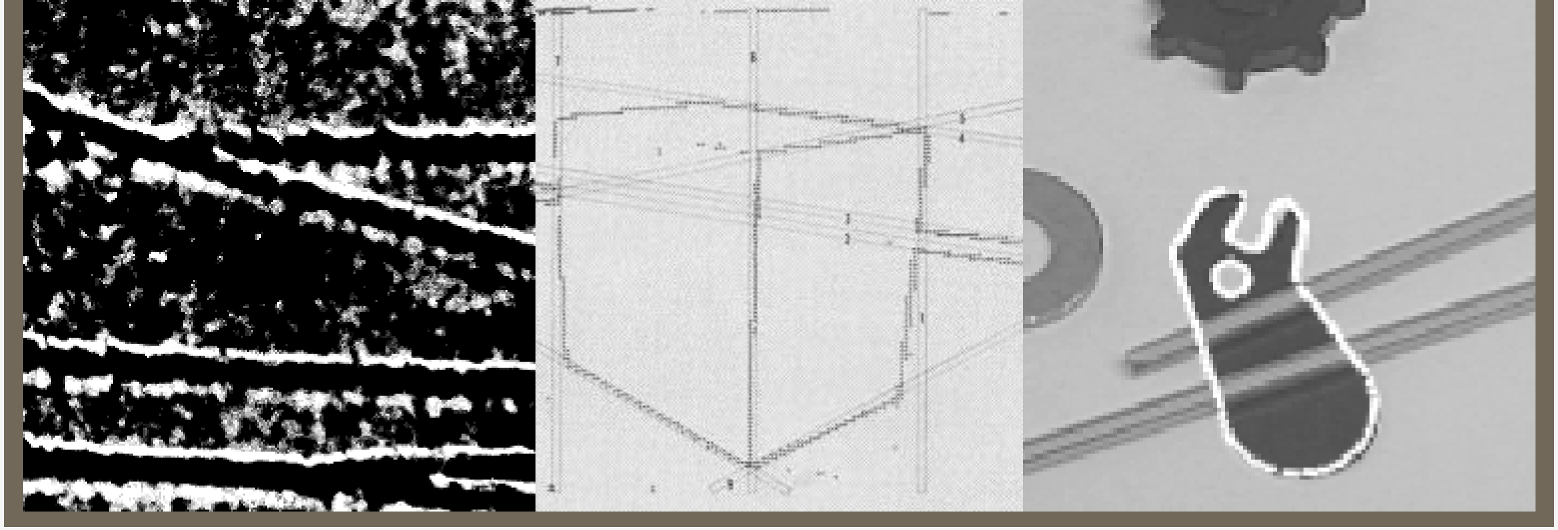


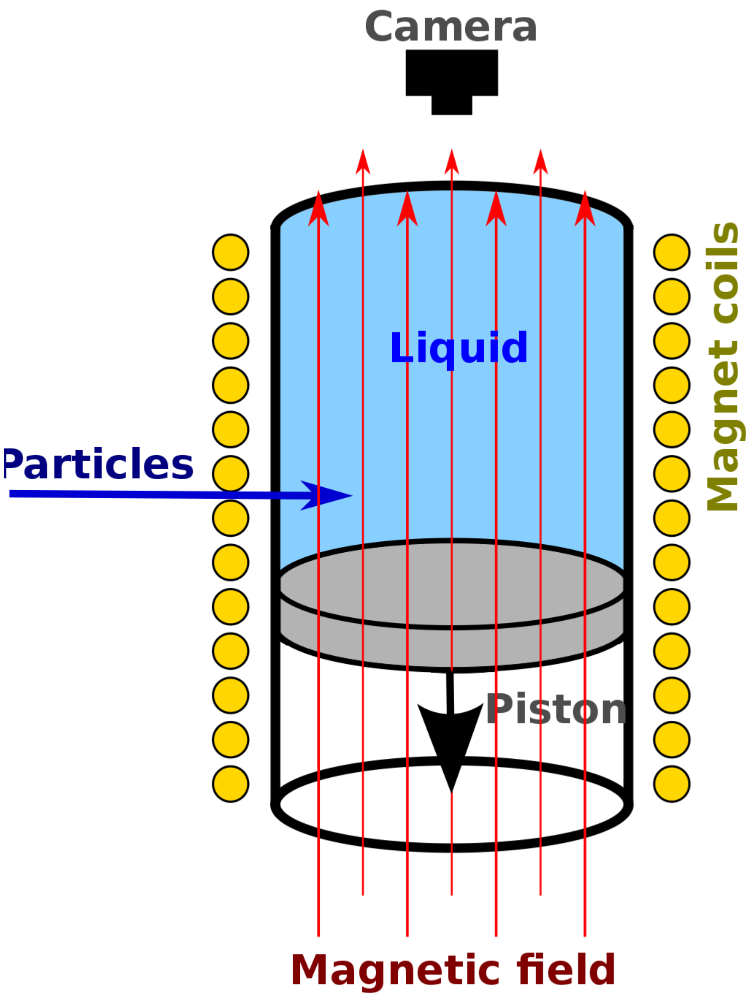
Hough Transform from bubbles through straight lines to arbitrary shapes

ZOI – UTIA

Adam Novozámský (novozamsky@utia.cas.cz)

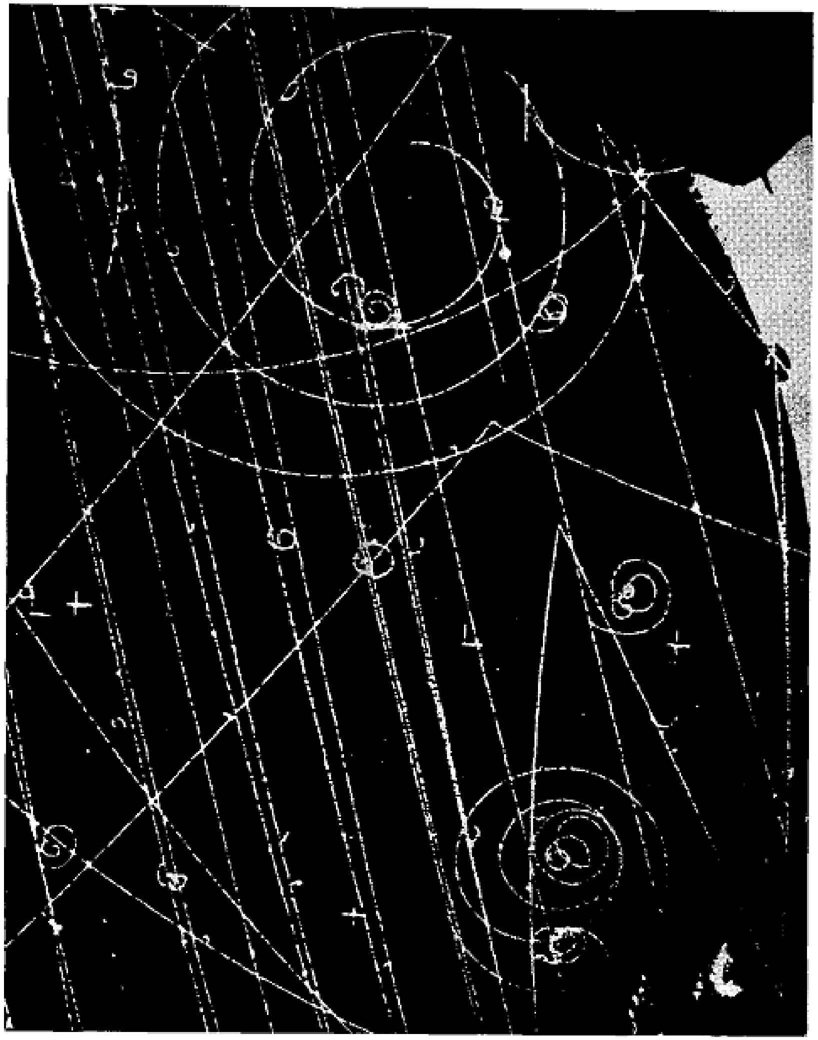
Beginning

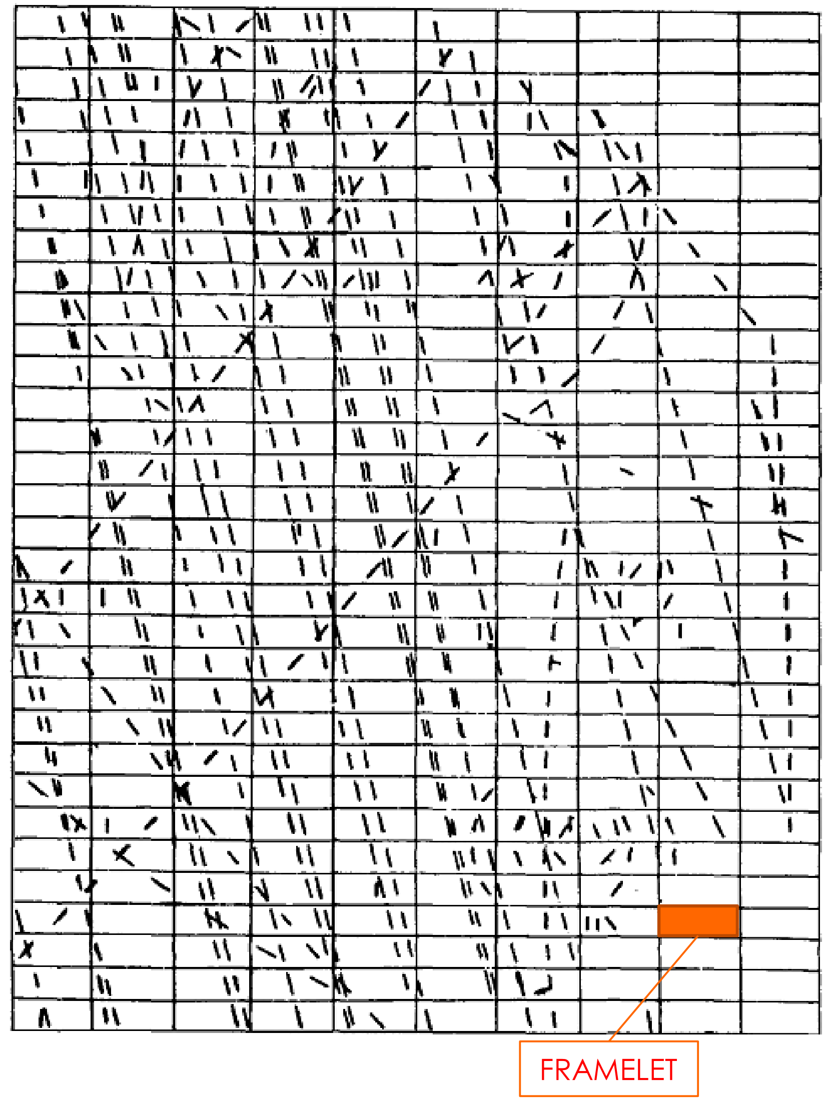
P.V.C. Hough, Machine Analysis of Bubble Chamber Pictures, Proc. Int. Conf. High Energy Accelerators and Instrumentation, 1959.



Beginning

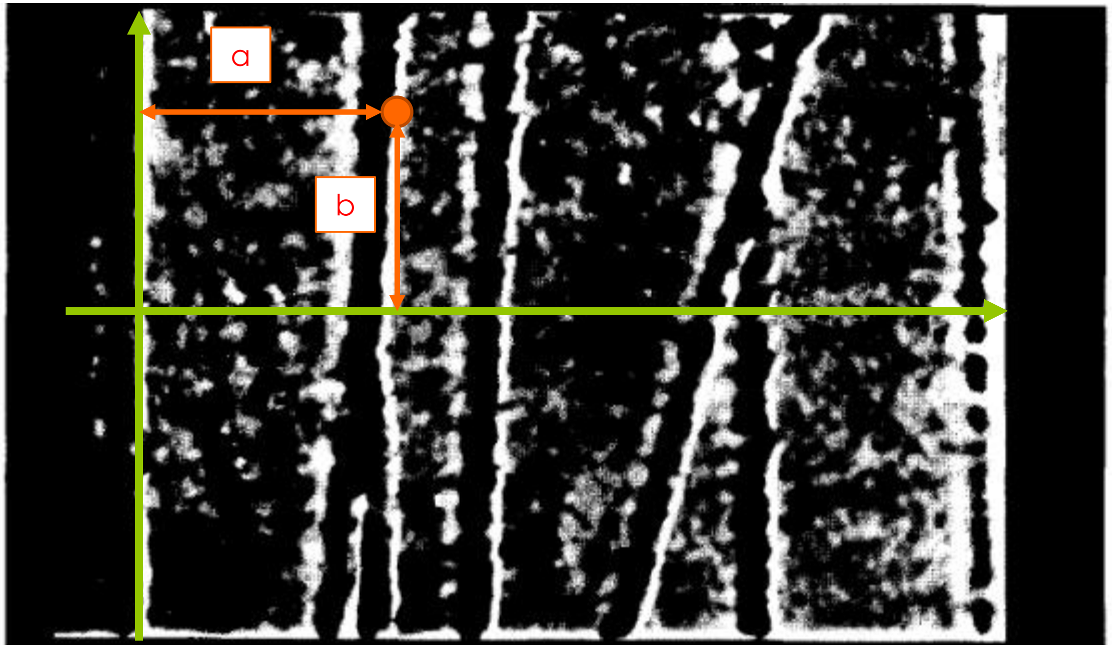
P.V.C. Hough, Machine Analysis of Bubble Chamber Pictures, Proc. Int. Conf. High Energy Accelerators and Instrumentation, 1959

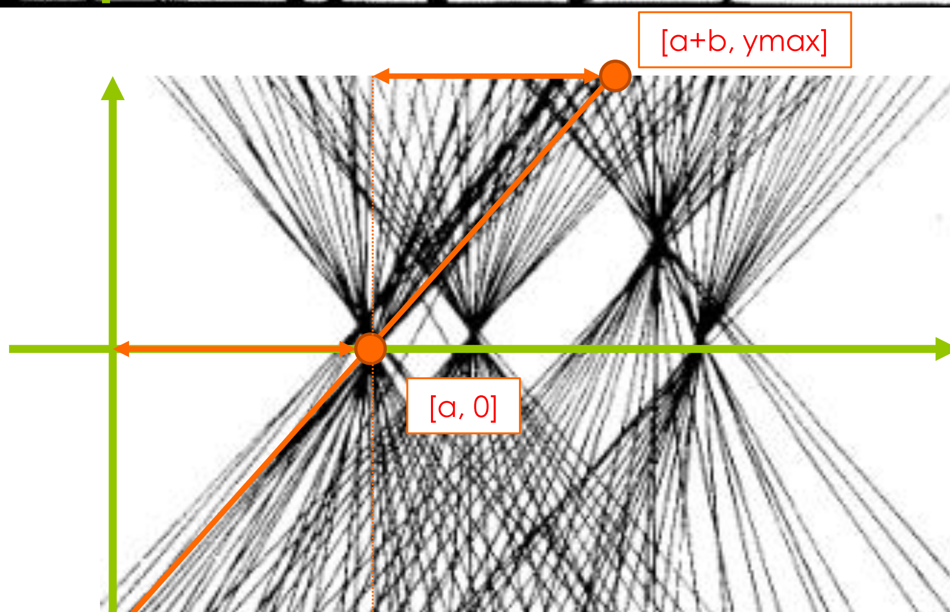




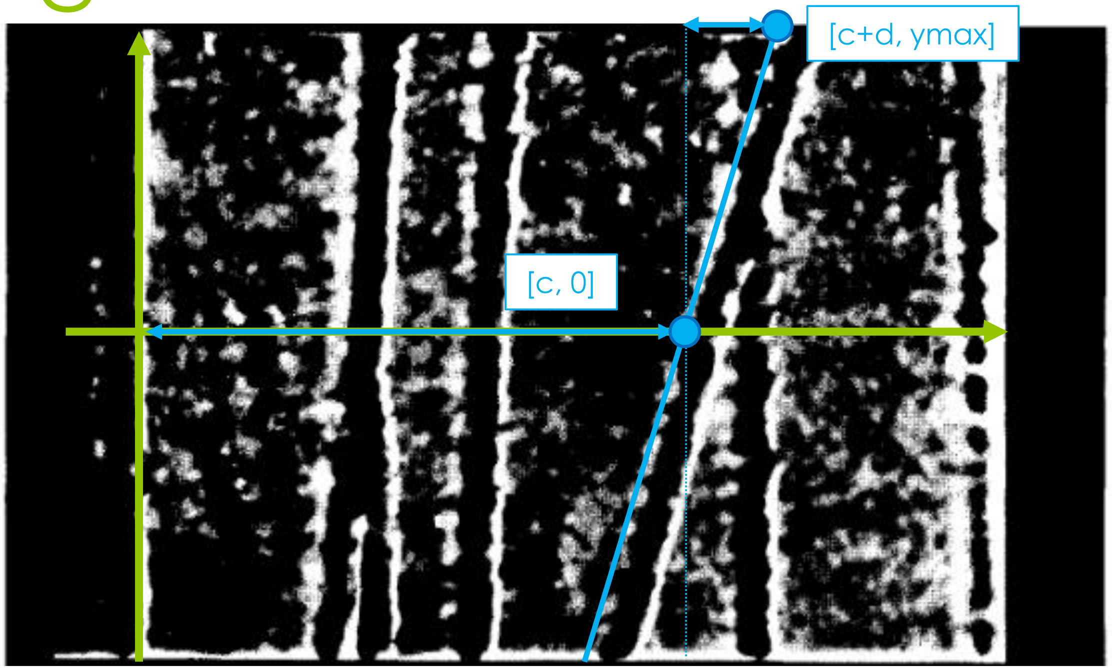
Beginning – The Plane Transform

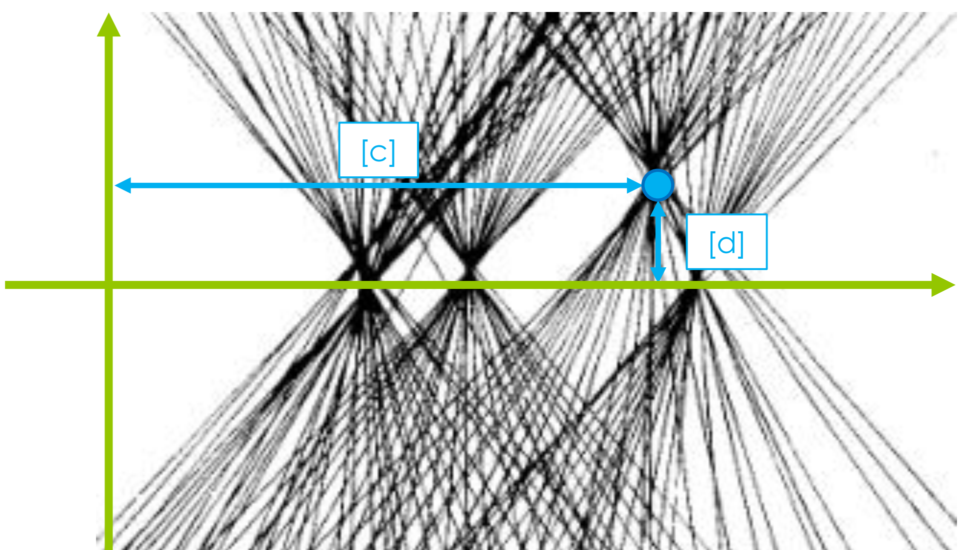
slope-intercept HT



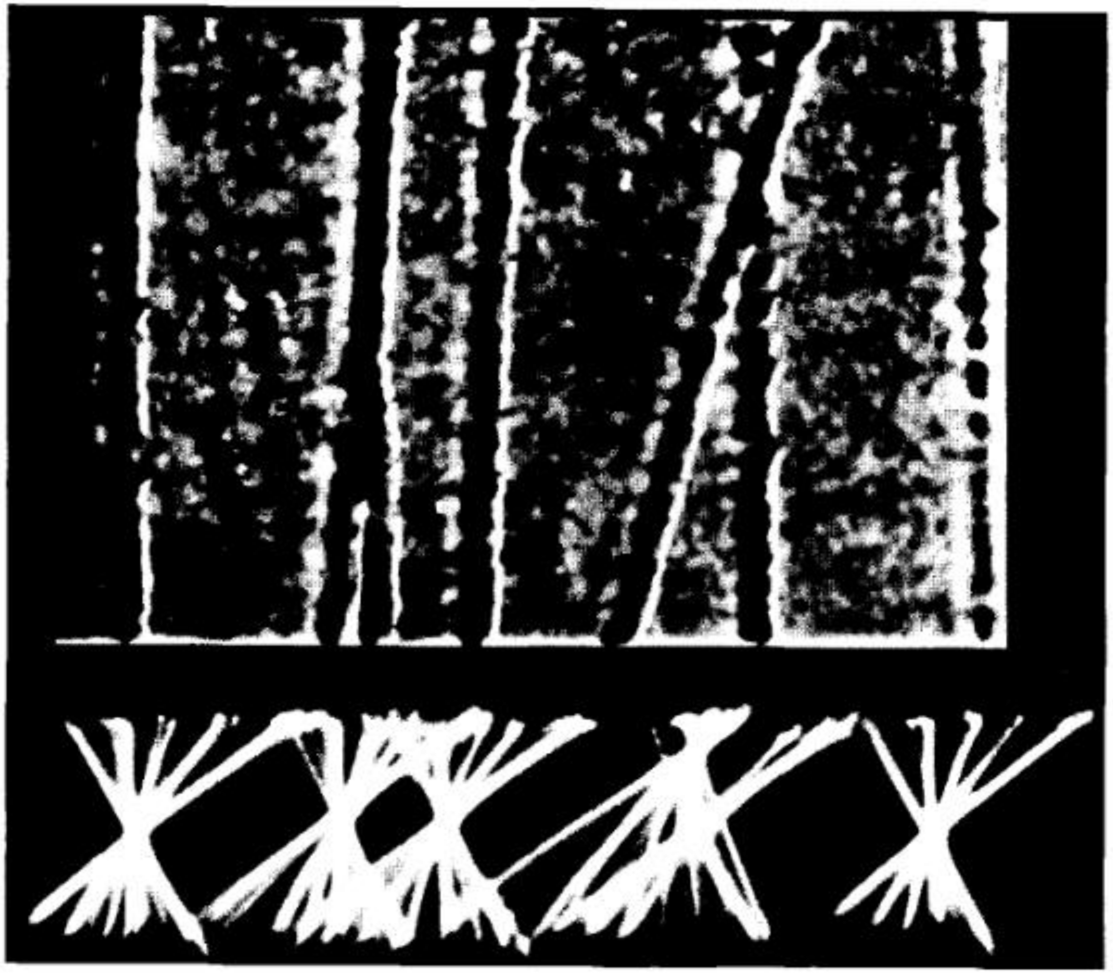


Beginning – The Plane Transform

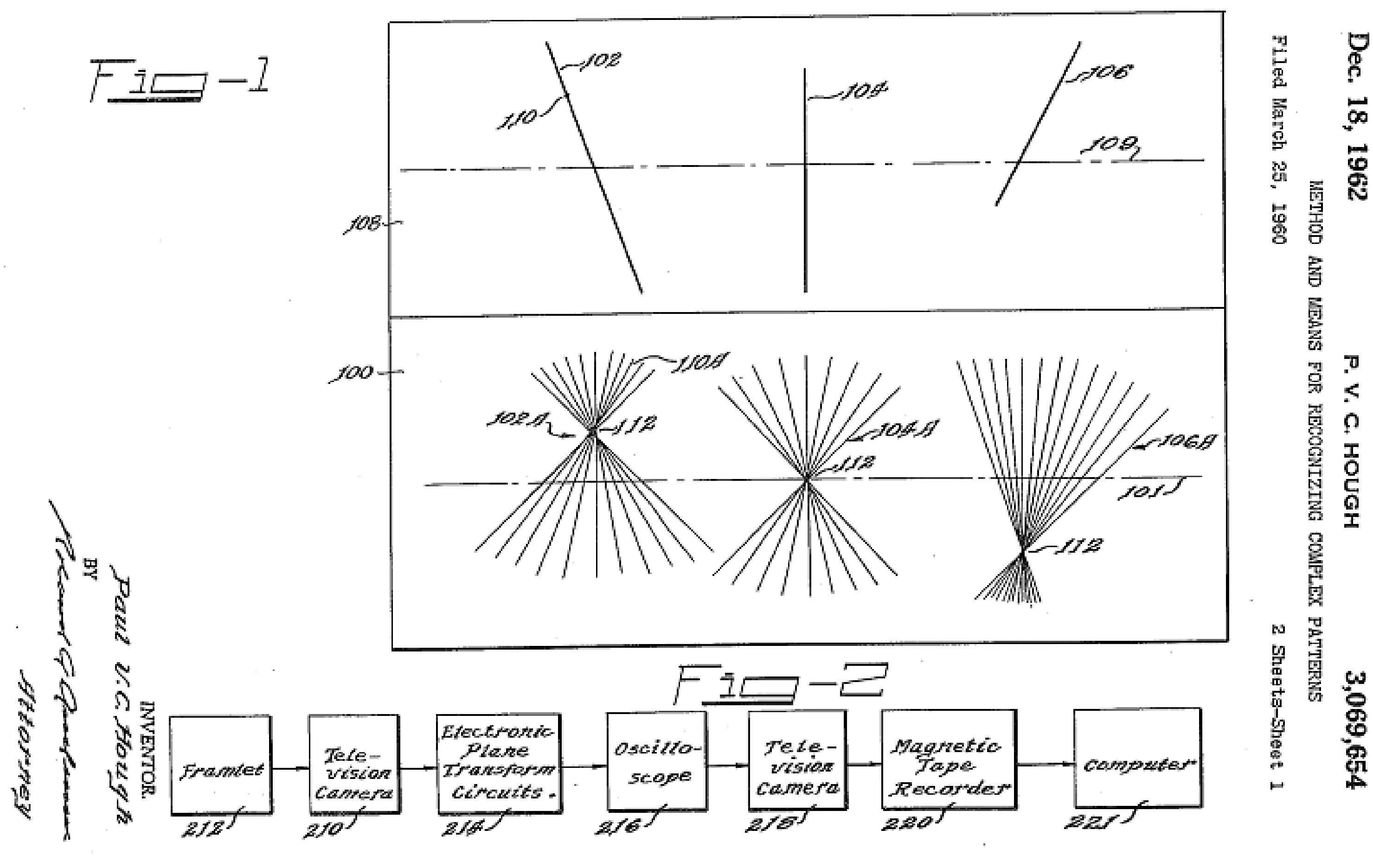




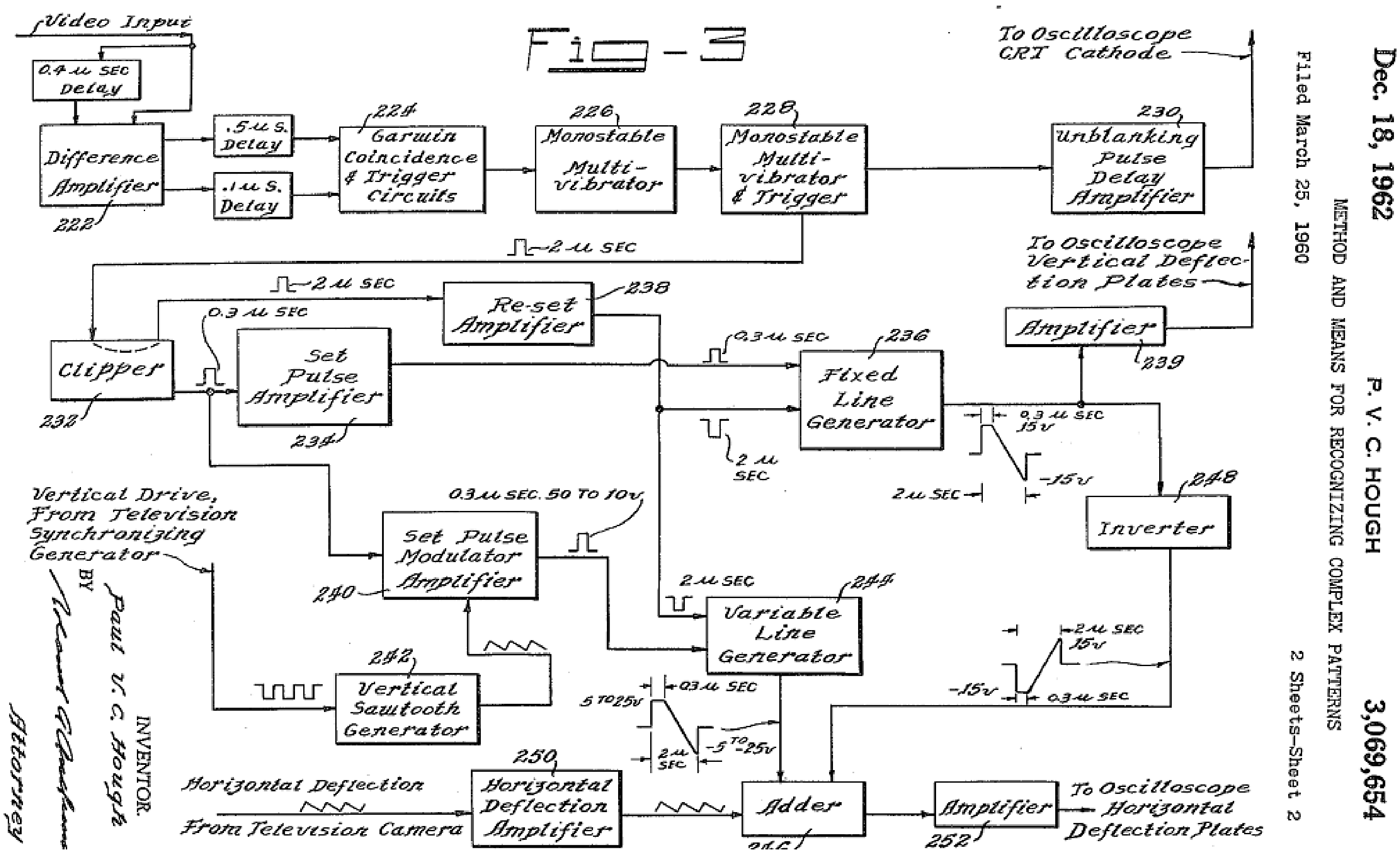
Beginning – The Plane Transform



Beginning – The Plane Transform



Beginning – The Plane Transform



Azriel Rosenfeld 1969

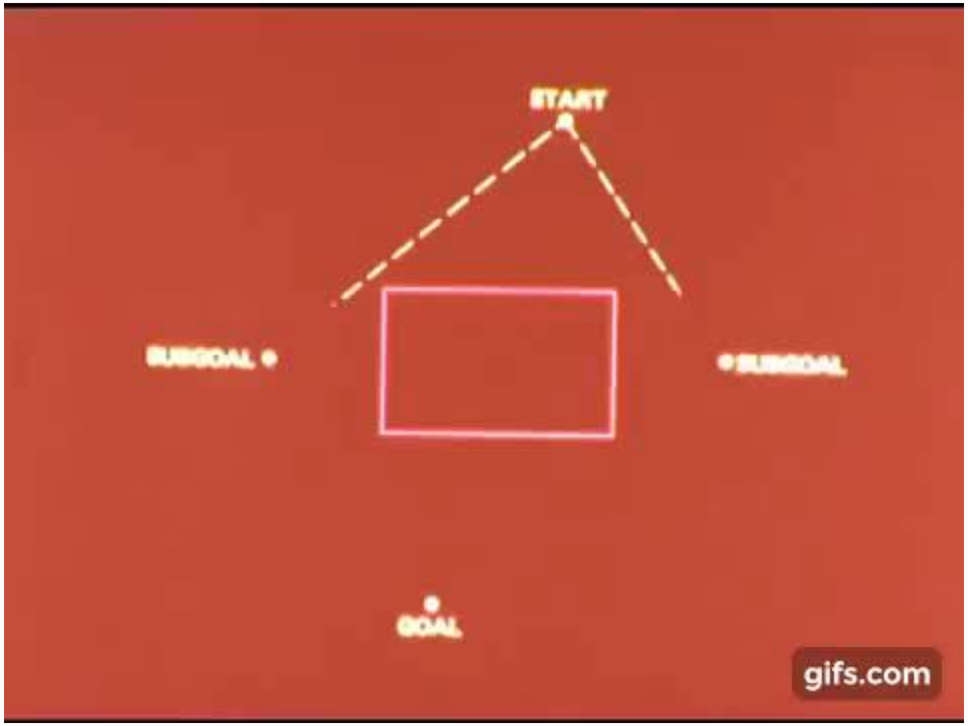
- gave the first explicit algebraic form for the transform:

$$y = y_i x + x_i$$

- proposed a simple digital implementation of the transform space as an array of counters
- introduced to the computer science community this idea

