Theories of Canny edge detector and Hough Transformation

Jr Fang, Liou

Canny edge detector

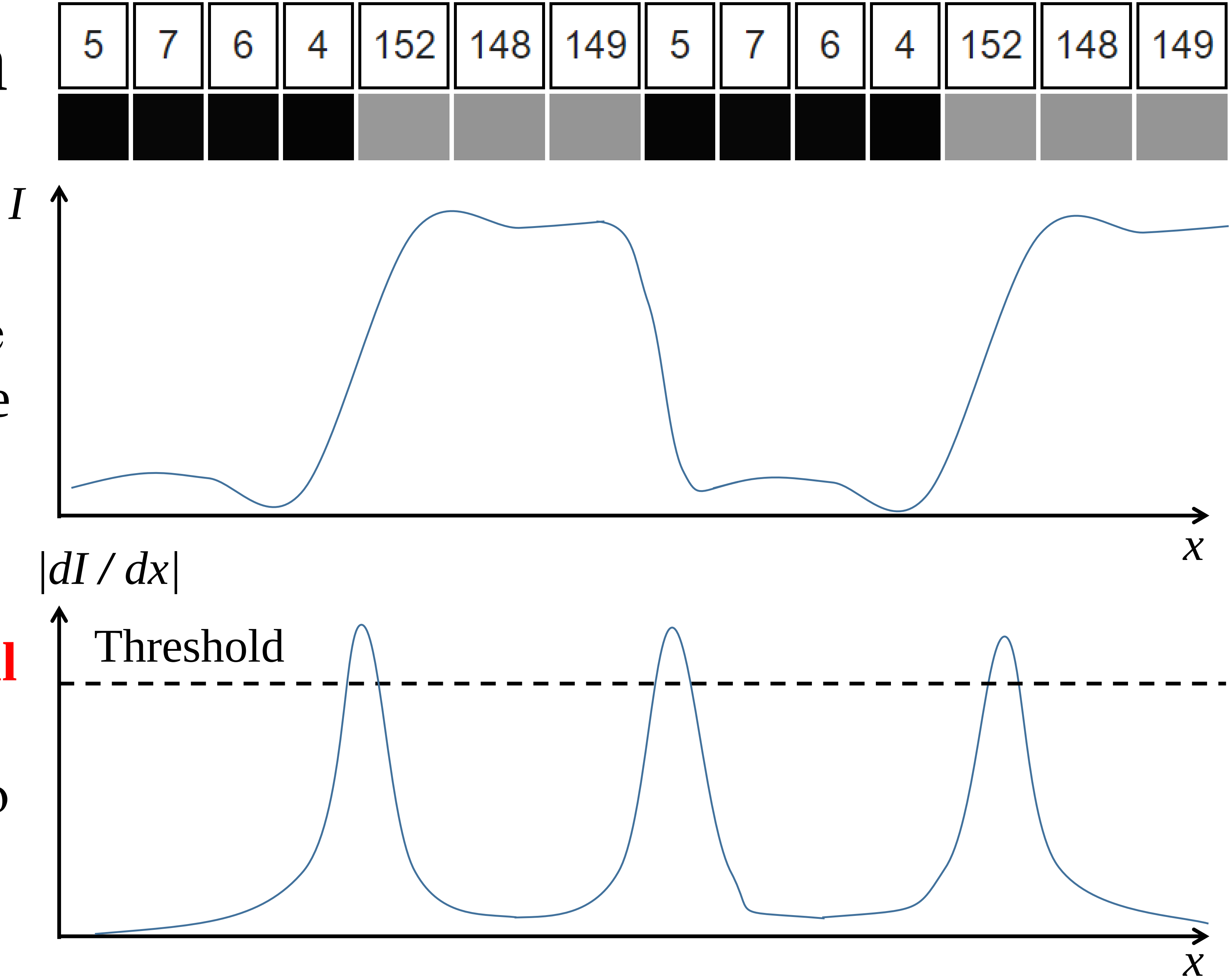
- There are 5 steps in this algorithm:
    1. Eliminate the noise
    2. Get the gradient vector
    3. Non-maximum suppression
    4. Double threshold
    5. Edge tracking by hysteresis

Edge detection

1D-Concept

Suppose the intensity I is a function of x, and it's curve as the $I - x$ figure, then take the derivative of I, we can get the $|dI / dx| - x$ figure.

