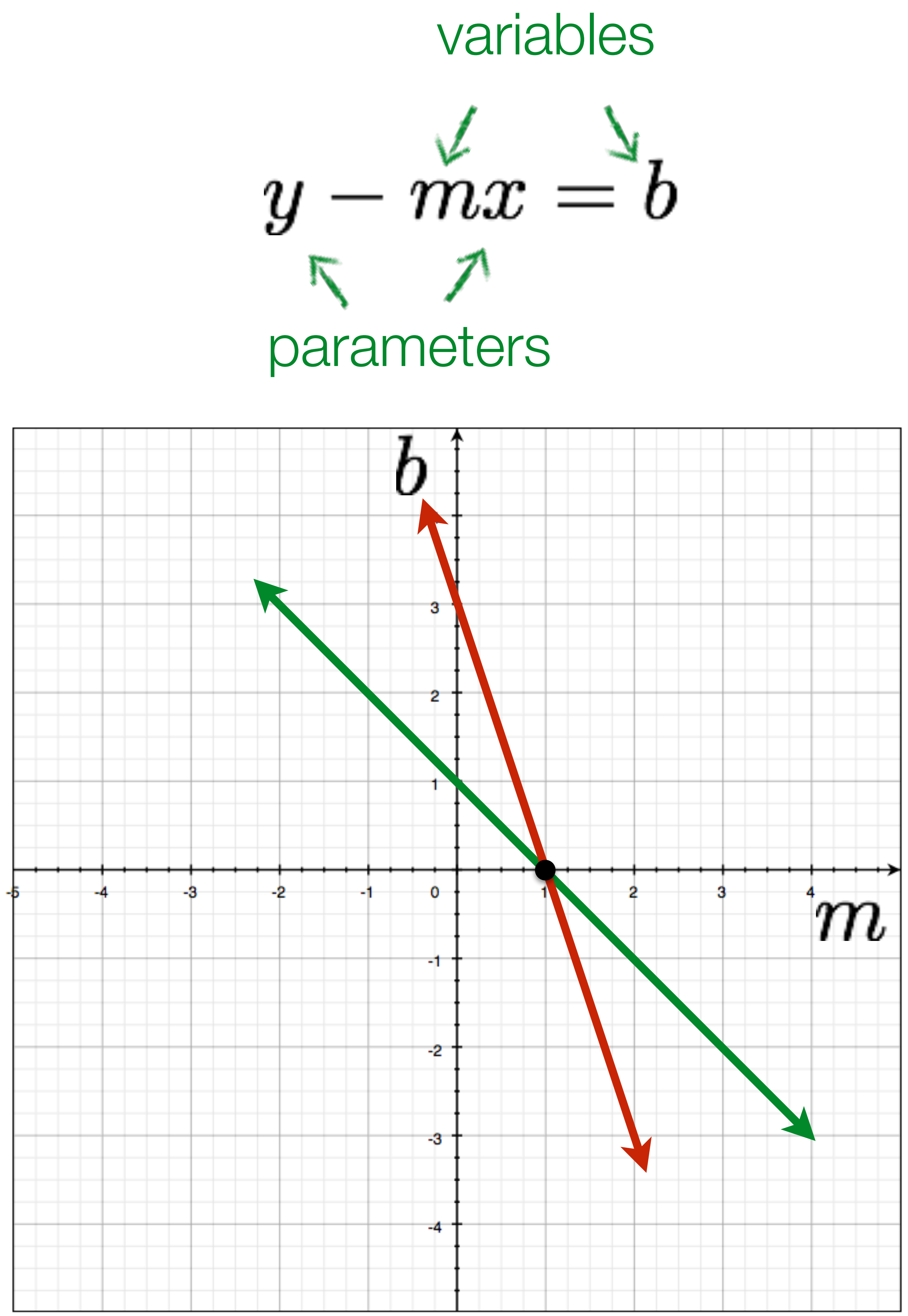


Image space



Parameter space

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Hough Transform: Lines

variables

$$y = mx + b$$

parameters

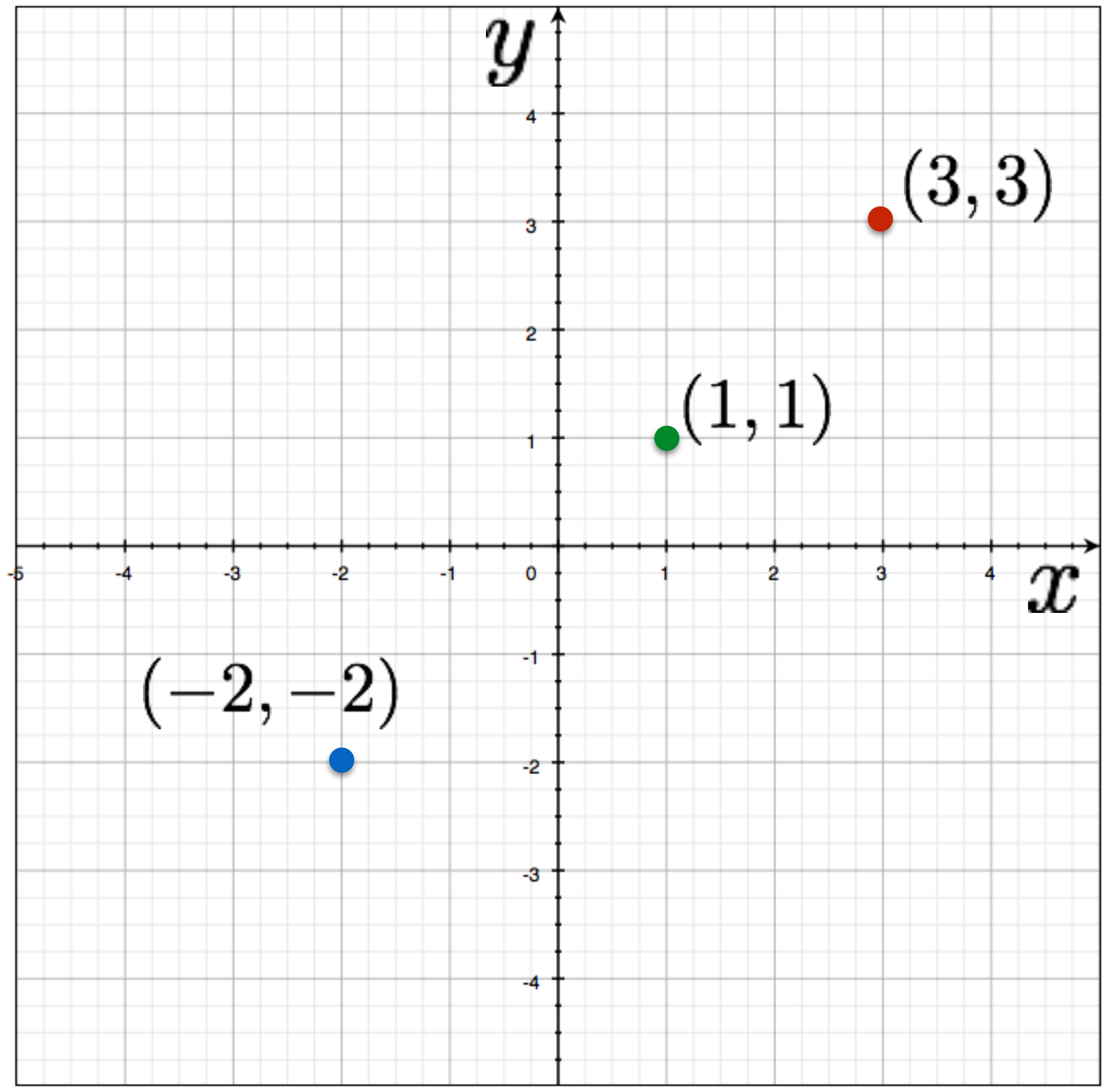


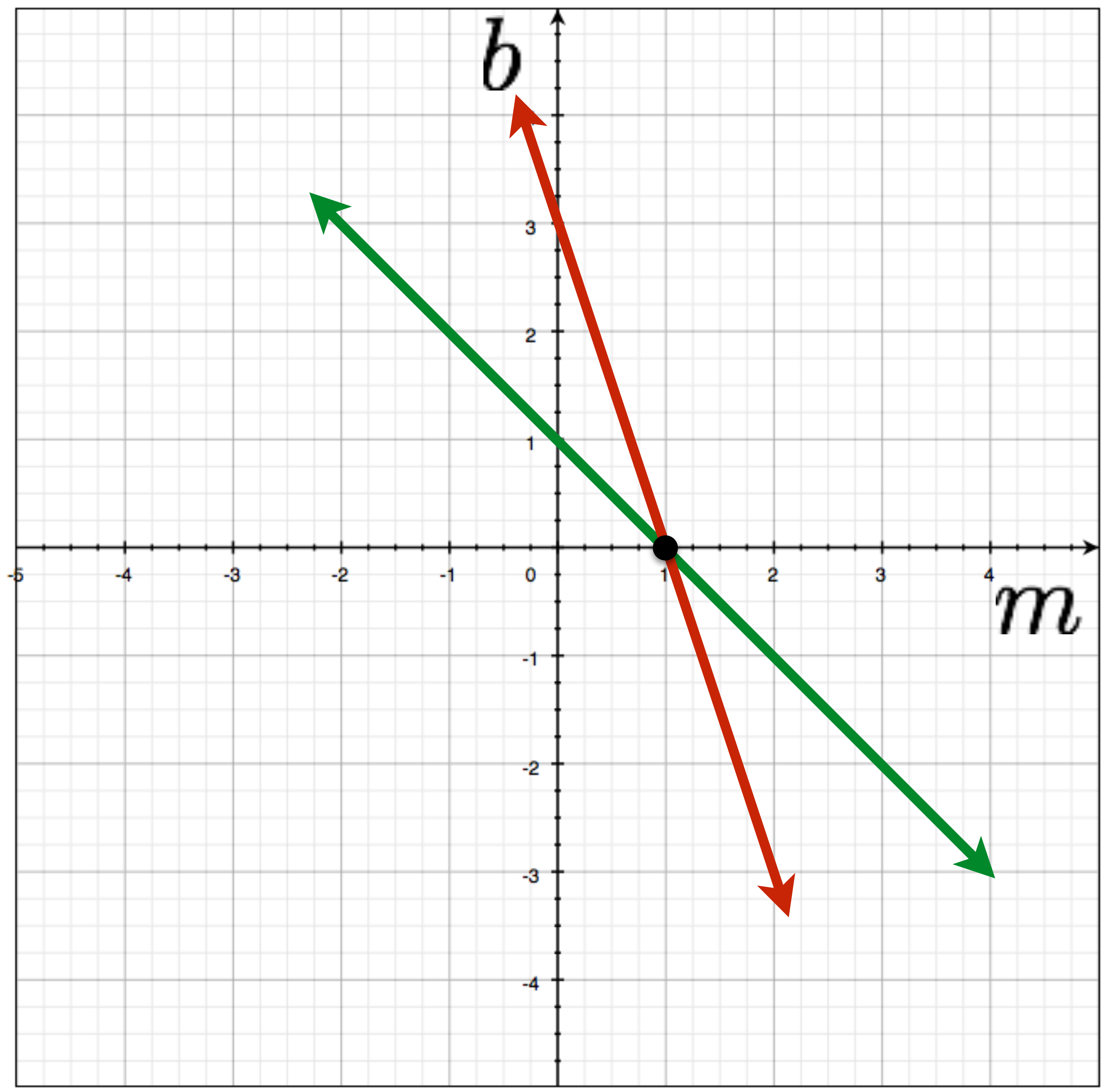
Image space

three points?

variables

$$y - mx = b$$

parameters



Parameter space

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Hough Transform: Lines

variables

$$y = mx + b$$

parameters

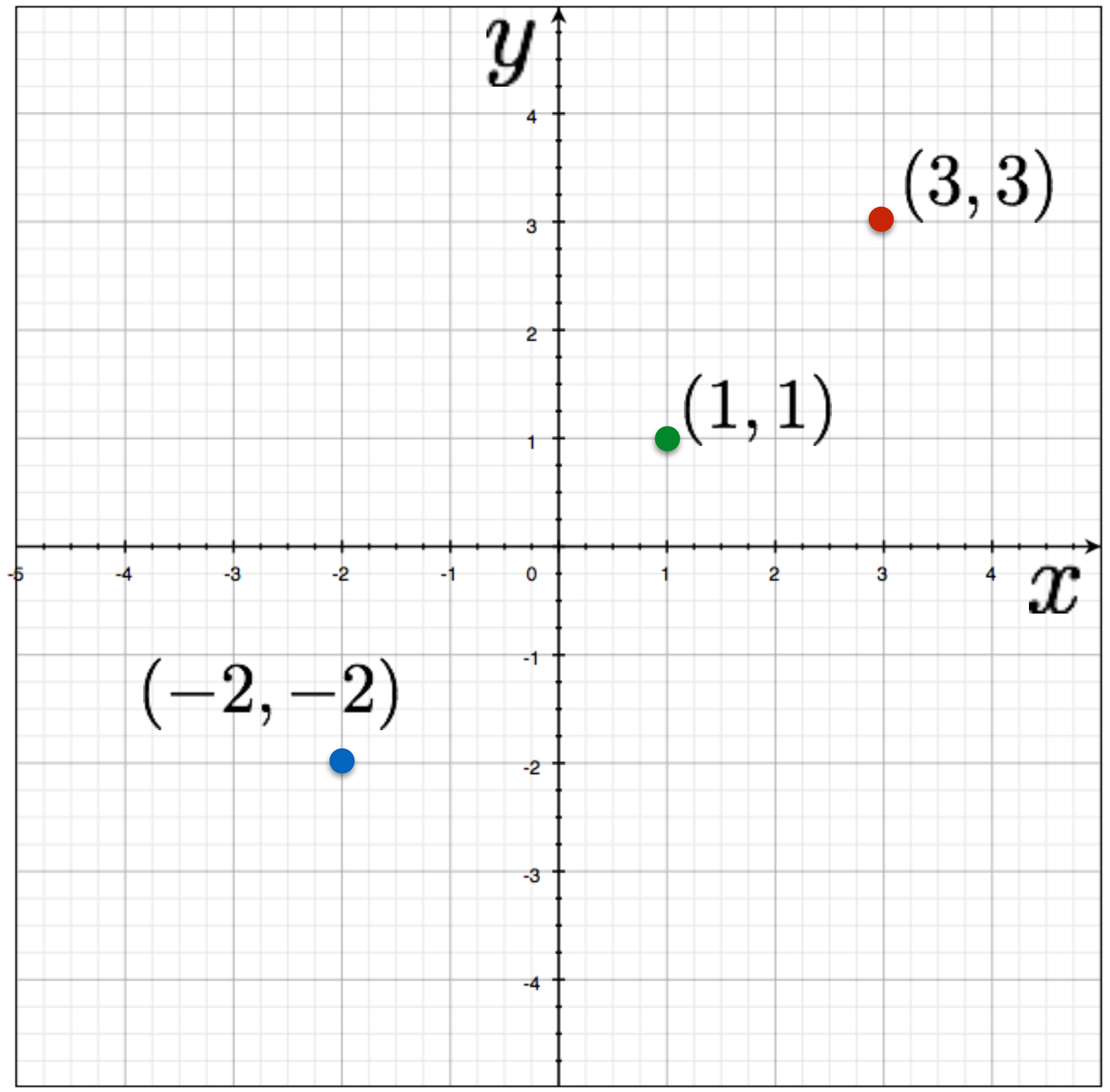


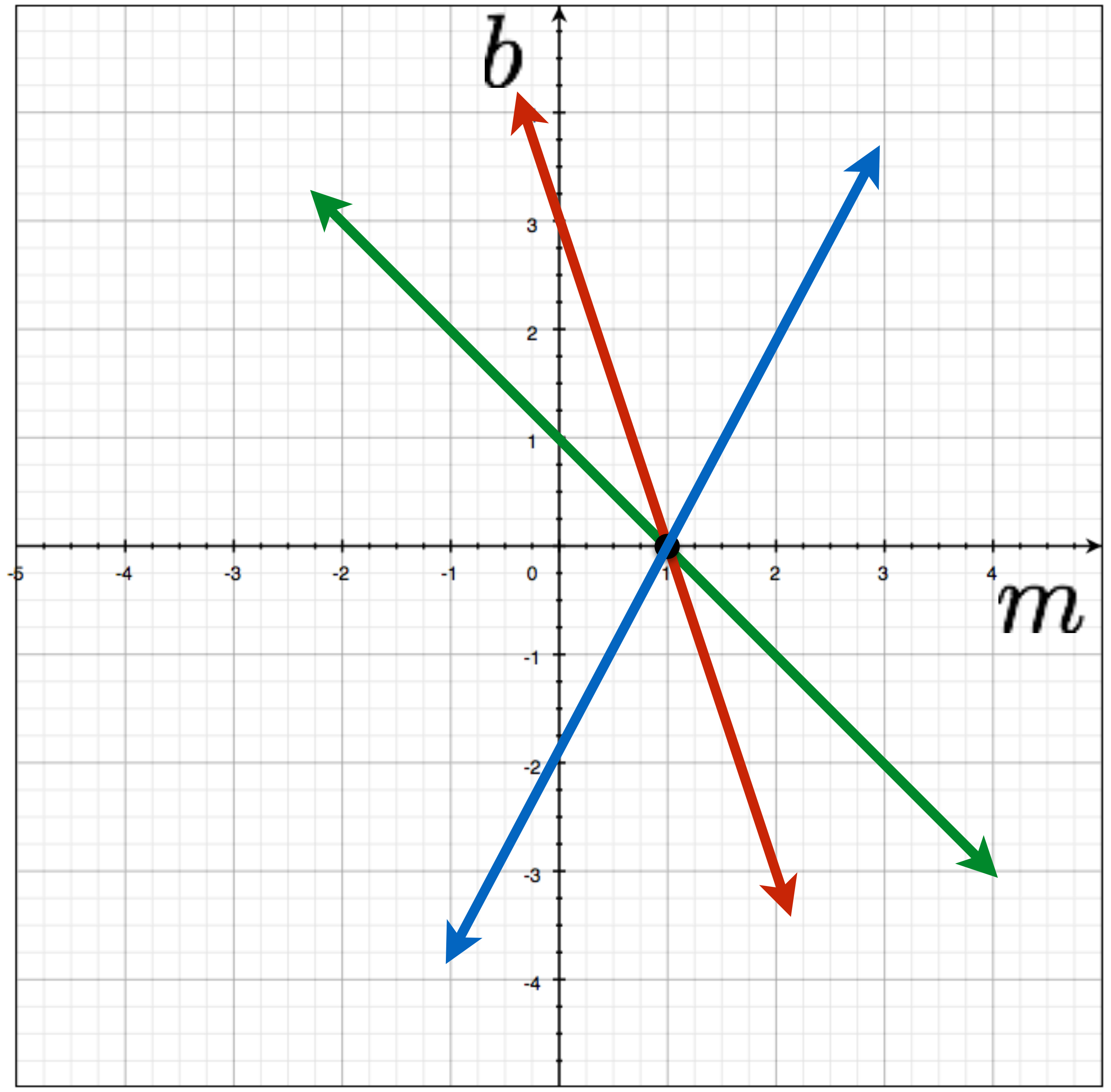
Image space

three points?

variables

$$y - mx = b$$

parameters



Parameter space

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Hough Transform: Lines

variables

$$y = mx + b$$

parameters

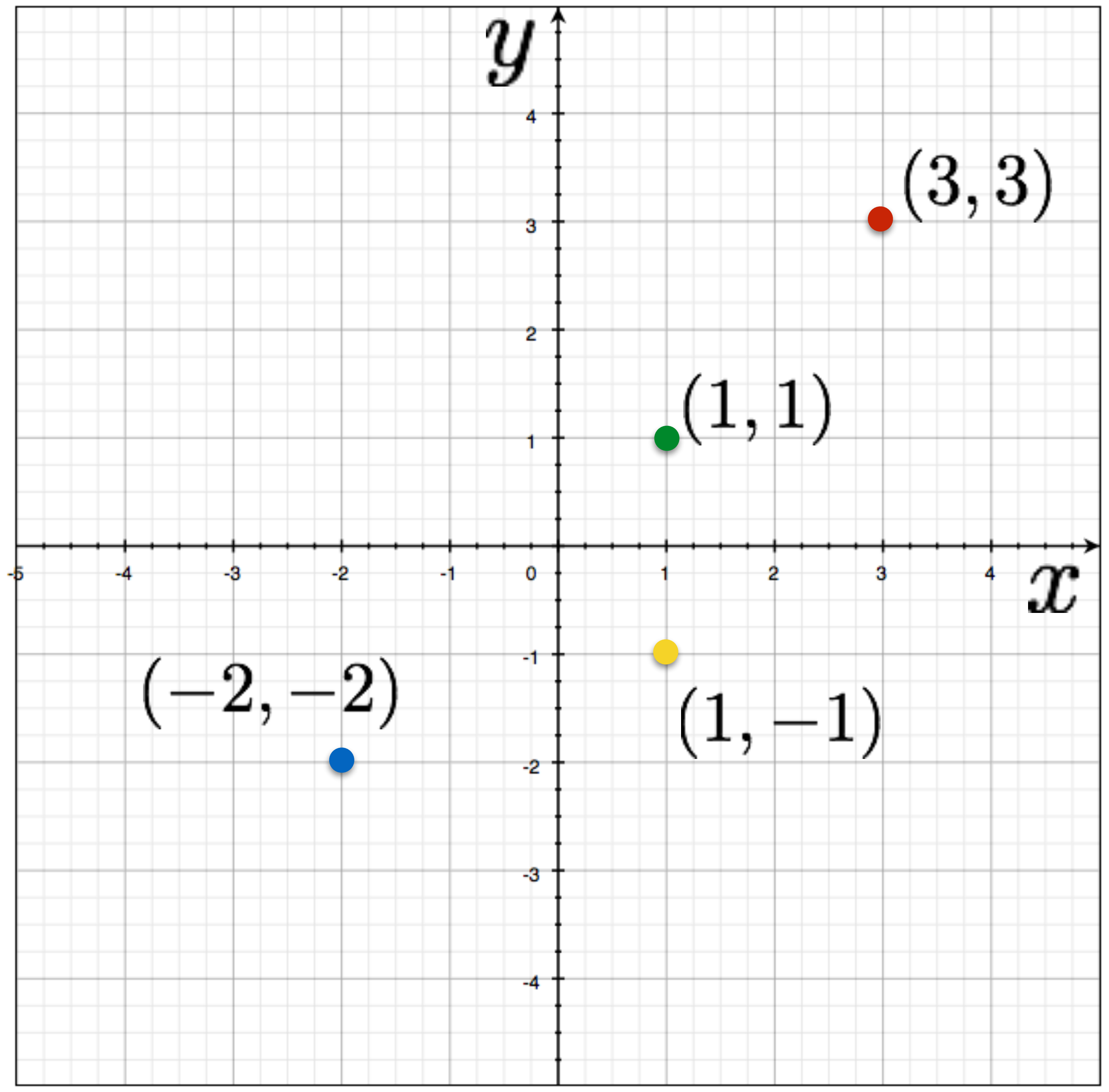


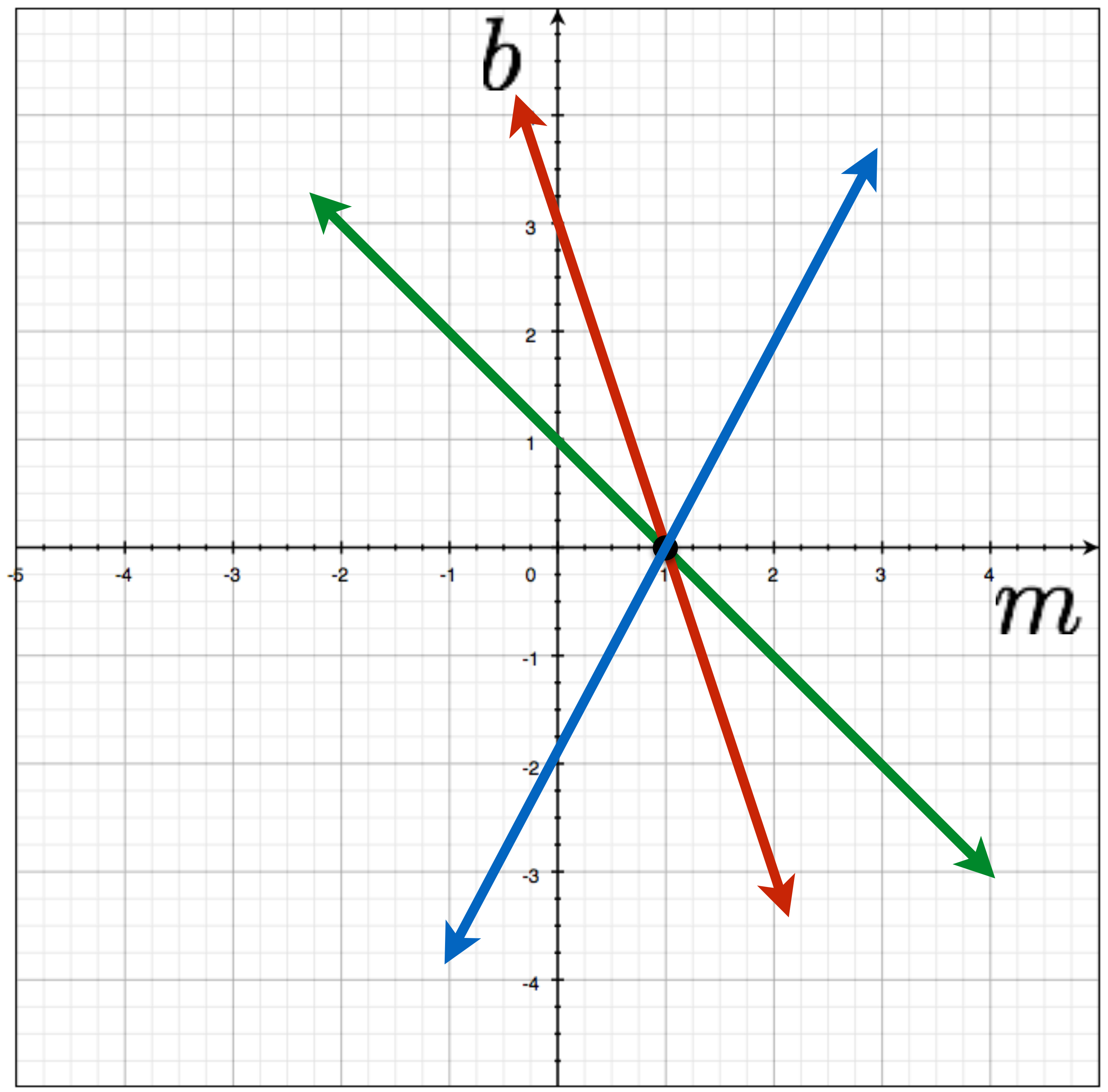
Image space

four points?

variables

$$y - mx = b$$

parameters



Parameter space

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Hough Transform: Lines

variables

$$y = mx + b$$

parameters

variables

$$y - mx = b$$

parameters

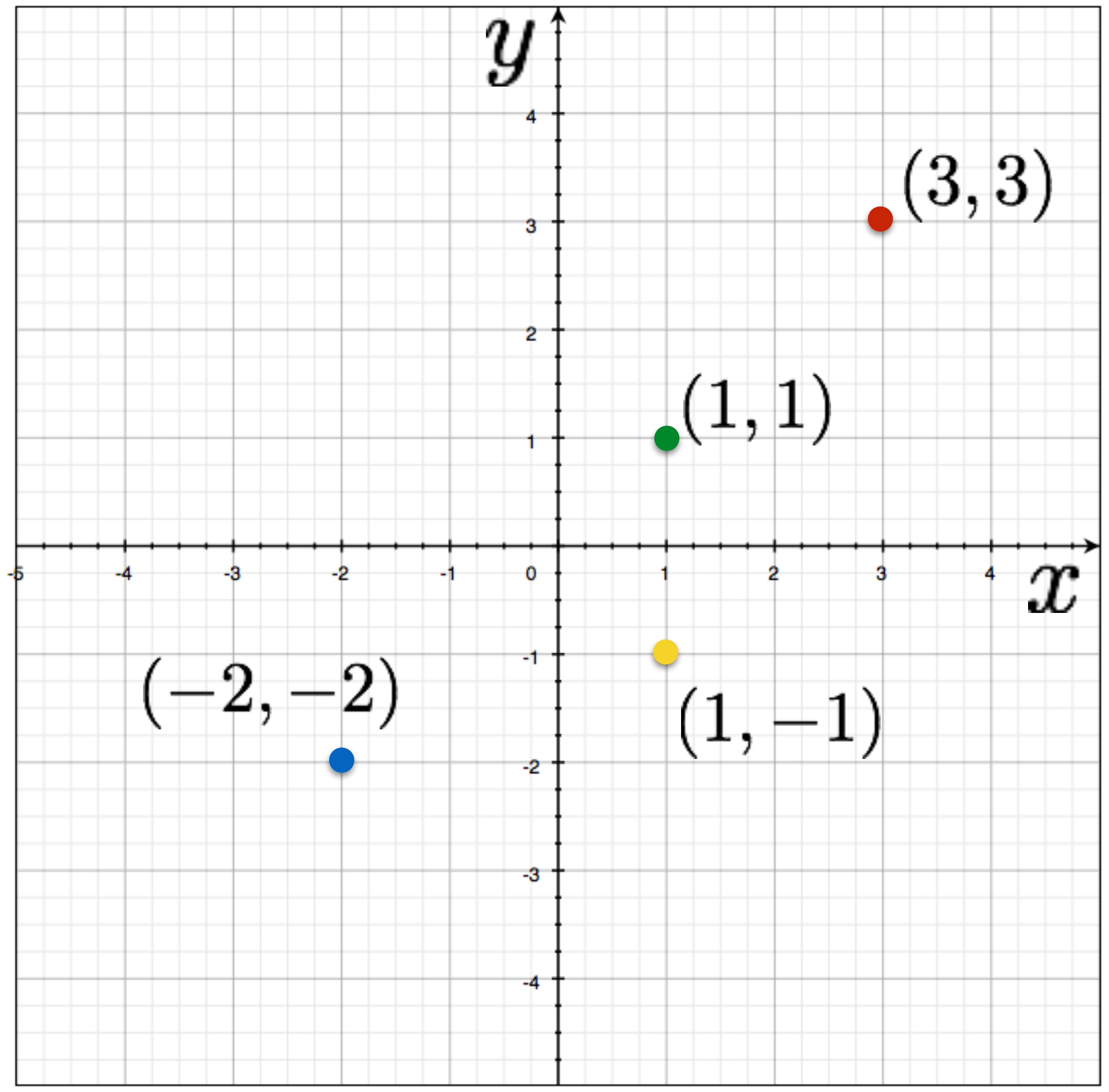
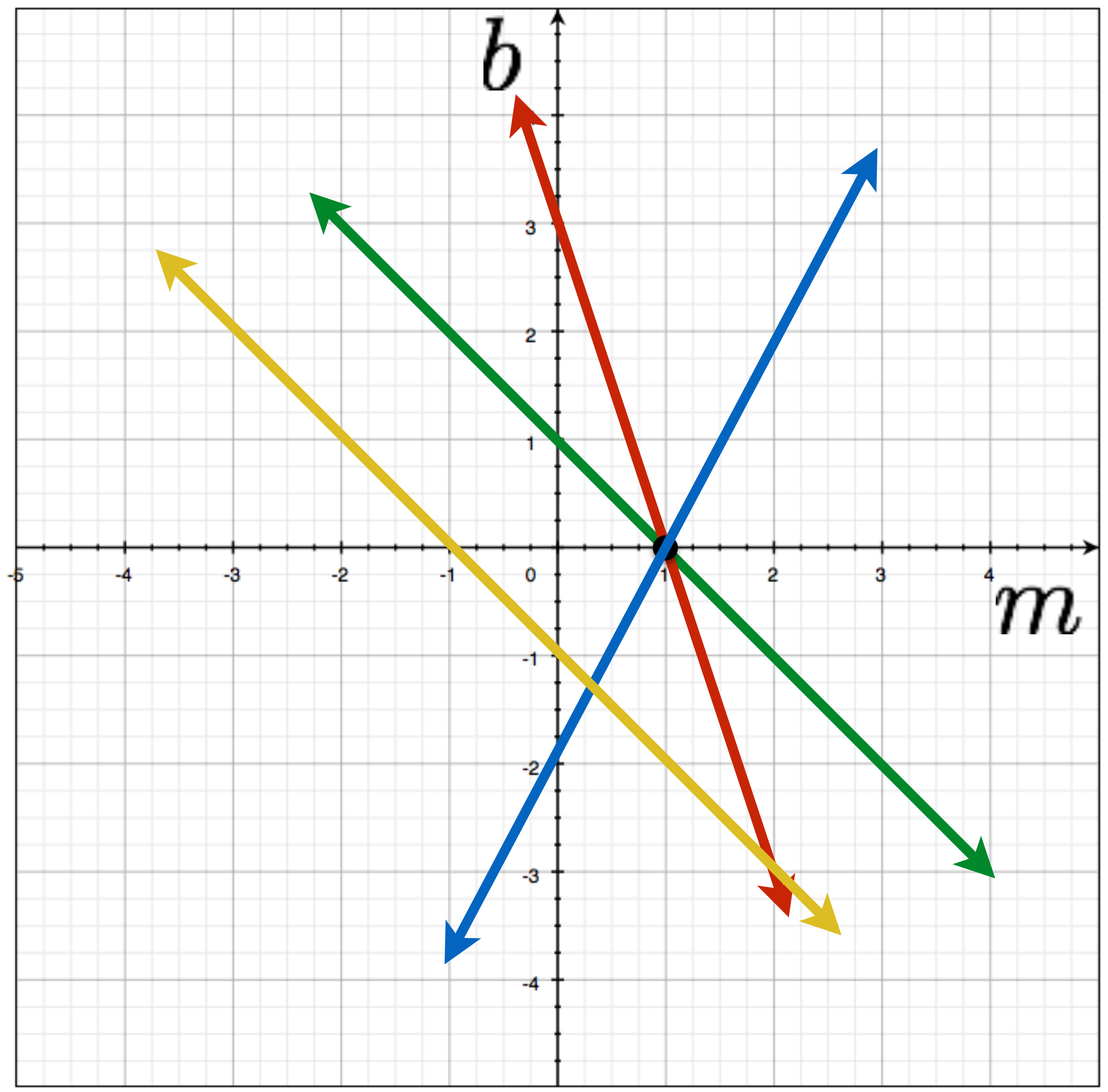


Image space



Parameter space

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Hough Transform: Lines

How would you find the best fitting line?

