Line detection

- The masks shown below can be used to detect lines at various orientations

−1	−1	−1
2	2	2
−1	−1	−1

Horizontal

−1	−1	2
−1	2	−1
2	−1	−1

+ 45°

−1	2	−1
−1	2	−1
−1	2	−1

Vertical

2	−1	−1
−1	2	−1
−1	−1	2

−45°

mask

-1	-1	-1
2	2	2
-1	-1	-1

*

horizontal line

0	0	0	0
1	1	1	1
0	0	0	0

=

convolved image

-	-	-	-
-	6	6	-
-	-	-	-

vertical line

0	0	1	0
0	0	1	0
0	0	1	0

=

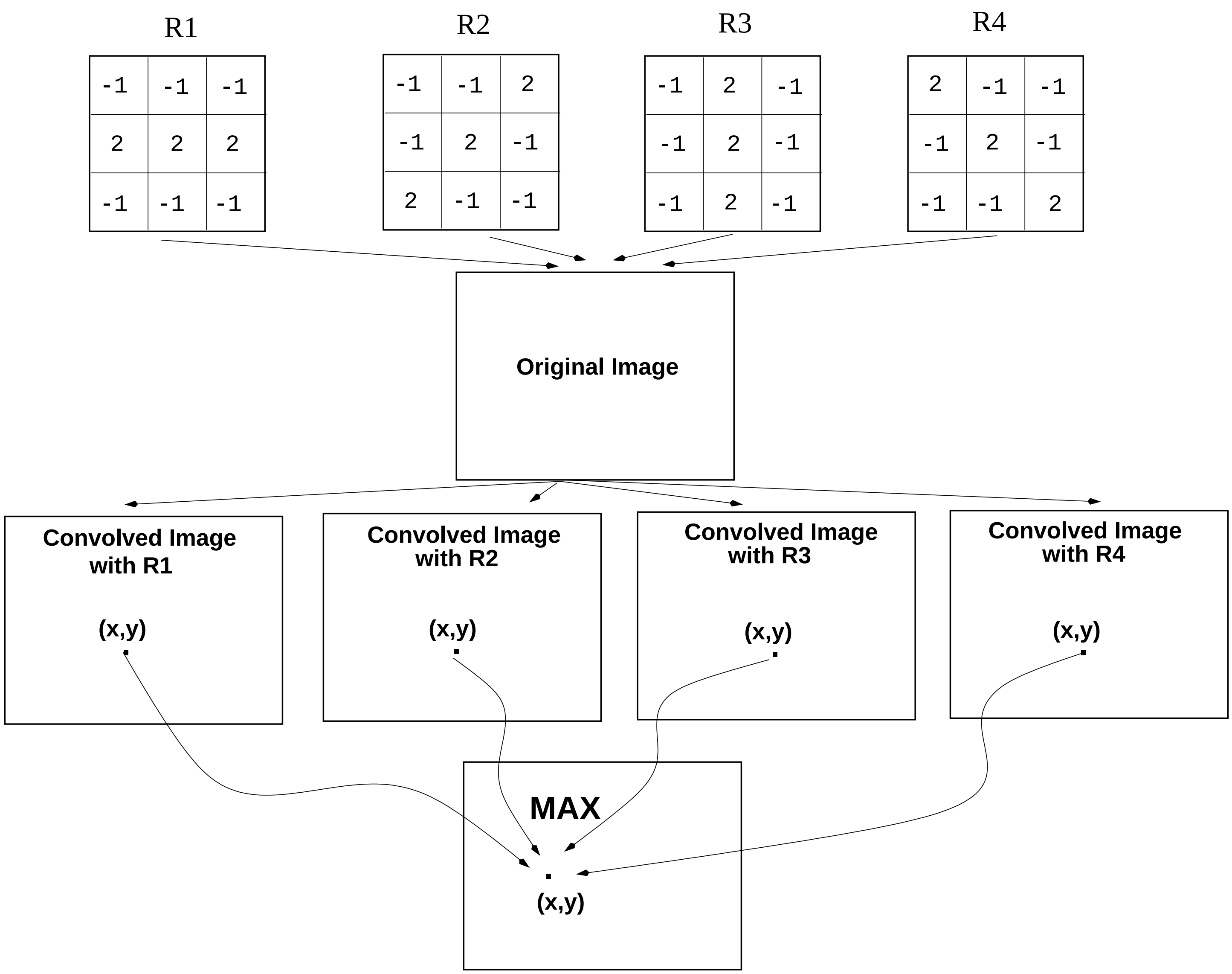
convolved image

-	-	-	-
-	0	0	-
-	-	-	-

- In practice, we run every mask over the image and we combine the responses:

$$R(x, y) = max(|R_1(x, y)|, |R_2(x, y)|, |R_3(x, y)|, |R_4(x, y)|)$$

If $R(x, y) > T$, then discontinuity



Using Hough Transform to detect lines

(Trucco, Chapt. 5)