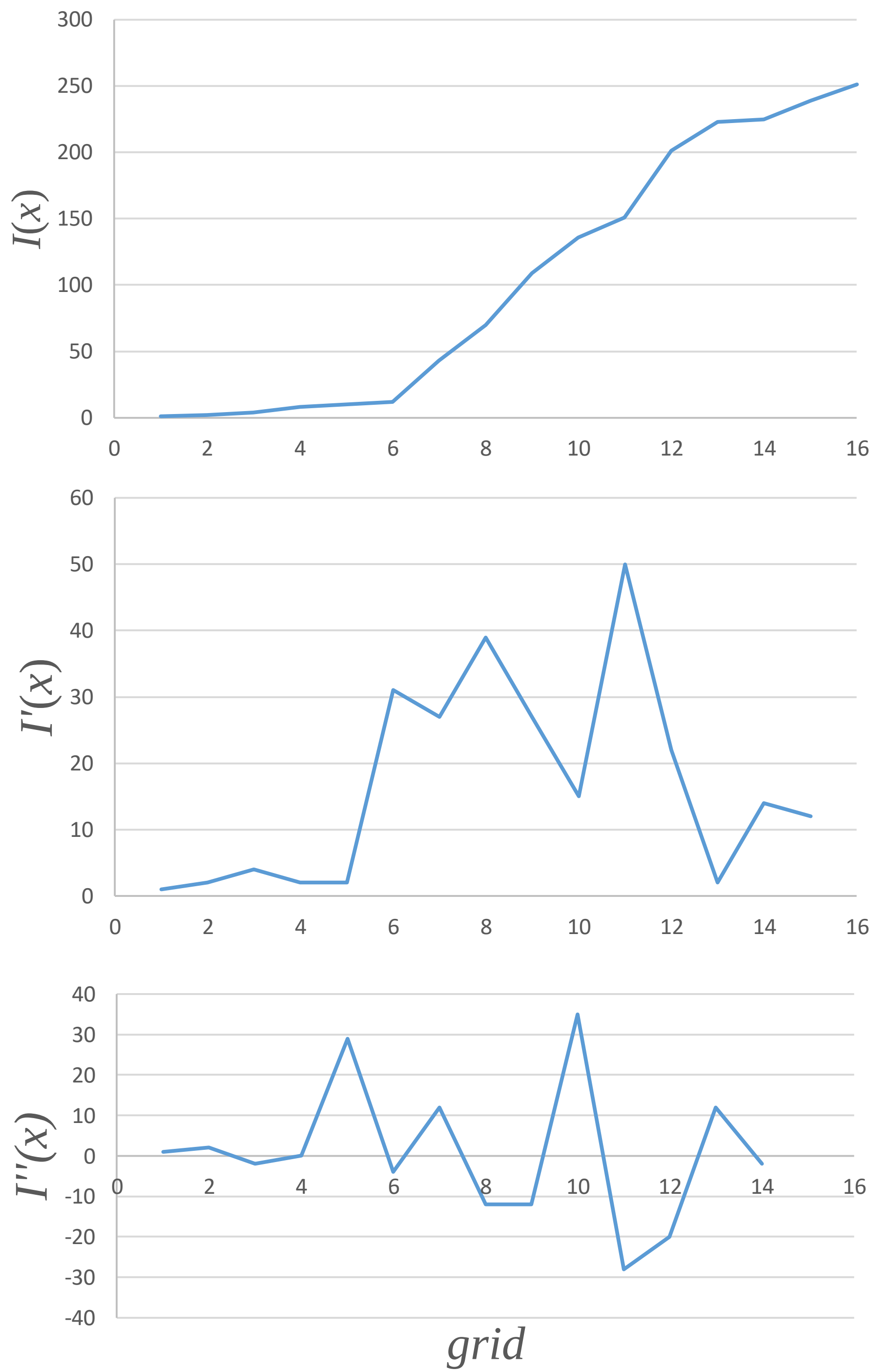
The **absolute value of local extreme value** is the edge, here we'll set a threshold to determine is the edge or not.