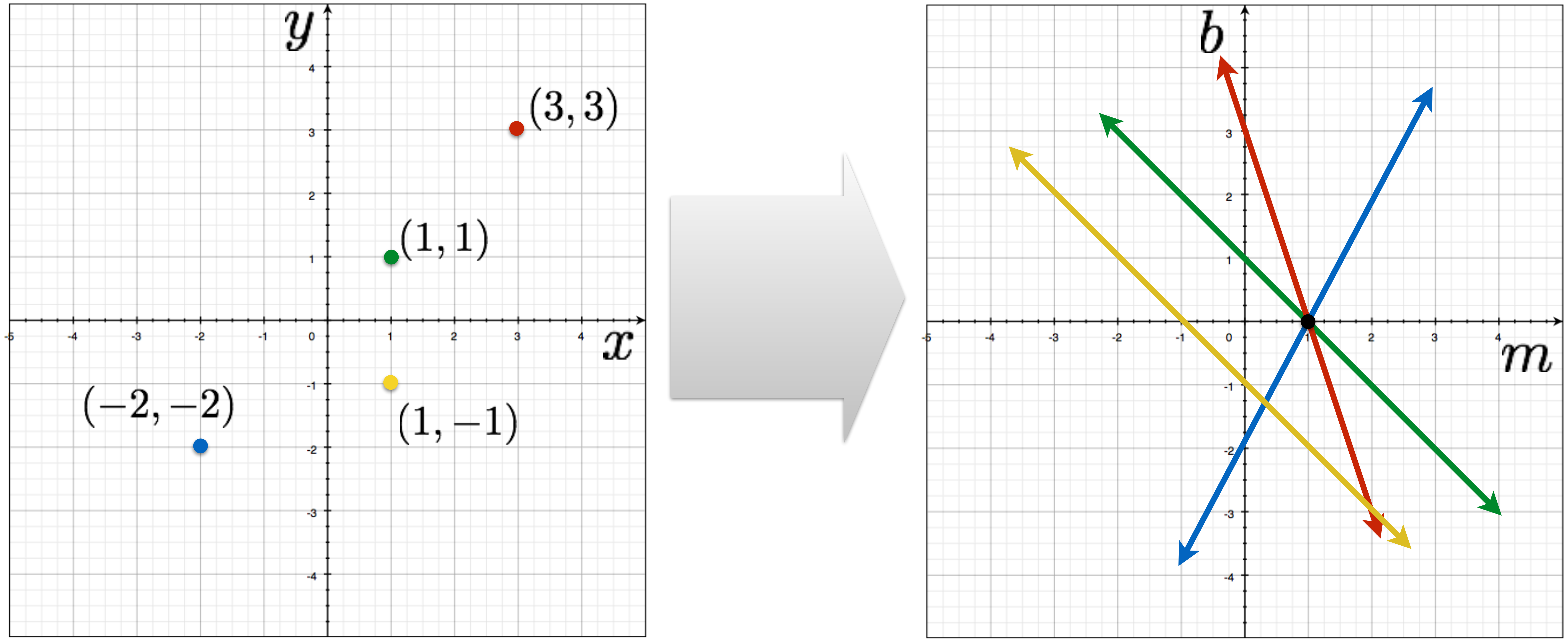


Image space

Parameter space

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Hough Transform: Lines

Is this method robust to measurement noise? clutter?

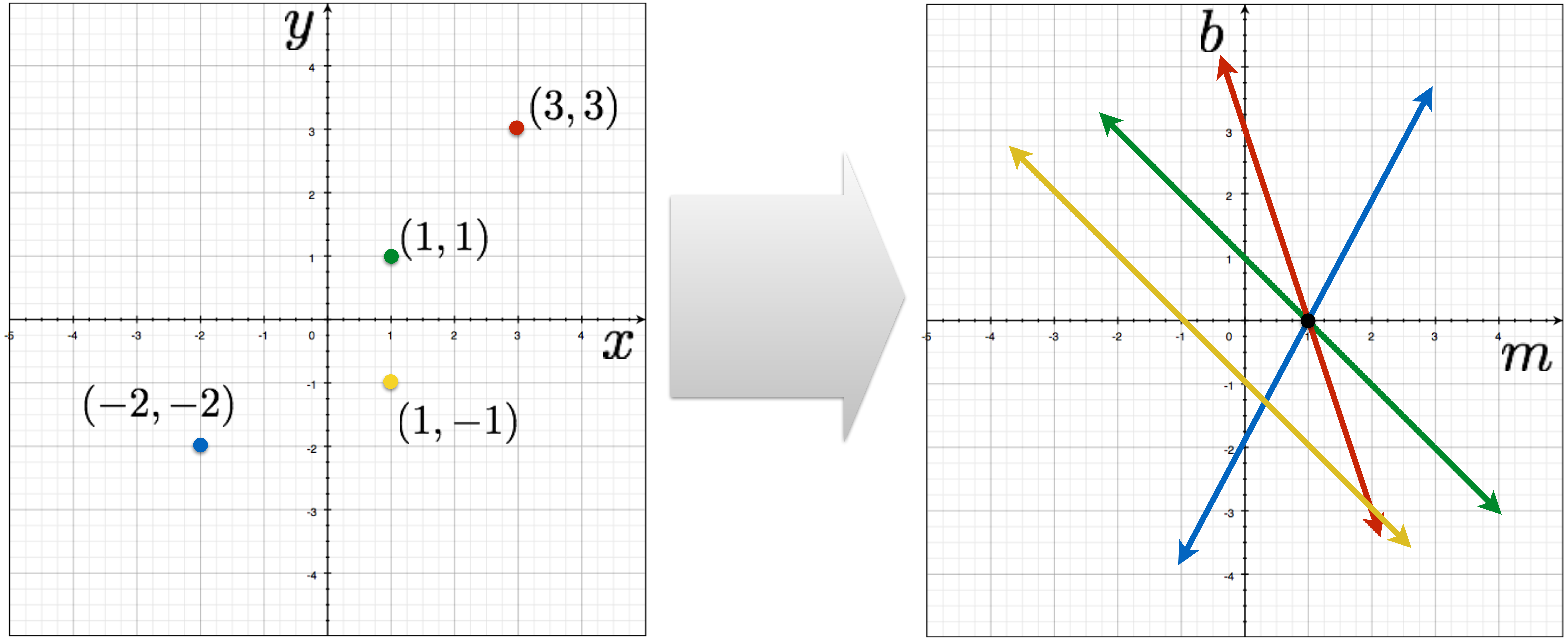


Image space

Parameter space

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Line Detection by Hough Transform

Algorithm:

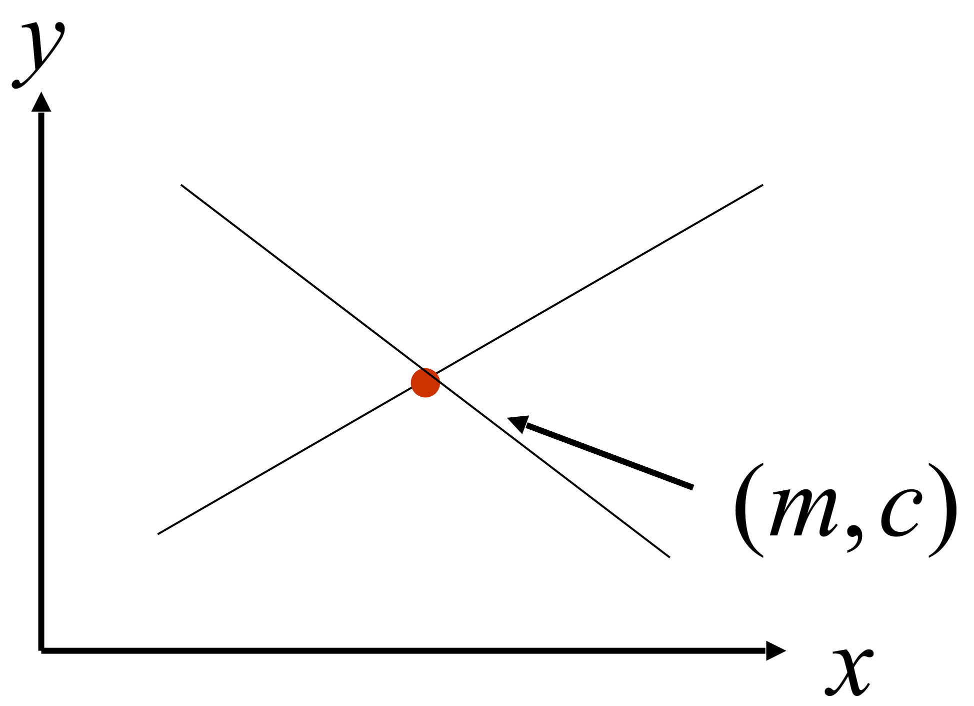
1. Quantize Parameter Space (m,c)

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		

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Problems with **Parametrization**

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		

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Problems with **Parametrization**

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		

The space of m is huge!

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

The space of c is huge!

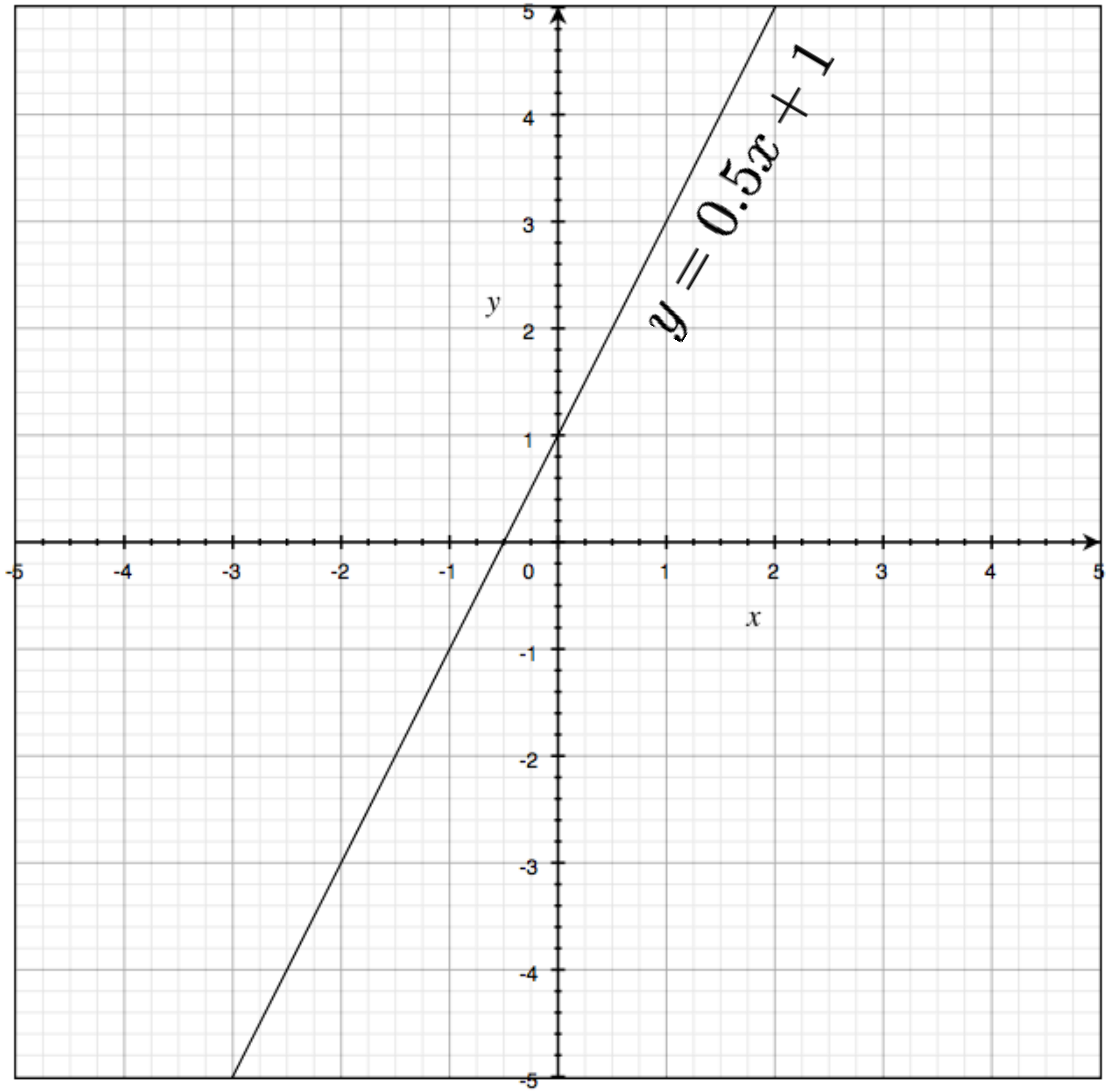
$$-\infty \leq c \leq \infty$$

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Lines: Slope intercept form

$$y = mx + b$$

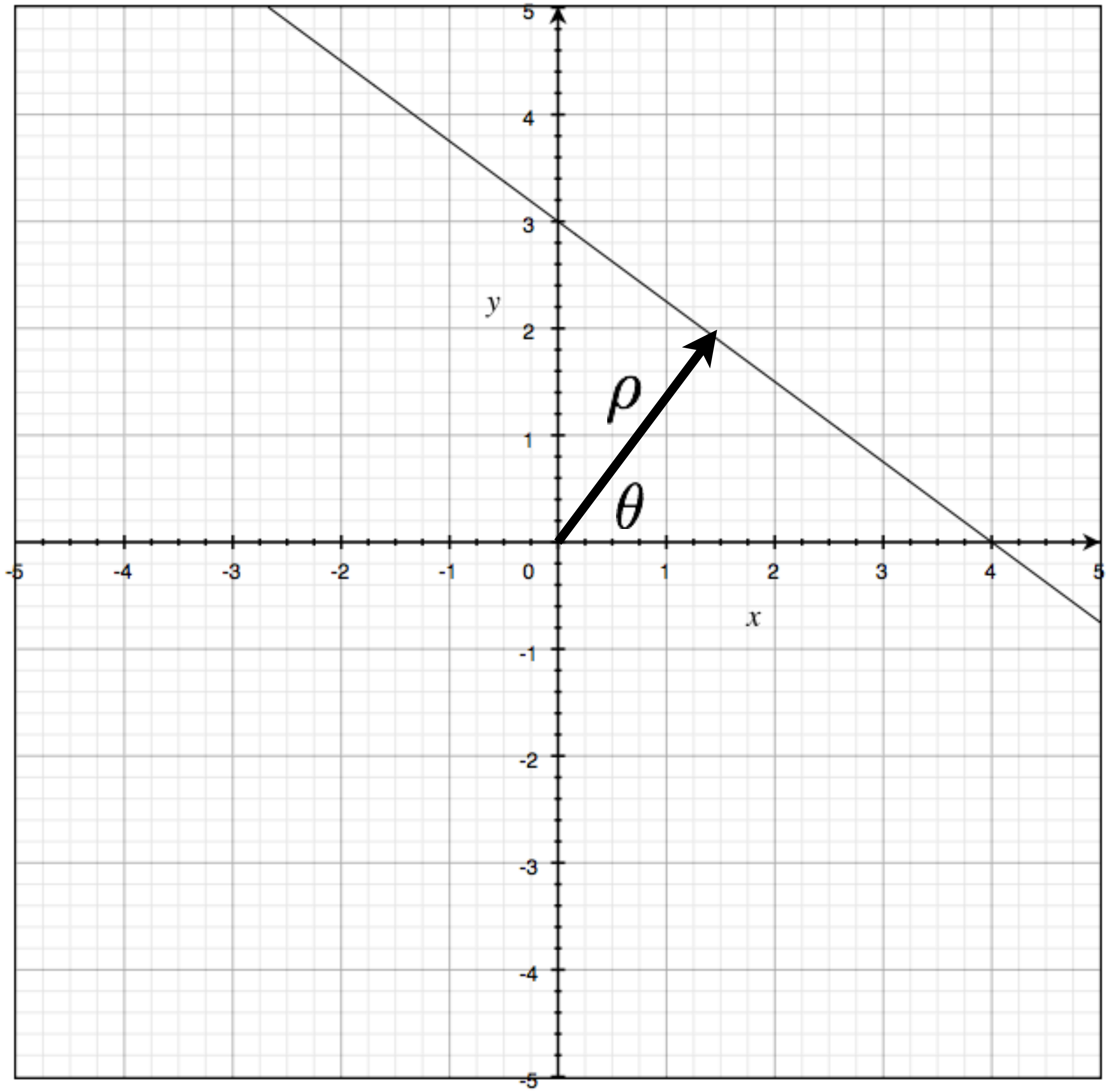
slope y-intercept



Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Lines: Normal form

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



Book's convention

$$x \sin\theta + y \cos\theta + r = 0$$

$$r \geq 0$$

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

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Hough Transform: Lines

variables

$$y = mx + b$$

parameters

parameters

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

variables

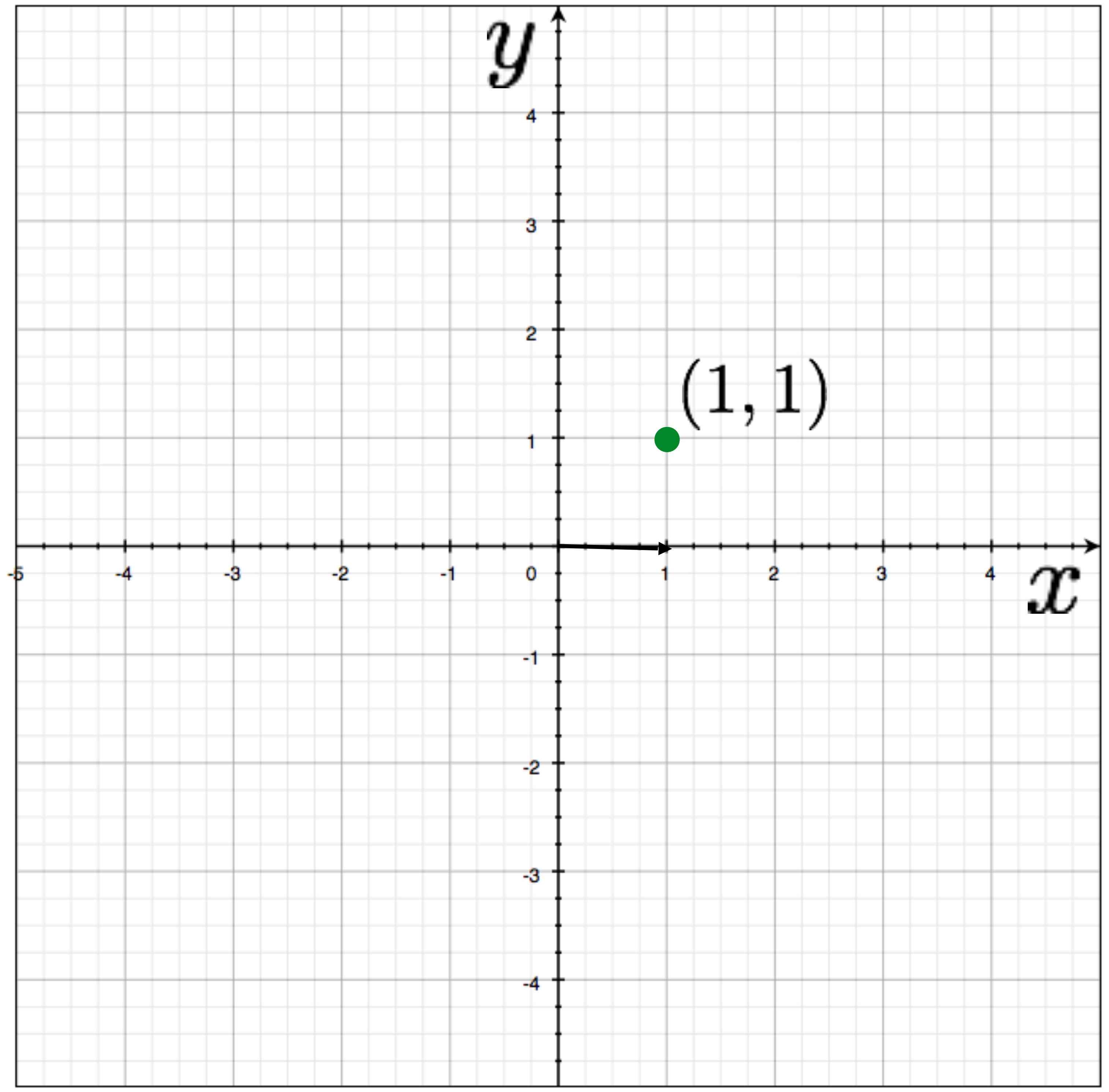
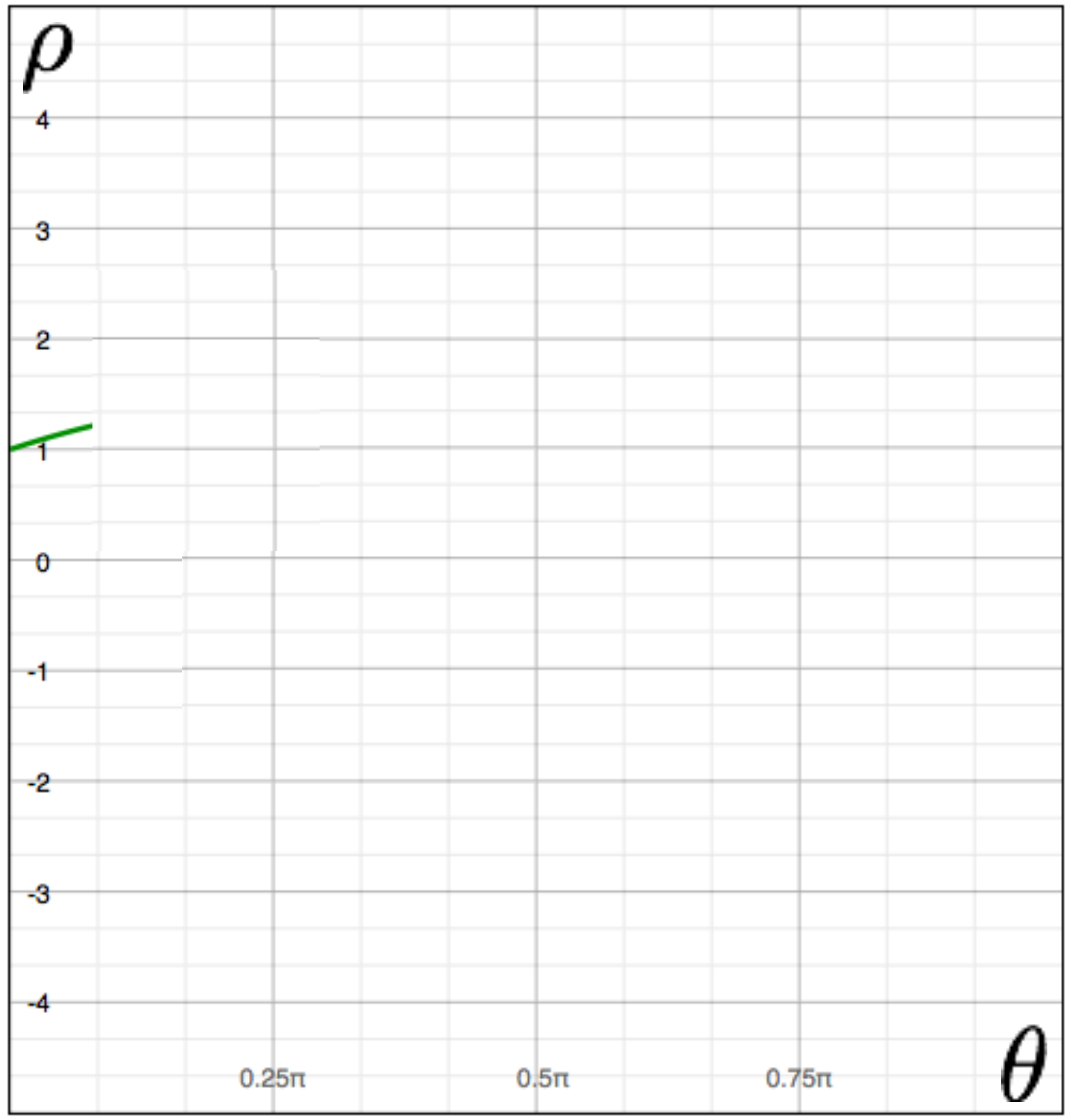


Image space

a point becomes?



Parameter space

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Hough Transform: Lines

variables

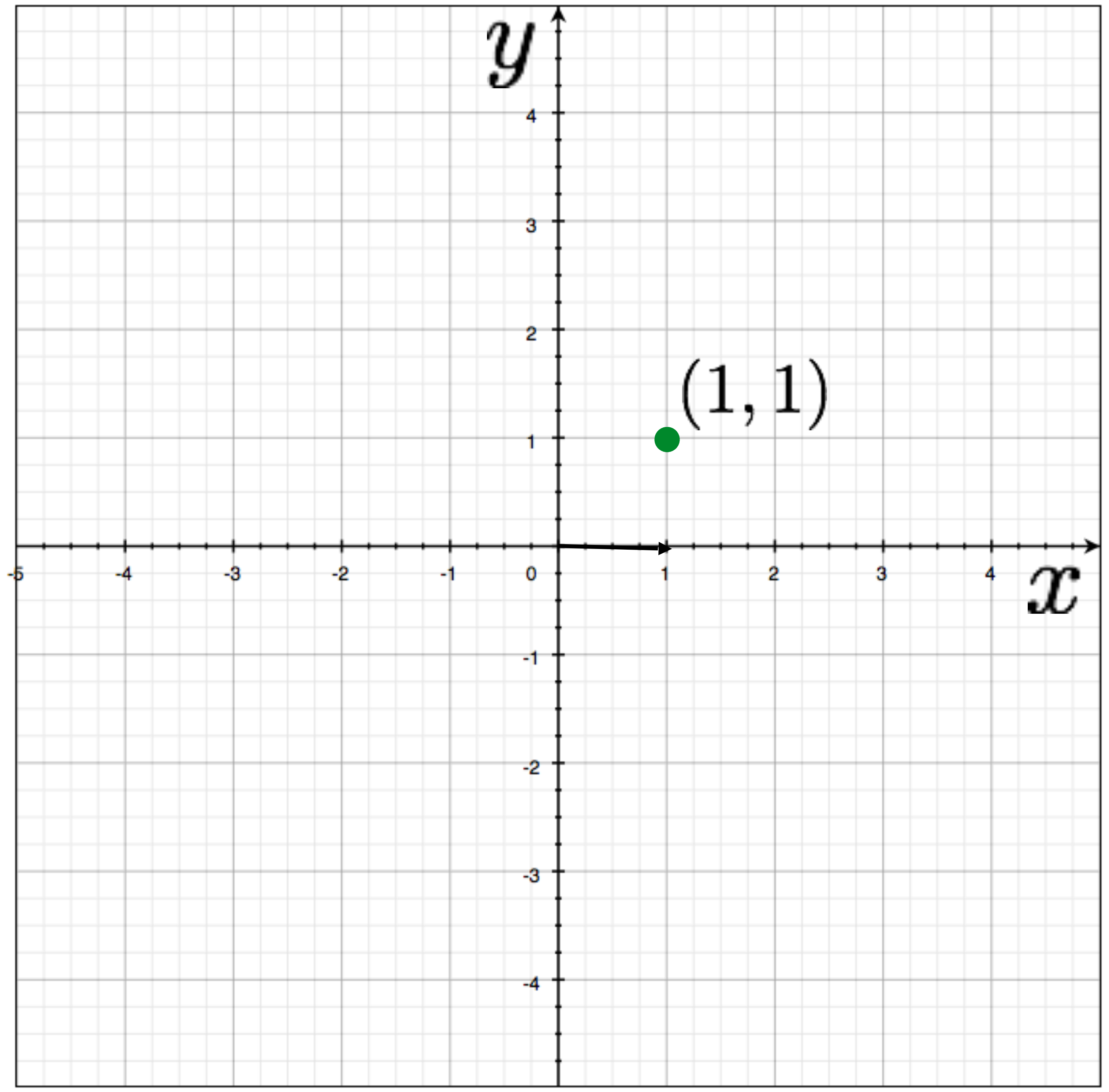
$$y = mx + b$$

parameters

parameters

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

variables



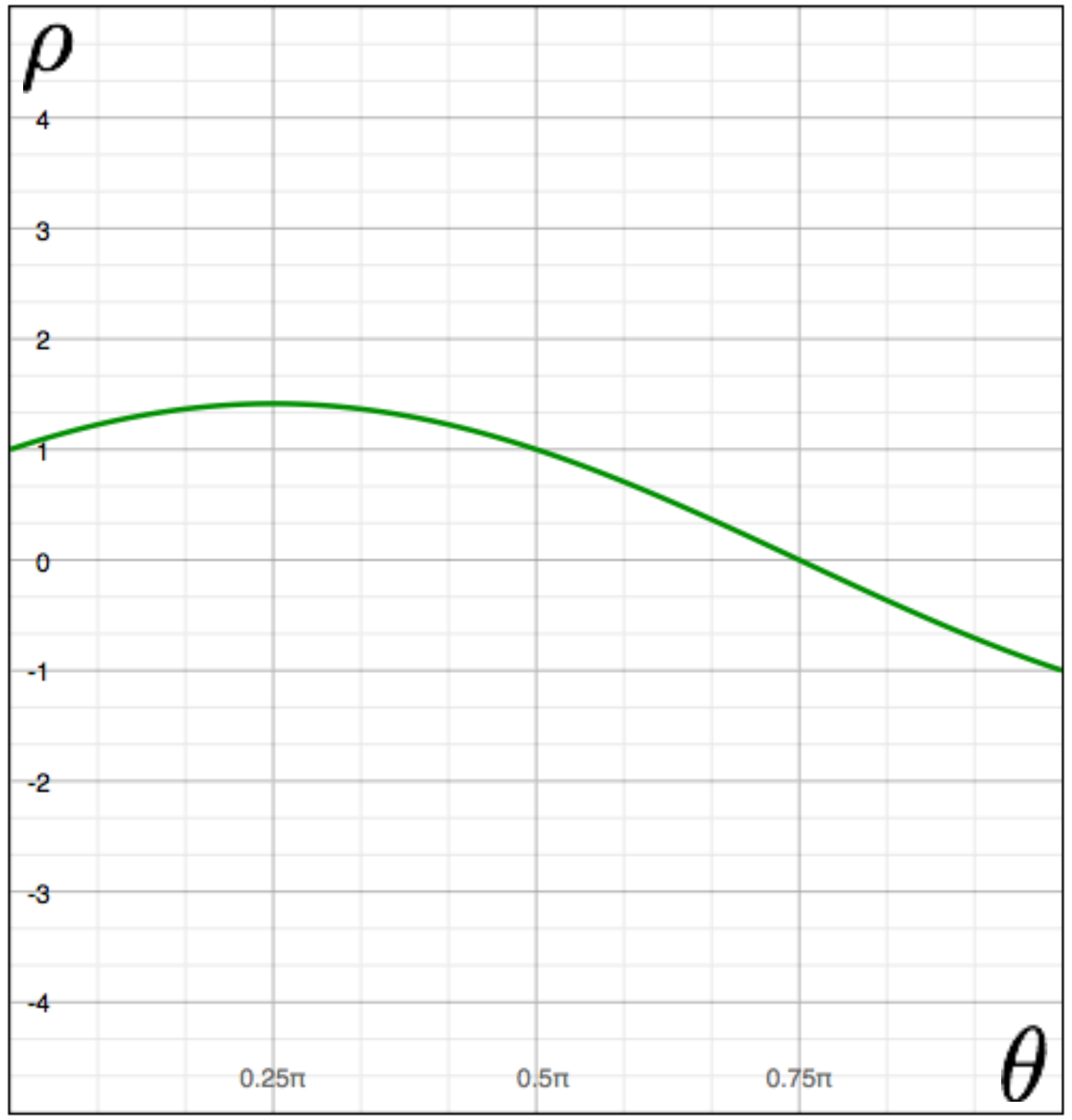


Image space

Parameter space

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Hough Transform: Lines

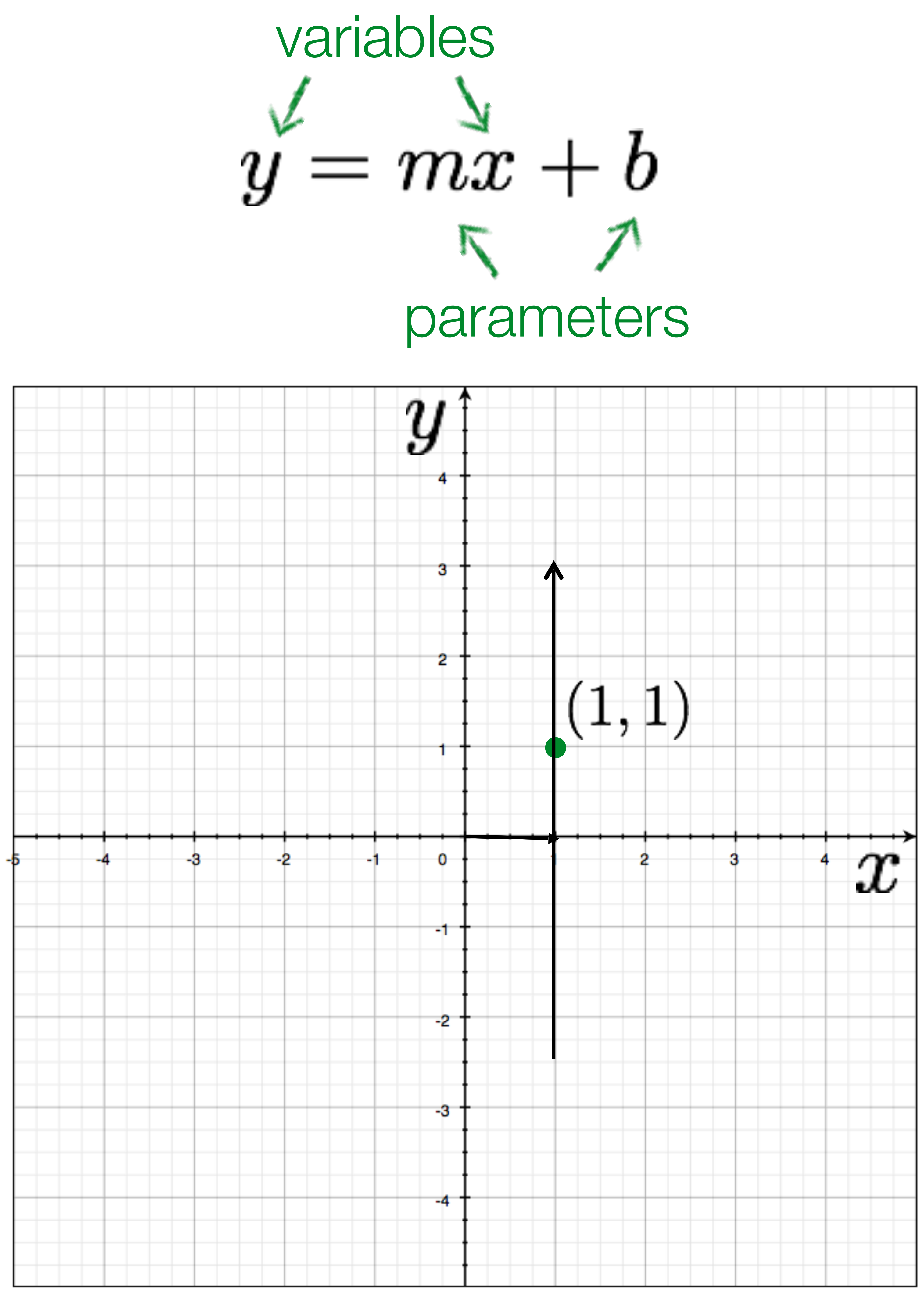
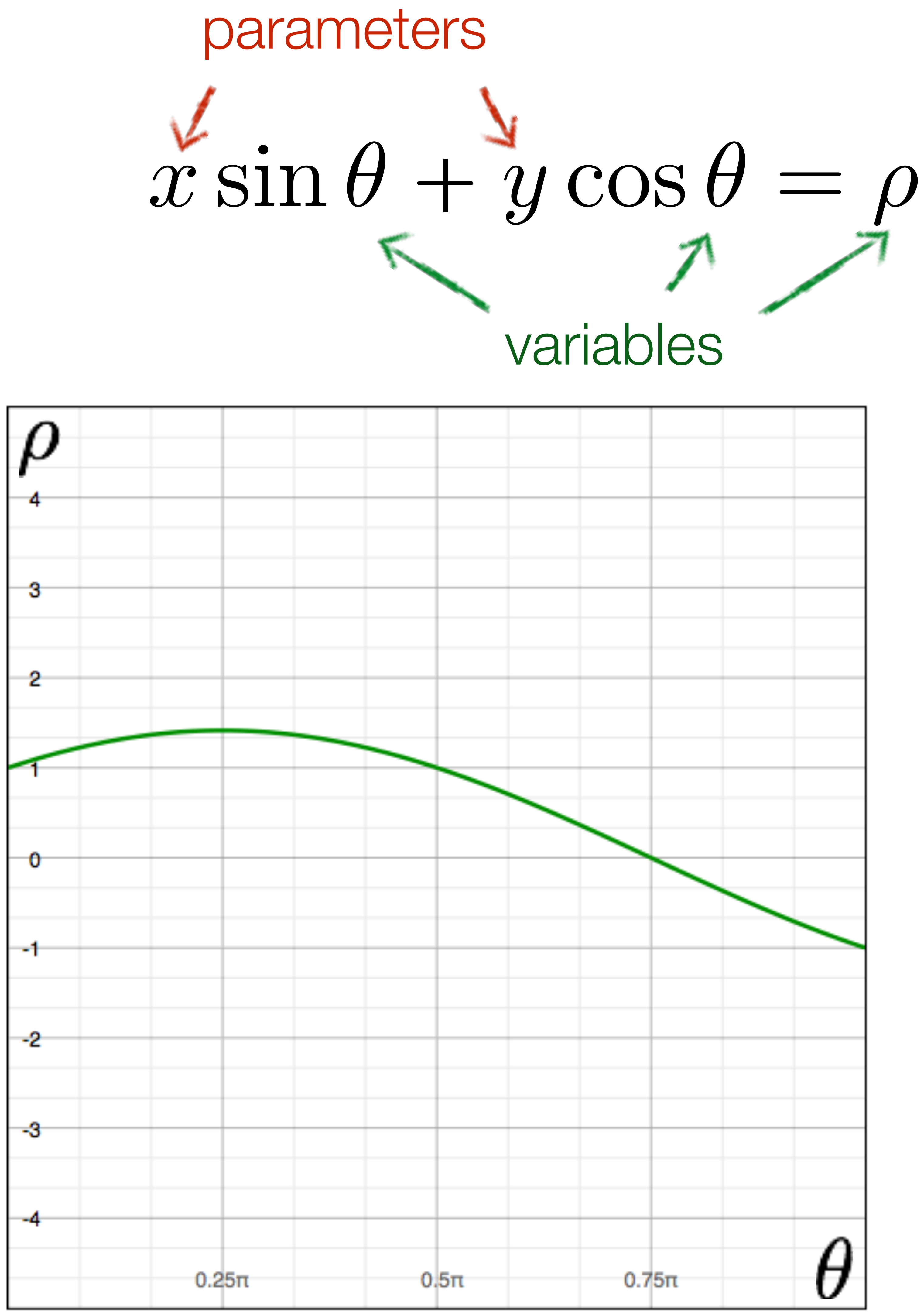