Duda & Hart

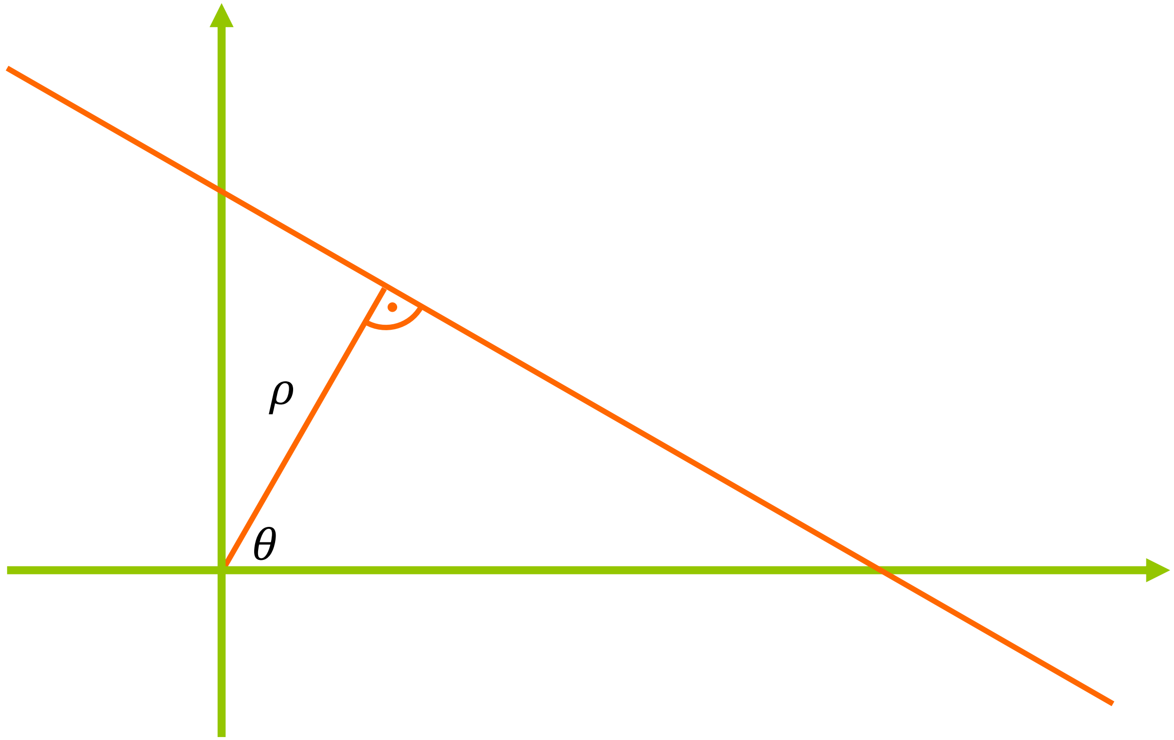
- Shakey the robot [1966-1972]
  - SRI nonprofit scientific research institute
  - Artificial Intelligence Center



Duda & Hart

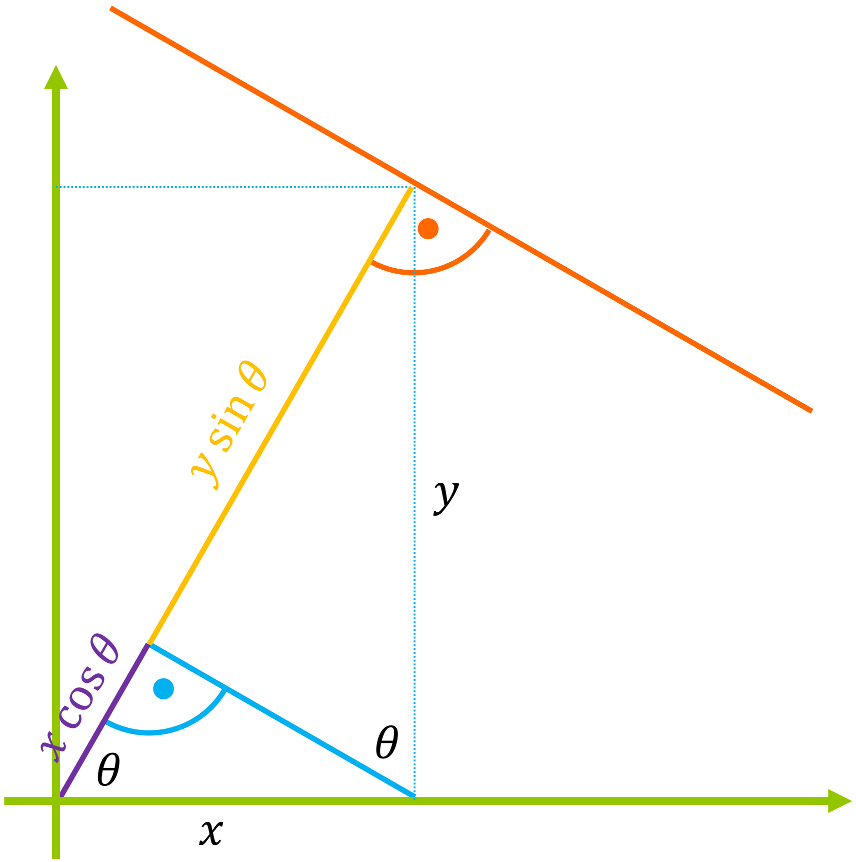
Richard O. Duda and Peter E. Hart, Use of the Hough transformation to detect lines and curves in pictures, Commun. ACM 15, 1, 1972.

slope-intercept ⟶ angle-radius



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

Duda & Hart



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

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

$$1 = \sin^2\theta + \cos^2\theta$$

$$1 = \frac{y\sin\theta}{\rho} + \frac{x\cos\theta}{\rho}$$

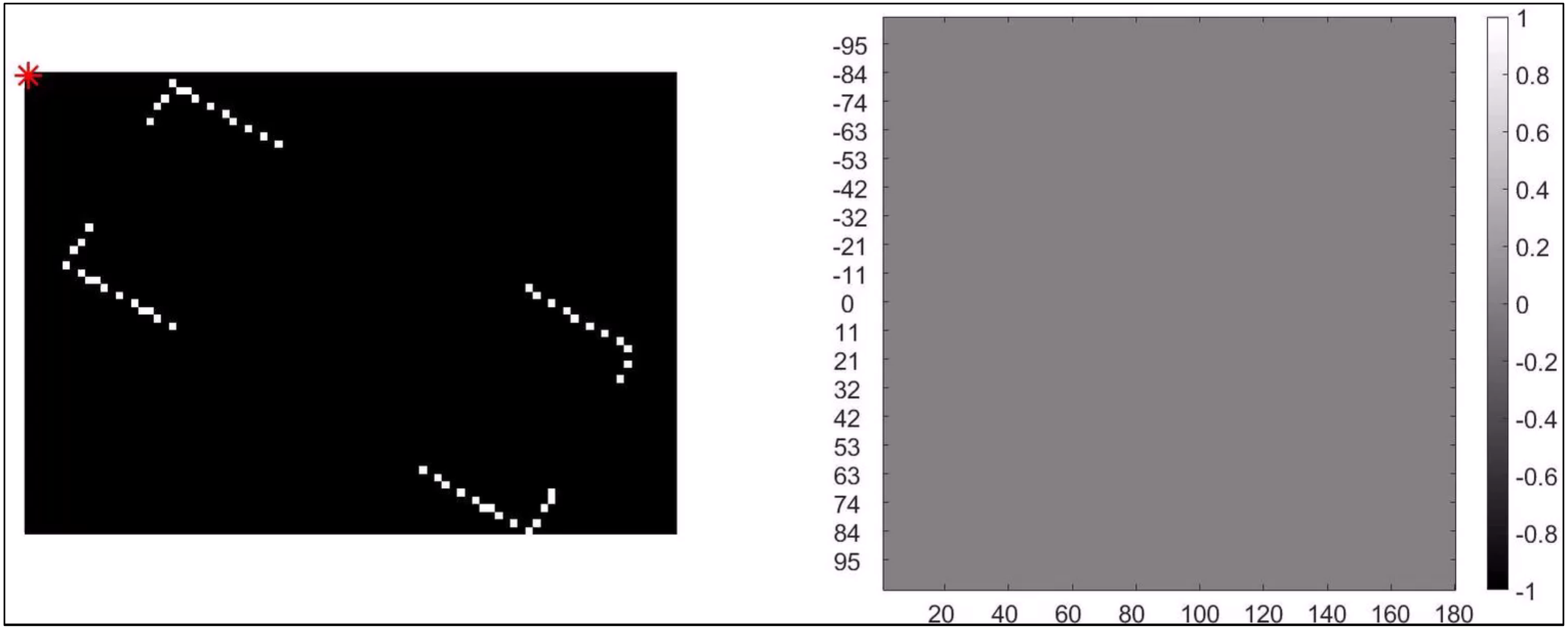
$$\rho = x\cos\theta + y\sin\theta$$

Duda & Hart

- **Property 1:** A point in the picture plane corresponds to a sinusoidal curve in the parameter plane.
- **Property 2:** A point in the parameter plane corresponds to a straight line in the picture plane.
- **Property 3:** Points lying on the same straight line in the picture plane correspond to curves through a common point in the parameter plane.
- **Property 4:** Points lying on the same curve in the parameter plane correspond to lines through the same point in the picture plane.

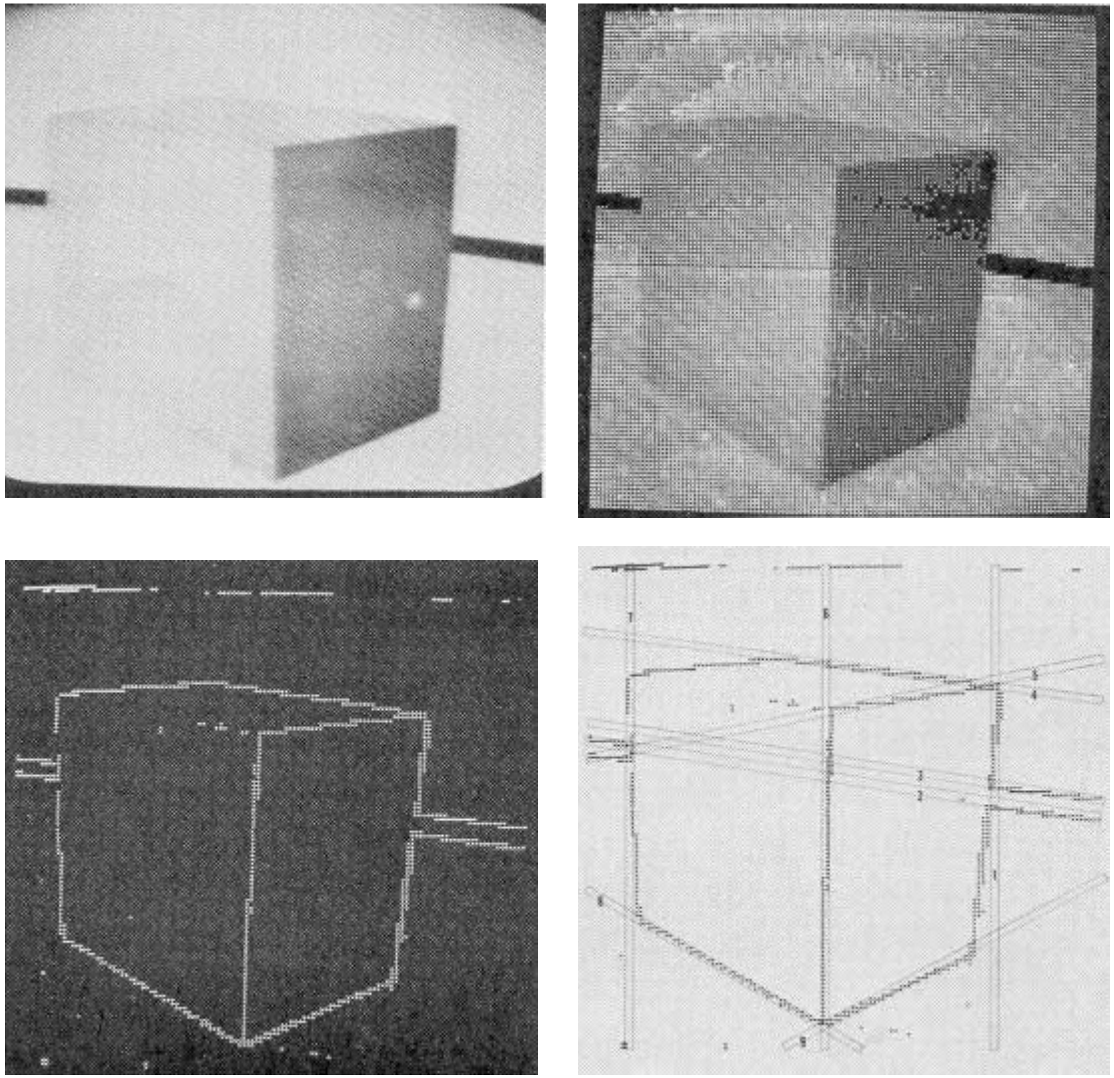
Duda & Hart

demonstration of the algorithm



Duda & Hart

An illustrative example



Duda & Hart

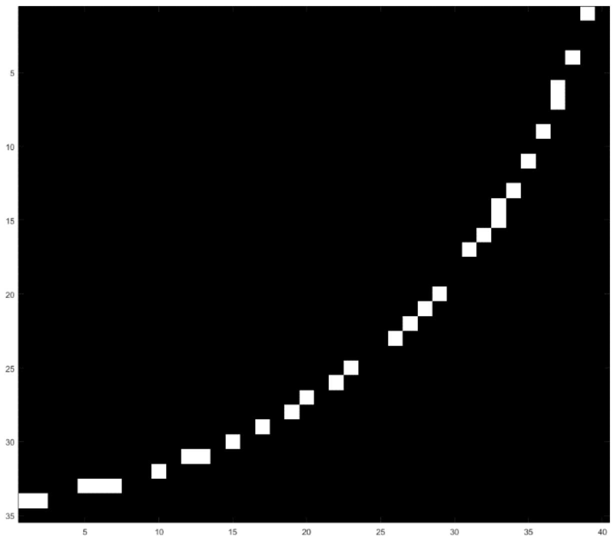
An illustrative example

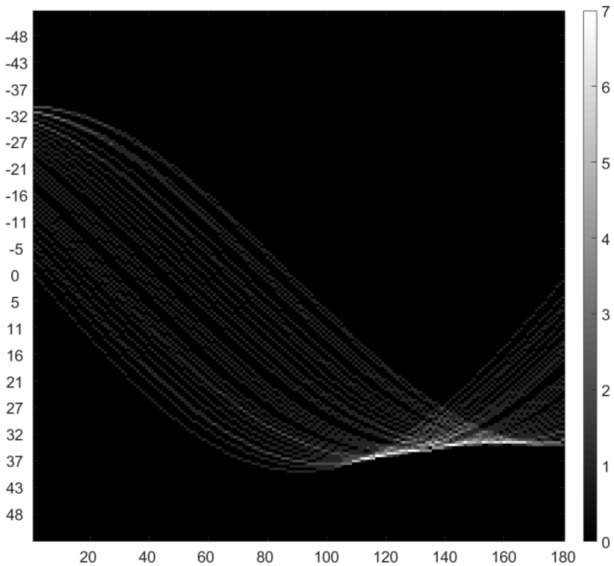
ρ \ θ	0°	20°	40°	60°	80°	100°	120°	140°	160°
85								2	
83							1	2	
81							4	4	
79			2				6	5	
77				2			8	4	2
75							6	6	2
73							4	3	3
71		2		1		1	4	2	3
69			1	4		12	4	3	5
67			3		2	14	2	3	4
65			1			11	1	2	4
63					5	2		2	4
61						1		3	9
59	4	1			11	9	1	8	12
57	4	3		3	10	12	3	10	15
55	9	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	(38)	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

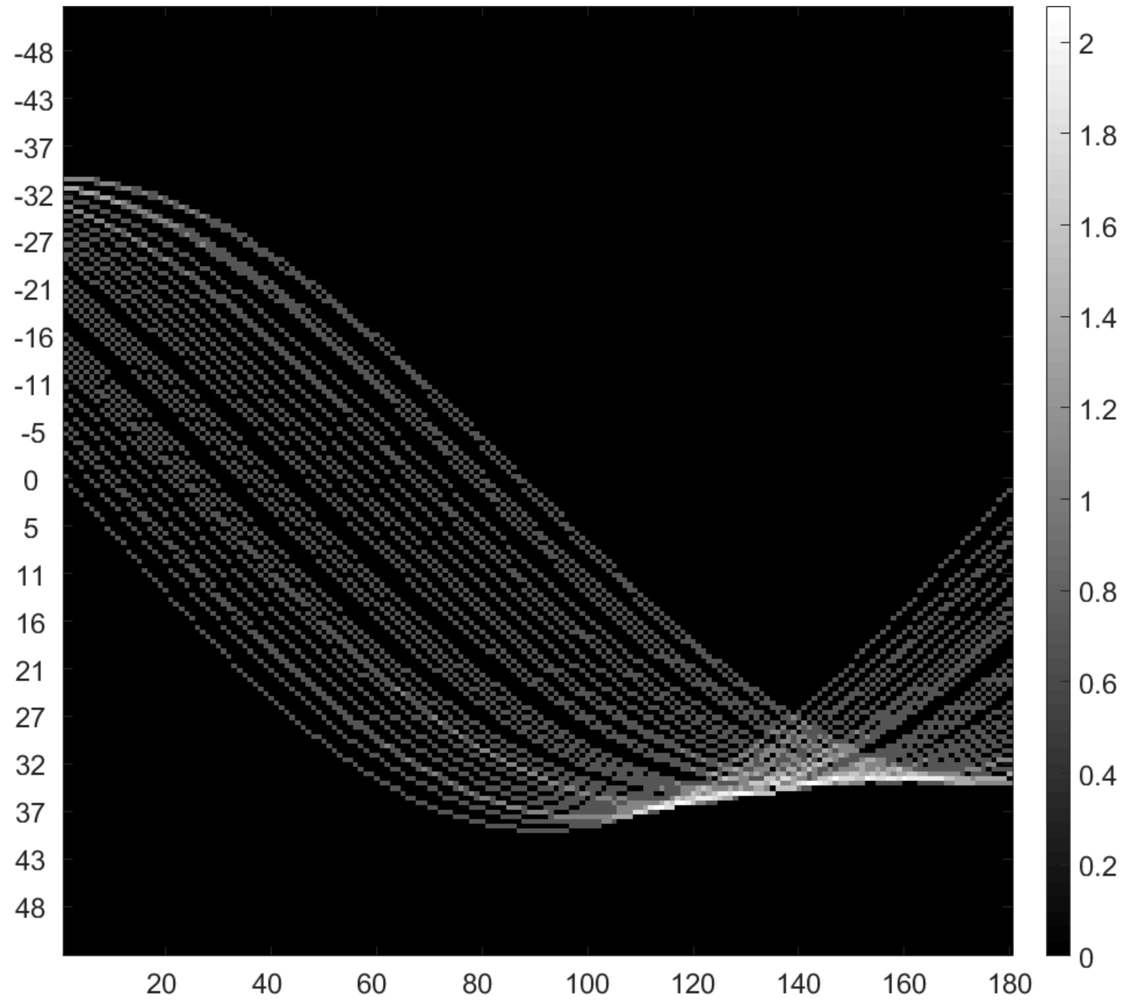
ρ \ θ	0°	20°	40°	60°	80°	100°	120°	140°	160°
−85									
−83									
−81									
−79			3	1					
−77			1	3					
−75									
−73									
−71		2							
−69		2							
−67									
−65			1	2					
−63		1			2				
−61	5								
−59	7		2	1		1			
−57	6					2	1		2
−55	0	10	6			1	1		4
−53	16	13	12		18	4	1		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

