The Hough Transform

CS 482, Prof. Stein

Lecture 6E

Reading & Slide Credits

Readings:

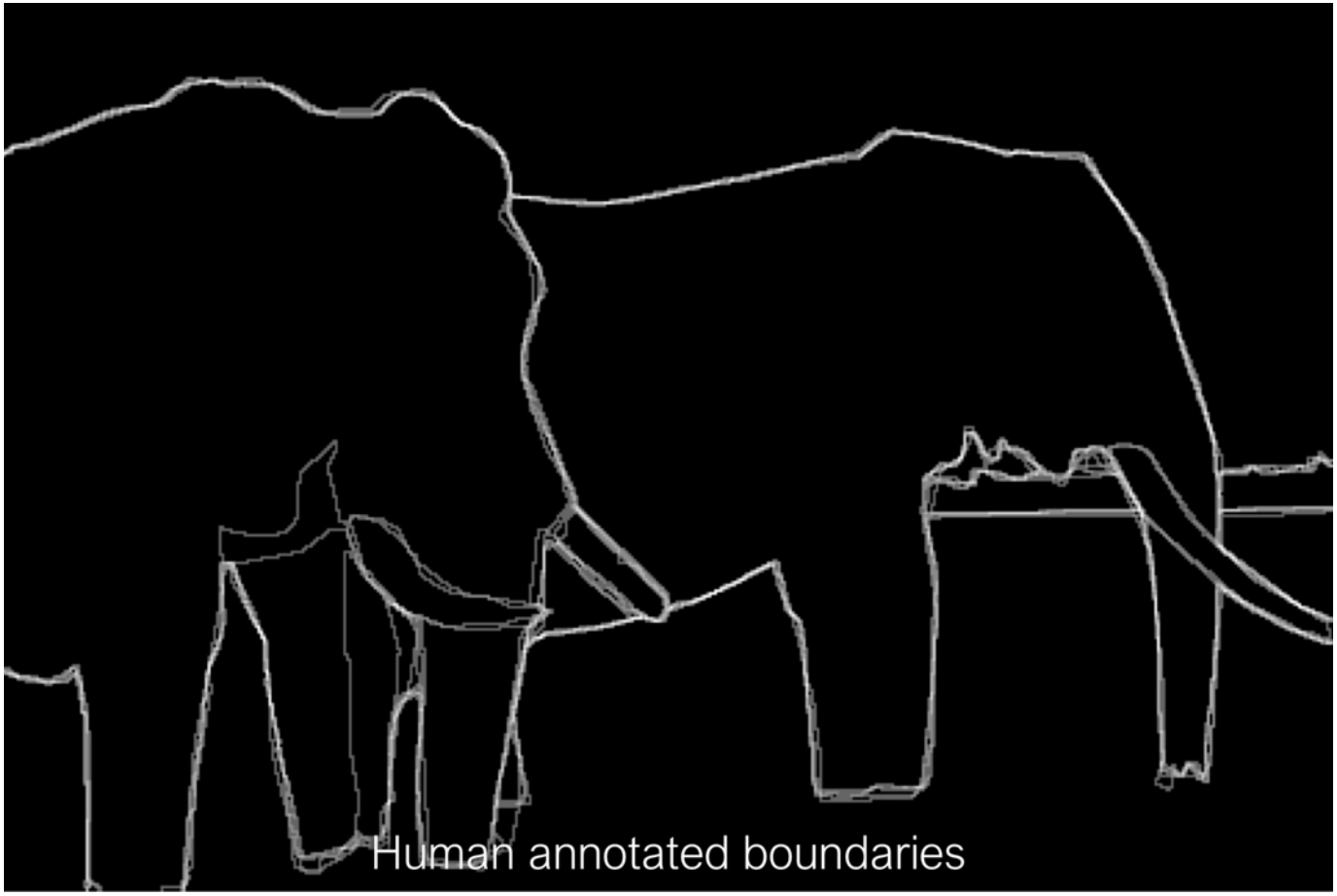
- Szeliski: 4.2, 4.3

Slide Credits (from which many of these slides are either directly taken or adapted)

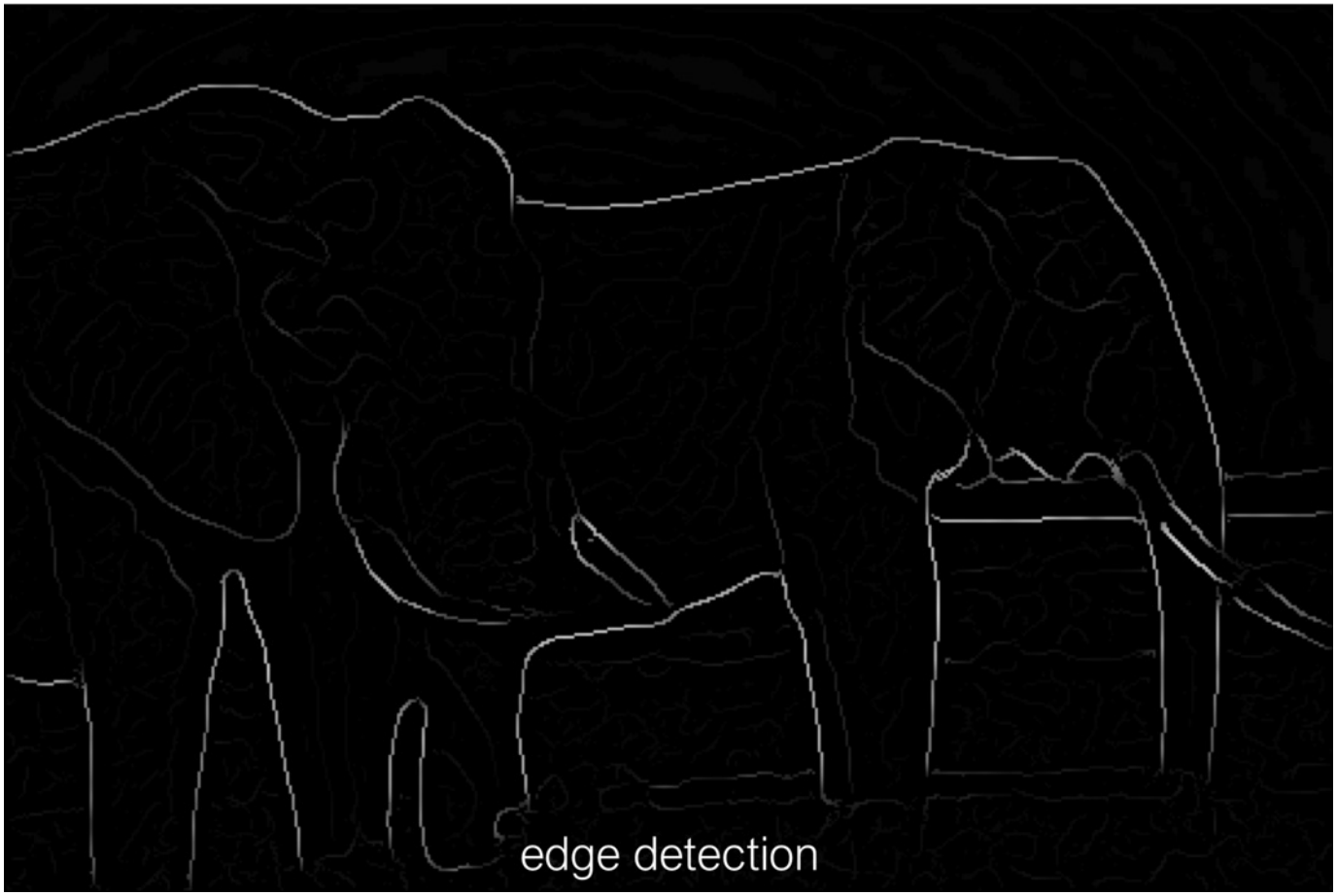
- CMU Computer Vision Course (Yannis Gkioulekas)