Image space



Parameter space

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Hough Transform: Lines

variables

$$y = mx + b$$

parameters

parameters

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

variables

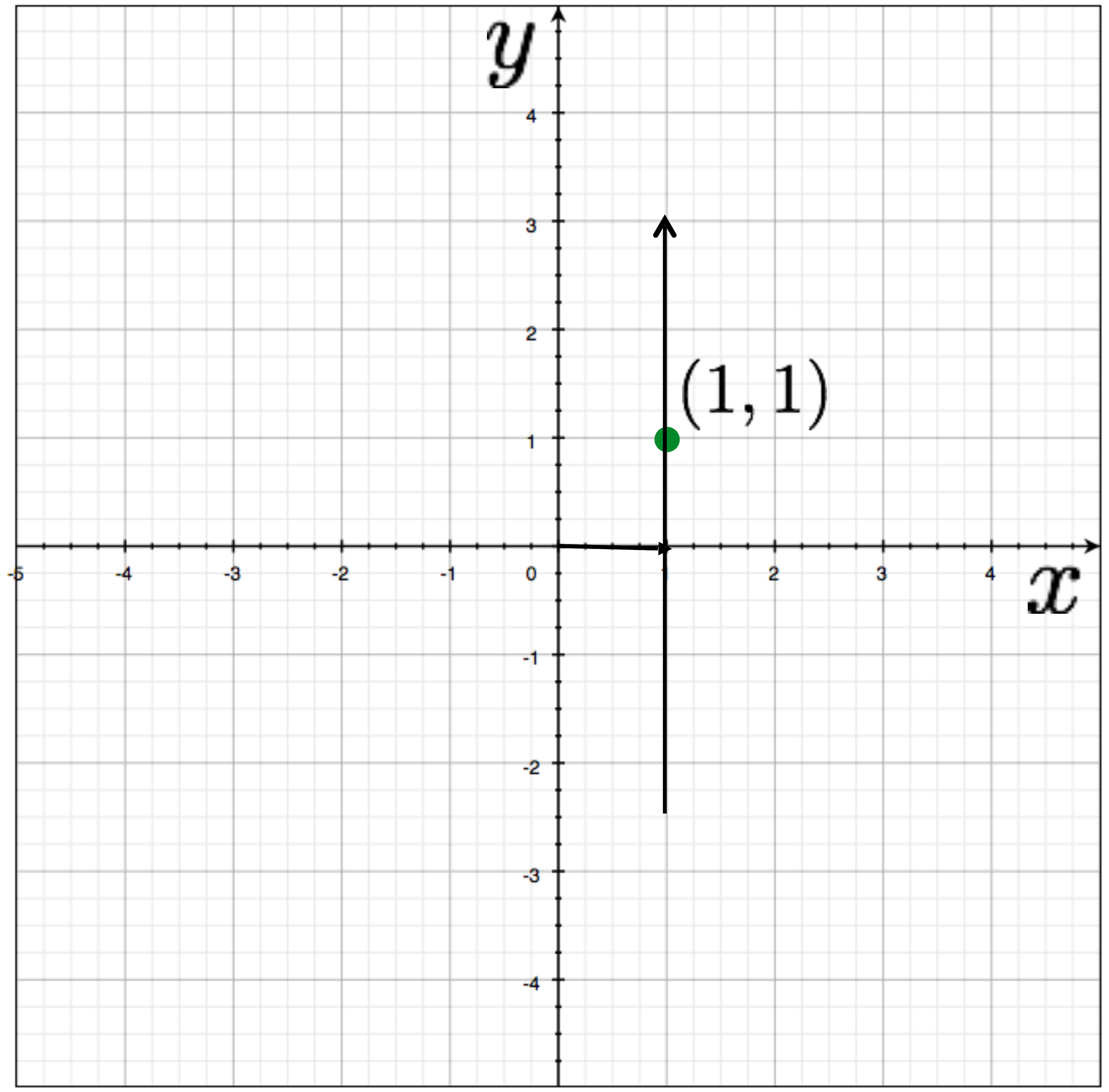
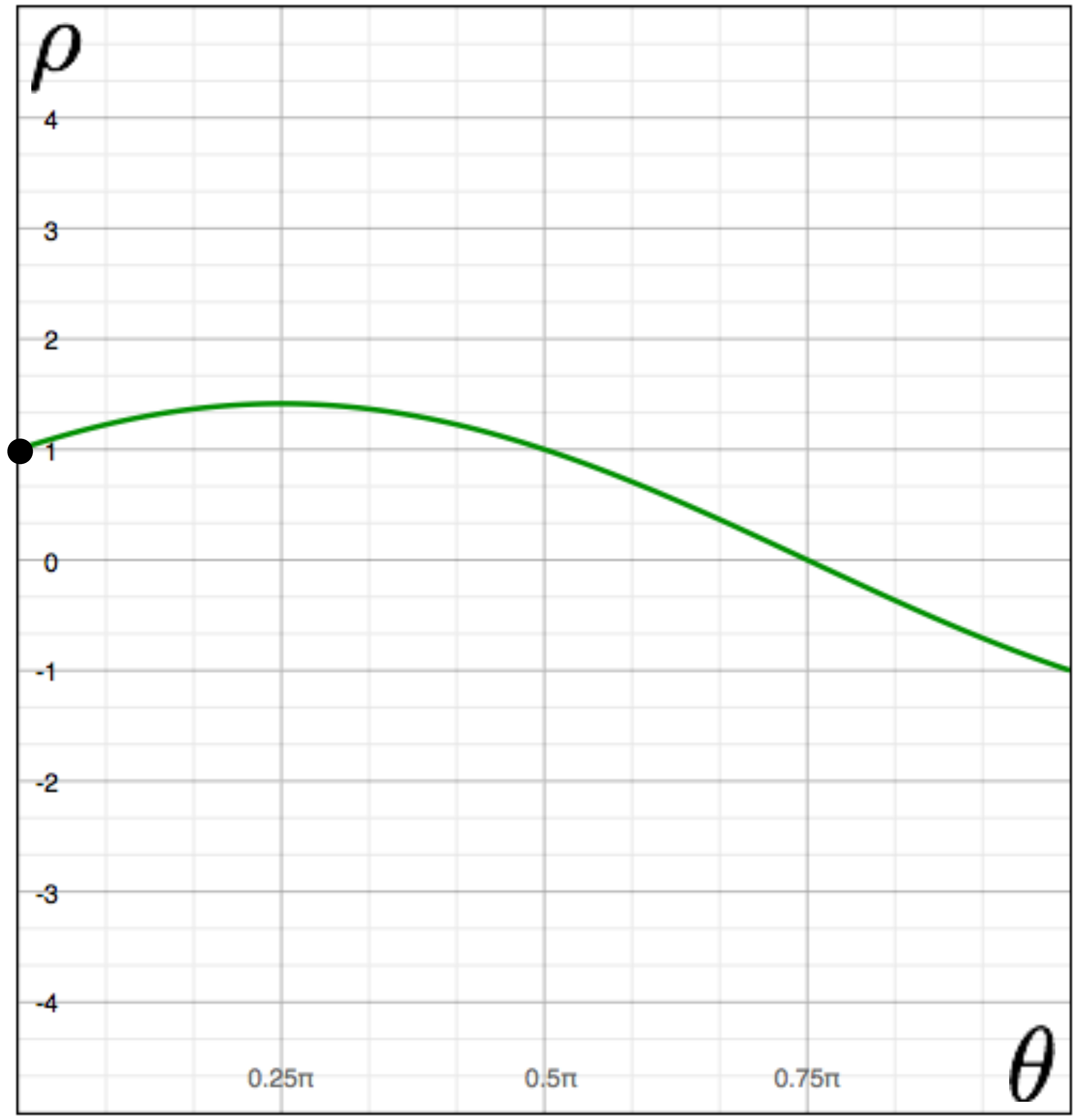


Image space

a line becomes a point



Parameter space

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Hough Transform: Lines

variables

$$y = mx + b$$

parameters

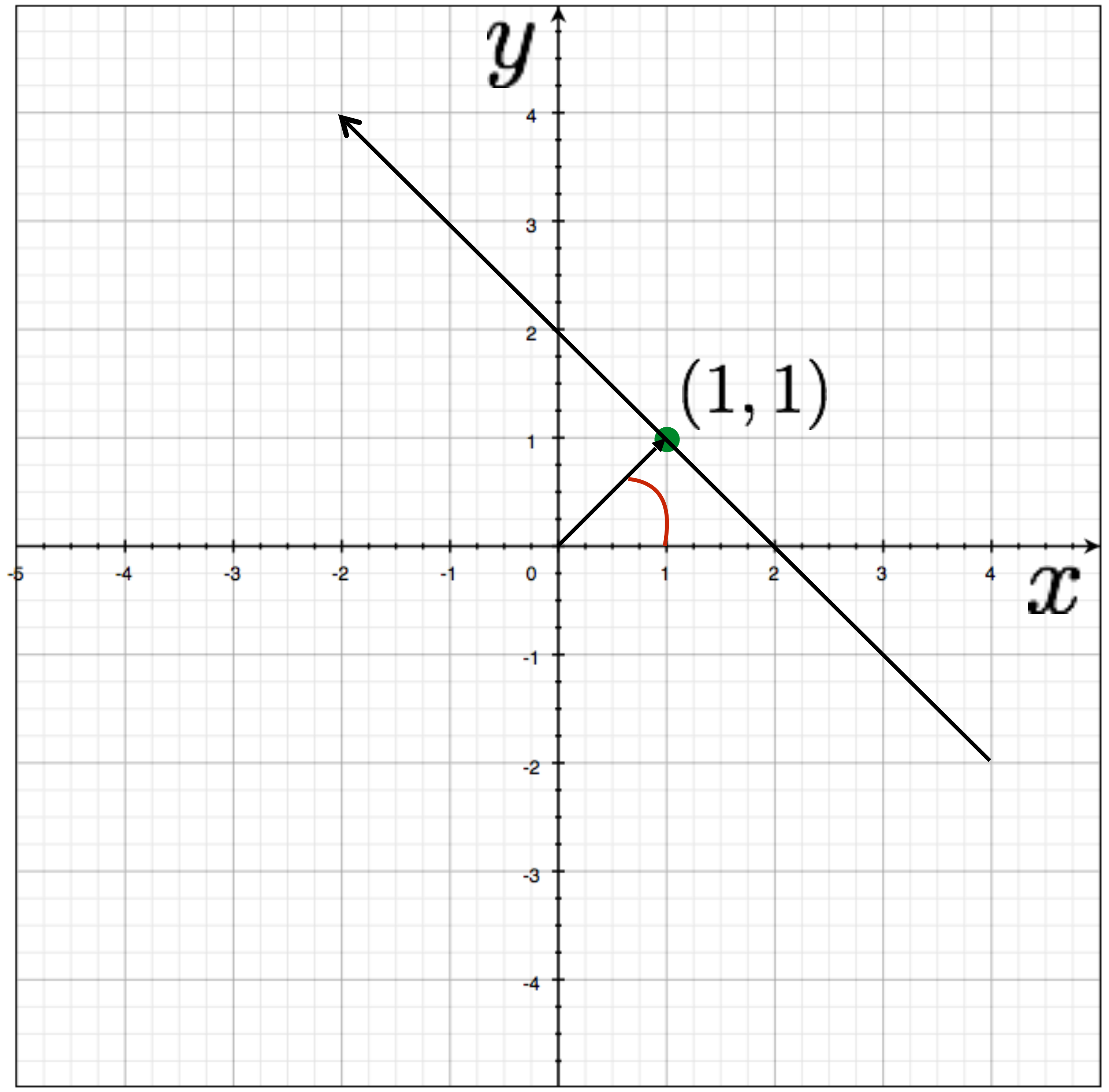


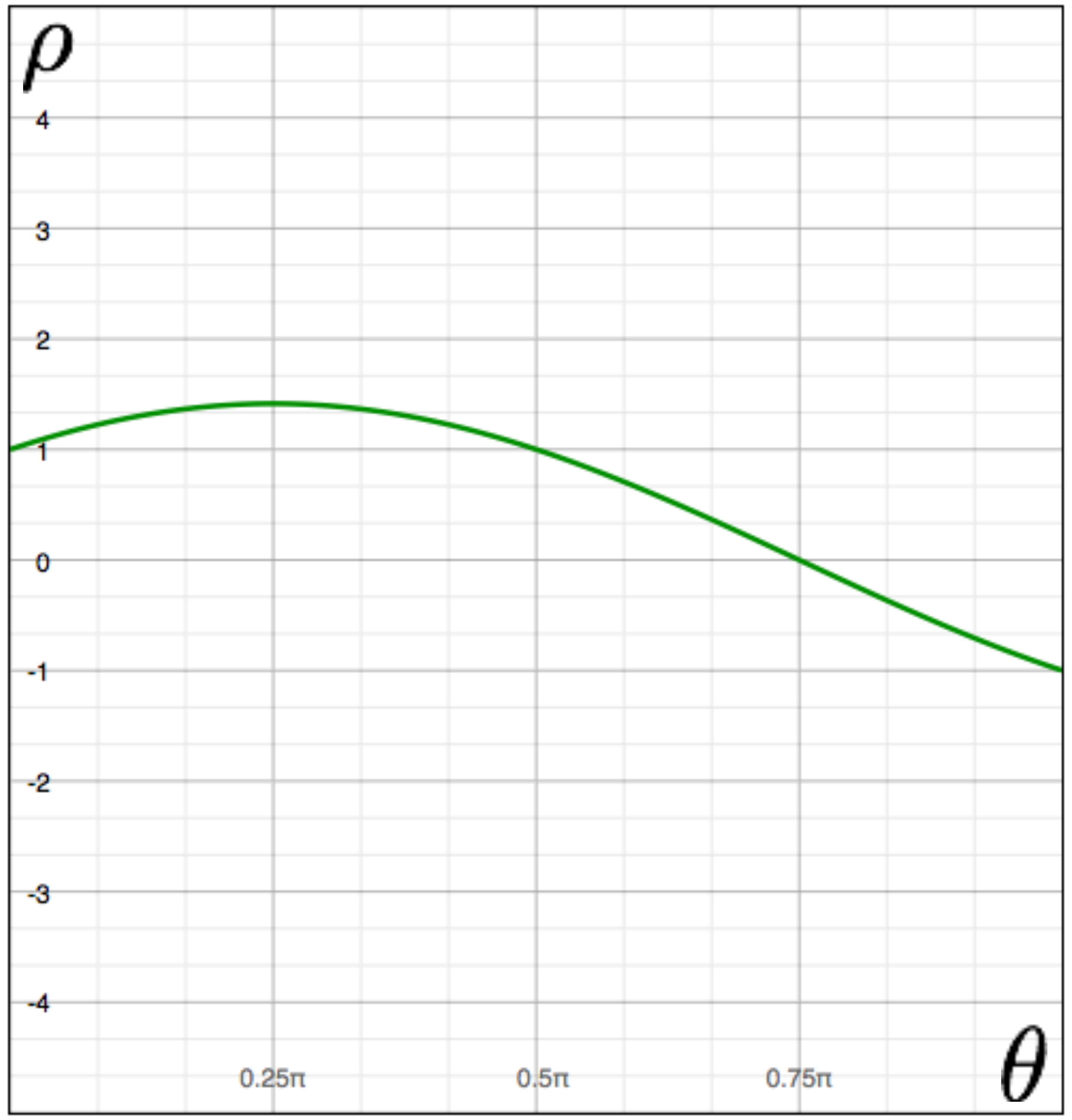
Image space

a line becomes?

parameters

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

variables



Parameter space

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Hough Transform: Lines

variables

$$y = mx + b$$

parameters

parameters

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

variables

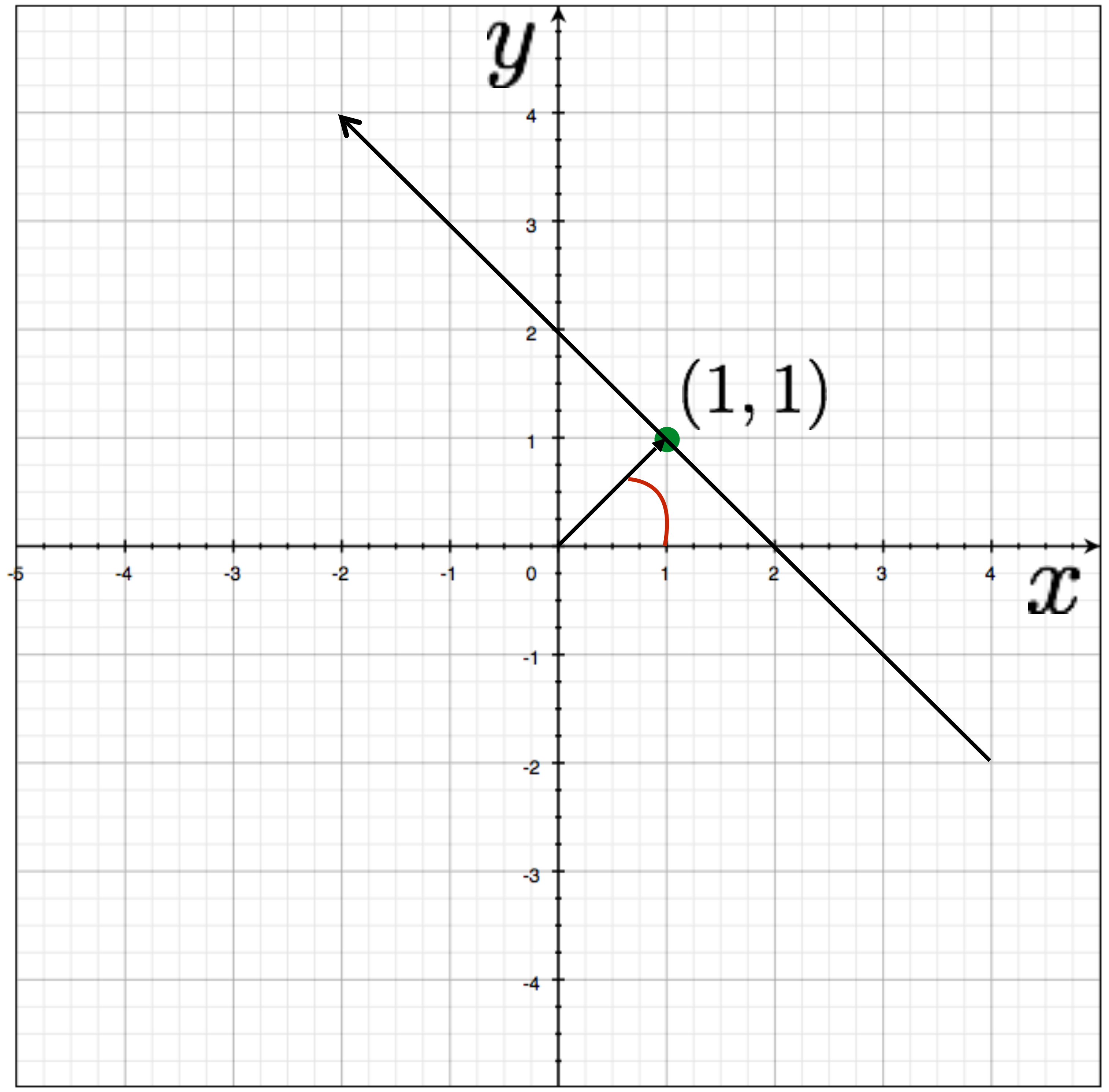
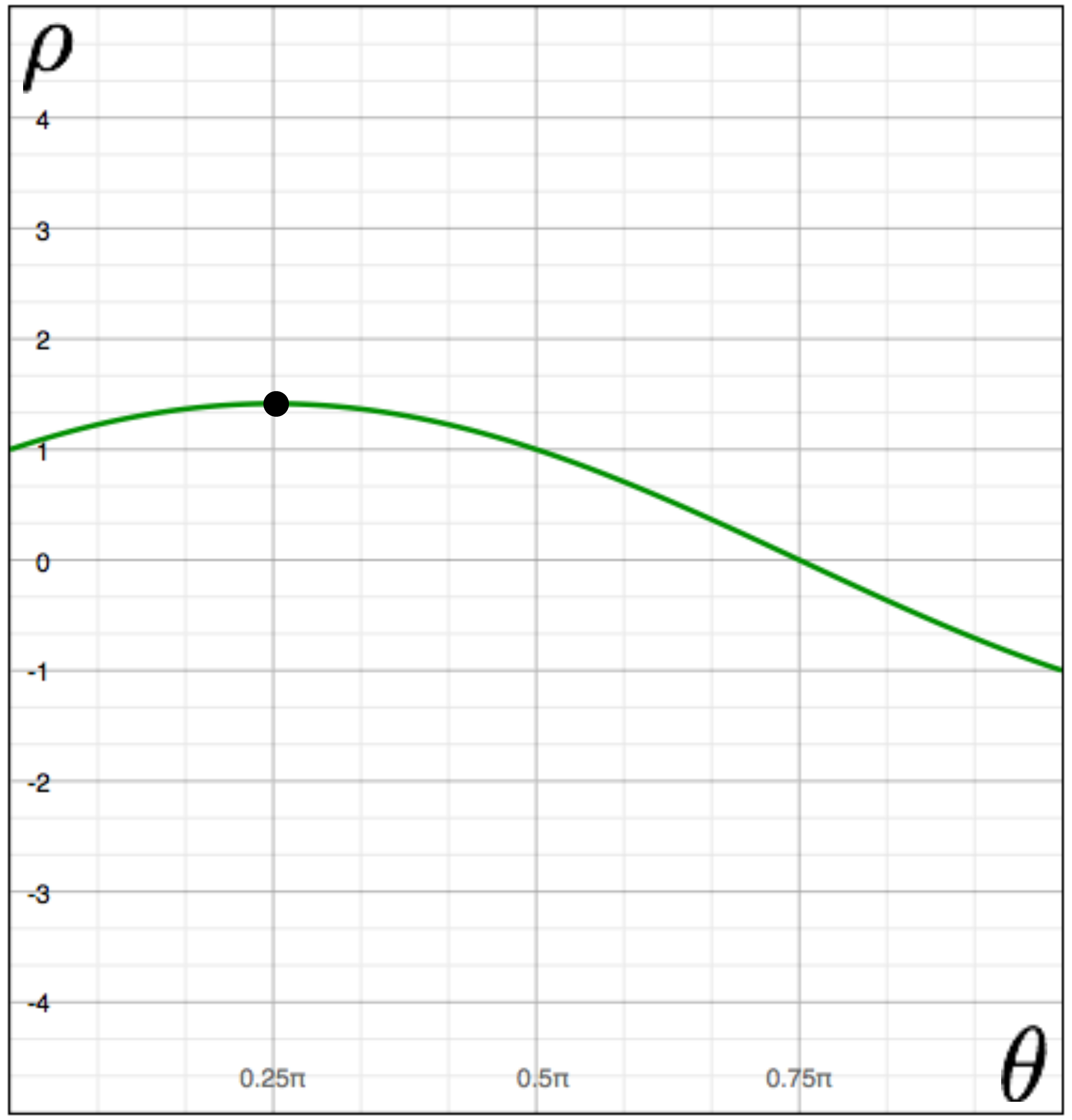


Image space

a line becomes a point



Parameter space

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Hough Transform: Lines

variables

$$y = mx + b$$

parameters

parameters

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

variables

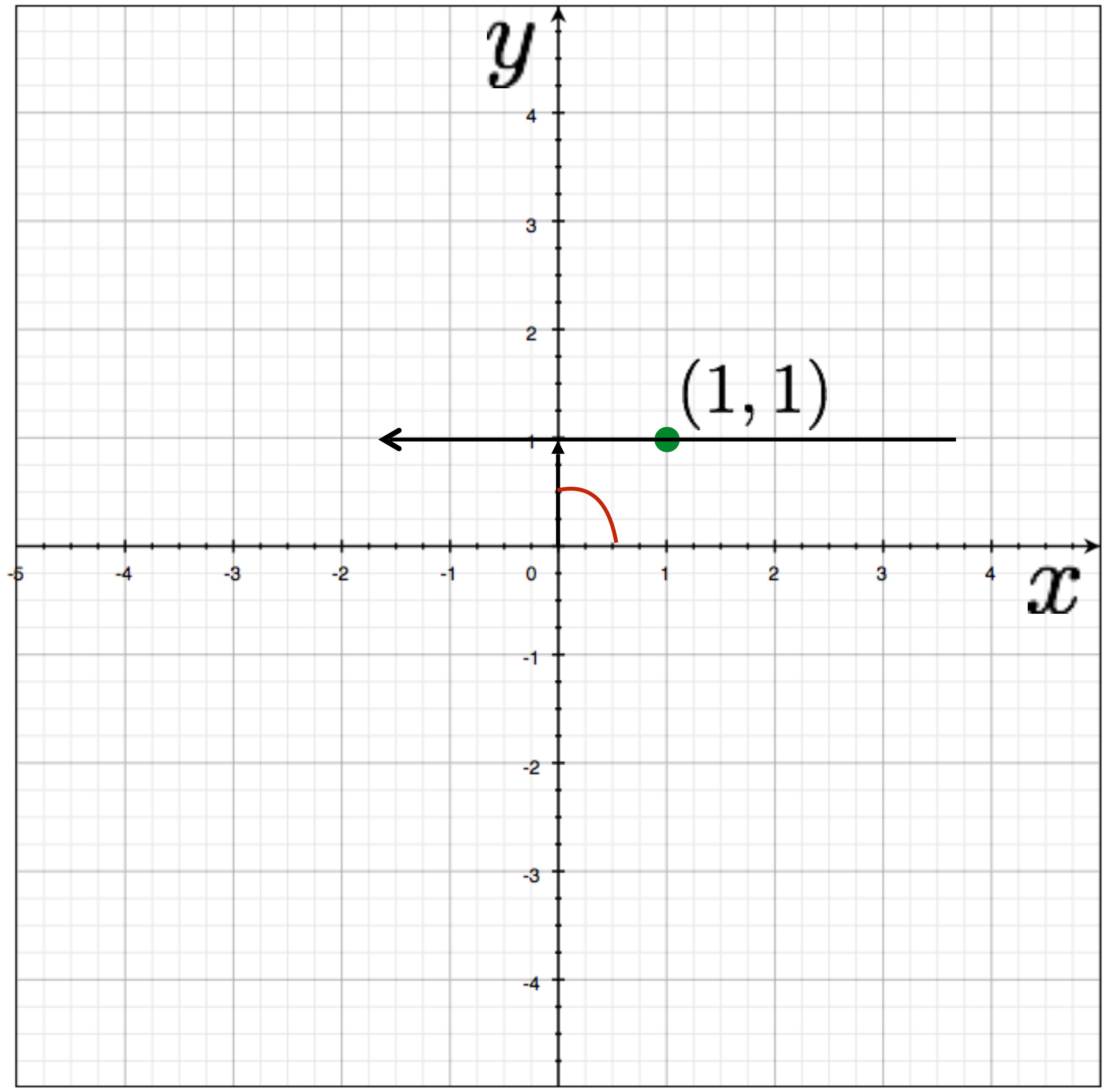
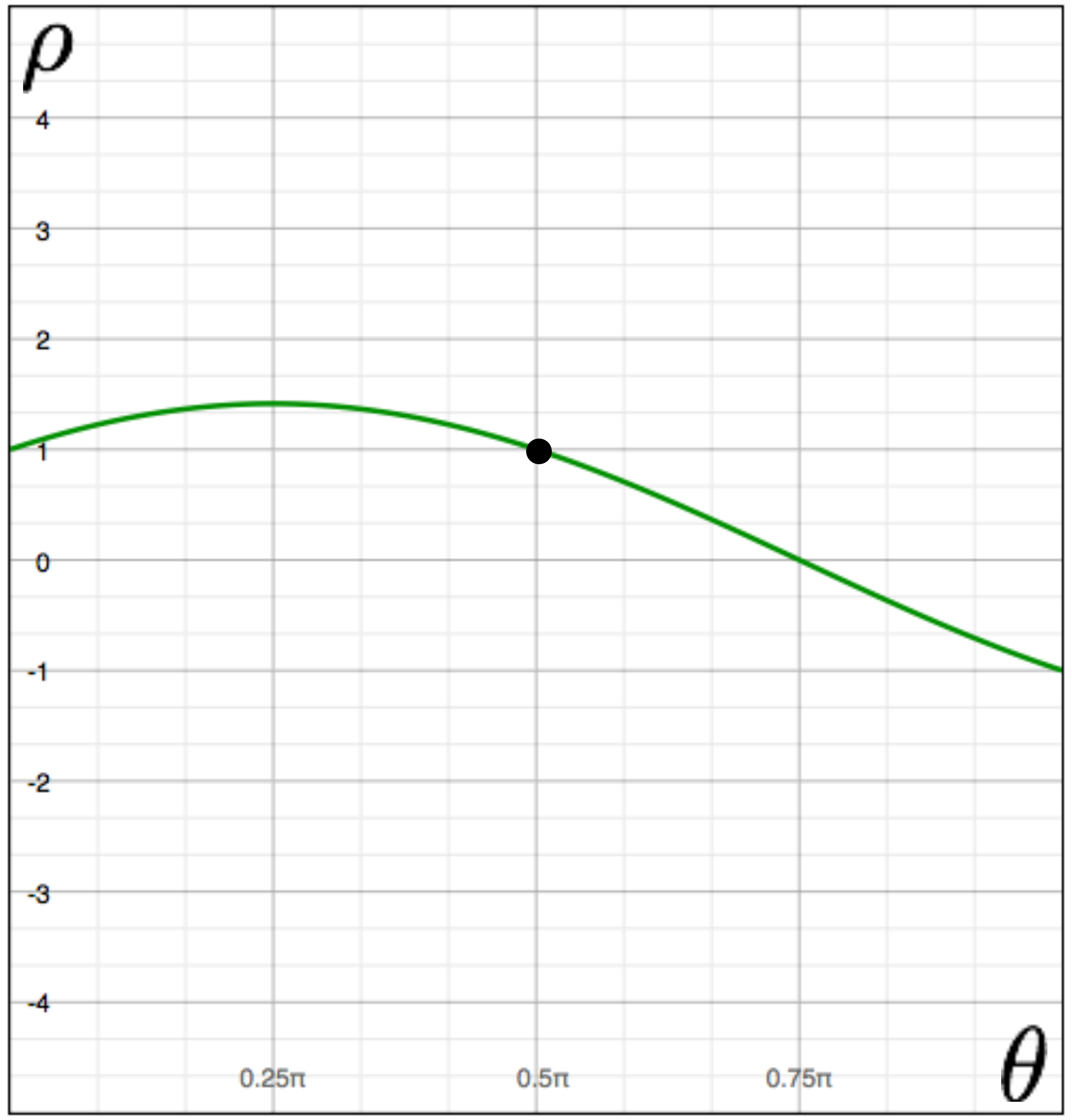


Image space

a line becomes a point



Parameter space

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Hough Transform: Lines

variables

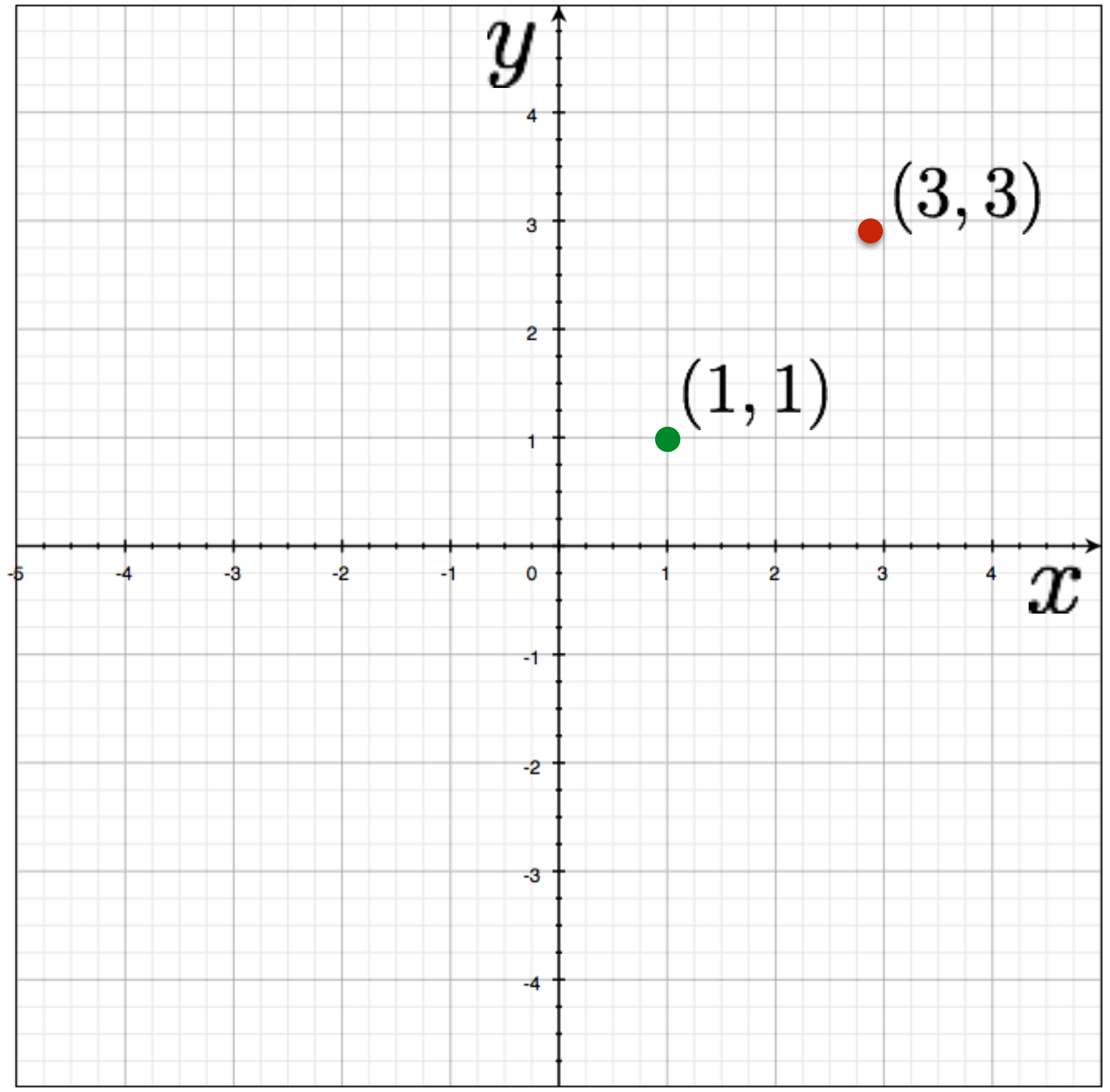
$$y = mx + b$$

parameters

parameters

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

variables



two points become?