Duda & Hart

- Rotation

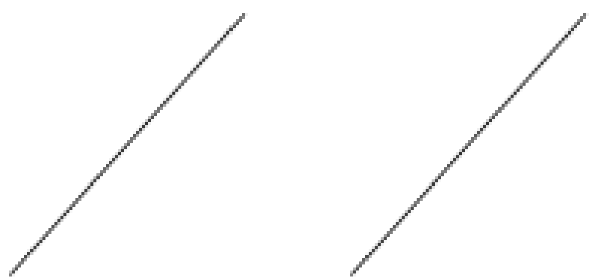


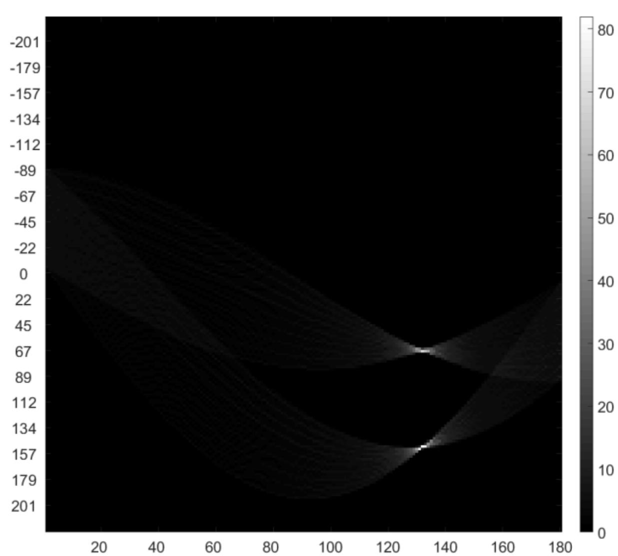


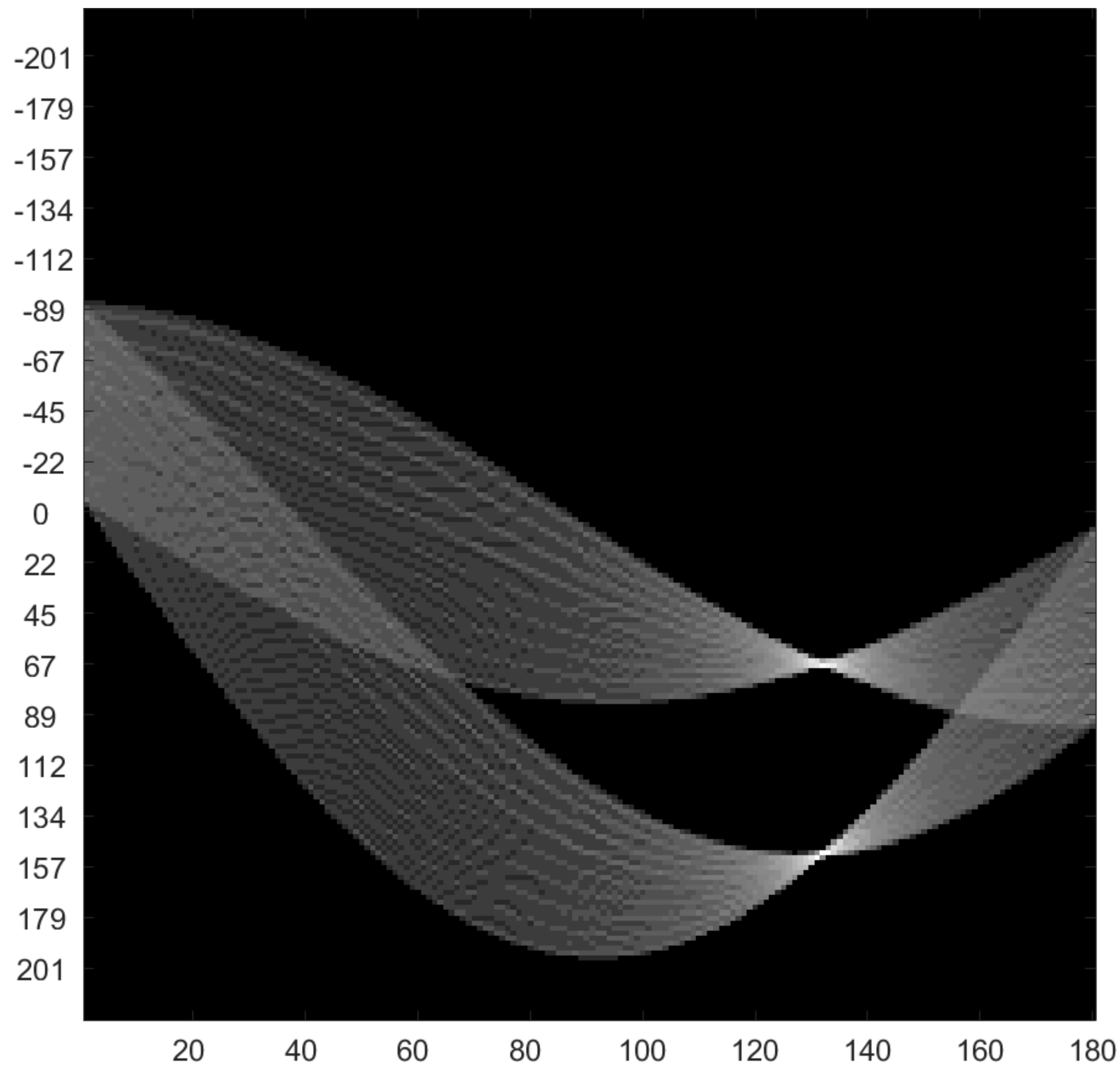


Duda & Hart

- Translation

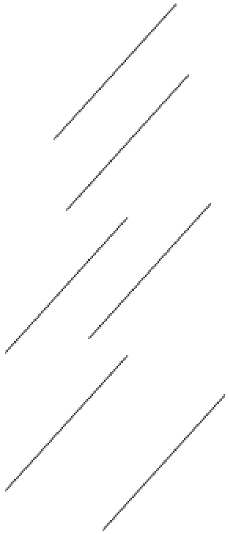


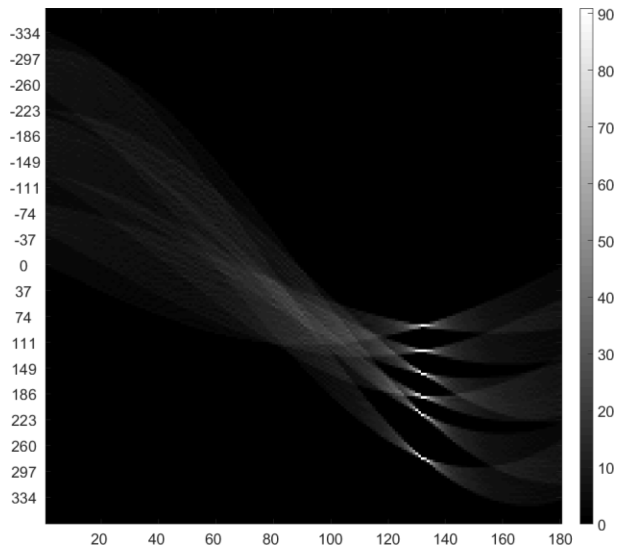


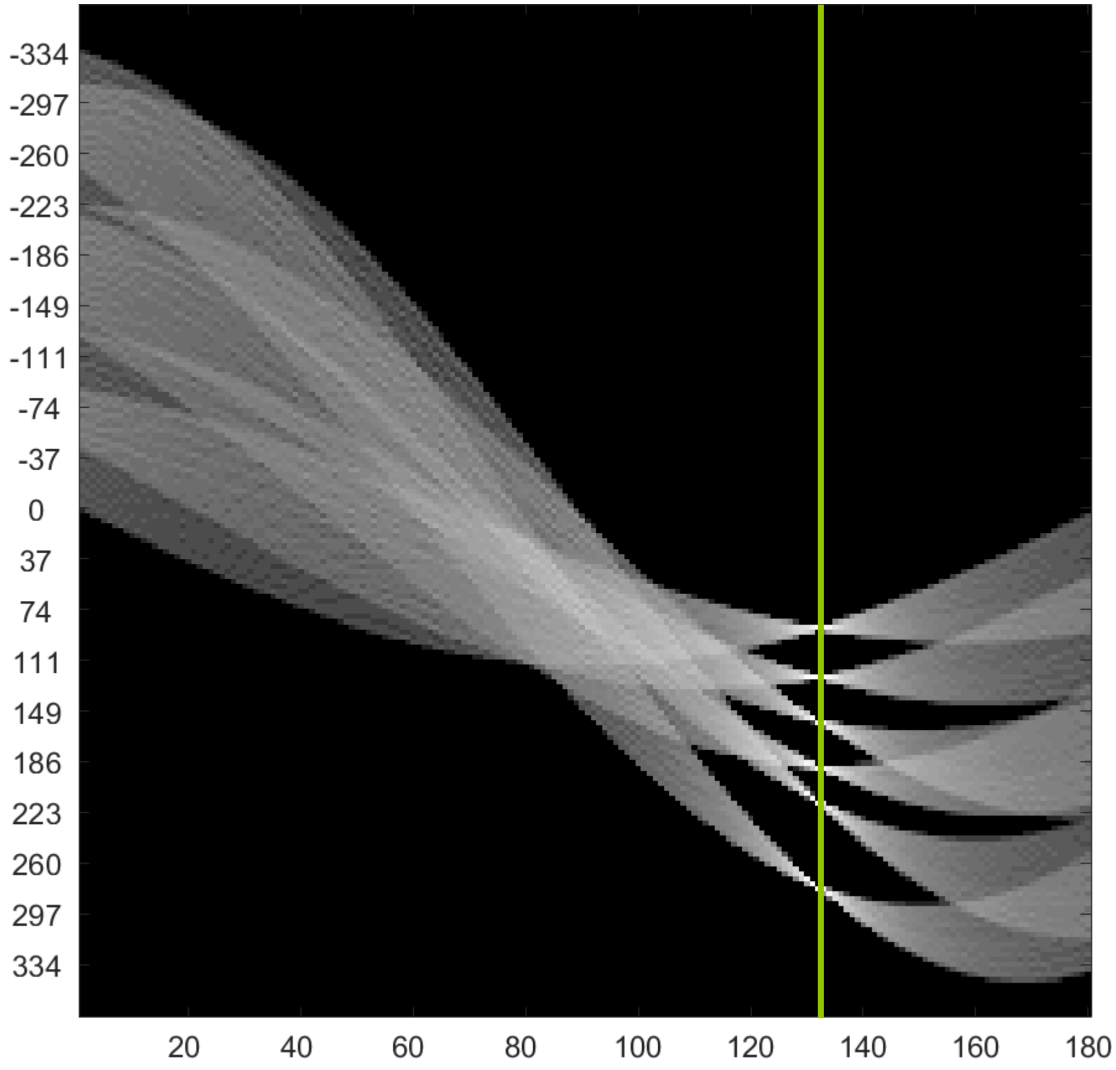


Duda & Hart

- Translation







Duda & Hart

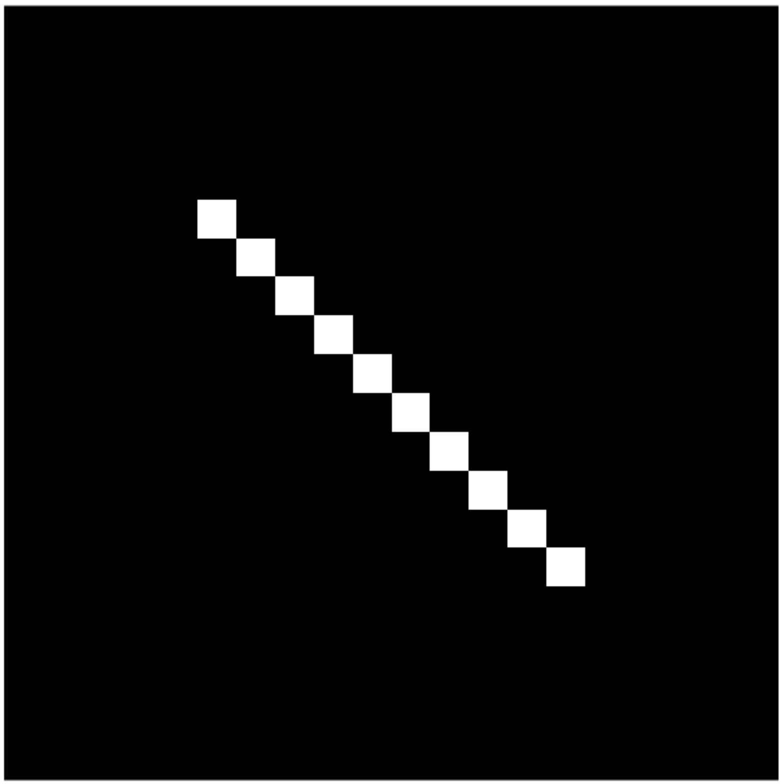
Limitations of the HT

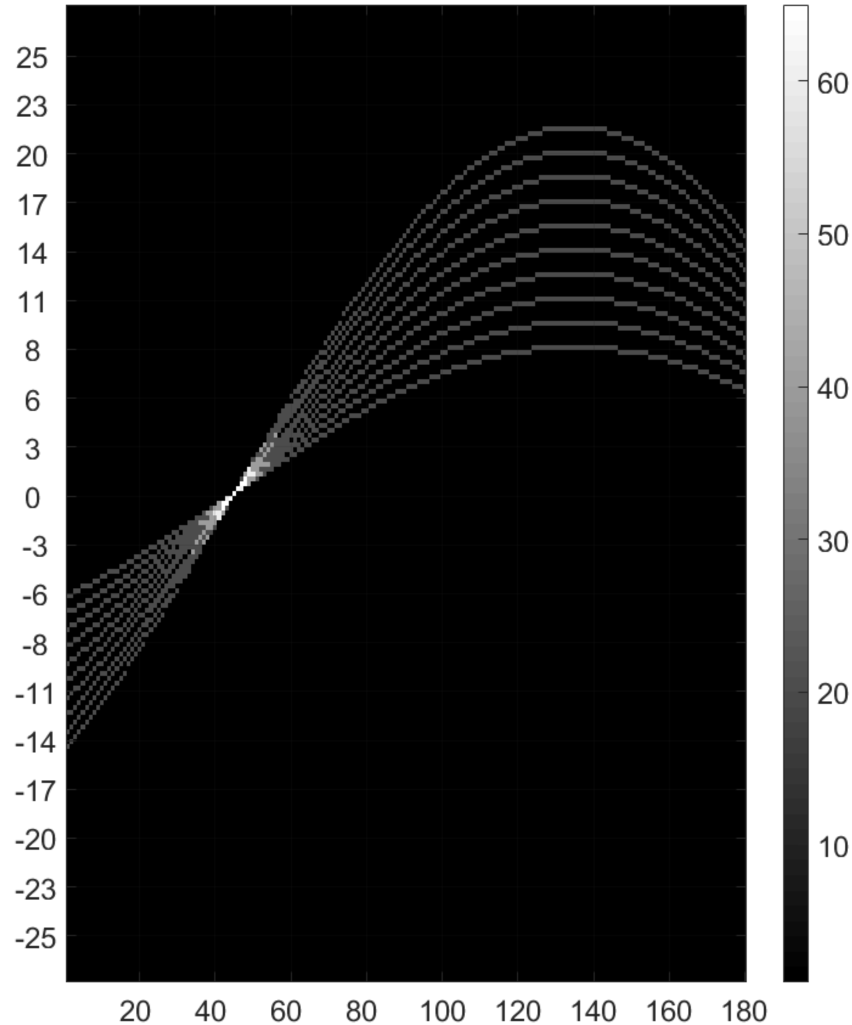
- sensitive to the quantization of both θ and ρ
  - Finer quantization gives better resolution
    - increases the computation time
    - exposes the problem with nearly collinear points

- technique finds collinear points without regard to contiguity
  - meaningless groups of collinear points being detected

Duda & Hart

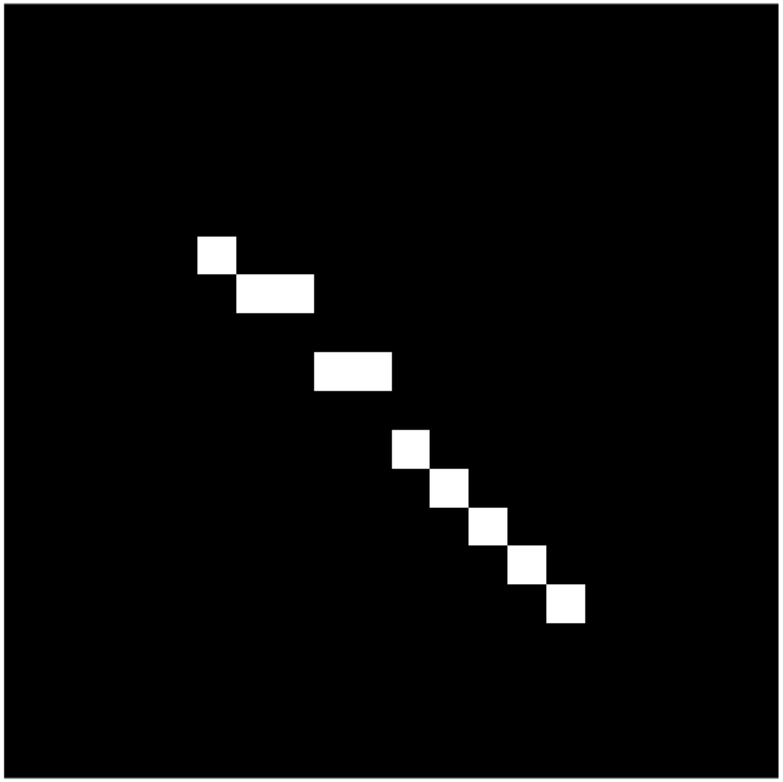
Limitations of the HT

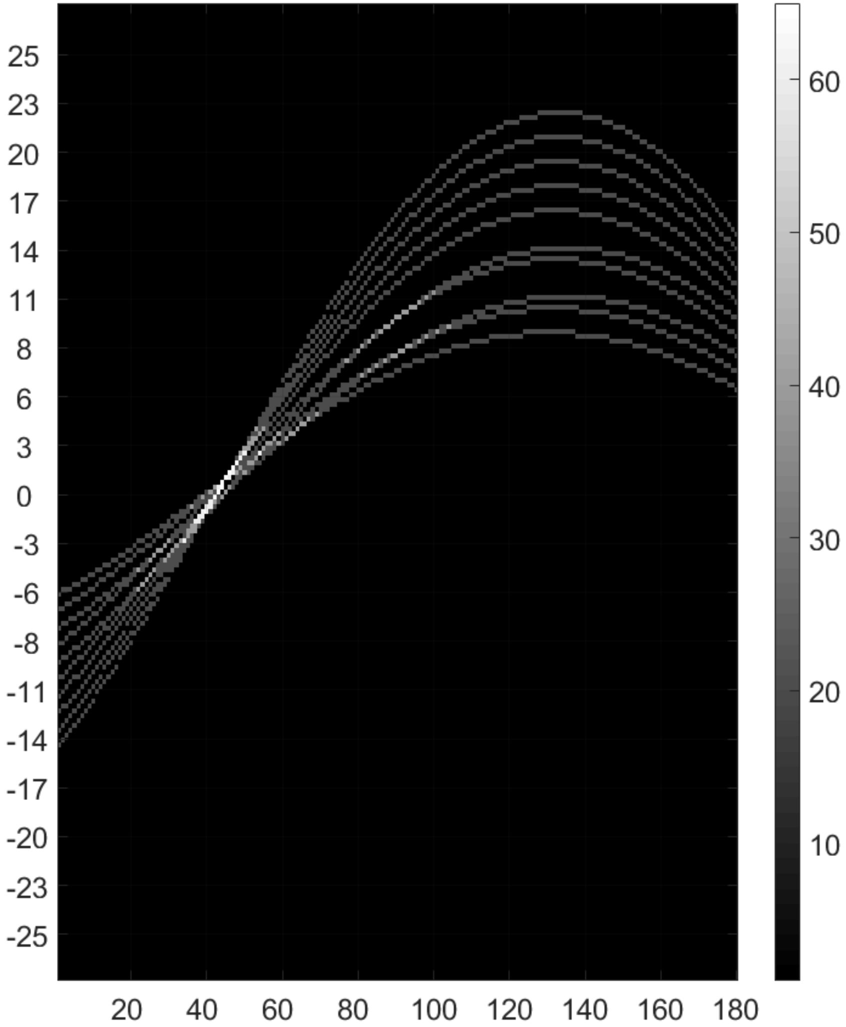




Duda & Hart

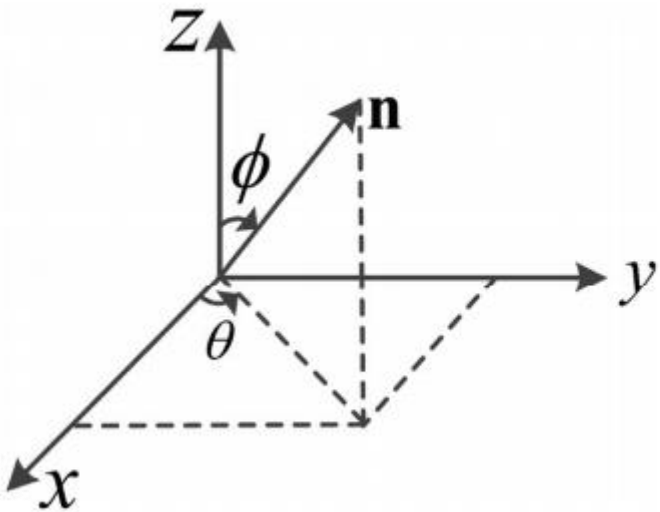
Limitations of the HT

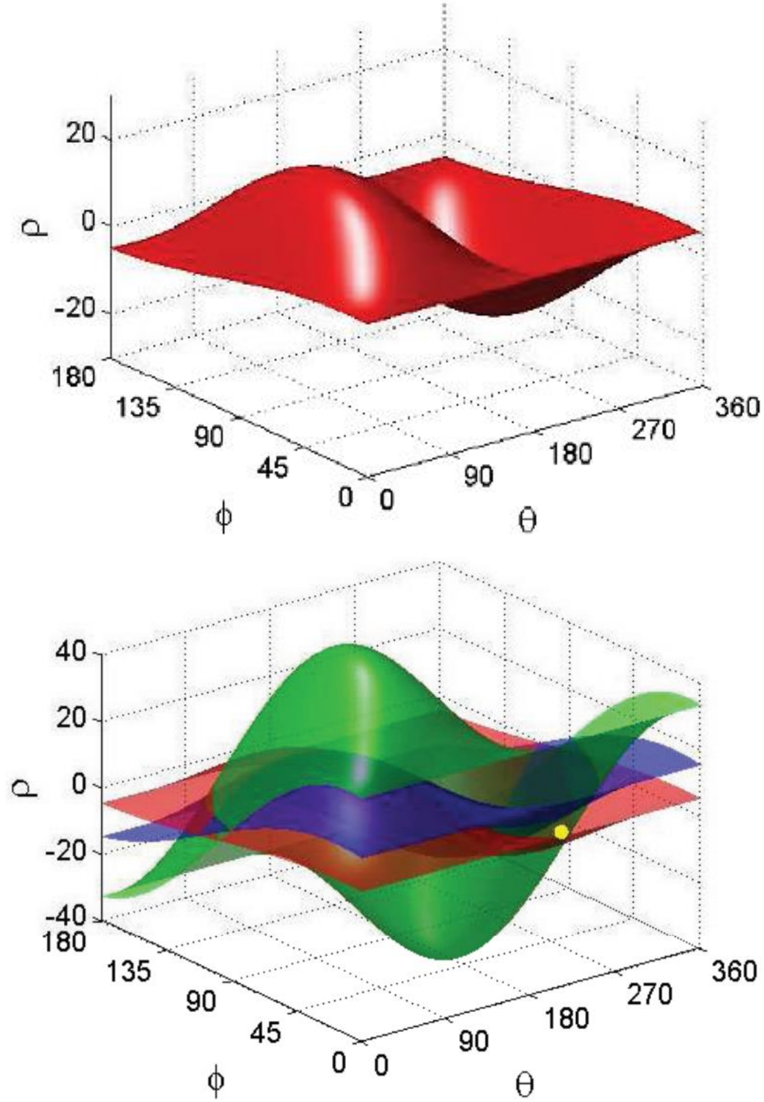




Duda & Hart

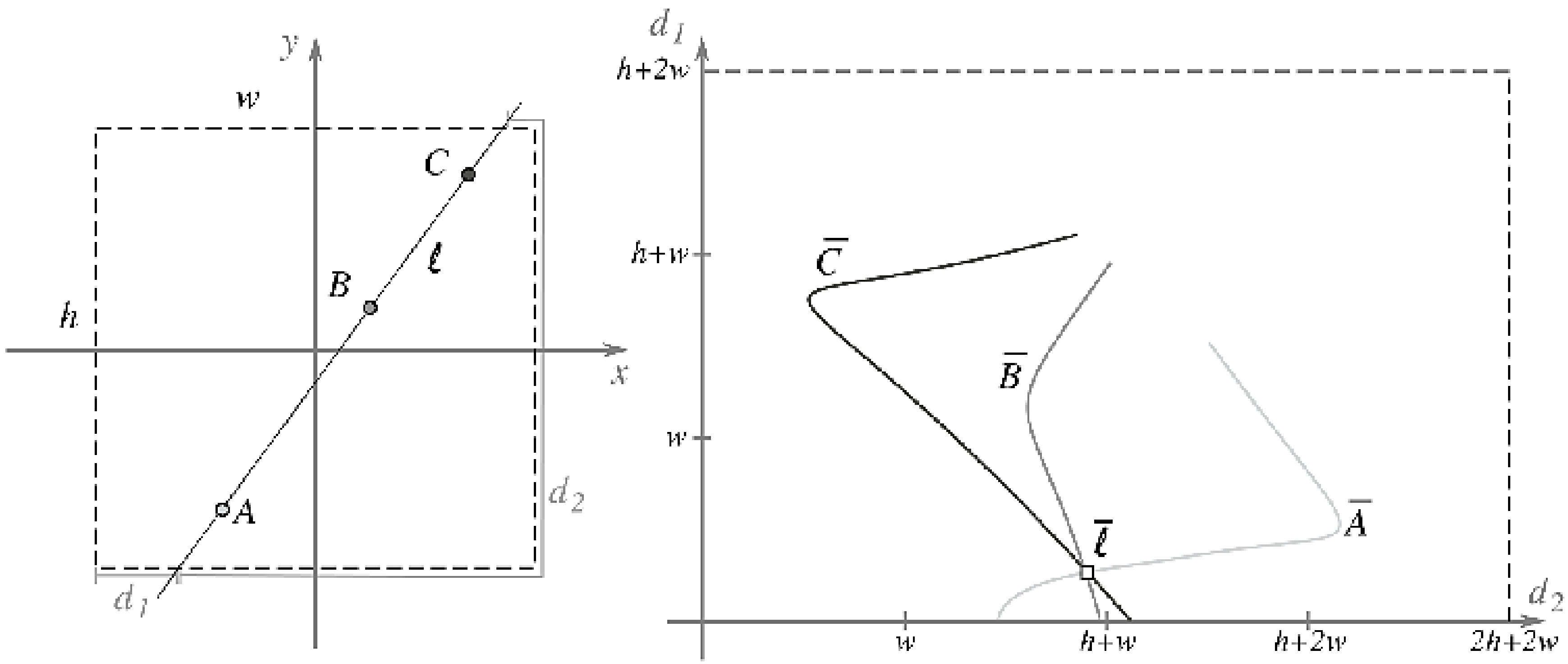
HT in 3D





Zhen Wang and Ghassan AlRegib, Automatic fault surface detection by using 3D Hough transform, SEG Technical Program Expanded Abstracts 2014,p. 1439-1444,2014.

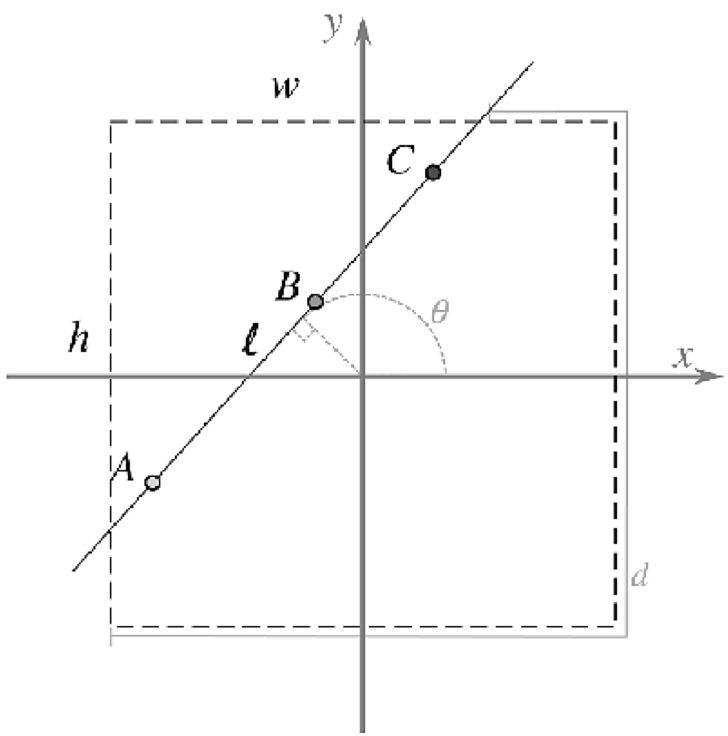
Muff Parametrization



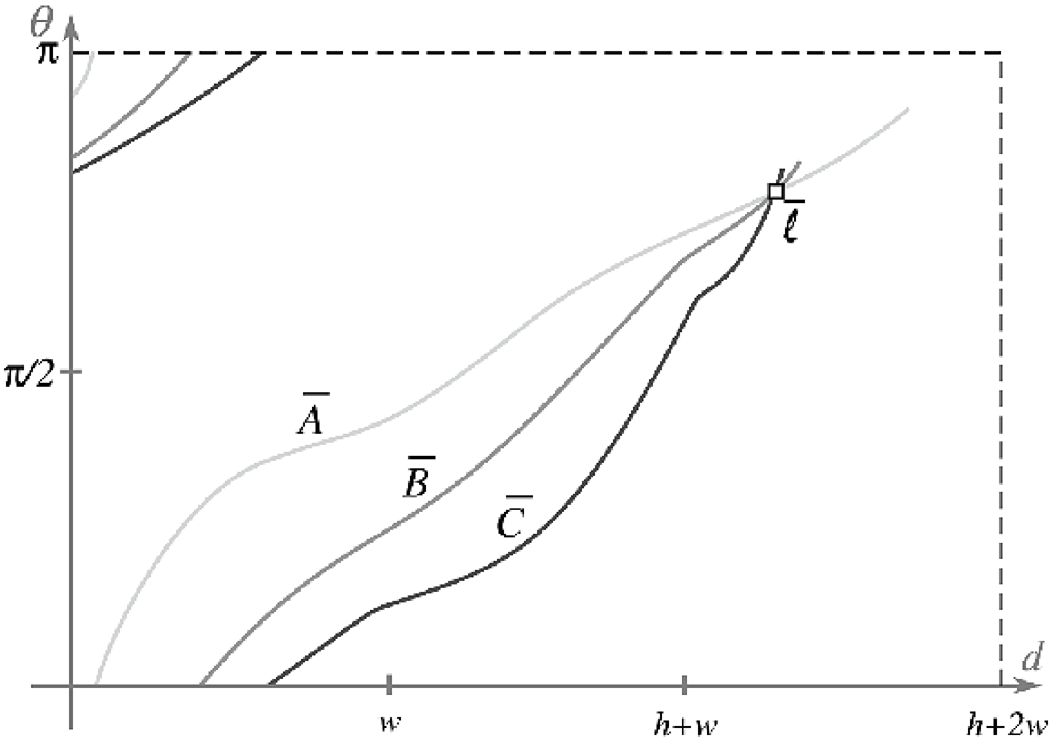
bounding rectangle
around the image

Wallace,R.: A modified Hough transform for lines. In: Proceedings of CVPR 1985,pp.665–667, 1985.