Detecting Object Boundaries

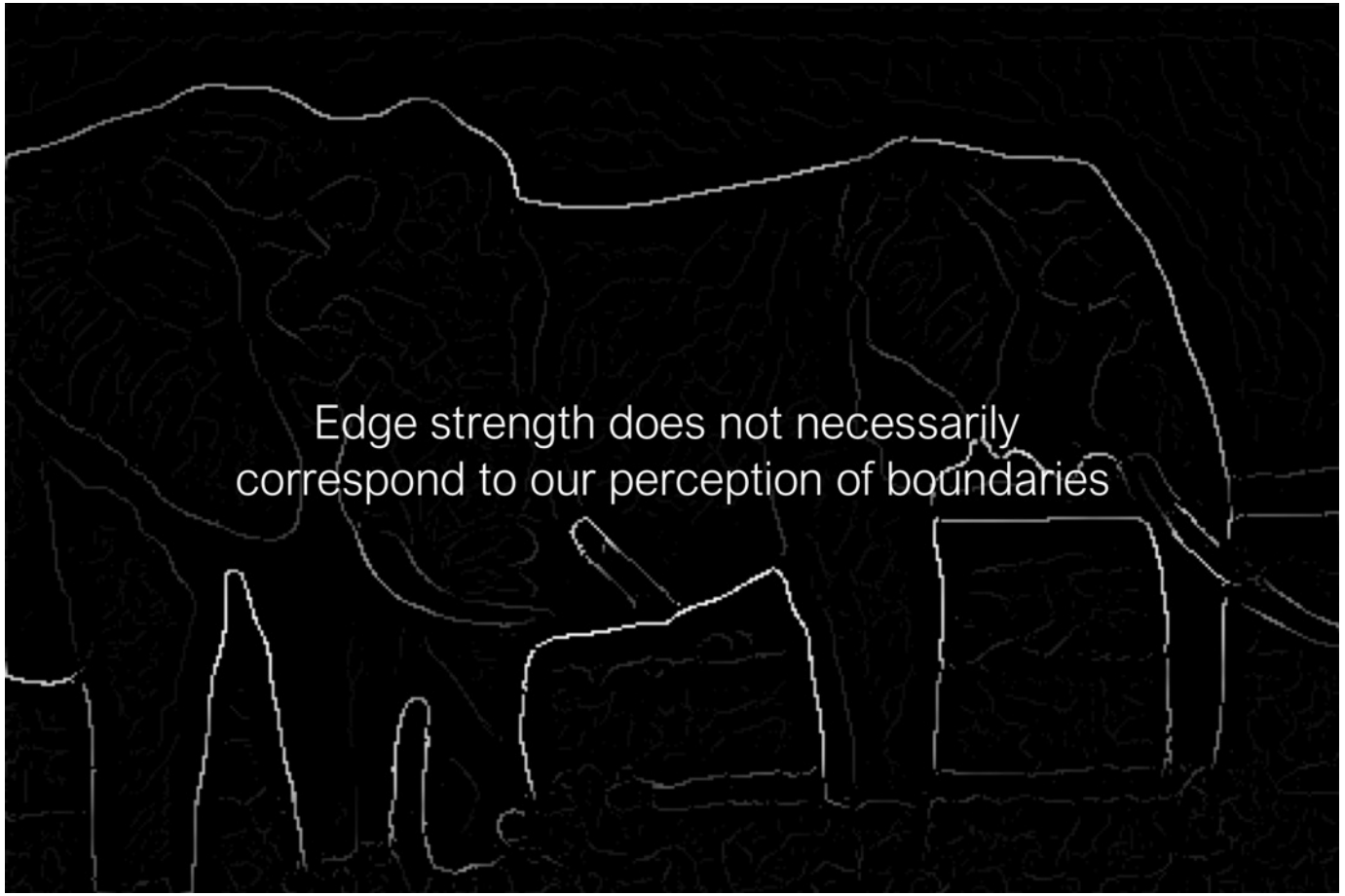
Detecting Object Boundaries



Detecting Object Boundaries

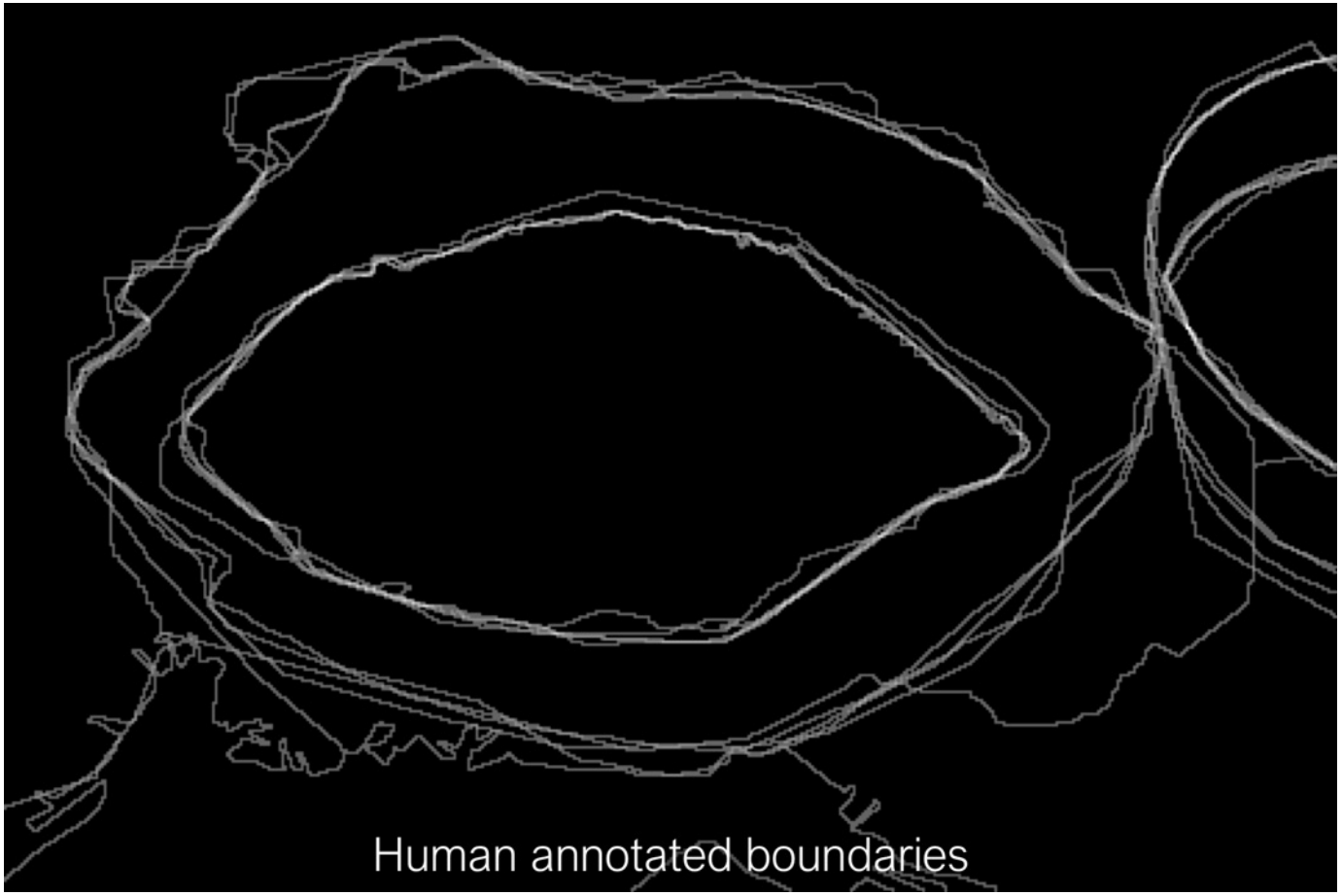


Detecting Object Boundaries



Detecting Object Boundaries

Detecting Object Boundaries

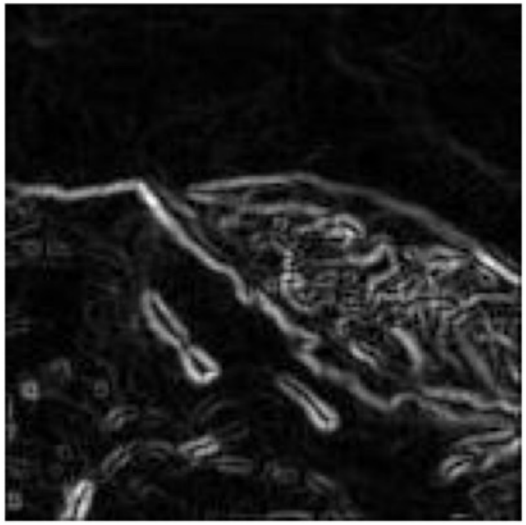


Detecting Object Boundaries

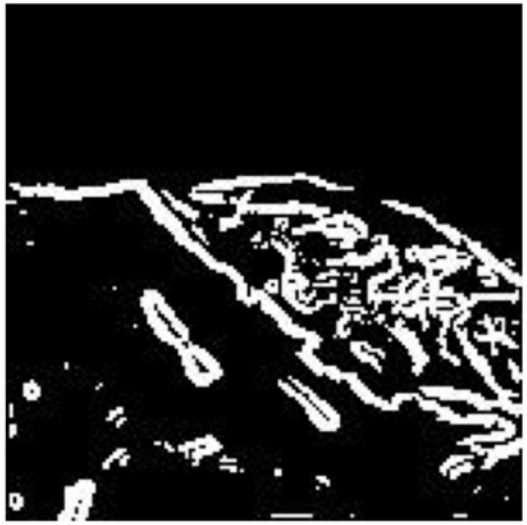
Detecting Object Boundaries

Lines can be difficult to find reliably

Original image



Edge detection



Thresholding

Noisy edge image
Incomplete boundaries

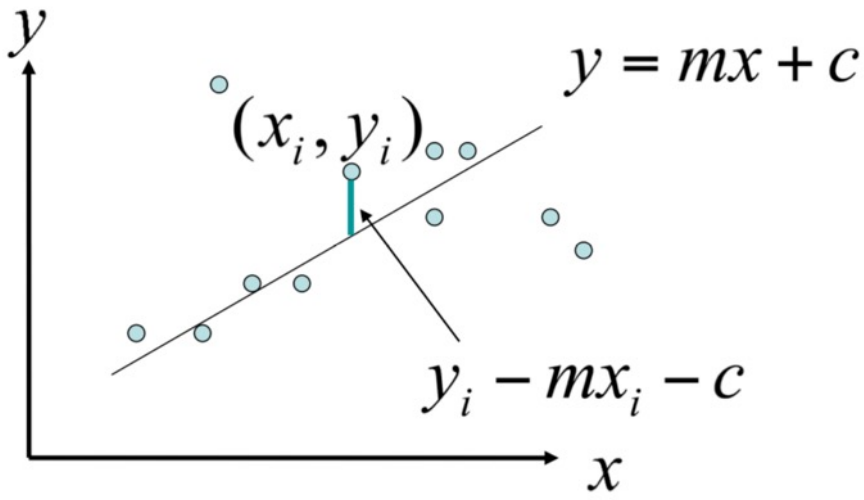
Let's say we want to fit a line to some data

Given: Many (x_i, y_i) pairs

Find: Parameters (m, c)

Minimize: Average square distance:

$$E = \sum_i \frac{(y_i - mx_i - c)^2}{N}$$



Let's say we want to fit a line to some data

Given: Many (x_i, y_i) pairs

Find: Parameters (m, c)

Minimize: Average square distance:

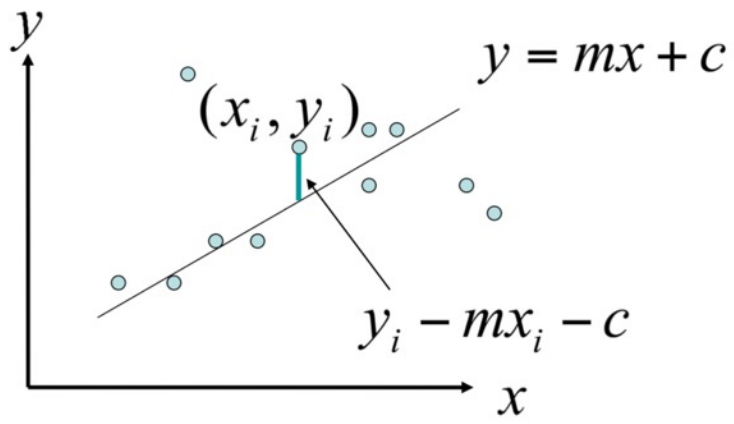
$$E = \sum_i \frac{(y_i - mx_i - c)^2}{N}$$

Using:

$$\frac{\partial E}{\partial m} = 0 \quad \& \quad \frac{\partial E}{\partial c} = 0$$

Note:

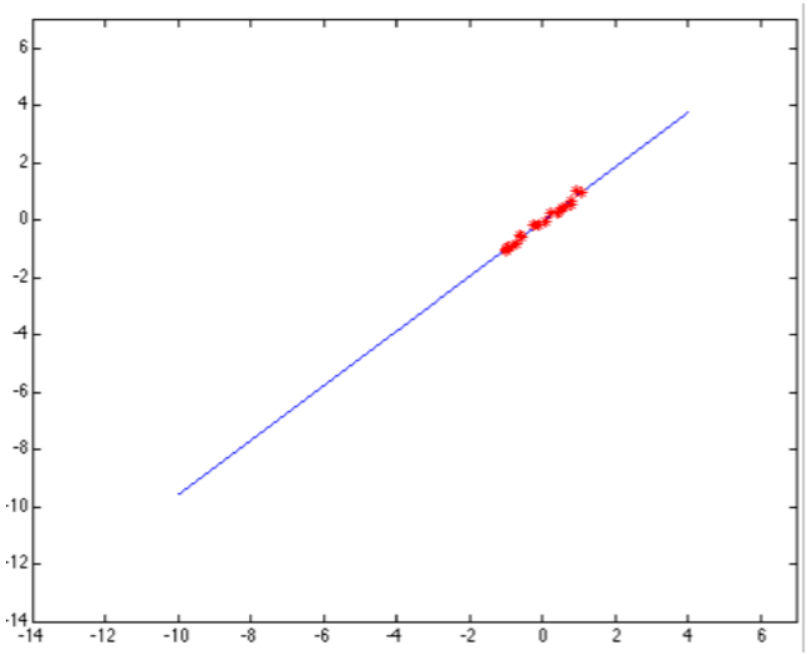
$$\bar{y} = \frac{\sum_i y_i}{N} \qquad \bar{x} = \frac{\sum_i x_i}{N}$$