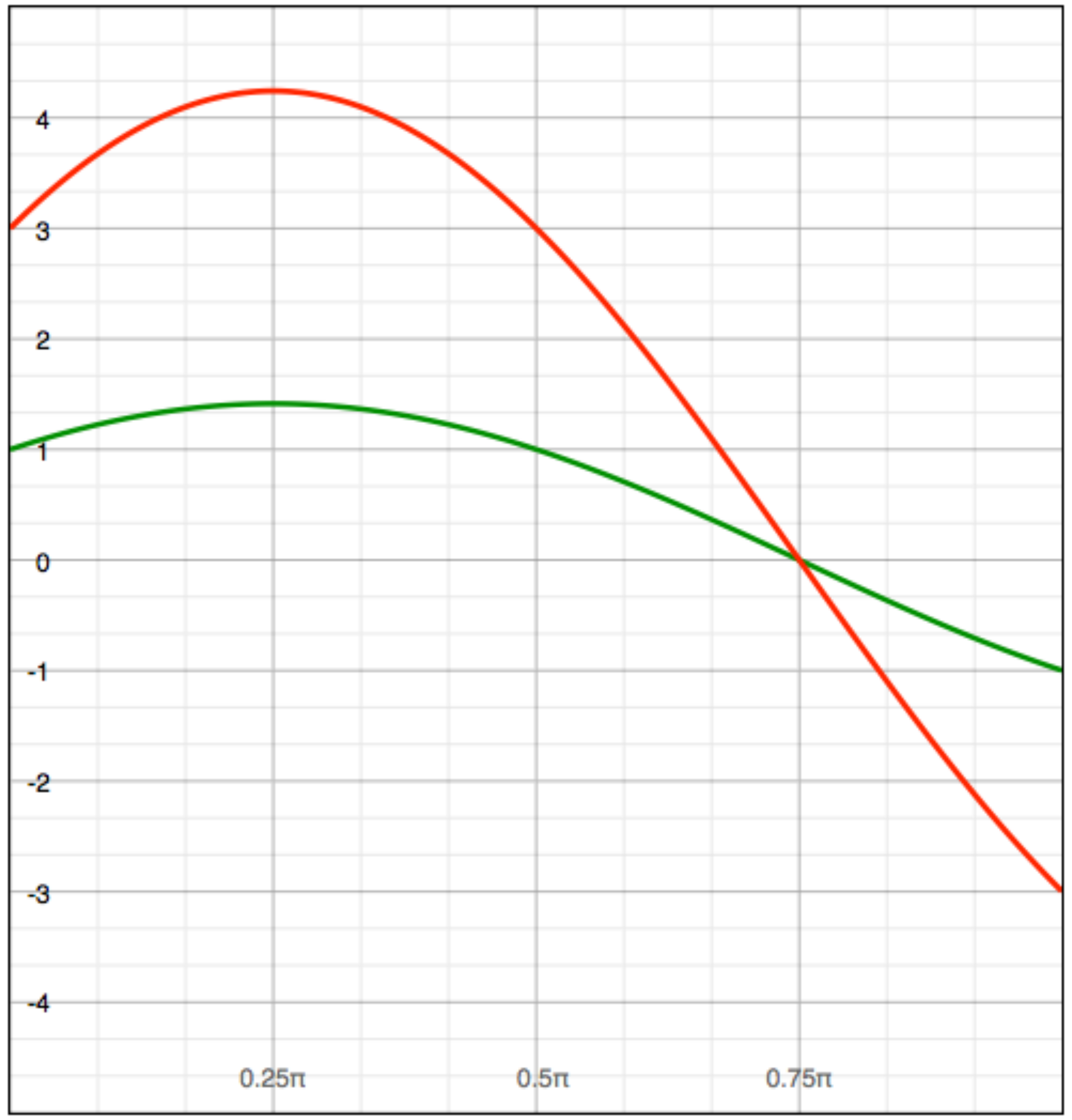


Image space

Parameter space

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Hough Transform: Lines

variables

$$y = mx + b$$

parameters

parameters

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

variables

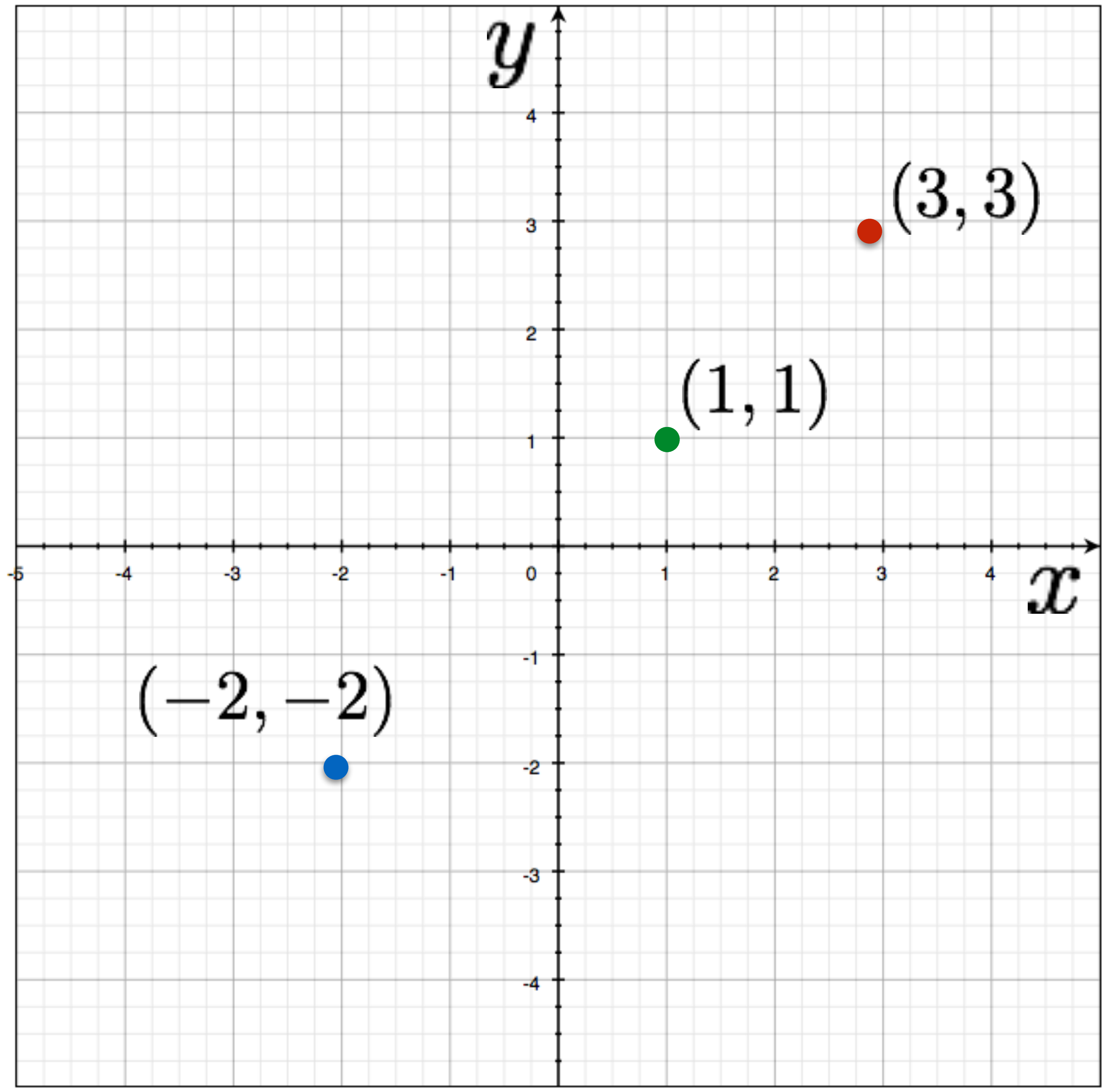
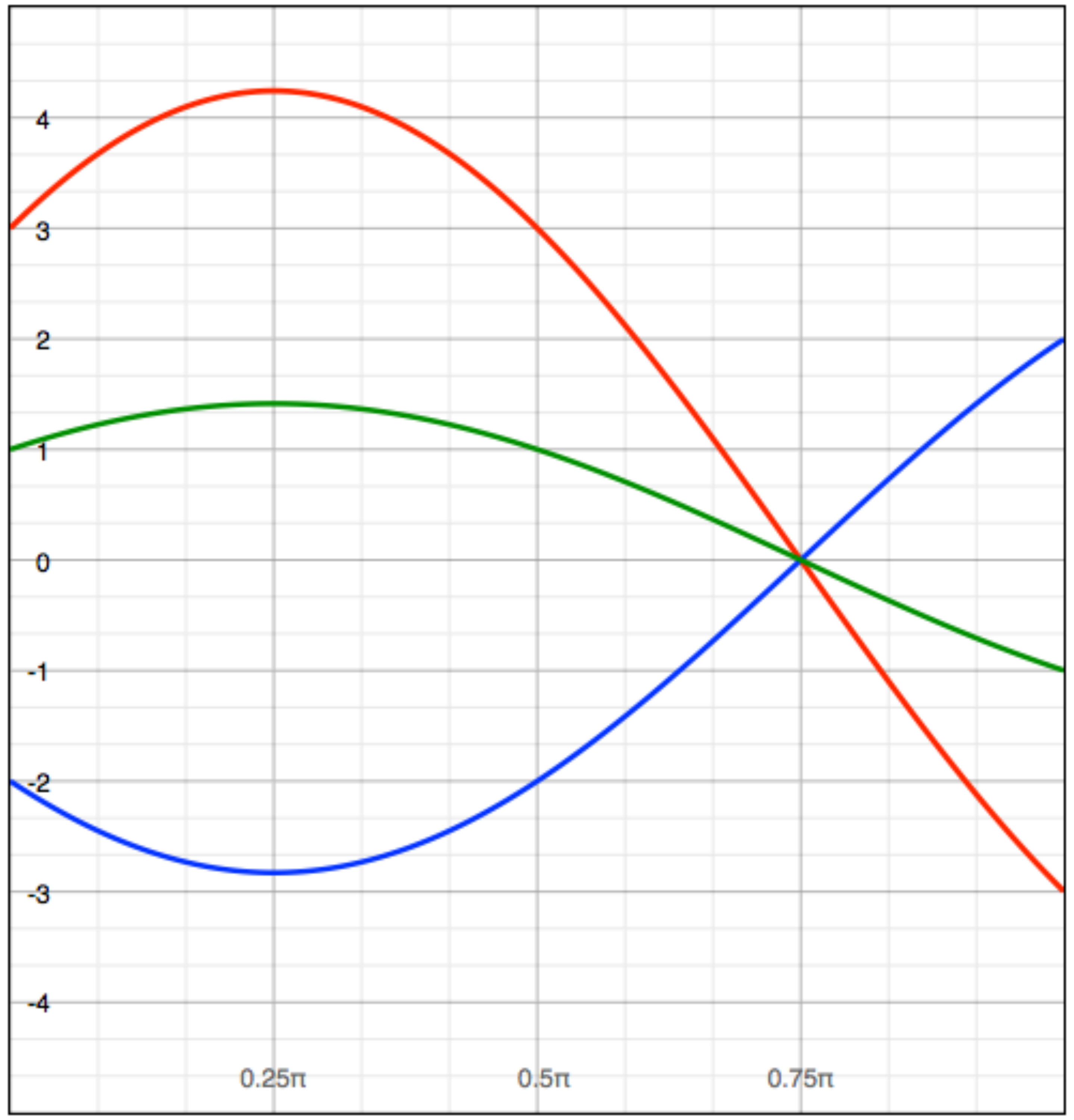


Image space

three points become?



Parameter space

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Hough Transform: Lines

variables

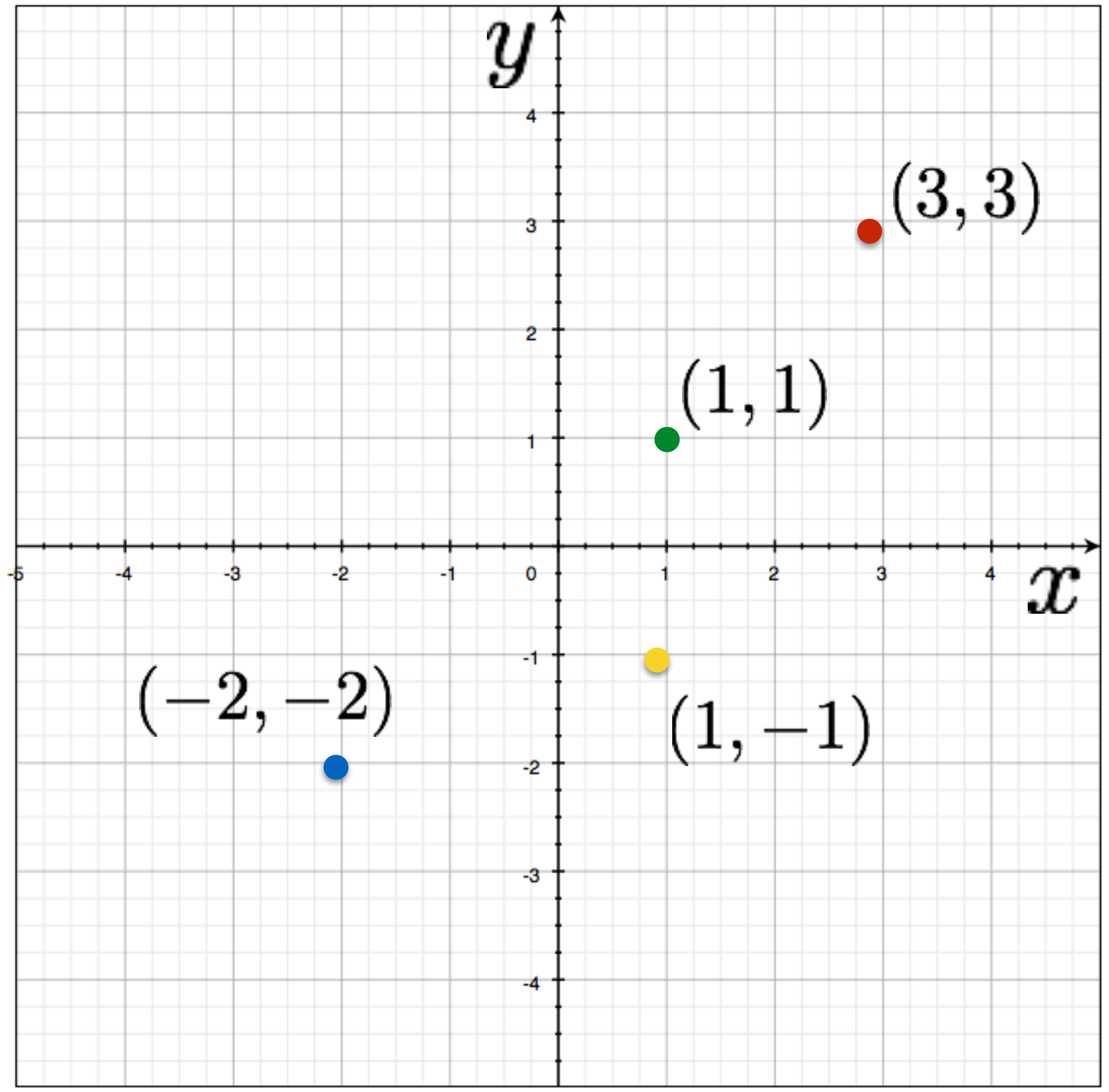
$$y = mx + b$$

parameters

parameters

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

variables



four points become?

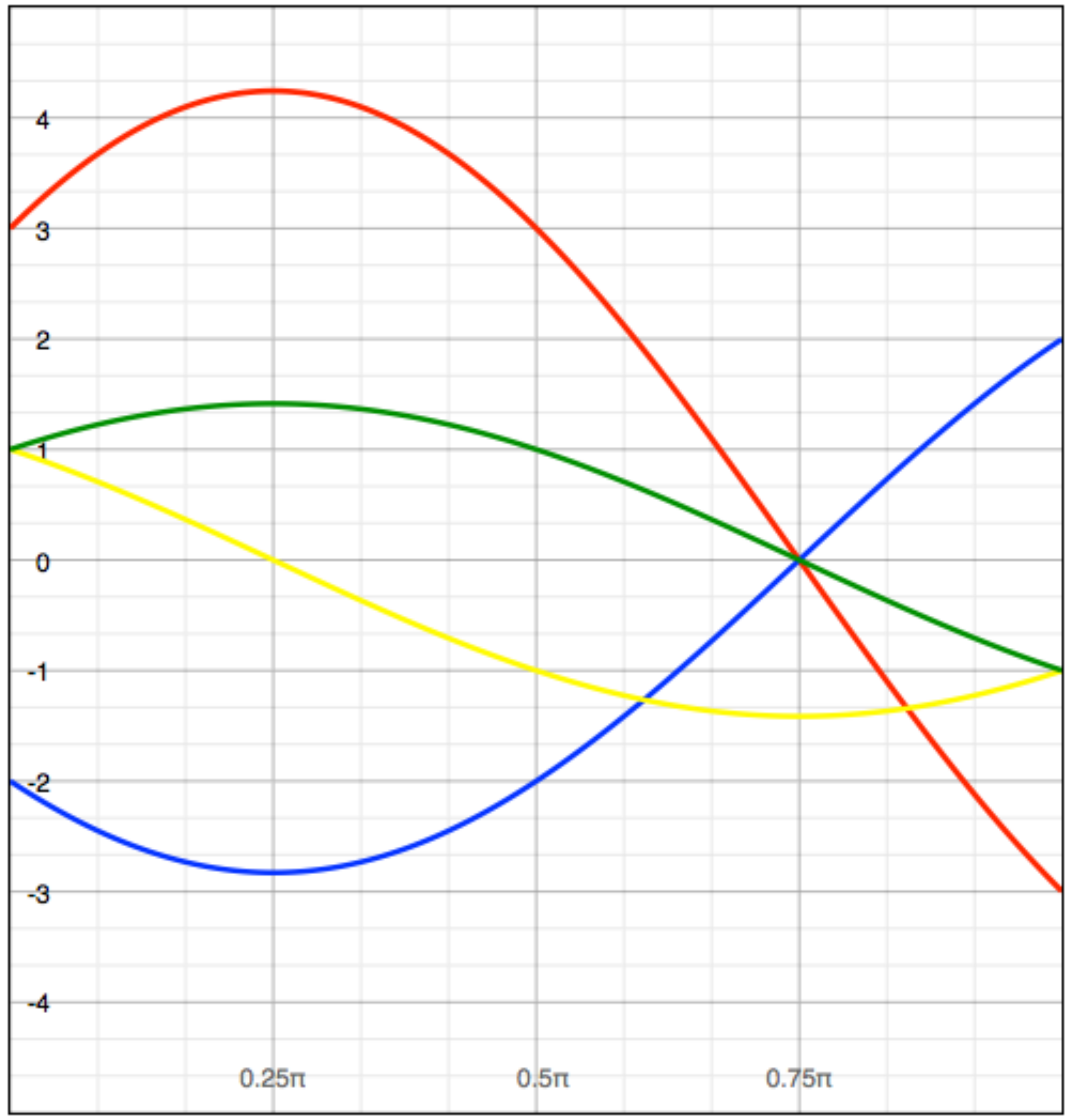


Image space

Parameter space

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Hough Transform for Lines (switching to books notation)

Idea: Each point votes for the lines that pass through it

— A line is the set of points, (x, y), such that

$$x \sin\theta + y \cos\theta + r = 0$$

— Different choices of θ, r give different lines

Hough Transform for Lines (switching to books notation)

Idea: Each point votes for the lines that pass through it

— A line is the set of points, (x, y), such that

$$x \sin\theta + y \cos\theta + r = 0$$

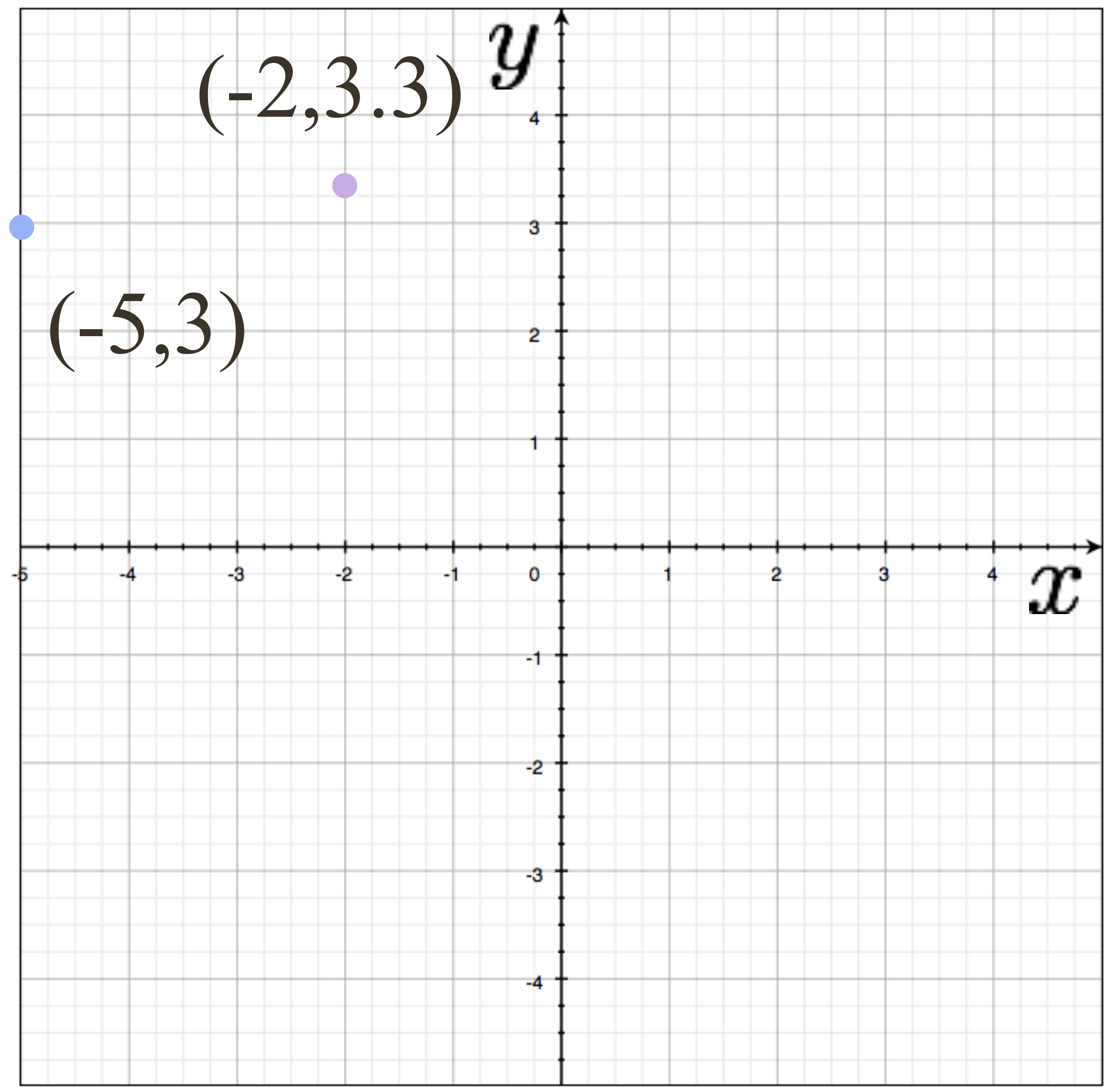
— Different choices of θ, r give different lines

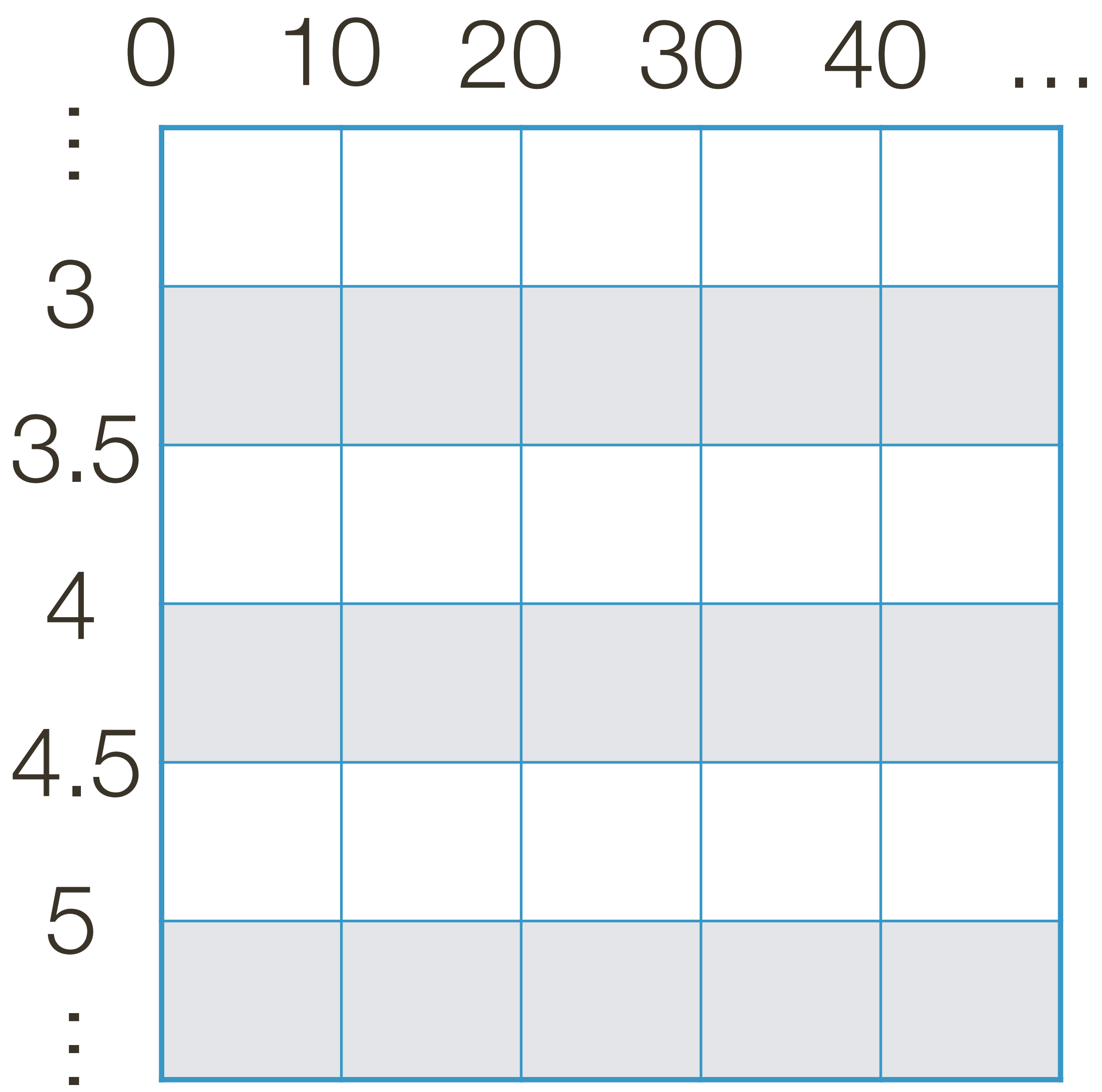
— For any (x, y) there is a one parameter family of lines through this point. Just let (x, y) be constants and for each value of θ the value of r will be determined

— Each point enters votes for each line in the family

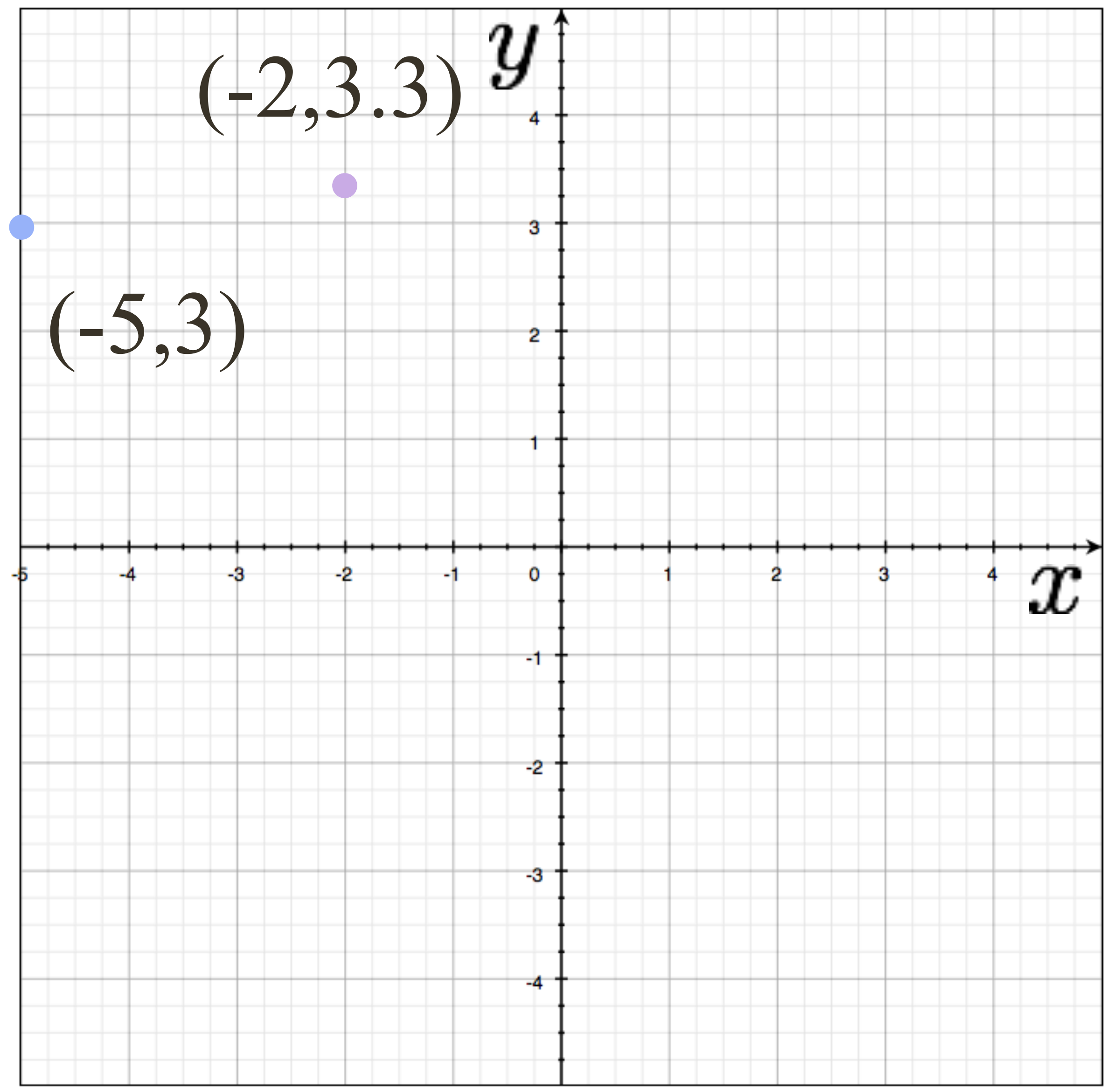
— If there is a line that has lots of votes, that will be the line passing near the points that voted for it

