

SciComp with Py

Hough Transform

Vladimir Kulyukin
Department of Computer Science
Utah State University



Outline

- Basic Concepts
 - Point-slope representation of lines
 - Parameterized representation of lines
 - 2D points & sine curves
- Hough Transform Algorithm
- Hough Transform in OpenCV



Basic Concepts

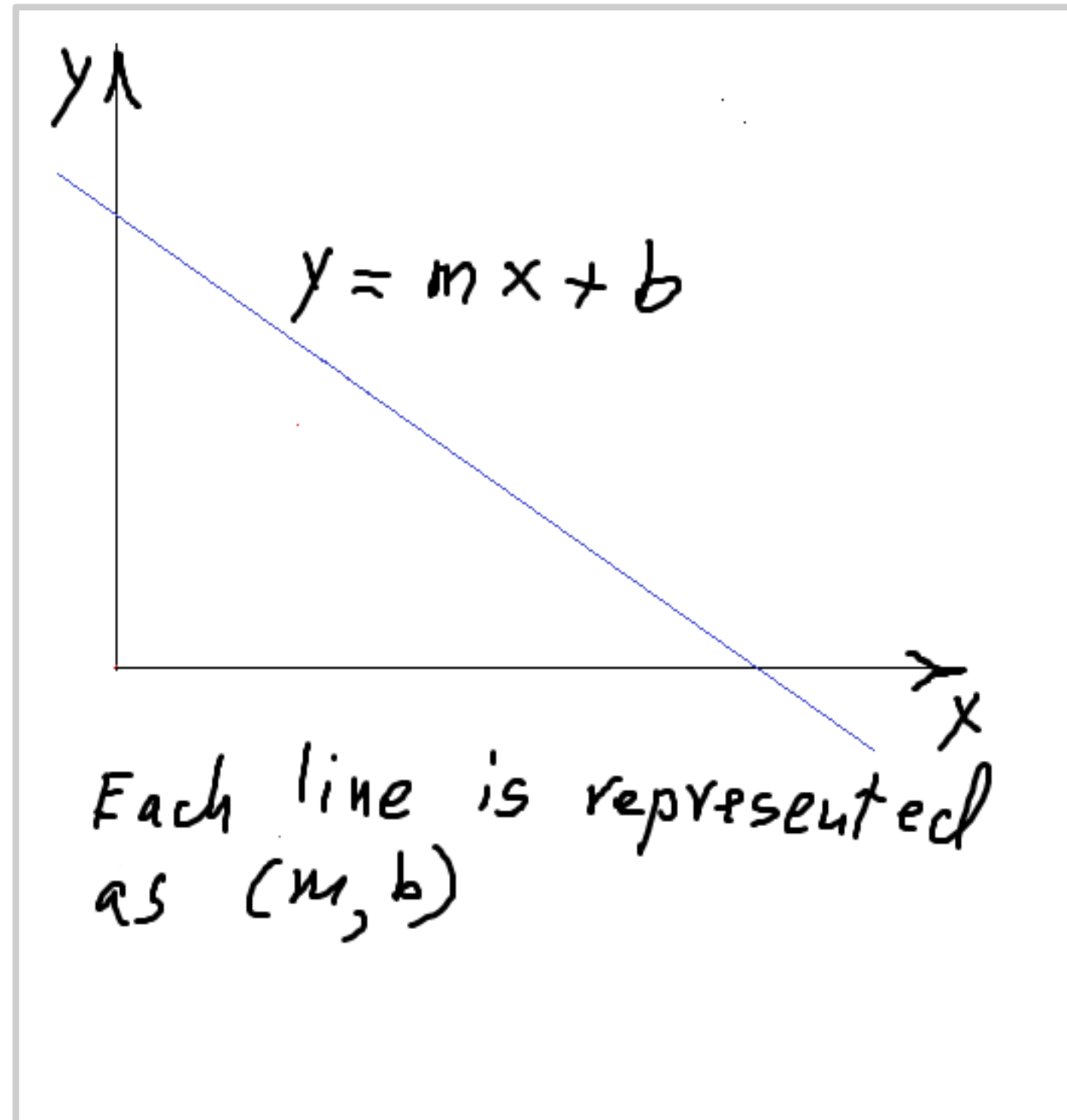


Hough Transform

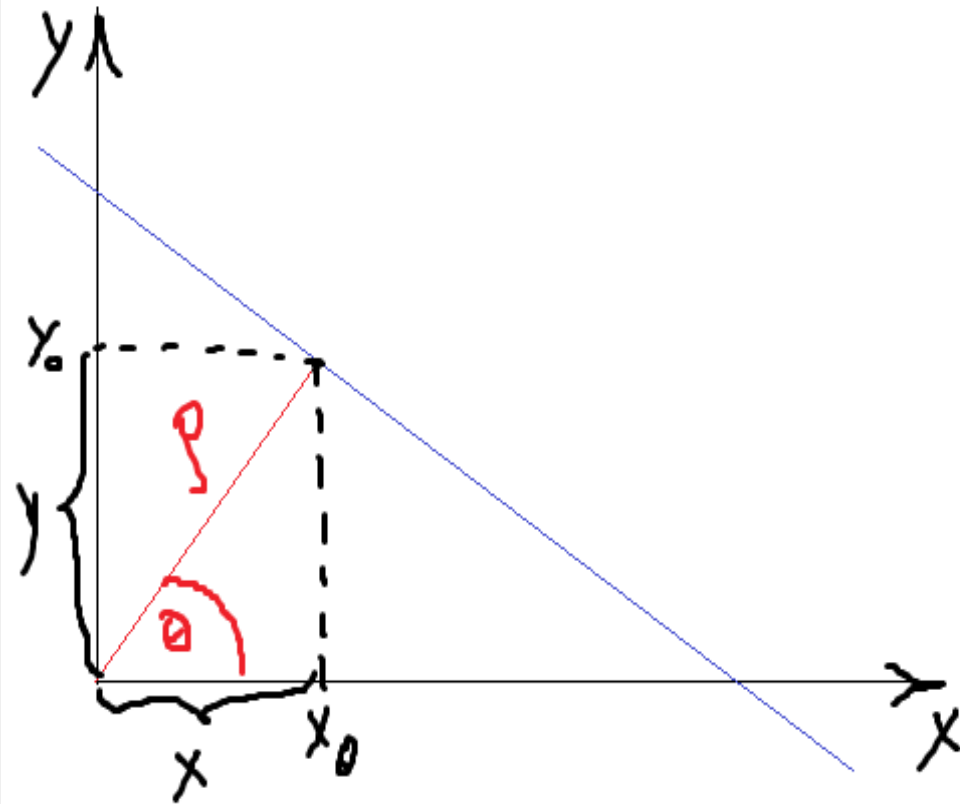