Forman Parameterization

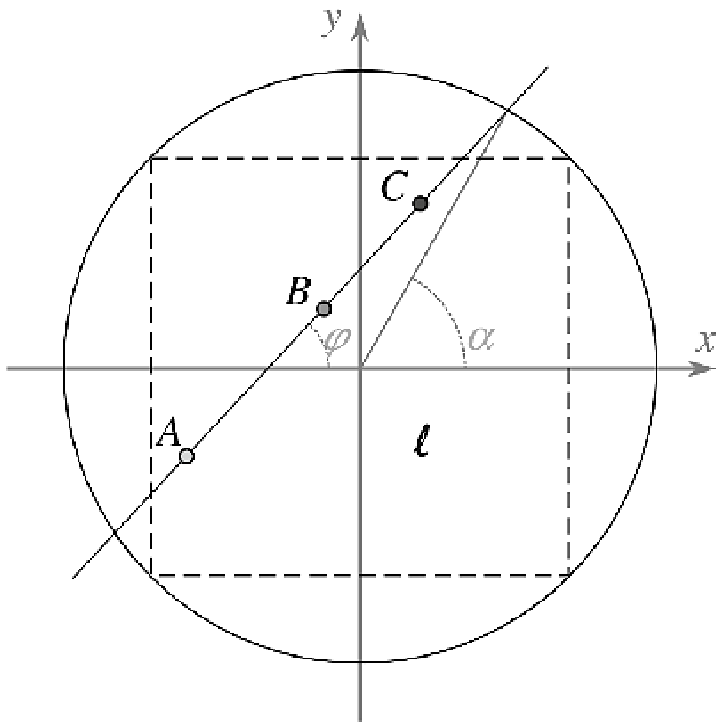


bounding rectangle
around the image

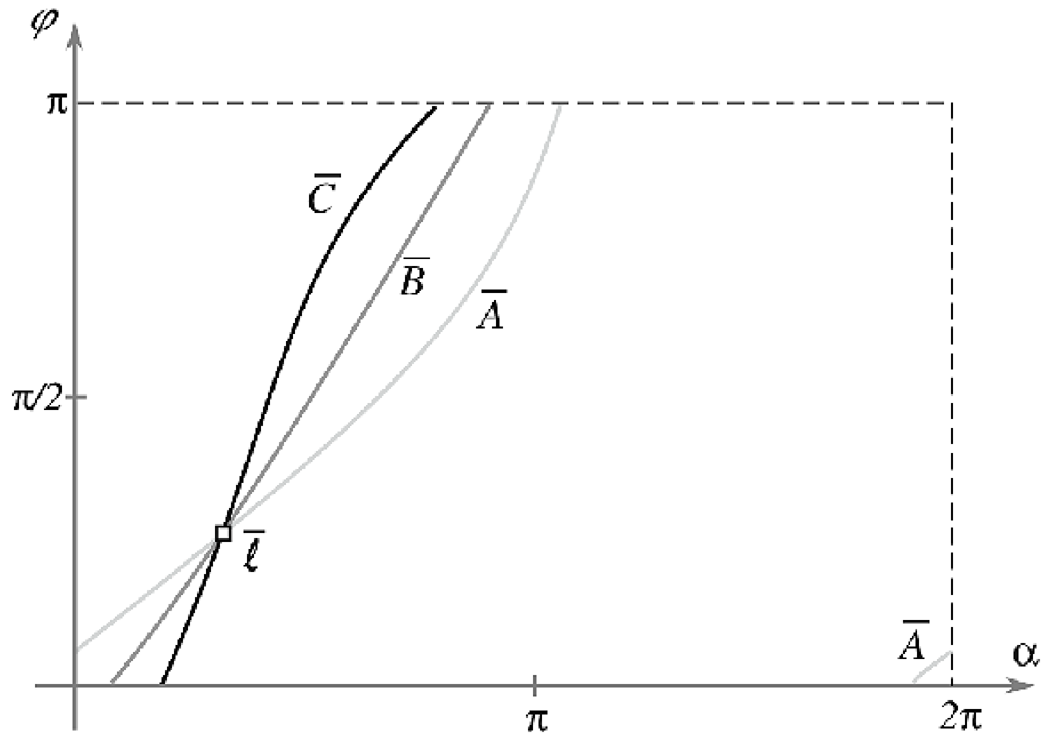


Forman, A.V.: A modified Hough transform for detecting lines in digital imagery. Appl. Artif. Intell. III, 151–160, 1986

Fan–Beam Parameterization

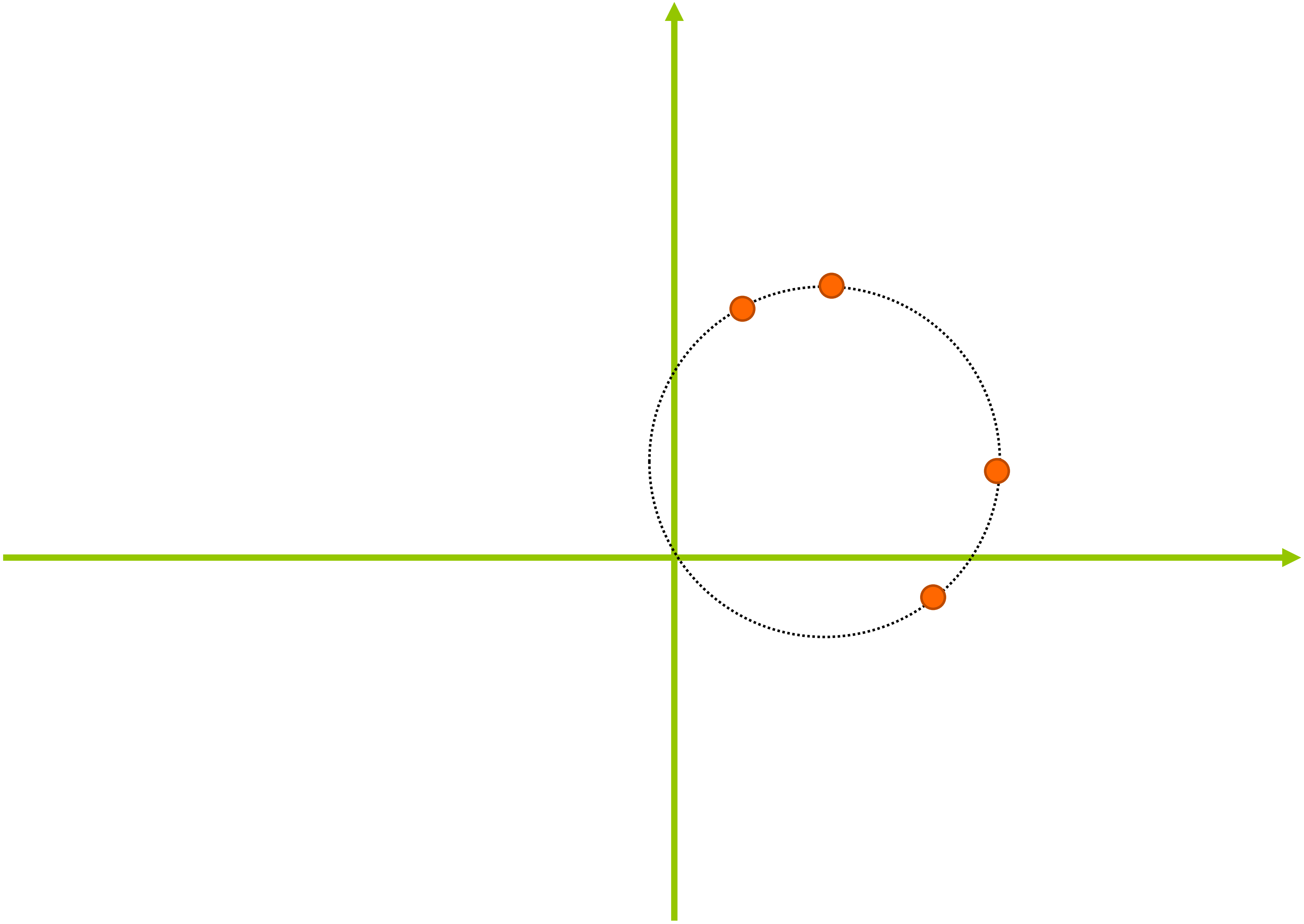


bounding circle
around the image

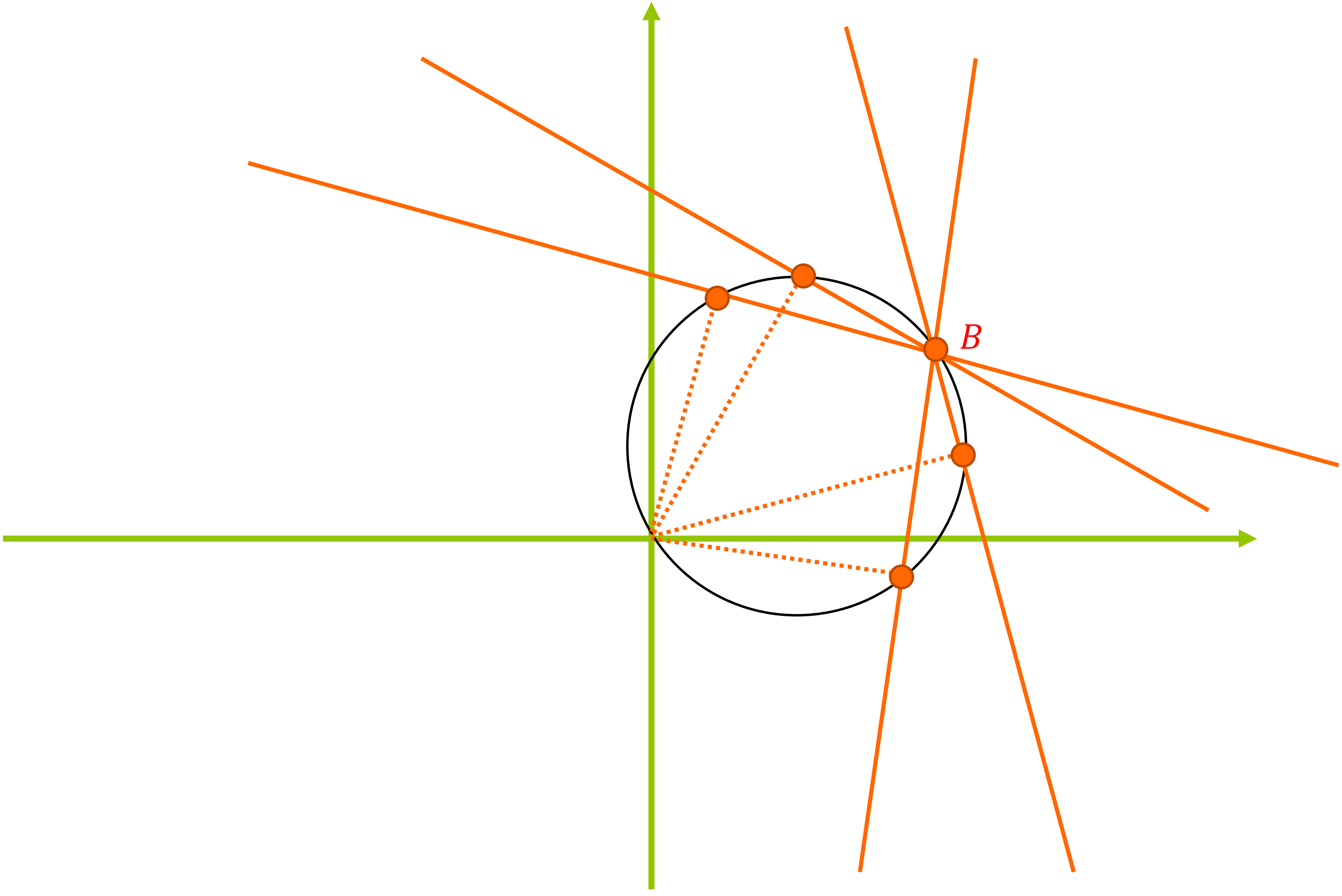


Eckhardt, U., Maderlechner, G.: Application of the projected Hough transform in picture processing. In: Proceedings of the 4th International Conference on Pattern Recognition, pp. 370–379. Springer, London, UK, 1988

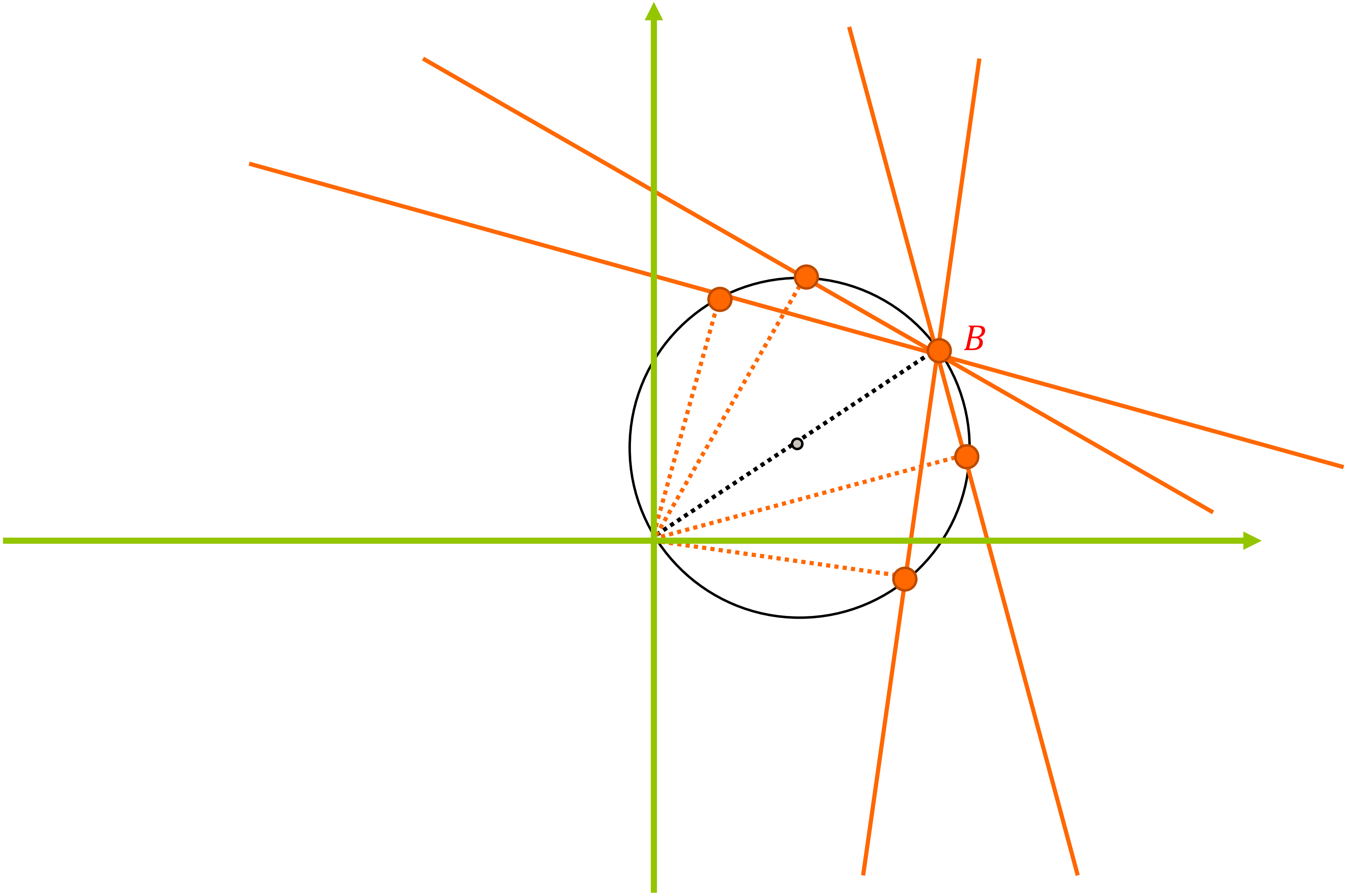
Circle Transform



Circle Transform



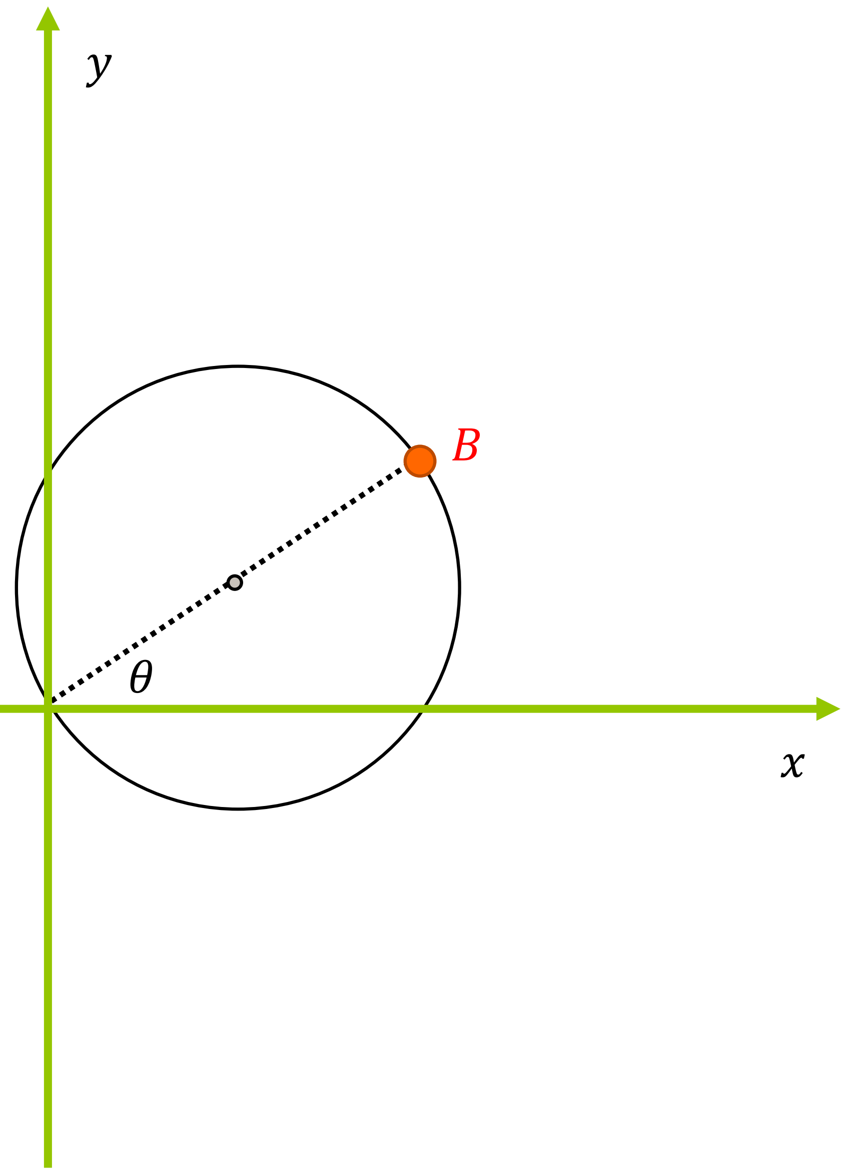
Circle Transform



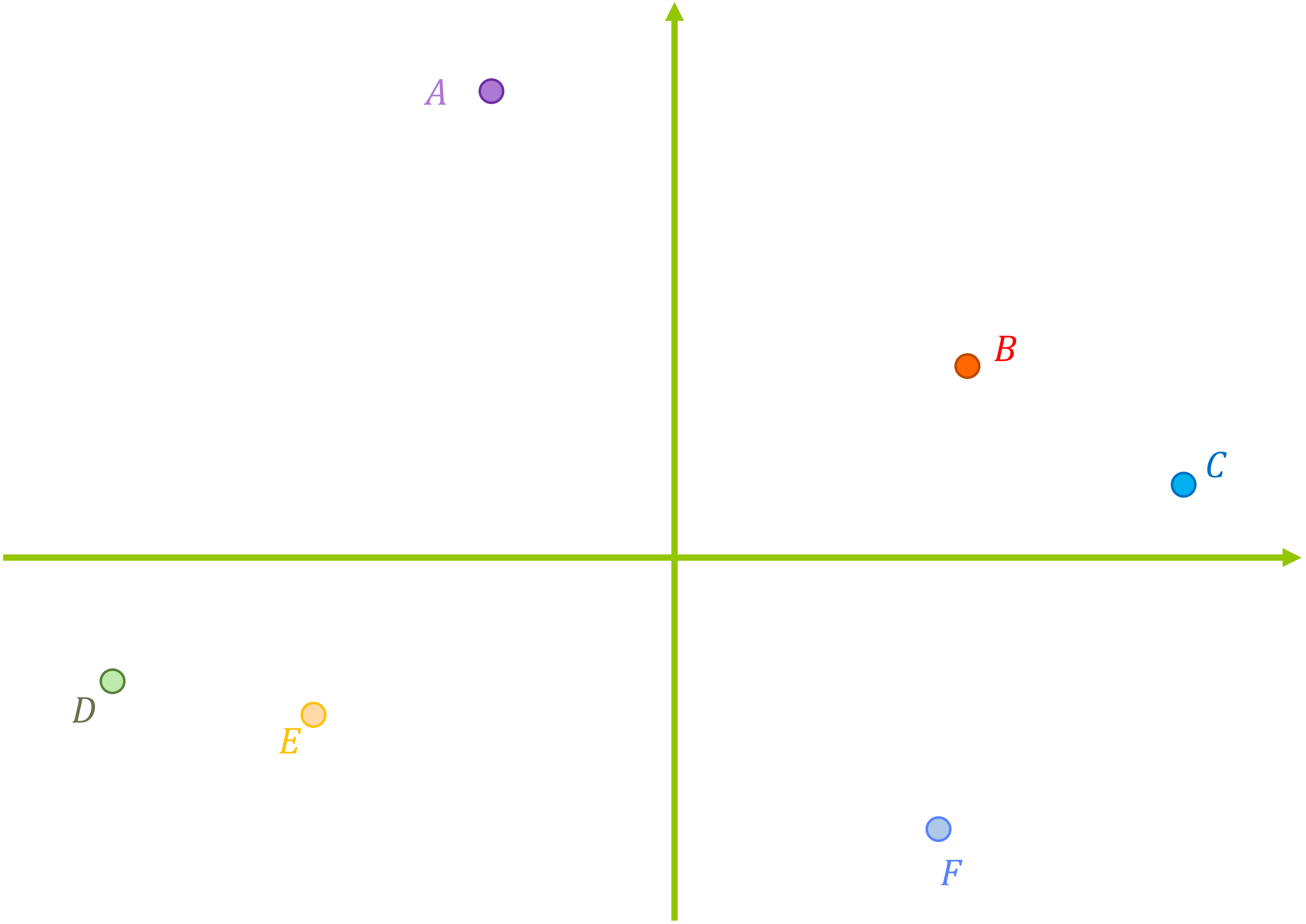
Circle Transform

$$P = (a, b)_{\mathbb{E}^2} = (\rho\cos\theta\,, \rho\sin\theta)$$

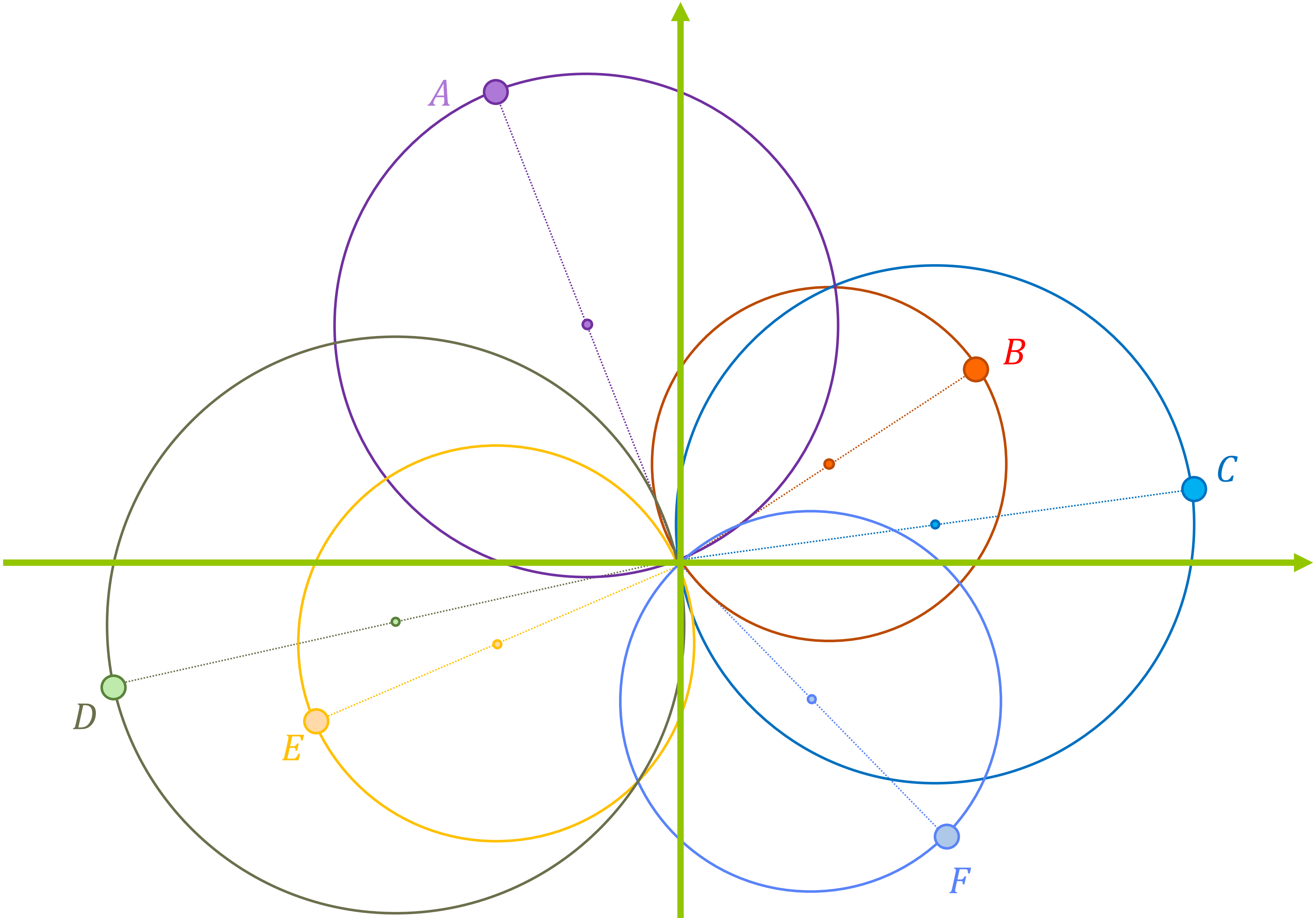
$$Circle: \begin{matrix} x = \frac{1}{2}(\rho\cos\theta + O_x) \\ y = \frac{1}{2}(\rho\sin\theta + O_y) \end{matrix}$$