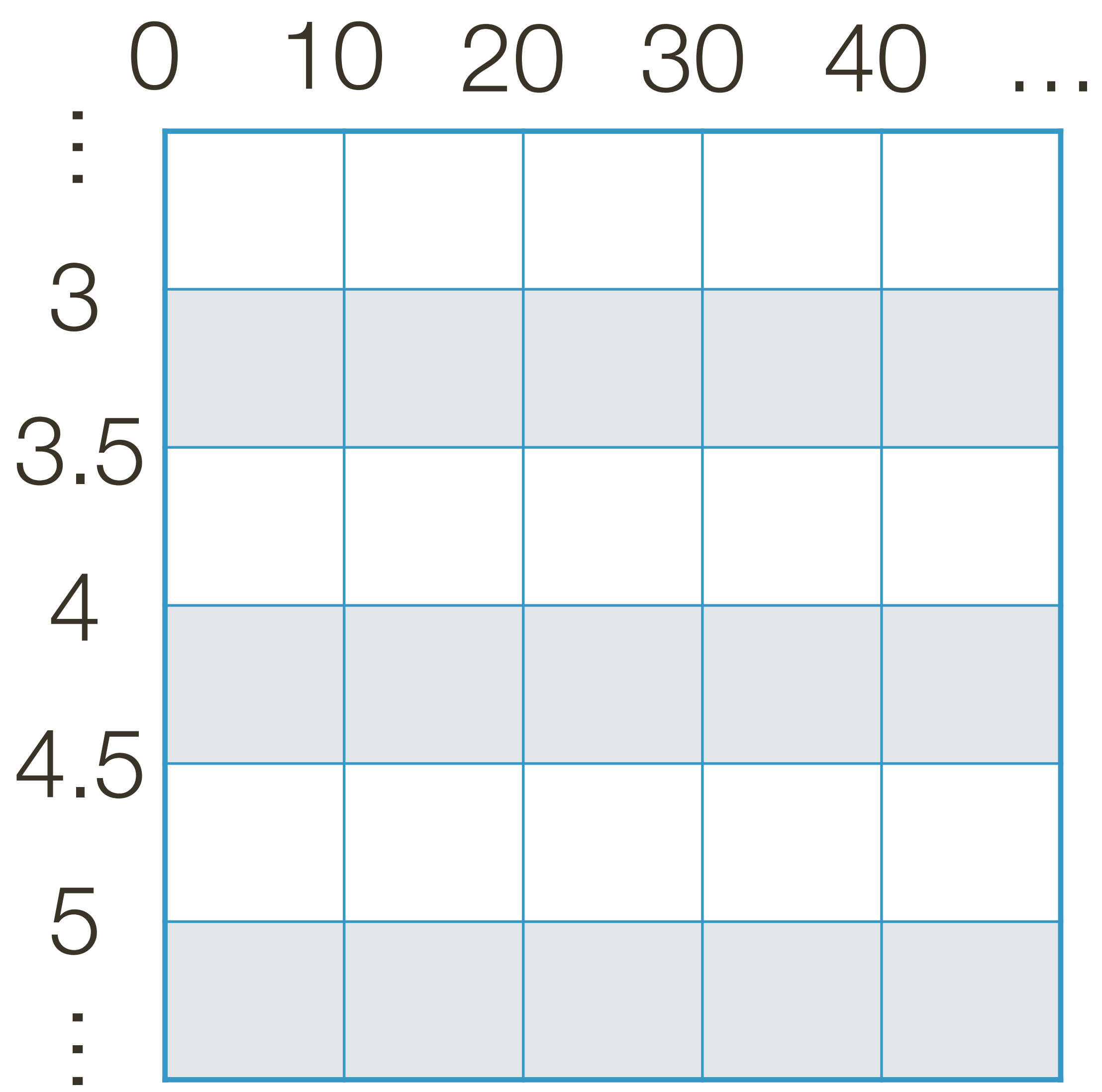
Example: Hough Transform for Lines





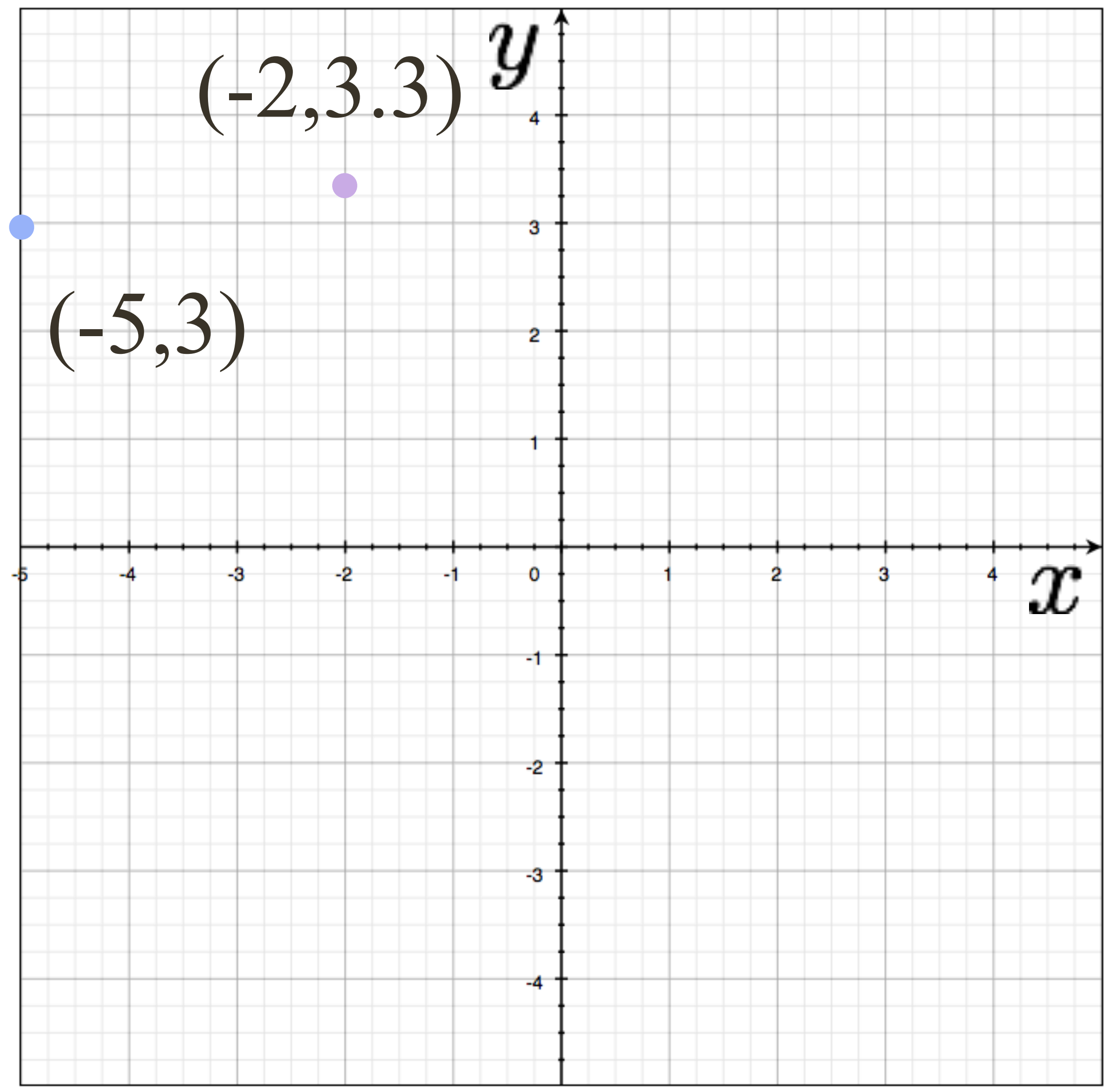
Example: Hough Transform for Lines

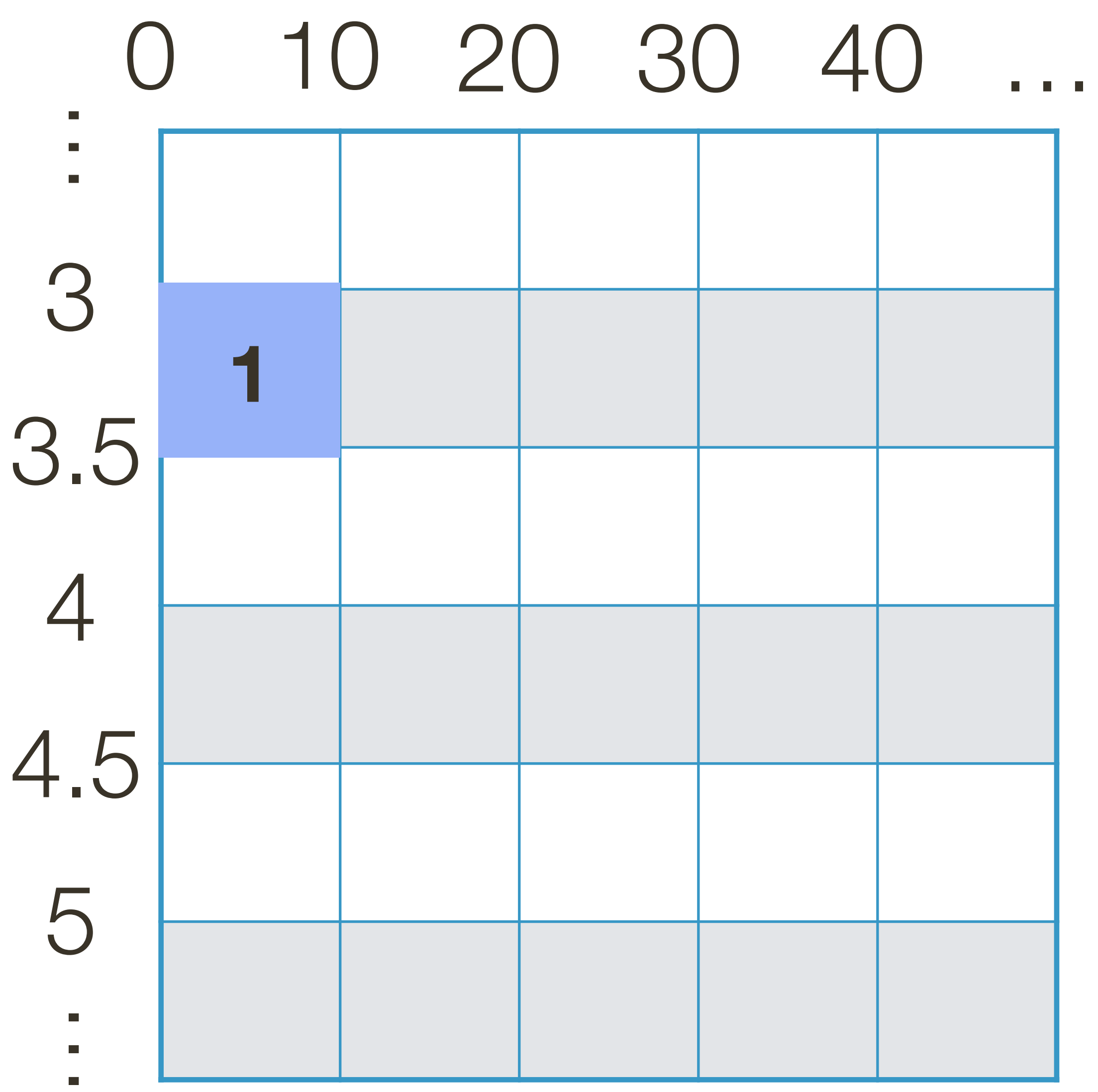




$$-5\sin(5^\circ) - 3\cos(5^\circ) + r = 0 => r = 3.42$$

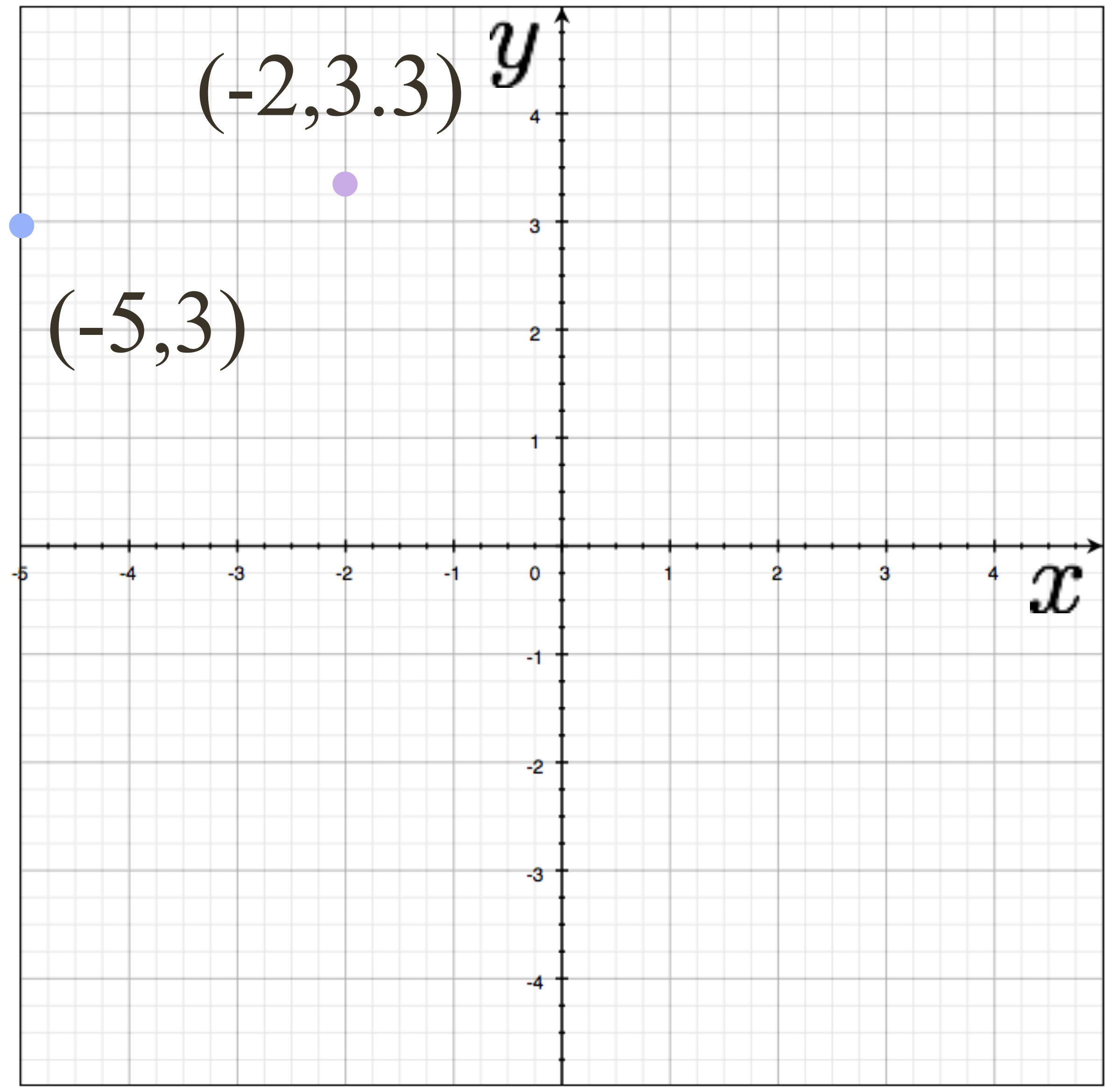
Example: Hough Transform for Lines

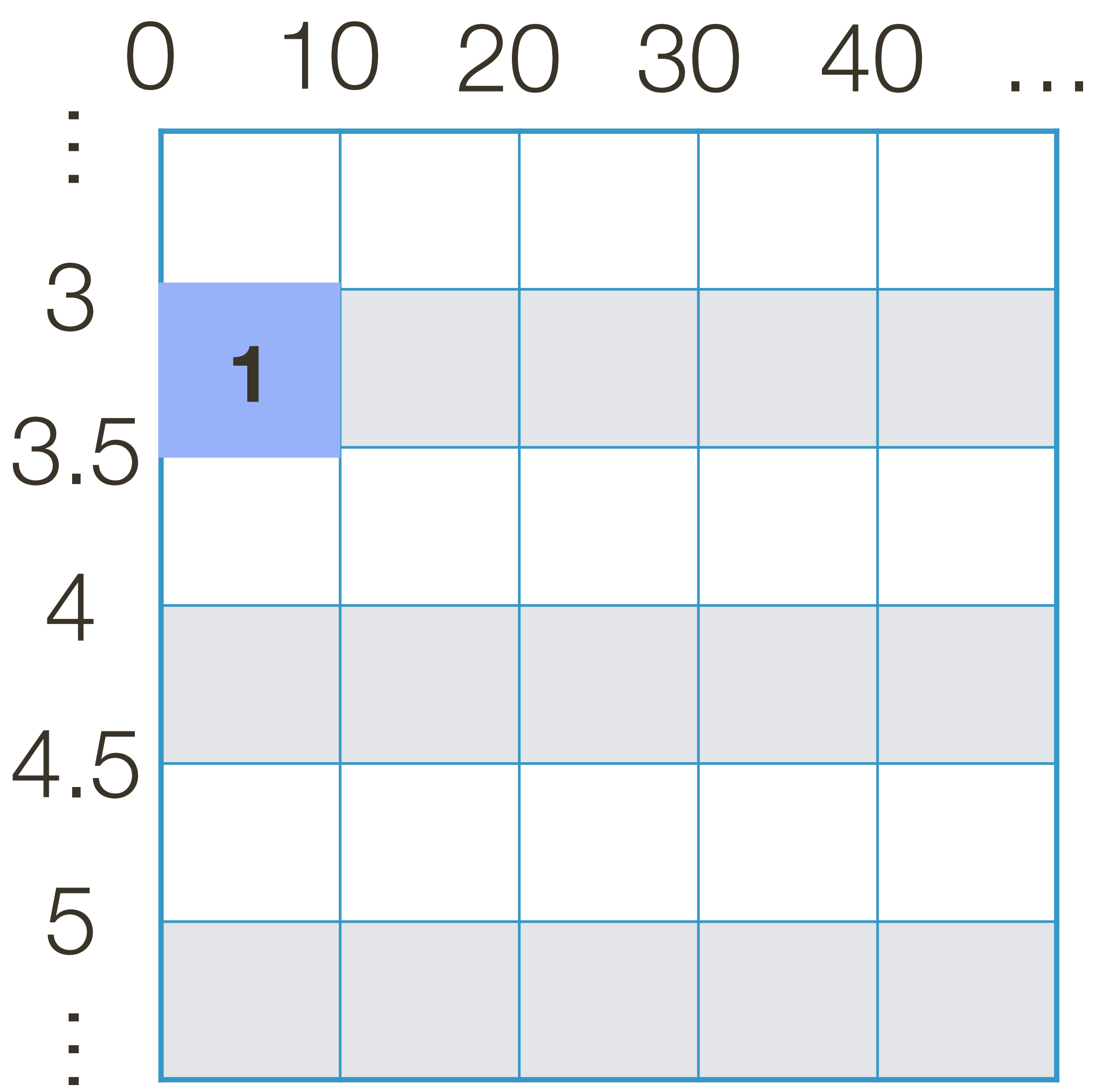




$$-5\sin(5^\circ) - 3\cos(5^\circ) + r = 0 \Rightarrow r = 3.42$$

Example: Hough Transform for Lines

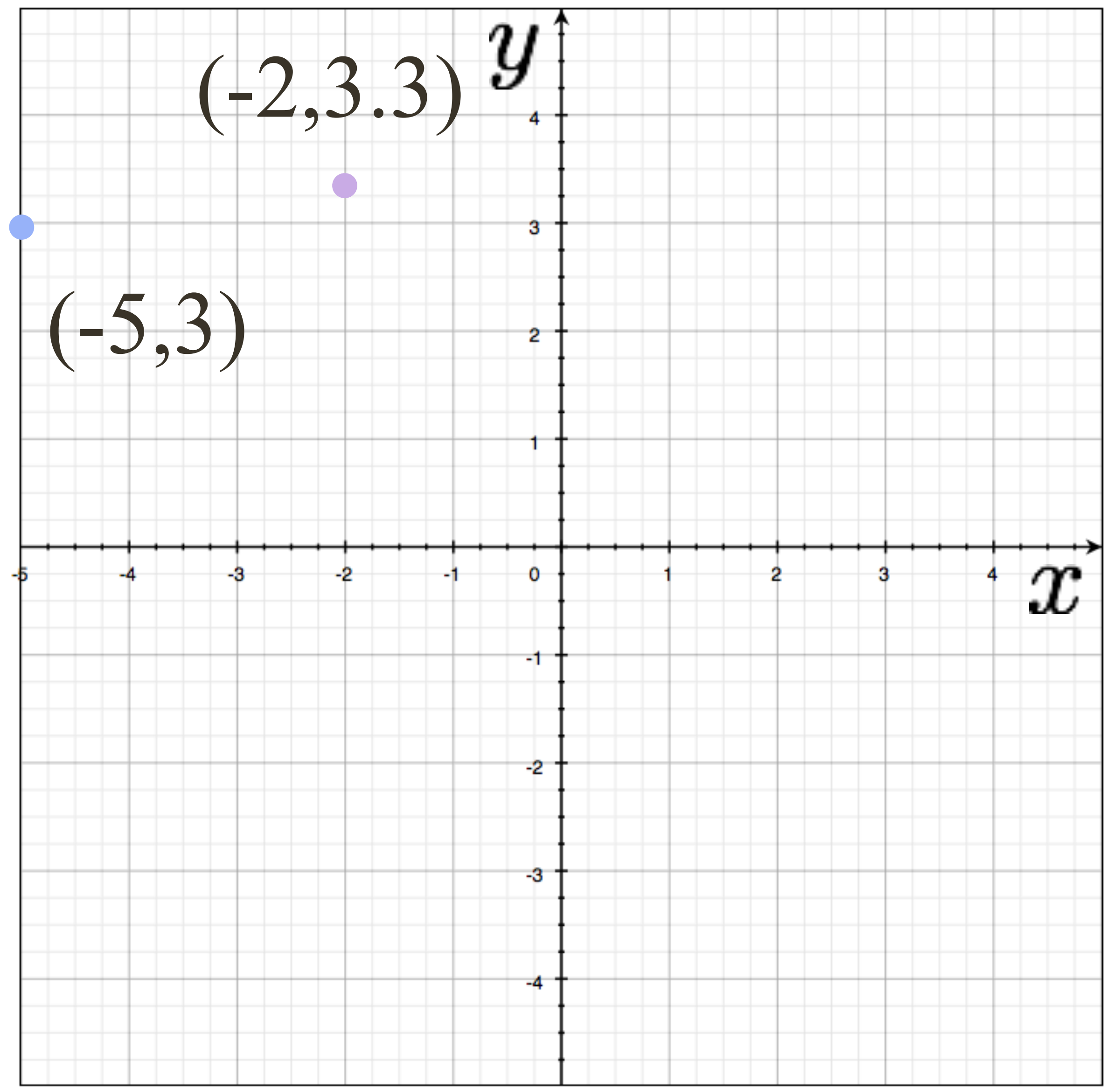


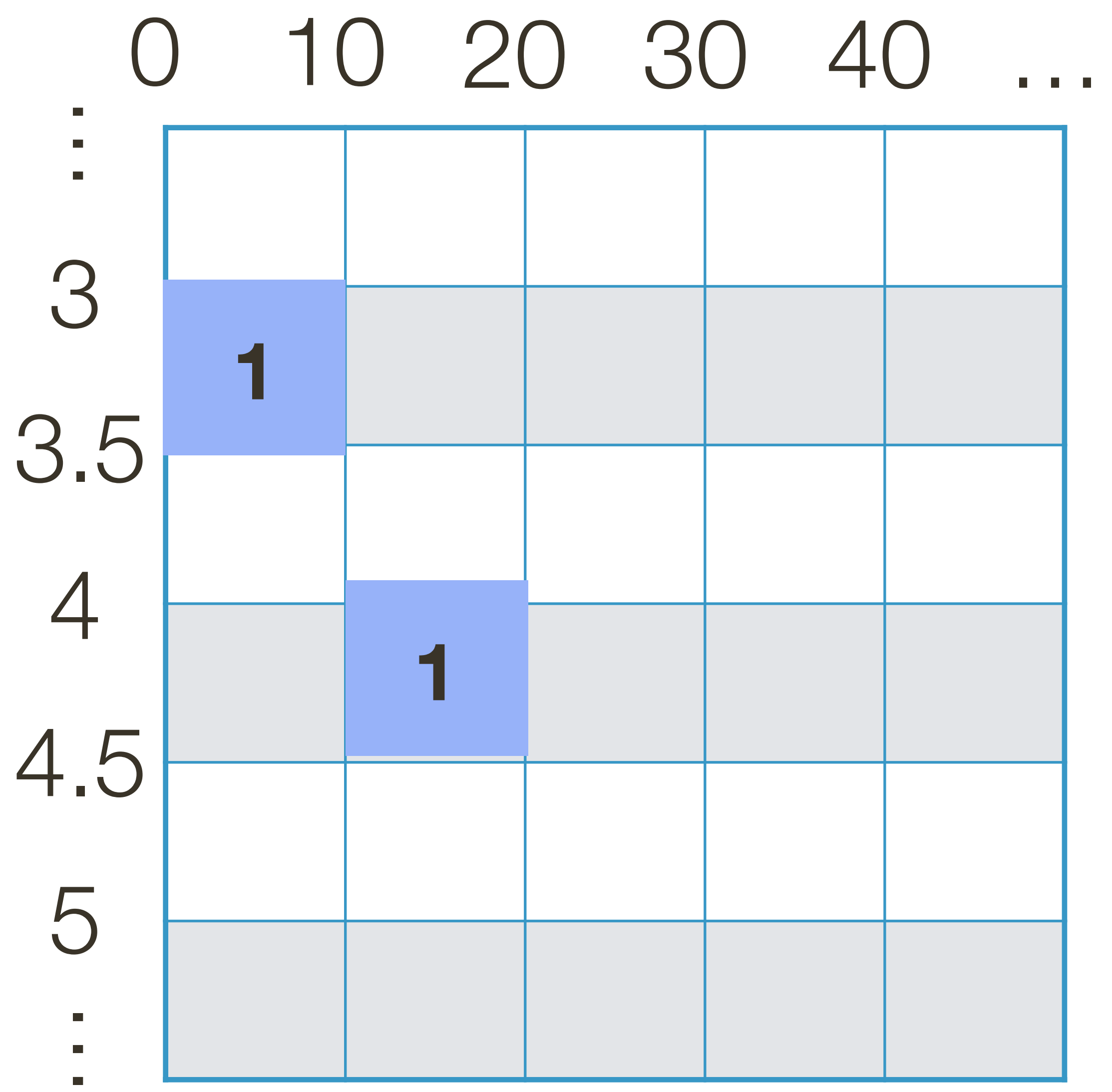


$$-5\sin(5^\circ) - 3\cos(5^\circ) + r = 0 => r = 3.42$$

$$-5\sin(15^\circ) - 3\cos(15^\circ) + r = 0 => r = 4.18$$

Example: Hough Transform for Lines

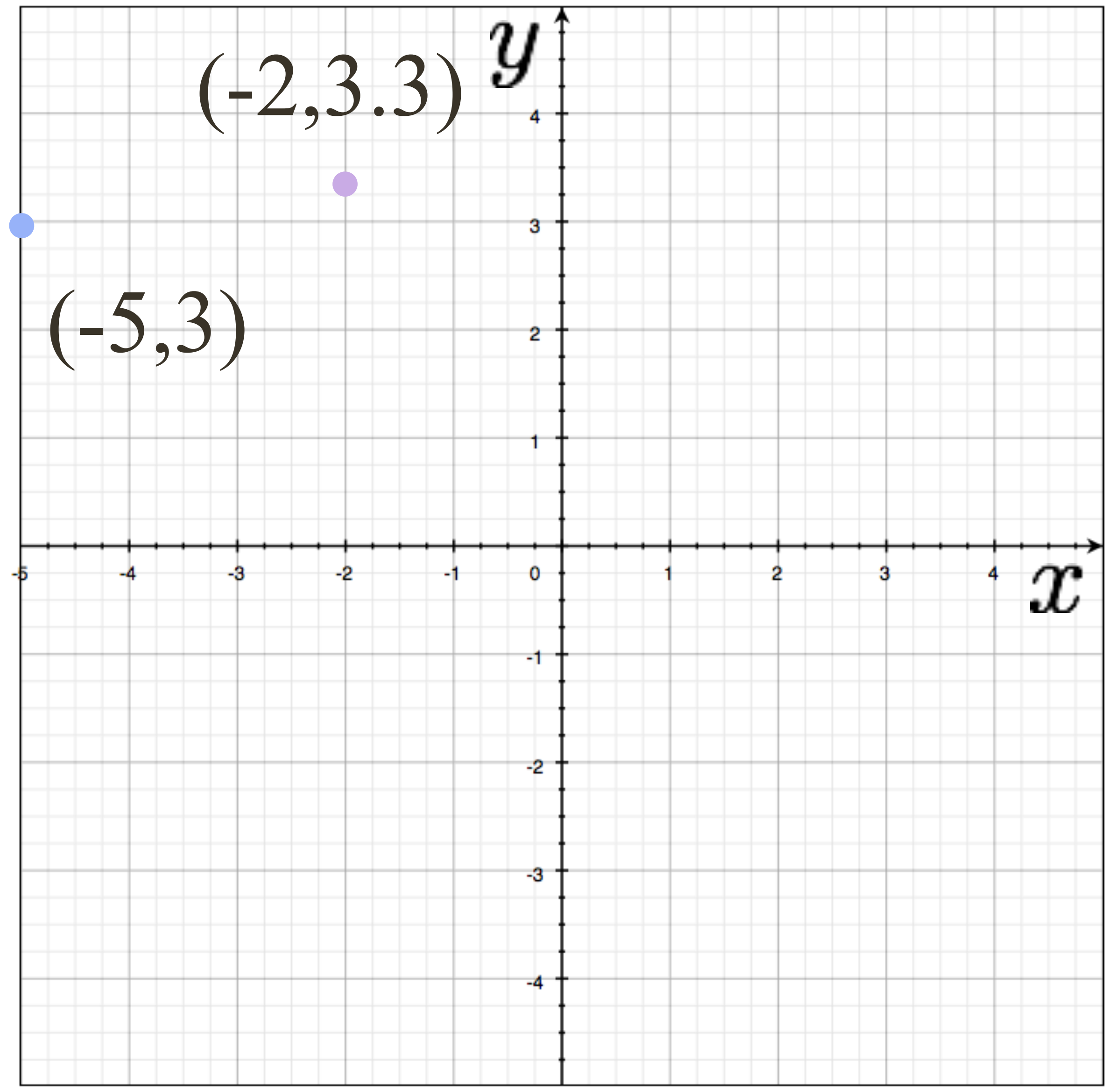


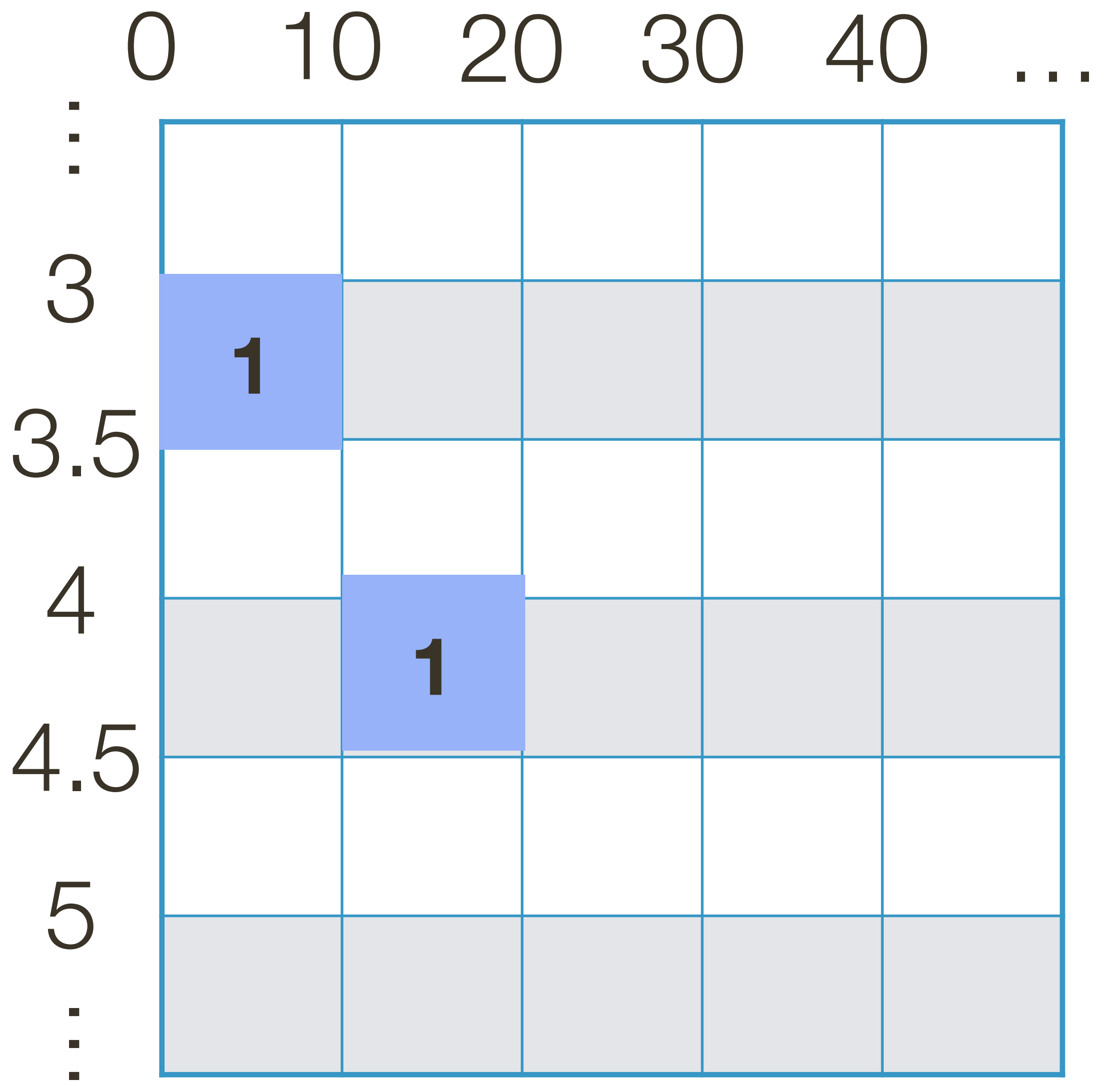


$$-5\sin(5^\circ) - 3\cos(5^\circ) + r = 0 => r = 3.42$$

$$-5\sin(15^\circ) - 3\cos(15^\circ) + r = 0 => r = 4.18$$

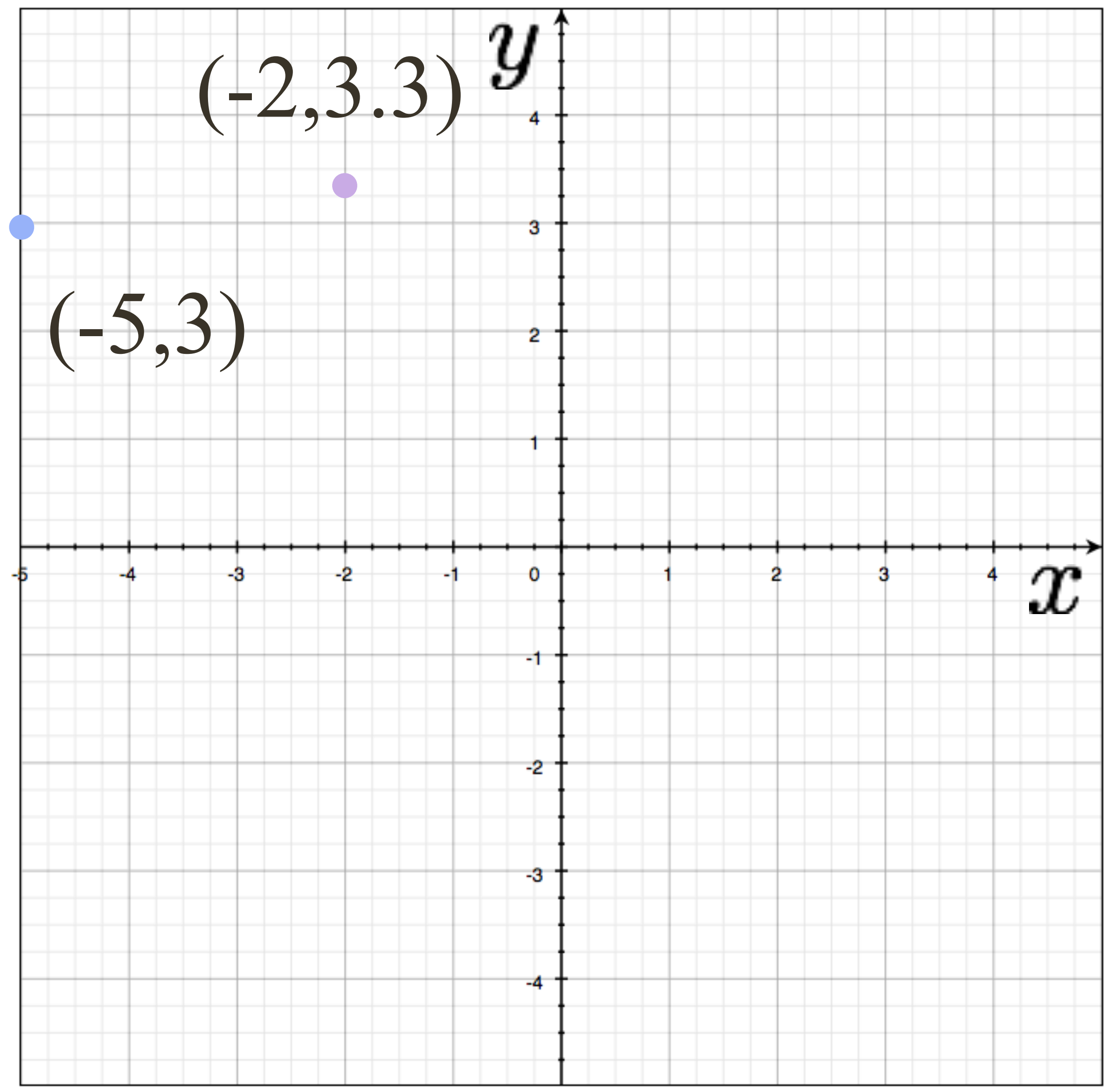
Example: Hough Transform for Lines

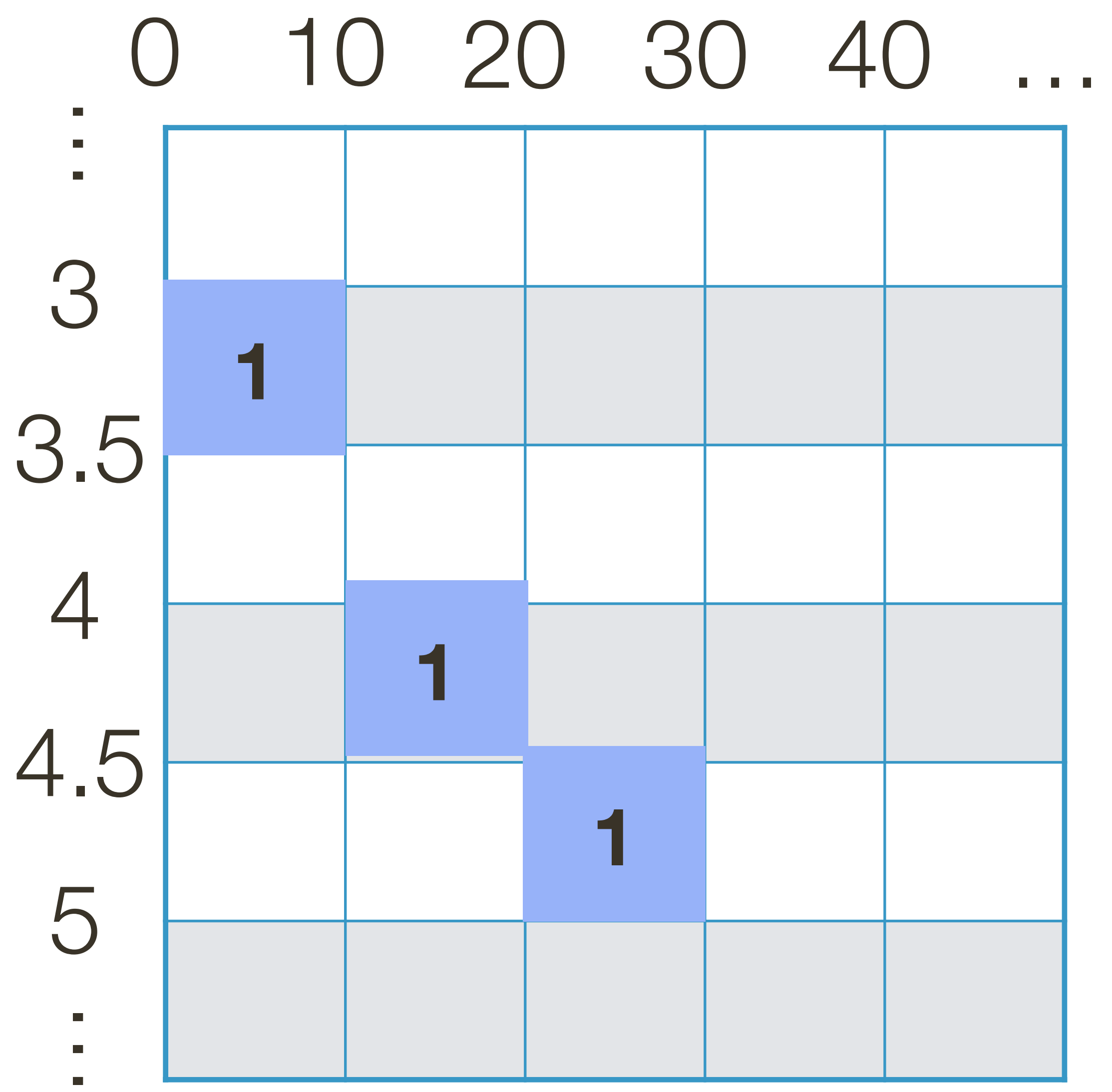




$$-5\sin(5^\circ) - 3\cos(5^\circ) + r = 0 => r = 3.42$$
$$-5\sin(15^\circ) - 3\cos(15^\circ) + r = 0 => r = 4.18$$
$$-5\sin(25^\circ) - 3\cos(25^\circ) + r = 0 => r = 4.83$$

