SAIR
Spatial AI & Robotics Lab

CSE 473/573-A

W5: HOUGH TRANSFORM & ALIGNMENT

Chen Wang
Spatial AI & Robotics Lab
Department of Computer Science and Engineering

University at Buffalo The State University of New York

Content

- Hough Transform
  - Line Parameterization,
  - Line Detection,
- Alignment
  - Homography
  - Interpolation
  - Stitching, Panorama

SAIR

Spatial AI & Robotics Lab

IMAGE PROCESSING

Hough Transform

Hough Transform

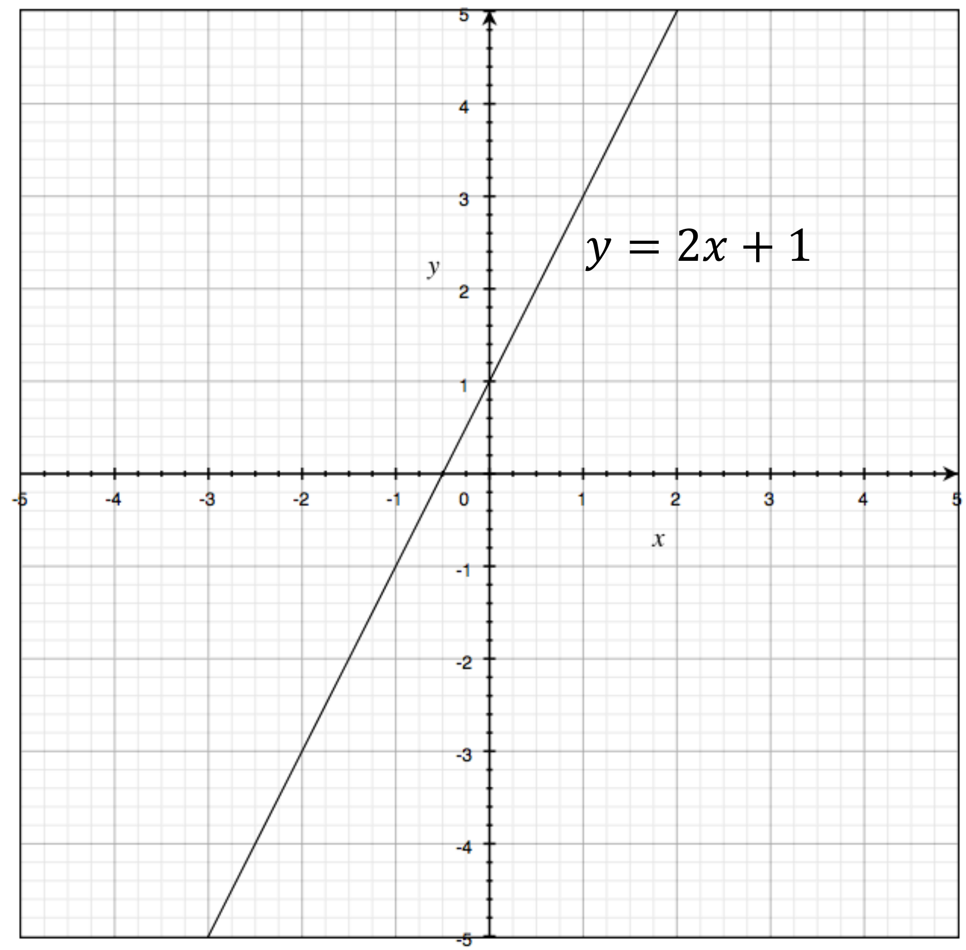
- Hough Transform can detect basic shapes
  - Detect points/edges → Find shapes.
  - Lines, Circles, etc.
- Line parameterizations
  - Slope intercept form
  - Double intercept form
  - Normal Form

SAIR
Spatial AI & Robotics Lab

Slope intercept form

$$y = mx + b$$

slope y-intercept

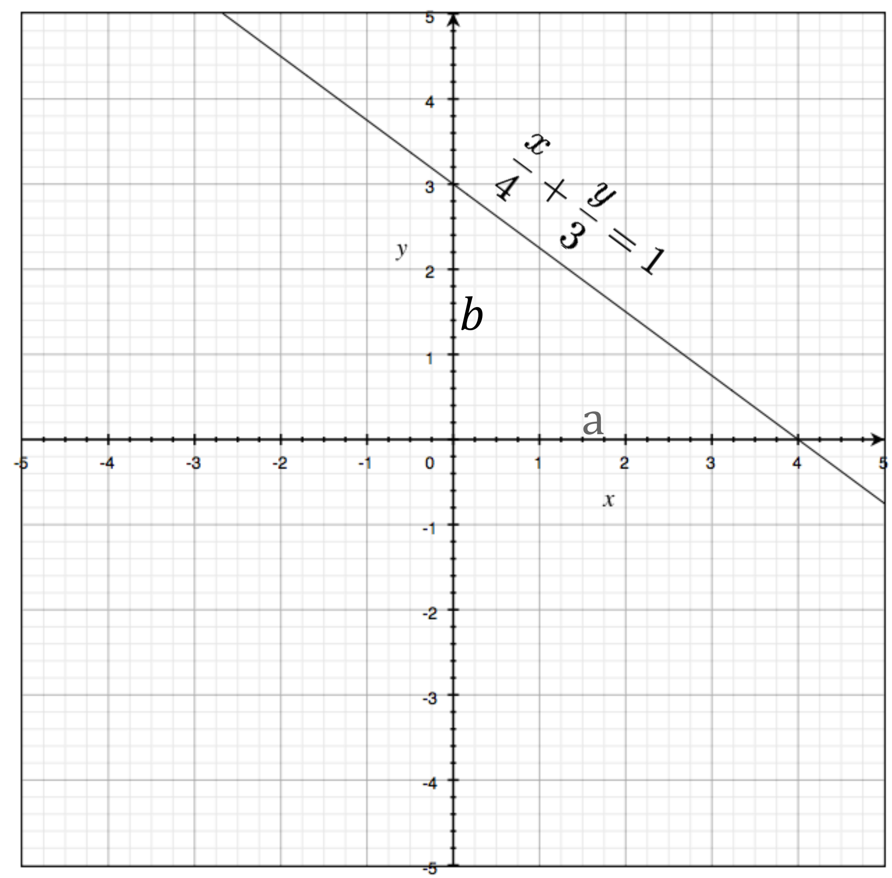


Double intercept form

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

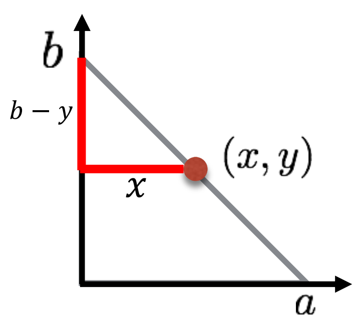
x-intercept

y-intercept



Derivation:

(Similar Triangles)



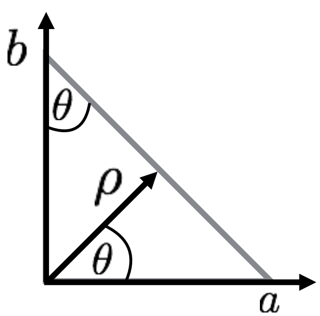
$$\frac{x}{a} = \frac{b-y}{b}$$

What are a and b?

Normal Form

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

Derivation:

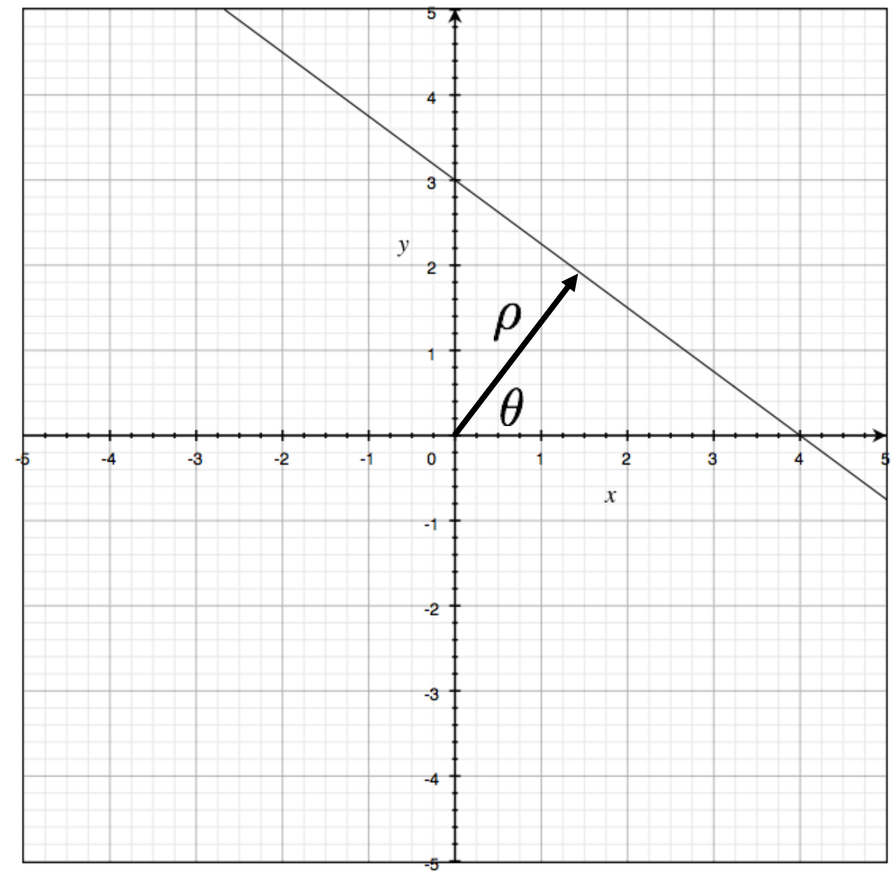


$$a = \frac{\rho}{\cos\theta}$$

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

plug into:

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



What are ρ and θ ?

S A I R
Spatial AI & Robotics Lab

Hough Transform

- Slope intercept form
- Normal Form

Image and parameter space

variables

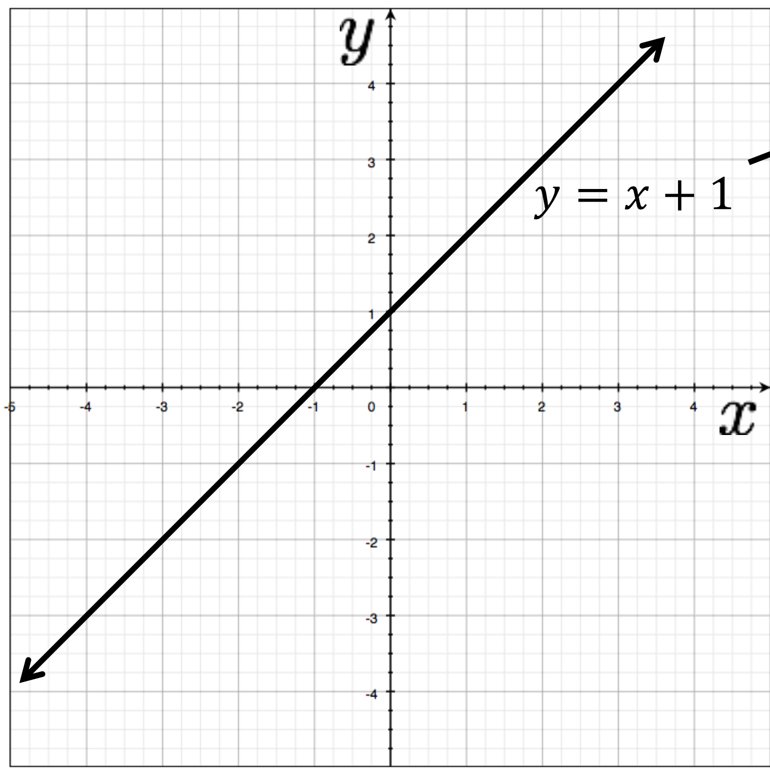
$$y = mx + b$$

parameters

variables

$$b = -xm + y$$

parameters



$m = 1$
$b = 1$

a line becomes a point

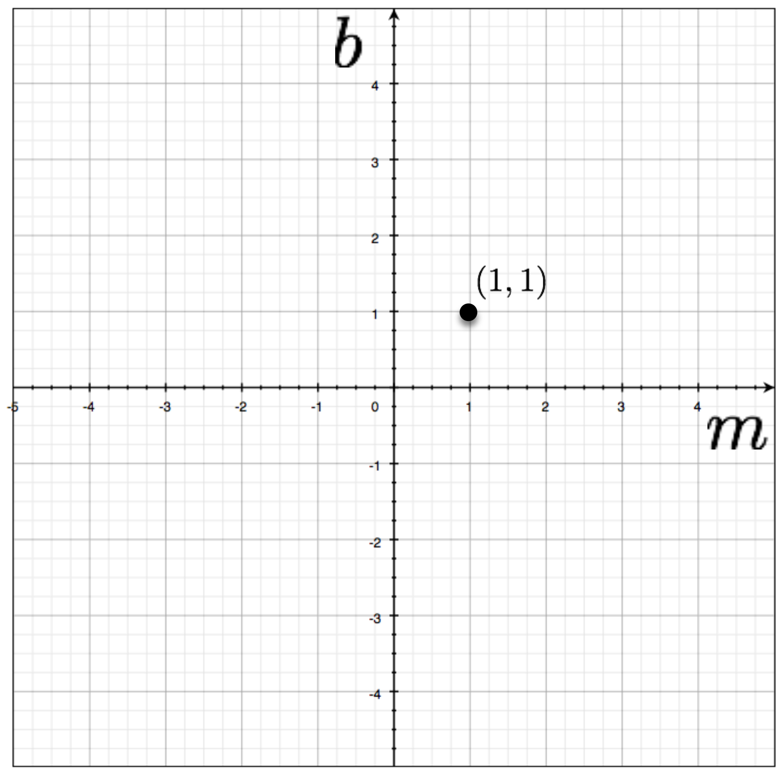


Image space

Parameter space

Image and parameter space

variables

$$y = mx + b$$

parameters

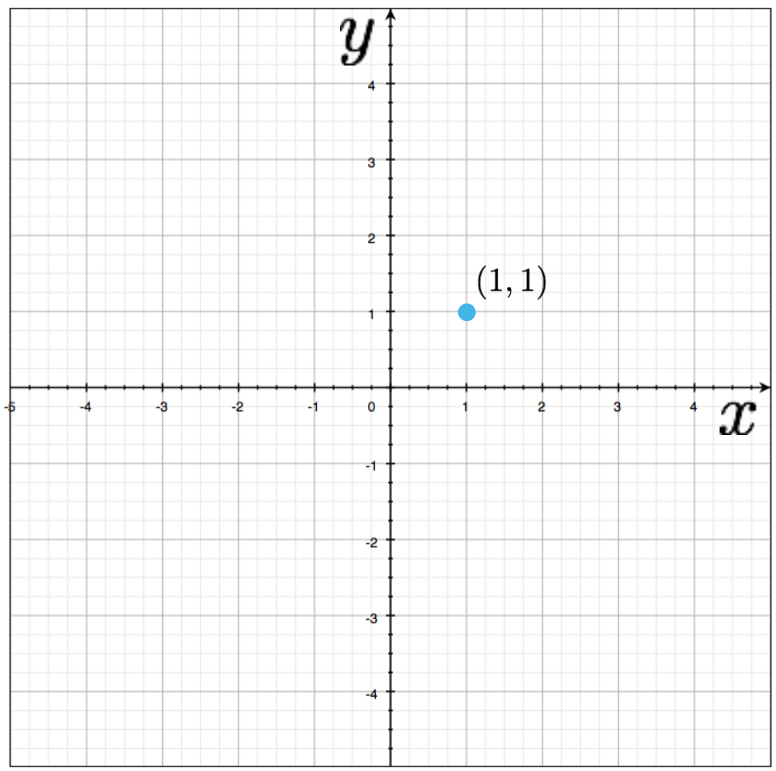


Image space

What would a point in image space become in parameter space?