- Paul Hough invented the concept in 1959 and patented it in 1962
- Richard Duda and Peter Hart used Hough's idea in image analysis and introduced it to computer vision and image analysis as the **Hough Transform**
- It is used in computer vision primarily for line detection



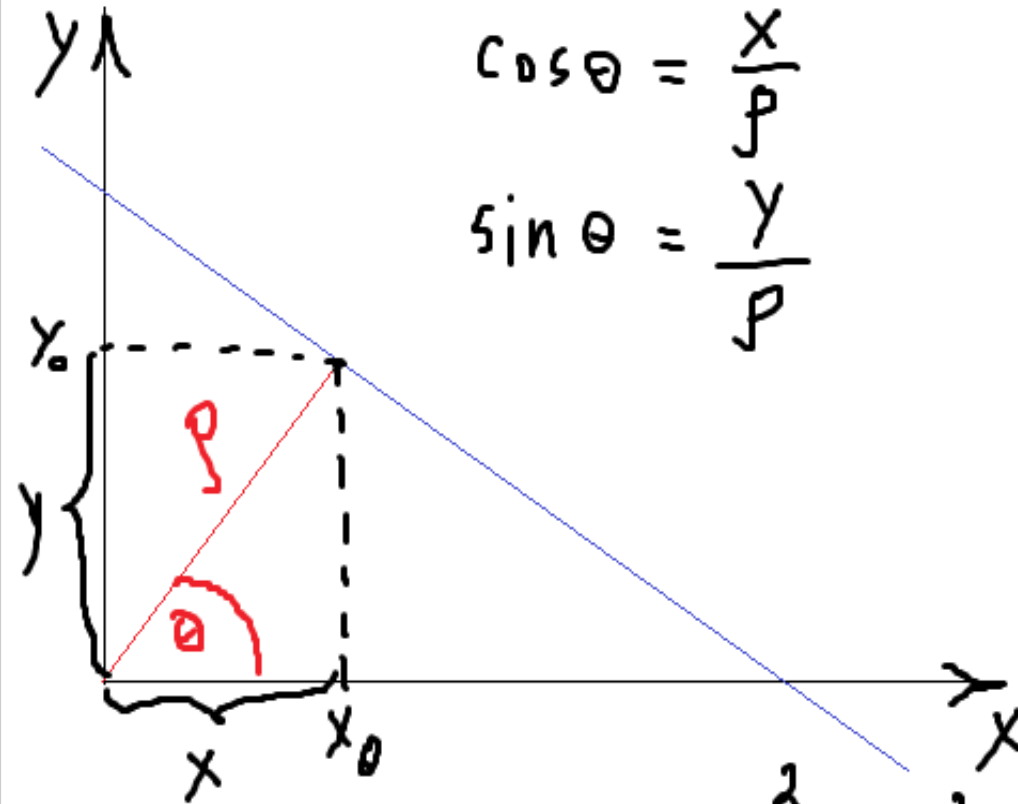
Point-Slope Representation of Lines



Parameterized Representation of Lines



Each line is represented
as (ρ, θ)