Example: Hough Transform for Lines



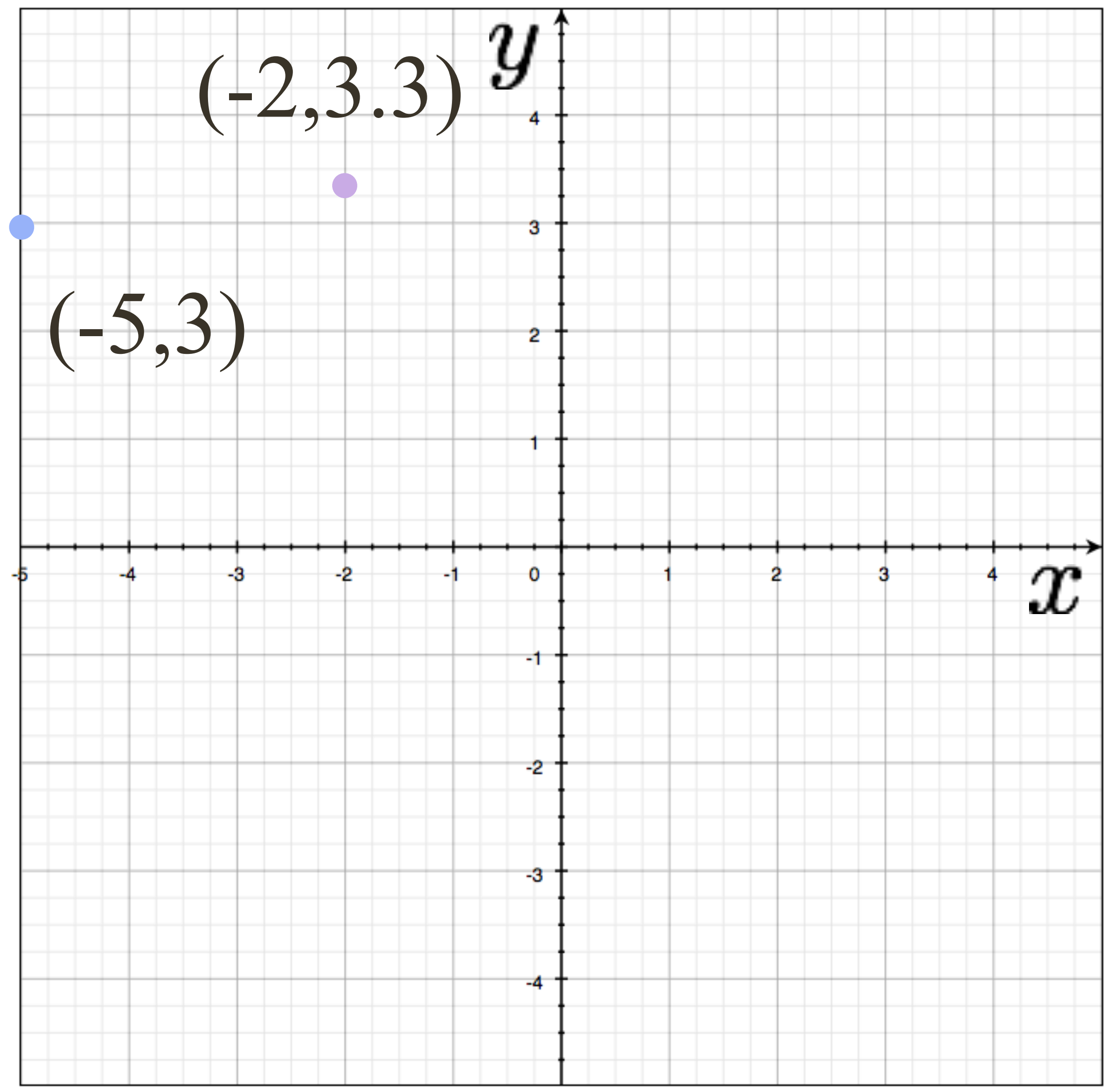


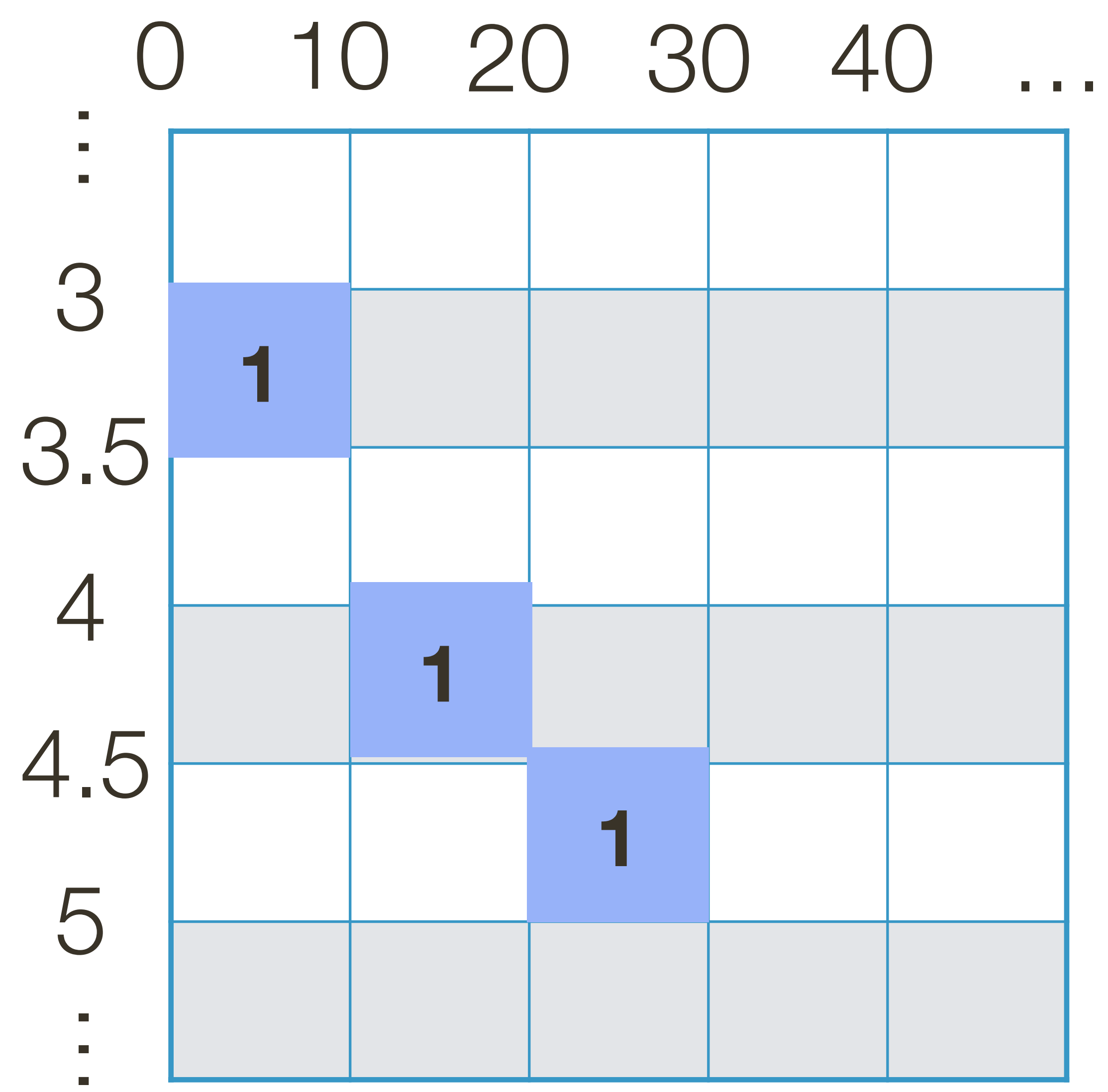
$$-5\sin(5^\circ) - 3\cos(5^\circ) + r = 0 => r = 3.42$$

$$-5\sin(15^\circ) - 3\cos(15^\circ) + r = 0 => r = 4.18$$

$$-5\sin(25^\circ) - 3\cos(25^\circ) + r = 0 => r = 4.83$$

Example: Hough Transform for Lines

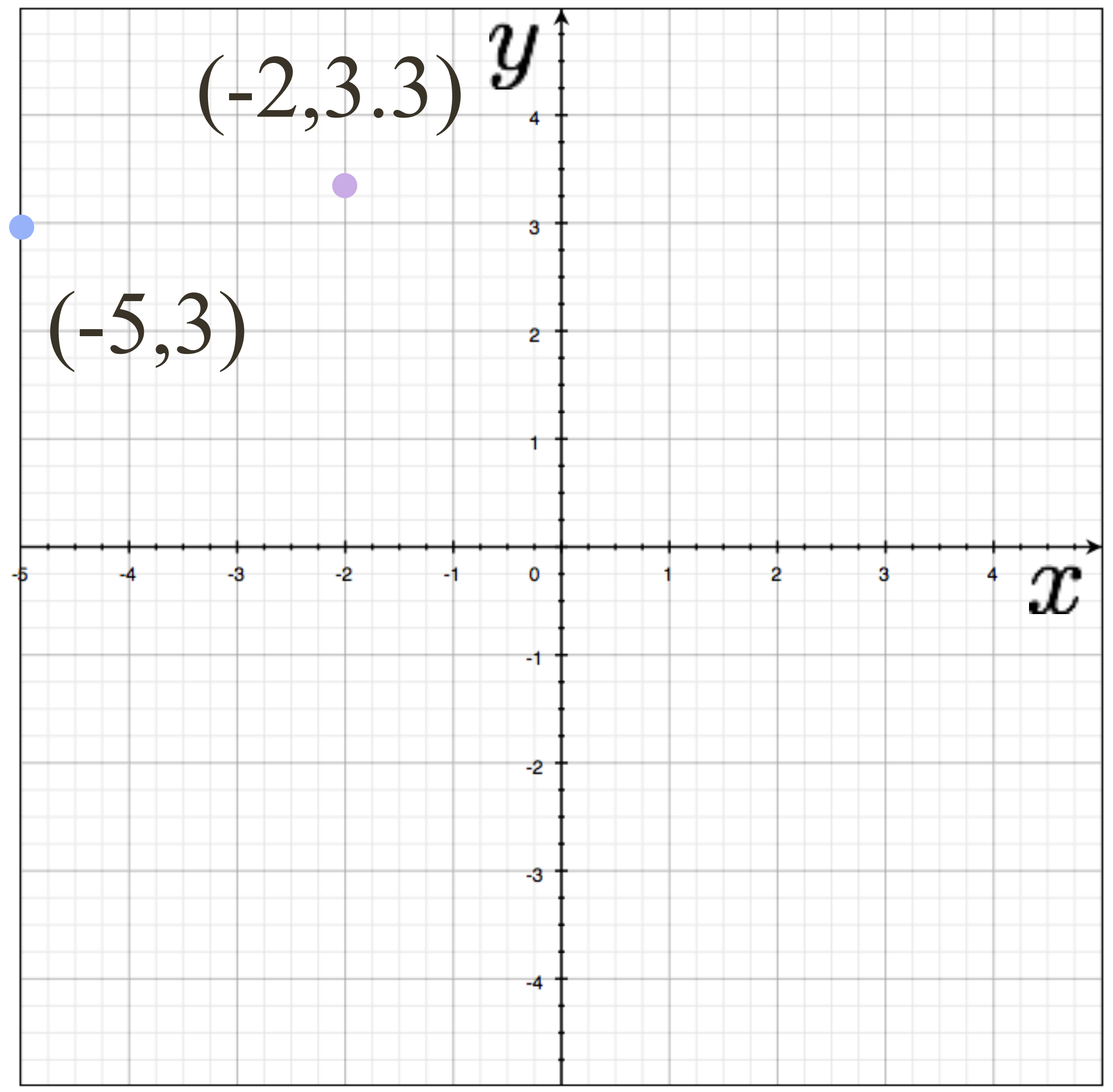


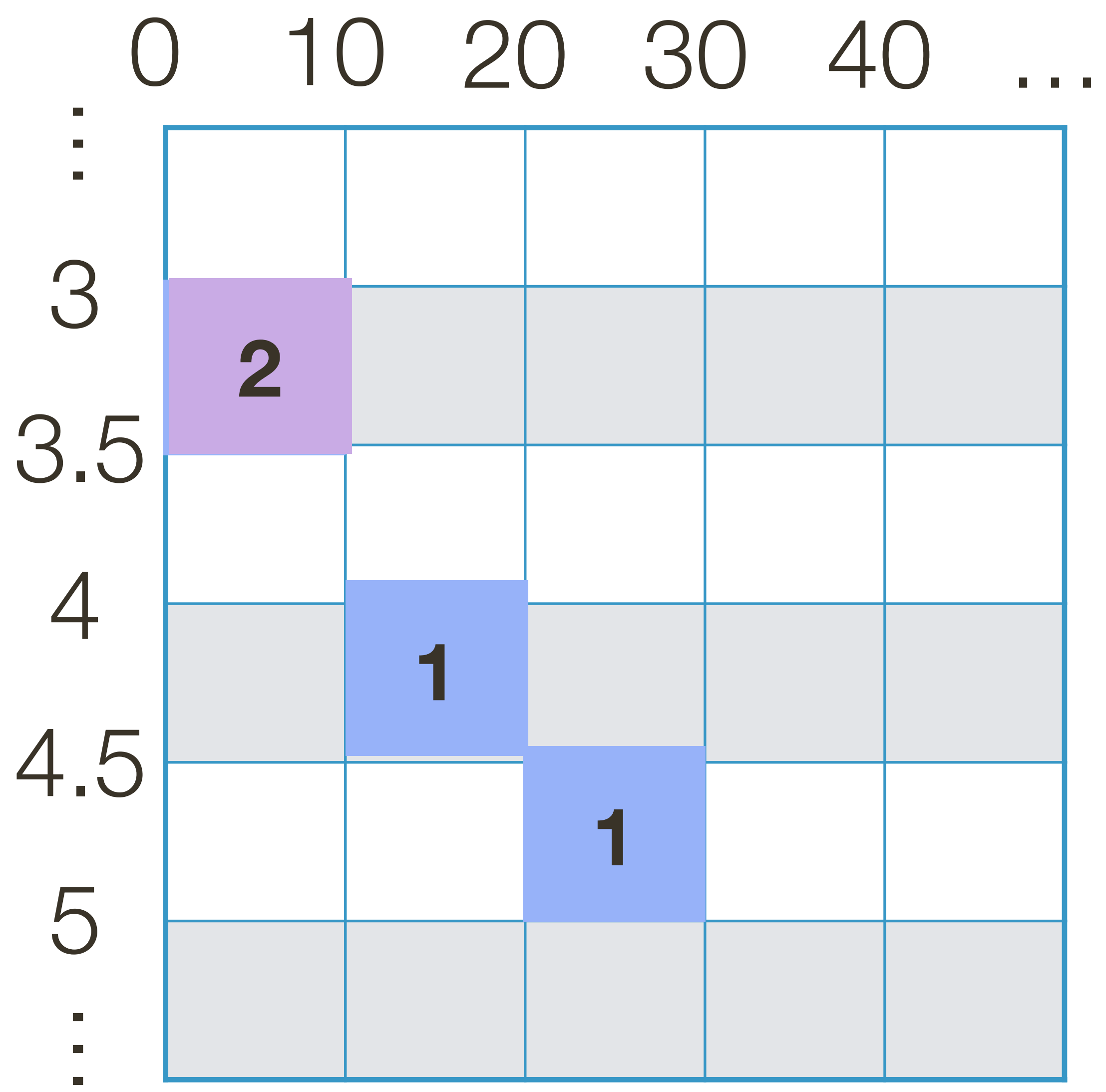


$$-5\sin(5^\circ) - 3\cos(5^\circ) + r = 0 => r = 3.42$$
$$-5\sin(15^\circ) - 3\cos(15^\circ) + r = 0 => r = 4.18$$
$$-5\sin(25^\circ) - 3\cos(25^\circ) + r = 0 => r = 4.83$$

$$-2\sin(5^\circ) - 3.3\cos(5^\circ) + r = 0 => r = 3.46$$

Example: Hough Transform for Lines





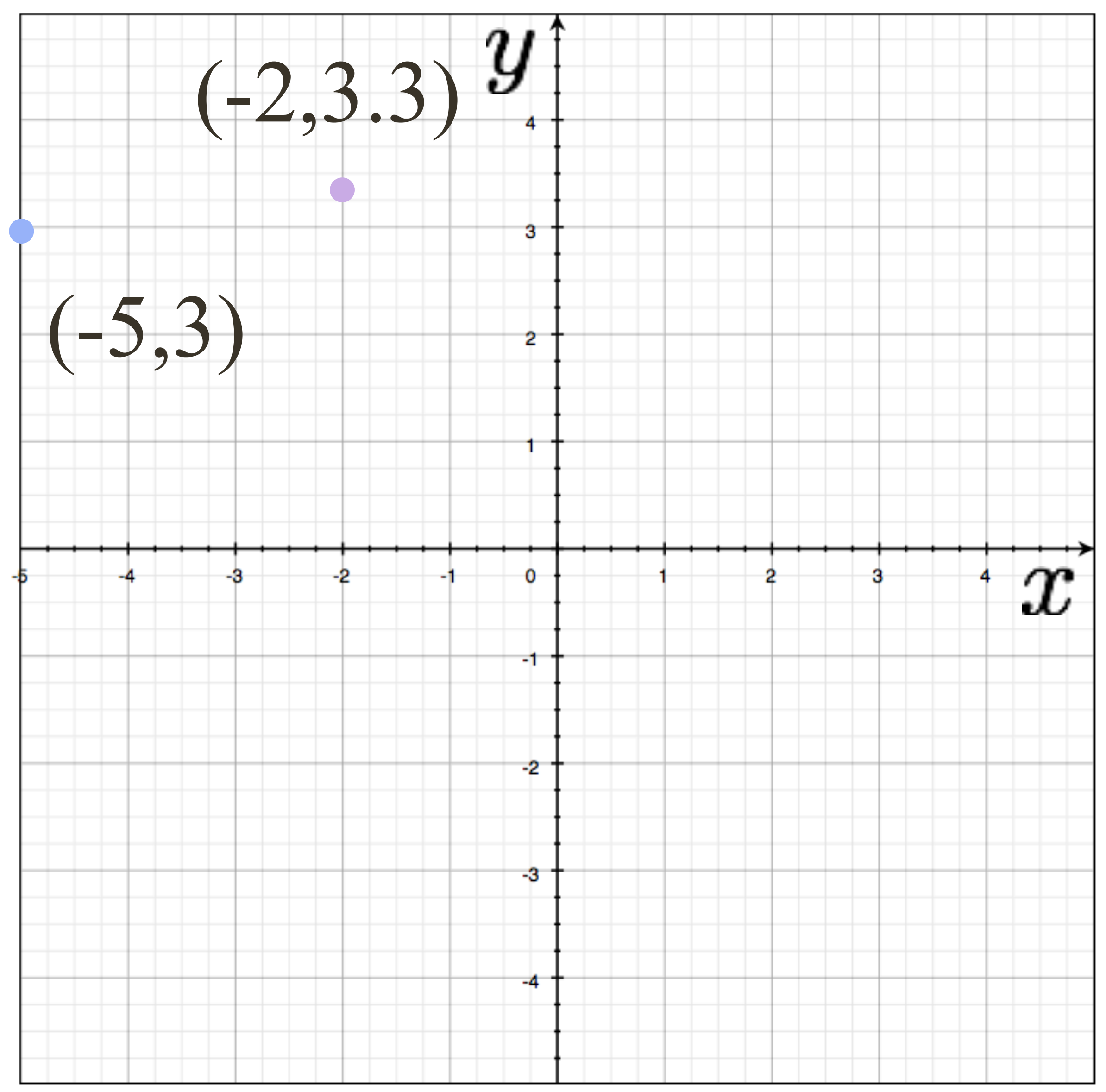
$$-5\sin(5^\circ) - 3\cos(5^\circ) + r = 0 => r = 3.42$$

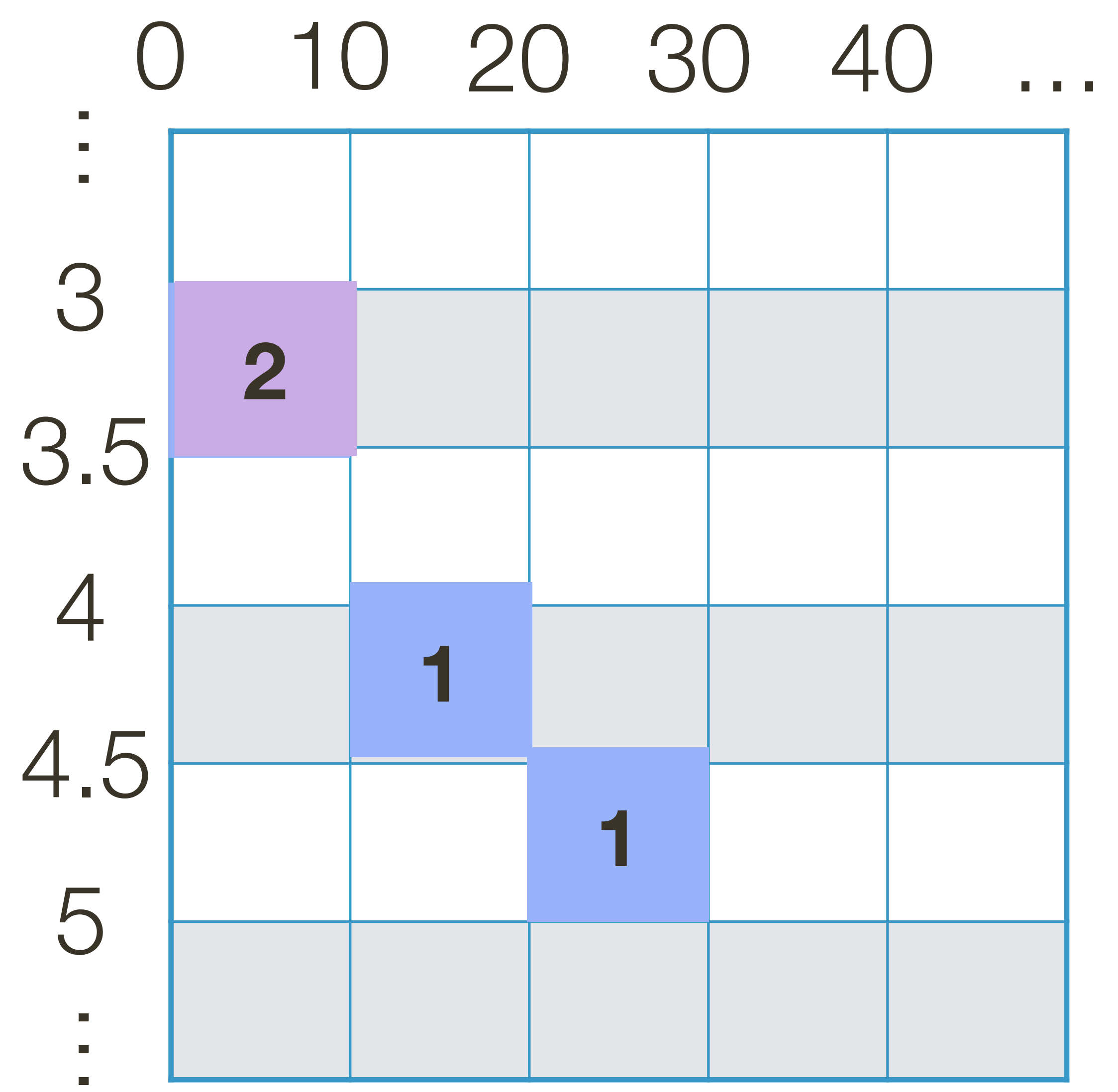
$$-5\sin(15^\circ) - 3\cos(15^\circ) + r = 0 => r = 4.18$$

$$-5\sin(25^\circ) - 3\cos(25^\circ) + r = 0 => r = 4.83$$

$$-2\sin(5^\circ) - 3.3\cos(5^\circ) + r = 0 => r = 3.46$$

Example: Hough Transform for Lines

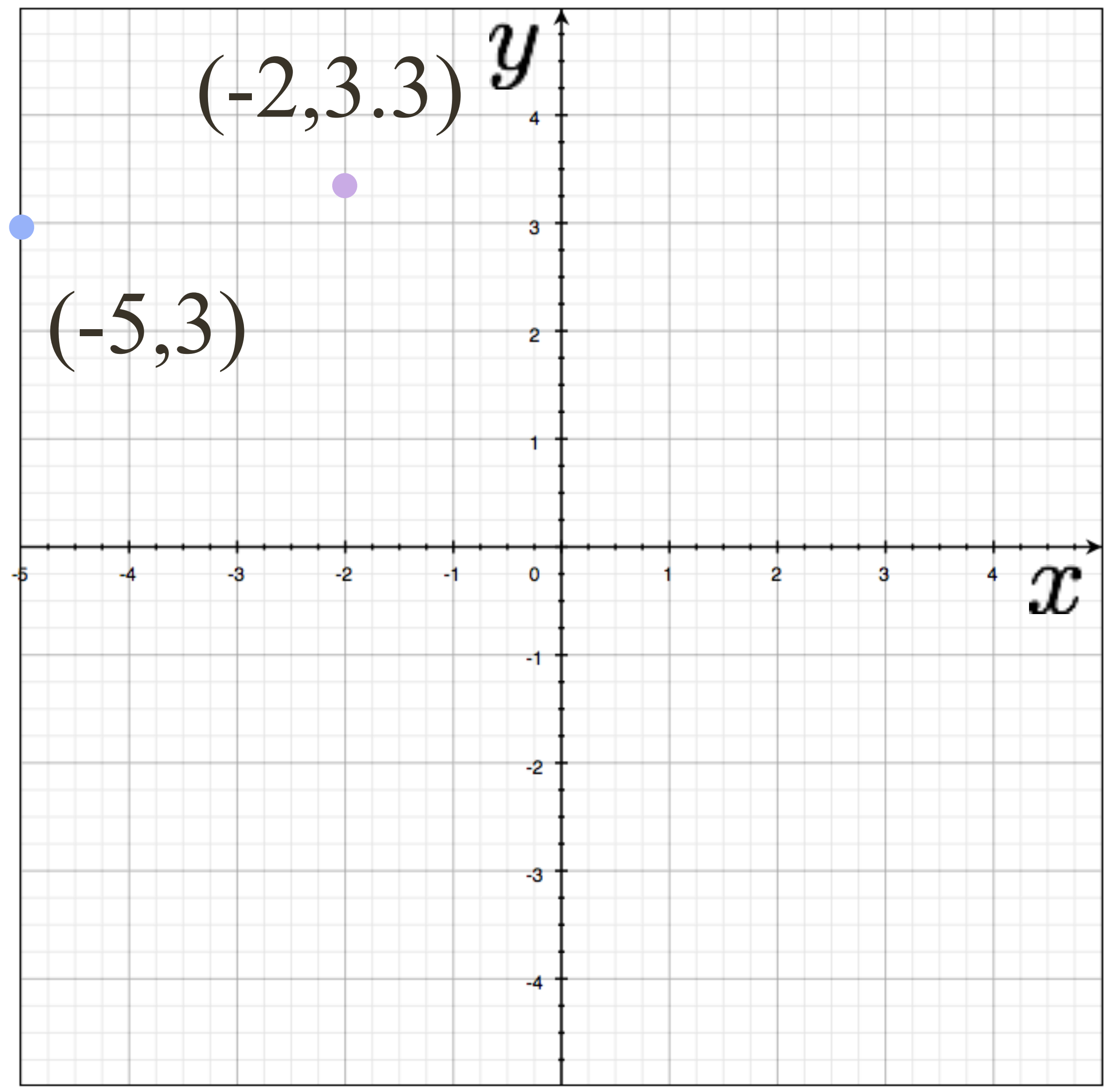


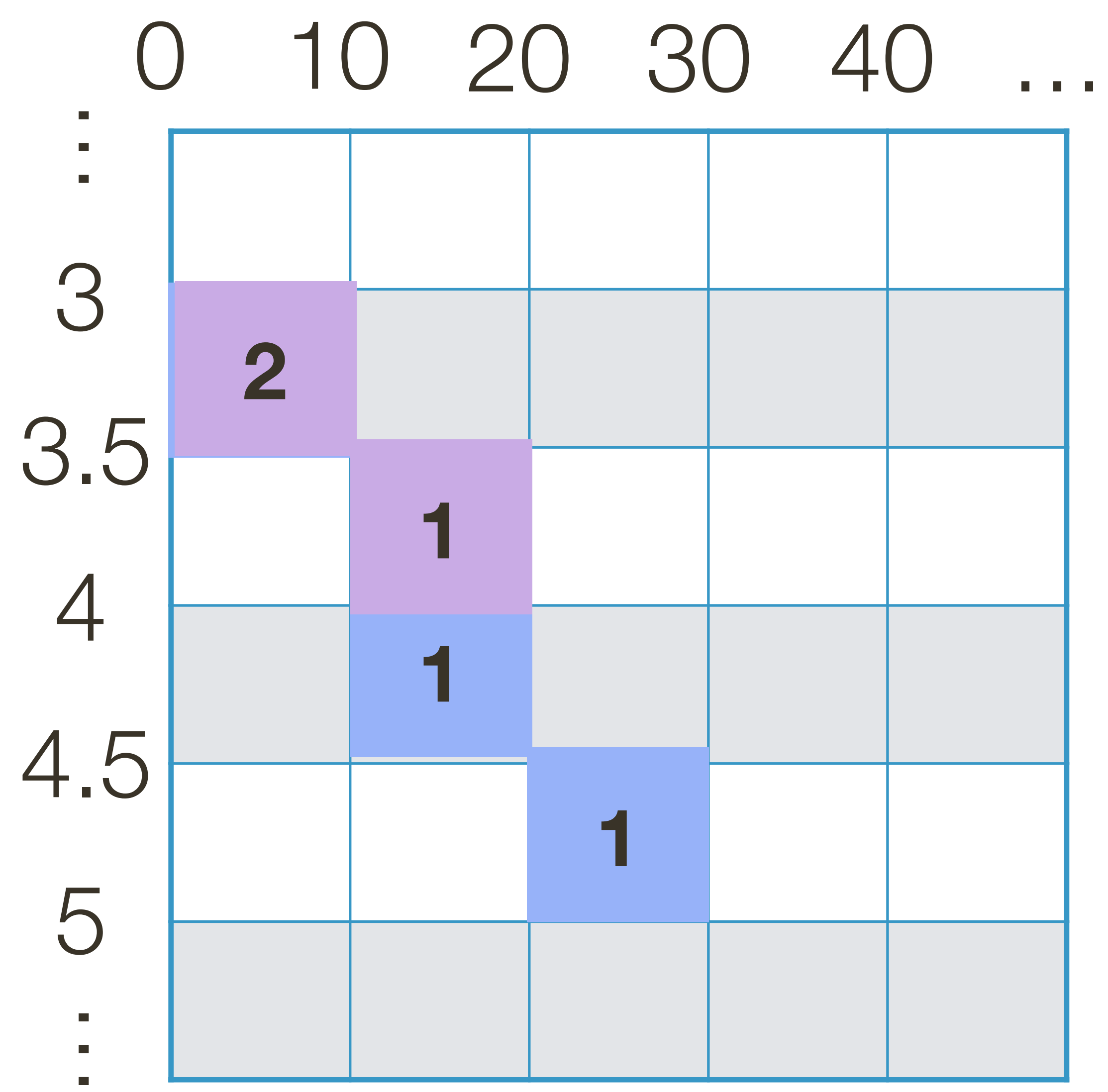


$$-5\sin(5^\circ) - 3\cos(5^\circ) + r = 0 => r = 3.42$$
$$-5\sin(15^\circ) - 3\cos(15^\circ) + r = 0 => r = 4.18$$
$$-5\sin(25^\circ) - 3\cos(25^\circ) + r = 0 => r = 4.83$$

$$-2\sin(5^\circ) - 3.3\cos(5^\circ) + r = 0 => r = 3.46$$
$$-2\sin(15^\circ) - 3.3\cos(15^\circ) + r = 0 => r = 3.71$$