- Consider the slope-intercept equation of line

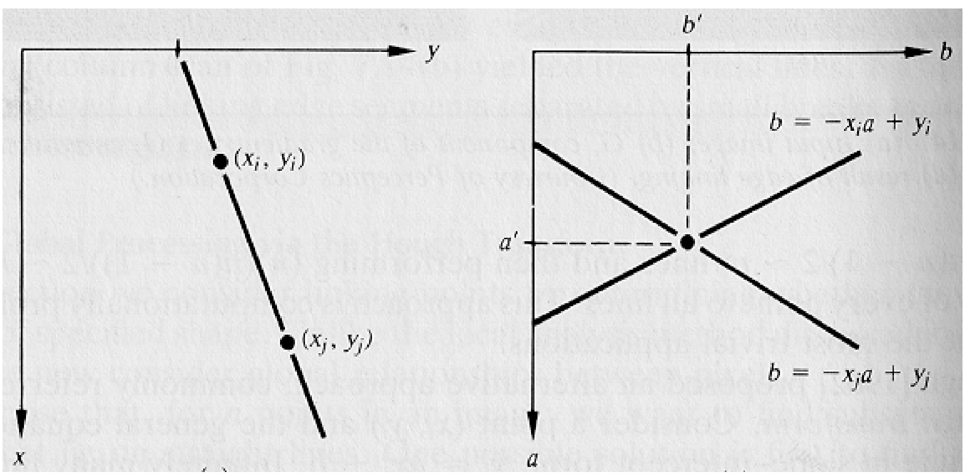
$$y = ax + b,$$

(a, b are constants, x is a variable, y is a function of x)

- Rewrite the equation as follows:

$$b = -xa + y$$

(now, x, y are constants, a is a variable, b is a function of a)



- The following properties are true:

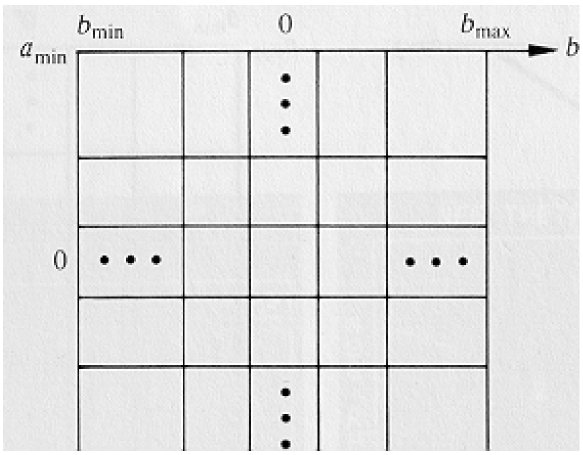
Each point (x_i, y_i) defines a line in the $a - b$ space (parameter space)

Points lying on the same line in the $x - y$ space, define lines in the parameter space which all intersect at the same point

The coordinates of the point of intersection define the parameters of the line in the $x - y$ space