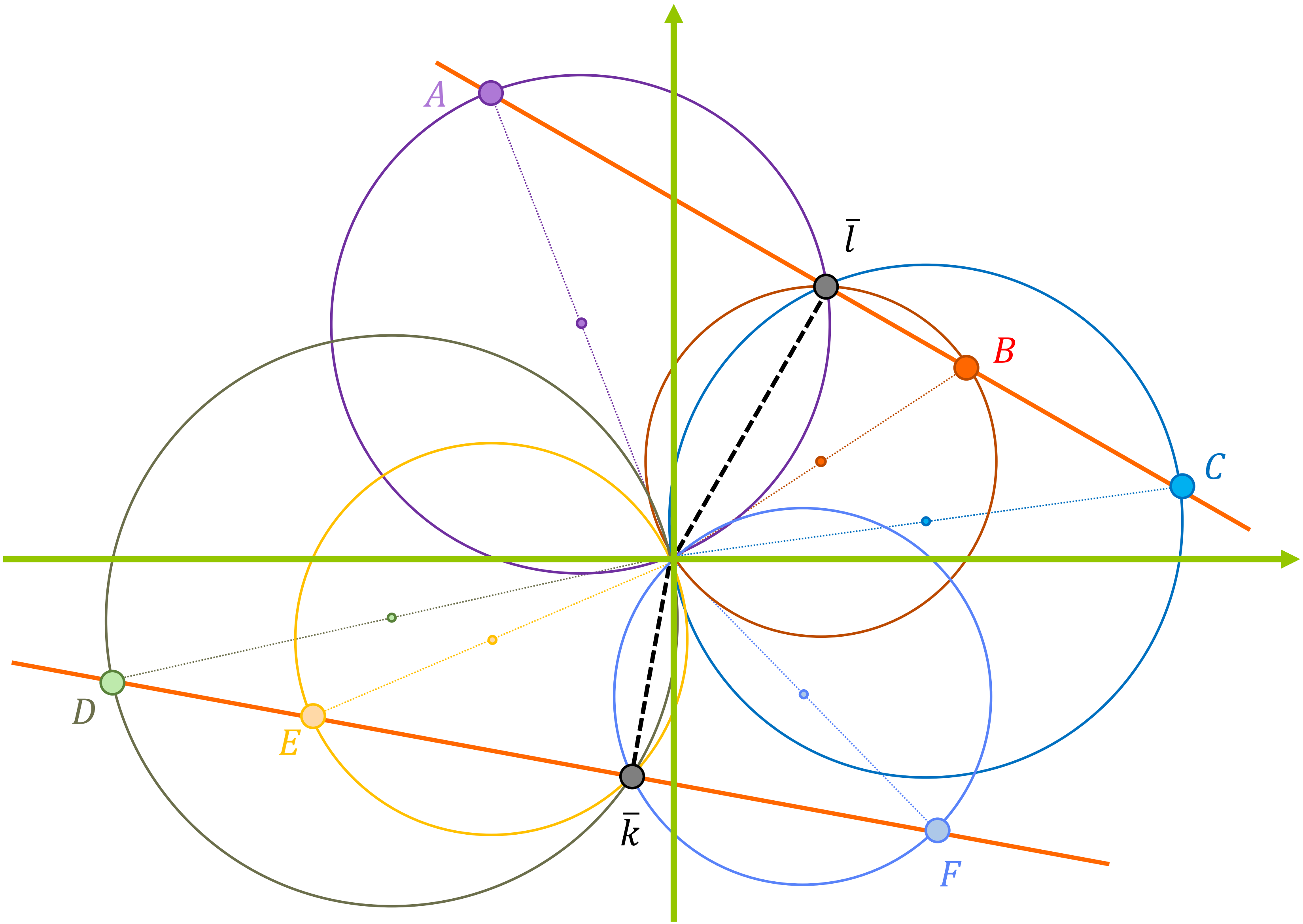
Circle Transform



Circle Transform

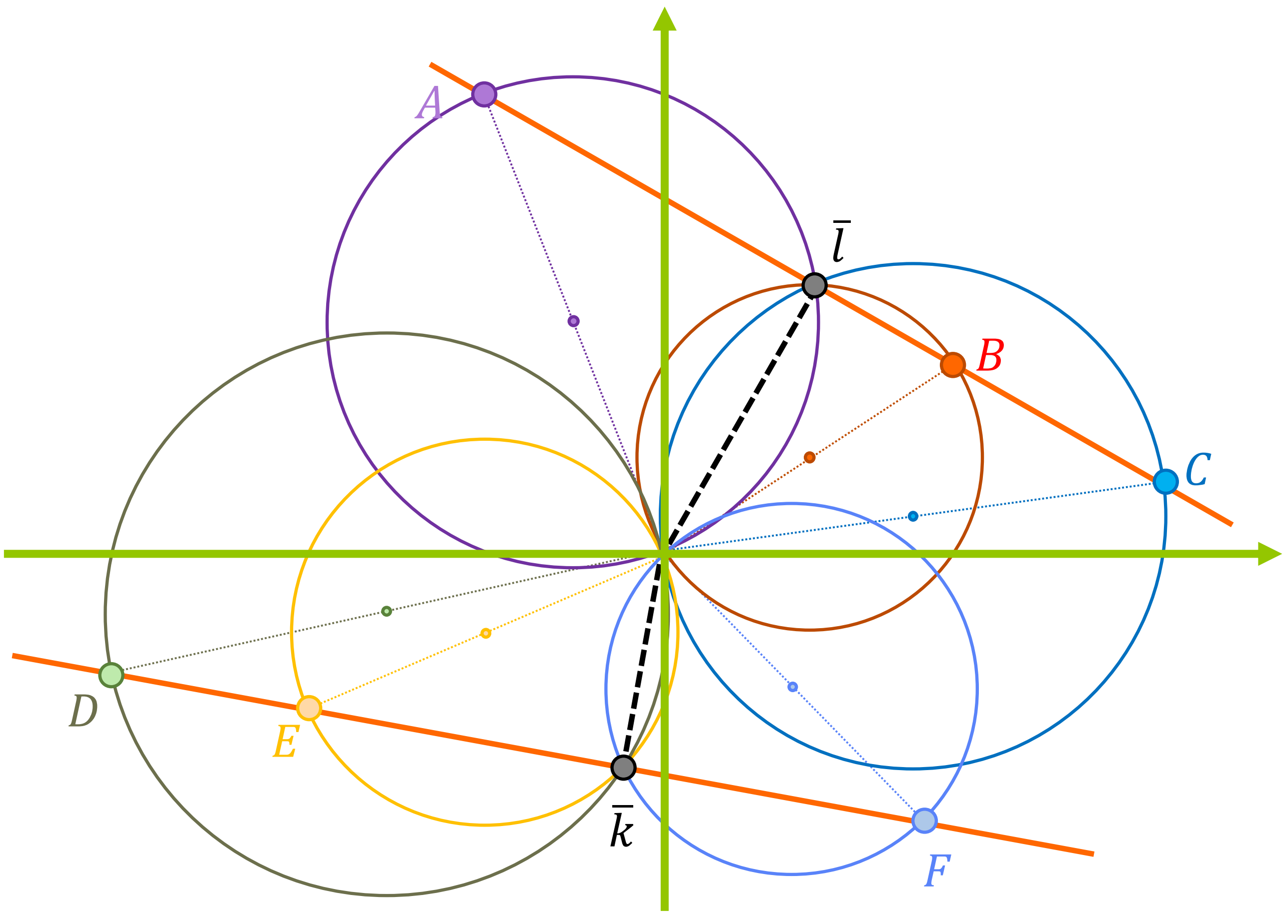


Circle Transform



Circle Transform

Problem: the number of votes is much higher by origin than in the rest of the accumulator space



HT for Analytic Curve

$$f(\boldsymbol{x}, \boldsymbol{a}) = 0$$
$$\boldsymbol{x} \,..\, image\ point$$
$$\boldsymbol{a} \,..\, parameter\ vector$$

Analytic form	Parameters	Equation
Line	ρ, θ	$x \cos\theta + y \sin\theta = \rho$
Circle	a, b, ρ	$(x-a)^2 + (y-b)^2 = \rho^2$
Parabola	a, b, ρ_x, θ	$(y-b)^2 = 2\rho_x(x-a)^*$
Ellipse	$a, b, \rho_x, \rho_y, \theta$	$\frac{(x-a)^2}{{\rho_x}^2} + \frac{(y-b)^2}{{\rho_y}^2} = 1^*$
$*$ Plus rotation by θ		

HT for Analytic Curve – Circle

[Duda&Hart 1972]

Unknown radius ρ

$$(x-a)^2 + (y-b)^2 = \rho^2$$

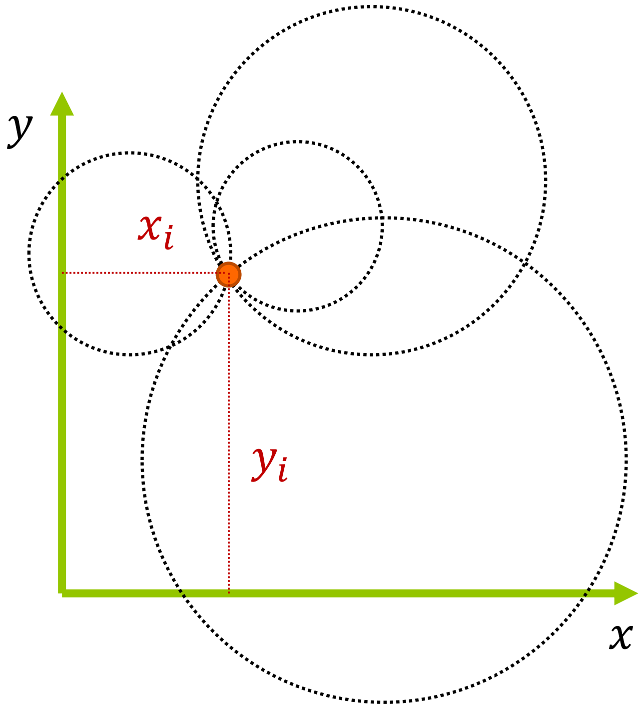
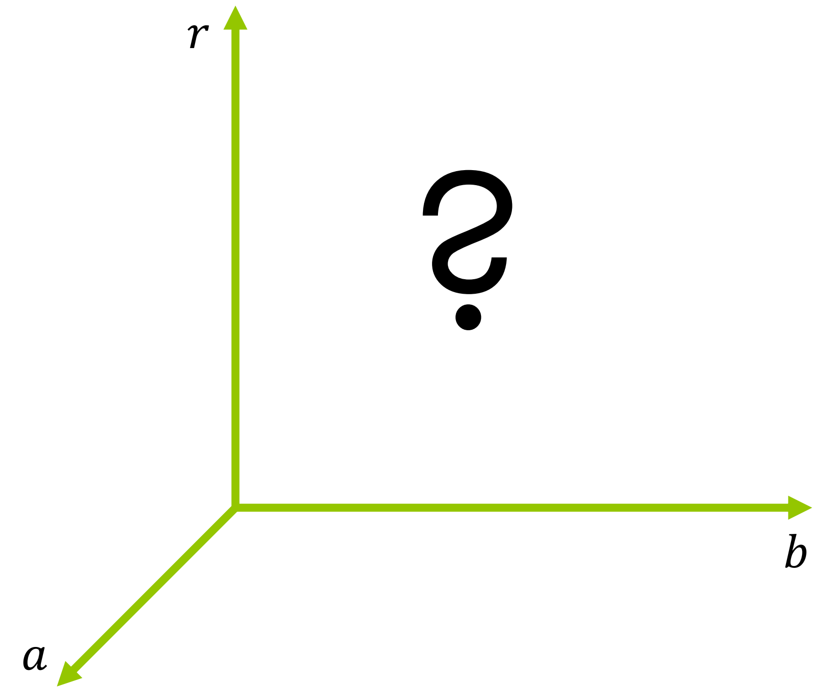


Image space



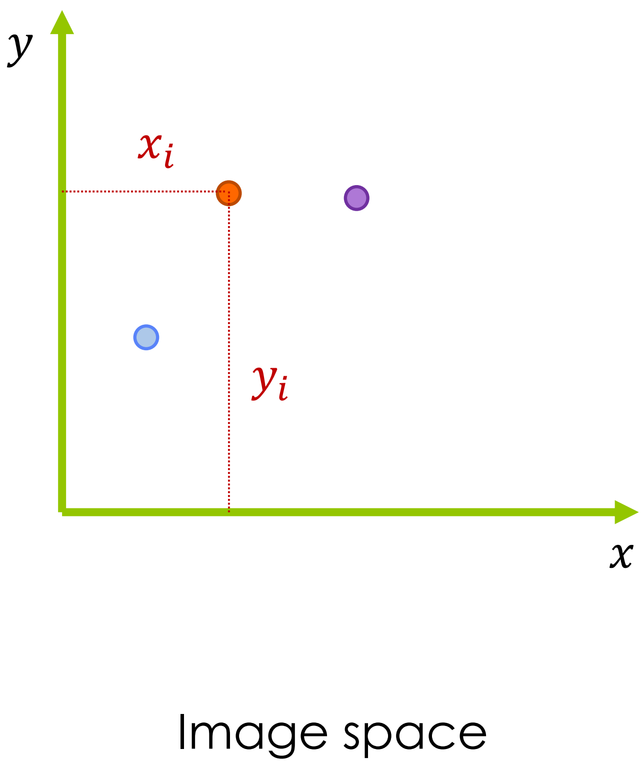
Hough space

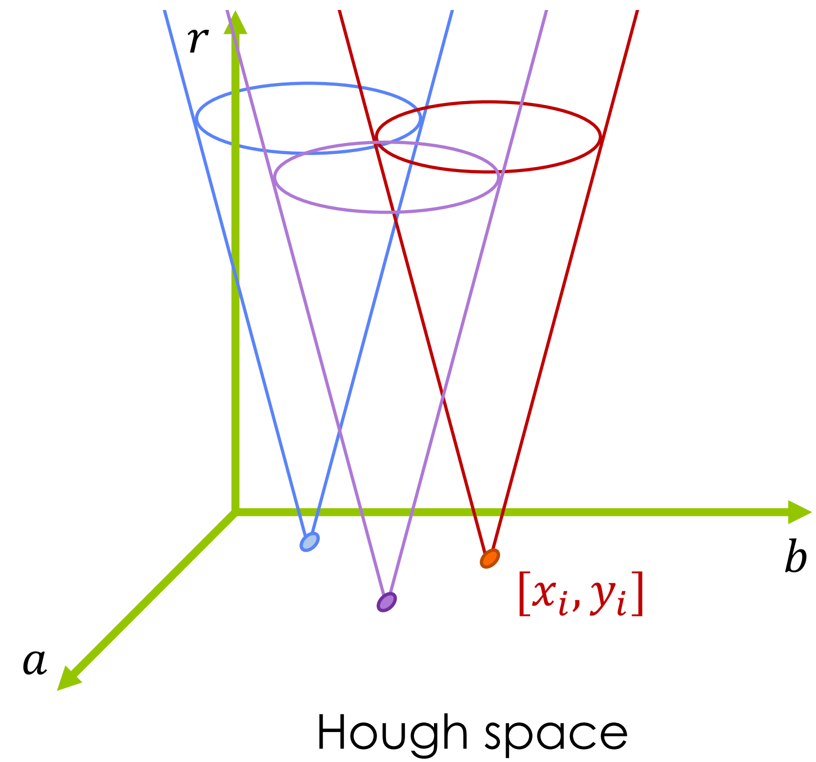
HT for Analytic Curve – Circle

[Ballard 1981]

Unknown radius ρ

$$(x - a)^2 + (y - b)^2 = \rho^2$$





HT for Analytic Curve – Circle

[Ballard 1981]

Known radius ρ

$$(x-a)^2 + (y-b)^2 = \rho^2$$

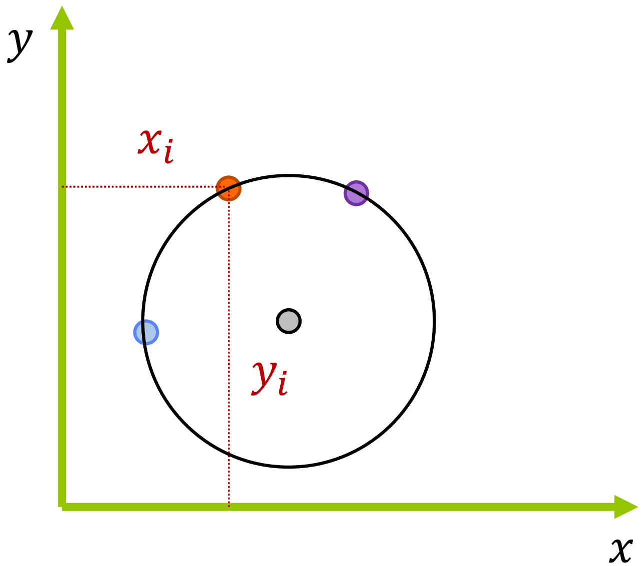
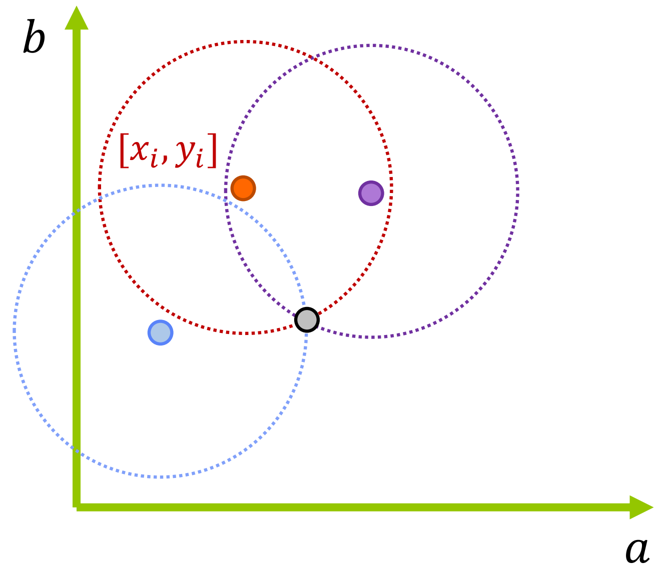


Image space

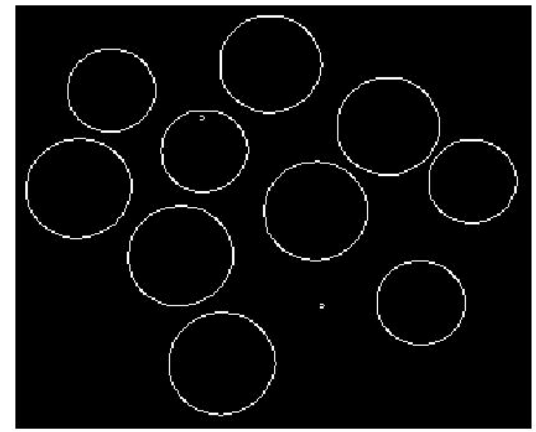


Hough space

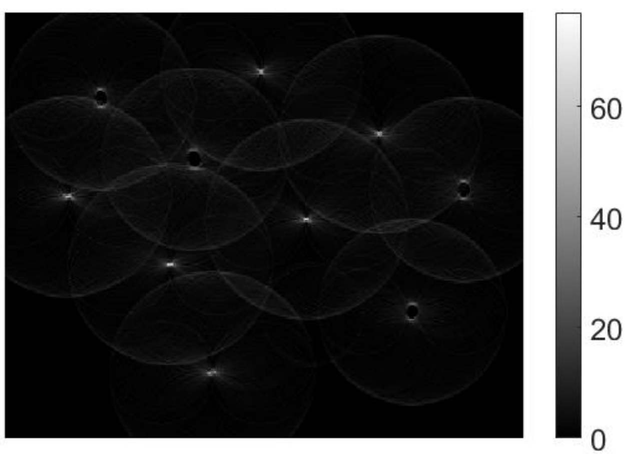
HT for Analytic Curve – Circle

Known radius ρ

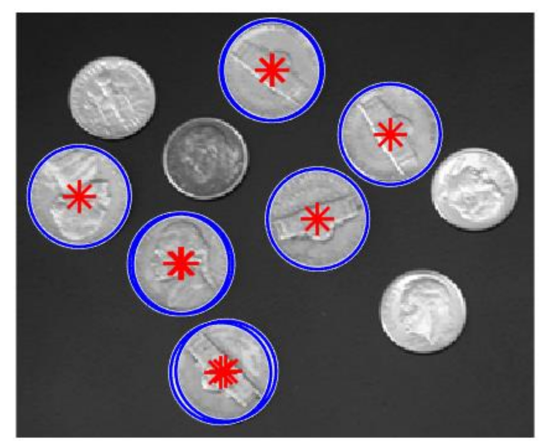
Image space



Thresholding & Canny detector



Hough space

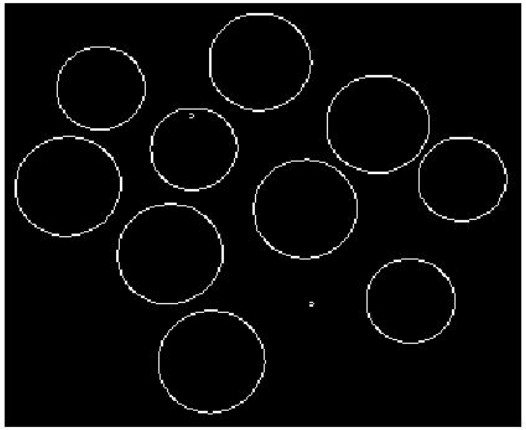


Detected circles

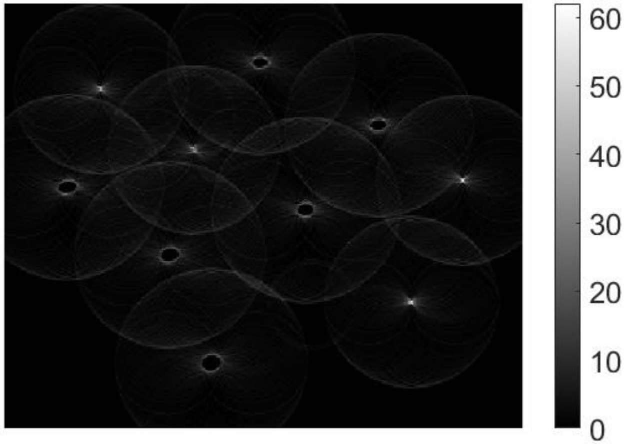
HT for Analytic Curve – Circle

Known radius ρ

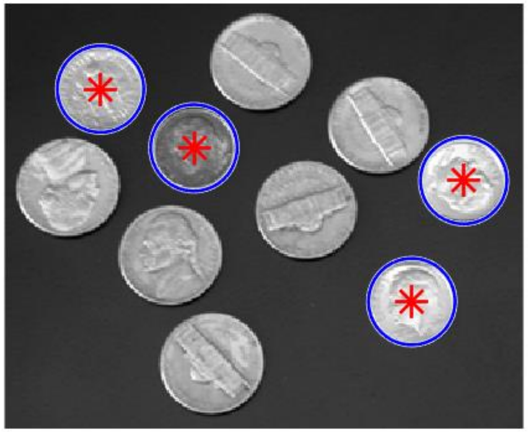
Image space



Thresholding & Canny detector



Hough space

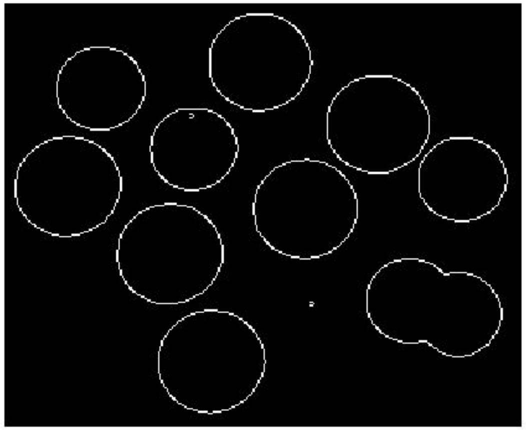


Detected circles

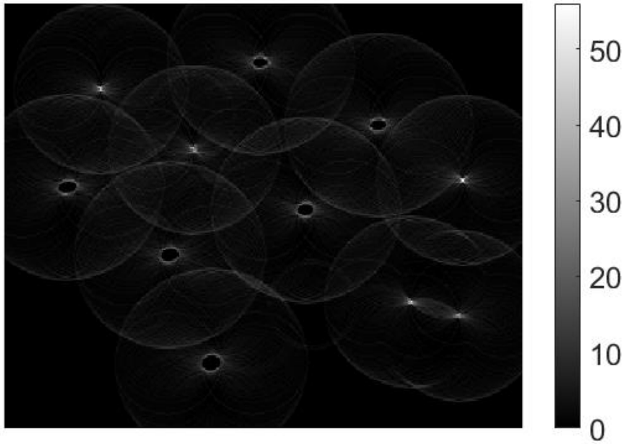
HT for Analytic Curve – Circle

Known radius ρ

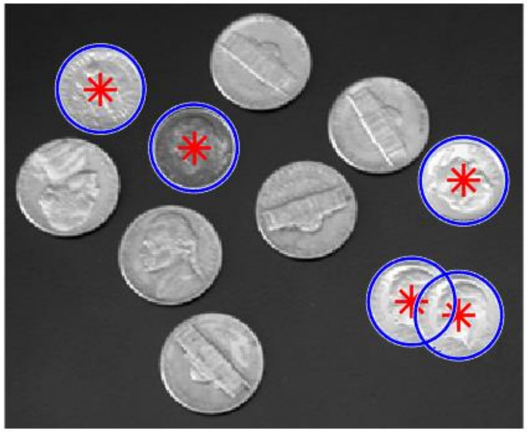
Image space



Thresholding & Canny detector



Hough space

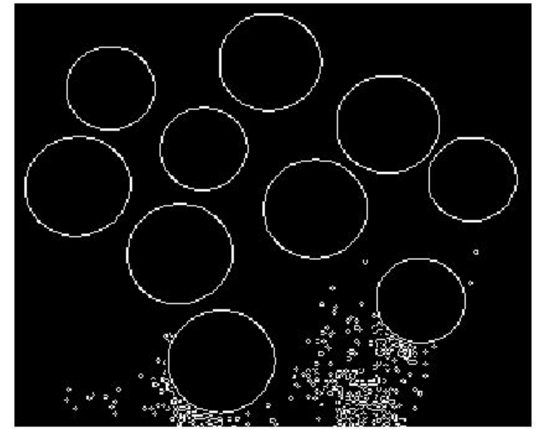


Detected circles

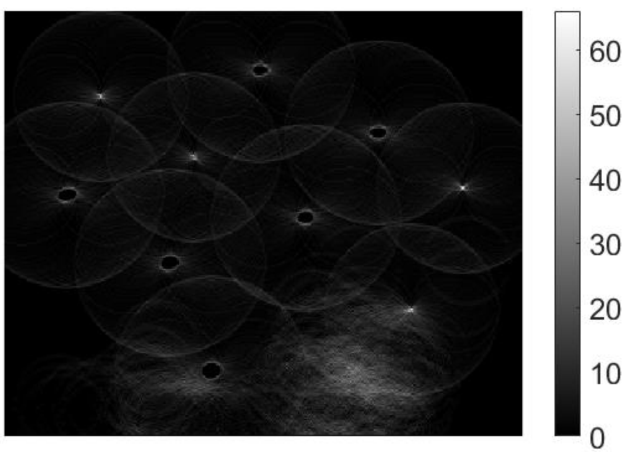
HT for Analytic Curve – Circle

Known radius ρ

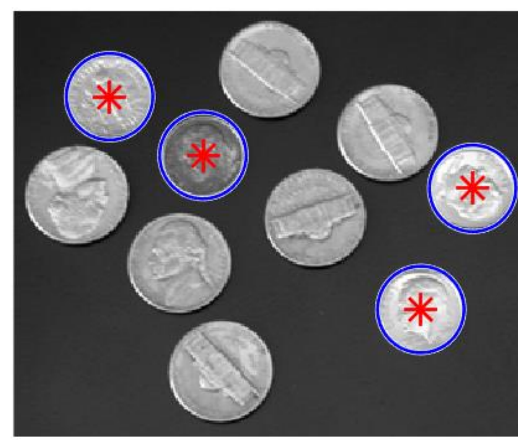
Image space



Thresholding & Canny detector



Hough space

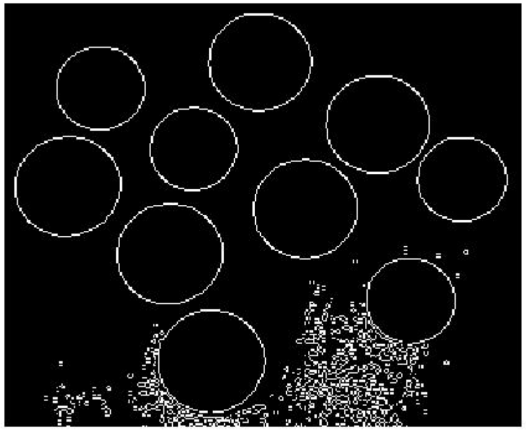


Detected circles

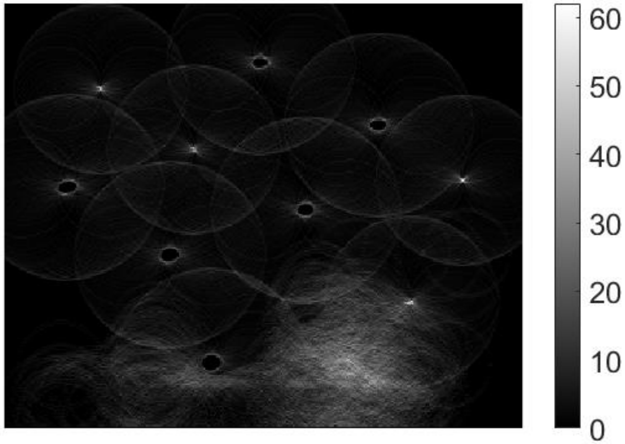
HT for Analytic Curve – Circle

Known radius ρ

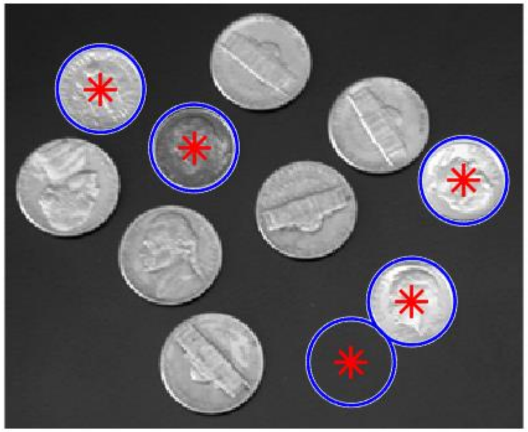
Image space



Thresholding & Canny detector



Hough space



Detected circles

HT for Analytic Curve – Circle

[Ballard 1981]