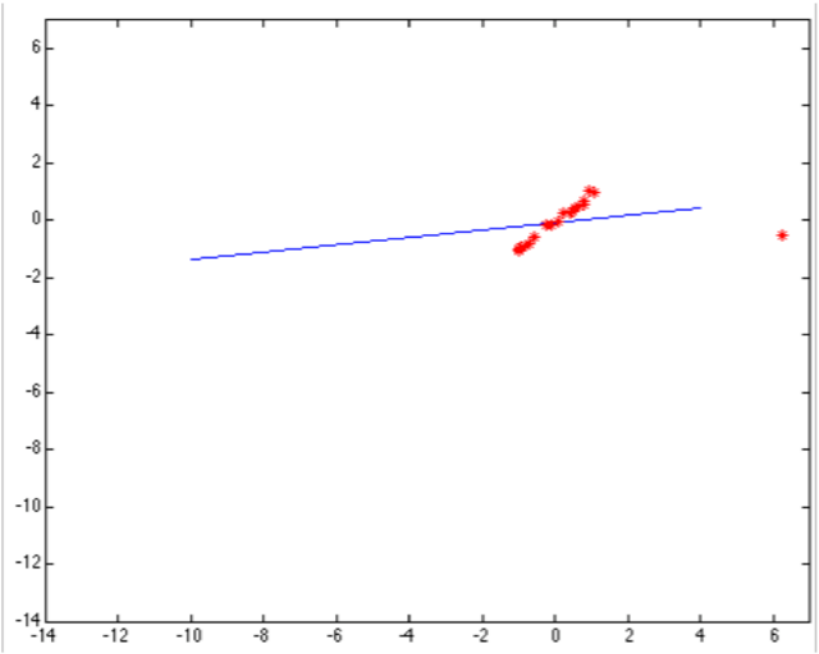
$$c = \bar{y} - m\bar{x}$$

$$m = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sum_i (x_i - \bar{x})^2}$$

Let's say we want to fit a line to some data

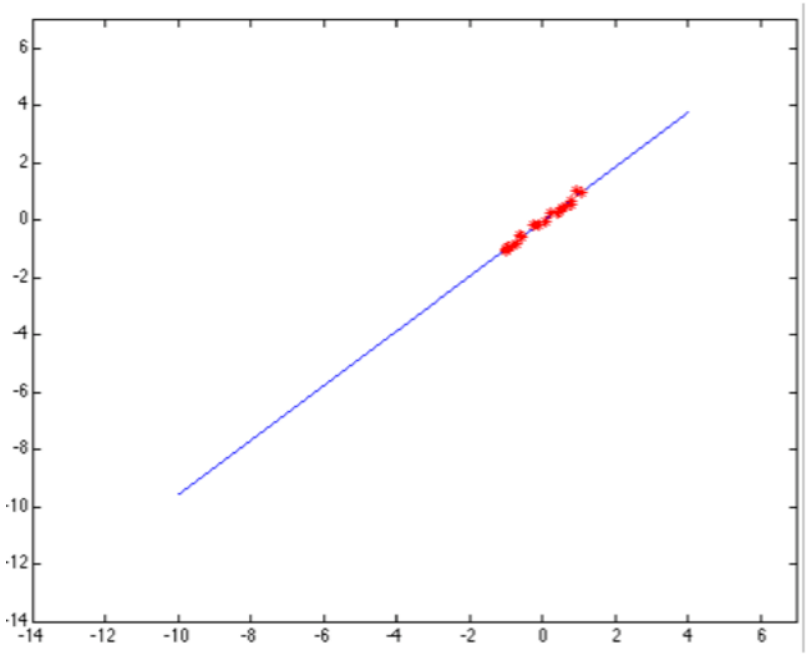


Least-squares error fit

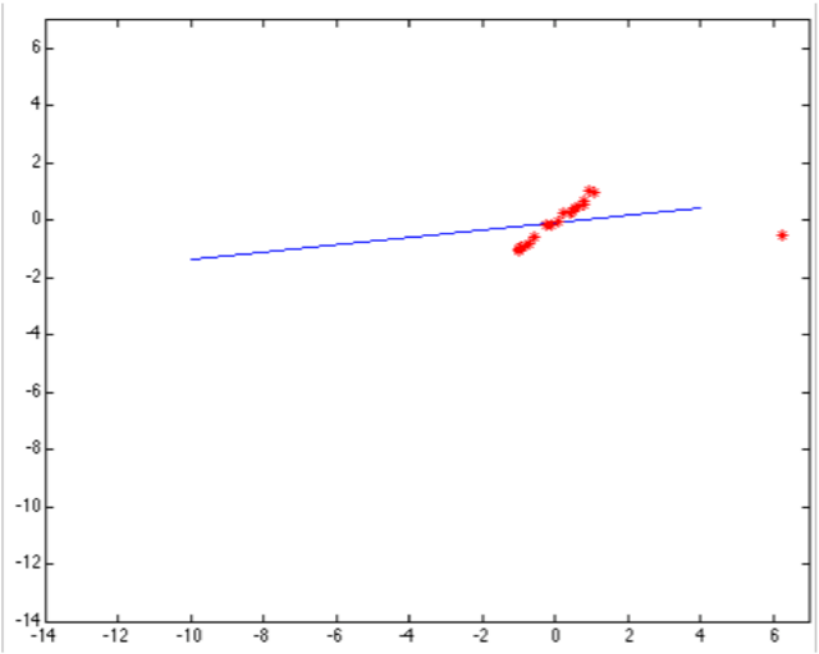


Squared error heavily penalizes outliers

Let's say we want to fit a line to some data



Least-squares error fit



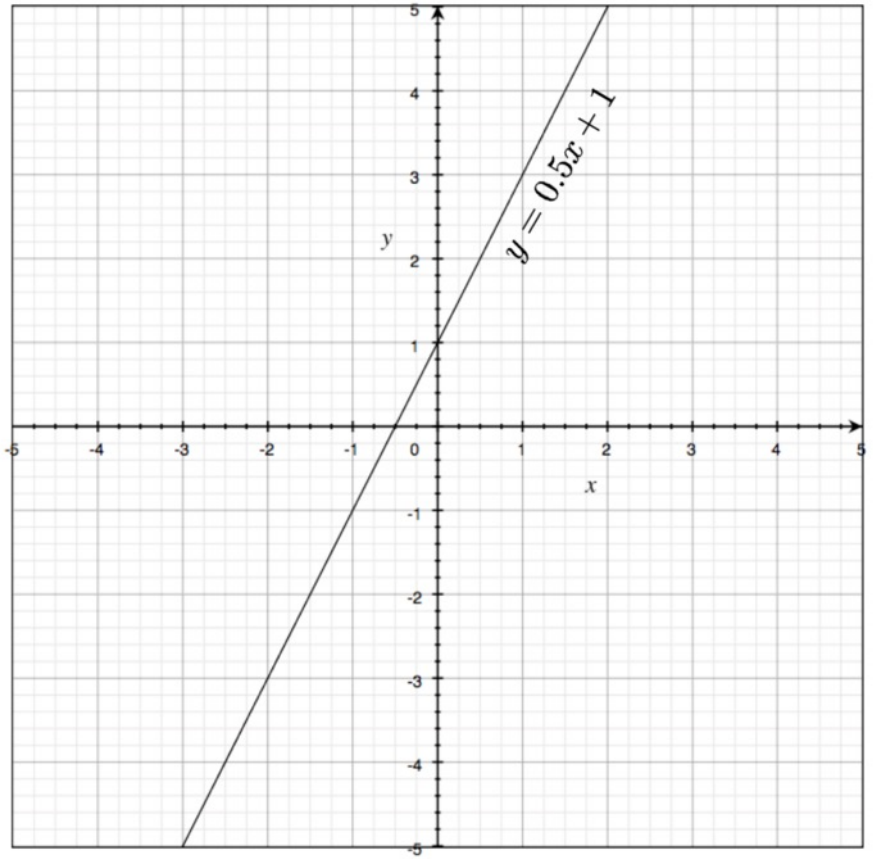
Squared error heavily penalizes outliers

Maybe we need to think about this problem differently…

Line Models
Slope-Intercept Form

$$y = mx + b$$

slope y-intercept

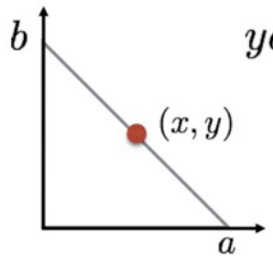


Line Models
Double-Intercept Form

$$\frac{x}{a} + \frac{y}{b} = 1$$

x-intercept y-intercept

Derivation:

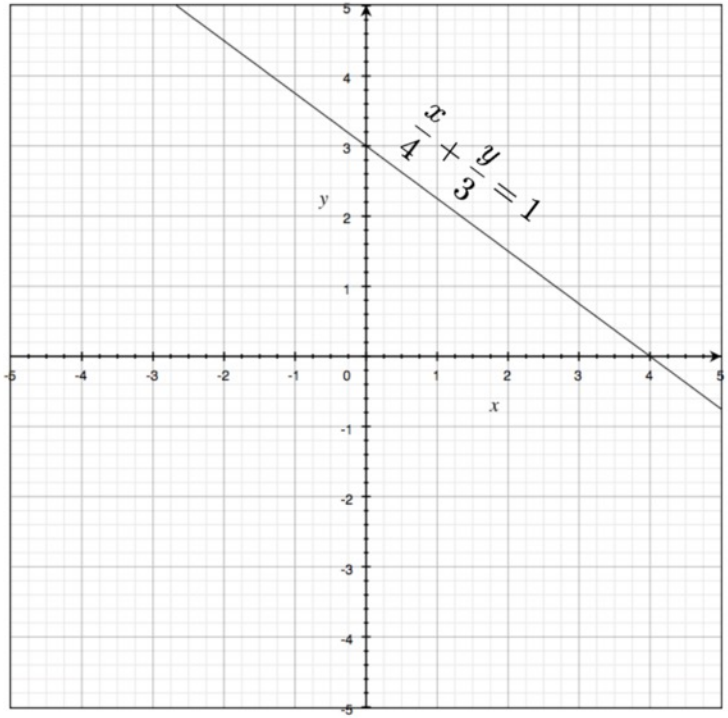


(Similar slope) $\frac{y-b}{x-0} = \frac{0-y}{a-x}$

$$ya + yx - ba + bx = -yx$$

$$ya + bx = ba$$

$$\frac{y}{b} + \frac{x}{a} = 1$$



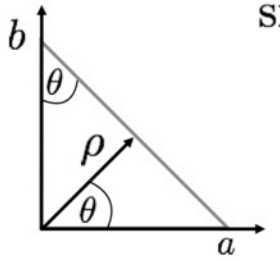
Line Models
Normal Form

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

Derivation:

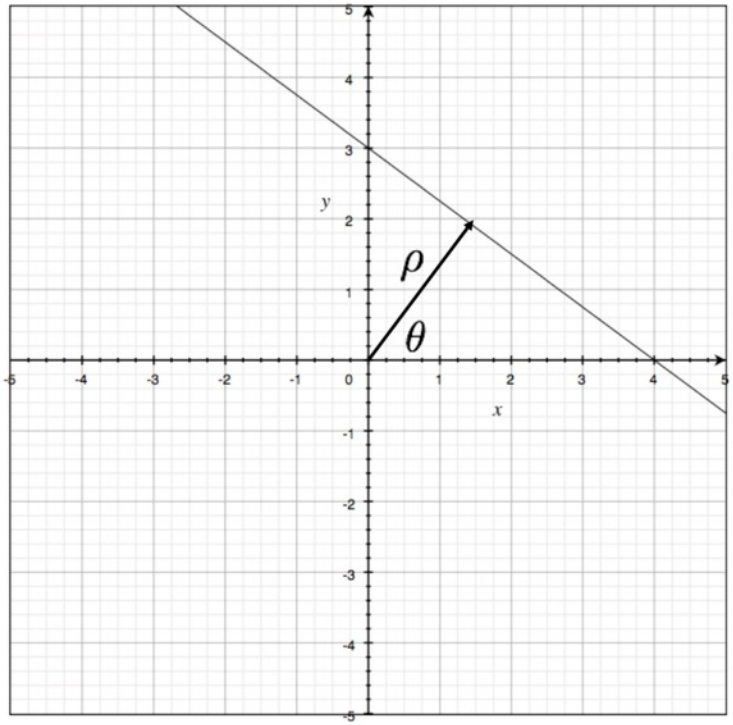
$$\cos\theta = \frac{\rho}{a} \rightarrow a = \frac{\rho}{\cos\theta}$$

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



plug into: $\frac{x}{a} + \frac{y}{b} = 1$

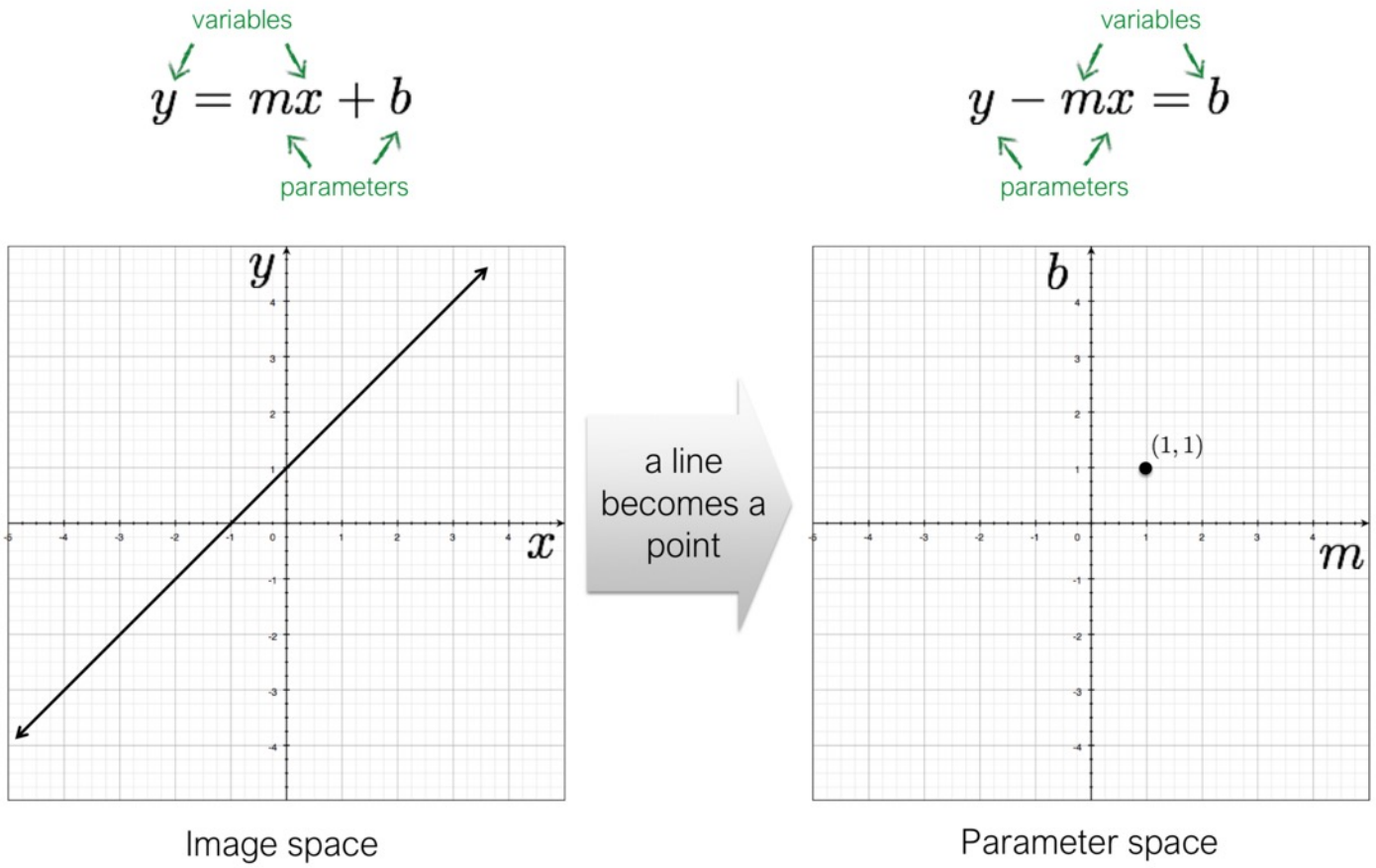
$$x\cos\theta + y\sin\theta = \rho$$



The Hough Transform is a procedure for finding lines in an image

- 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

Line-parameter space



Line-parameter space

variables

$$y = mx + b$$

parameters

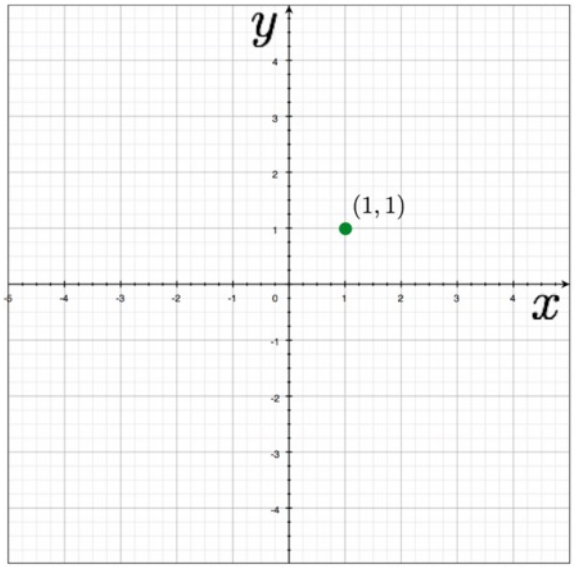
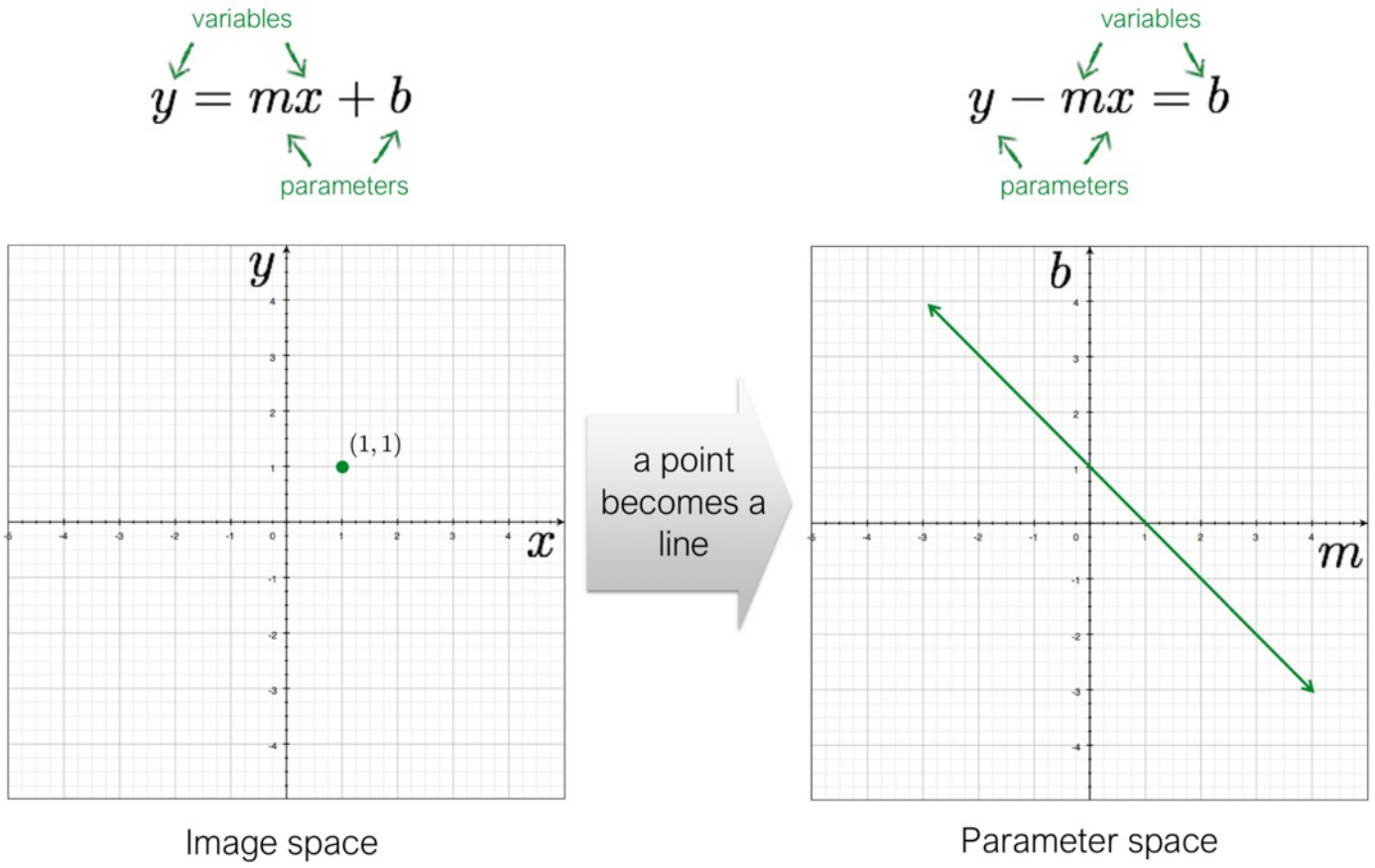


Image space

What if we have a point in image space? What does that become in line parameter space?