S A I R
Spatial AI & Robotics Lab

Image and parameter space

variables

$$y = mx + b$$

parameters

variables

$$b = -xm + y$$

parameters

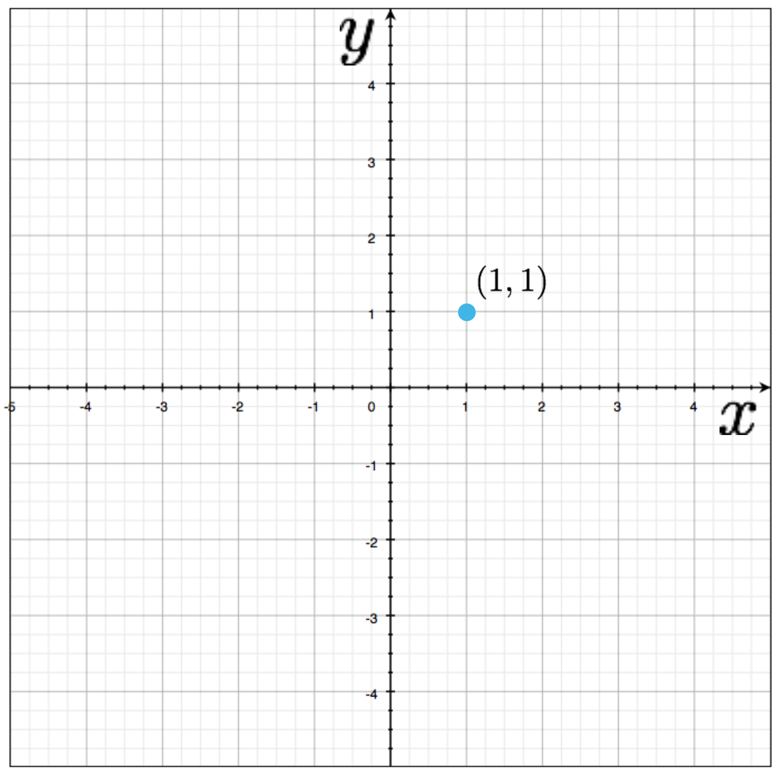
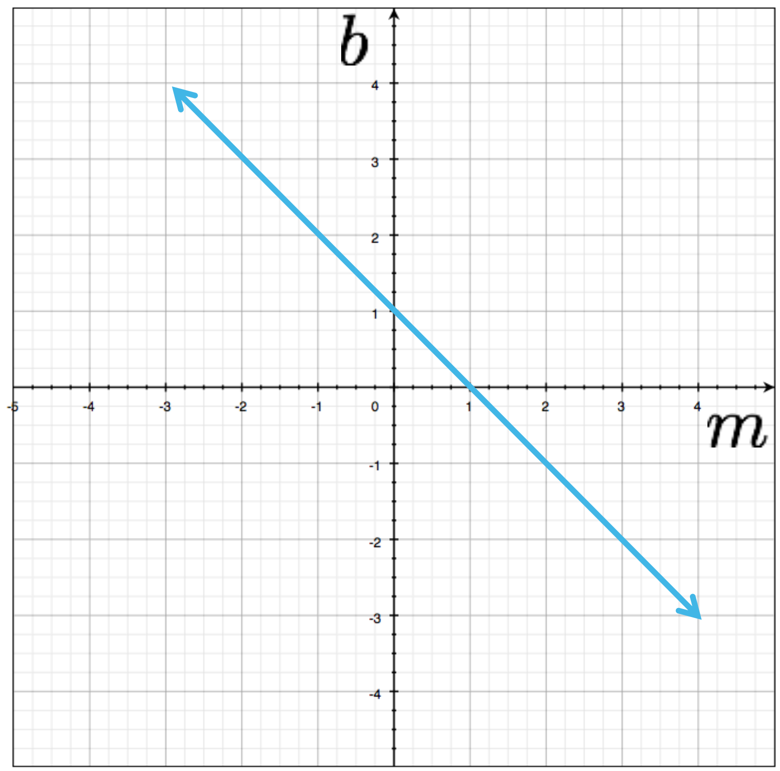


Image space

a point becomes a line



Parameter space

SAIR Spatial AI & Robotics Lab

Image and parameter space

variables

$$y = mx + b$$

parameters

variables

$$b = -xm + y$$

parameters

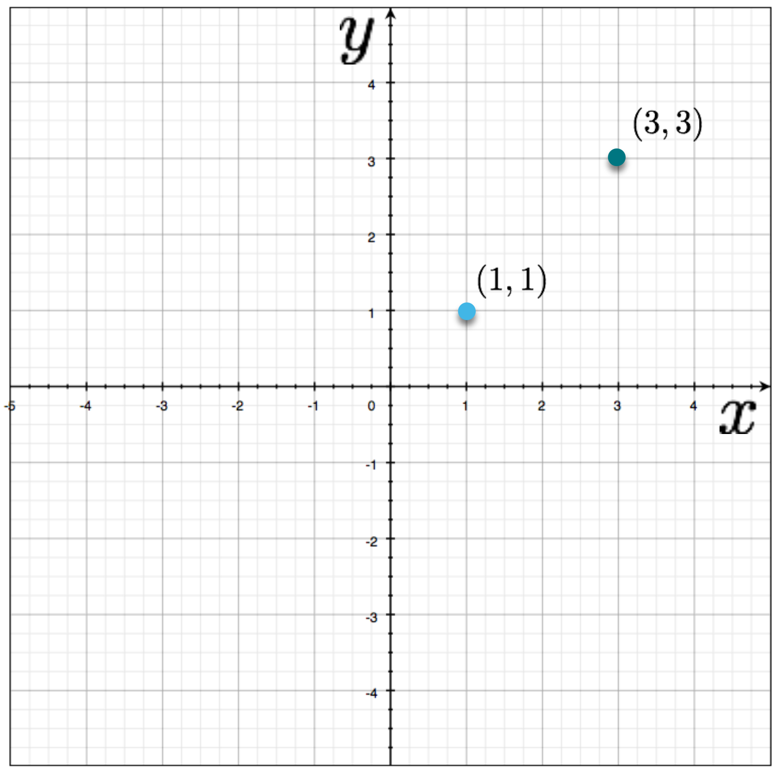
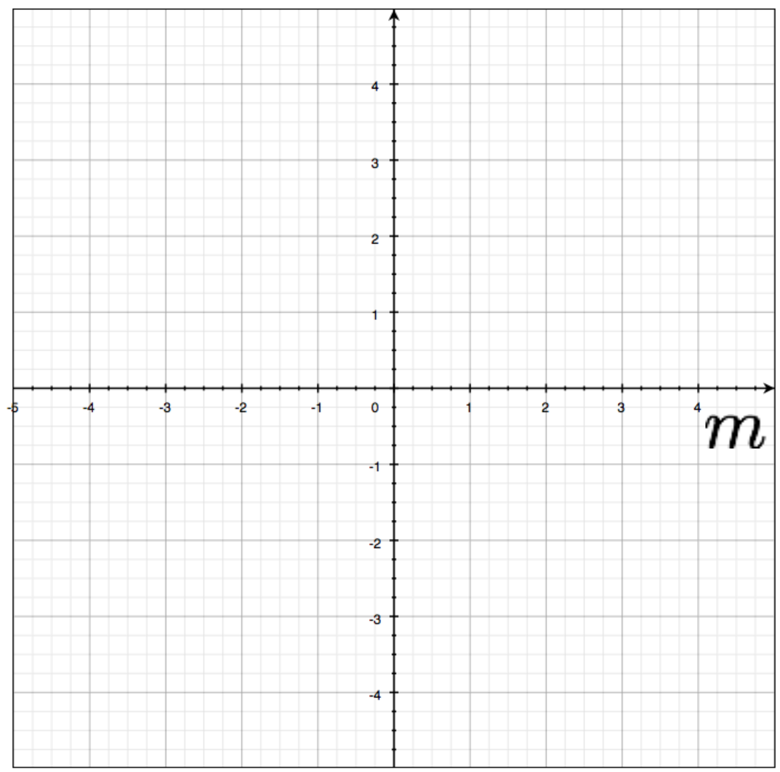


Image space

two points become ?



Parameter space

Image and parameter space

variables

$$y = mx + b$$

parameters

variables

$$b = -xm + y$$

parameters

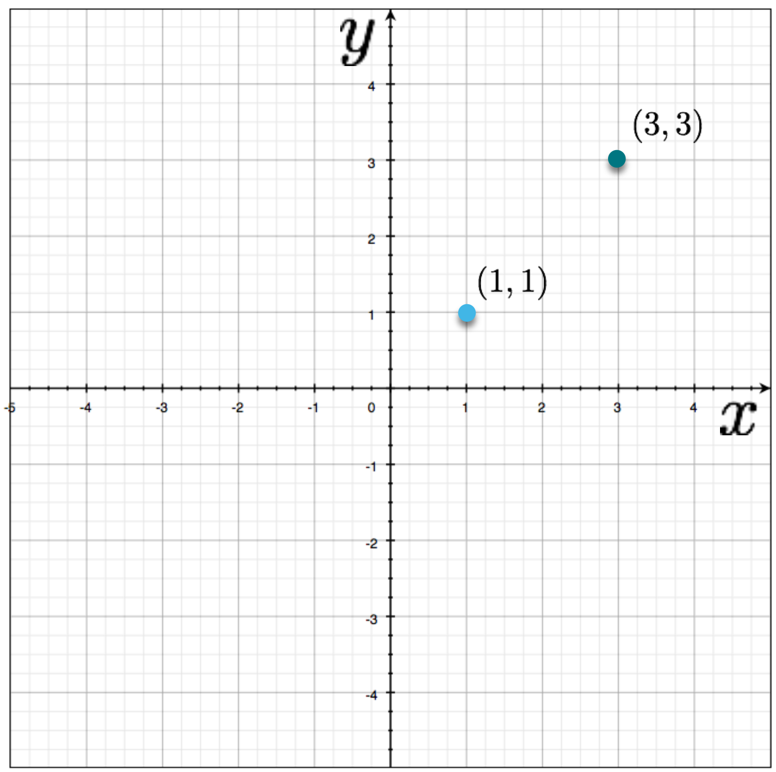
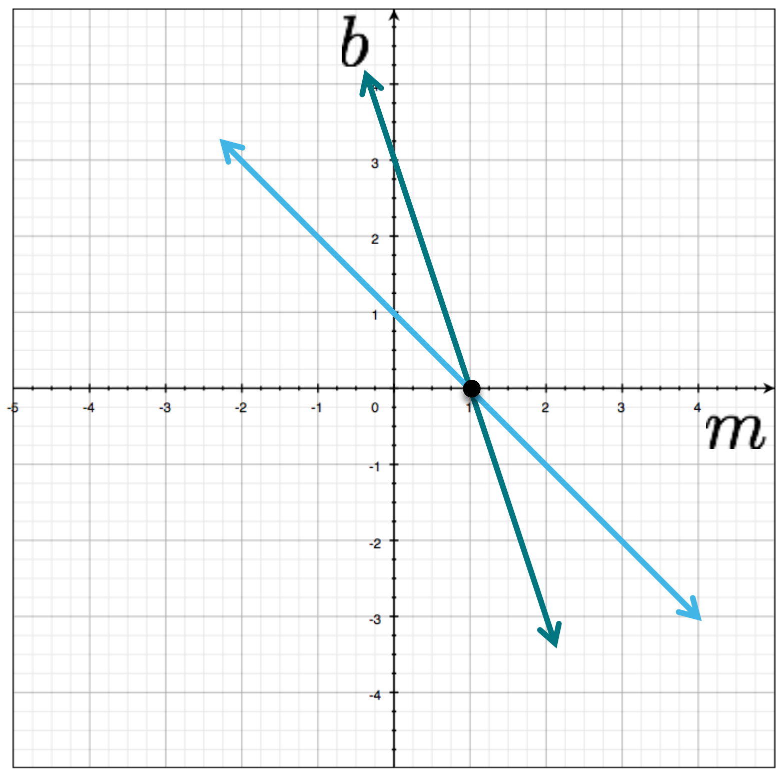


Image space

two points become ?



Parameter space

SAIR
Spatial AI & Robotics Lab

Image and parameter space

variables

$$y = mx + b$$

parameters

variables

$$b = -xm + y$$

parameters

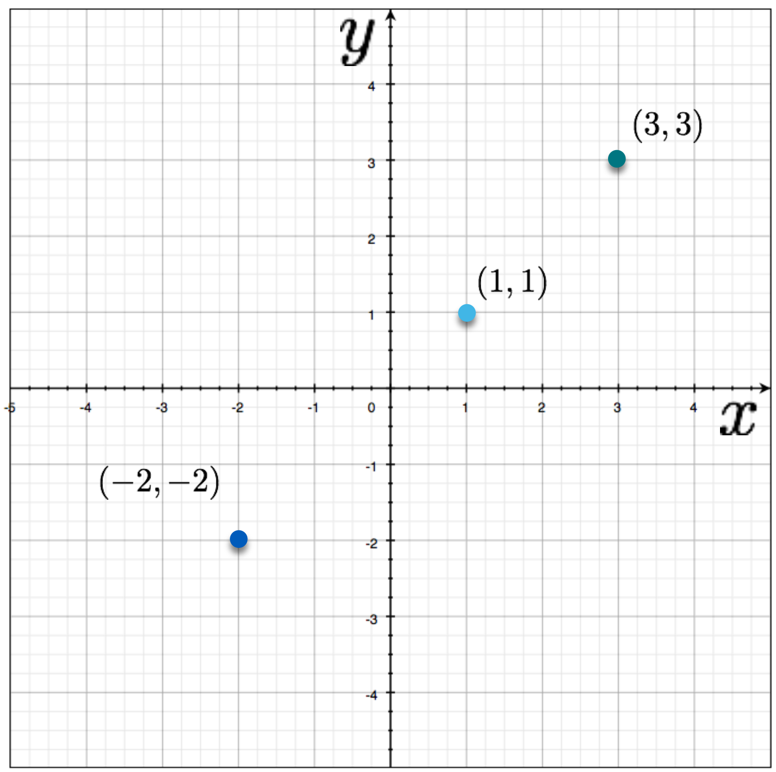
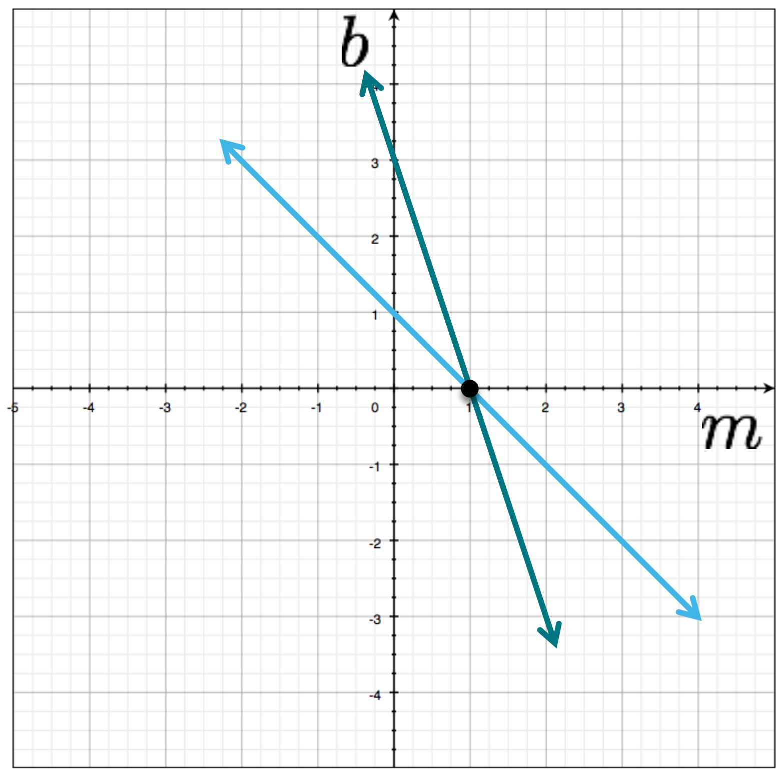


Image space

three points become ?



Parameter space

Image and parameter space

variables

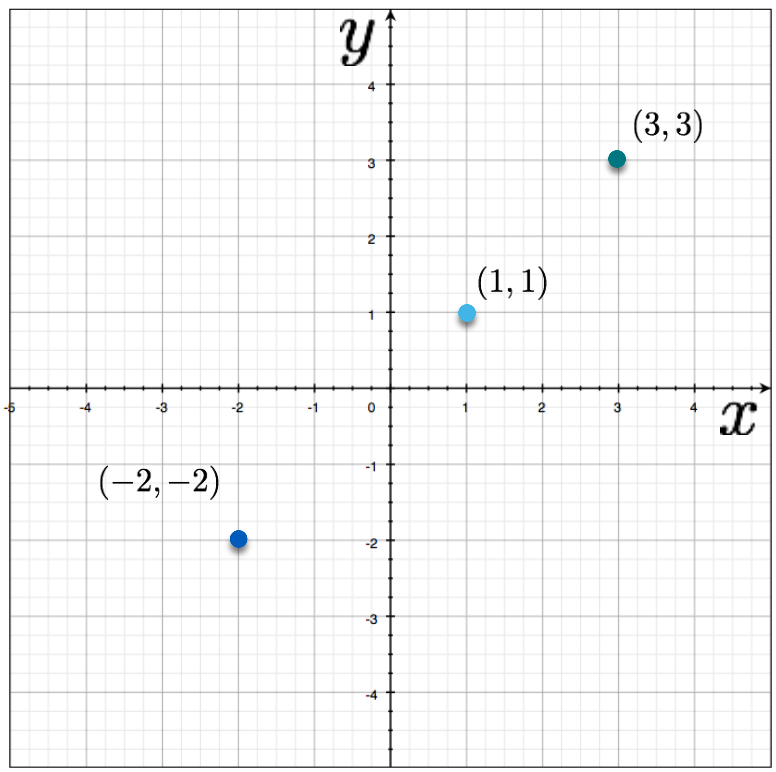
$$y = mx + b$$

parameters

variables

$$b = -xm + y$$

parameters



three points become ?