Unknown radius ρ

Known directional information of the edge

$$(x-a)^2 + (y-b)^2 = \rho^2$$

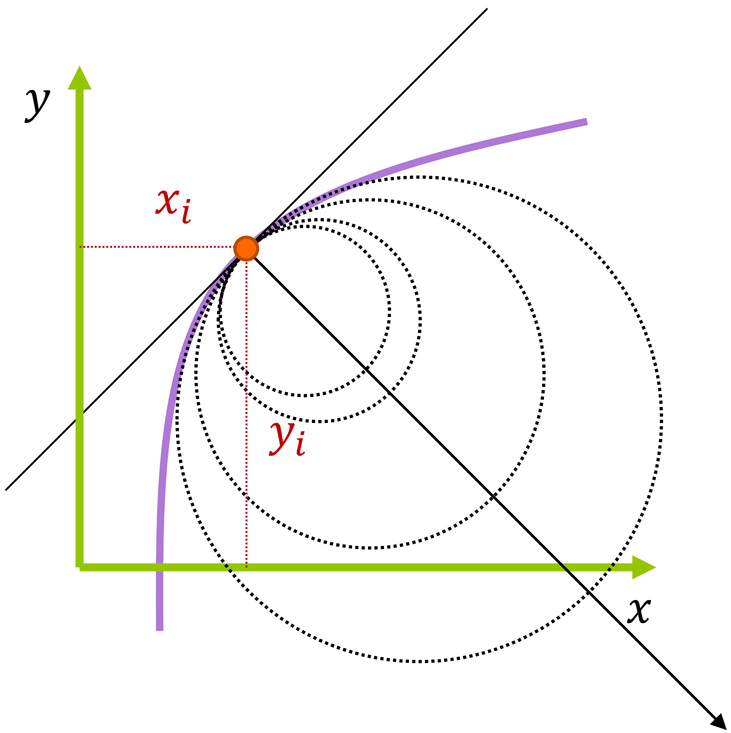
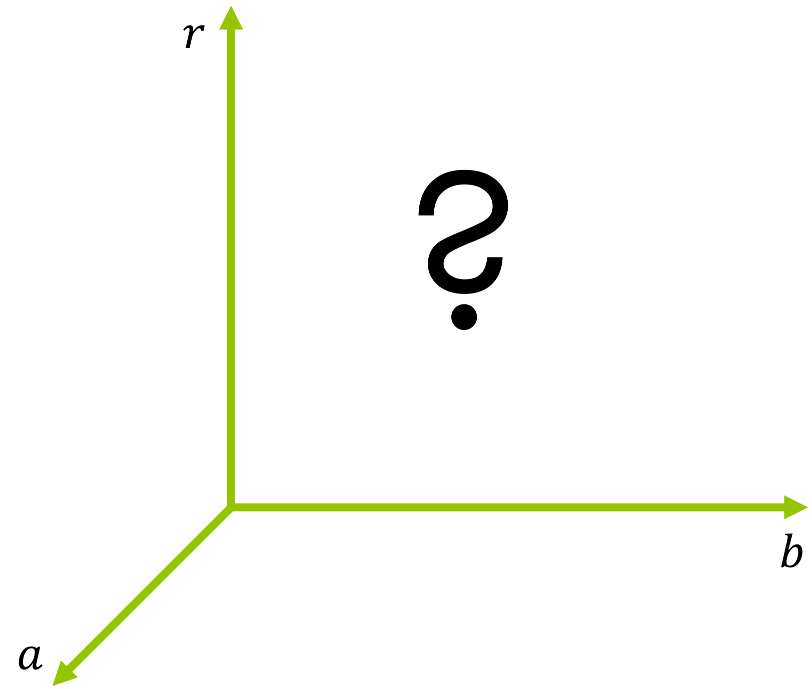


Image space



Hough space

HT for Analytic Curve – Circle

[Ballard 1981]

Unknown radius ρ

Known directional information of the edge

$$(x-a)^2 + (y-b)^2 = \rho^2$$

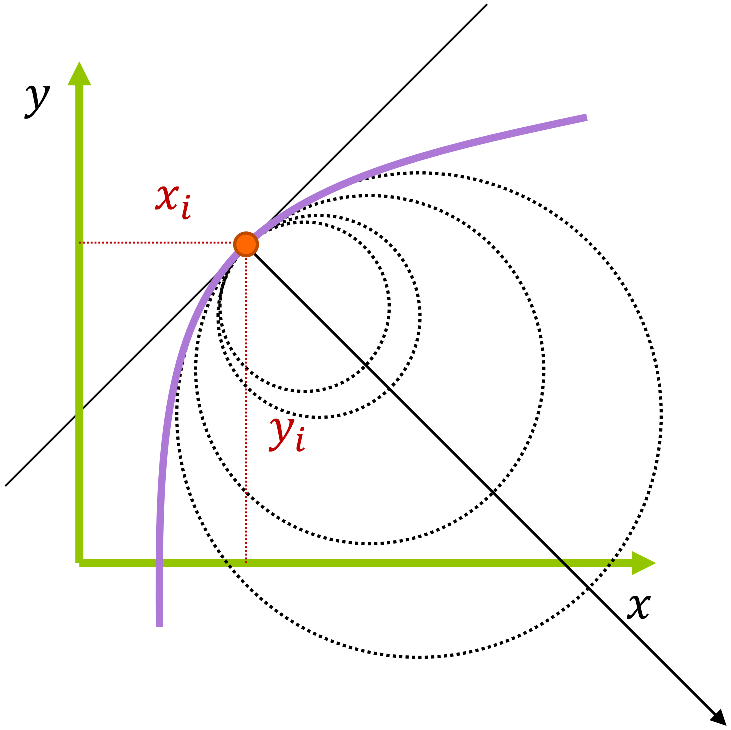
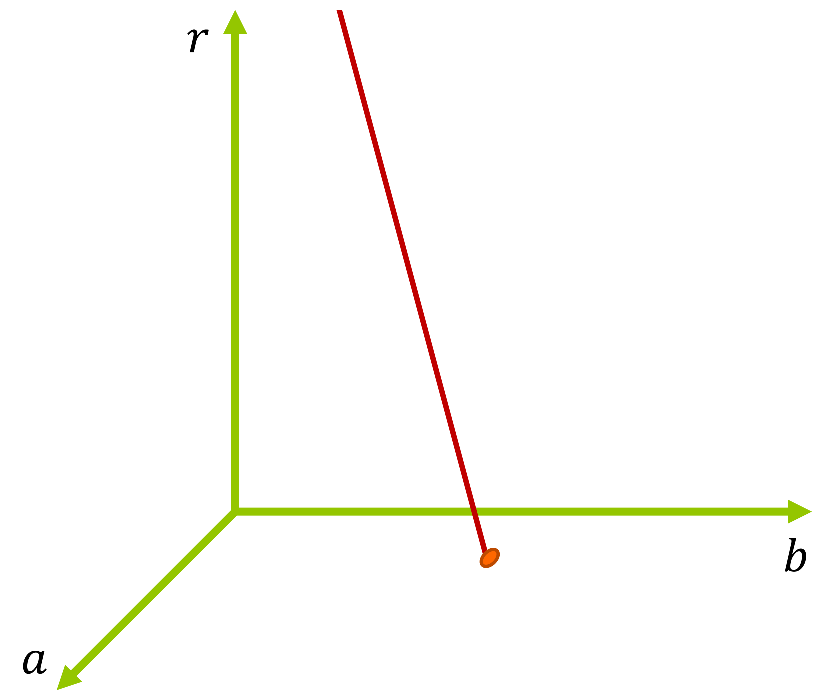


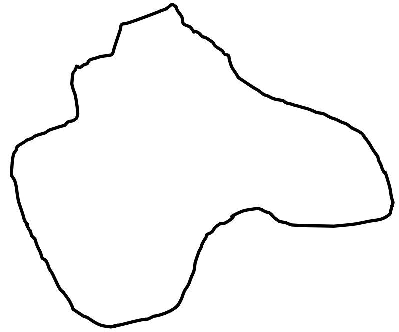
Image space



Hough space

HT for any given curve

in a specific orientation and size



P. M. Merlin and D. J. Farber, "A Parallel Mechanism for Detecting Curves in Pictures," in IEEE Transactions on Computers, vol. C-24, no. 1, pp. 96-98, Jan. 1975.

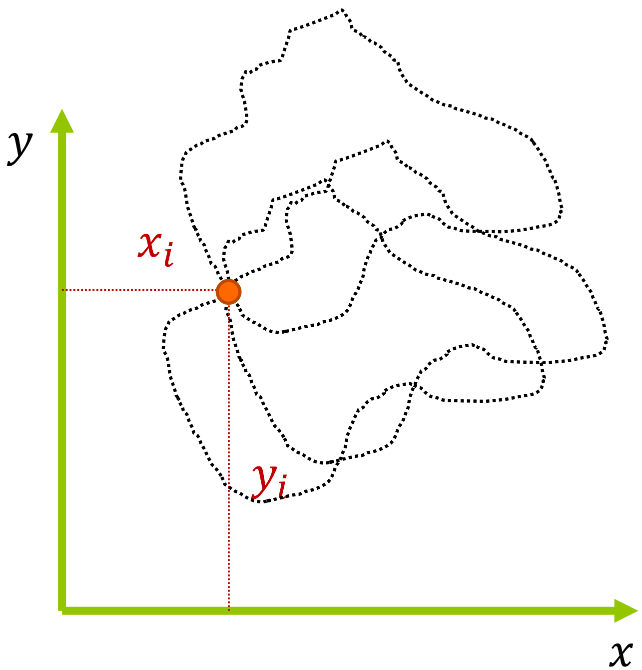
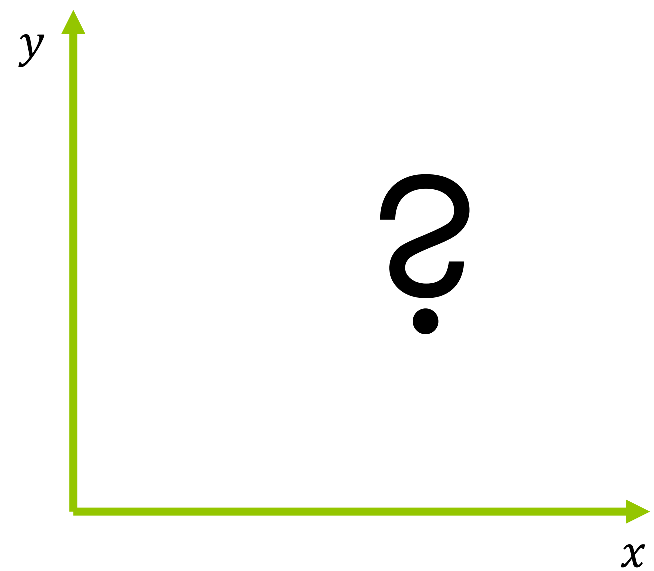


Image space



Hough space

HT for any given curve

[Merlin 1975]

in a specific orientation and size

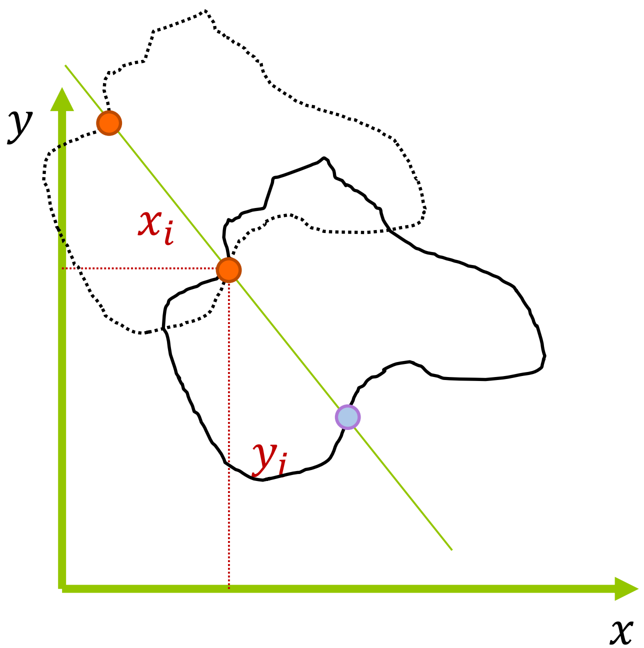
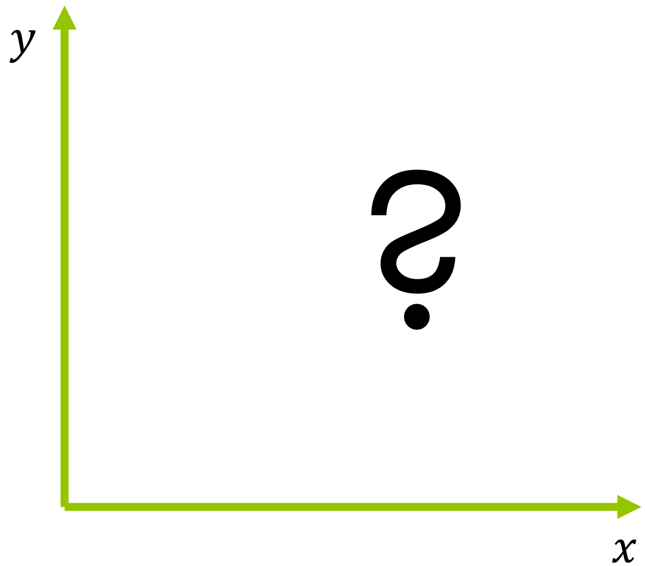


Image space

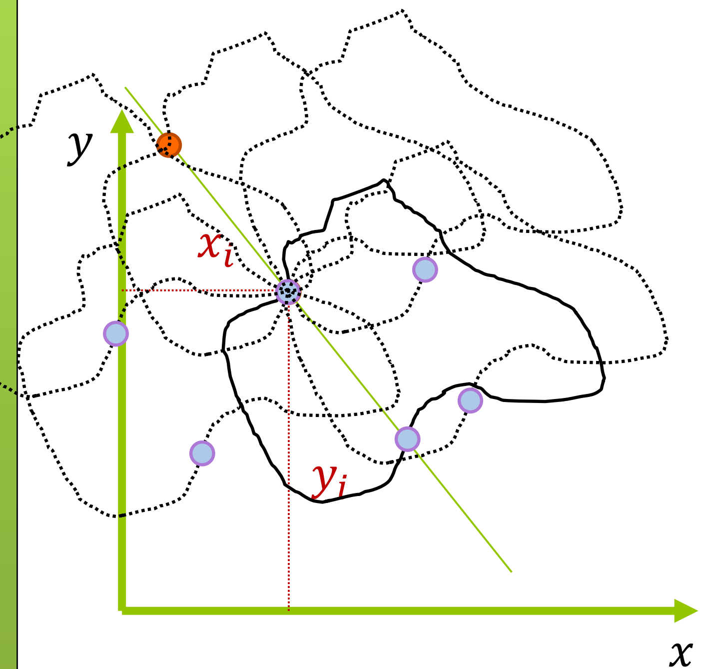


Hough space

HT for any given curve

[Merlin 1975]

in a specific orientation and size



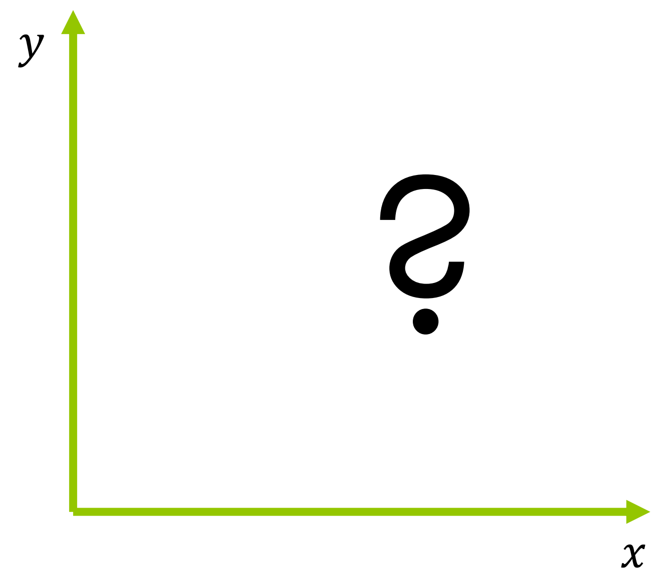


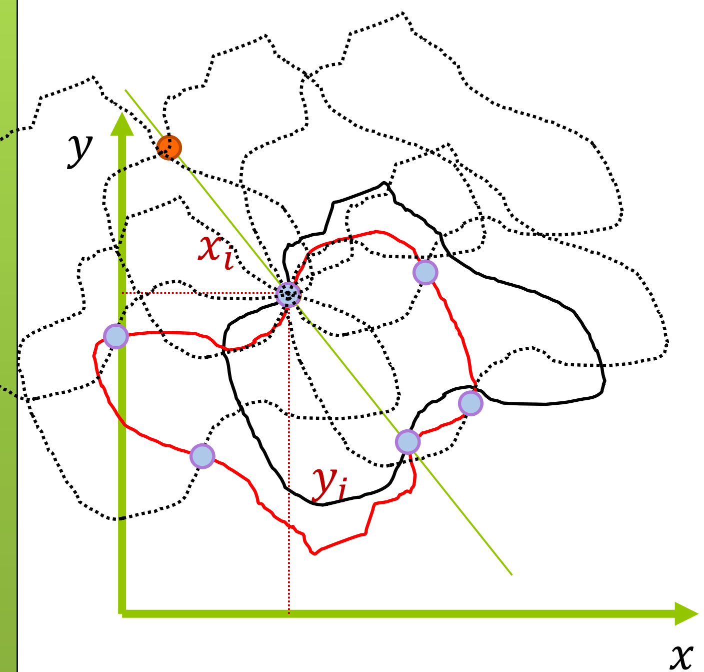
Image space

Hough space

HT for any given curve

[Merlin 1975]

in a specific orientation and size



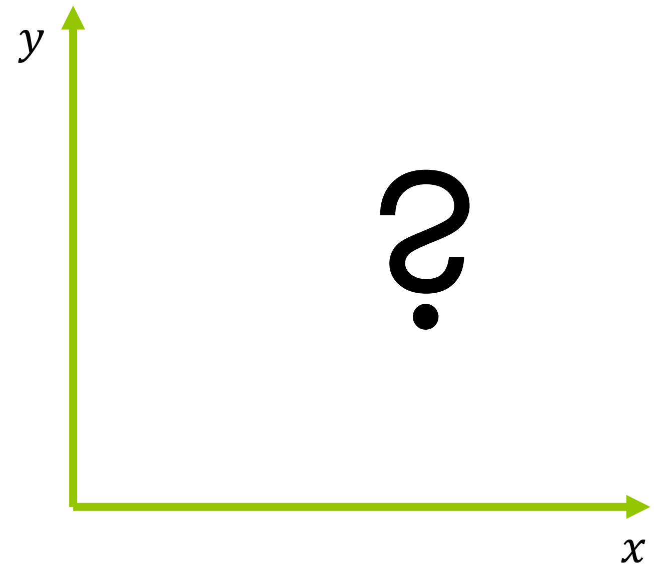


Image space

Hough space

HT for any given curve

[Merlin 1975]

in a specific orientation and size

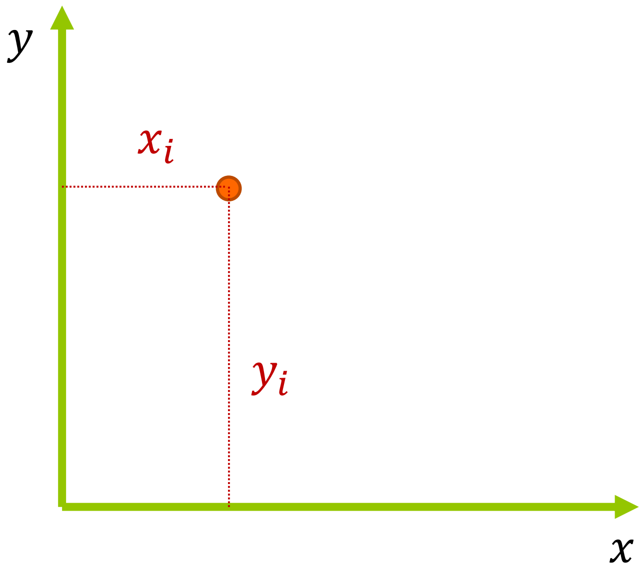
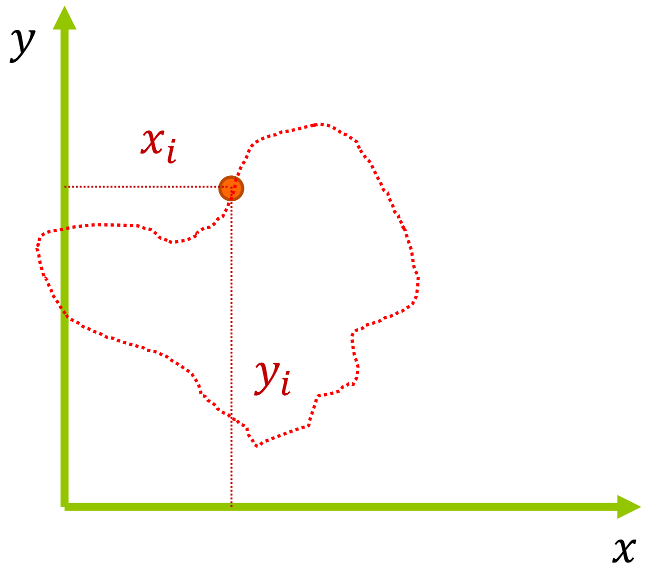


Image space



Hough space

HT for any given curve

[Merlin 1975]

in a specific orientation and size

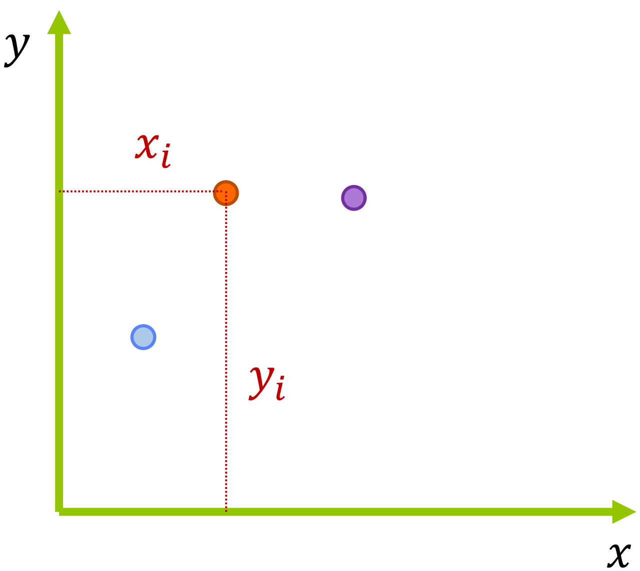
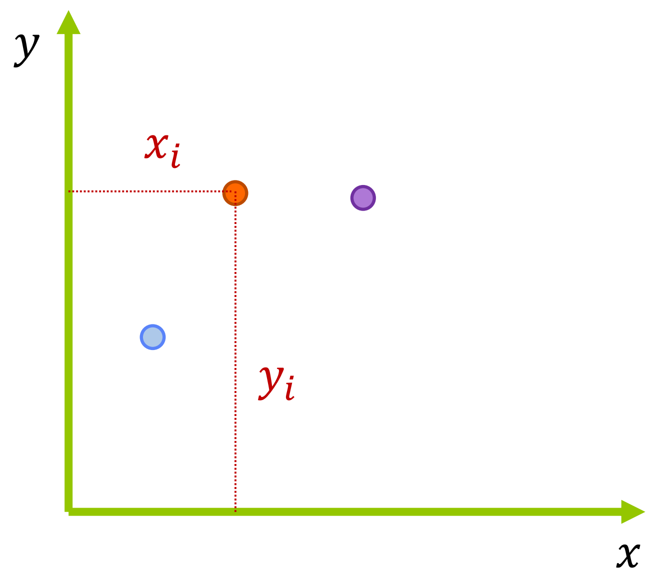


Image space



Hough space

HT for any given curve

[Merlin 1975]

in a specific orientation and size

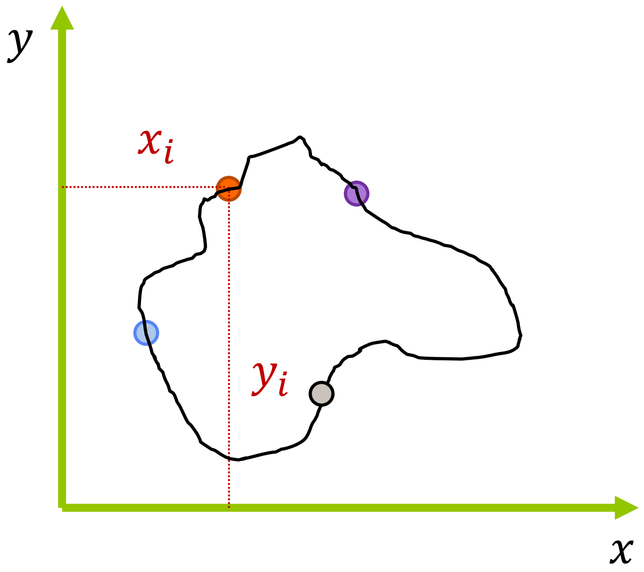
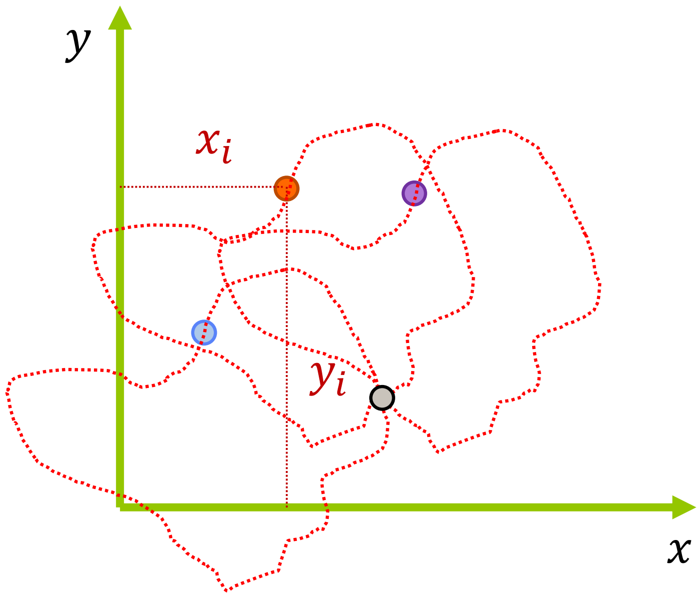


Image space



Hough space

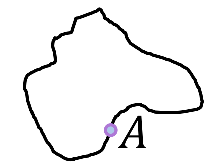
HT for any given curve

[Merlin 1975]

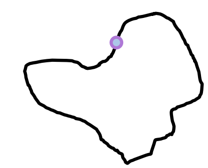
in a specific orientation and size

- **We can summarize the algorithm as follows:**

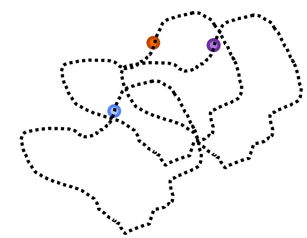
1. Mark a reference point called A in the given curve.



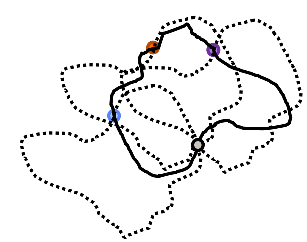
2. Rotate the given curve 180°.



3. Trace the rotated curve with A on each of the points of the picture.



4. Find the point where the maximum number of curves traced in Step 3 intersect.
5. The best fit to the given curve is the trace with A on the point found in Step 4.



- Most of the computation time is spent in Step 3.
  - the same curve for all the points → **parallel**

HT for any given curve

[Merlin 1975]

in a specific orientation and size

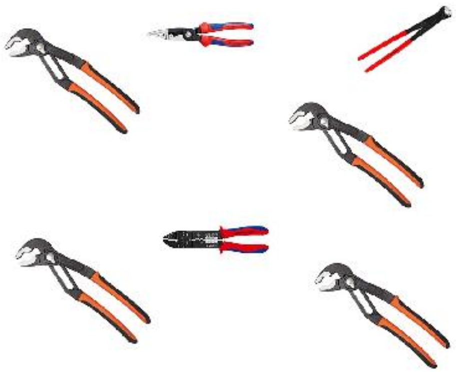
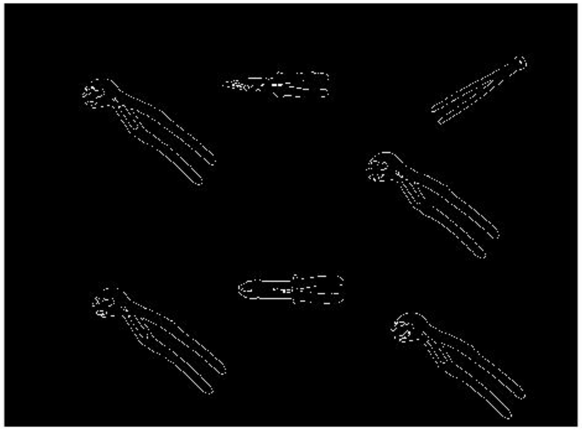
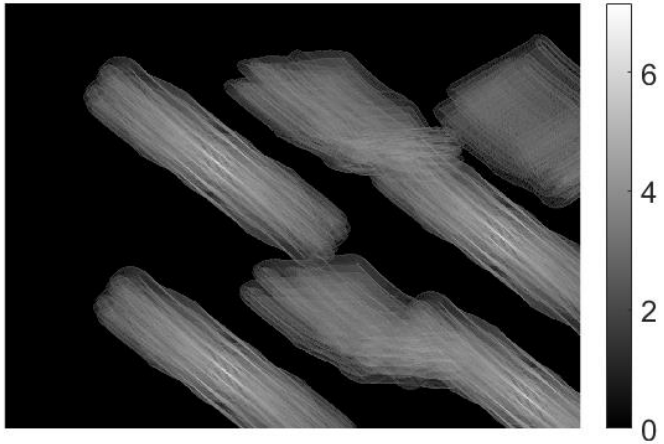
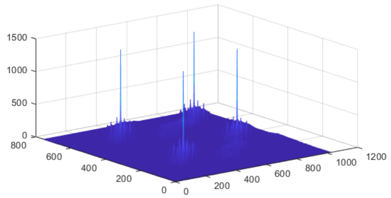


Image space

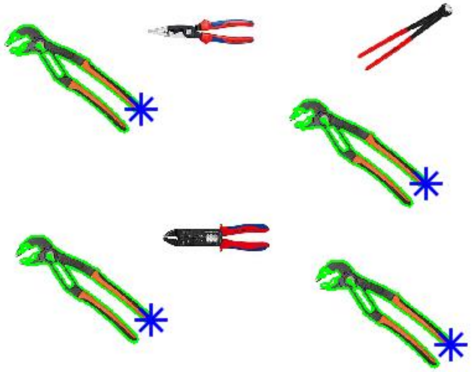


Thresholding & Morphological operations & Canny detector





Hough space (log)



Detected objects

Why not CONV or NCC?