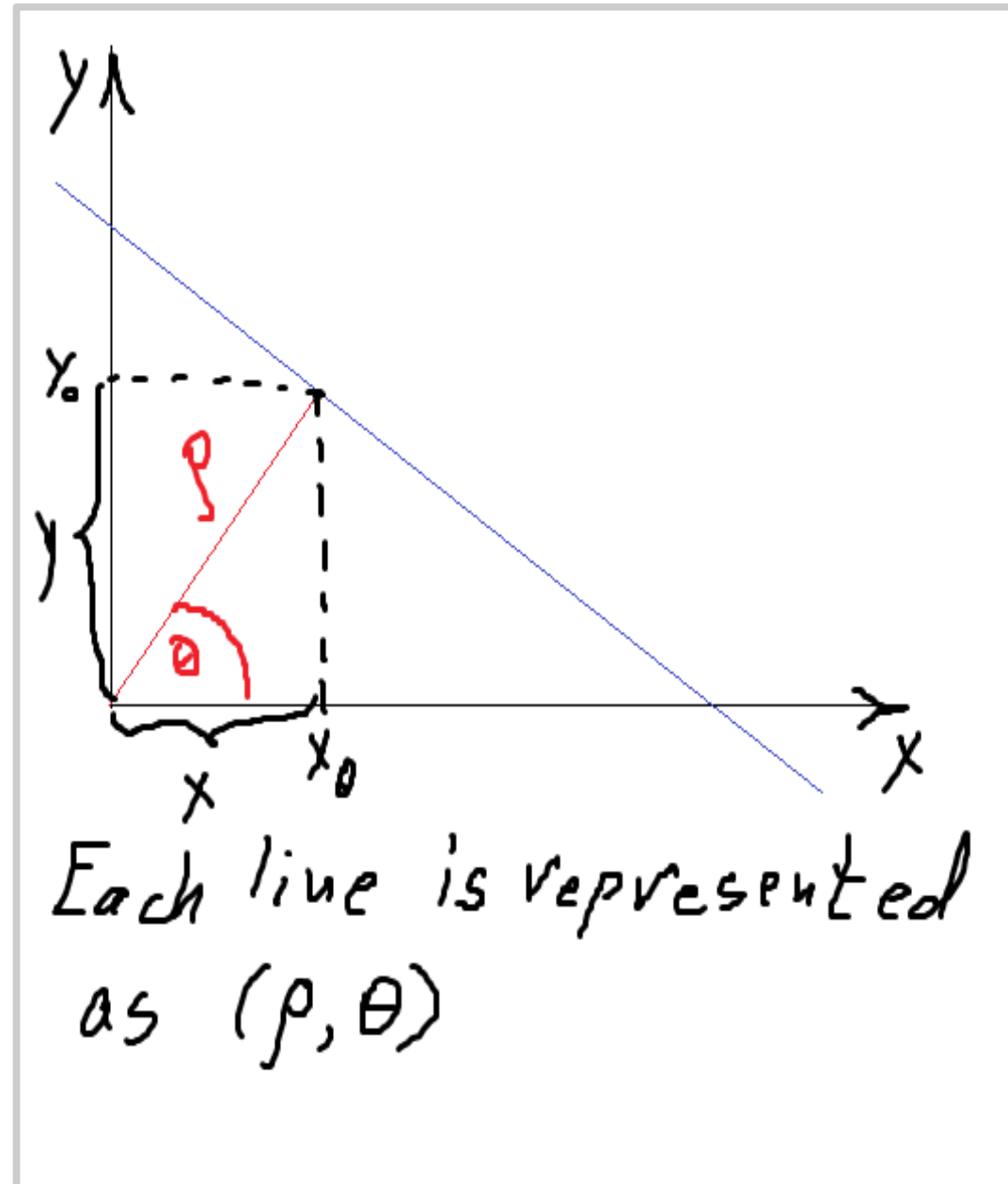
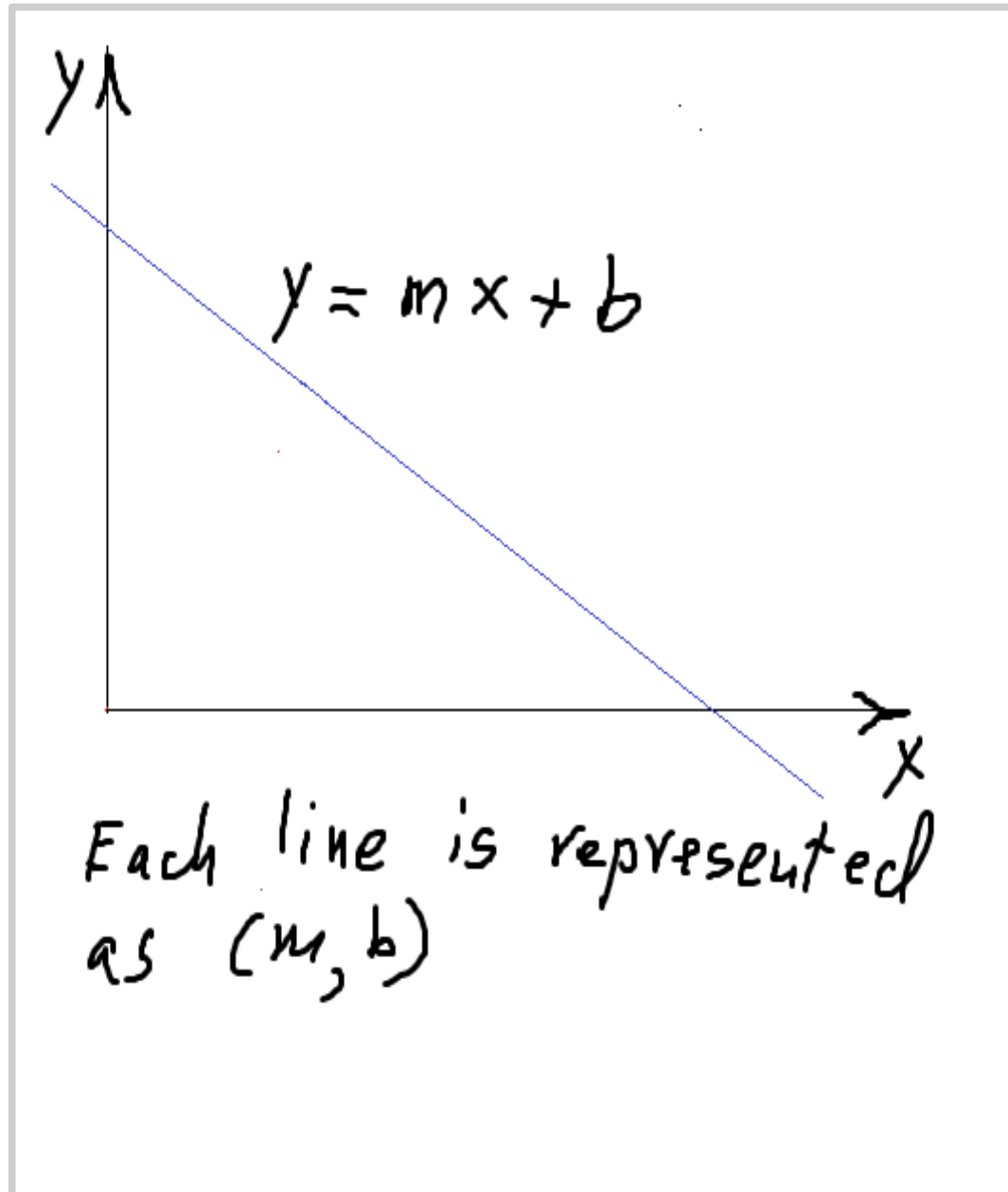