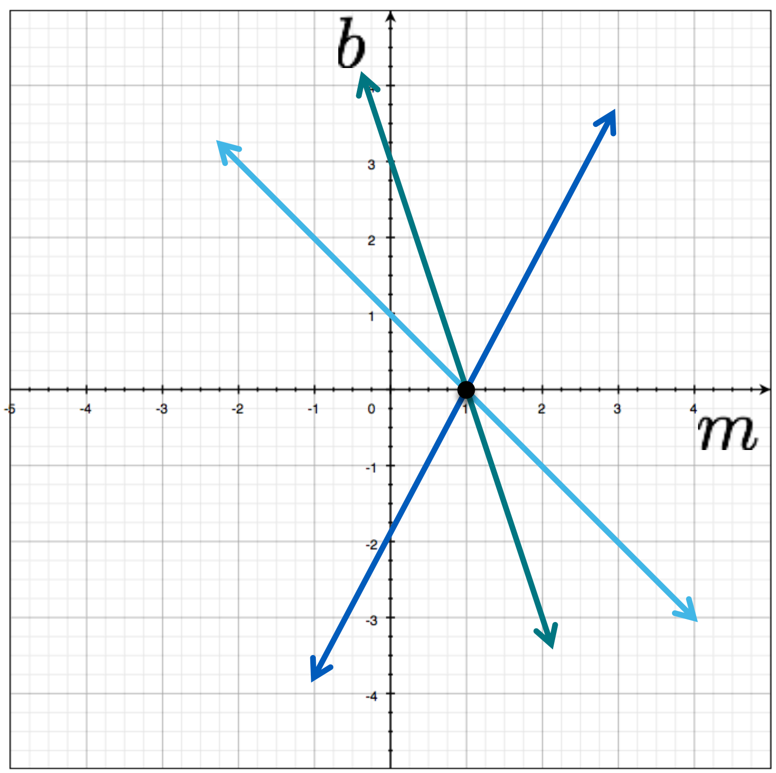


Image space

Parameter space

Image and parameter space

variables

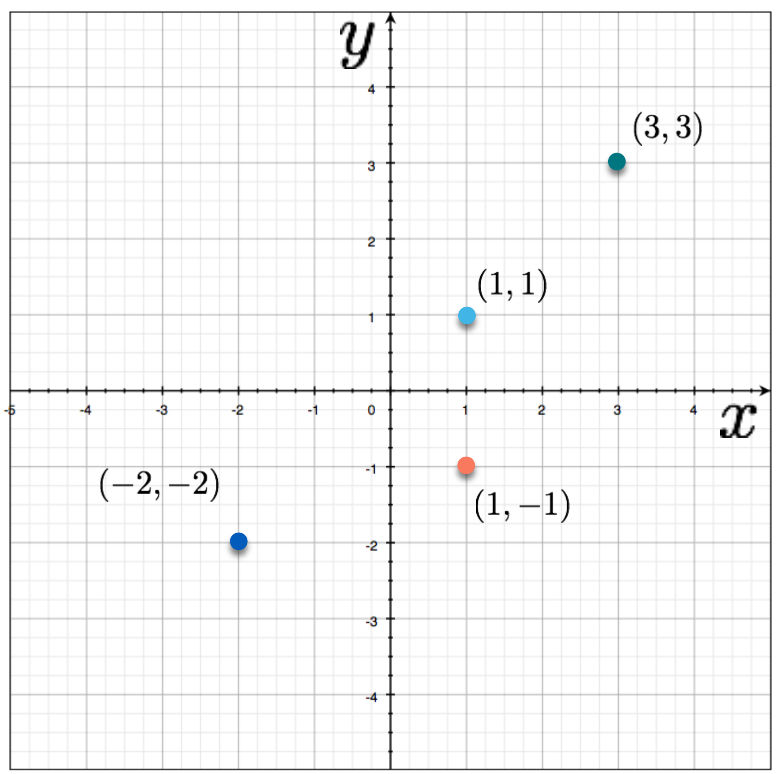
$$y = mx + b$$

parameters

variables

$$b = -xm + y$$

parameters



four points become ?

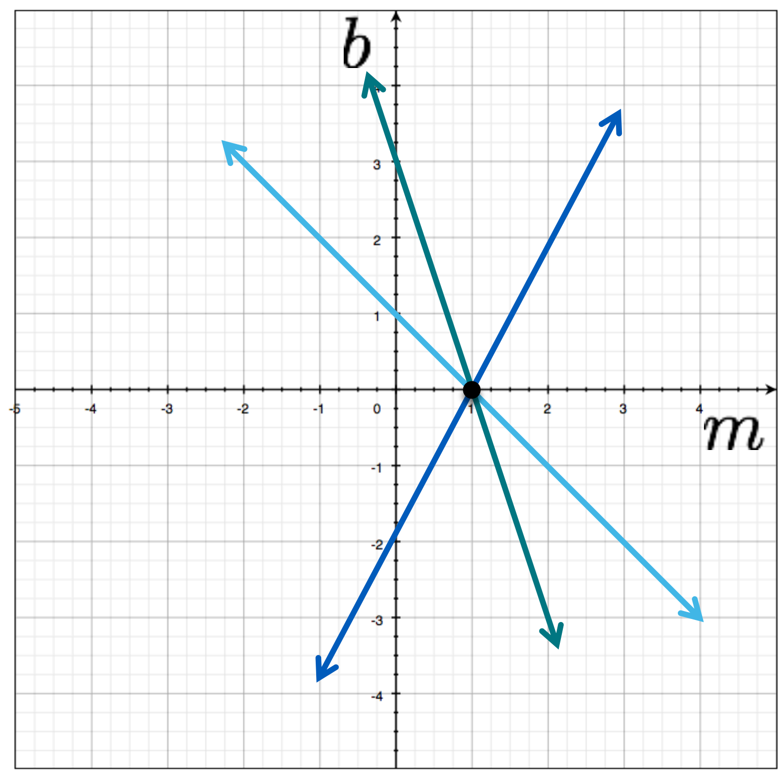


Image space

Parameter space

SAIR
Spatial AI & Robotics Lab

Image and parameter space

variables

$$y = mx + b$$

parameters

variables

$$b = -xm + y$$

parameters

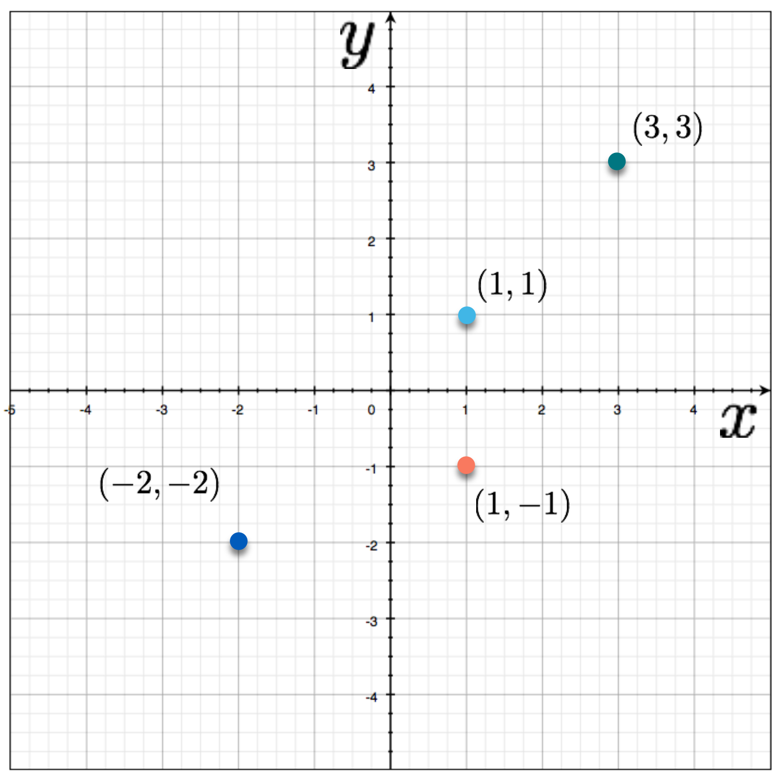
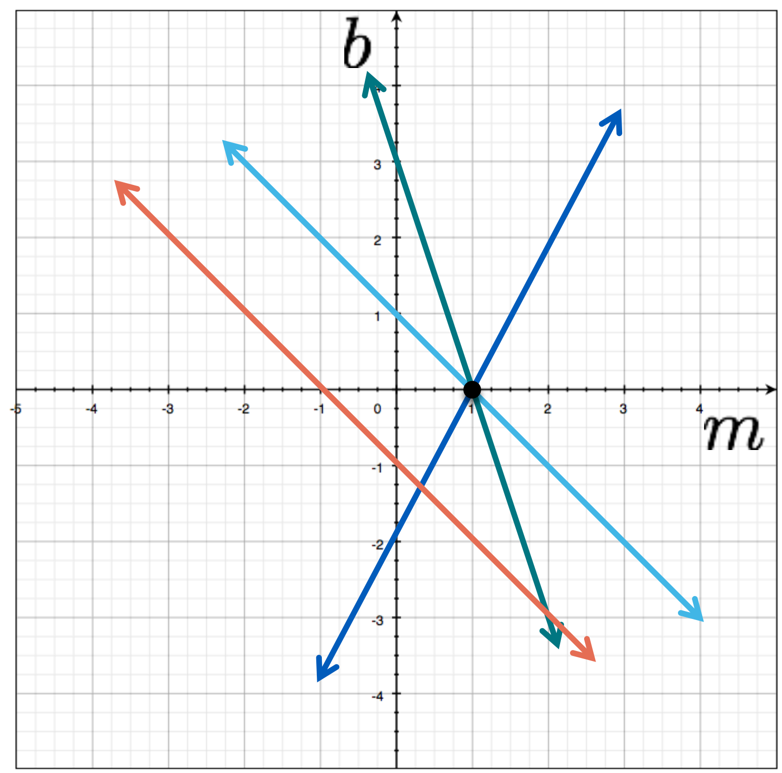


Image space

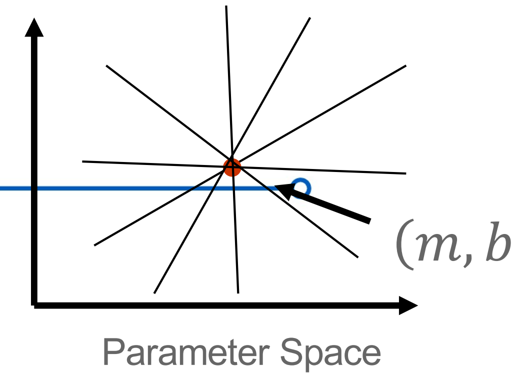
four points become ?



Parameter space

Is this method robust to noise?

Hough Voting



Line Detection Algorithm:

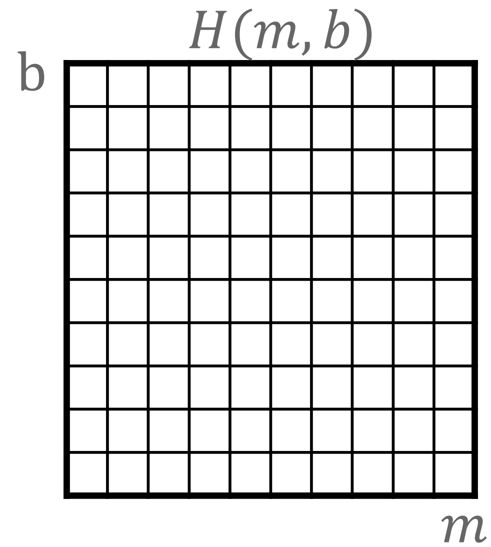
1. Quantize Parameter Space (m, b).

2. Create Hough Space Array $H(m, b) = 0$.

3. For each image point (x_i, y_i):
   For all points (m, b) on $b = -x_i m + y_i$:
   $H(m, b) = H(m, b) + 1$

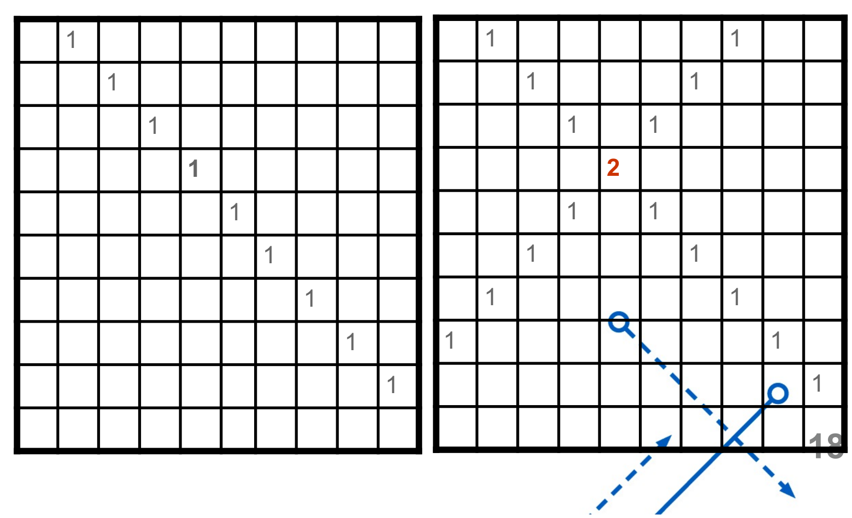
4. Find local maxima in $H(m_m, b_m)$.

5. The detected line: $y = m_m\, x + b_m$.



Is it able to detect multiple lines?

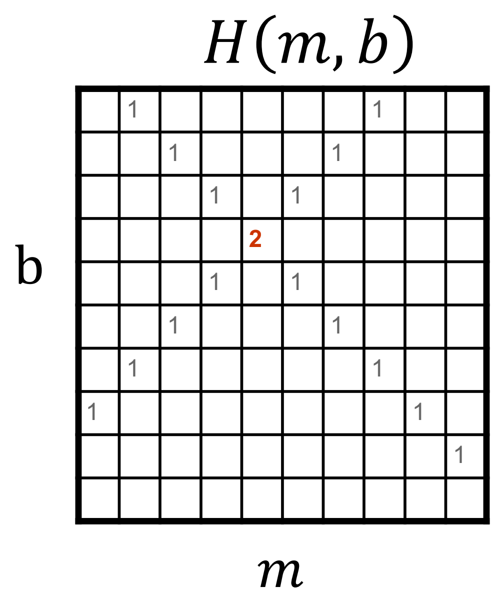
Is this solution good enough?



SAIR
Spatial AI & Robotics Lab

Problems with slope intercept form

How big does the Hough array have to be?



The space of m is huge! The space of b is huge!

$$-\infty \le m \le \infty \qquad -\infty \le b \le \infty$$

Hough Transform with Normal Form

Use normal form:

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

The Hough space become $H(\rho, \theta)$

Hough Space

$$0 \leq \theta \leq \pi$$
$$0 \leq \rho \leq \rho_{max}$$

(Finite Hough Array Size)

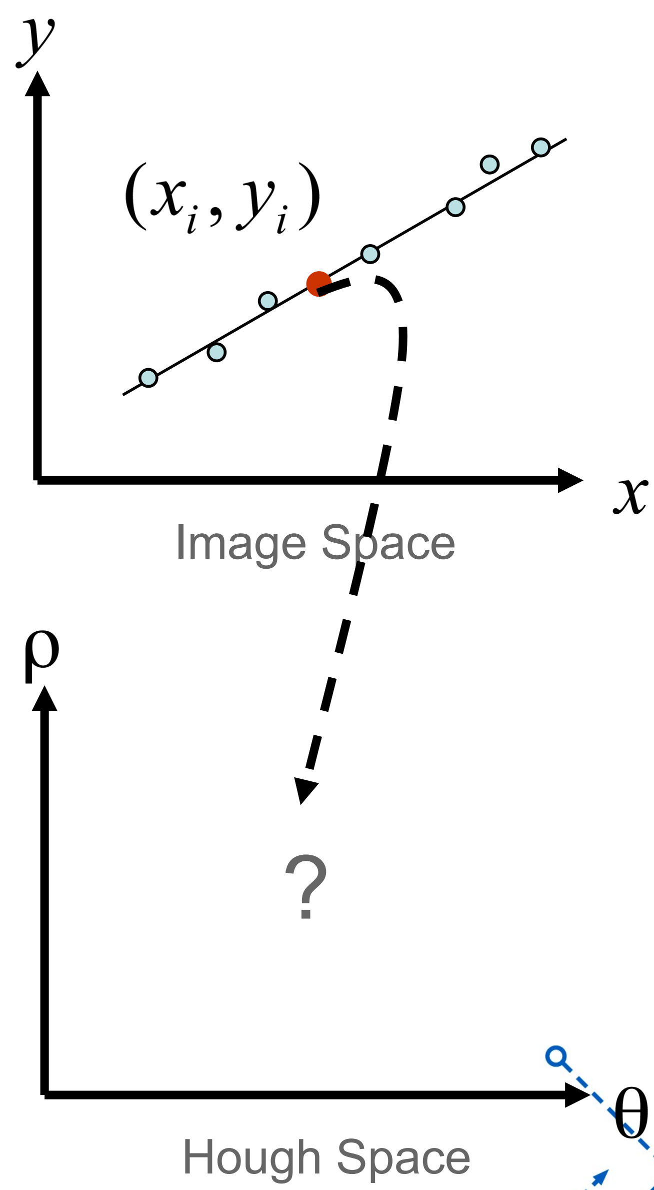


Image and parameter space

variables

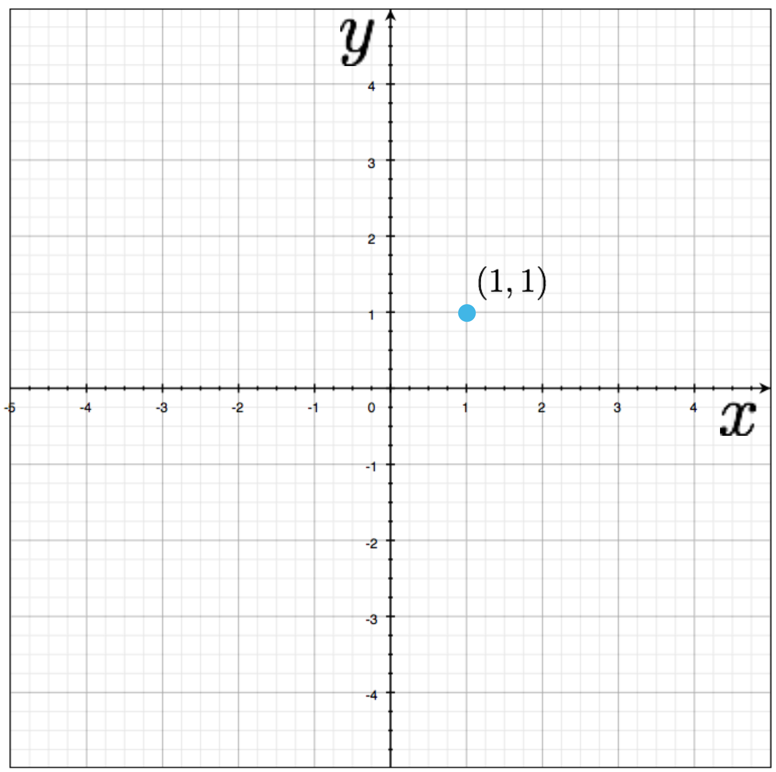
$$x \cos\theta + y \sin\theta = \rho$$

parameters

parameters

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

variables



a point becomes?