Line-parameter space



Line-parameter space

variables

$$y = mx + b$$

parameters

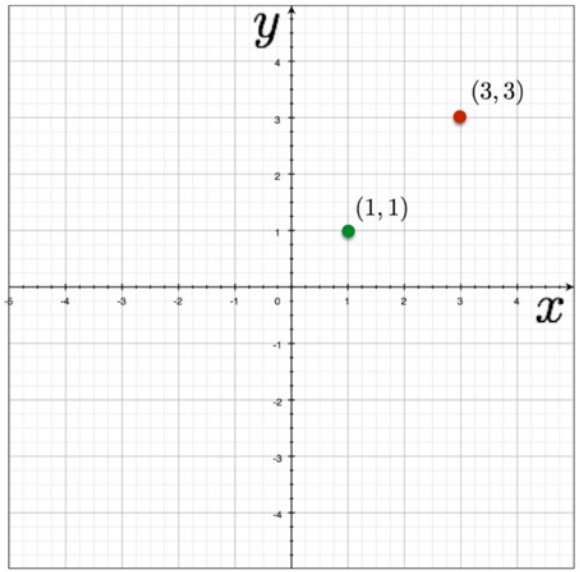
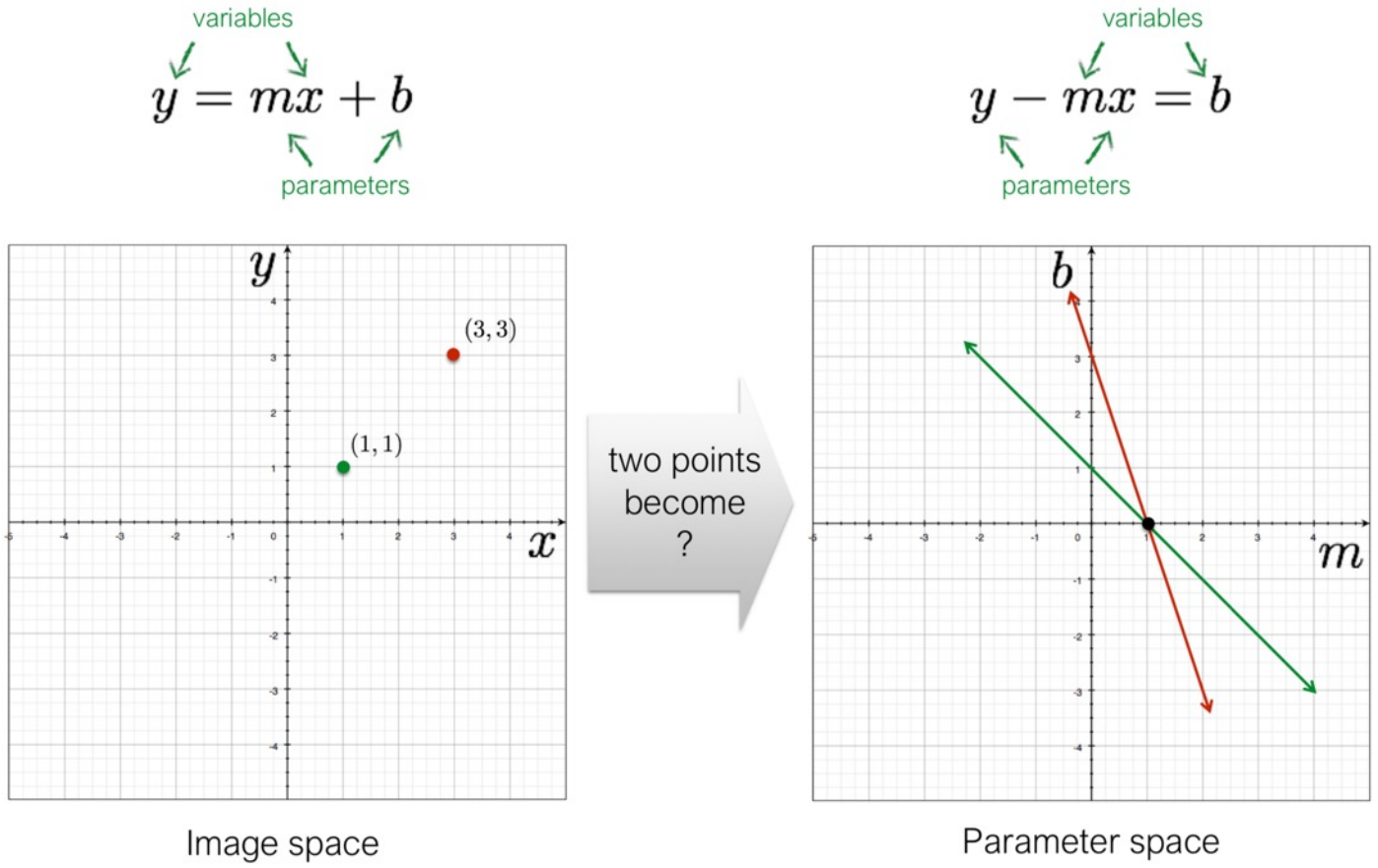


Image space

Line-parameter space



Line-parameter space

variables

$$y = mx + b$$

parameters

variables

$$y - mx = b$$

parameters

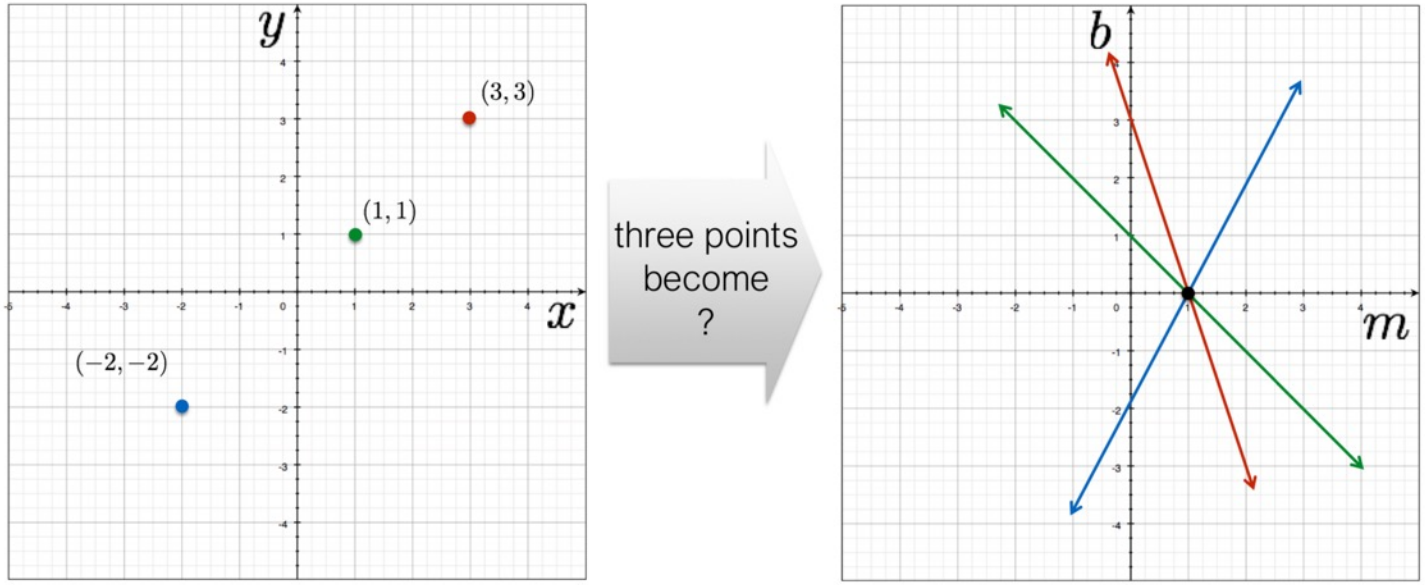


Image space

Parameter space

Line-parameter space

variables

$$y = mx + b$$

parameters

variables

$$y - mx = b$$

parameters

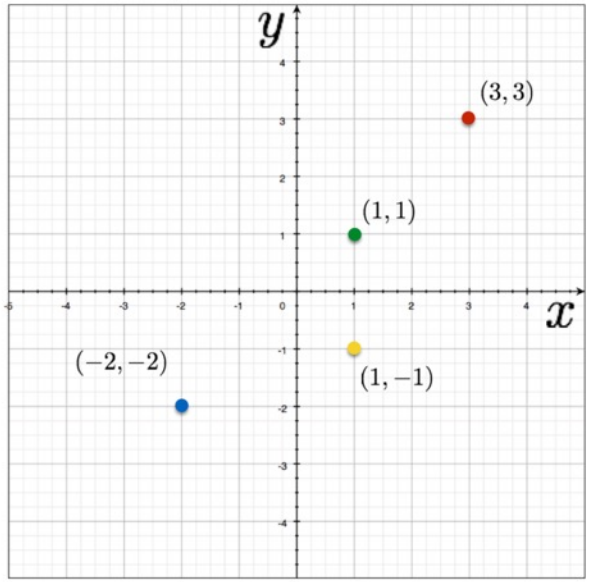
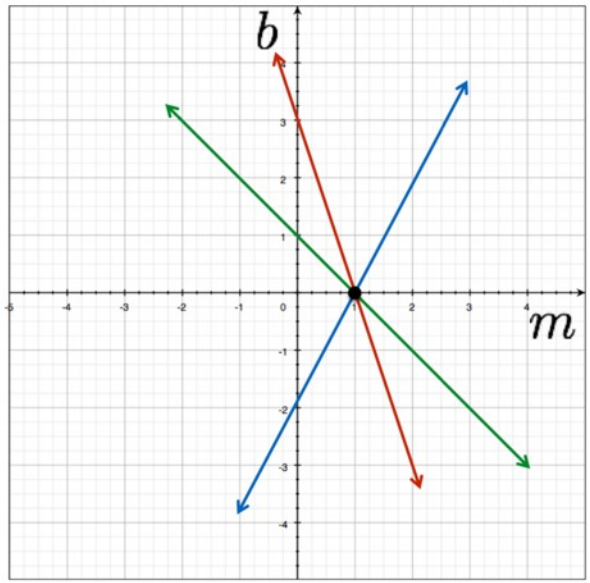


Image space

four points become ?