$$\cos \theta = \frac{x}{\rho}$$
$$\sin \theta = \frac{y}{\rho}$$

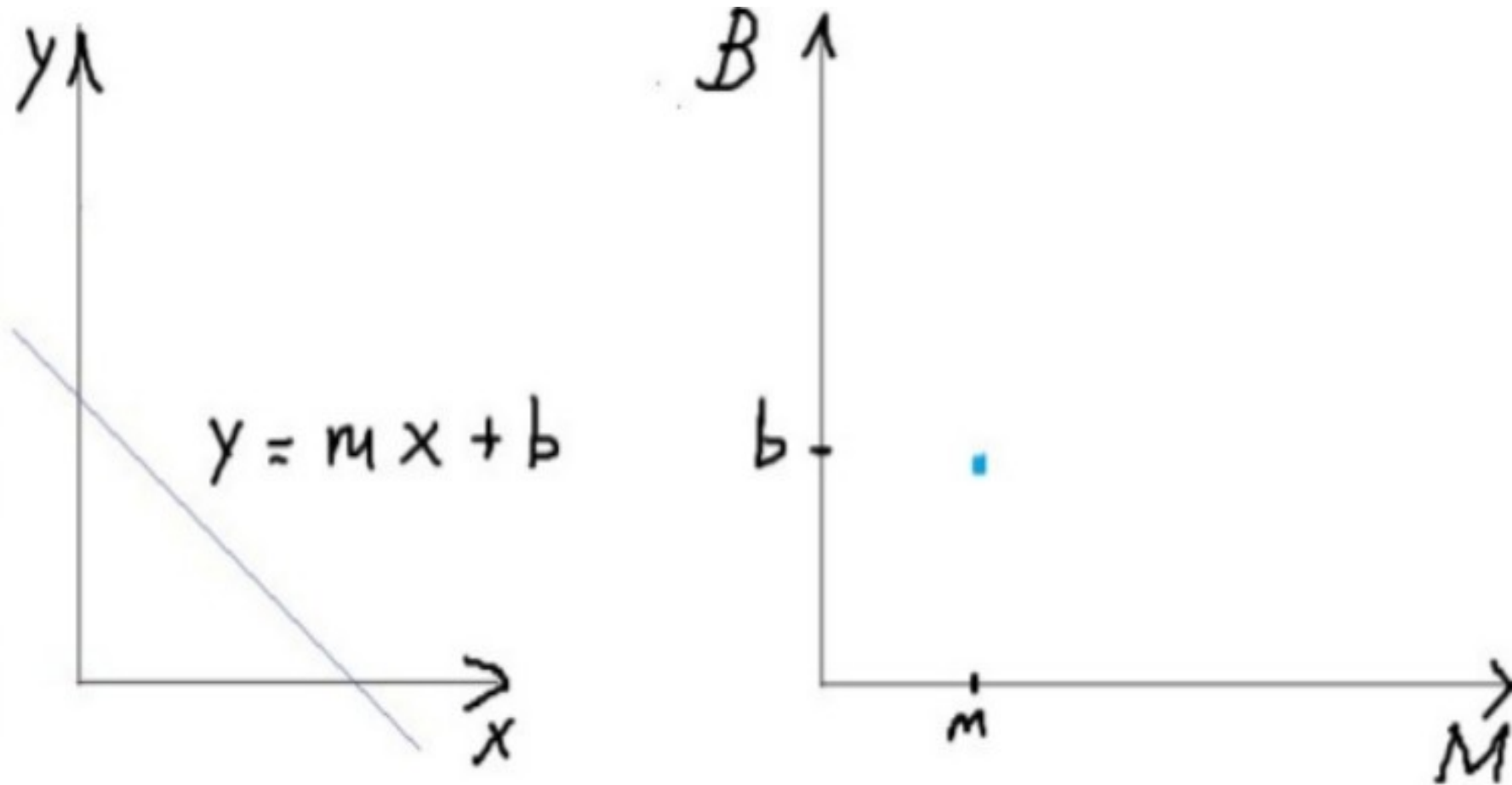
$$\rho^2 = x^2 + y^2 \Rightarrow \rho = \frac{x^2}{\rho} + \frac{y^2}{\rho} =$$
$$x \frac{x}{\rho} + y \frac{y}{\rho} = x \cos \theta + y \sin \theta$$