Example: Hough Transform for Lines

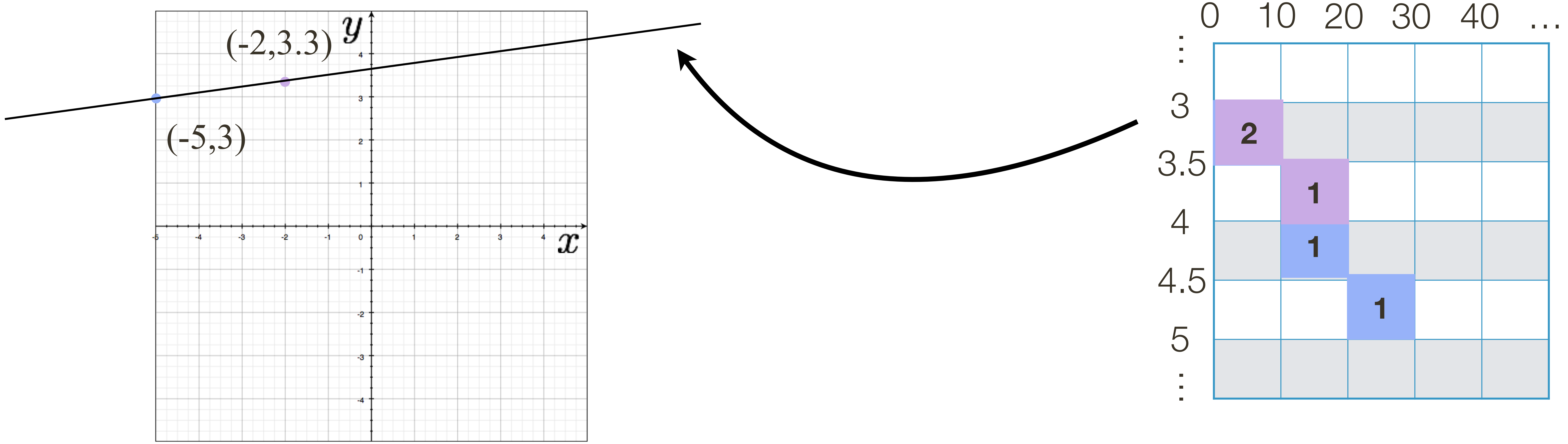




$$-5\sin(5^\circ) - 3\cos(5^\circ) + r = 0 => r = 3.42$$
$$-5\sin(15^\circ) - 3\cos(15^\circ) + r = 0 => r = 4.18$$
$$-5\sin(25^\circ) - 3\cos(25^\circ) + r = 0 => r = 4.83$$

$$-2\sin(5^\circ) - 3.3\cos(5^\circ) + r = 0 => r = 3.46$$
$$-2\sin(15^\circ) - 3.3\cos(15^\circ) + r = 0 => r = 3.71$$

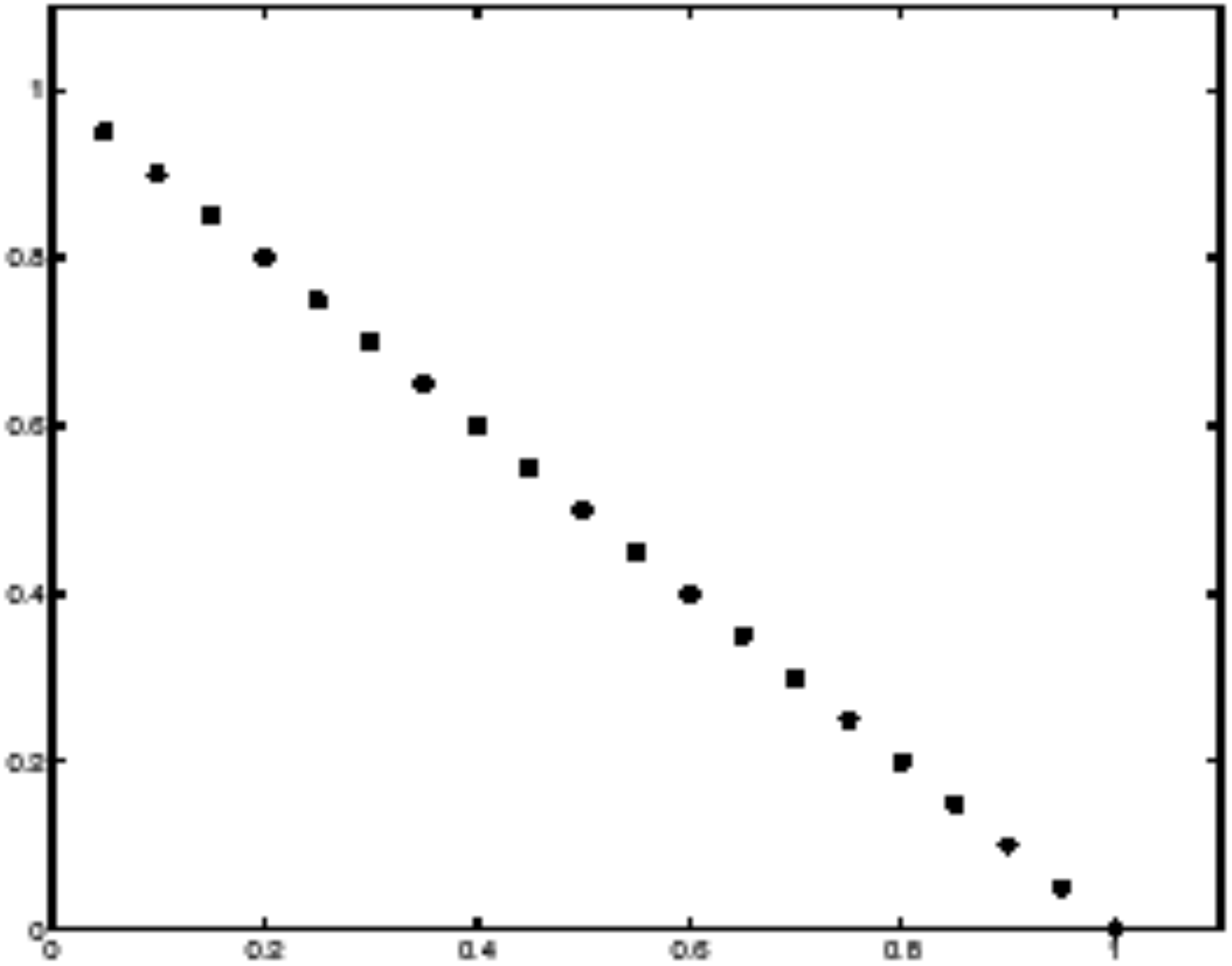
Example: Hough Transform for Lines



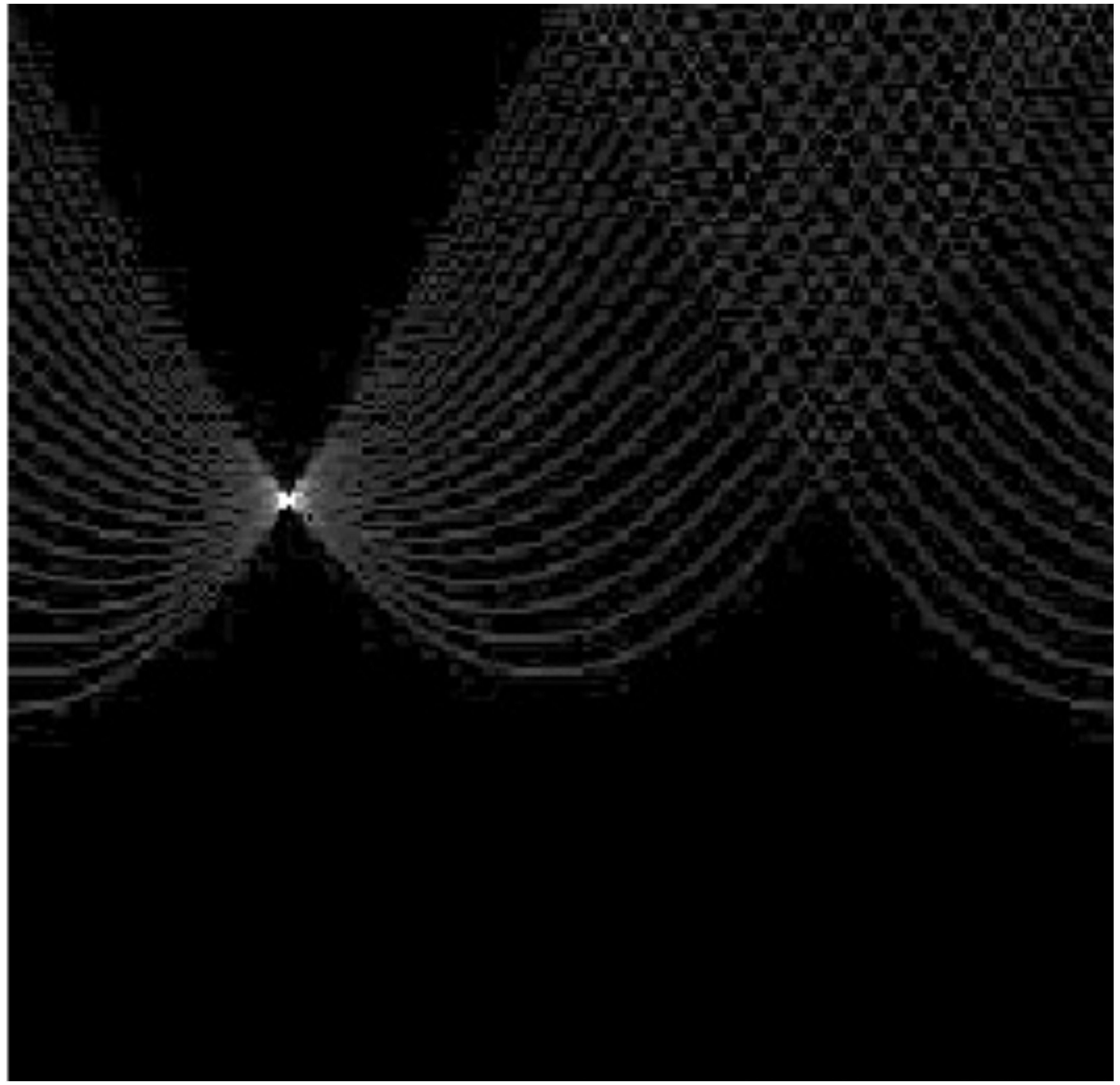
$$-5\sin(5^\circ) - 3\cos(5^\circ) + r = 0 => r = 3.42$$
$$-5\sin(15^\circ) - 3\cos(15^\circ) + r = 0 => r = 4.18$$
$$-5\sin(25^\circ) - 3\cos(25^\circ) + r = 0 => r = 4.83$$

$$-2\sin(5^\circ) - 3.3\cos(5^\circ) + r = 0 => r = 3.46$$
$$-2\sin(15^\circ) - 3.3\cos(15^\circ) + r = 0 => r = 3.71$$

Example: Clean Data



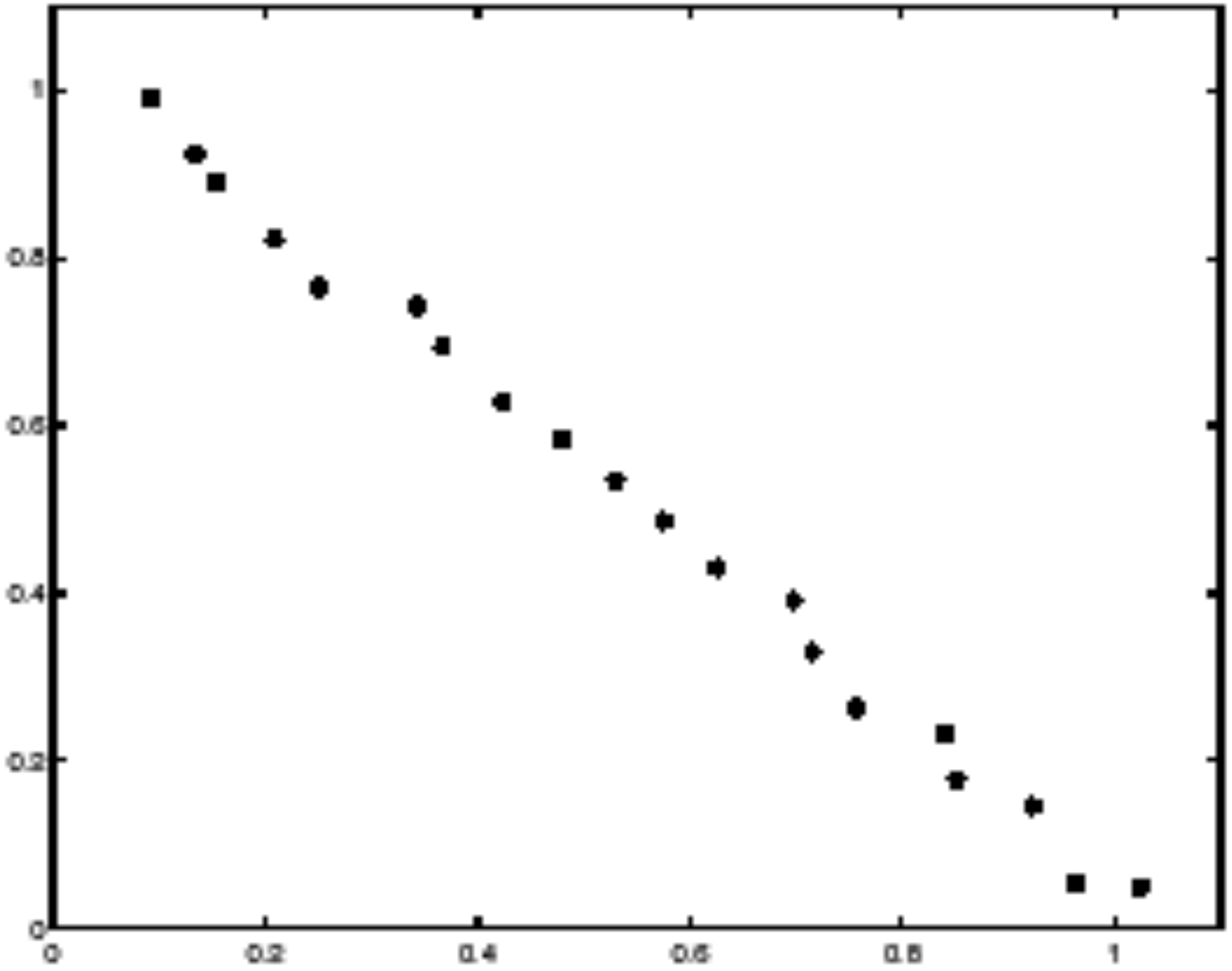
Tokens



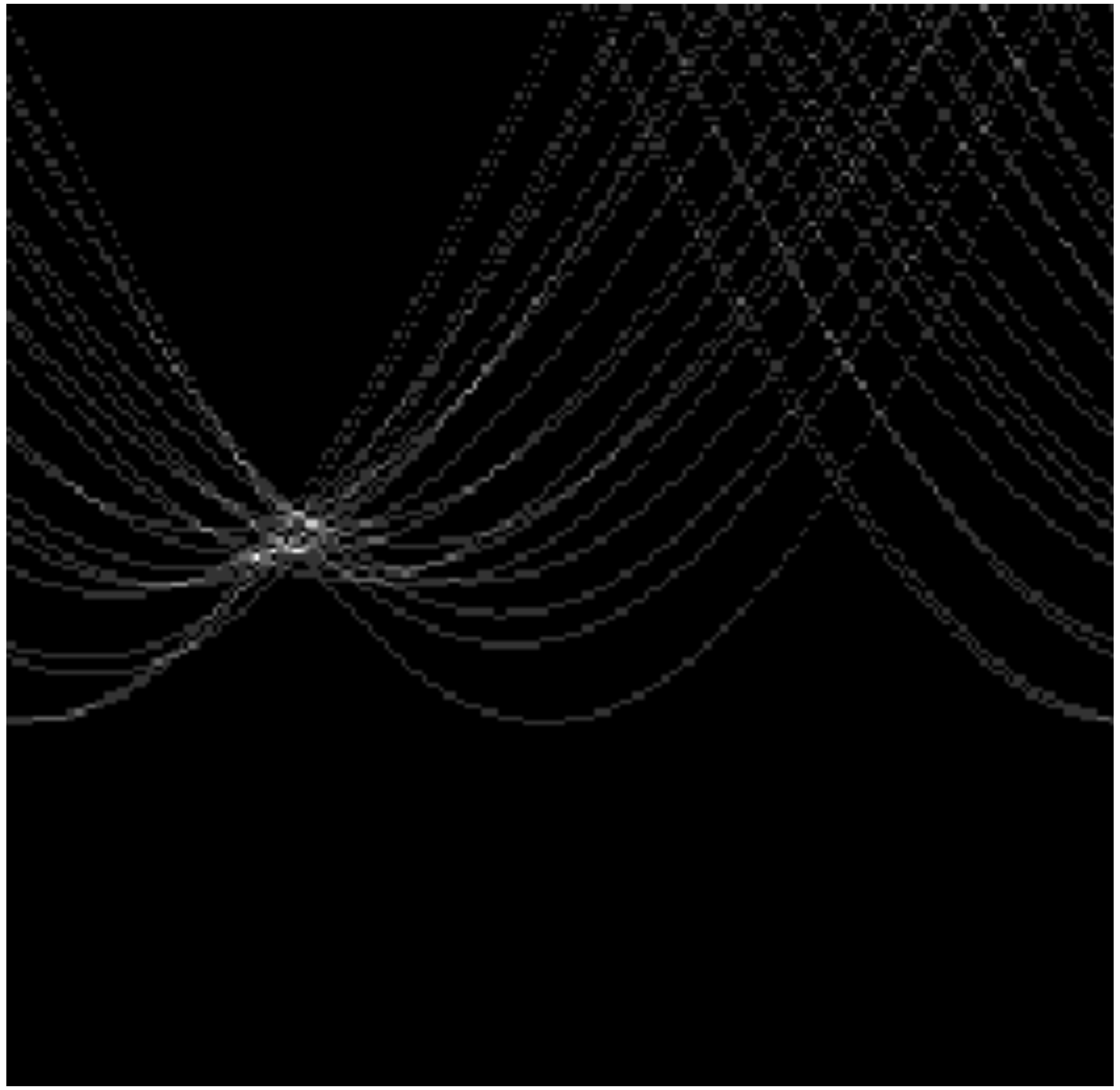
Votes
Horizontal axis is θ
Vertical Axis is r

Forsyth & Ponce (2nd ed.) Figure 10.1 (Top)

Example: Some Noise



Tokens



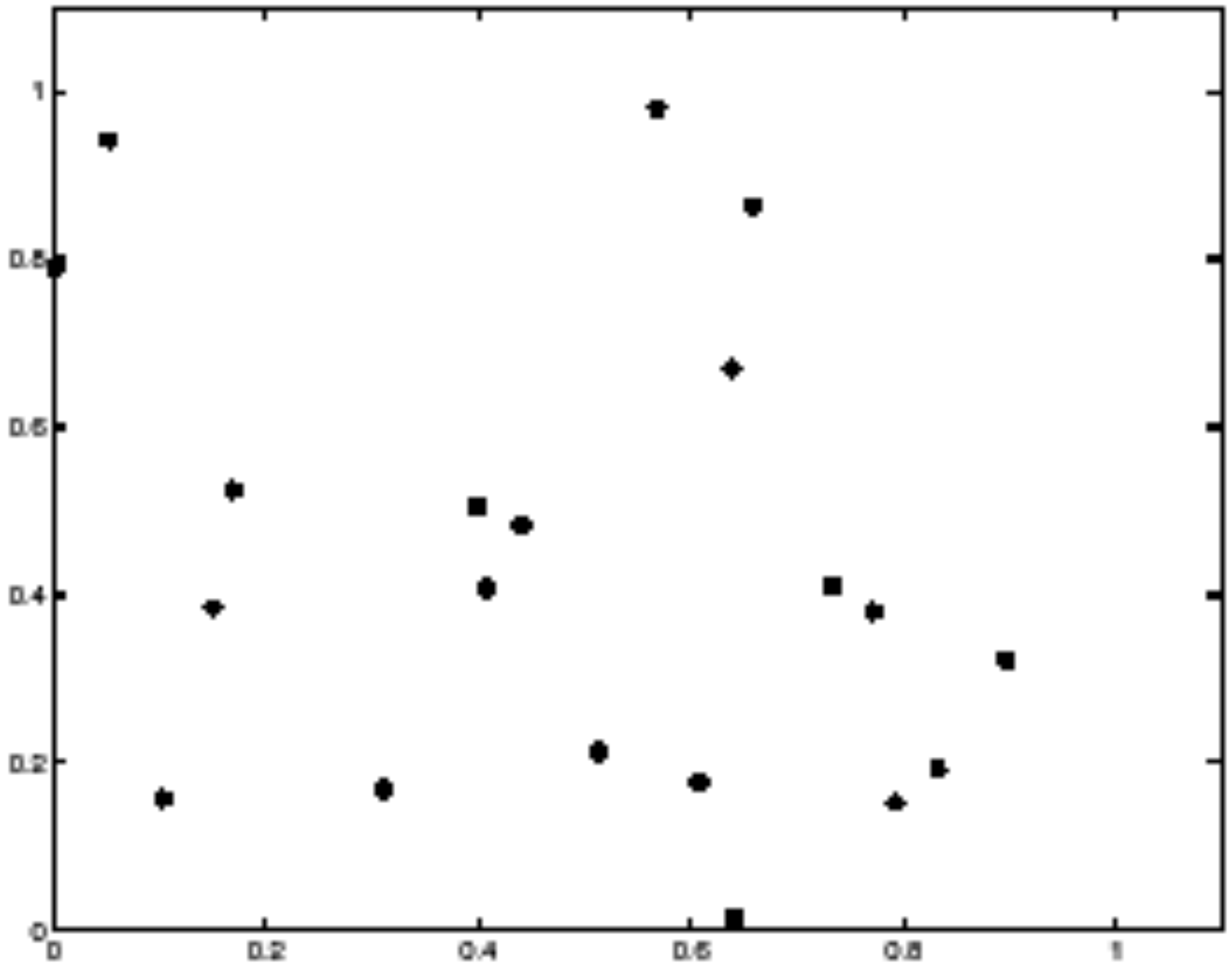
Votes

Horizontal axis is θ

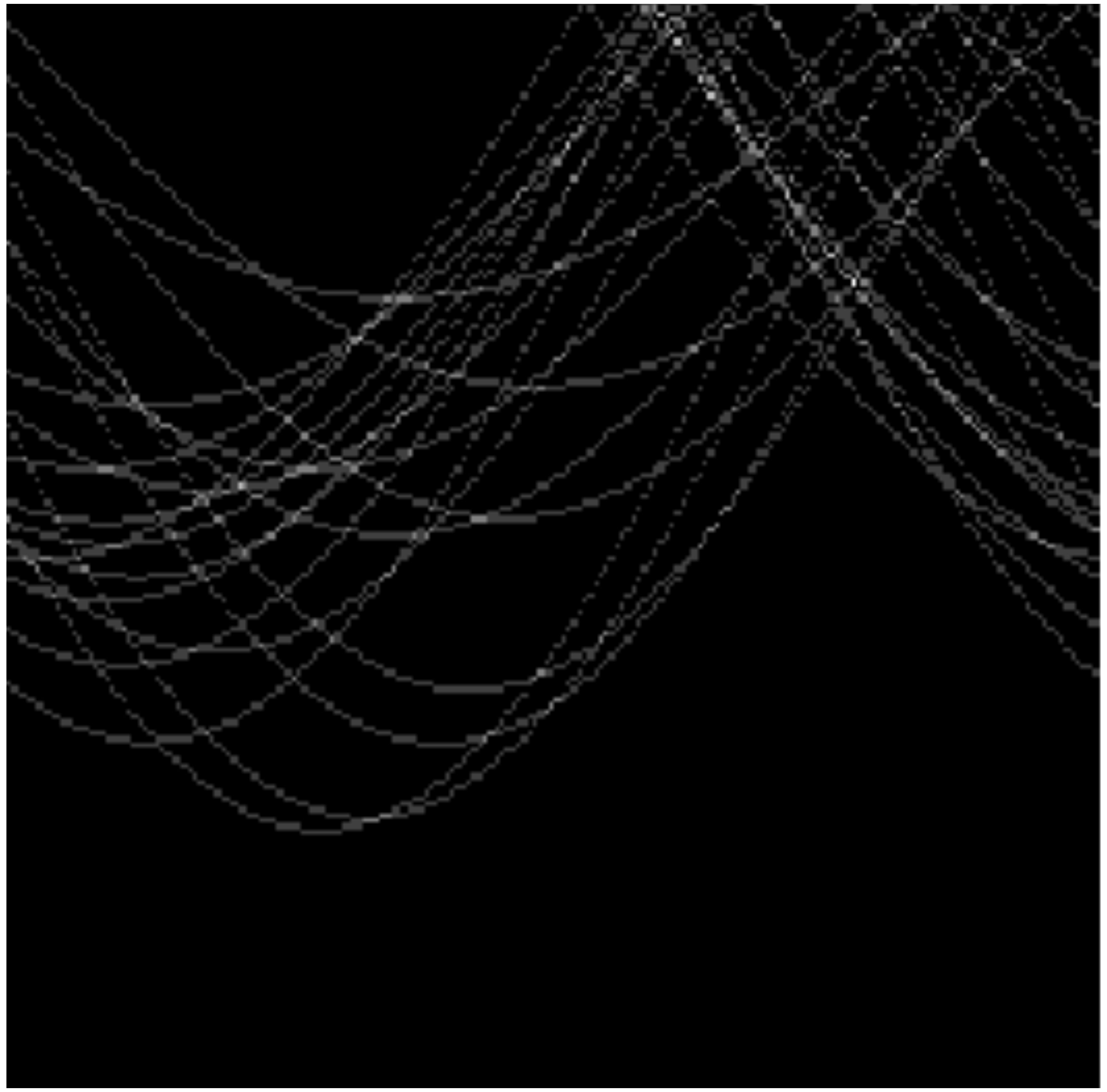
Vertical Axis is r

Forsyth & Ponce (2nd ed.) Figure 10.1 (Bottom)

Example: Too Much Noise



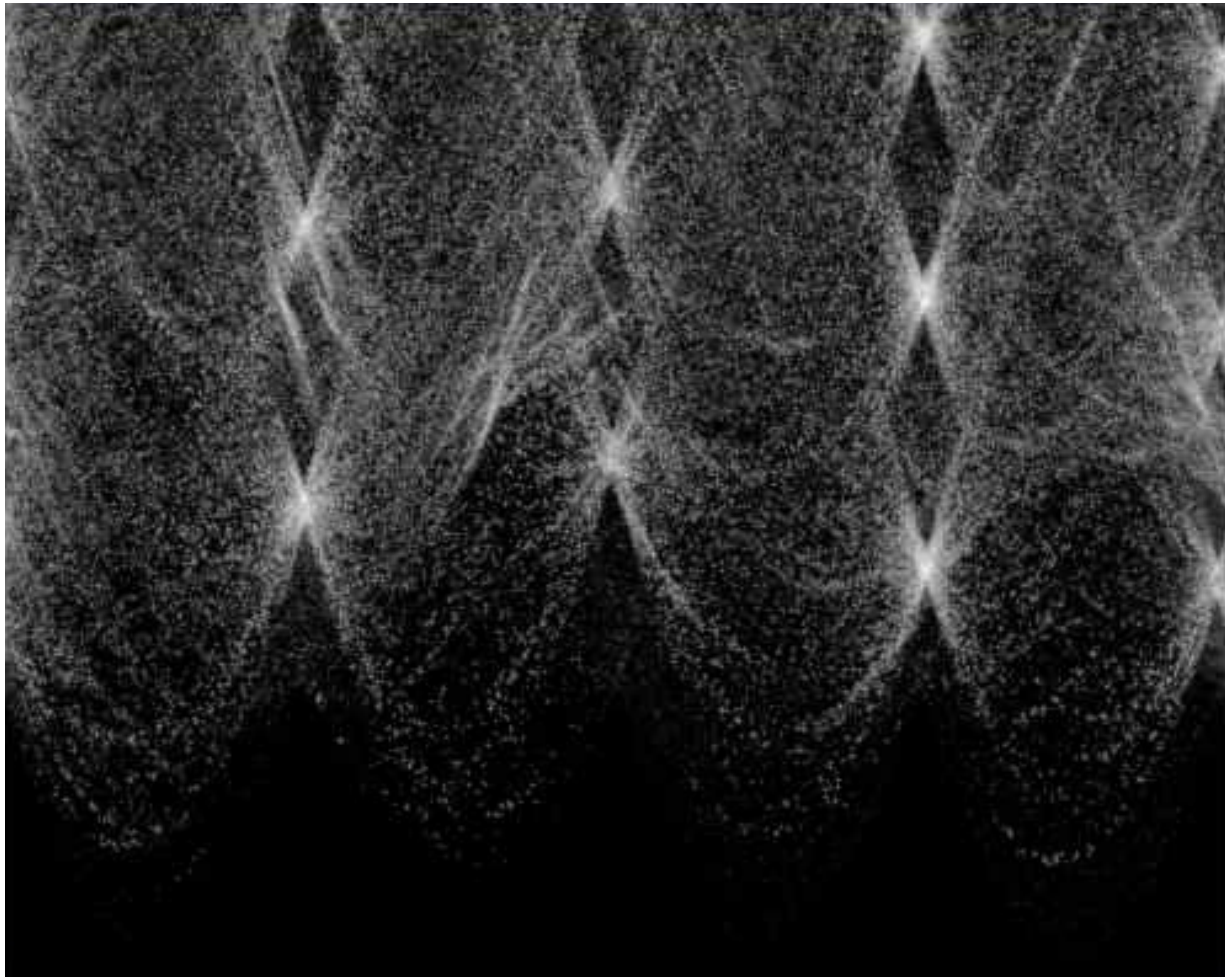
Tokens



Votes
Horizontal axis is θ
Vertical Axis is r

Forsyth & Ponce (2nd ed.) Figure 10.2

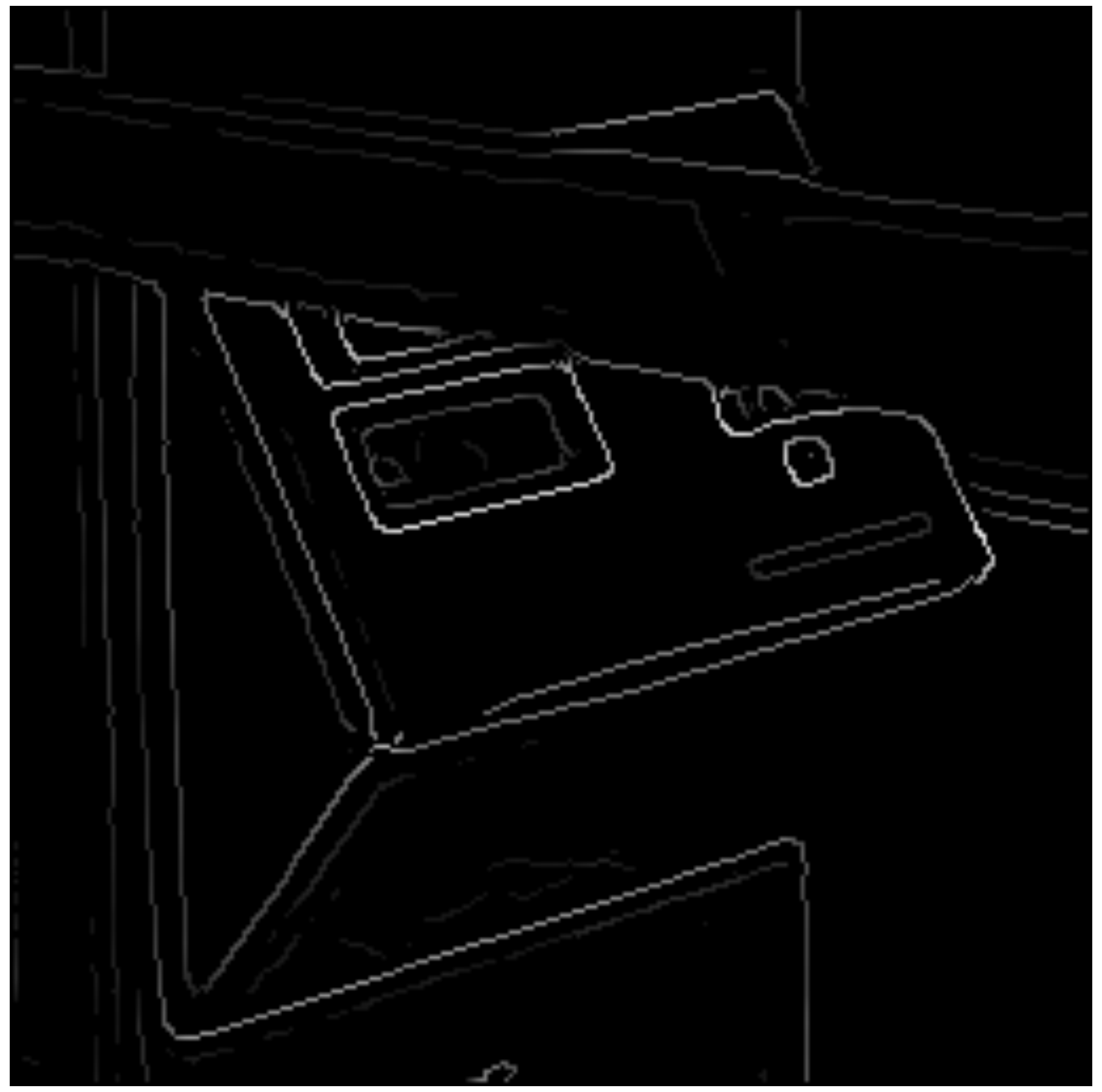
Real World **Example**



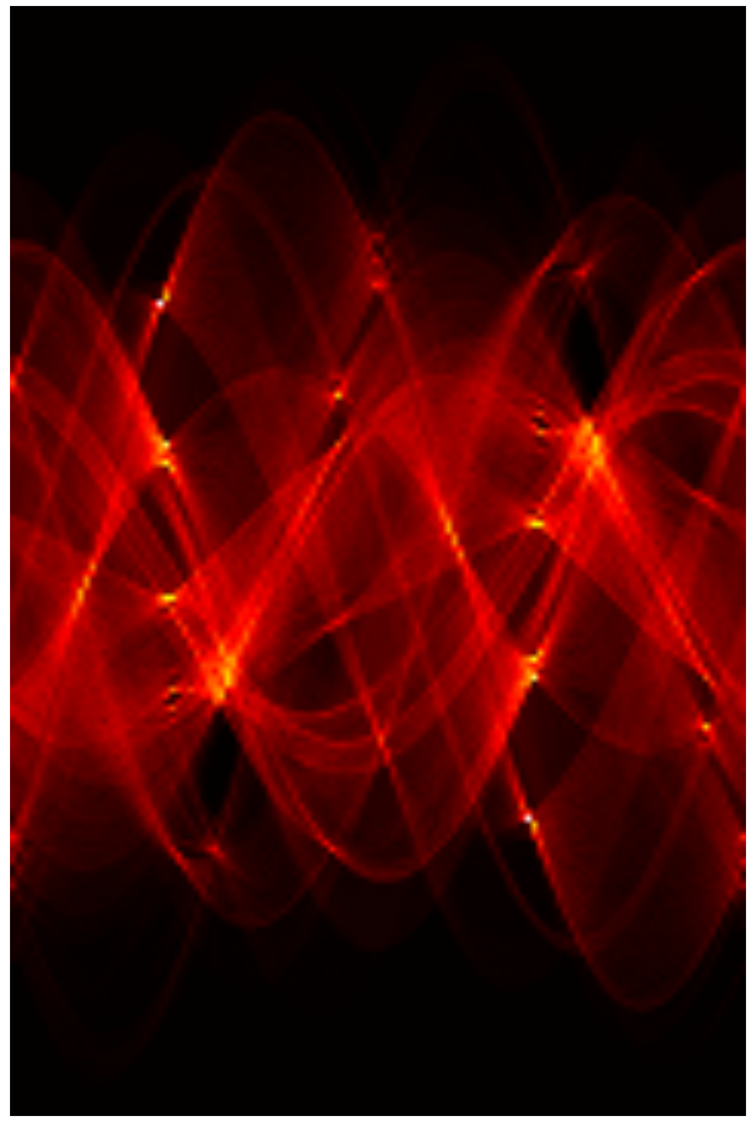
Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Real World **Example**