Edge detection

1^{st} derivative and 2^{nd} derivative

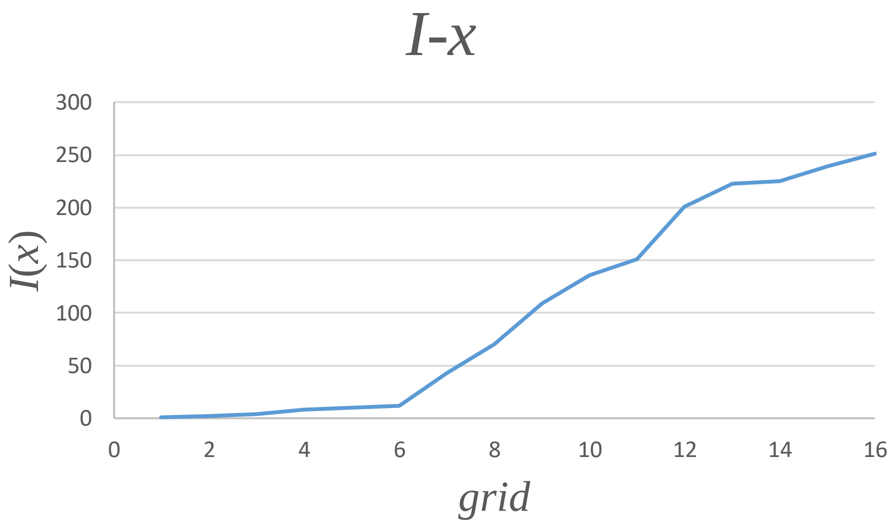
- Assume we have $I(x)$, then take the derivative on $I(x)$, get $dI(x)/dx$ and find its **local extreme value**. It can be done by taking the 2^{nd} derivative on I(x), getting $d^2I(x)/dx^2$ and finding its **0 value**.
- The edge found by 1^{st} derivative is thicker than 2^{nd} derivative, but it can be solve by taking threshold, and the latter is more sensitive to noise than the former.

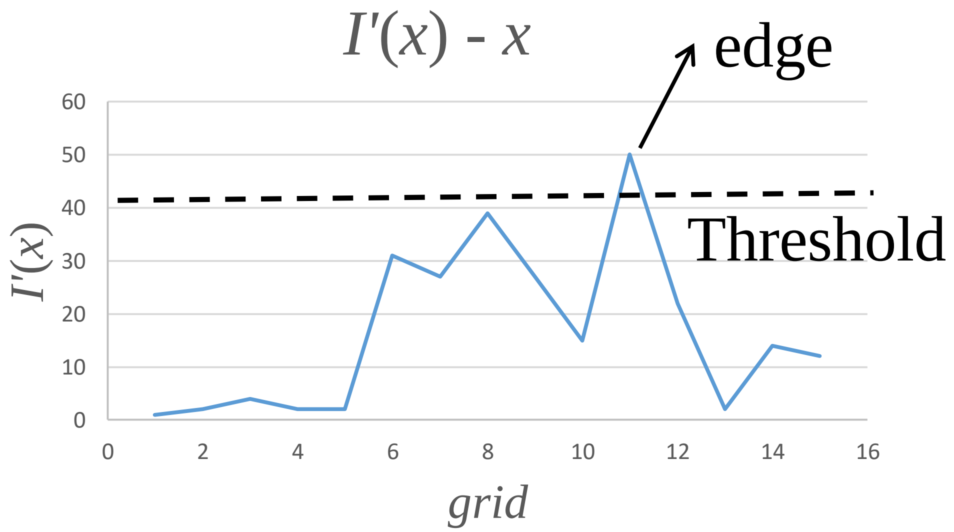


Edge detection

Non-maximum suppression

- If the $I - x$ curve is near a line, then there are many local maximum values, the edge line gets bolder. Here we take the derivative ***dI/dx*** and get its **absolute value** (generally).
- Compare the grids which are value bigger than the threshold with the adjacent grid, so that we can specify the edge, and the edge line get thinner.





Edge detection

2D-Concept

The same as 1D-concept, but there is one more quantity "direction", namely, it's a vector, called "gradient".
In mathematics, the gradient is define as:

$$\vec{\nabla} G(x,y) = \frac{\partial}{\partial x} G(x,y)\hat{i} + \frac{\partial}{\partial y} G(x,y)\hat{j} = G_x \hat{i} + G_y \hat{j}$$

Magnitude: $\left|\vec{\nabla} G(x,y)\right| = \sqrt{G_x{}^2 + G_y{}^2}$ Direction $\theta = \pm \tan^{-1}(G_x / G_y)$

Magnitude is the value of gradient at that point, direction is used to compare the value of gradient at the next point. Since we take the absolute value of magnitude, so there are 2 direction.