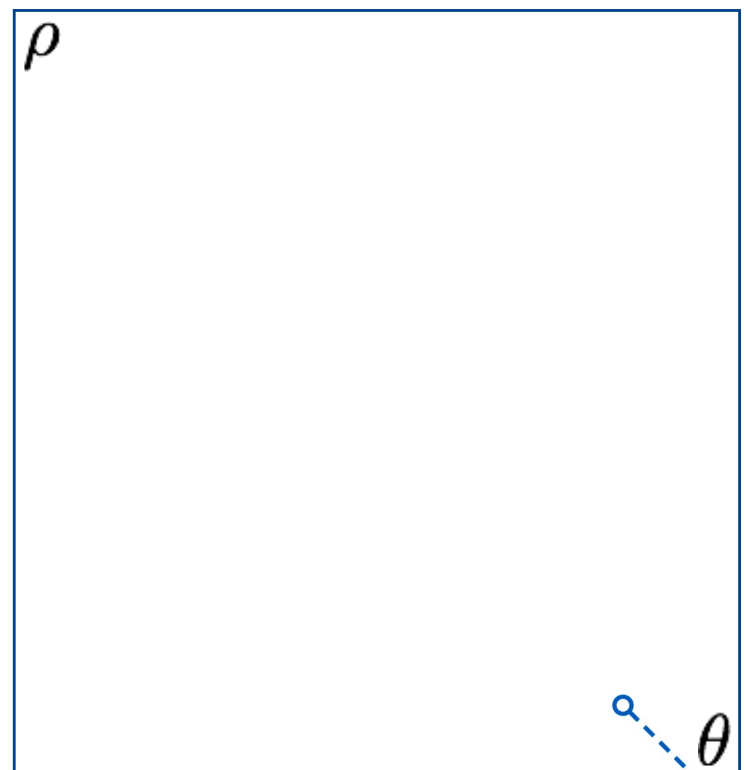


Image space

Parameter space

Image and parameter space

variables

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

parameters

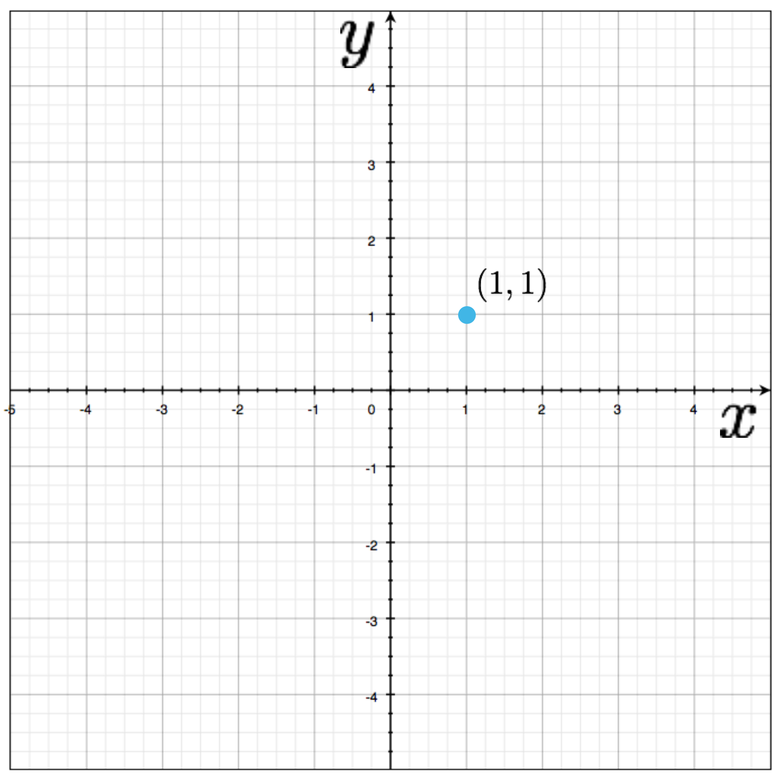


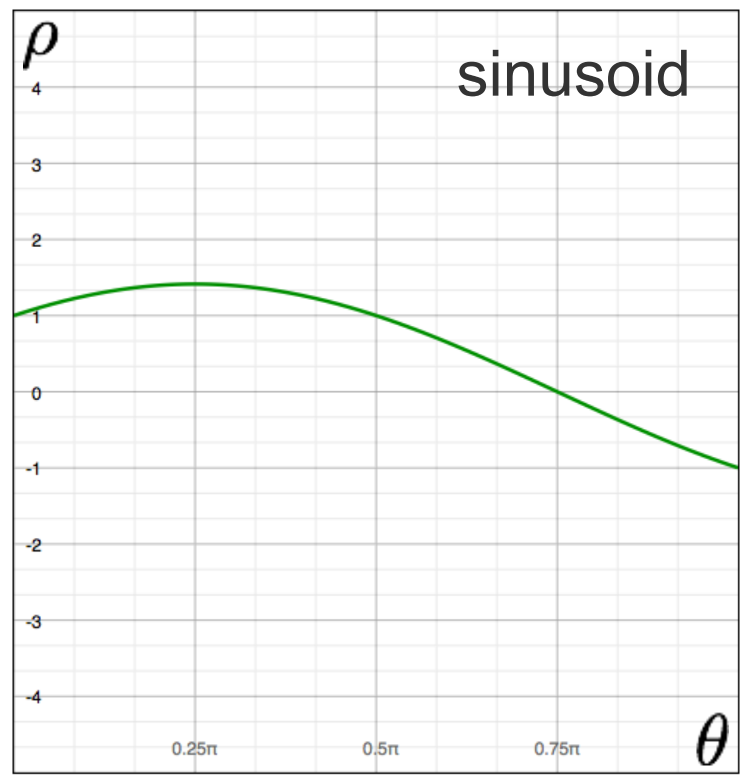
Image space

a point becomes a wave

parameters

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

variables



Parameter space

Image and parameter space

variables

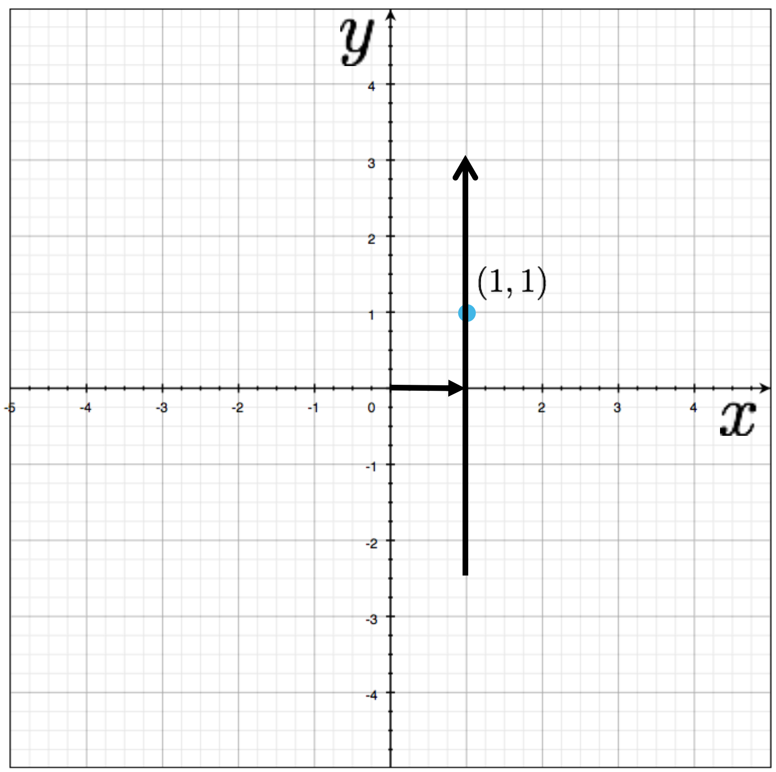
$$x \cos\theta + y \sin\theta = \rho$$

parameters

parameters

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

variables



a line becomes?

Image space

Parameter space

Image and parameter space

variables

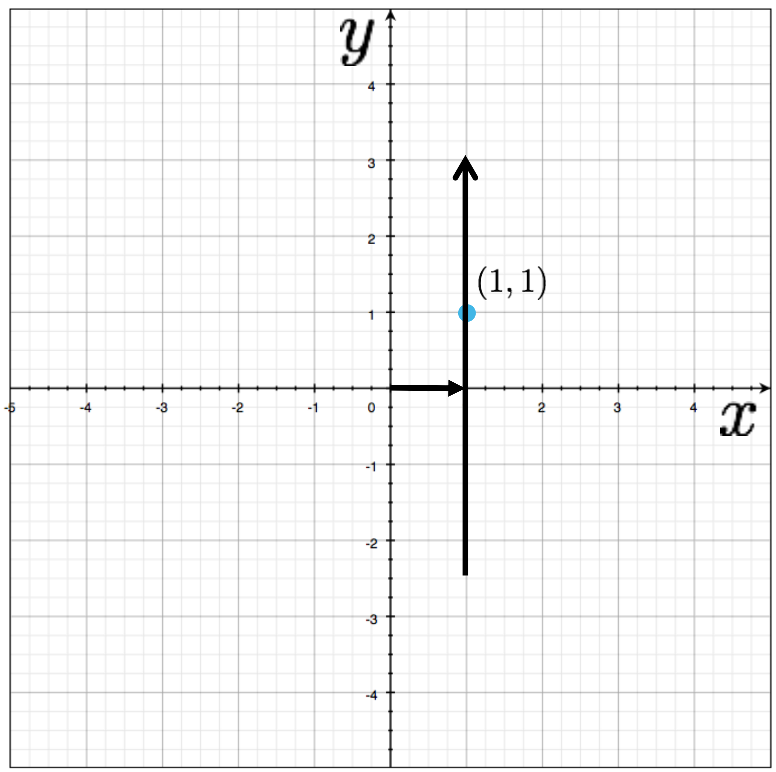
$$x\cos\theta + y\sin\theta = \rho$$

parameters

parameters

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

variables



a line becomes a point

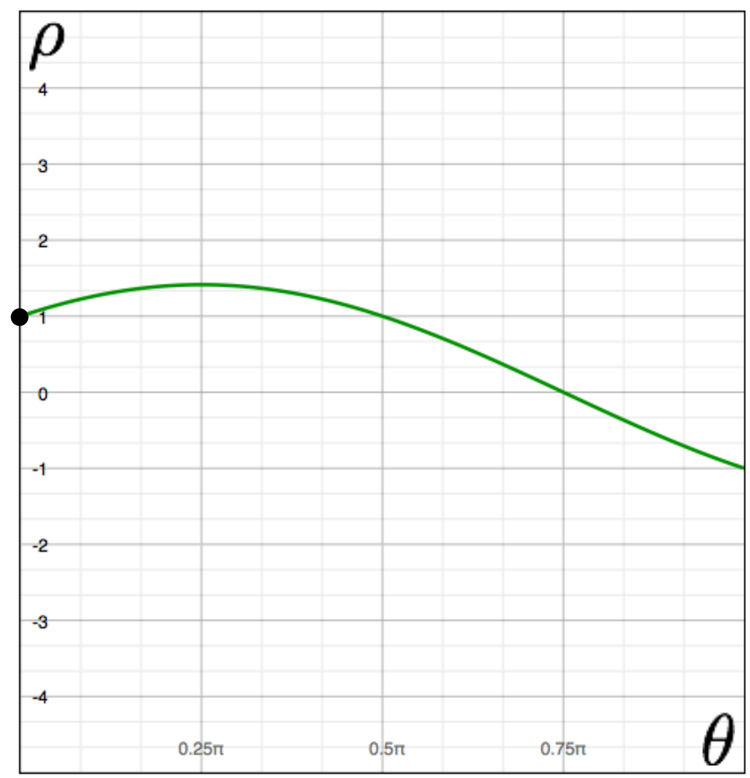


Image space

Parameter space

Image and parameter space

variables

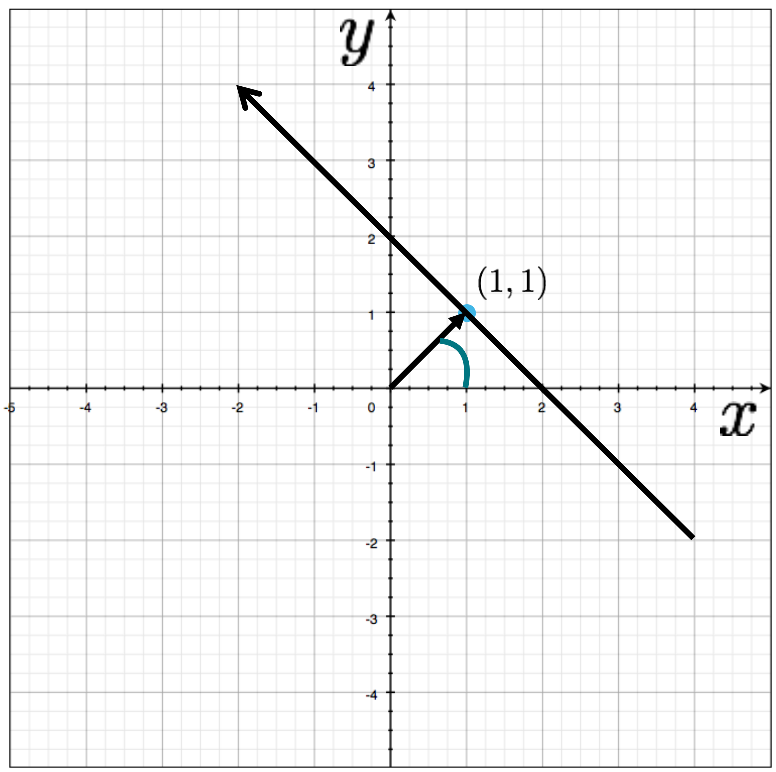
$$x\cos\theta + y\sin\theta = \rho$$

parameters

parameters

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

variables



a line becomes?