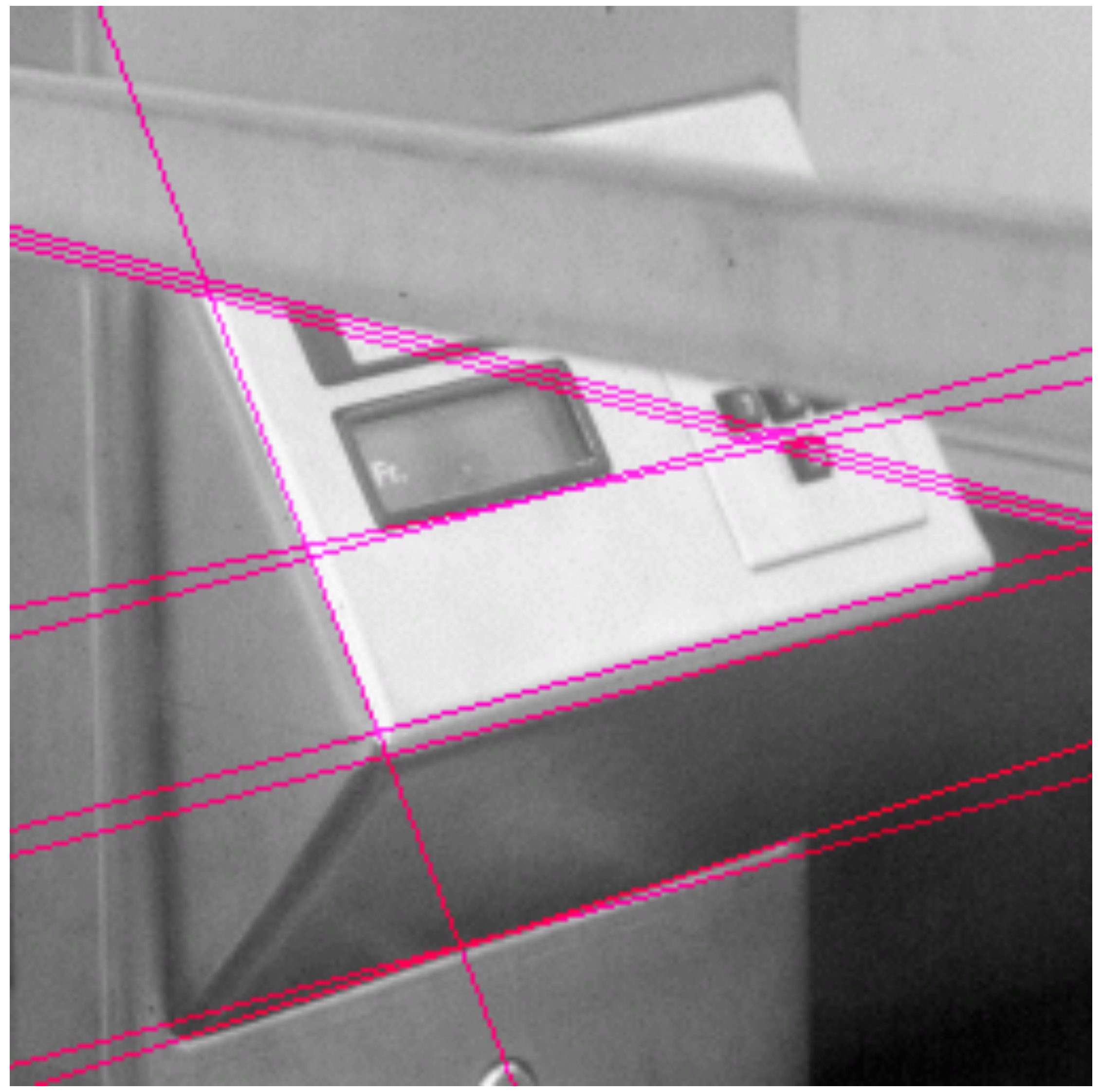
Original



Edges



Parameter space



Hough Lines

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Mechanics of Hough Transform

1. Construct a quantized array to represent θ and r

2. For each point, render curve (θ, r) into this array adding one vote at each cell

Difficulties:
— How big should the cells be? (too big, and we merge quite different lines; too small, and noise causes lines to be missed)

How many lines?

— Count the peaks in the Hough array

— Treat adjacent peaks as a single peak

Some **Practical Details** of Hough Transform

It is best to **vote** for the two closest bins in each dimension, as the locations of the bin boundaries are arbitrary

— This means that peaks are “blurred” and noise will not cause similar votes to fall into separate bins

Can use a **hash table** rather than an array to store the votes

— This means that no effort is wasted on initializing and checking empty bins

— It avoids the need to predict the maximum size of the array, which can be non-rectangular

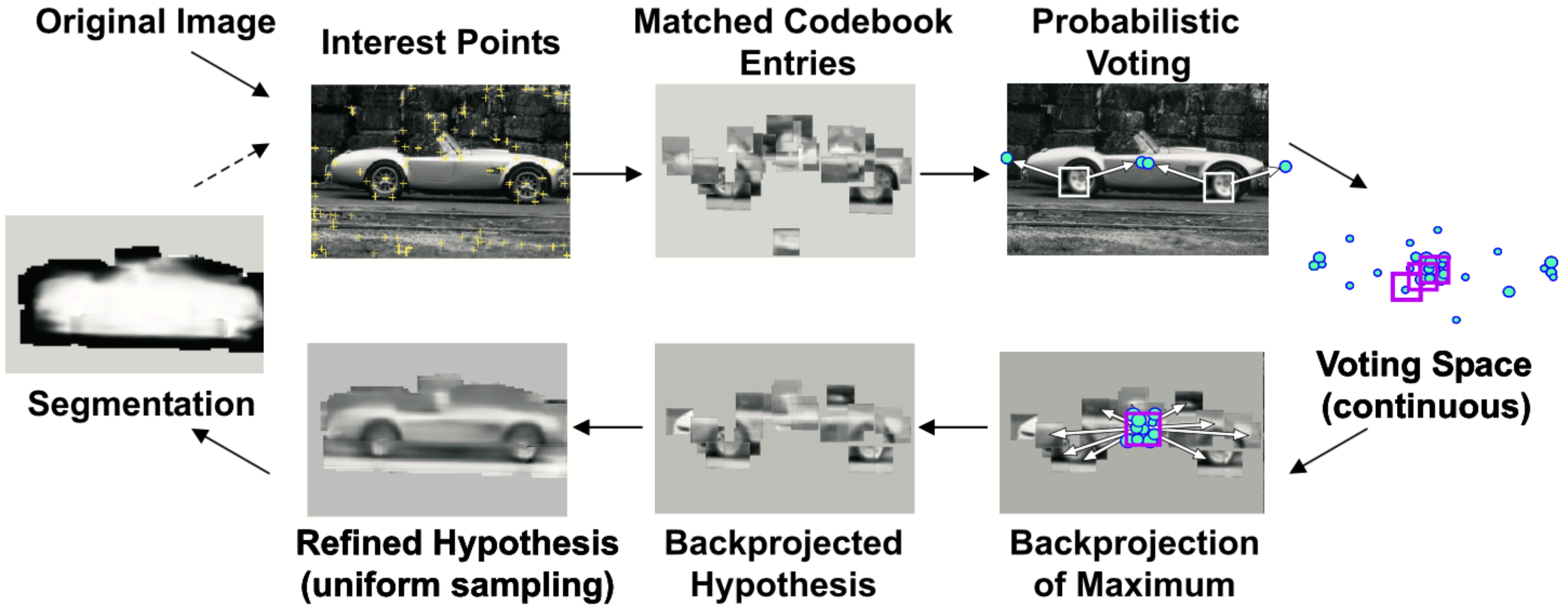
Some **Practical Details** of Hough Transform

A key is to have each feature (token) determine as many parameters as possible
— Lines are detected more effectively from edge elements with
both position and orientation
— For object recognition, each token should predict **position, orientation, and scale**

The Hough transform can extract feature groupings from clutter in linear time

Example 1: Object Recognition — Implicit Shape Model

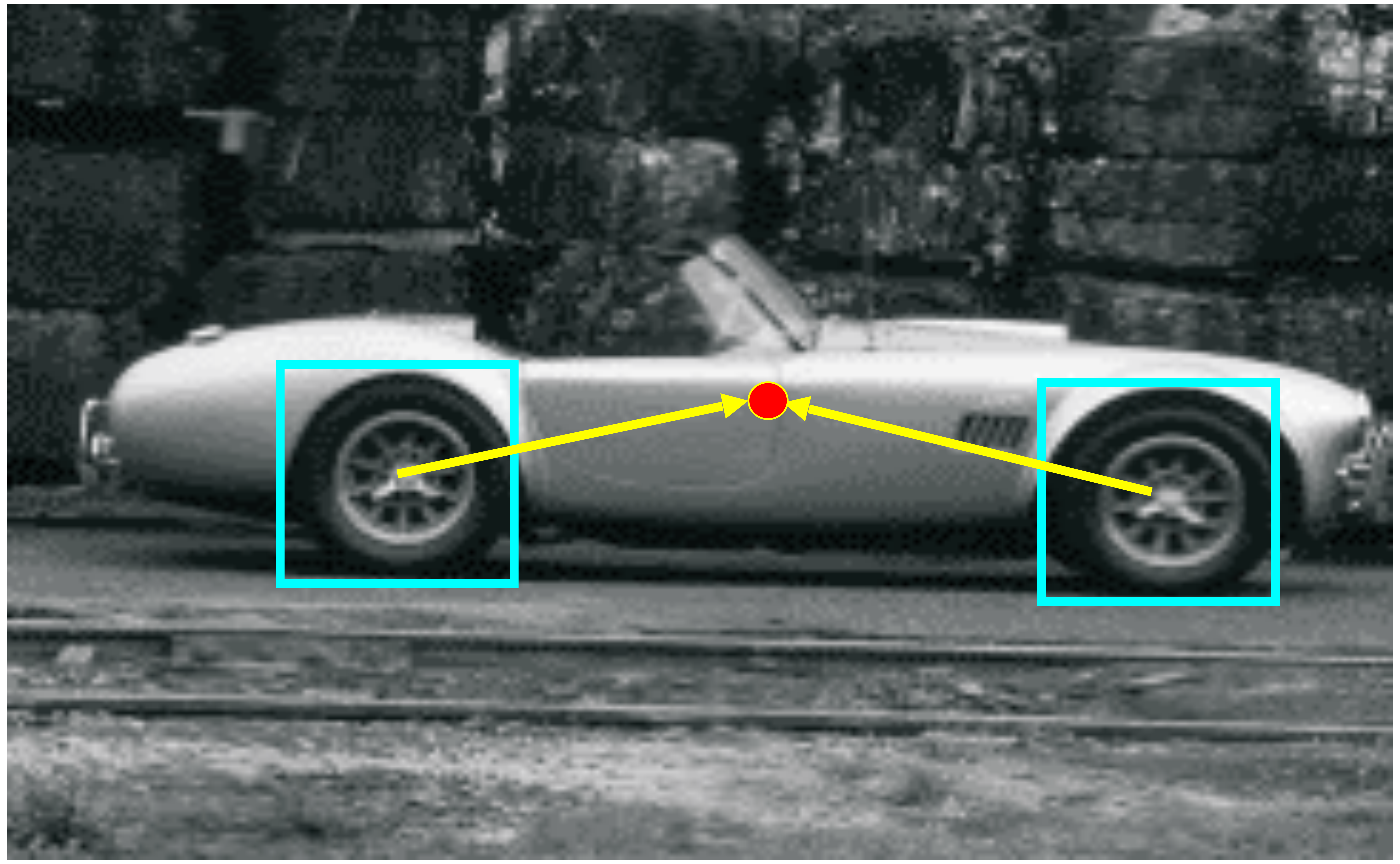
Combined object detection and segmentation using an implicit shape model.
Image patches cast weighted votes for the object centroid.



B. Leibe, A. Leonardis, and B. Schiele, Combined Object Categorization and Segmentation with an Implicit Shape Model, ECCV Workshop on Statistical Learning in Computer Vision 2004

Example 1: Object Recognition — Implicit Shape Model

Index displacements by “visual codeword”



training image

visual codeword with
displacement vectors

B. Leibe, A. Leonardis, and B. Schiele, Combined Object Categorization and Segmentation with an Implicit Shape Model, ECCV Workshop on Statistical Learning in Computer Vision 2004

Example 1: Object Recognition — Implicit Shape Model

B. Leibe, A. Leonardis, and B. Schiele, Combined Object Categorization and Segmentation with an Implicit Shape Model, ECCV Workshop on Statistical Learning in Computer Vision 2004

Example 2: Object Recognition — Boundary Fragments

Boundary fragments cast weighted votes for the object centroid. Also obtains an estimate of the object's contour.

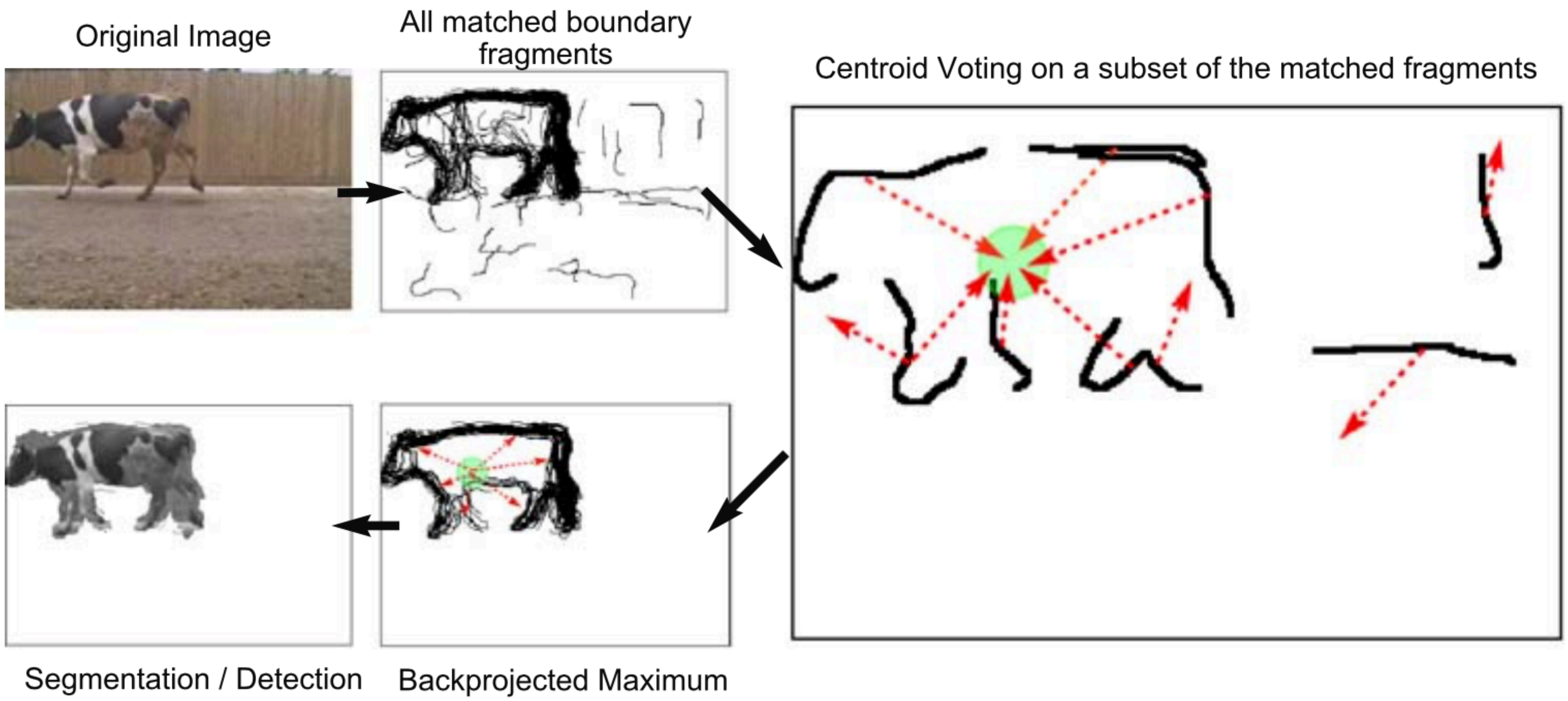


Image credit: Opelt et al., 2006

Example 2: Object Recognition — Boundary Fragments

Boundary fragments cast weighted votes for the object centroid. Also obtains an estimate of the object's contour.

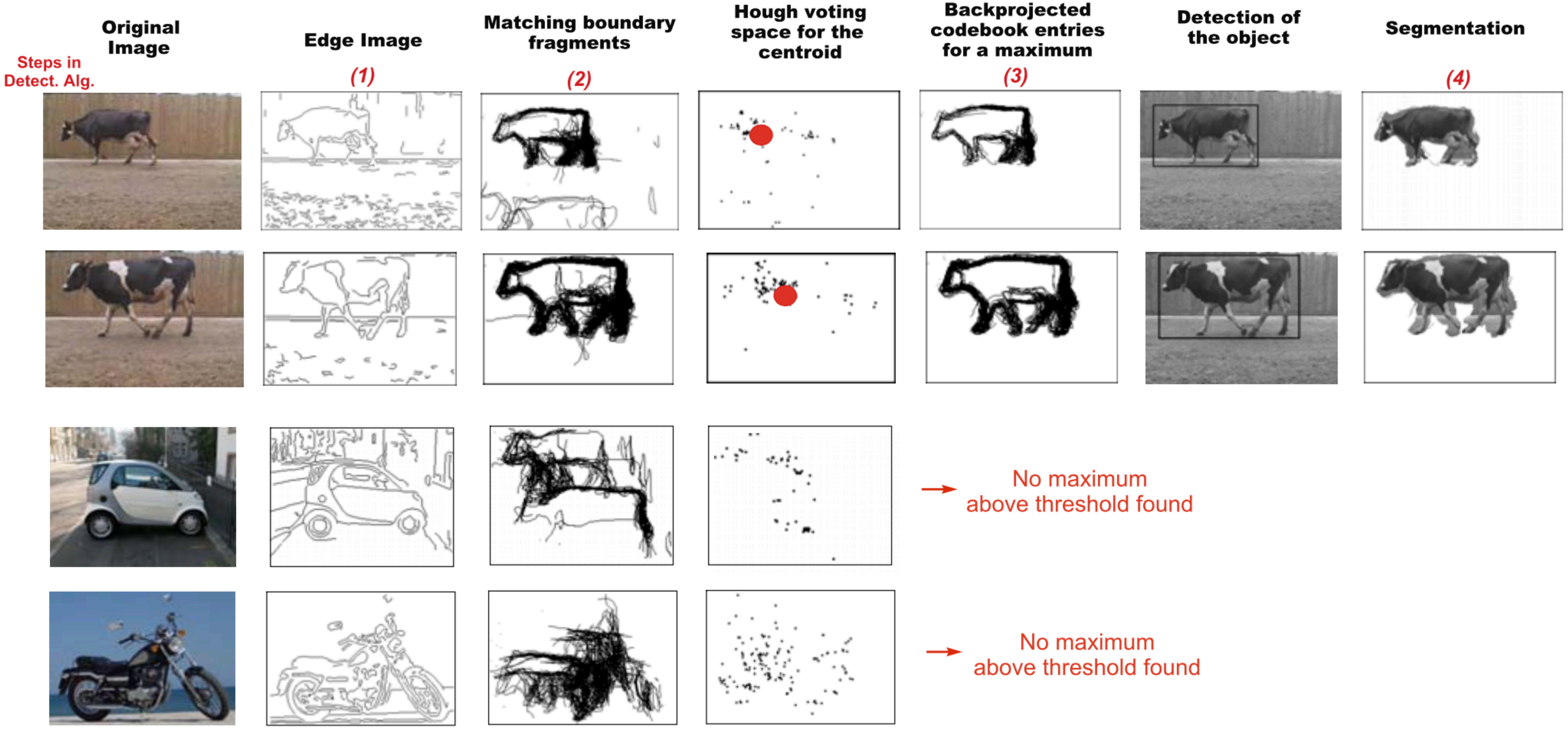


Image credit: Opelt et al., 2006

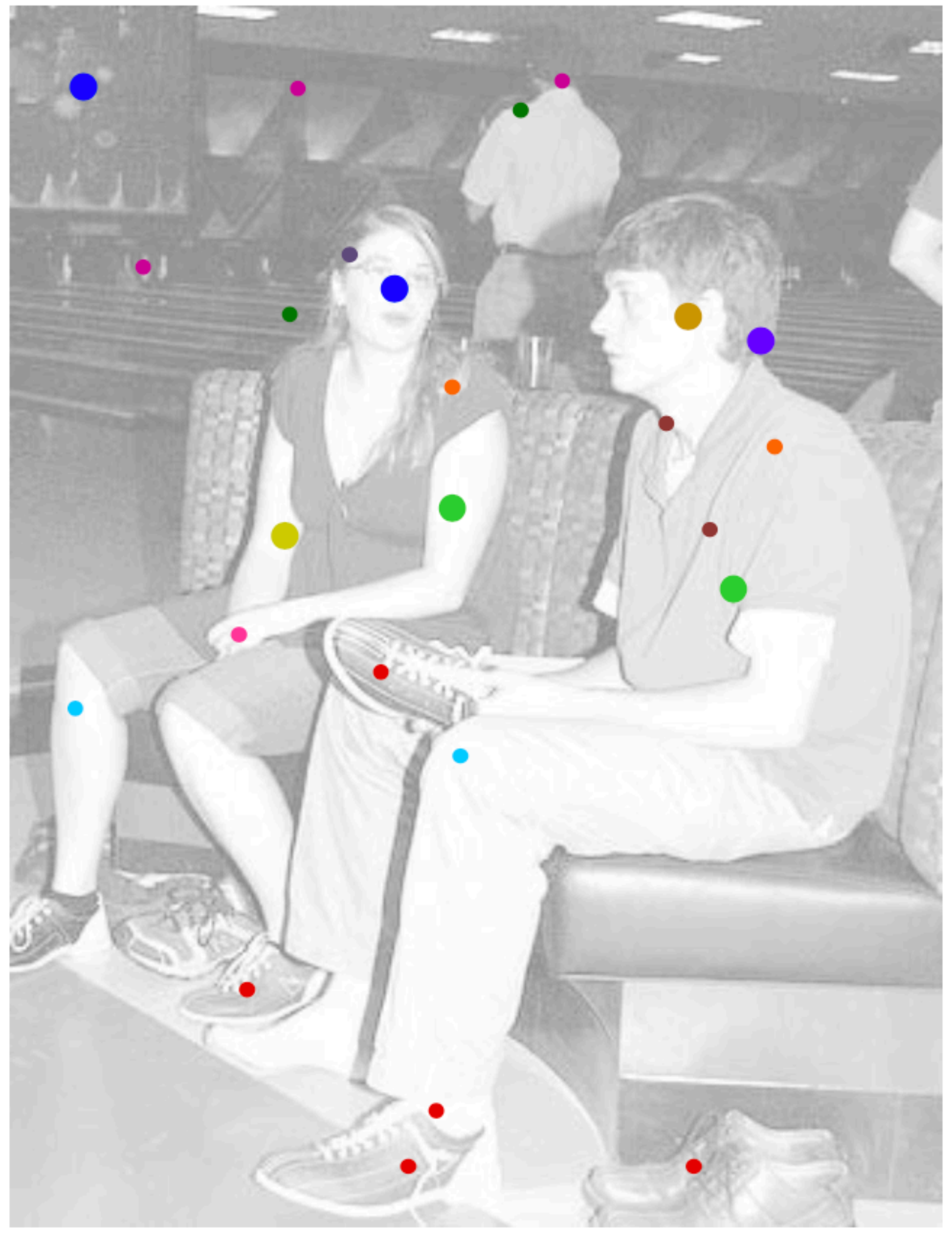
Example 3: Object Recognition — Poselets

Poselets are image patches that have distinctive appearance and can be used to infer some of the configuration of a parts-based object. Detected poselets vote for the object configuration.

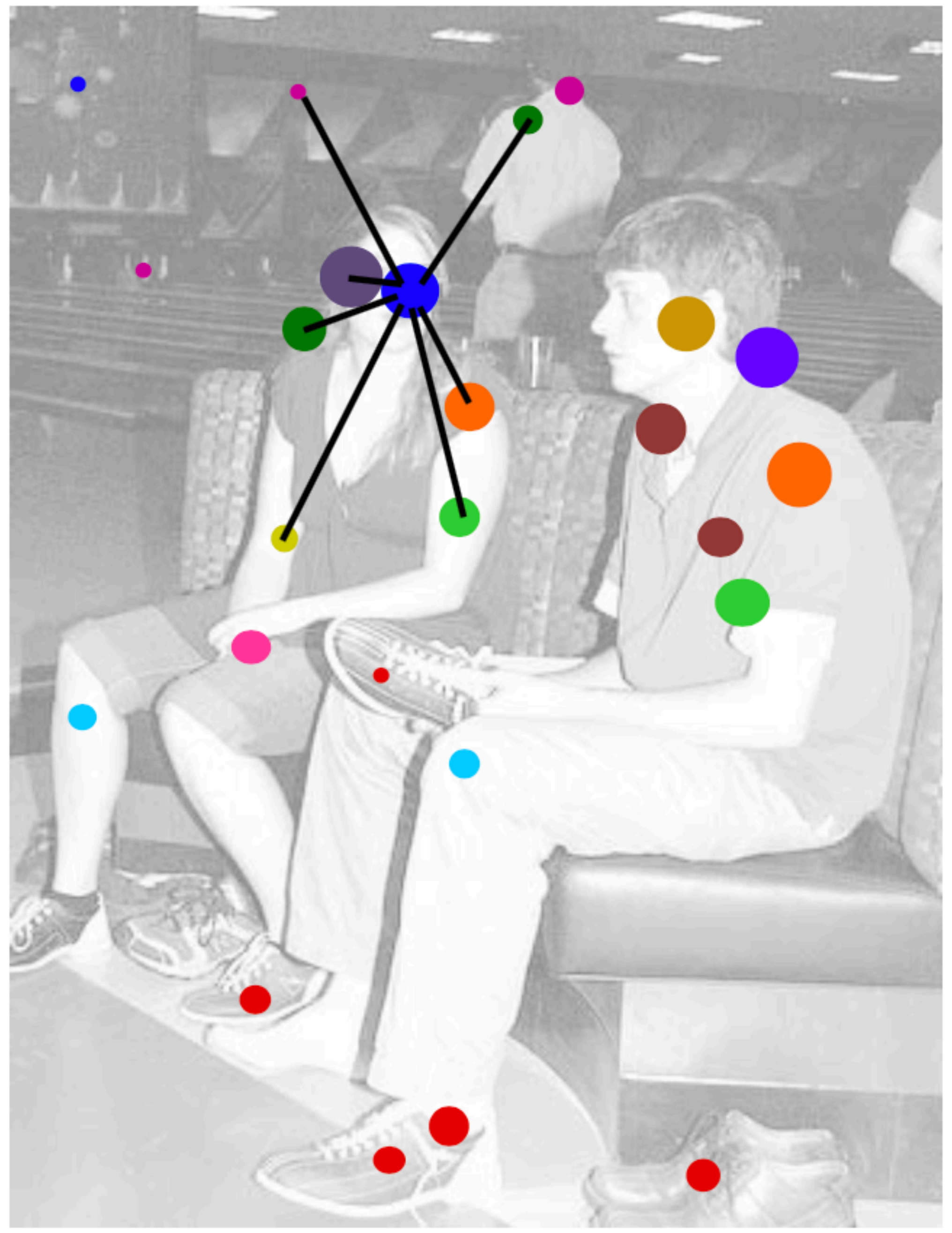
Image credit: Bourdev and Malik, 2009

Example 3: Object Recognition — Poselets

Poselets are image patches that have distinctive appearance and can be used to infer some of the configuration of a parts-based object. Detected poselets vote for the object configuration.

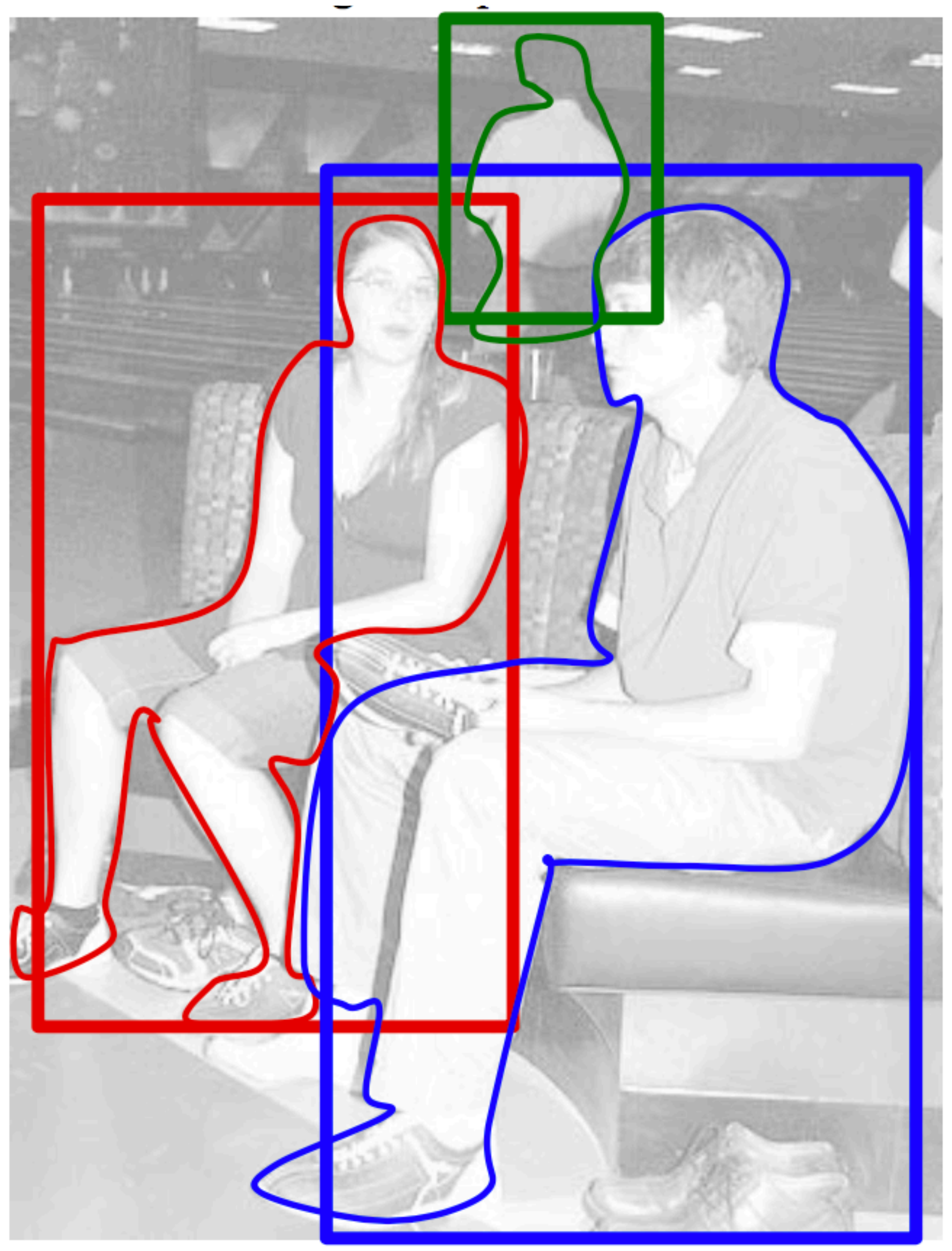


1. q-scores. Different colors illustrate different poselet detectors firing in the image. The blob size illustrates the score of the independent poselet classifier.



2. Q-scores (Section 4). Evidence from consistent poselet activations leads to a reranking based on mutual activation (Q-scores). Weaker activations consistent with others gain importance, whereas inconsistent ones get damped.

3. Clustering (Section 5). Activations are merged in a greedy manner starting with the strongest activation. Merging is based on pairwise consistency.



4. Bounding boxes (Section 6) and segmentations (Section 7). We predict the visible bounds and the contour of the person using the poselets within the cluster.

Image credit: Bourdev and Malik, 2009

Discussion of Hough Transform

Advantages:

— Can handle high percentage of outliers: each point votes separately

— Can detect multiple instances of a model in a single pass

Disadvantages:

— Complexity of search time increases exponentially with the number of model parameters

— Can be tricky to pick a good bin size

Summary of Hough Transform

The **Hough transform** is another technique for fitting data to a model

— a voting procedure

— possible model parameters define a quantized accumulator array

— data points “vote" for compatible entries in the accumulator array

A key is to have each data point (token) constrain model parameters as tightly as possible