Two Representations Side by Side



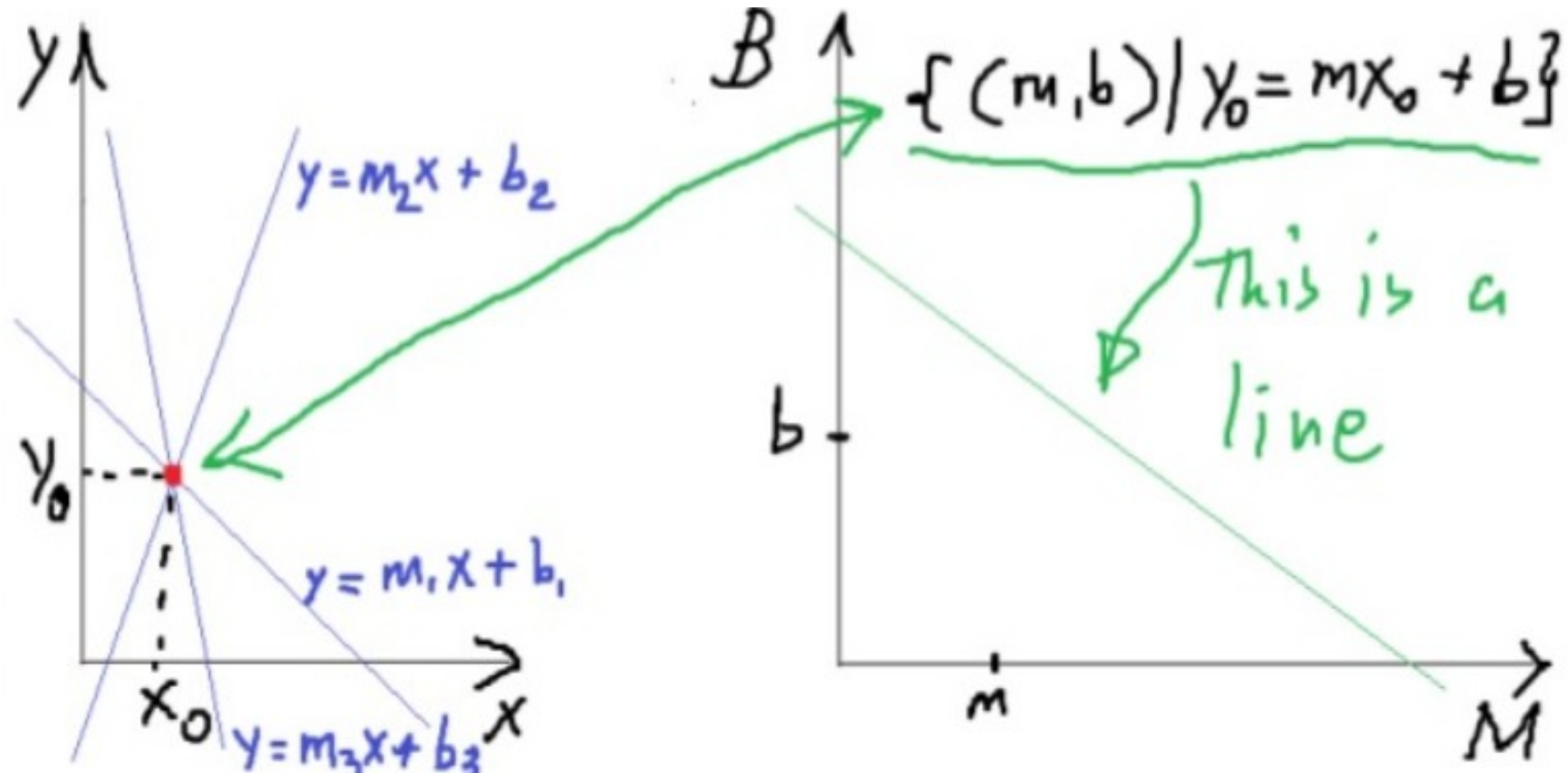
From Euclid Plane Lines to Hough Plane Points



A line $y = mx+b$ in Euclid Plane (left) corresponds to a point (m, b) in Hough Plane (right)



From Euclid Points to Hough Plane Lines



A point (x_0, y_0) in Euclid Plane (left) corresponds to a line in Hough Plane (right)