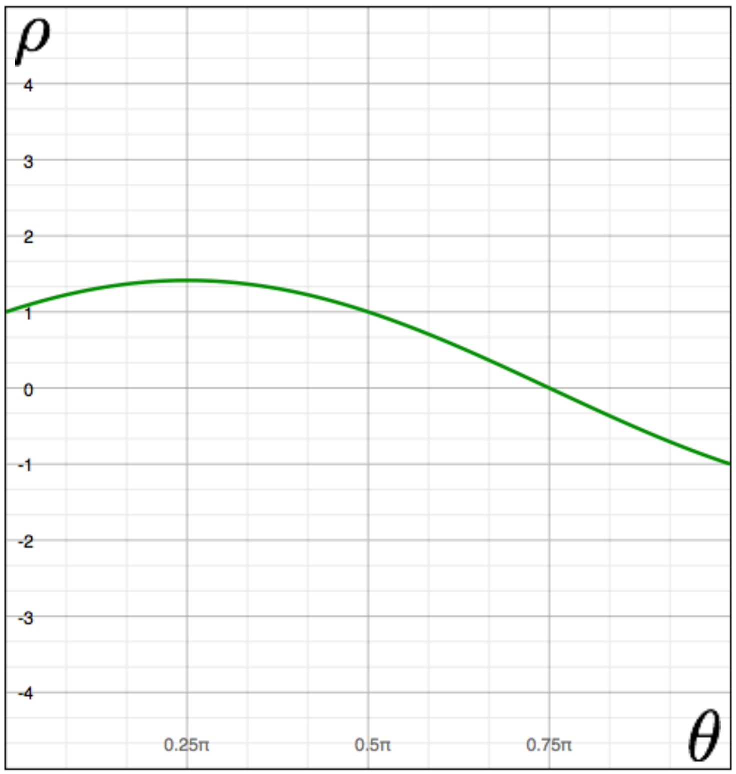


Image space

Parameter space

Image and parameter space

variables

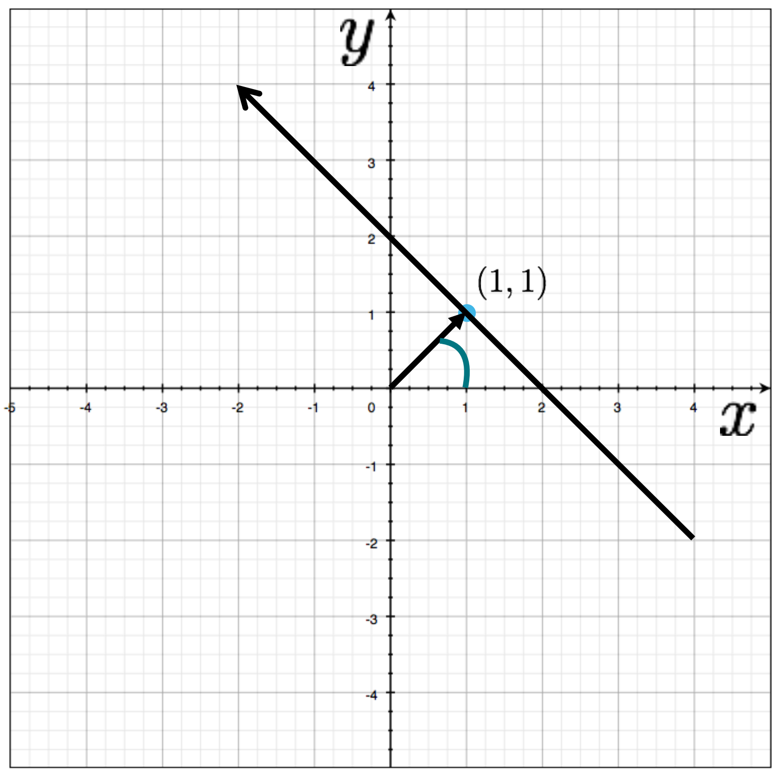
$$x \cos\theta + y \sin\theta = \rho$$

parameters

parameters

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

variables



a line becomes a point

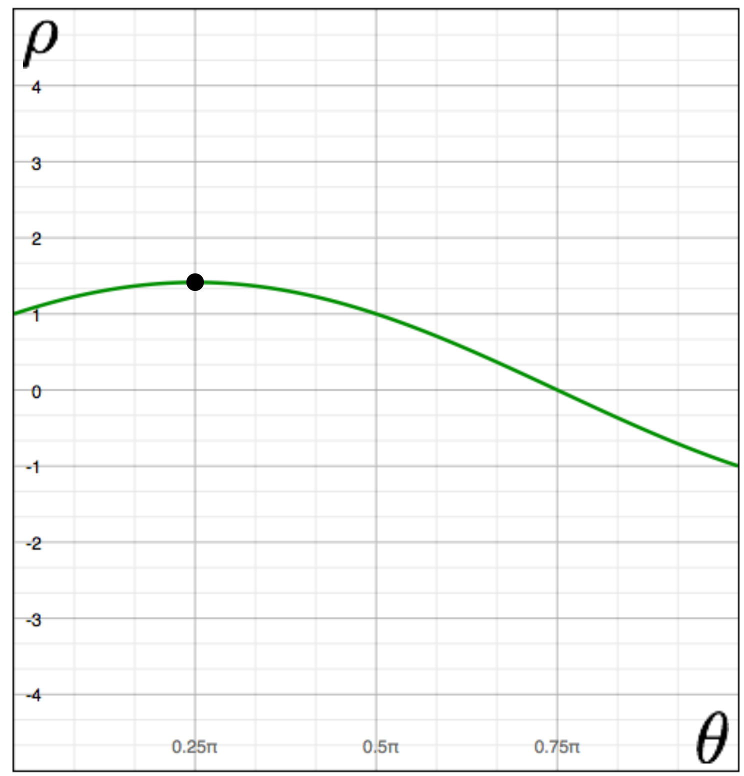


Image space

Parameter space

Image and parameter space

variables

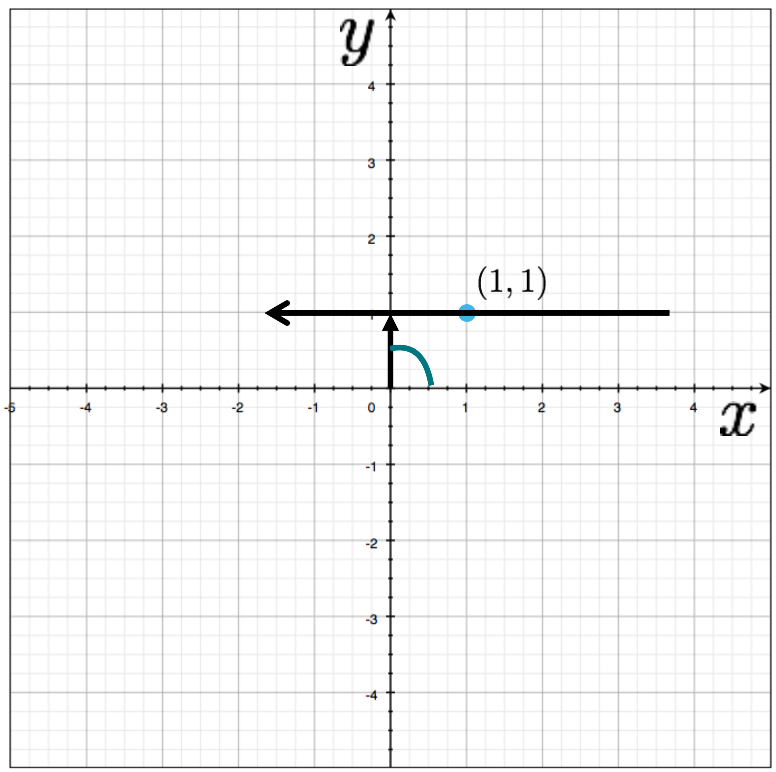
$$x \cos\theta + y \sin\theta = \rho$$

parameters

parameters

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

variables



a line becomes a point

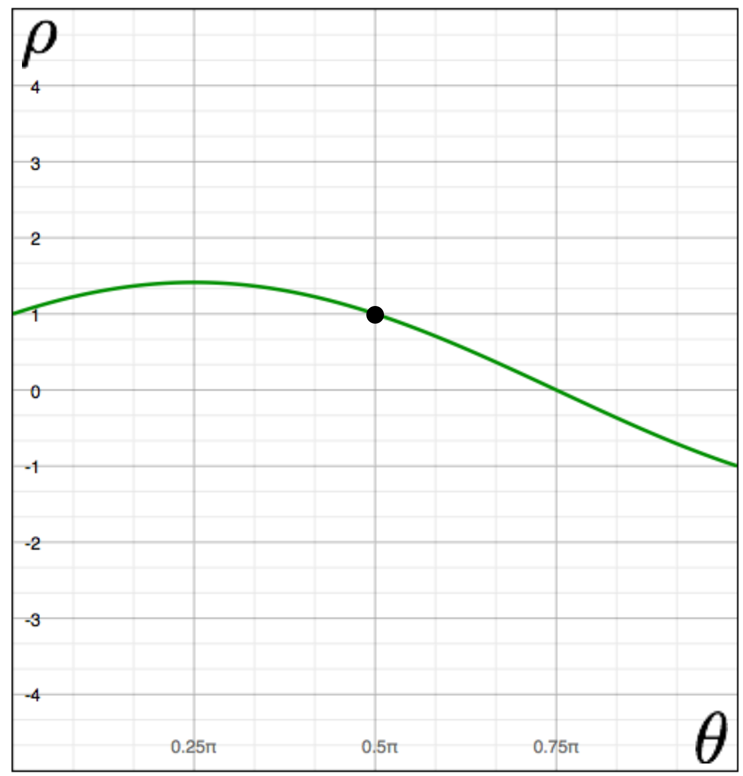


Image space

Parameter space

Image and parameter space

variables

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

parameters

parameters

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

variables

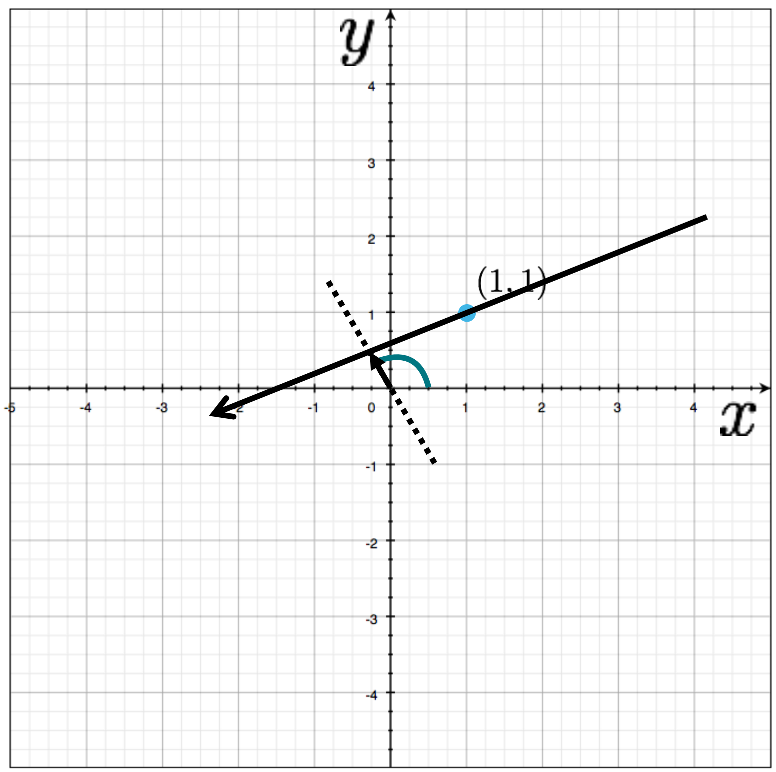
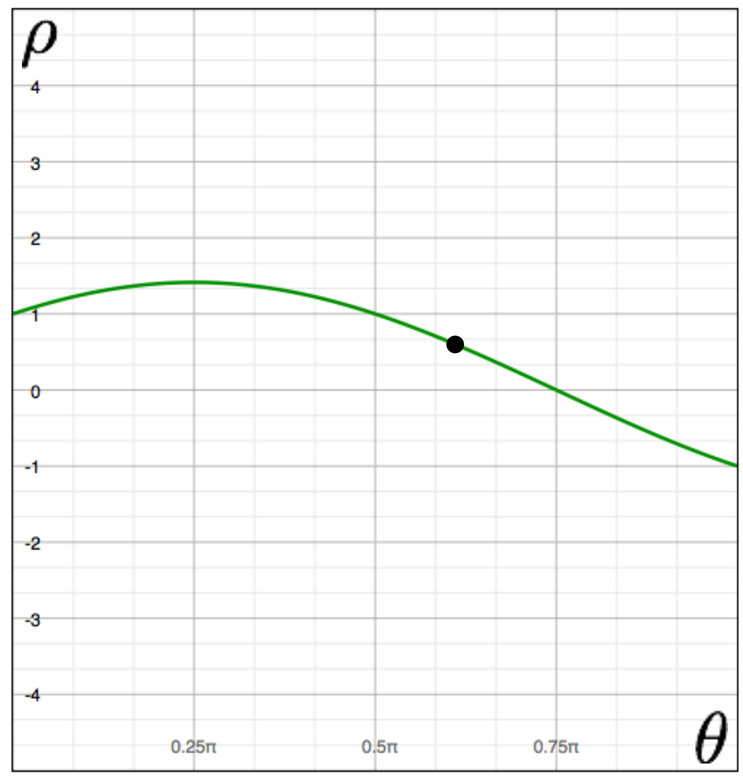


Image space

a line
becomes
a point



Parameter space

Image and parameter space

variables

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

parameters

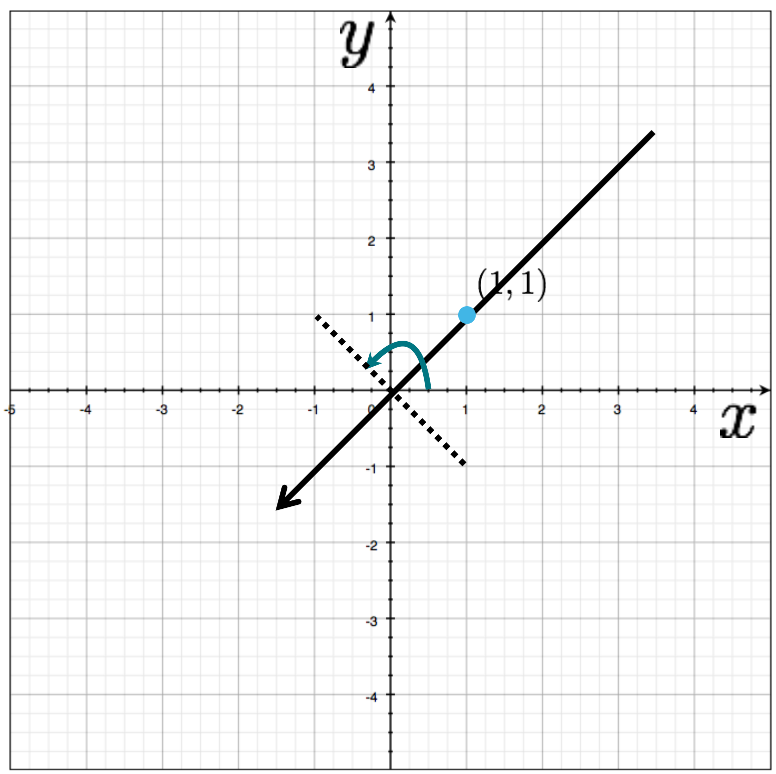


Image space

a line becomes a point

parameters

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

variables

Parameter space

Image and parameter space

variables

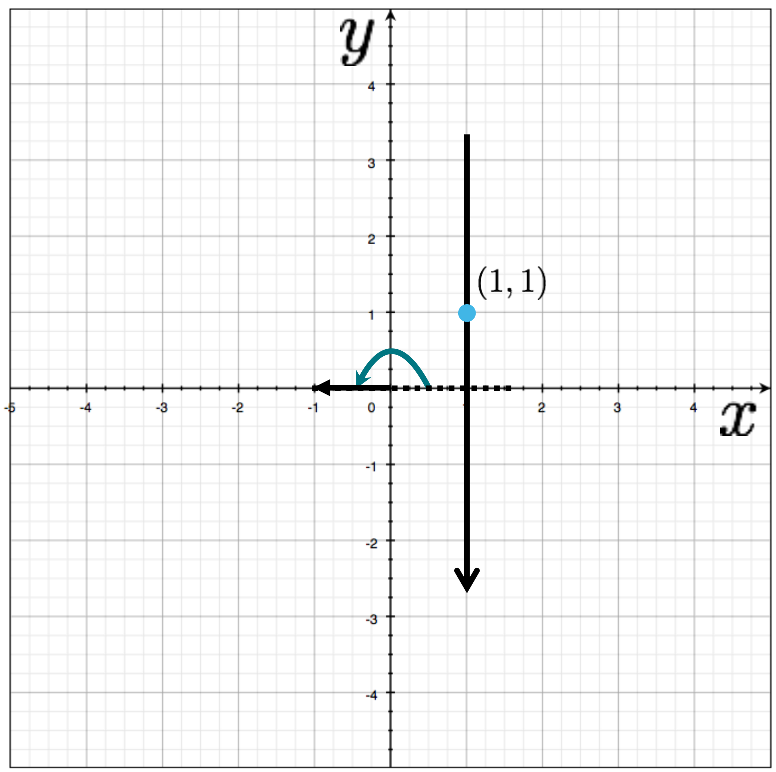
$$x \cos\theta + y \sin\theta = \rho$$

parameters

parameters

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

variables



a line becomes a point

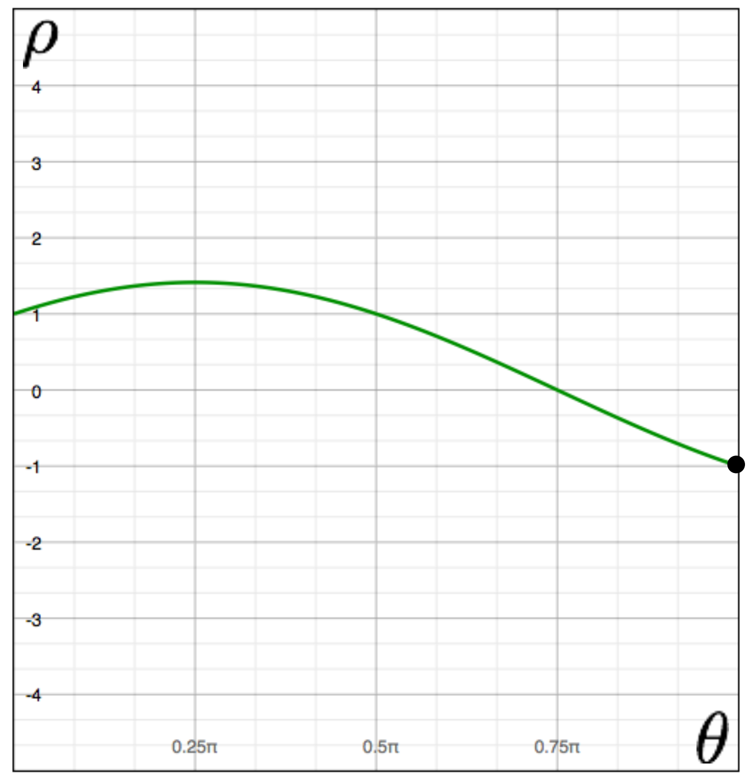


Image space

Parameter space

Image and parameter space

variables

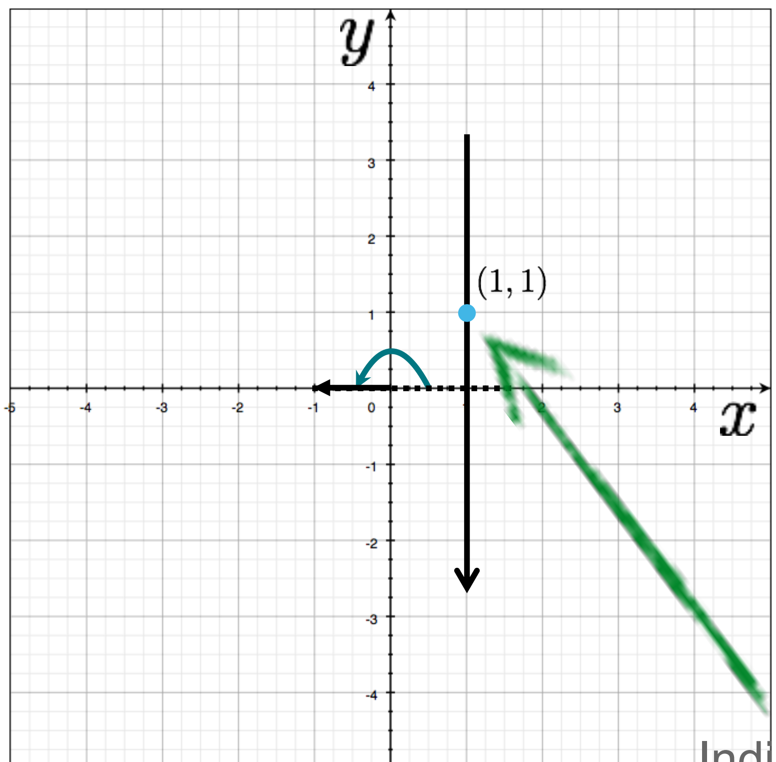
$$x\cos\theta + y\sin\theta = \rho$$

parameters

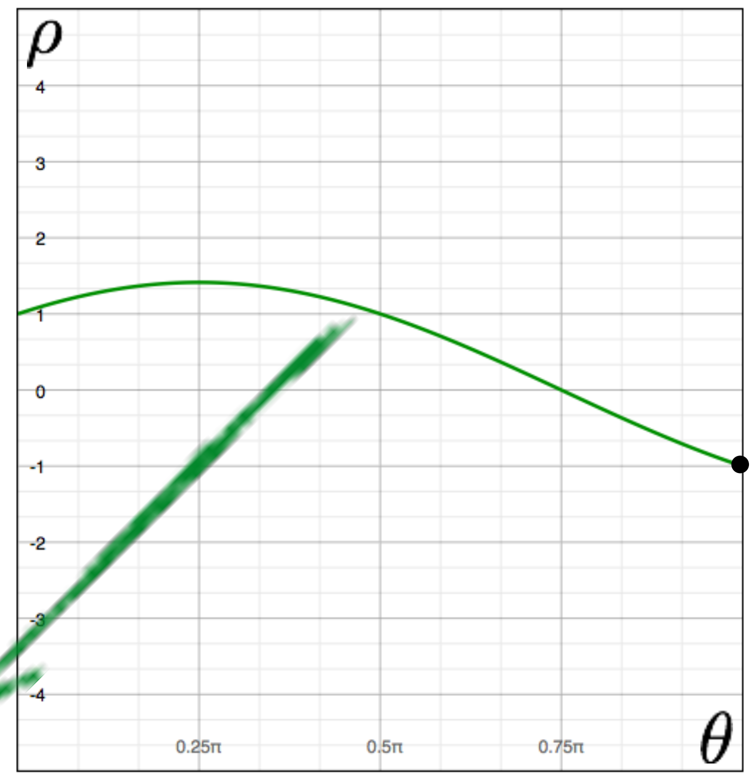
parameters

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

variables



a line becomes a point



Indicate all lines pass through (1,1)