- Fast Normalized Cross-Correlation (NCC)

John P. Lewis, “Fast Normalized Cross-Correlation” (1995).

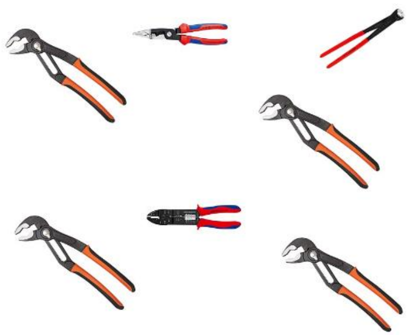
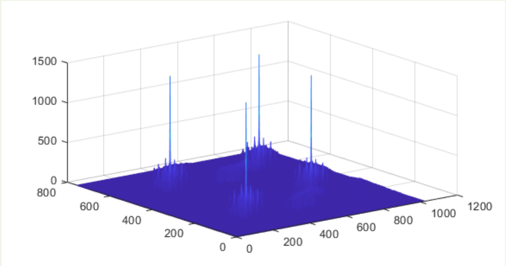
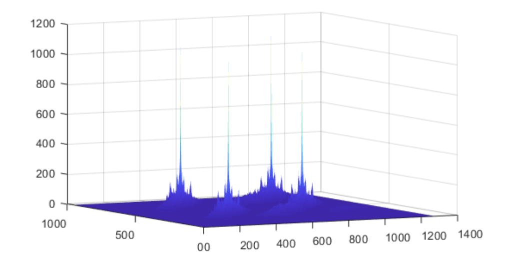
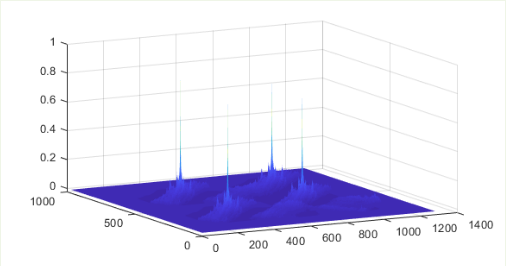


Image – 756x1024 px

Template – 215x254 px

Method	time [s]	output
HT Merlin 1975	0.1407	
CONV	0.0242	
NCC	0.2293	

HT for any given curve

[Merlin 1975]

in a specific orientation and size – with occlusion

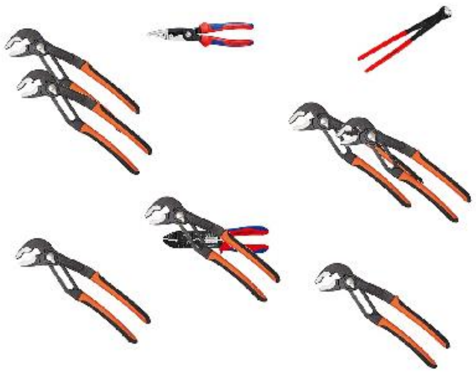
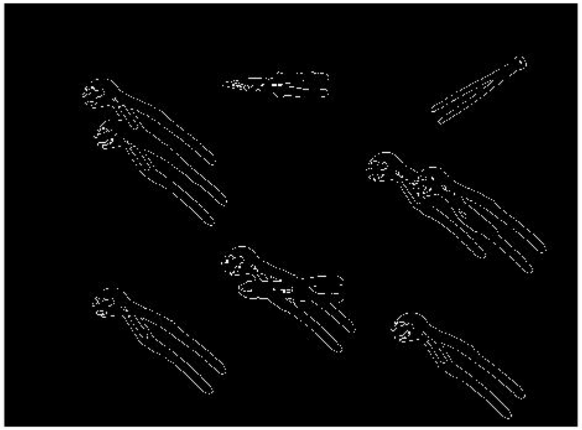
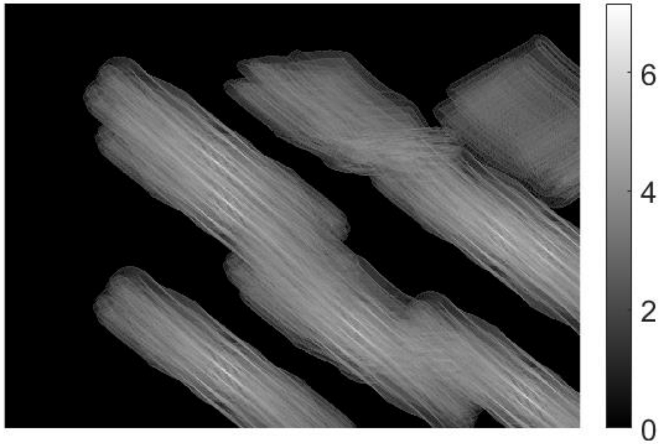


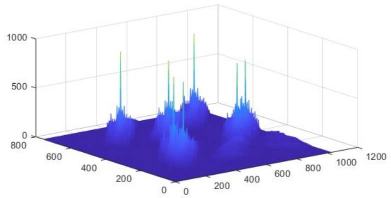
Image space

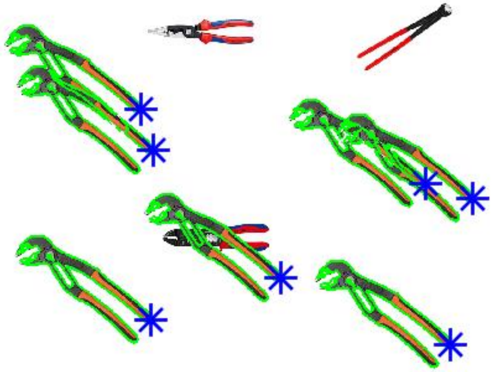


Thresholding & Morphological operations & Canny detector



Hough space (log)





Detected objects

Why not CONV or NCC?

- Fast Normalized Cross-Correlation (NCC)

John P. Lewis, “Fast Normalized Cross-Correlation” (1995).

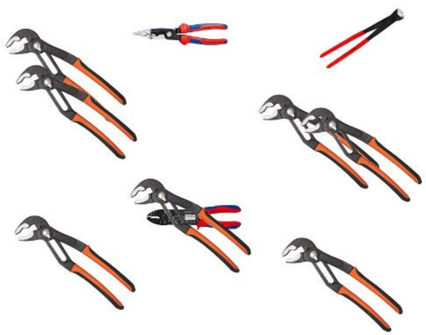
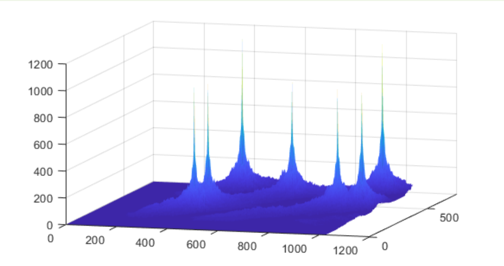
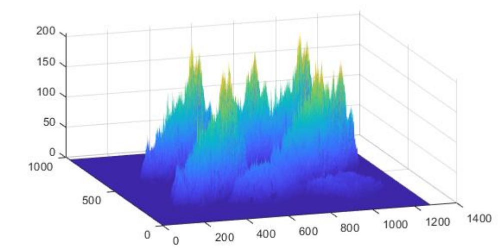
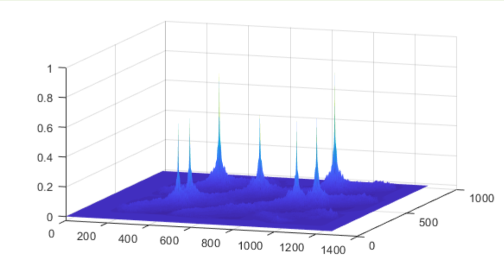


Image – 756x1024 px

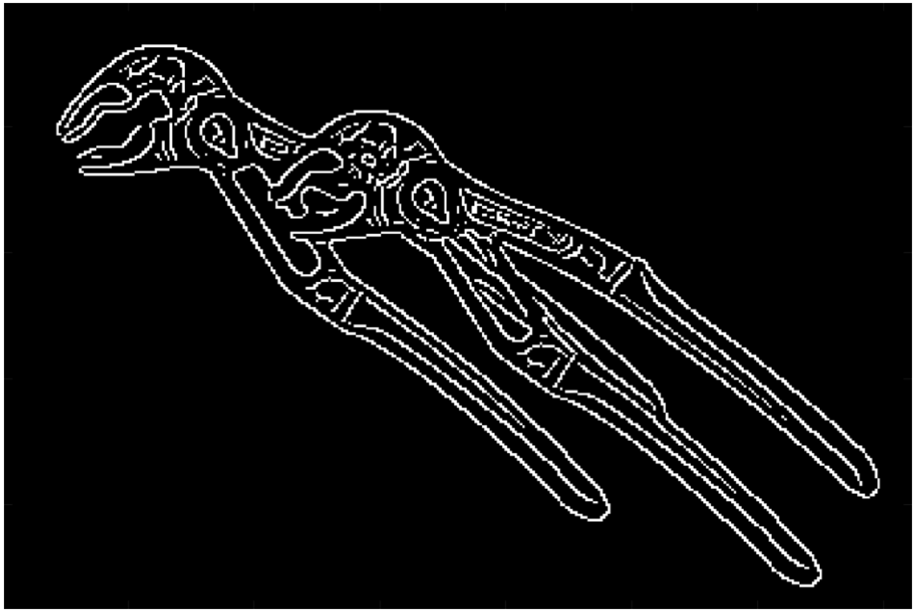
Template – 215x254 px

Method	time [s]	output
HT Merlin 1975	0.1407	
CONV	0.0242	
NCC	0.2293	

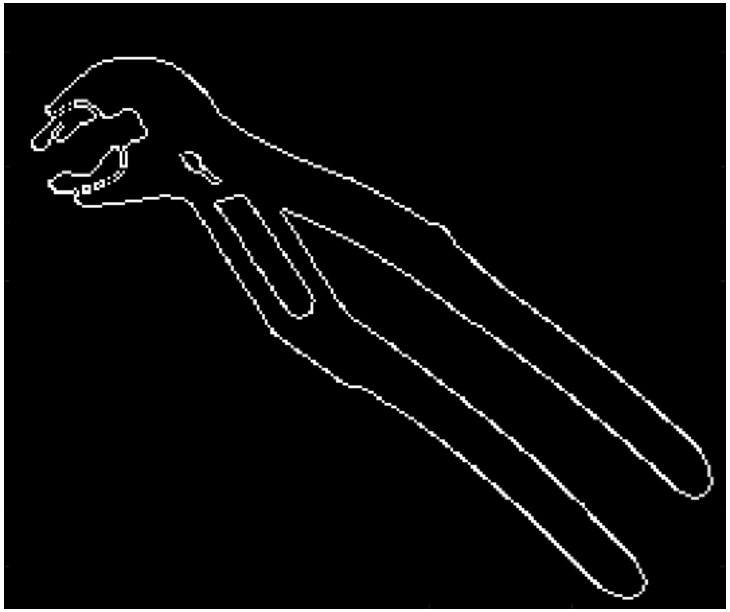
HT for any given curve

[Merlin 1975]

in a specific orientation and size – with occlusion
+ different edge detectors



detected edges in image



detected edges in template

HT for any given curve

[Merlin 1975]

in a specific orientation and size – with occlusion
+ different edge detectors

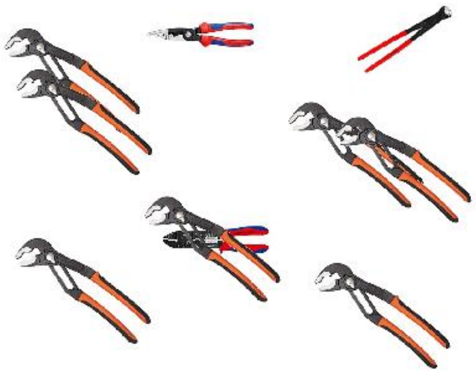
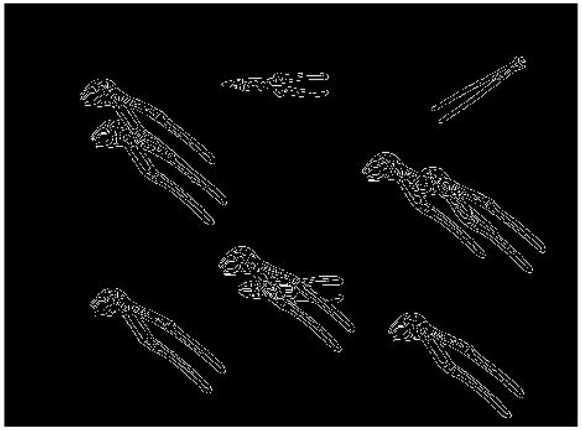
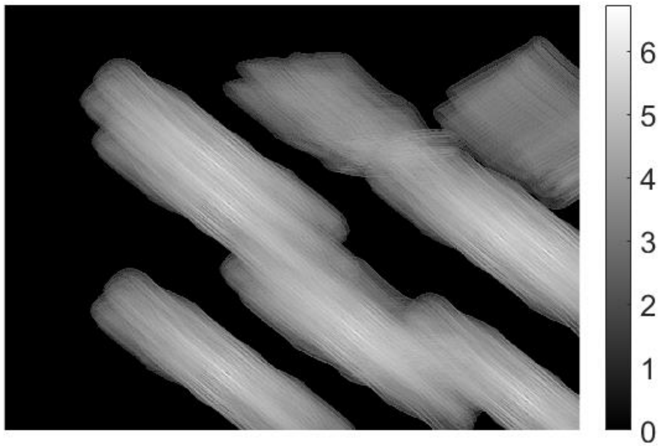


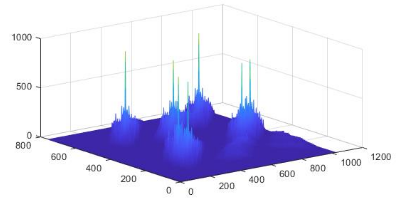
Image space

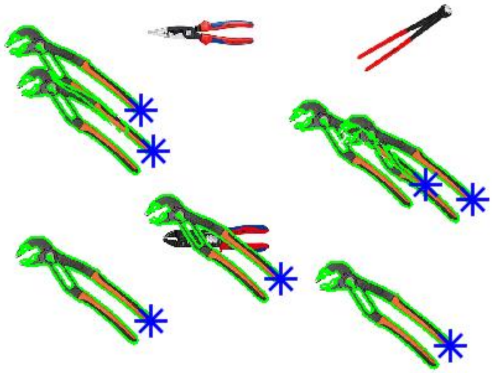


Thresholding & Morphological operations & Canny detector



Hough space (log)





Detected objects

HT for any given curve

[Merlin 1975]

in a specific orientation and size – with AWGN $\mathcal{N}(0,0.02)$

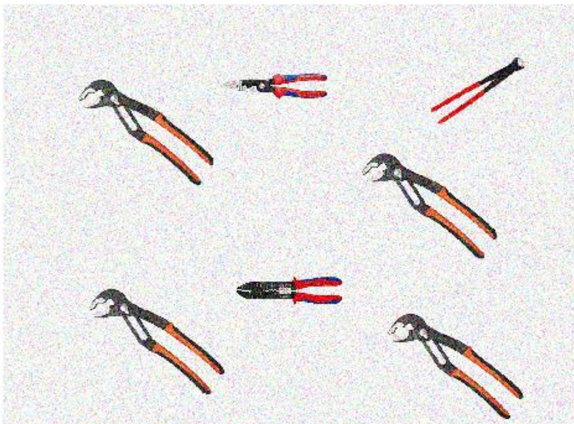
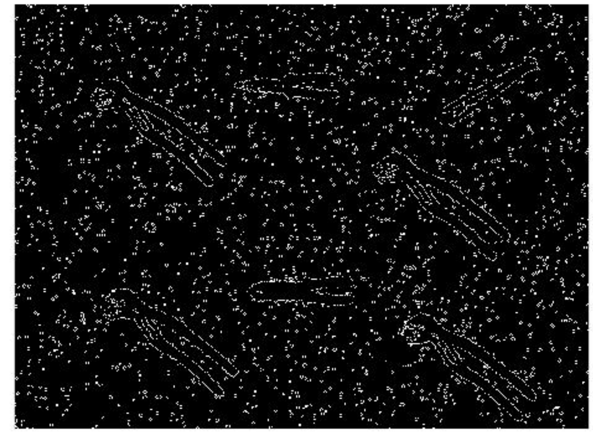
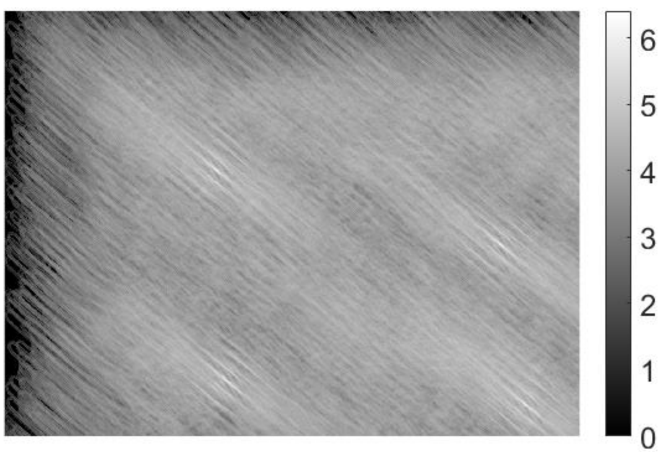


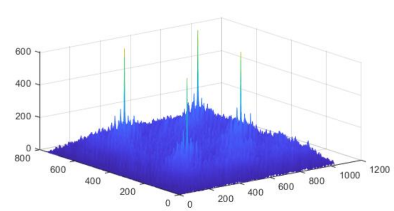
Image space

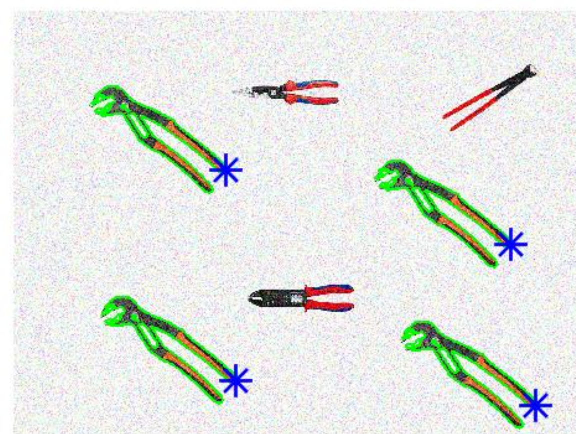


Thresholding & Morphological operations & Canny detector



Hough space (log)





Detected objects

HT for any given curve

in a specific orientation and size – with AWGN $\mathcal{N}(0,0.02)$

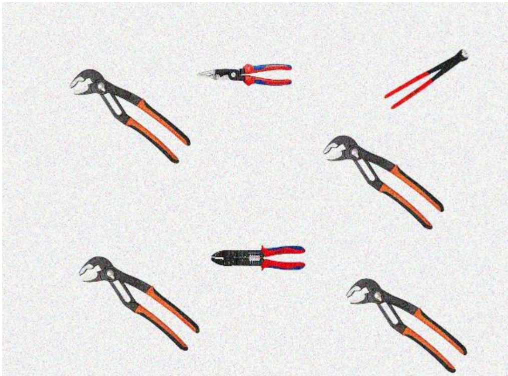


Image – 756x1024 px

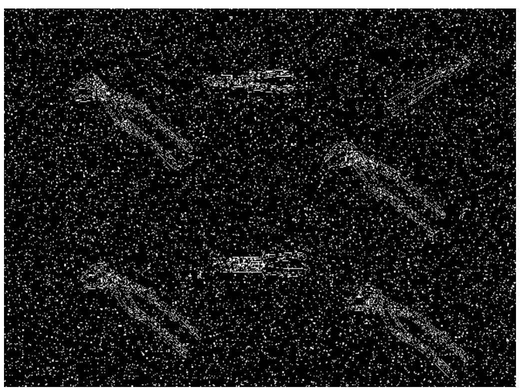
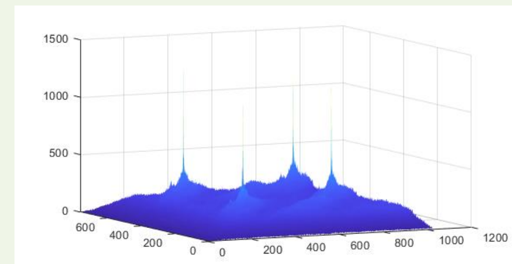
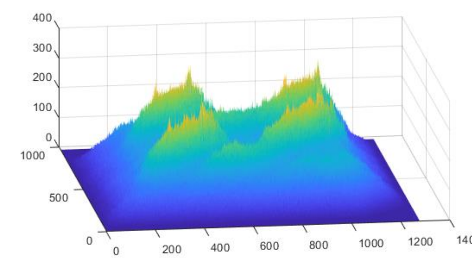
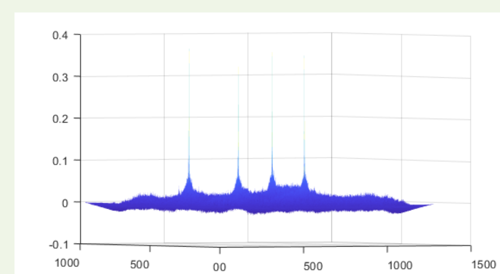


Image – 756x1024 px

Template – 215x254 px

Method	time [s]	output
HT Merlin 1975	1.2961	
CONV	0.0242	
NCC	0.2293	

Generalizing Hough Transform

D.H. Ballard, Generalizing the Hough transform to detect arbitrary shapes, Pattern Recognition, Volume 13, Issue 2, 1981.

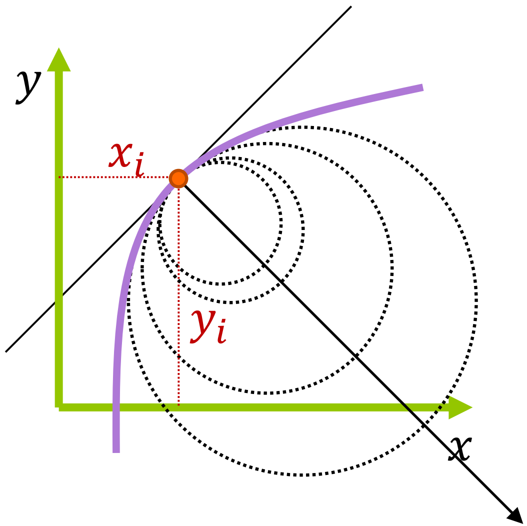
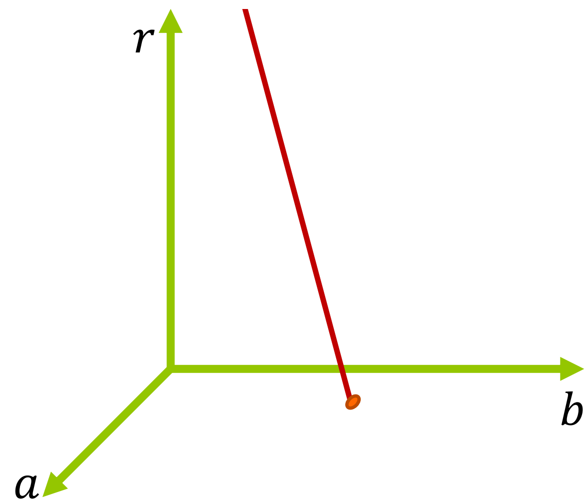
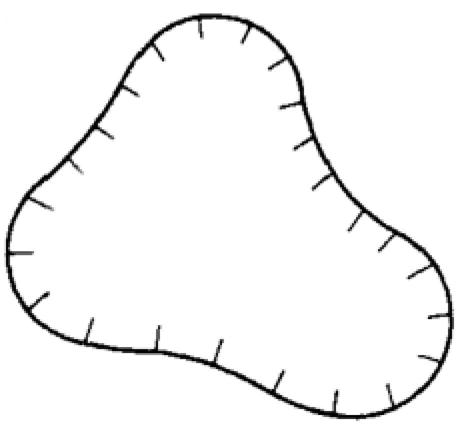
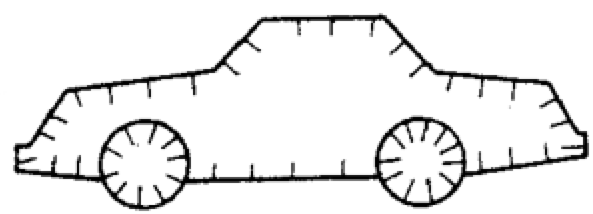


Image space



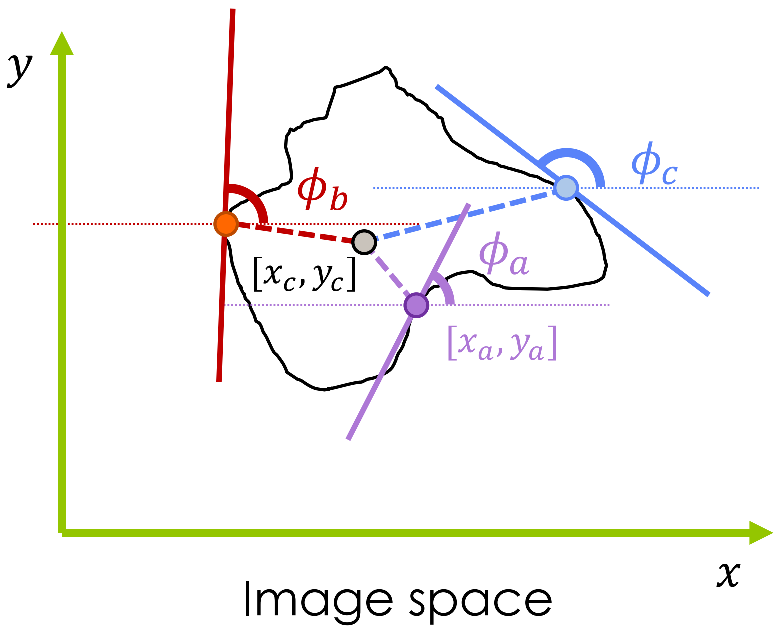
Hough space





Generalizing Hough Transform

[Ballard 1975]



R-table	
…	…
ϕ_a	$\boldsymbol{r}_1^a, \boldsymbol{r}_2^a, \ldots, \boldsymbol{r}_{m_a}^a$
…	…
ϕ_b	$\boldsymbol{r}_1^b, \boldsymbol{r}_2^b, \ldots, \boldsymbol{r}_{m_b}^b$
…	…
ϕ_c	$\boldsymbol{r}_1^c, \boldsymbol{r}_2^c, \ldots, \boldsymbol{r}_{m_c}^c$
…	…

$$\boldsymbol{centroid} = [x_c, y_c]$$

$$\boldsymbol{r}_k^i = [x_k^i, y_k^i]$$

$$x_c = x_j + x_k^i$$

$$y_c = y_j + y_k^i$$

Generalizing Hough Transform

[Ballard 1975]

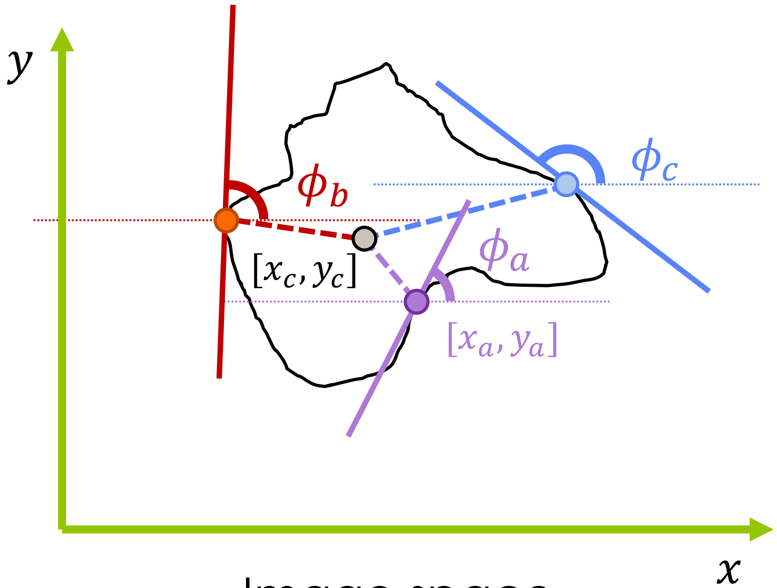
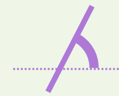
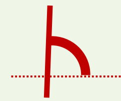
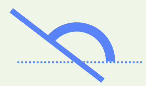
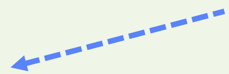


Image space

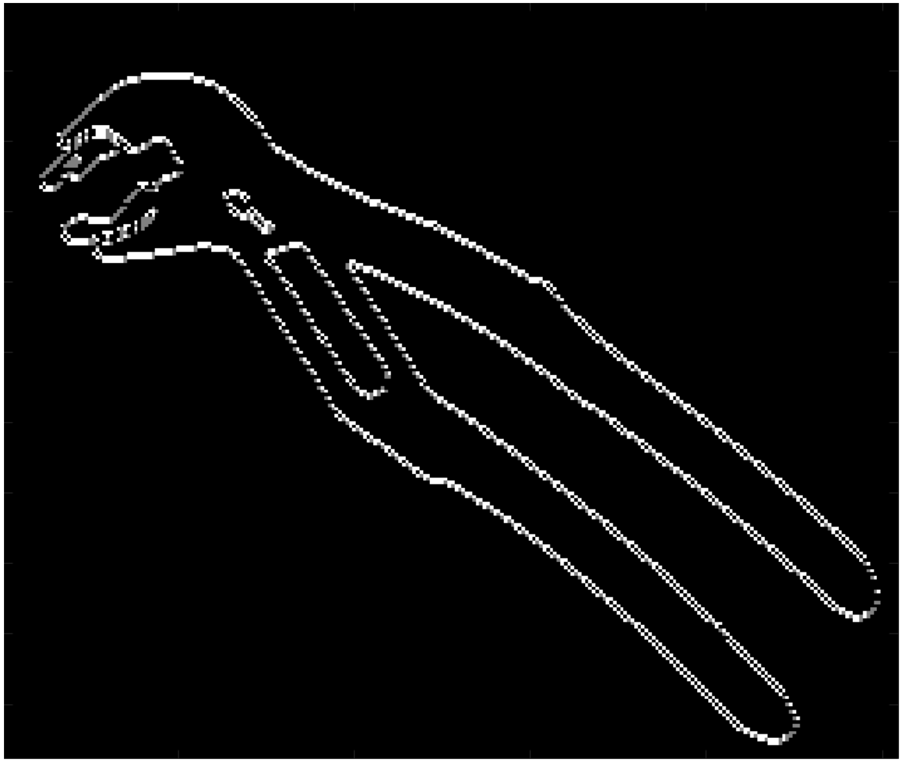
R-table	
...	...
	 ...
...	...
	 ...
...	...
	 ...
...	...