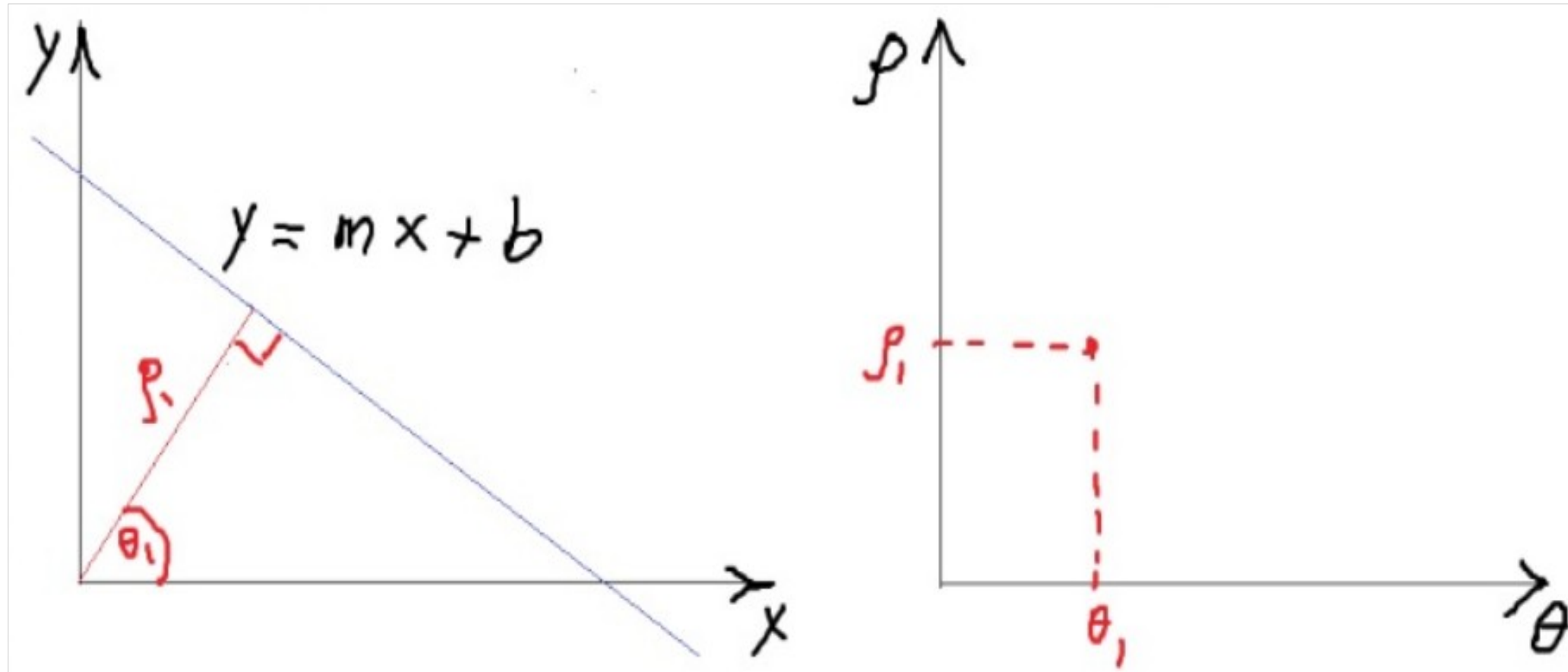


Problems with Hough Plane

- Problem 1: Vertical lines cannot be represented because their slopes are not defined
- Problem 2: Theoretically, values of M and B range from $-\infty$ to $+\infty$
- Problem 1 can be solved with meaningful domain-specific thresholds
- Problem 2 can be solved with finite $M \times B$ matrices