Parameter space

Line-parameter space

variables

$$y = mx + b$$

parameters

variables

$$y - mx = b$$

parameters

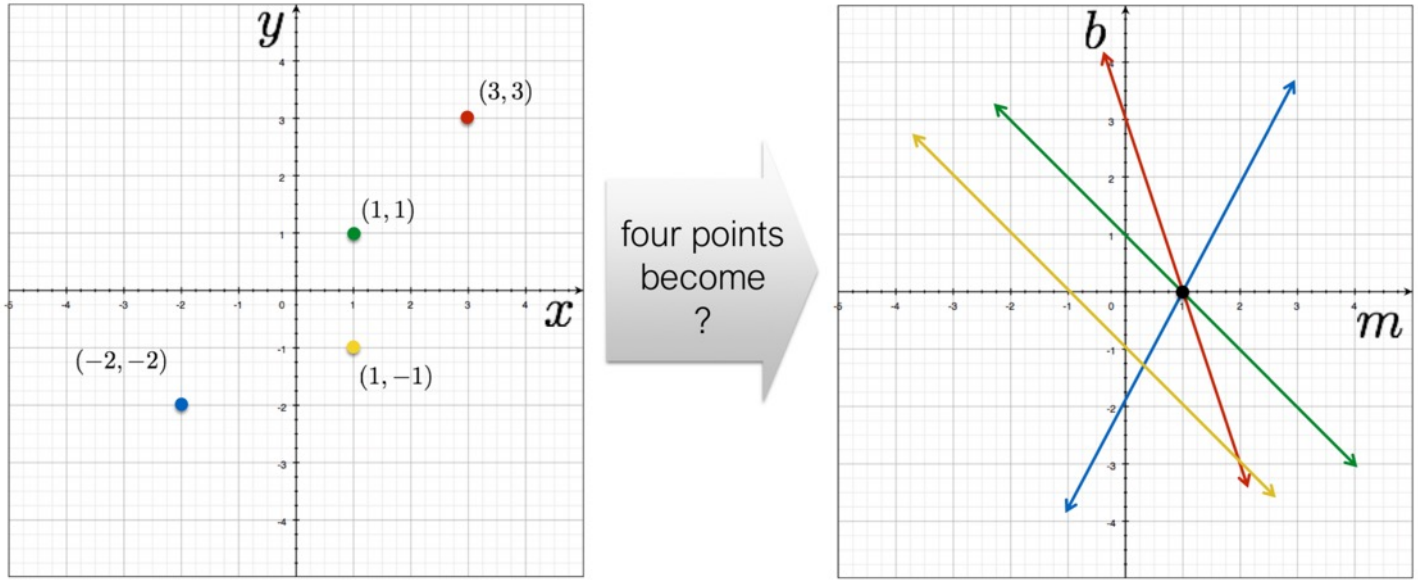


Image space

Parameter space

Line-parameter space

variables

$$y = mx + b$$

parameters

variables

$$y - mx = b$$

parameters

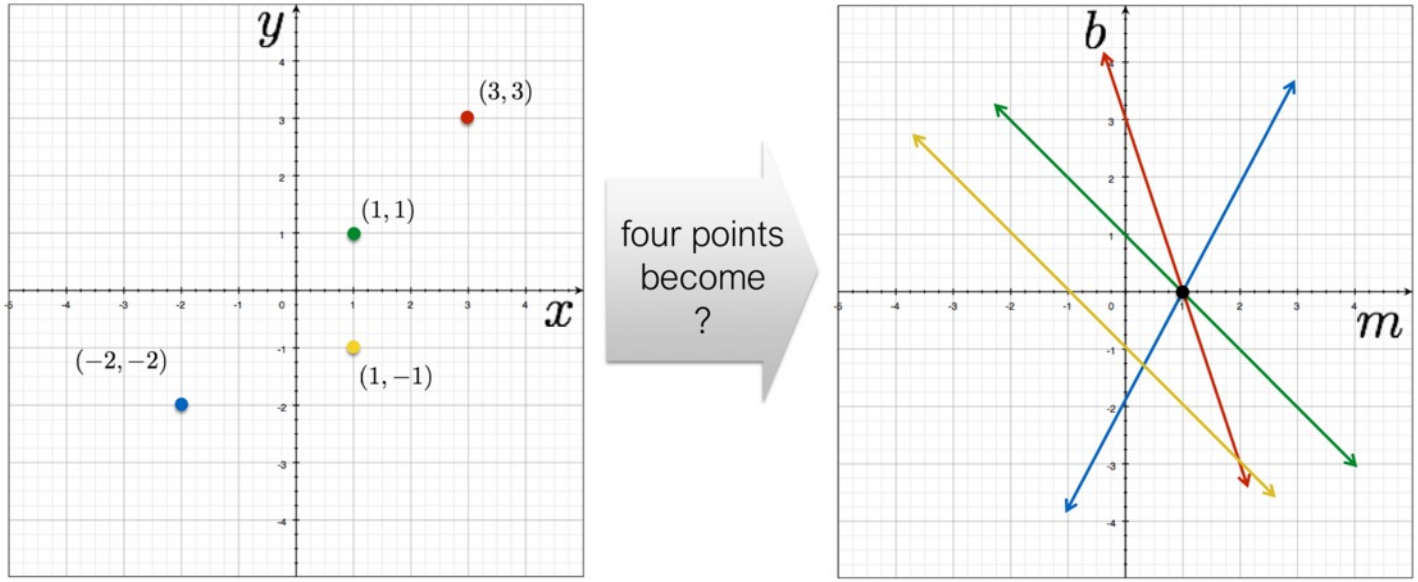


Image space

Parameter space

How do we pick the best-fitting line?

Line Detection by Hough Transform

Algorithm:

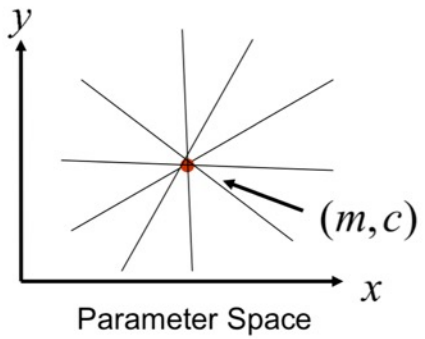
1. Quantize Parameter Space (m,c)

2. Create Accumulator Array $A(m,c)$

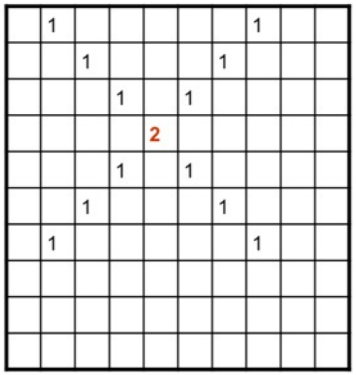
3. Set $A(m,c) = 0 \quad \forall m,c$

4. For each image edge (x_i, y_i)
   For each element in $A(m,c)$
   If (m,c) lies on the line: $c = -x_i m + y_i$
   Increment $A(m,c) = A(m,c) + 1$

5. Find local maxima in $A(m,c)$



$A(m,c)$



Usually, the Normal Form Parameterization is more robust

Use normal form:

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

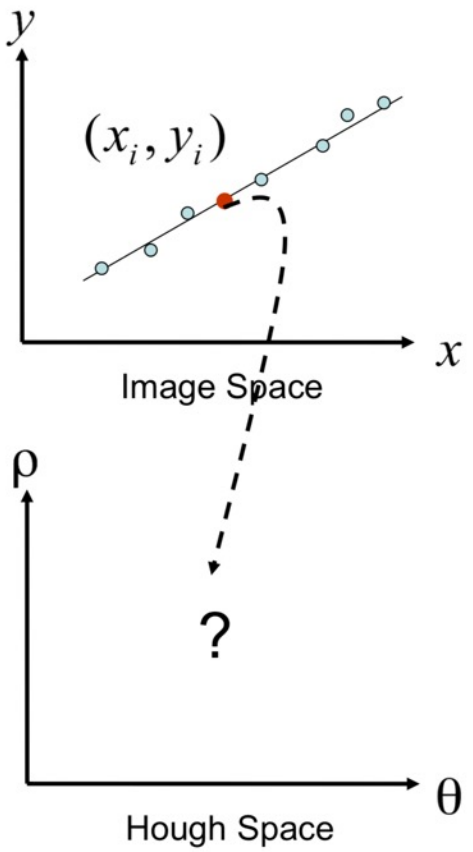
Given points (x_i, y_i) find (ρ, θ)