Generalizing Hough Transform

[Ballard 1975]

$$\tan \phi = \frac{dy}{dx}$$

```
Dy=imfilter(double(Img),[1; -1],'same');
Dx=imfilter(double(Img),[1  -1],'same');
Fi=mod(atan2(Dy,Dx)+pi, pi);
```



Generalizing Hough Transform [Ballard 1975]

in a specific orientation and size

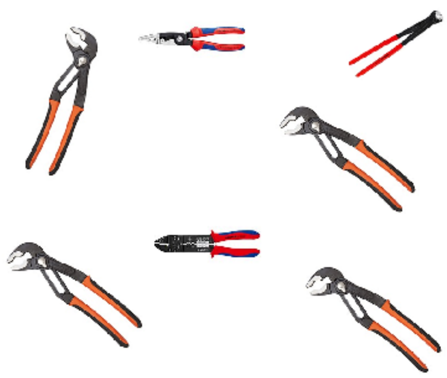
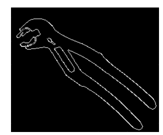
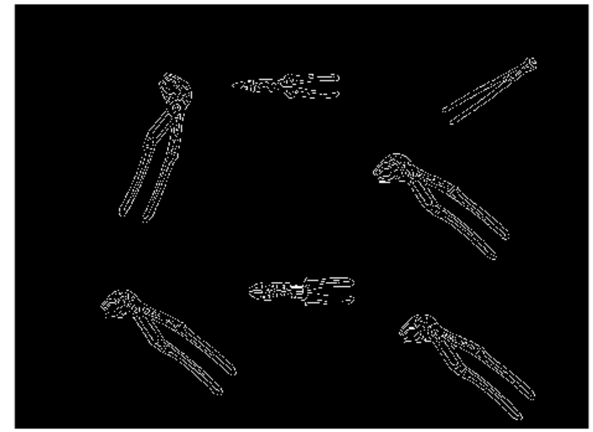
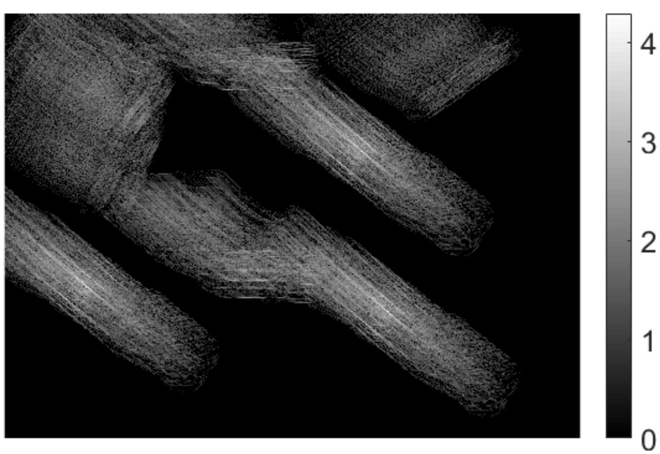


Image space

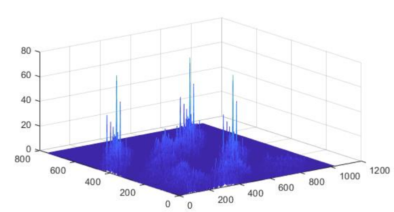


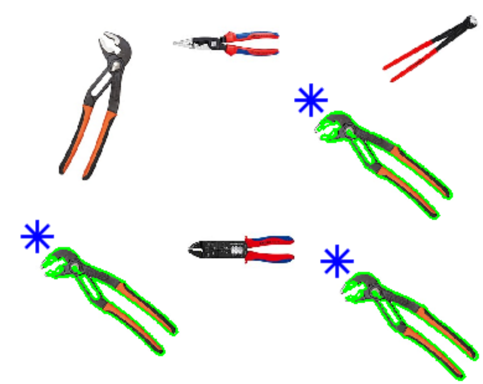


Thresholding & Morphological operations & Canny detector



Hough space (log)





Detected objects

Generalizing Hough Transform [Ballard 1975]

in an arbitrary orientation and certain range of size

- **Scale by** s: $T_s[R(\phi)] = sR(\phi)$

  all vectors scaled by s

- **Rotation by** θ: $T_\theta[R(\phi)] = Rot\{R[(\phi - \theta) mod 2\pi], \theta\}$

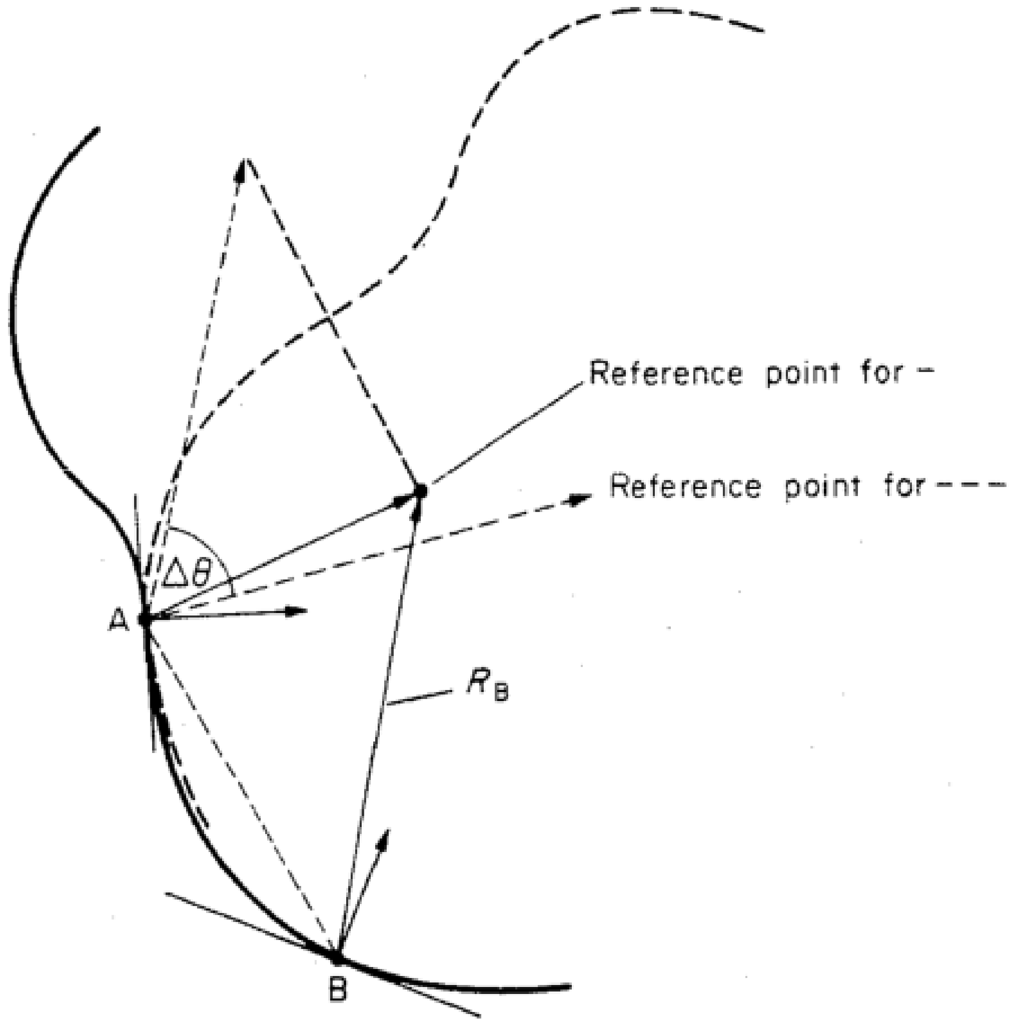
  all the indices are incremented by $-\theta \; mod 2\pi$

  the appropriate vectors $\boldsymbol{r}$ are found and then rotated by θ

Generalizing Hough Transform

[Ballard 1975]

in an arbitrary orientation and certain range of size



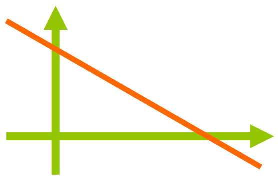
Generalizing Hough Transform [Ballard 1975]

in an arbitrary orientation and certain range of size

DEMO

Application - Detection of vanishing points and lines

T. Tuytelaars, M. Proesmans and L. Van Gool, "The cascaded Hough transform," Proceedings of International Conference on Image Processing, Santa Barbara, CA, 1997, pp. 736-739 vol.2.



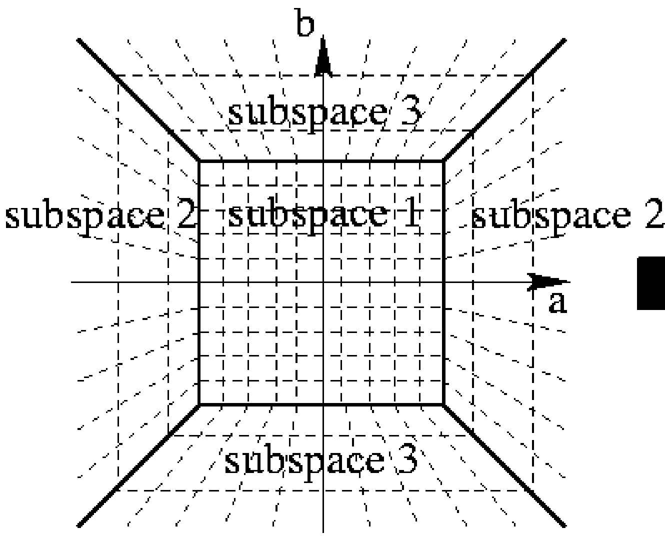
Slope-intercept: $ax + b + y = 0$
lines corresponds to points
point corresponds to lines

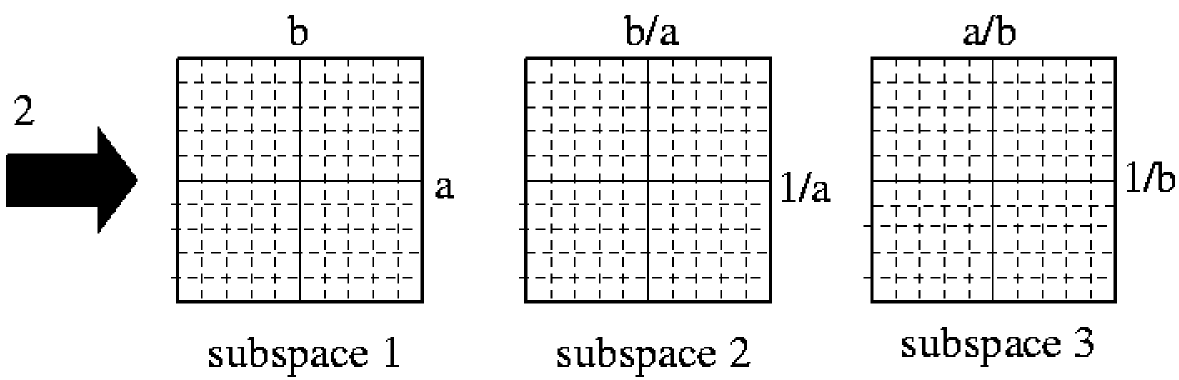
duality between lines and points

$|a| \le 1$, $|b| \le 1$

$|a| > 1$, $|b| \le |a|$

$|a| \le |b|$, $|b| > 1$





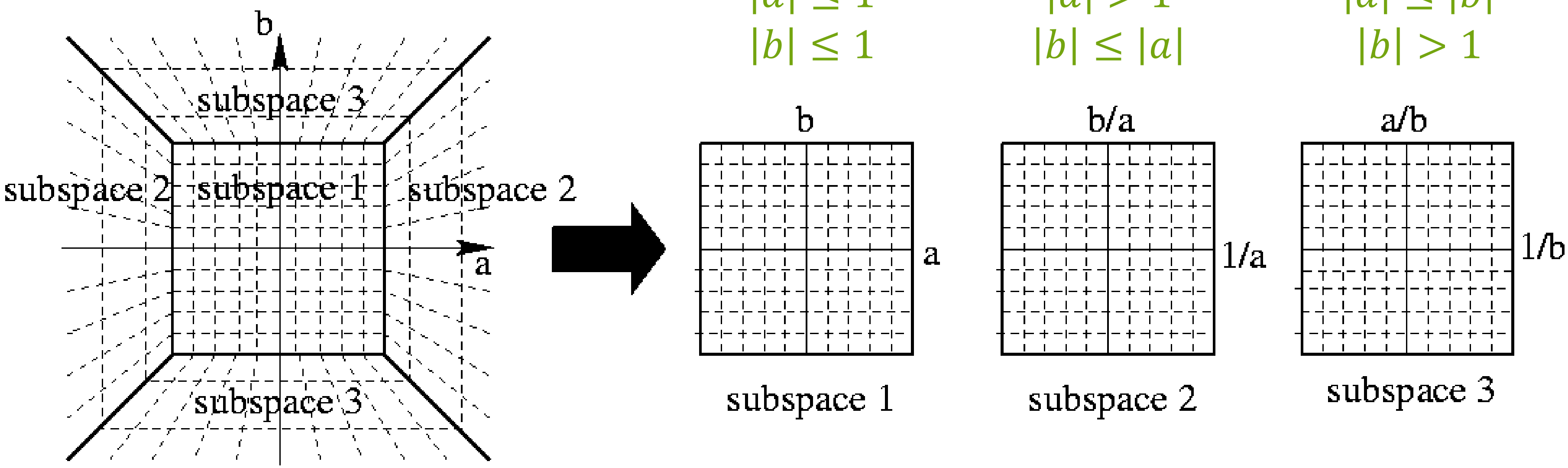
inhomogeneous discretization of the unbounded parameter space

(a, b)-space is split into three subspaces on the interval [−1,1]

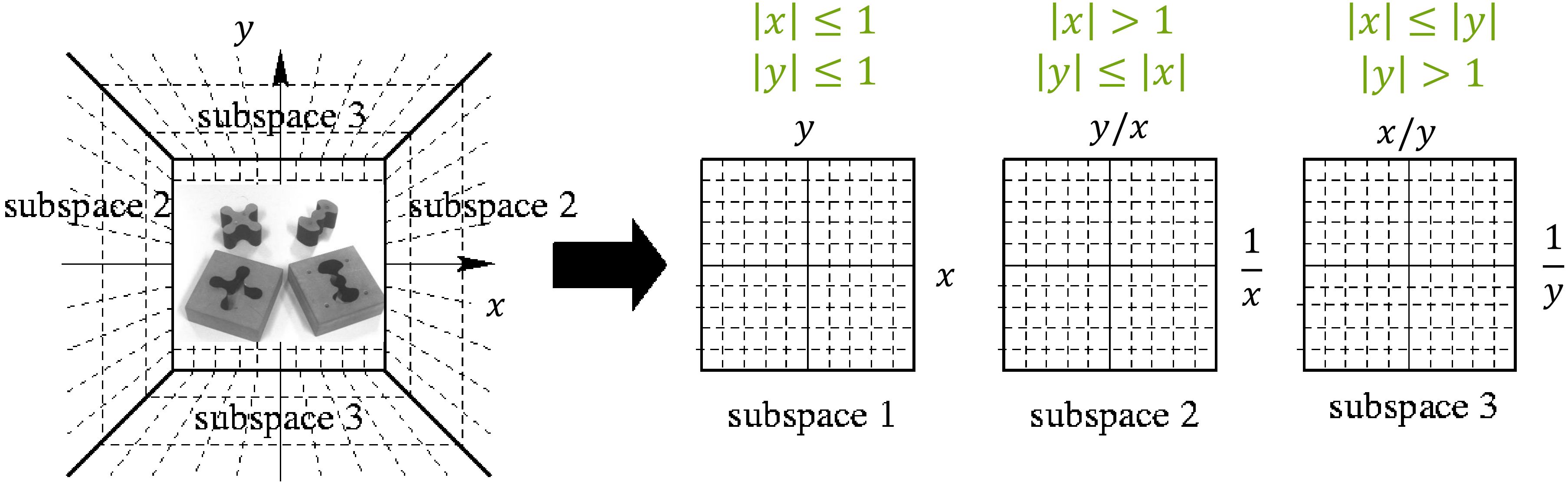
[Tuytelaars 1997]

Application - Detection of vanishing points and lines

$|a| \leq 1$, $|b| \leq 1$ | $|a| > 1$, $|b| \leq |a|$ | $|a| \leq |b|$, $|b| > 1$

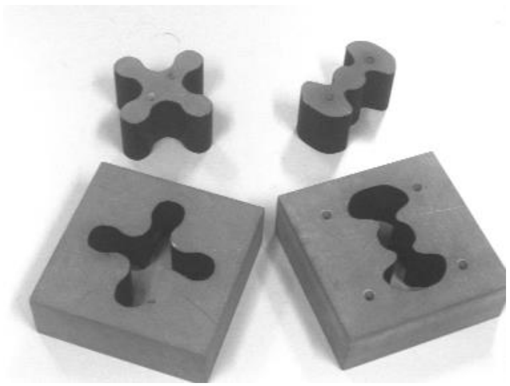


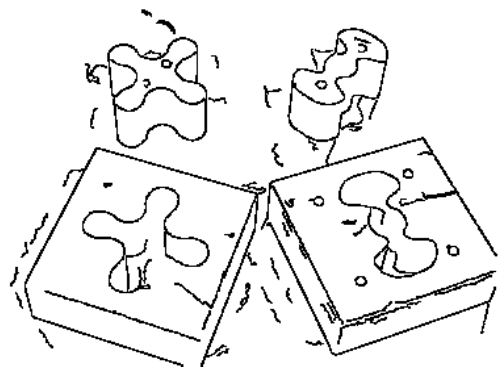
$|x| \leq 1$, $|y| \leq 1$ | $|x| > 1$, $|y| \leq |x|$ | $|x| \leq |y|$, $|y| > 1$



[Tuytelaars 1997]

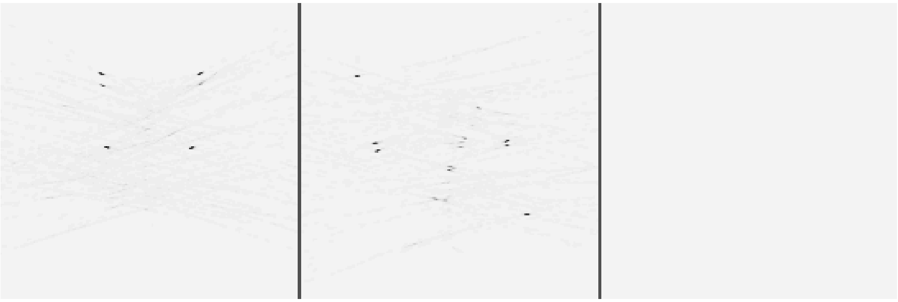
Application - Detection of vanishing points and lines





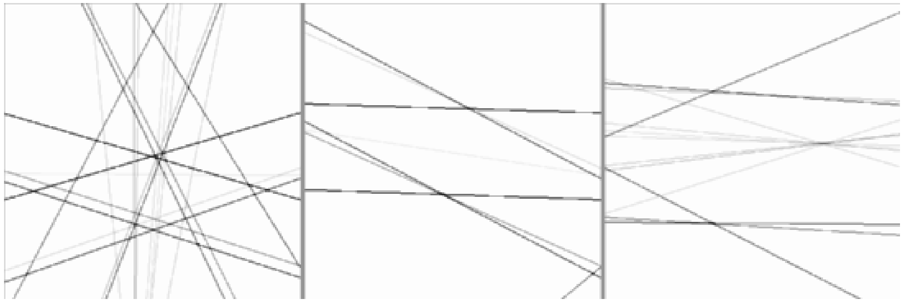
1.HT

straight lines



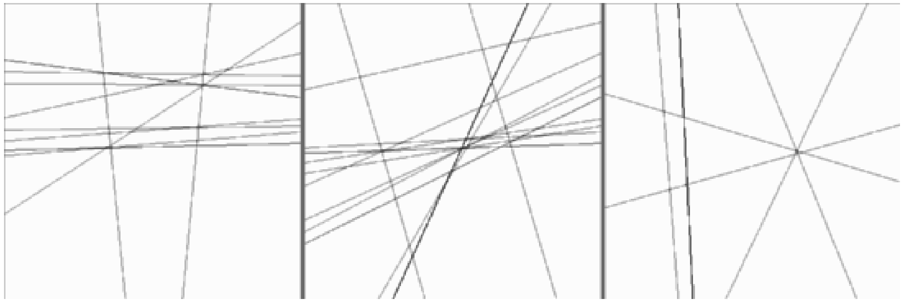
2.HT

vanishing points



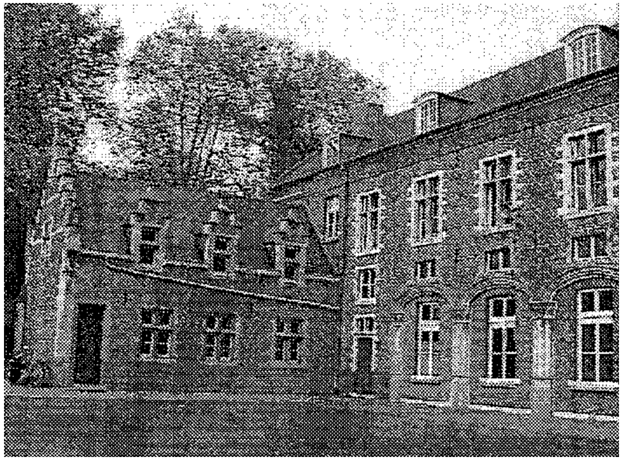
3.HT

vanishing lines

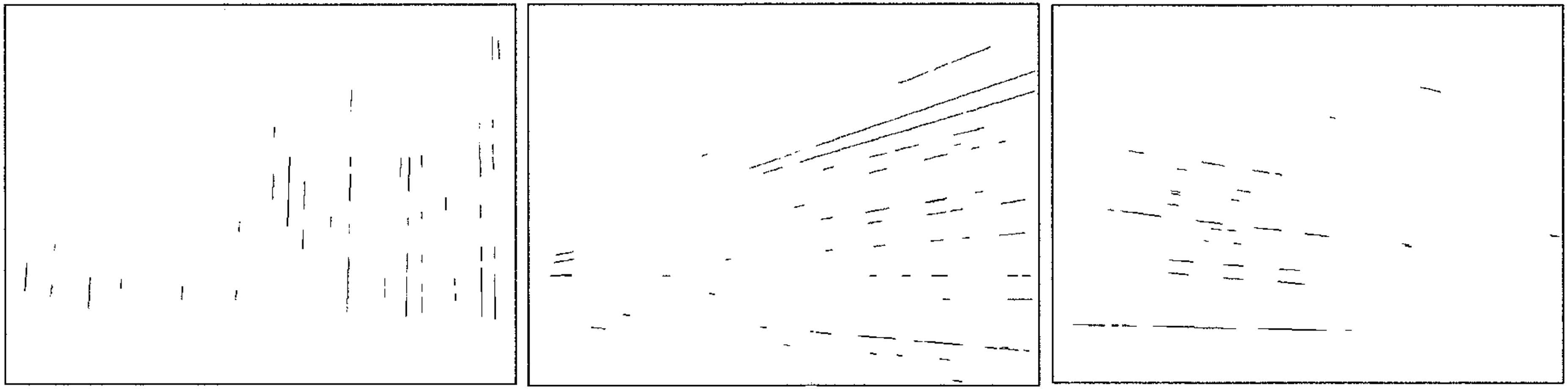


Application - Detection of vanishing points and lines

[Tuytelaars 1997]



Lines belonging to one of the three main directions



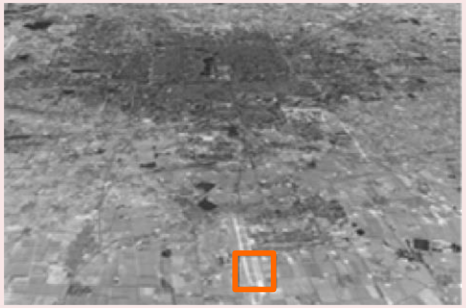
Application - Detection of vanishing points and lines

HT is used to find vanishing point and rectify building image

https://github.com/link2xt/fht/blob/master/README.md

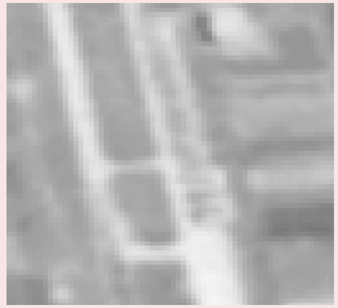
Application - Image matching by GHT

Image source



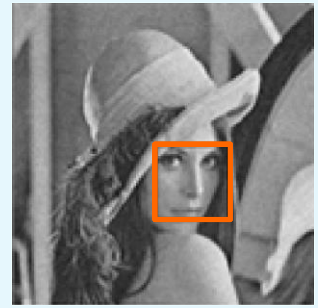
800 x 600

Image template



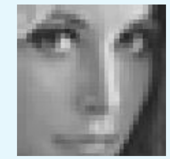
64 x 64

Image source



256 x 256
with noise

Image template



64 x 64

Method	**time [ms]**	**time [ms]**
GHT	40.1	5.0
Sum of absolute difference (SAD)	12107.4	1151.7
normalized cross-correlation NCC	26658.3	2543.6

Qiang Li and Bo Zhang, *Image matching under generalized hough transform*, IADIS AC (2005).

Thank you for attention!

ZOI – UTIA

Adam Novozámský (novozamsky@utia.cas.cz)