Rho-Theta Representation of Hough Plane



Instead of using M and B to represent Hough Plane, we can use Rho and Theta



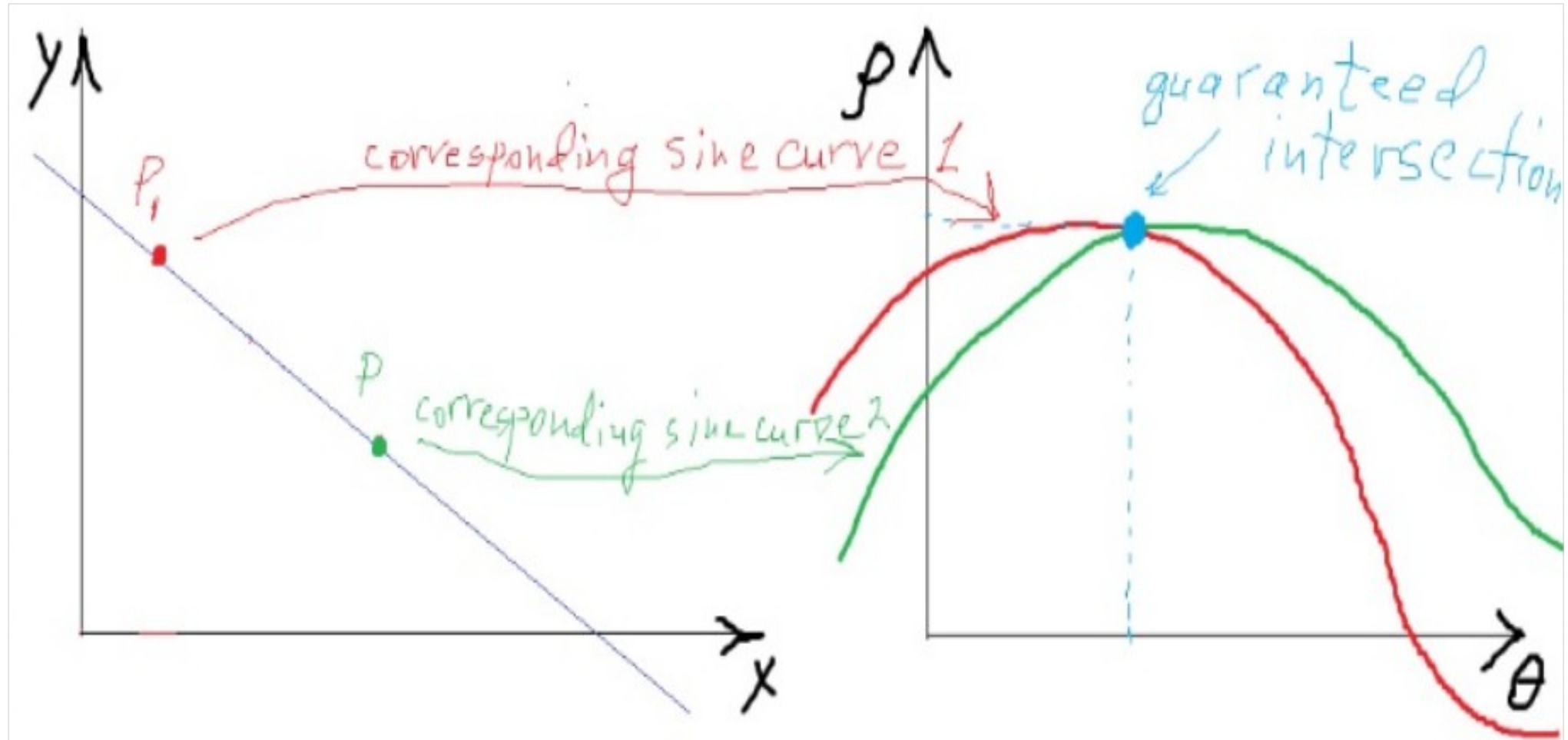
Rho-Theta Representation



All (r, θ) pairs corresponding to all lines passing through a point (x_0, y_0) in Euclid Plane form a sine curve in Hough Plane; this is astonishing when you think about it!



Rho-Theta Representation



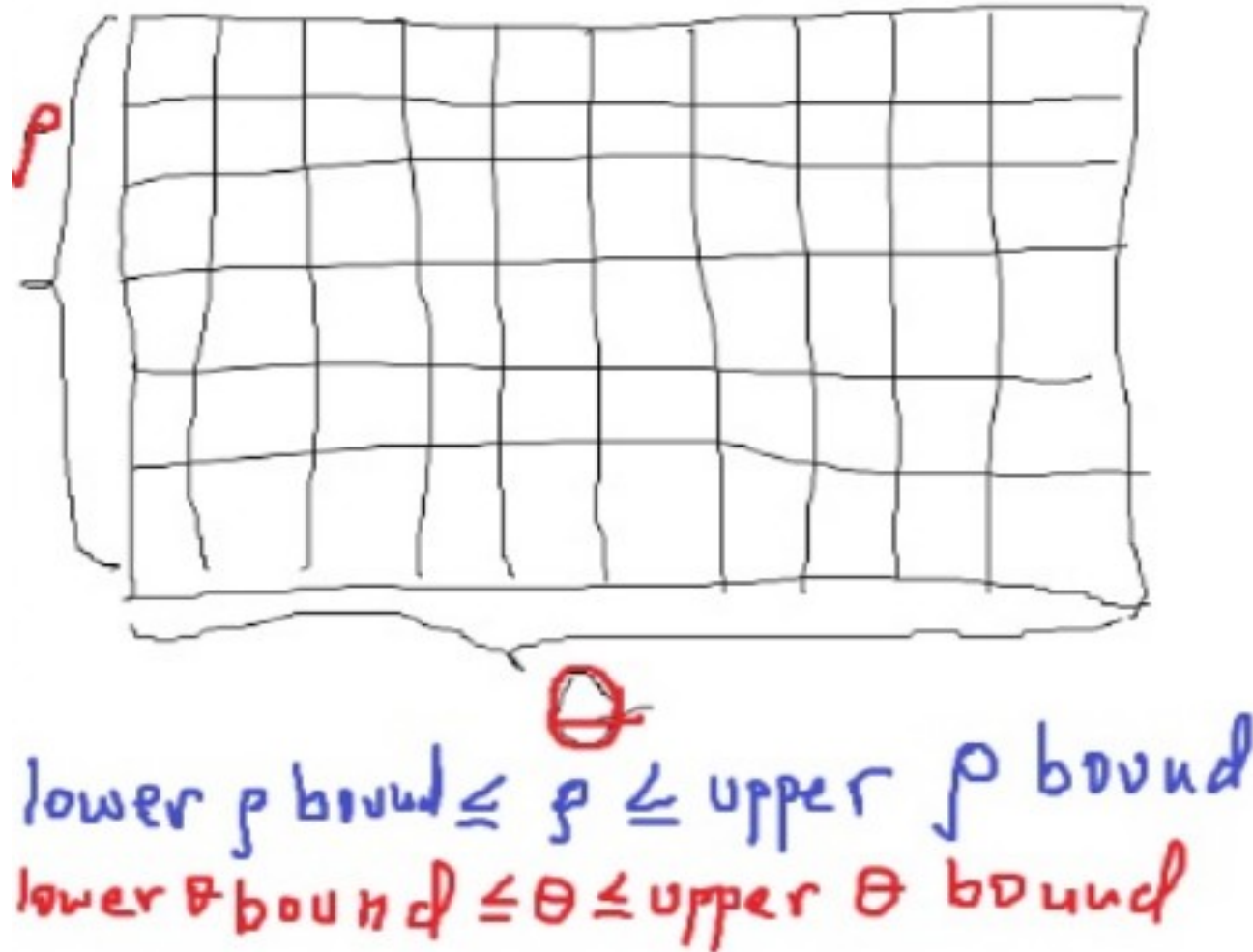
Another remarkable fact: The sine curves that correspond to any two collinear points in Euclid Plane (left) are guaranteed to intersect in Hough Plane (right)



HT Algorithm



Step 1: Create a Rho-Theta Table



Choose suitable integer bounds for Rho and Theta and create 2D matrix; let us call this matrix HT (i.e., Hough Transform)



Step 2: Compute Gradients

Given image Img (2D matrix), compute gradients at each cell of Img (see lecture on edge detection as derivative of light on how to compute gradients).