Hough Space Sinusoid

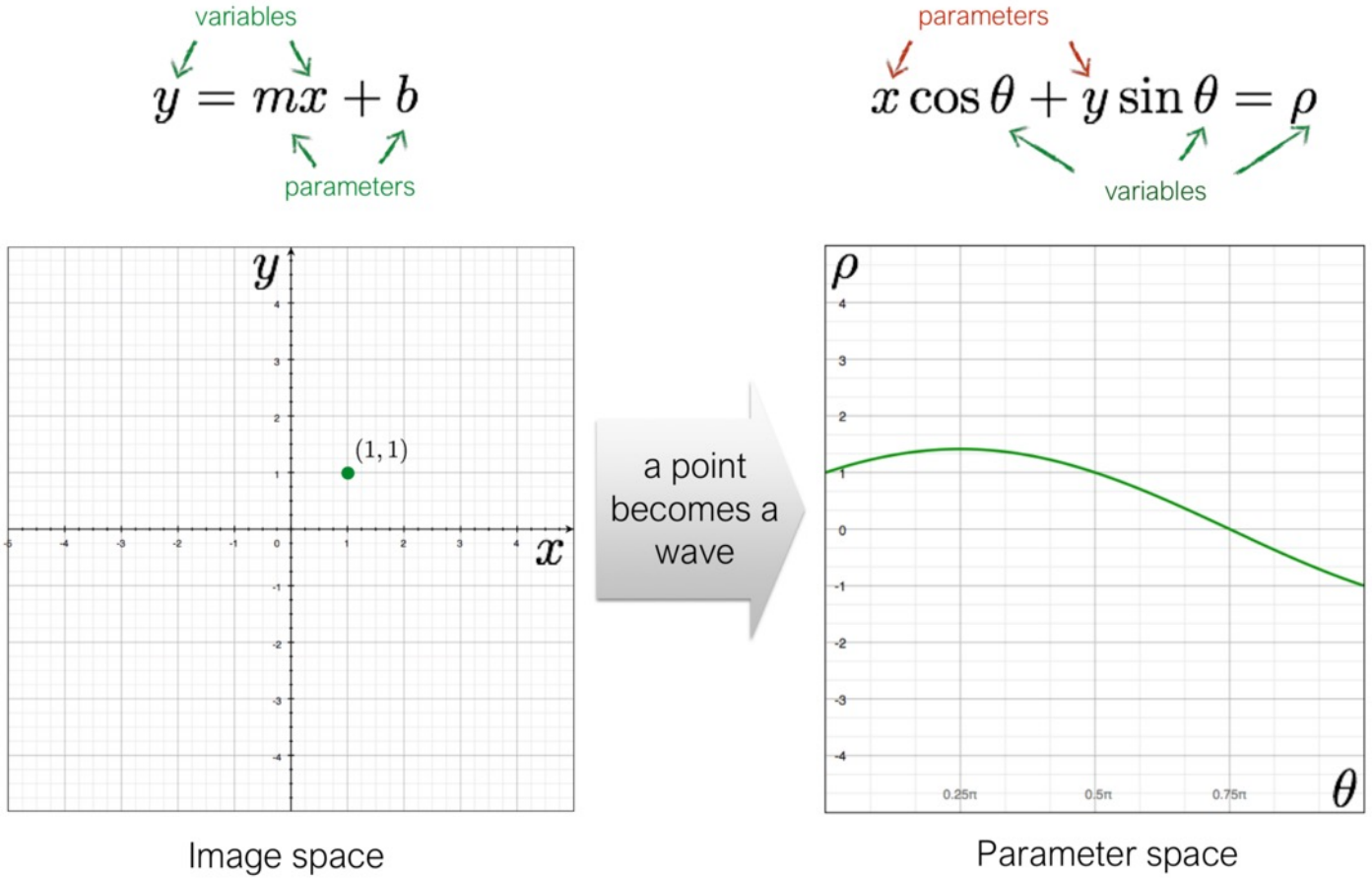
$$0 \le \theta \le 2\pi$$

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

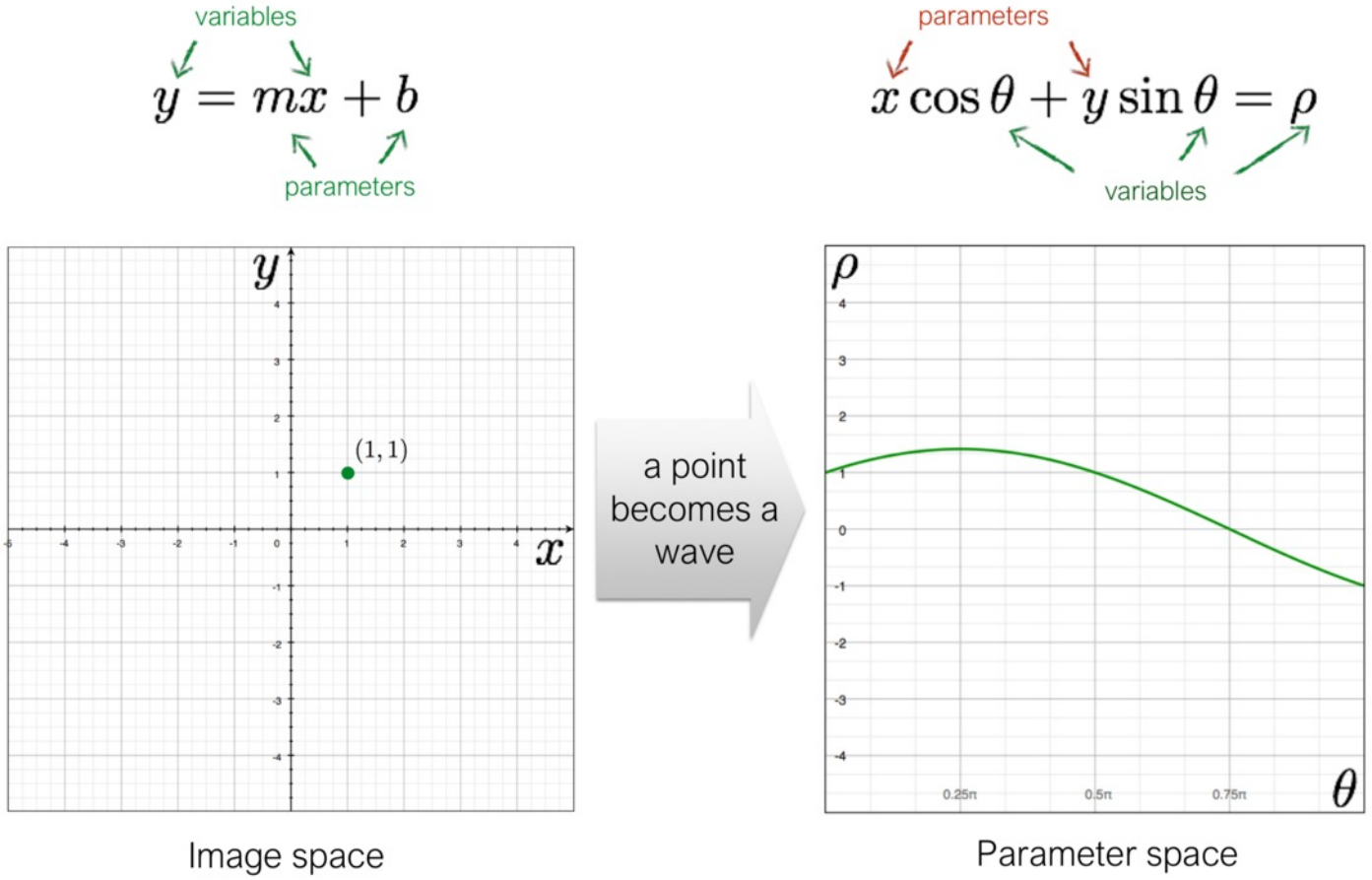
(Finite Accumulator Array Size)



In Normal Form Parameterization, points in image space map to sinusoids



In Normal Form Parameterization, points in image space map to sinusoids



In Normal Form Parameterization, points in image space map to sinusoids

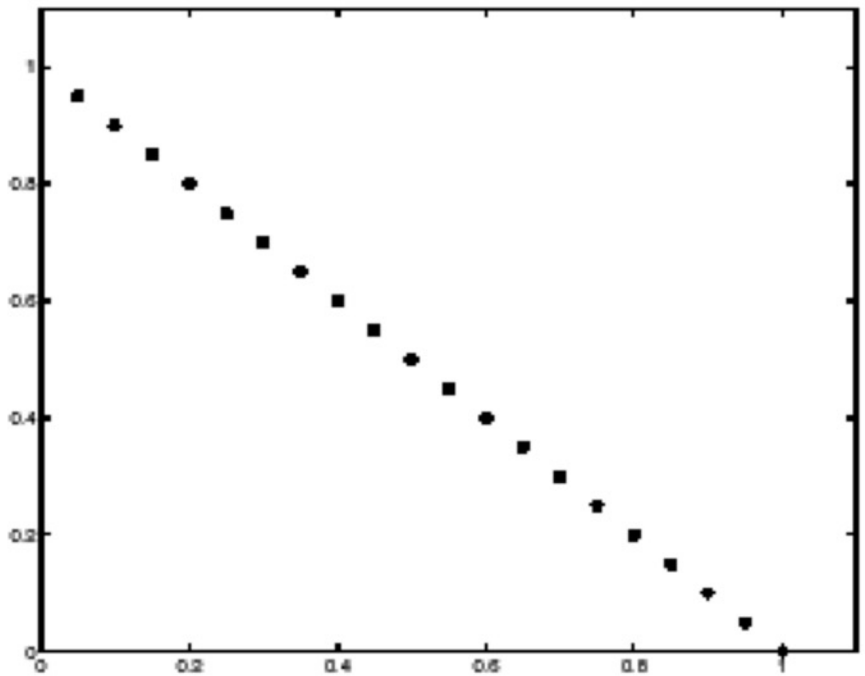
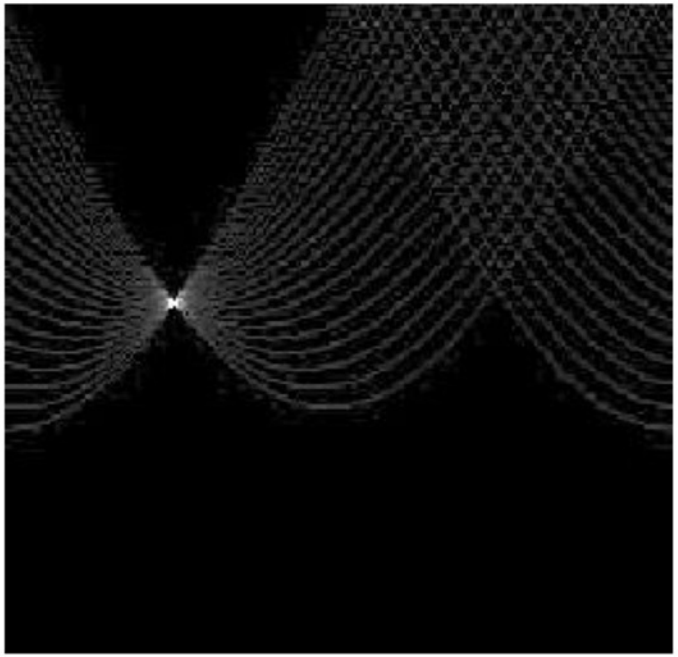
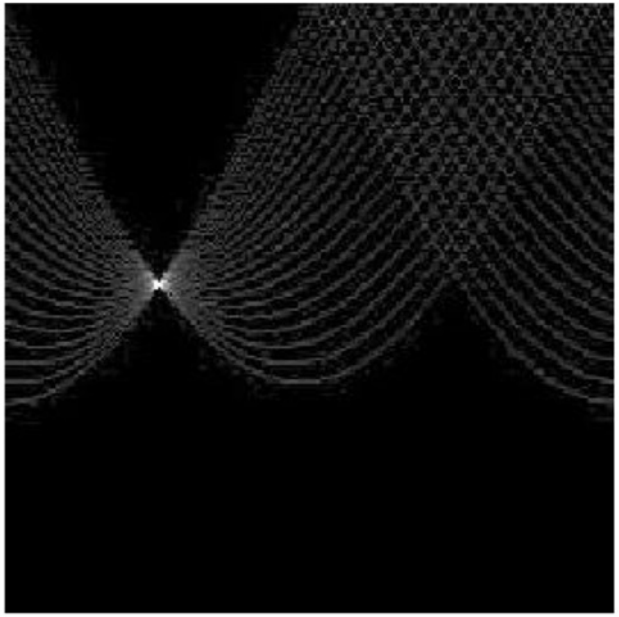


Image space

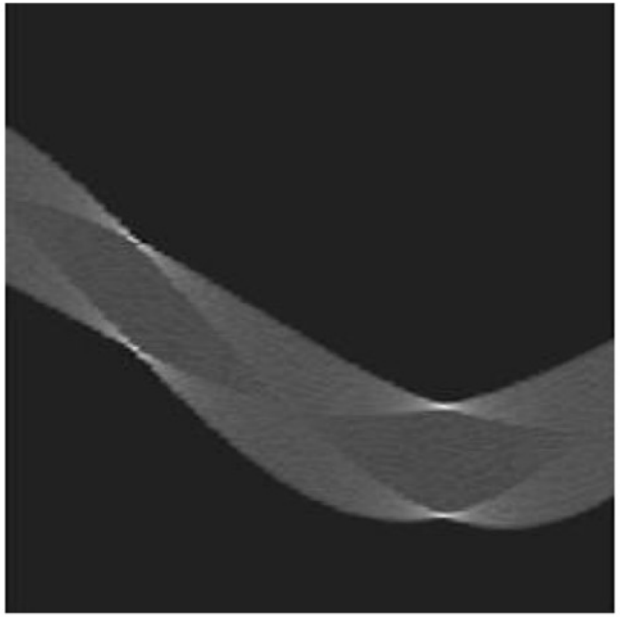


Votes

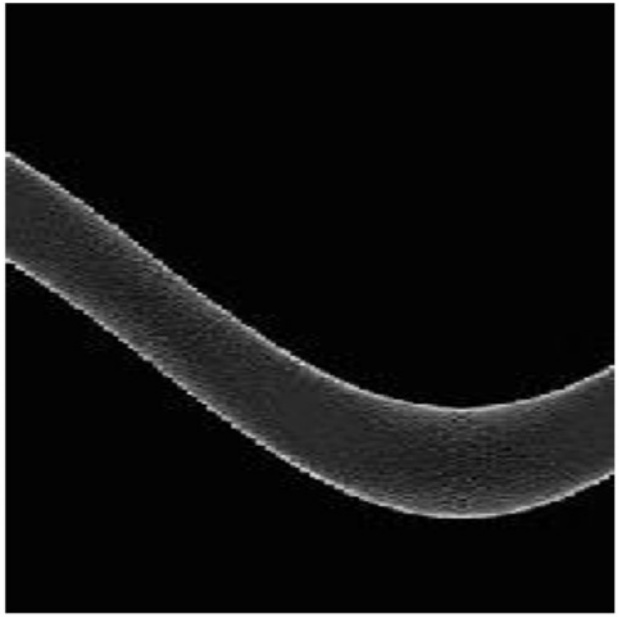
In Normal Form Parameterization, points in image space map to sinusoids