Double threshold

- Setting high threshold (HT) and low threshold (LT),
- local maximum values > HT → strong edge
- local maximum values < LT → not edge, and set the value 0
- LT < local maximum values < HT → mark as weak edge
- Where HT and LT are determined by the user

HT = 3.5

LT = 2.5

1	2	1	2	1	1	1	2
1	1	2	1	1	1	1	1
3	2	3	1	3	2	2	3
1	2	1	1	1	1	1	4
2	1	2	1	1	3	3	4
1	2	4	4	4	4	1	1
2	1	3	2	2	1	1	2
1	6	4	1	1	1	3	2
1	6	1	1	1	3	2	1
2	6	2	1	2	1	1	3

→

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
3	0	**3**	0	**3**	0	0	**3**
0	0	0	0	0	0	0	**4**
0	0	0	0	0	**3**	**3**	**4**
0	0	**5**	**5**	**4**	**4**	0	0
0	0	**3**	0	0	0	0	0
0	**6**	**4**	0	0	0	**3**	0
0	**6**	0	0	0	**3**	0	0
0	**6**	0	0	0	0	0	**3**

Edge tracking by hysteresis

- Weak edges that are connected to strong edges will be real edges. Weak edges that are not connected to strong edges will be removed.

Fake edge

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
3	0	3	0	3	0	0	3
0	0	0	0	0	0	0	4
0	0	0	0	0	3	3	4
0	0	5	5	4	4	0	0
0	0	3	0	0	0	0	0
0	6	4	0	0	0	3	0
0	6	0	0	0	3	0	0
0	6	0	0	0	0	0	3

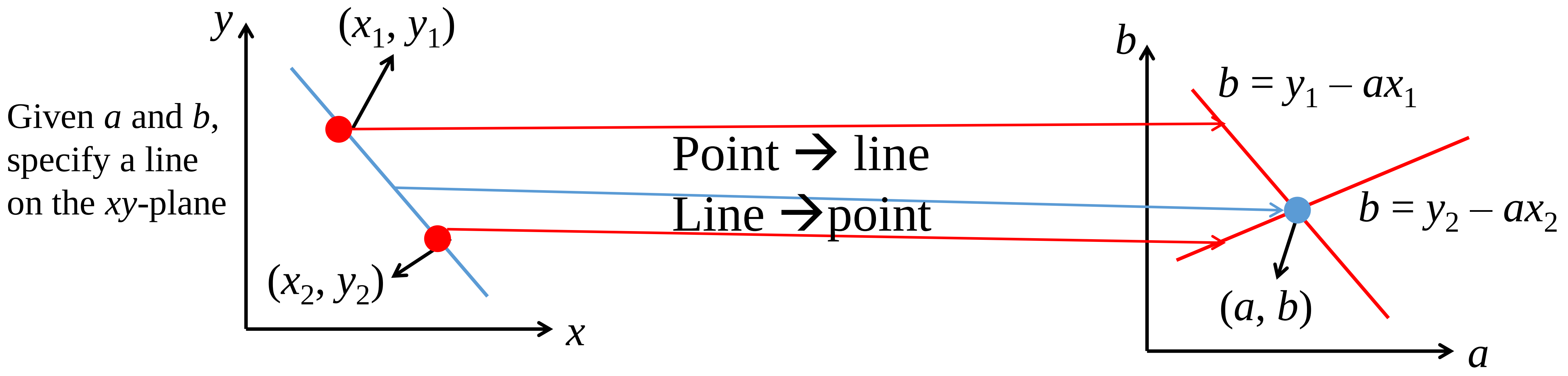
Fake edge

Houghlines

- Given n ($n > 2$) points $(x_1, y_1), \ldots, (x_n, y_n)$, want to find a straight line $y = ax + b$, **a and b are unknown**, so we re-write the line eq. as

$$b = y - ax = y + (-ax)$$

- If 2 straight lines intersect at a point, then the coordinate is the solution of the 2 straight lines. We can use this concept to determine a and b.