Step 3: Compute HT Values

For each point $P(x, y)$ in Img with sufficiently large gradient

For each value th of Theta in $[0, 180]$

$\rho = \text{int}(x * \cos(th) + y * \sin(th))$

$HT[th, \rho] += 1$

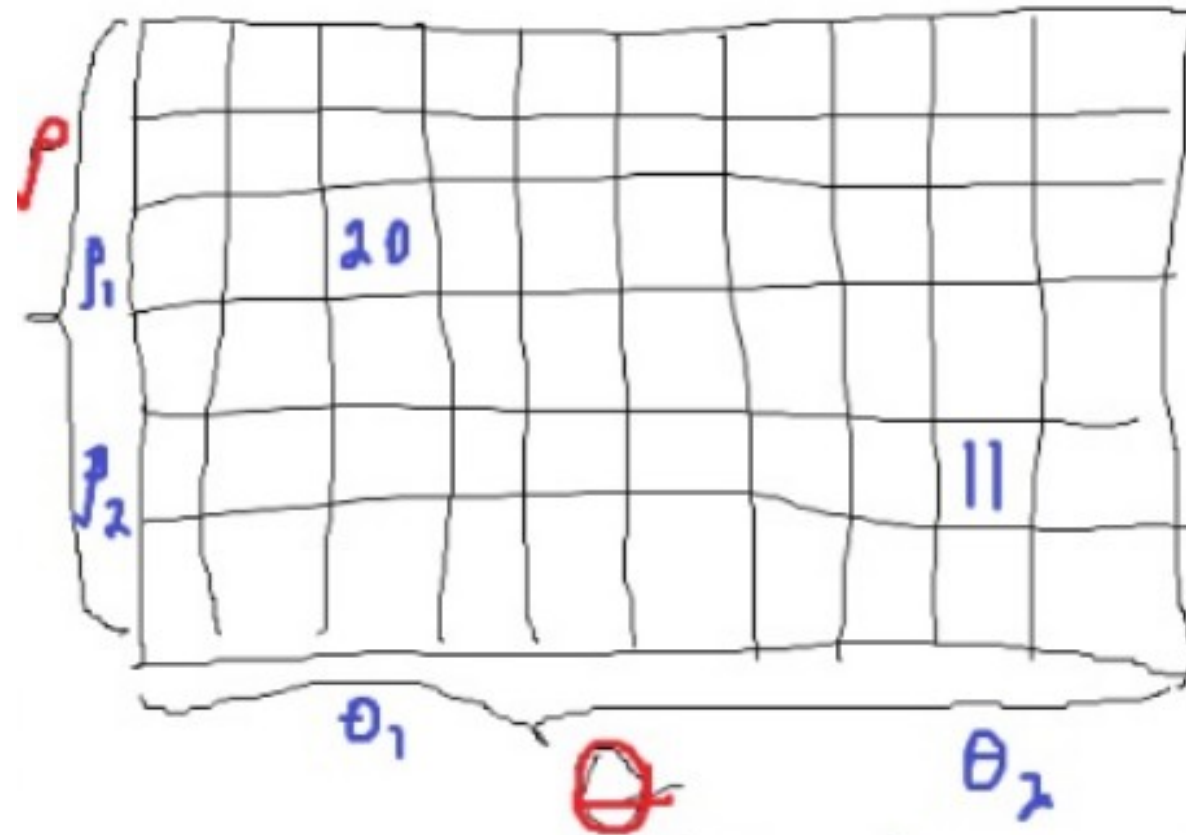


Step 4: Select HT Cells

Select those cells in $HT[th, \rho]$ for which the integer value in $HT[th, \rho]$ is above a threshold. Recall that each cell in $HT[th, \rho]$ represents a line in Euclid Space. The selected cells correspond to likely lines. The integer values in HT are sometimes called support levels.



Step 4: Example



Suppose support level = 10, then
 (p_1, θ_1) and (p_2, θ_2) clear it



Fundamental Question

What does it mean when $HT[\rho, \theta]$ has a large support level?



Answer

It means that there is likely to be a (ρ, θ) line in the image Img .



Hough Transform in OpenCV



Two HT Methods in OpenCV

Determines number of rows
in HT table

Determines number of
columns in HT table

This is support level
threshold

```
cv2.HoughLines(image, rho_accuracy, theta_accuracy, support_level)
```

```
cv2.HoughLinesP(image, rho_accuracy, theta_accuracy, support_level,  
min_len, max_gap)
```

Minimum length of lines

Max gap in lines



References

- <http://en.wikipedia.org/wiki/OpenCV>
- <http://opencv.org/>
- R. Laganier. OpenCV 2 Computer Vision Application Programming Cookbook.