Image space

Parameter space

Image and parameter space

variables

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

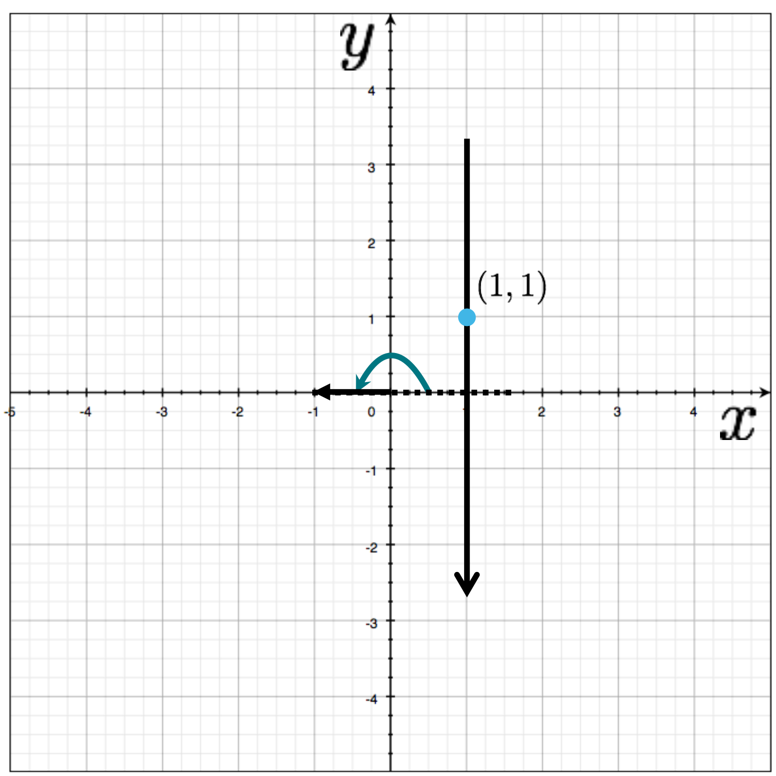
parameters

parameters

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

variables

Wait …why is ρ negative?



a line becomes a point

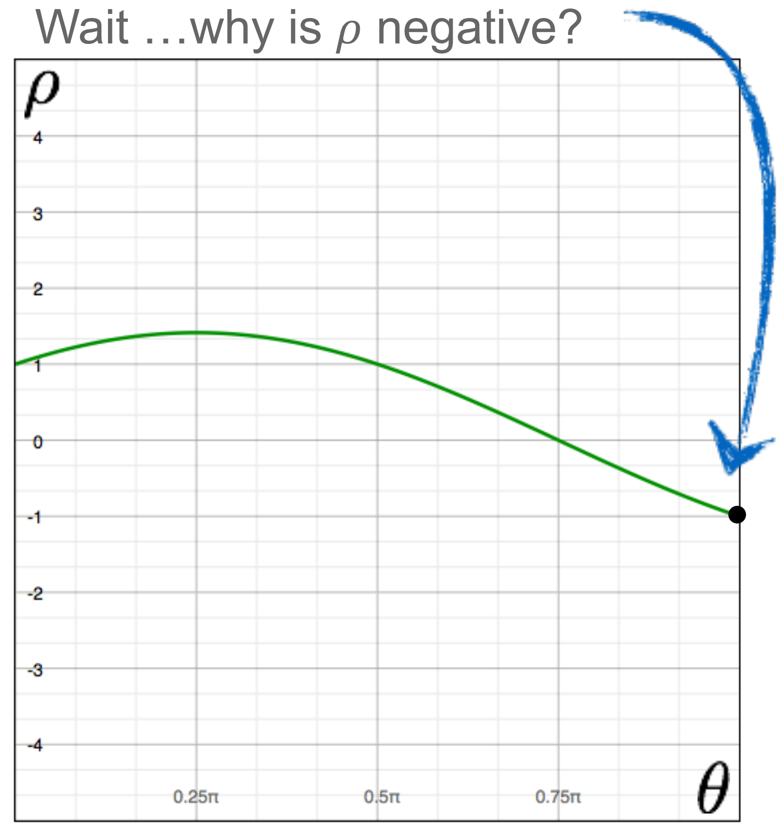


Image space

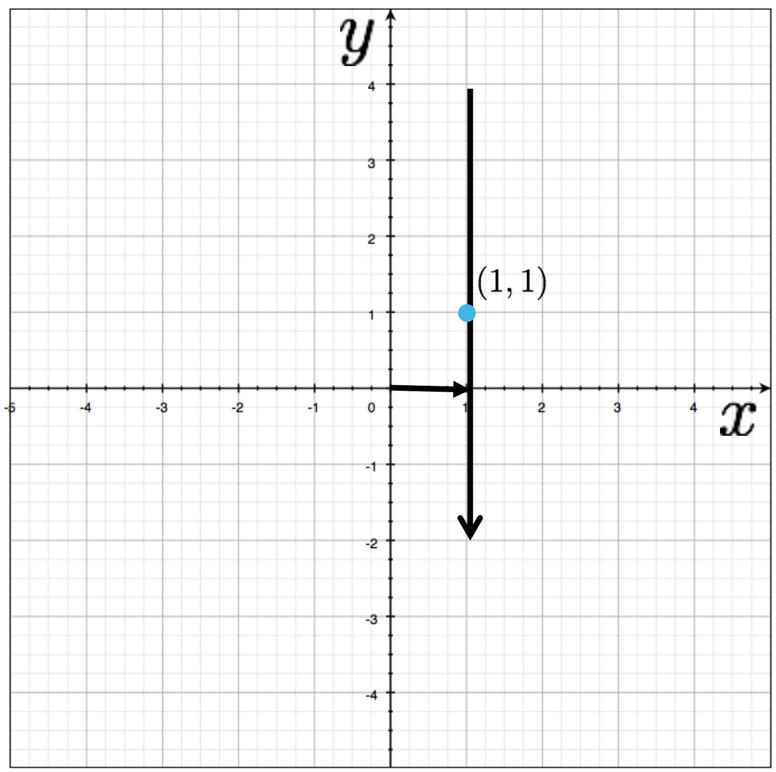
Parameter space

Image and parameter space

variables

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

parameters



a line becomes a point

parameters

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

variables

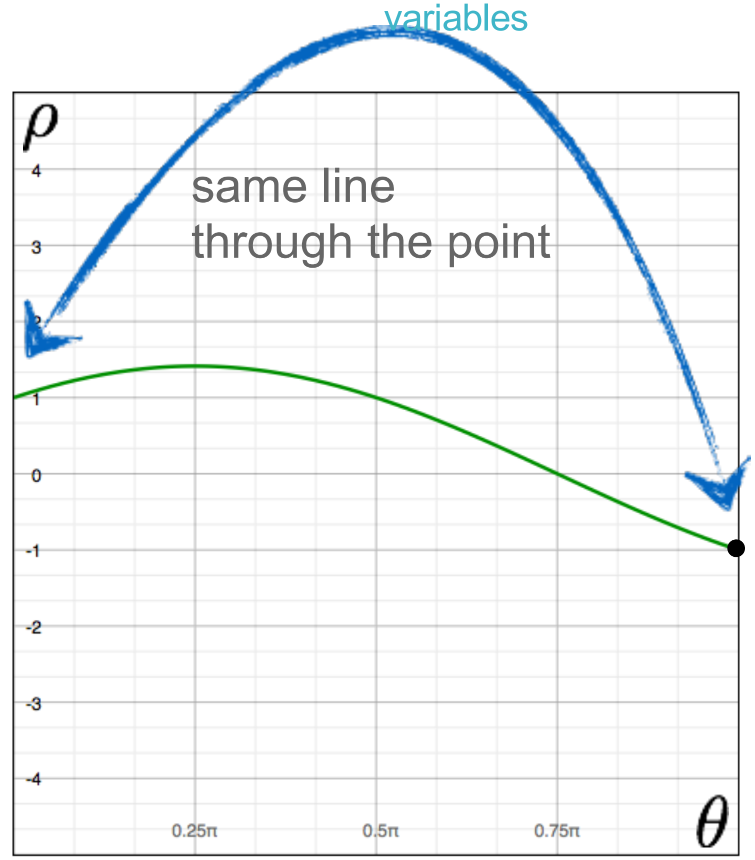


Image space

Parameter space

There are two ways to write the same line

Positive ρ version:

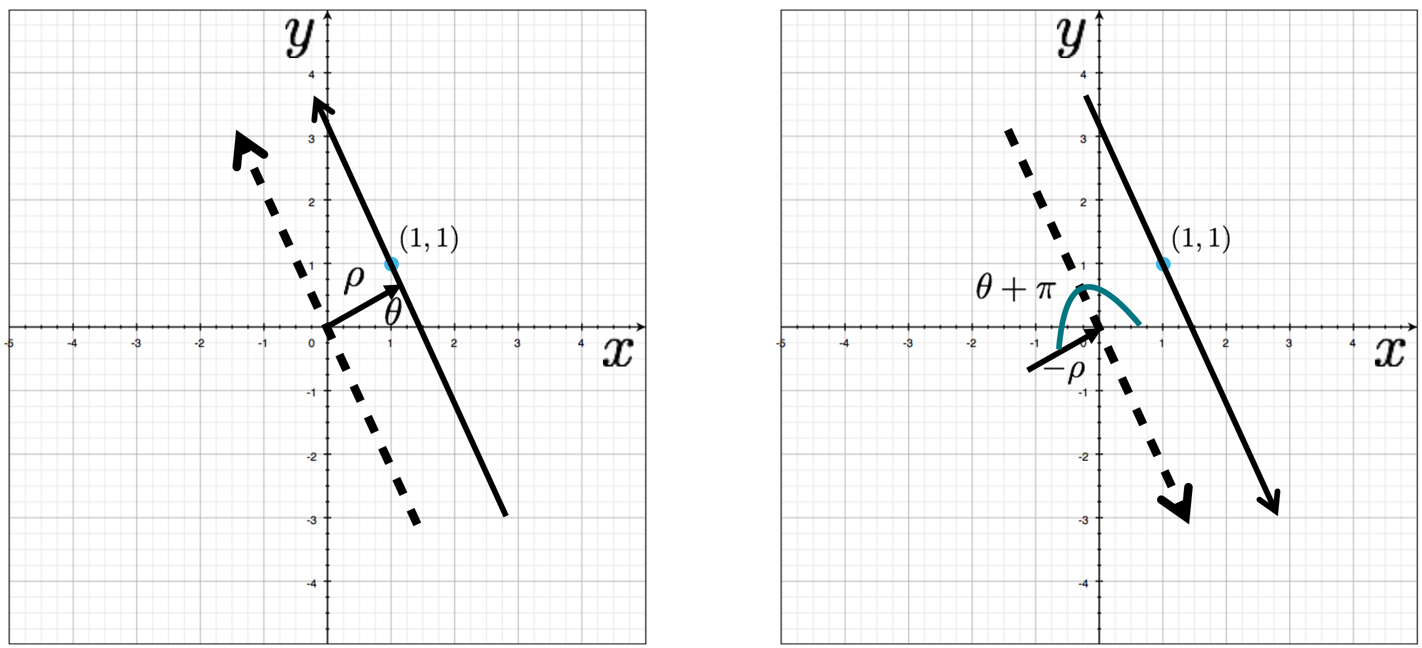
$$x\cos\theta + y\sin\theta = \rho$$

Negative ρ version:

$$x\cos(\theta+\pi) + y\sin(\theta+\pi) = -\rho$$

Recall:

$$\sin(\theta) = -\sin(\theta+\pi)$$
$$\cos(\theta) = -\cos(\theta+\pi)$$



SAIR
Spatial AI & Robotics Lab

Image and parameter space

variables

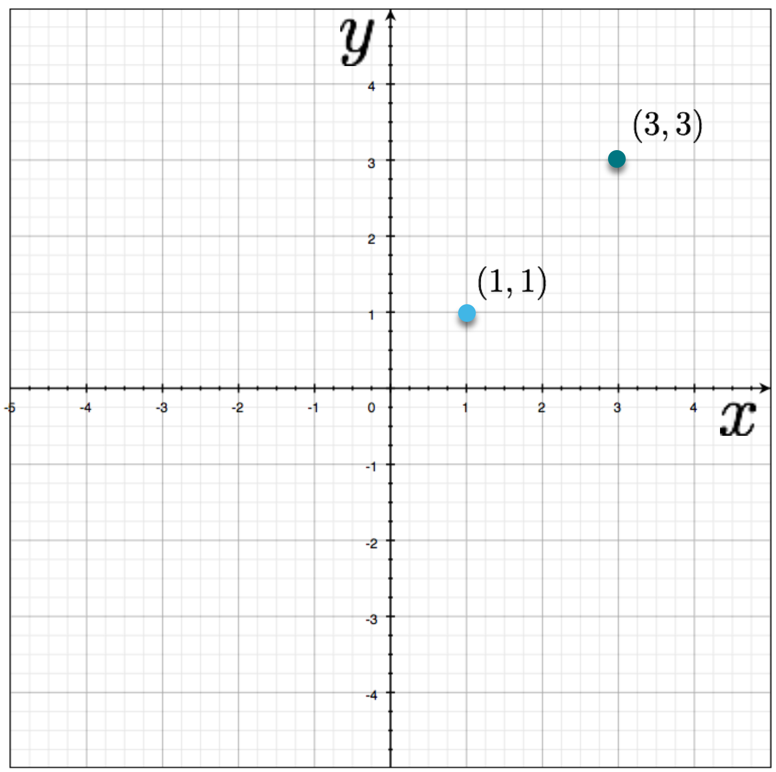
$$x \cos\theta + y \sin\theta = \rho$$

parameters

parameters

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

variables



two points become ?

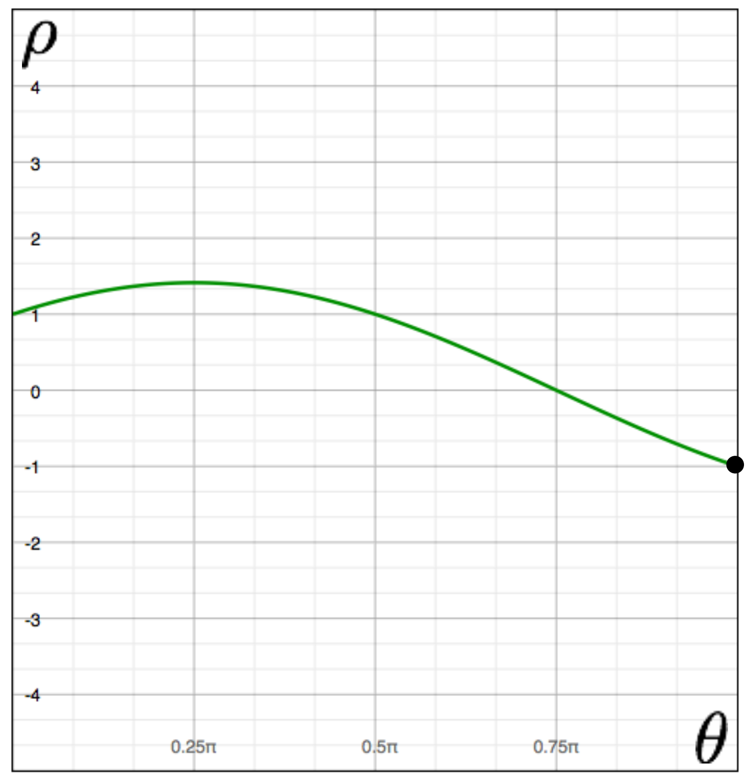


Image space

Parameter space

Image and parameter space

variables

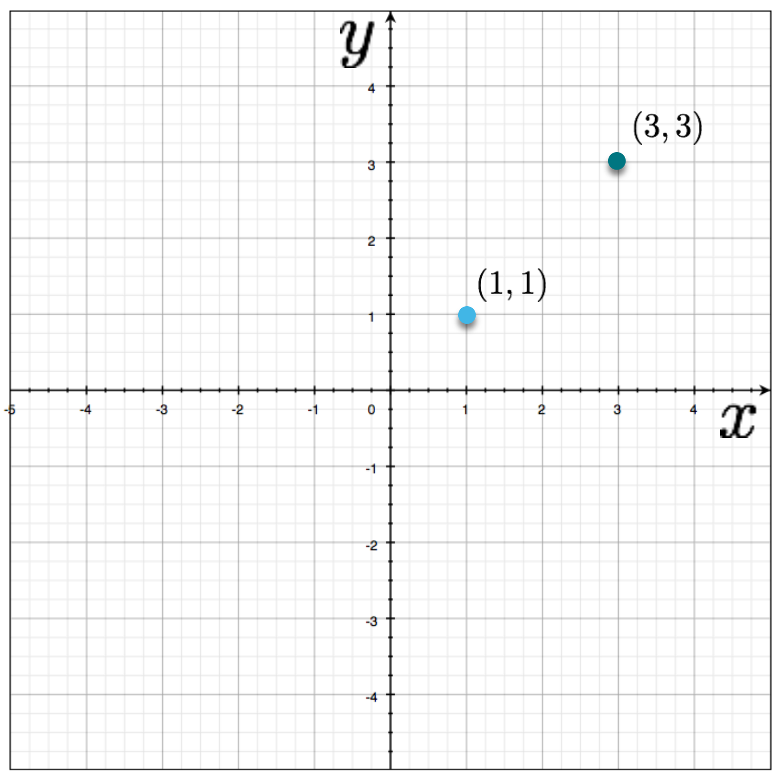
$$x \cos\theta + y \sin\theta = \rho$$

parameters

parameters

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

variables



two points become ?