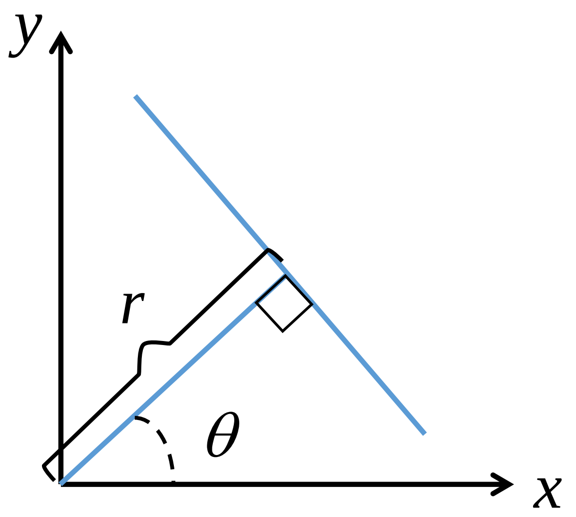


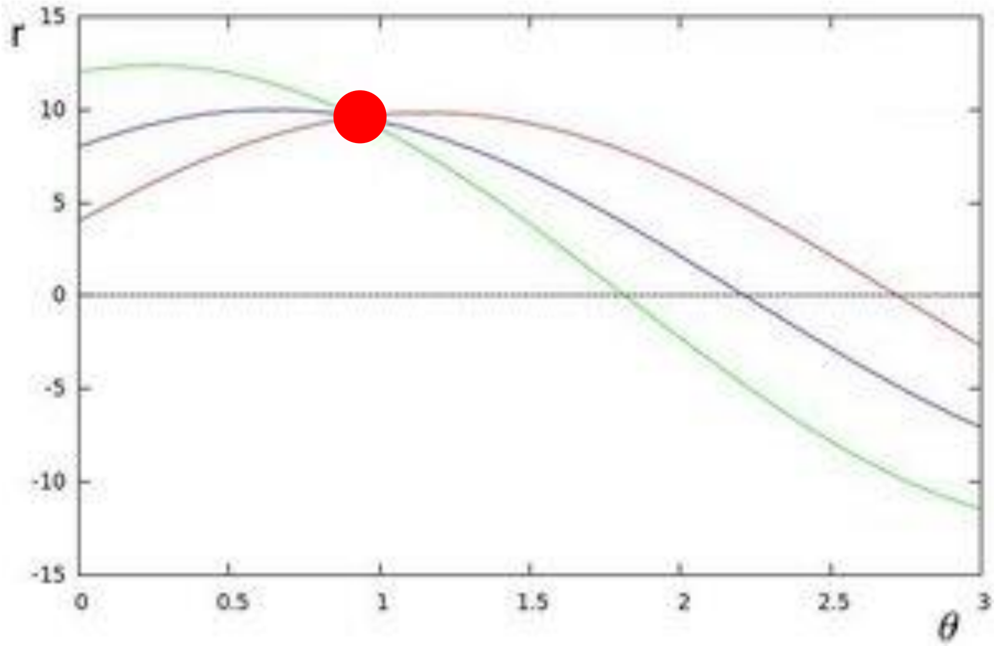
Hough Transform(HT)

- But for a vertical line, *a* is infinity, it's not a good property. So that we transform rectangle coordinate into polar coordinate $y_0 = ax_0 + b$