line

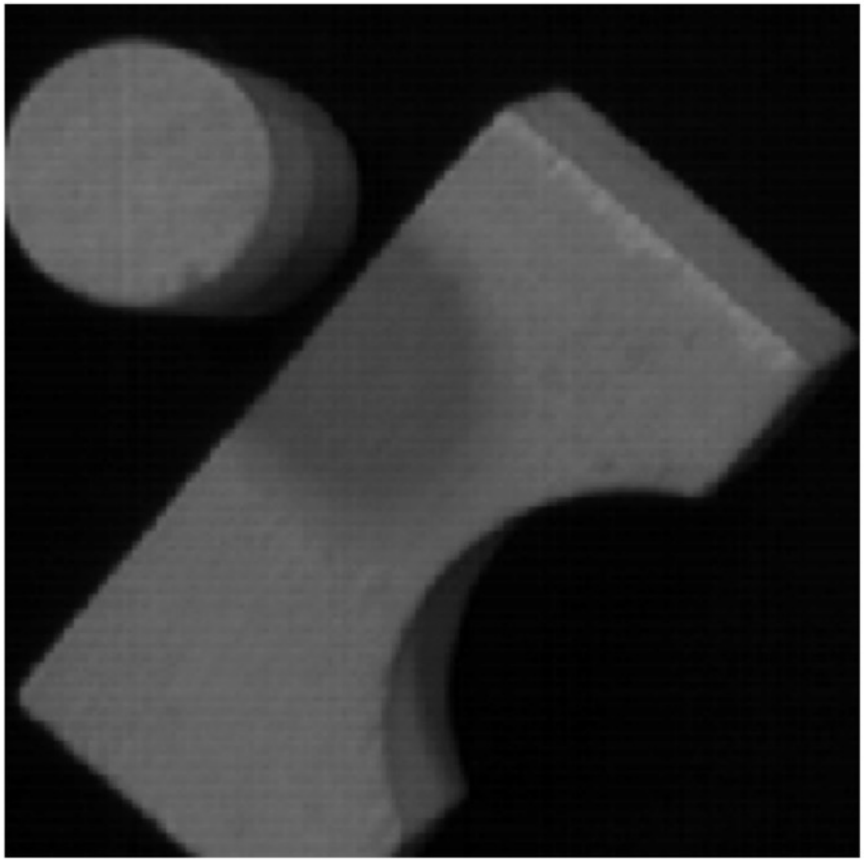


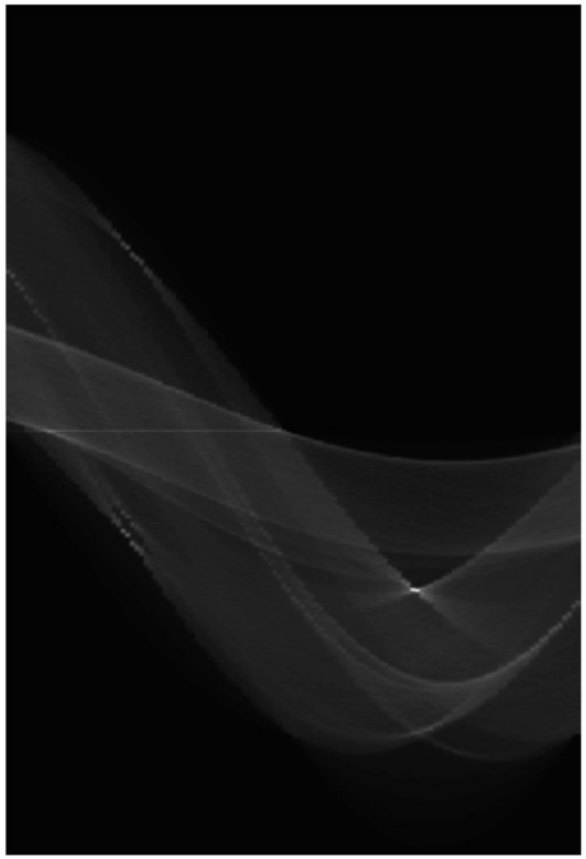
rectangle



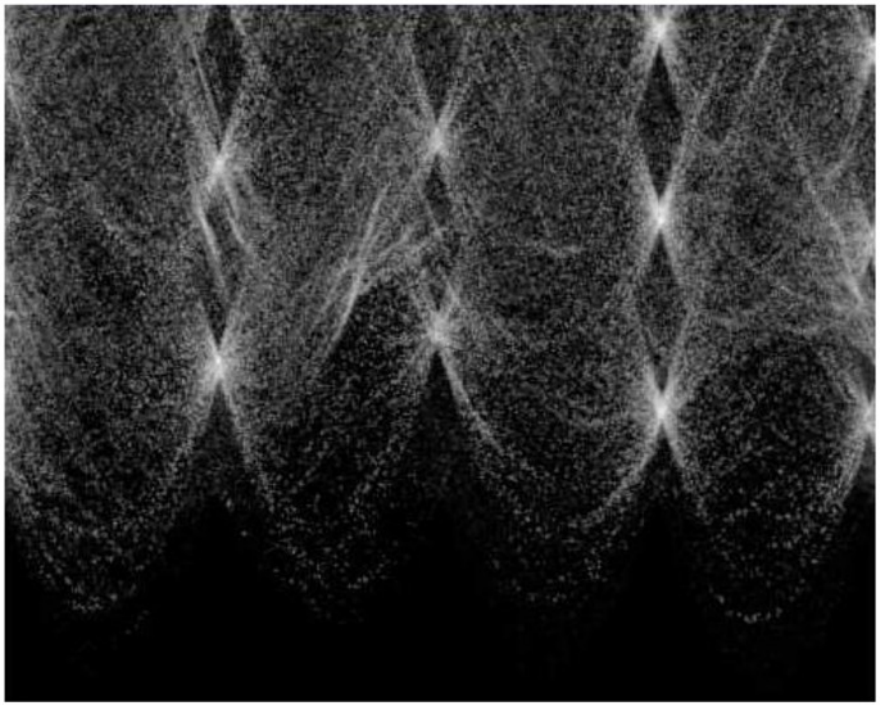
circle

Hough Transform Examples



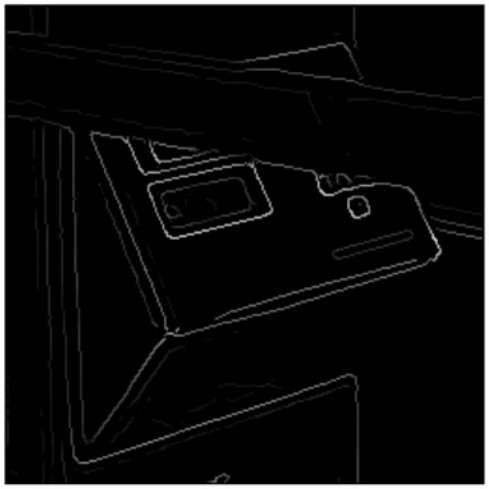


Hough Transform Examples

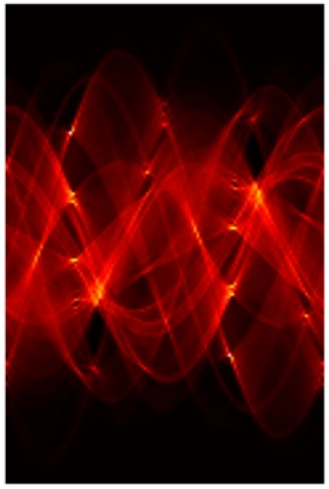


Hough Transform Examples

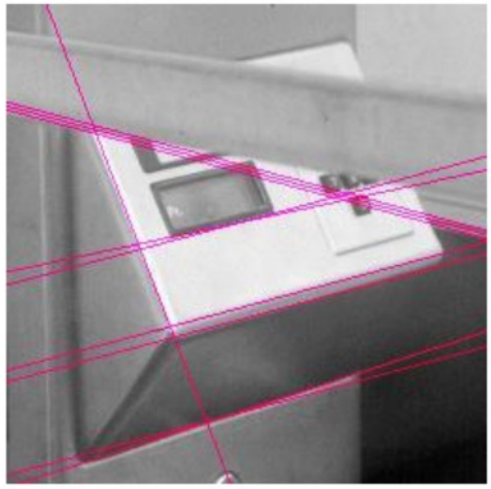
Original



Edges



parameter space



Hough Lines

In practice, measurements are noisy

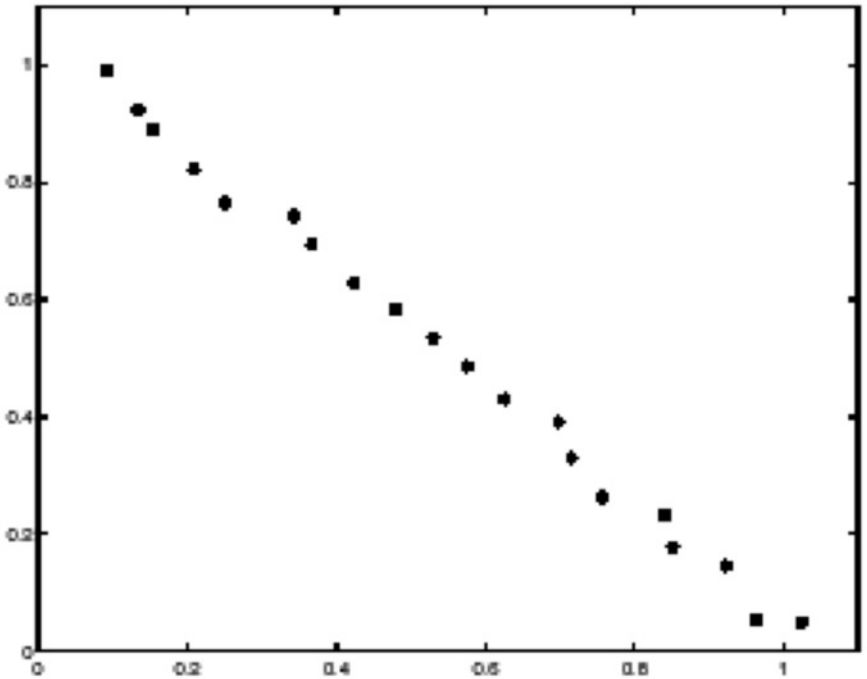
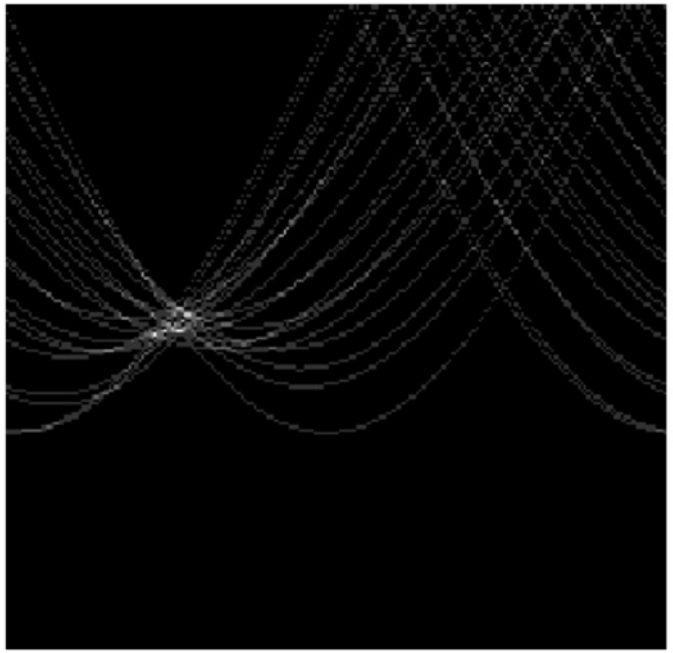


Image space



Votes

How might we handle noise?