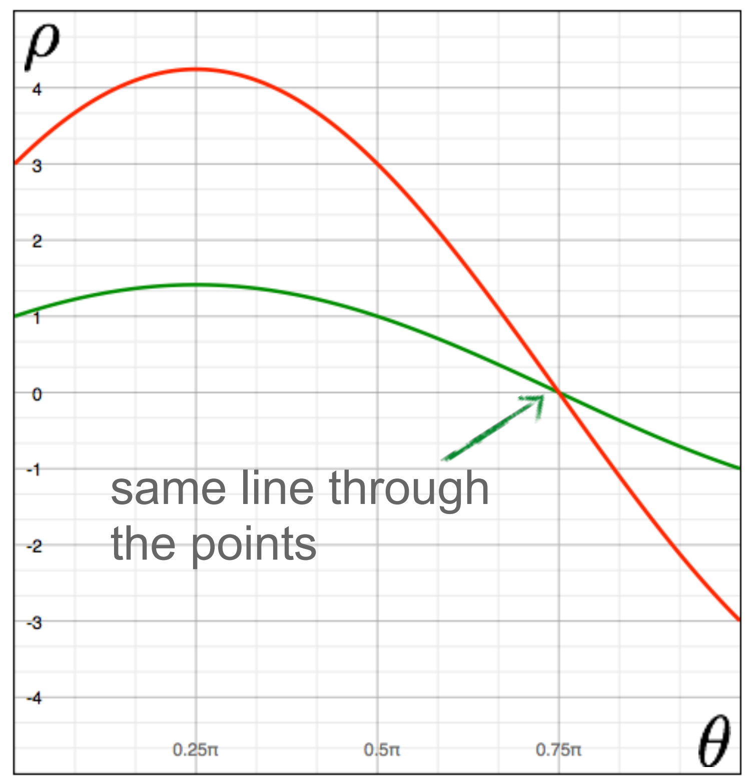


Image space

Parameter space

Image and parameter space

variables

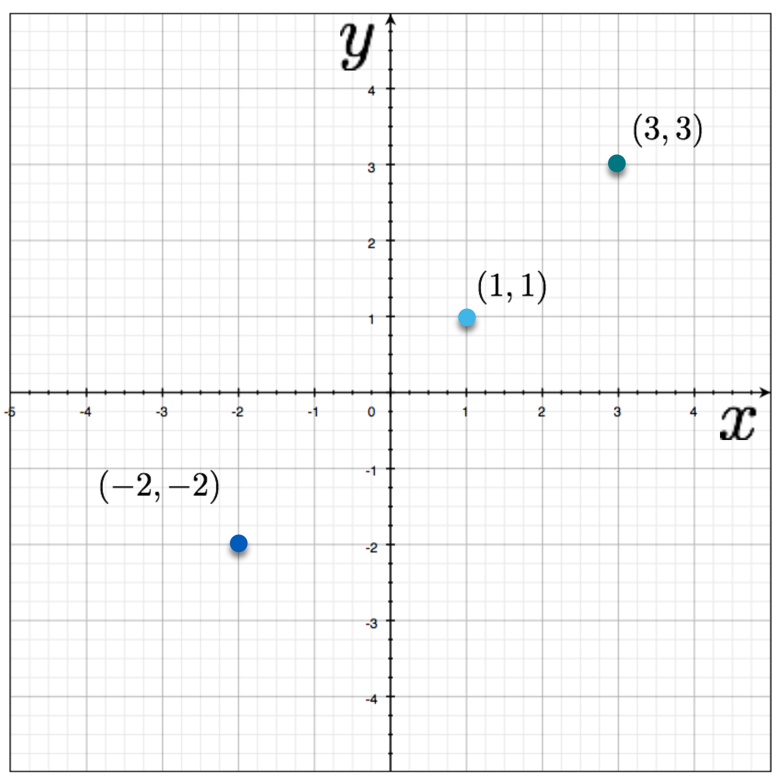
$$x \cos\theta + y \sin\theta = \rho$$

parameters

parameters

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

variables



three points become ?

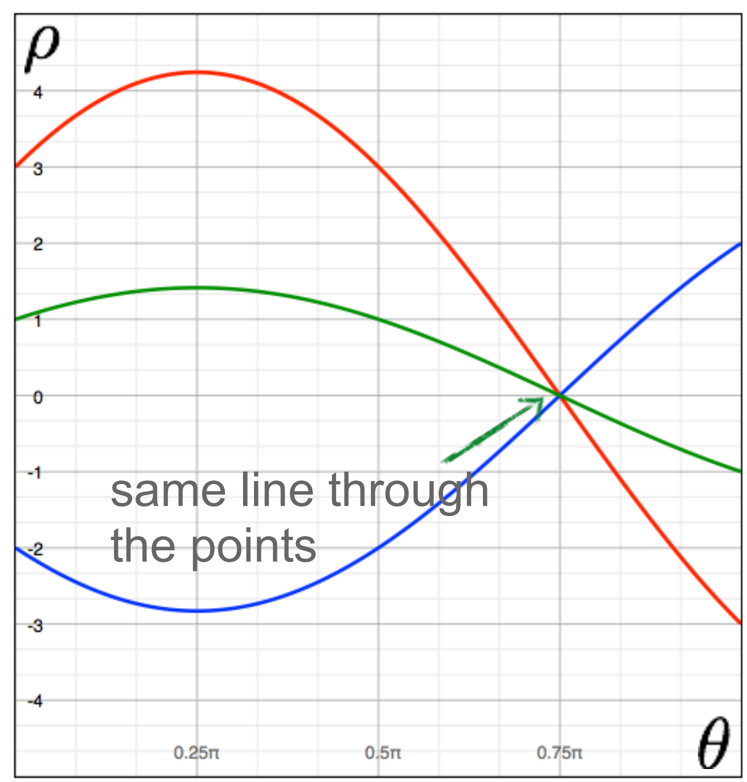


Image space

Parameter space

Image and parameter space

variables

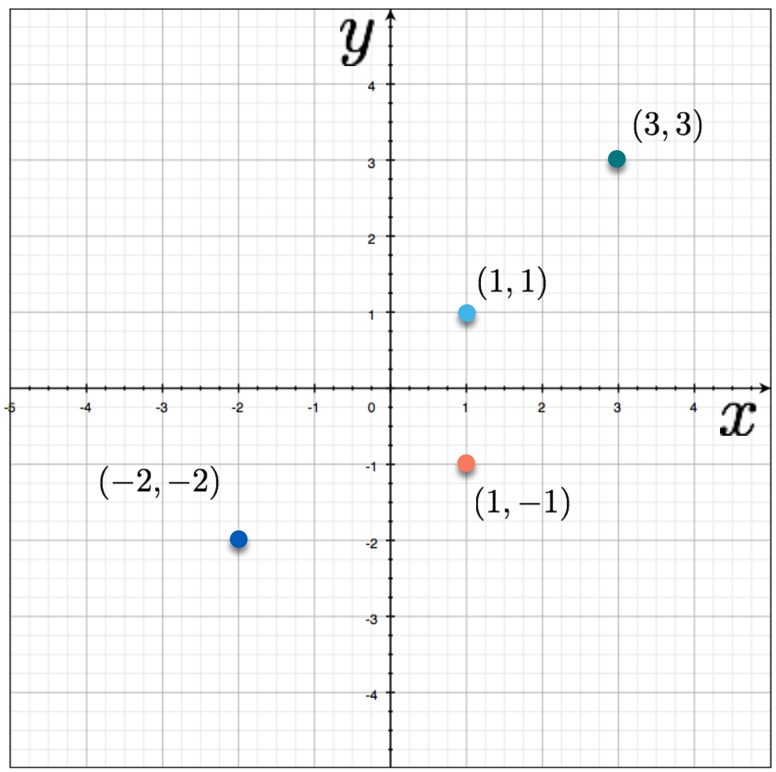
$$x \cos\theta + y \sin\theta = \rho$$

parameters

parameters

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

variables



four points become ?

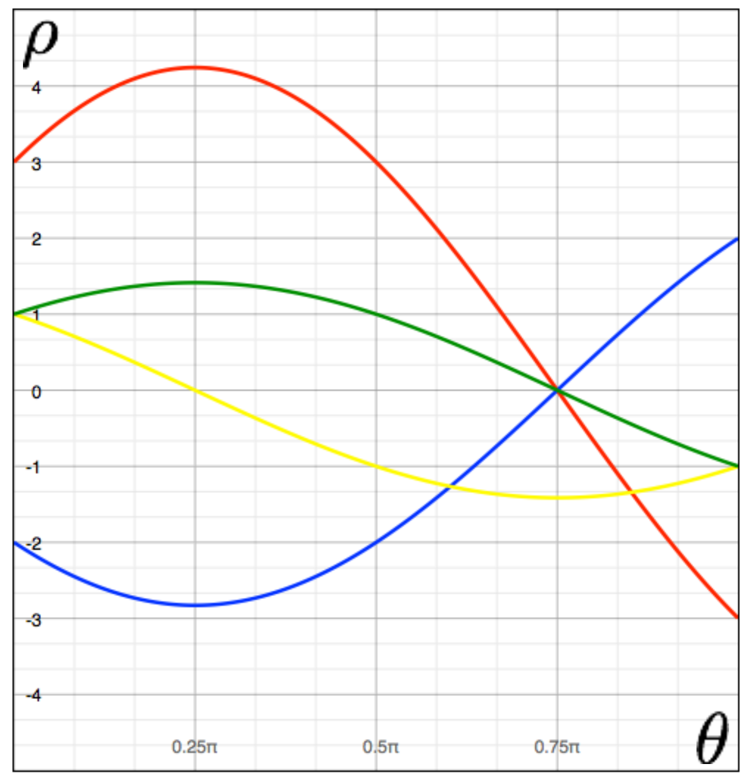
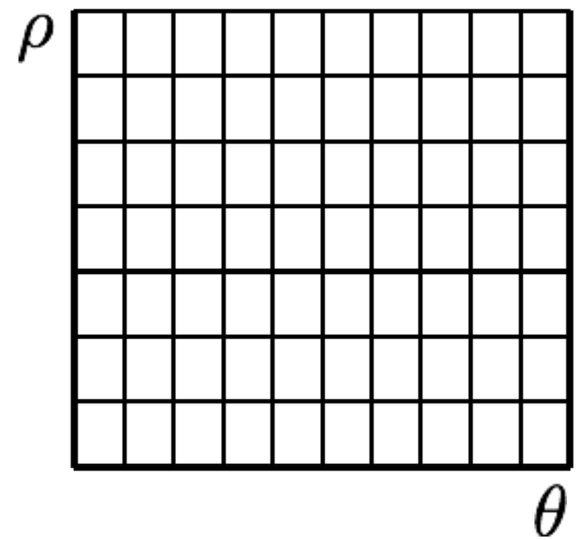


Image space

Parameter space

Line Detection by Hough Voting

H: accumulator array (votes)



Algorithm:

1. Quantize Parameter Space (θ, ρ).

2. Create Hough Space Array $H(\theta, \rho) = 0$.

3. For each image point (x_i, y_i):
   For all points (θ, ρ) on $\rho = x_i \cos\theta + y_i \sin\theta$:
   $H(\theta, \rho) = H(\theta, \rho) + 1$

4. Find local maxima $H(\theta_m, \rho_m)$.

5. The detected line: $x \cos\theta_m + y \sin\theta_m = \rho_m$

Line Detection by Hough Voting

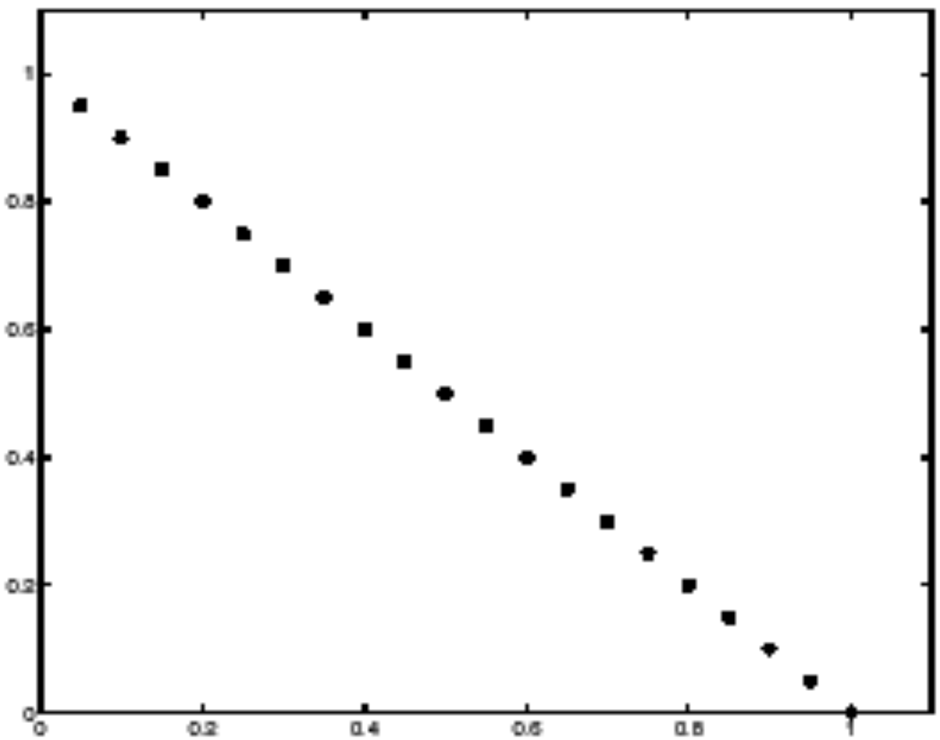
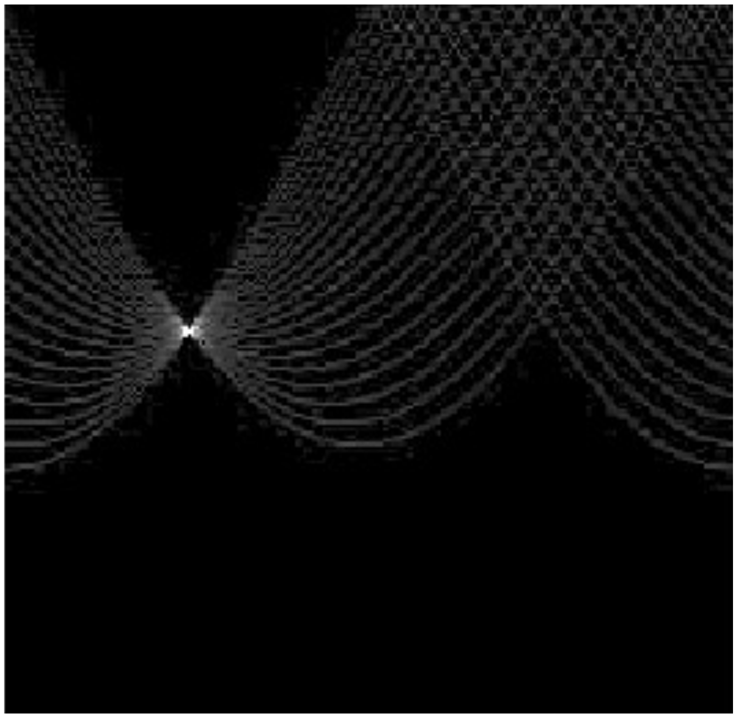


Image space



Votes

If images are noisy…

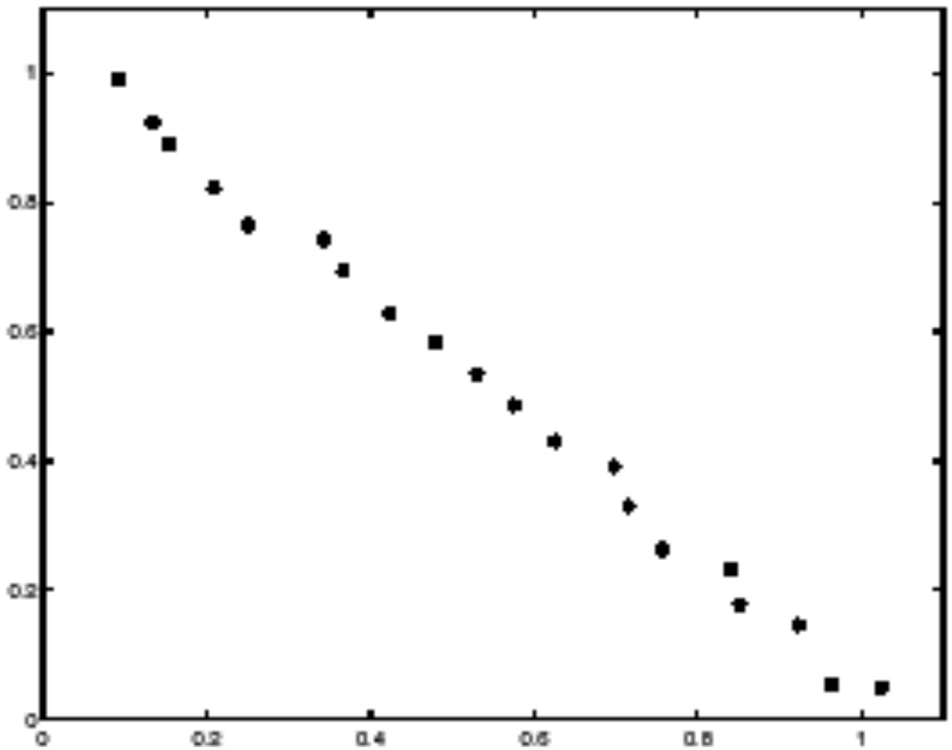
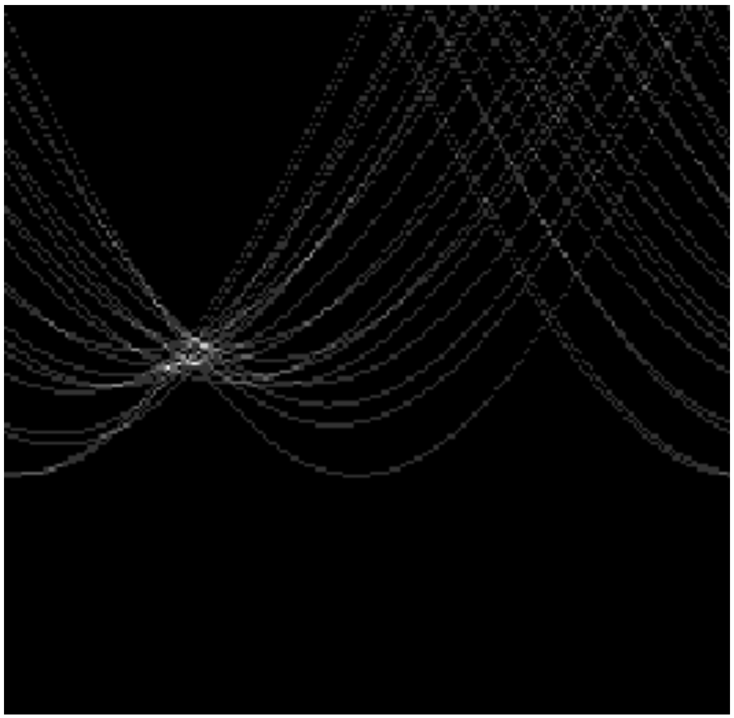