$$\rightarrow r = x_0\cos\theta + y_0\sin\theta = \sqrt{2}r_0\sin(\theta + 45)$$

$$r_0 = \sqrt{{x_0}^2 + {y_0}^2}$$



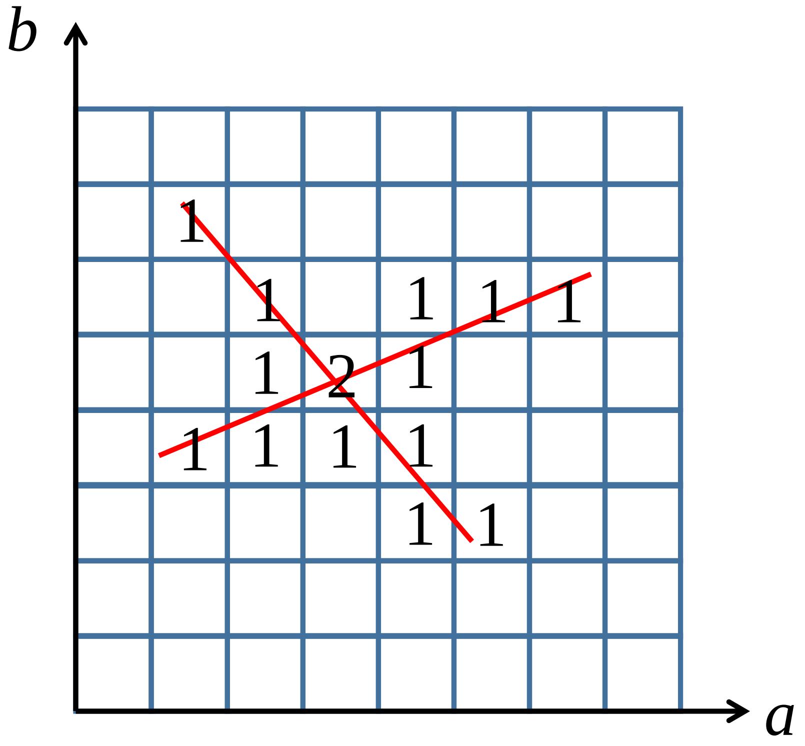


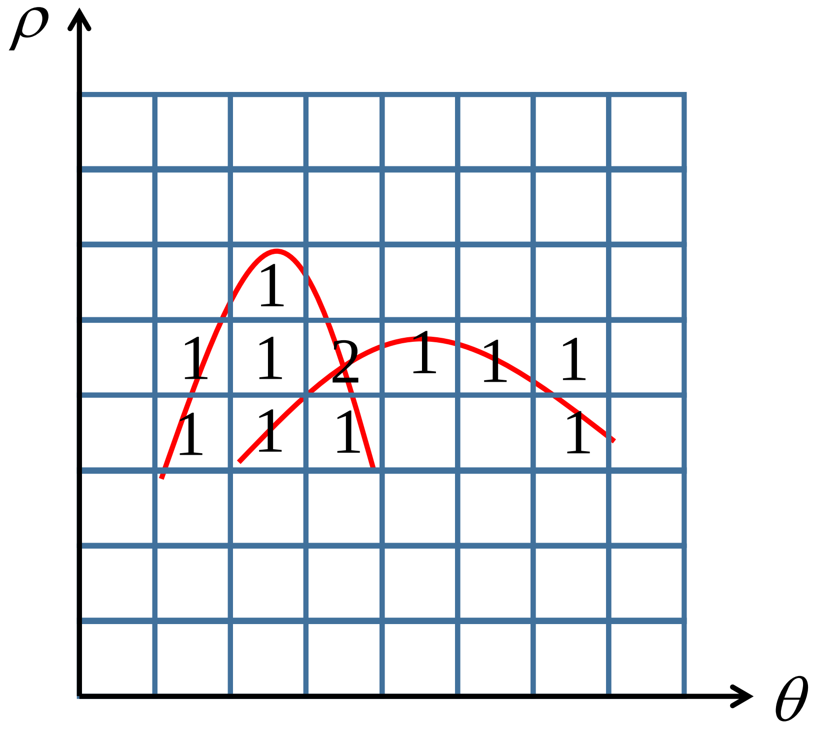
- If the mask is an $D \times D$ square, then r and θ is in the range:

$$r \in [-D/\sqrt{2}, D/\sqrt{2}] \qquad \theta \in [-\pi/2, \pi/2]$$

- By this transformation, we can express the vertical line.