Algorithm

1. Quantize the parameter space
$P[a_{\min}, \ldots, a_{\max}][b_{\min}, \ldots, b_{\max}]$ (accumulator array)



2. For each edge point (x, y)

For($a = a_{\min}$; $a \leq a_{\max}$; a++) {

$b = -xa + y$; /* round off if needed *

$(P[a][b])$++; /* voting */

}

3. Find local maxima in $P[a][b]$

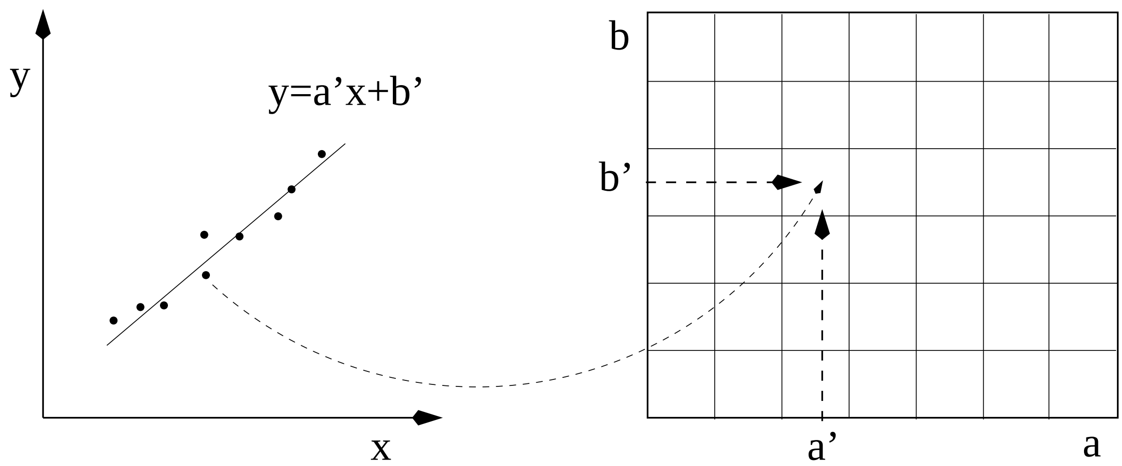
(If $P[a_j][b_k]$=M, then M points lie on the line $y = a_j x + b_k$)

• Effects of quantization

- The parameters of a line can be estimated more accurately using a finer quantization of the parameter space

- Finer quantization increases space and time requirements

- For noise tolerance, however, a coarser quantization is better



(it is very likely that every point will cast a vote in the (a', b') cell)

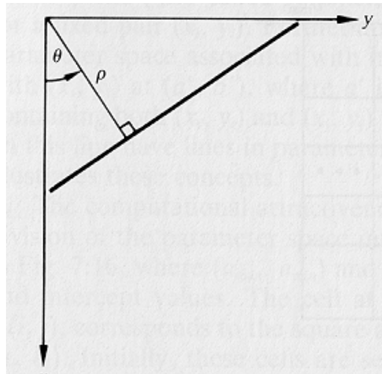
• Problem with slope-intercept equation

- The slope can become very large or even infinity !!

- It will be impossible to quantize such a large space

• Polar representation of lines

$$x\cos\theta + y\sin\theta = \rho \text{ (if the line is vertical, } \theta=0,\ x = \rho\text{)}$$

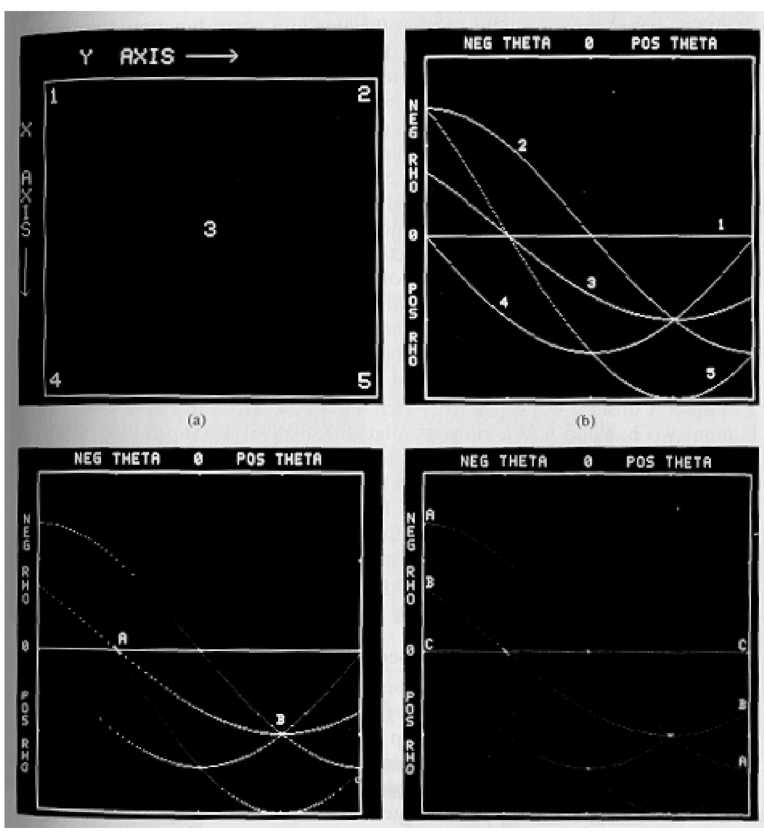


- The following properties are true:

Each point (x_i, y_i) defines a sinusoidal curve in the $\rho - \theta$ space (parameter space)

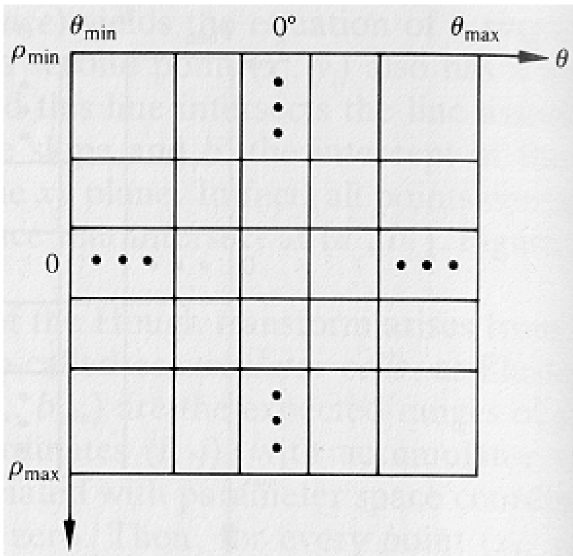
Points lying on the same line in the $x - y$ space, define curves in the parameter space which all intersect at the same point

The coordinates of the point of intersection define the parameters of the line in the $x - y$ space



Algorithm

1. Quantize the parameter space
   $P[\rho_{min}, \ldots, \rho_{max}][\theta_{min}, \ldots, \theta_{max}]$ (accumulator array)



2. For each edge point (x, y)

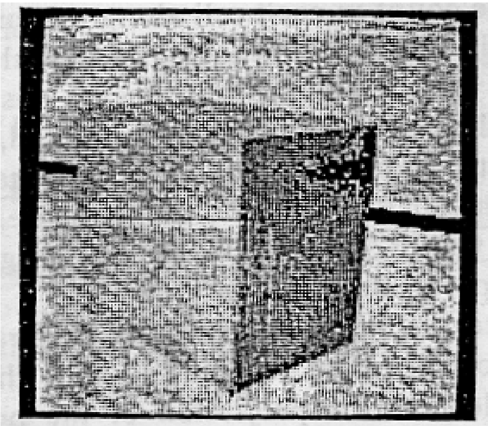
   For($\theta = \theta_{min}$; $\theta \leq \theta_{max}$; θ++) {

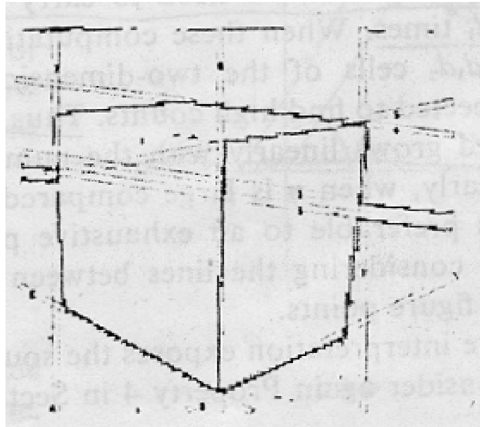
   $\rho = xcos\theta + ysin\theta$; /* round off if needed *

   $(P[\rho][\theta])$++; /* voting */

   }

3. Find local maxima in $P[\rho][\theta]$





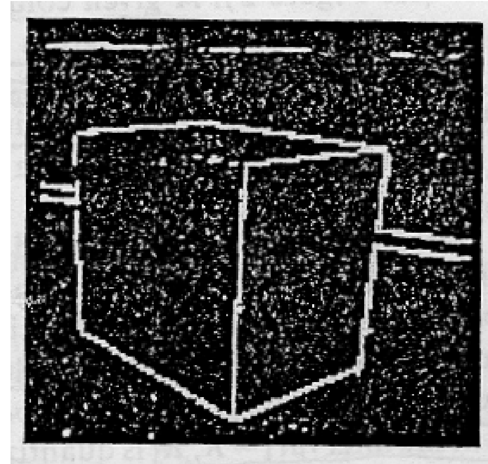


Table I. Accumulator Array for Figure 3(c)