Houghlines

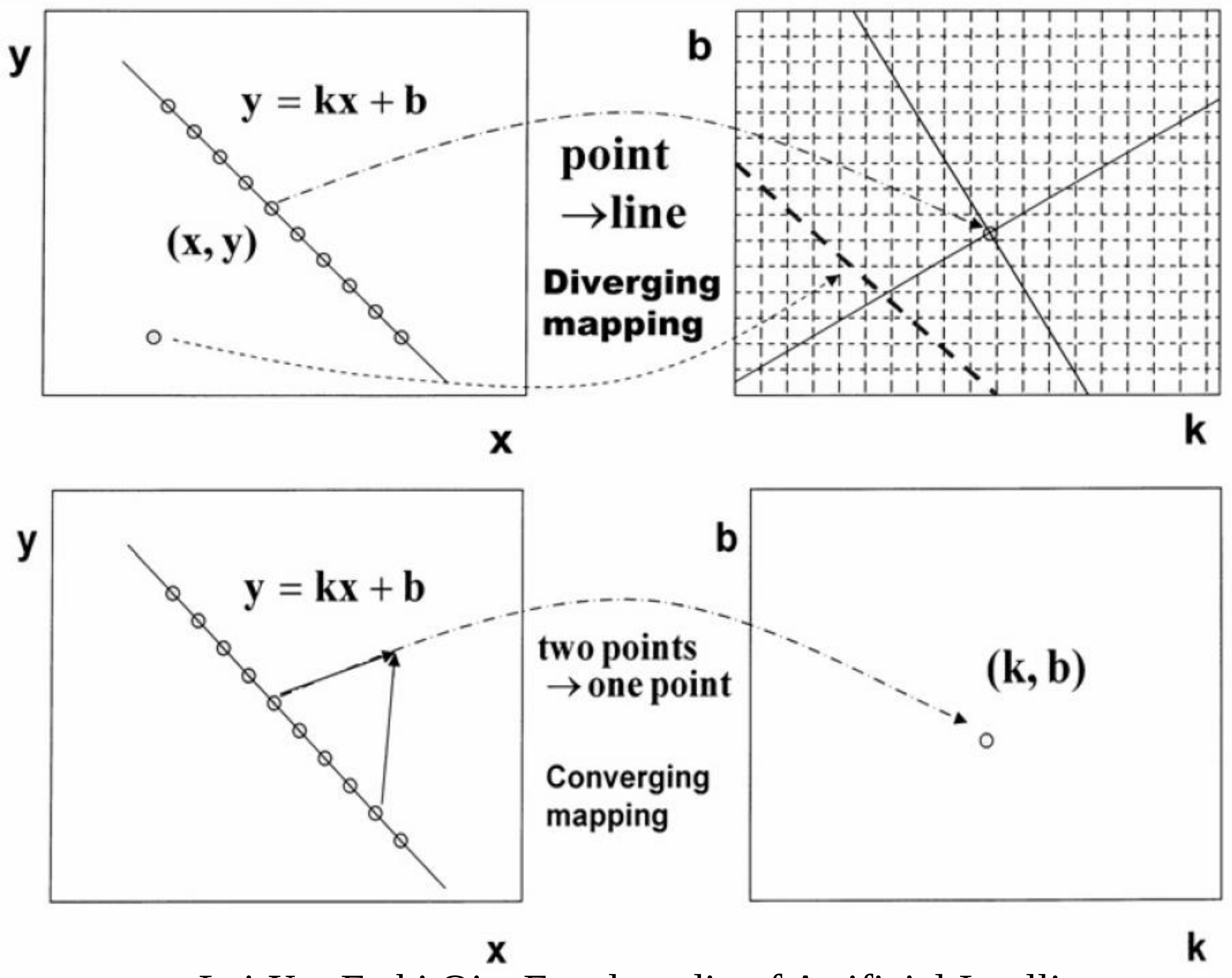




The highest accumulated number determine the coefficient. As the accumulated number is bigger than the threshold, then the line is specific.

Randomized Hough Transform(RHT)

- Most concepts of RHT are the same as HT, but it transforms the points on the same line into one point. This improves the memory usage and computing time, and make the result more accurate.



Lei Xu, Errki Oja, Ecyclopedia of Artificial Intelligence

Randomized Hough Transform(RHT)

- Suppose there are m points on a line, and n points are noise, total points $s = m + n$. If is the Bernoulli trial, then probability of getting 2 points that the two points can form a line in region of interesting (ROI) is

$$P_c = \frac{m(m-1)}{s(s-1)}$$

- If total is R times and at least k times to detect a line successfully, then the successful probability P_s and failure probability P_f are

$$P_s = \mathrm{C}_k^R \, {p_c}^k \left(1 - p_c\right)^{R-k} \qquad P_f = \mathrm{C}_k^R \, {p_c}^k \left(1 - p_c\right)^{R-k}$$

Code (python)