ρ \ θ	0°	20°	40°	60°	80°	100°	120°	140°	160°
85								2	
83							1	2	
81							4	4	
79			2				6	5	
77				2			8	4	2
75							6	6	2
73							4	3	3
71		2		1		1	4	2	3
69			1	4		12	4	3	5
67			3		2	14	2	3	4
65			1			11	1	2	4
63					5	2		2	4
61						1		3	9
59	4	1			11	9	1	8	12
57	4	3		3	10	12	3	10	15
55	9	5		4	5	4	5	11	12
53	6	6		4	10		11	9	14
51	4	9		4	20	2	11	10	8
49	5	6		2	10	3	11	13	8
47	8	4	4	4		2	13	10	10
45	4	7	14	3			11	6	8
43	4	18	21	5		1	12	10	8
41	9	17	21	15		25	18	7	8
39	8	20	21	13		22	11	11	7
37	12	17	22	17		9	10	9	10
35	18	14	17	17	38	8	7	9	6
33	37	16	22	21	42	10	5	9	9
31	35	11	21	23	23	8	11	9	10
29	13	18	18	23	20	14	13	9	9
27	7	16	12	30	20	20	7	9	6
25	7	18	12	32	19	27	8	7	8
23	8	12	11	20	17	52	11	6	7
21	7	17	12	23	8	11	15	11	10
19	9	14	12	16	7	7	14	6	7
17	9	12	12	16	6	9	16	12	7
15	8	13	13	11	7	10	16	14	10
13	10	9	15	11	7	10	16	13	6
11	12	11	13	14	40	10	16	13	13
9	10	10	16	14	8	9	14	21	22
7	10	8	22	12	41	6	7	12	21
5	11	12	15	11	23	6	11	14	14
3	13	15	15	8	18	7	11	16	15
1	10	14	17	11	7	8	9	10	12

ρ \ θ	0°	20°	40°	60°	80°	100°	120°	140°	160°
−85									
−83									
−81									
−79			3	1					
−77			1	3					
−75									
−73									
−71		2							
−69		2							
−67									
−65			1	2					
−63		1			2				
−61	5								
−59	7		2	1		1			
−57	6					2	1		2
−55	0	10	6			1	1		4
−53	16	13	12		18	4	1		6
−51	40	15	11		15	16			6
−49	32	18	11	57	15	23	1	1	5
−47	10	16	11	22	14	16	21	9	5
−45	7	17	11	11	16	18	41	21	6
−43	8	12	14	10	13	17	12	17	6
−41	6	7	14	11	14	14	7	19	12
−39	7	10	9	8	12	8	11	20	23
−37	7	7	14	8	17	9	12	18	24
−35	8	9	17	8	10	7	10	23	23
−33	6	12	15	8	12	9	11	22	26
−31	5	9	19	9	8	11	16	18	15
−29	9	10	12	9	8	9	18	18	15
−27	7	12	10	8	6	9	18	19	19
−25	5	10	8	8	7	7	22	9	14
−23	6	11	9	9	6	11	19	12	9
−21	7	15	9	7	10	10	16	10	11
−19	6	13	8	16	9	11	17	9	10
−17	7	17	9	15	7	11	16	14	13
−15	6	15	10	17	8	13	10	14	9
−13	10	15	9	15	9	17	11	13	12
−11	10	13	10	7	8	17	9	11	15
−9	7	14	8	7	8	23	8	12	15
−7	9	15	12	7	8	21	7	13	12
−5	77	13	15	9	7	14	10	12	15
−3	26	14	14	6	8	12	9	11	18
−1	10	13	18	9	8	8	11	12	15

Extending Hough Transform

- Hough transform can also be used for detecting circles, ellipses, etc.

- For example, the equation of circle is:

$$(x - x_0)^2 + (y - y_0)^2 = r^2$$

- In this case, there are three parameters: (x_0, y_0), r

- In general, we can use hough transform to detect any curve which can be described analytically by an equation of the form:

$$g(v, C) \quad (v\text{: vector of coordinates, } C\text{: parameters})$$

- Detecting arbitrary shapes, with no analytical description, is also possible (**Generalized Hough Transform**)