Image space



Votes

Too much noise

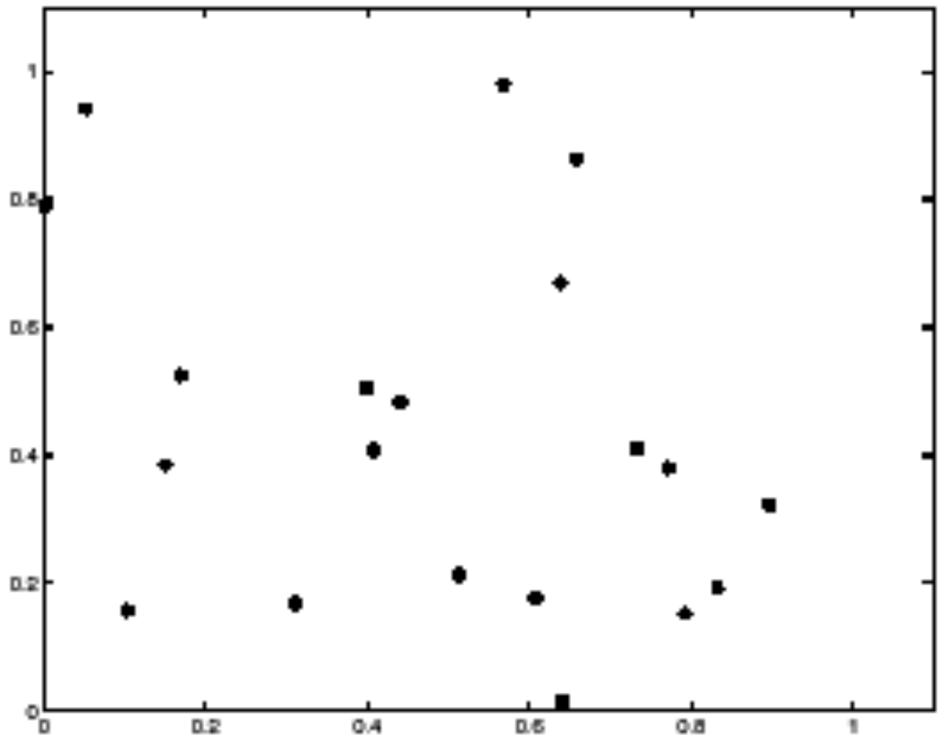
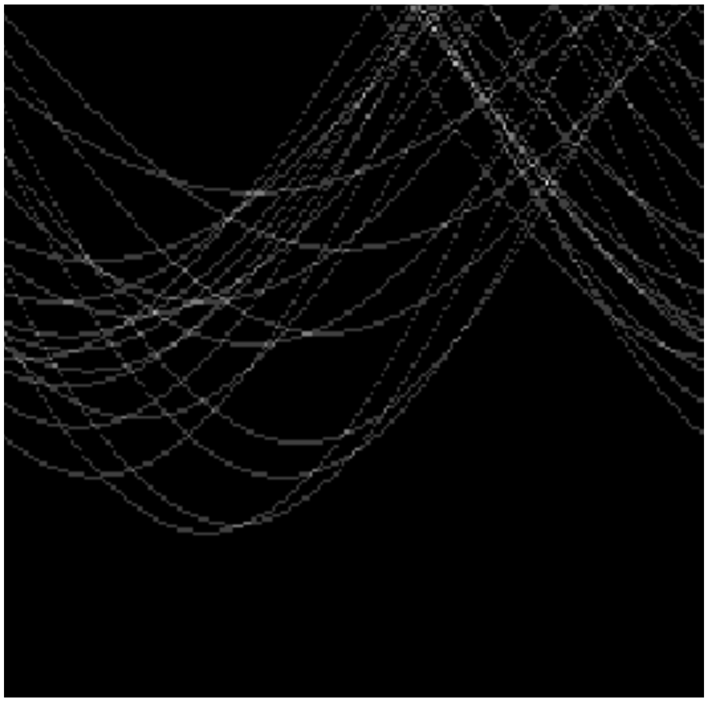


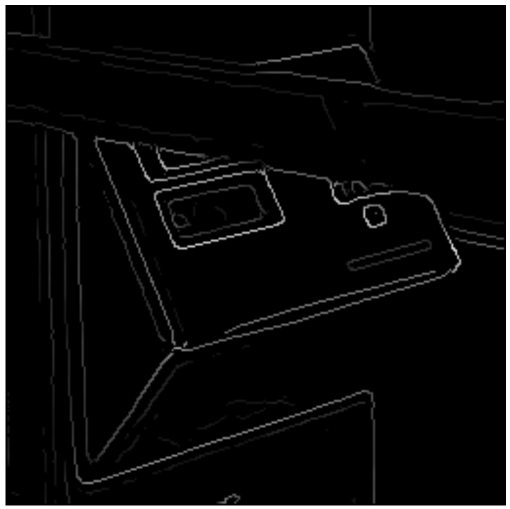
Image space



Votes

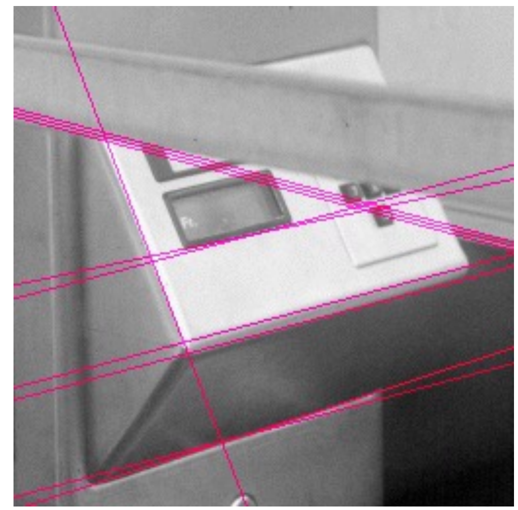
Real-world example

Original



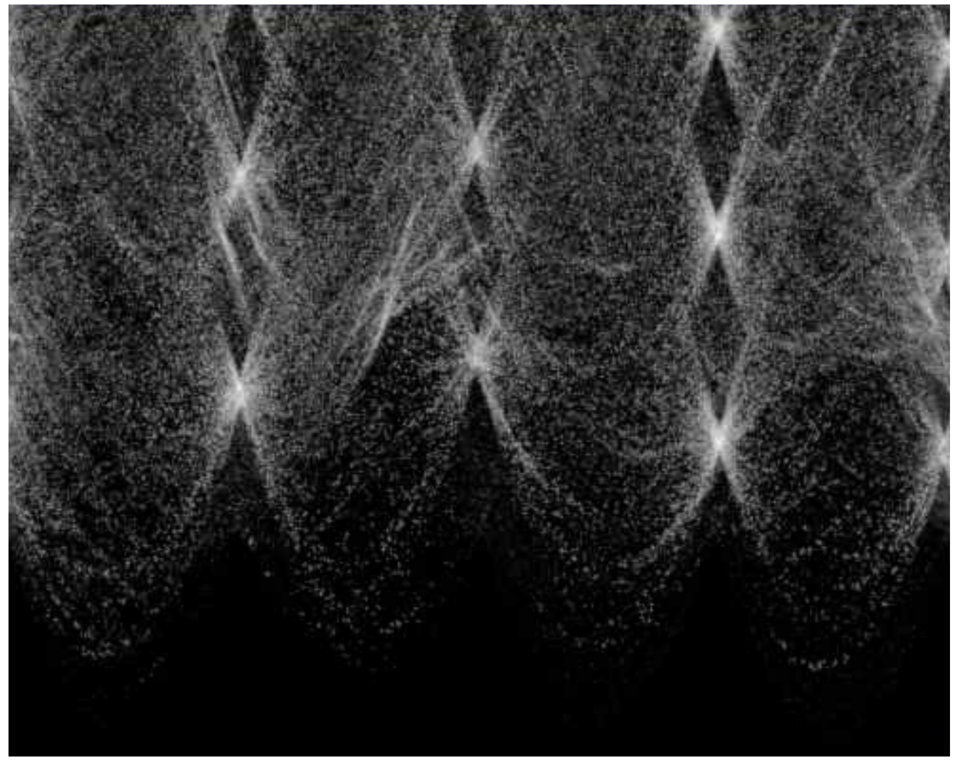
Edges

Parameter Space



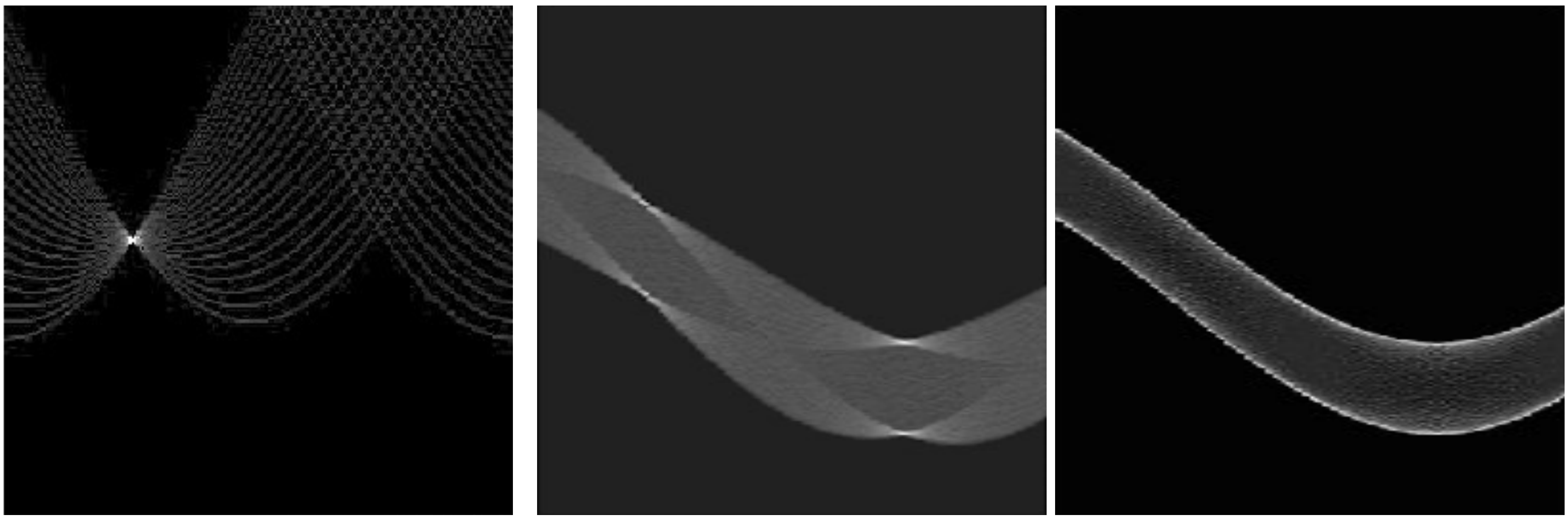
Hough Lines

More complex image



Basic Shapes

Parameter space



Line

rectangle
(parallelogram)

Circle

Can you guess the shape in image space?

Hough Circles

Let's assume **known** radius

parameters

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

variables

Fixed

parameters

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

variables

Fixed

What is the dimension of the parameter space?

SAIR
Spatial AI & Robotics Lab

Hough Circles

parameters

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

variables

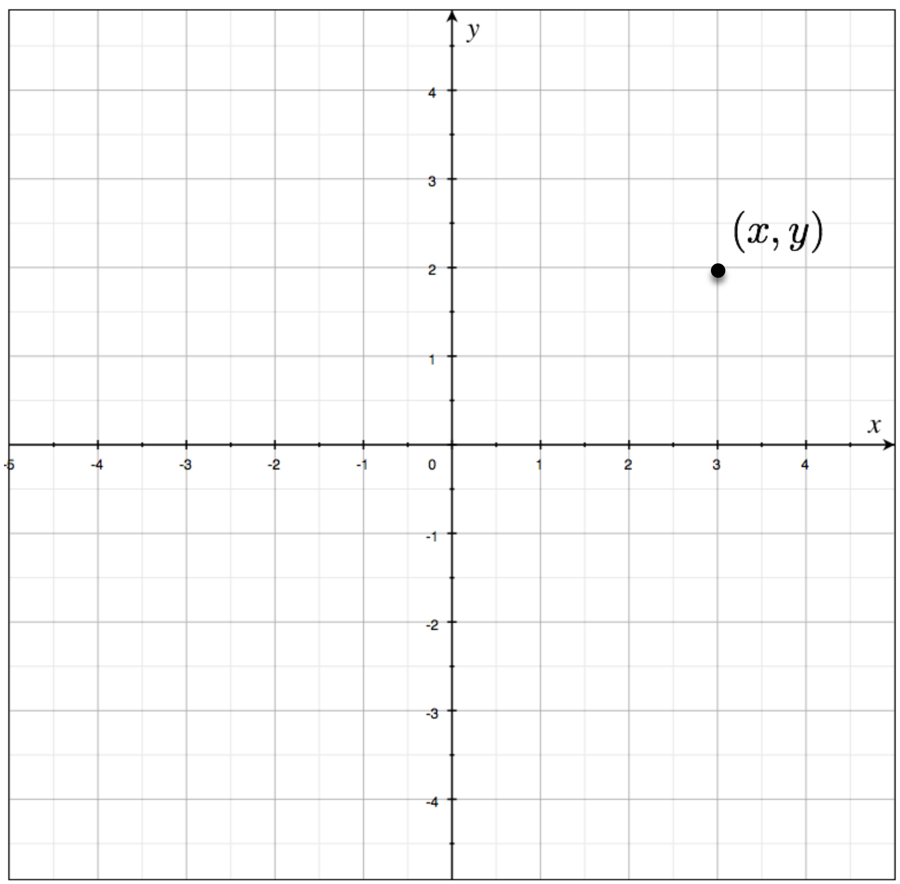
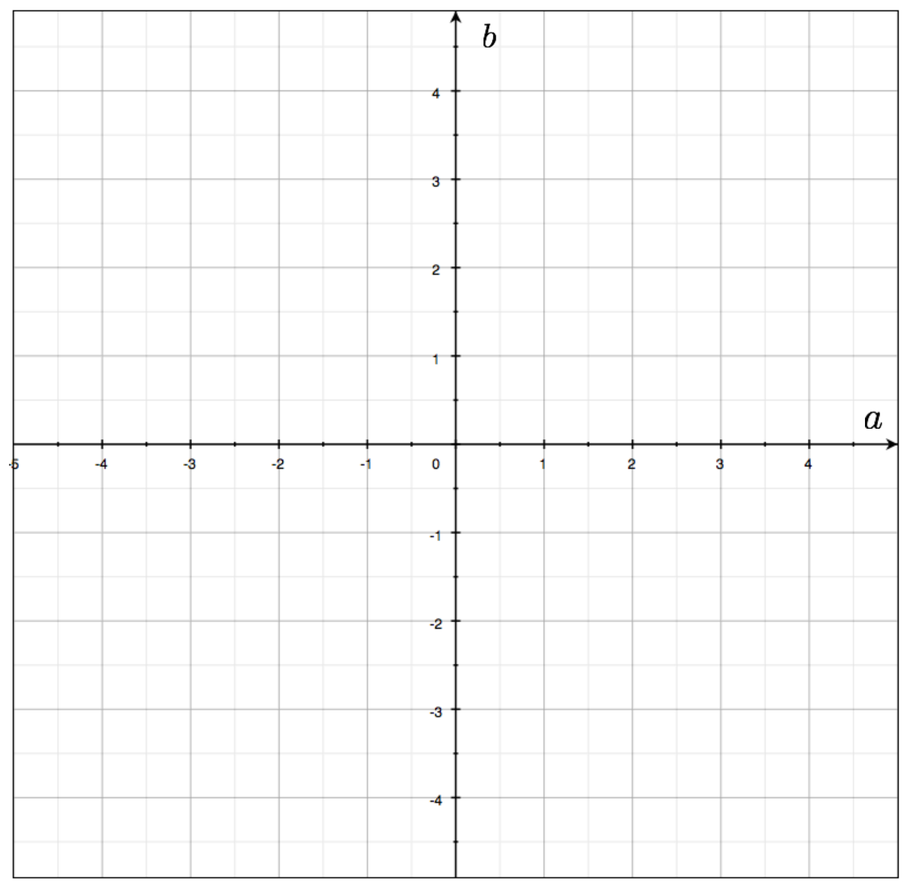


Image space

parameters

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

variables



Parameter space

SAIR Spatial AI & Robotics Lab

What does a point in image space correspond to in parameter space?

Hough Circles

parameters

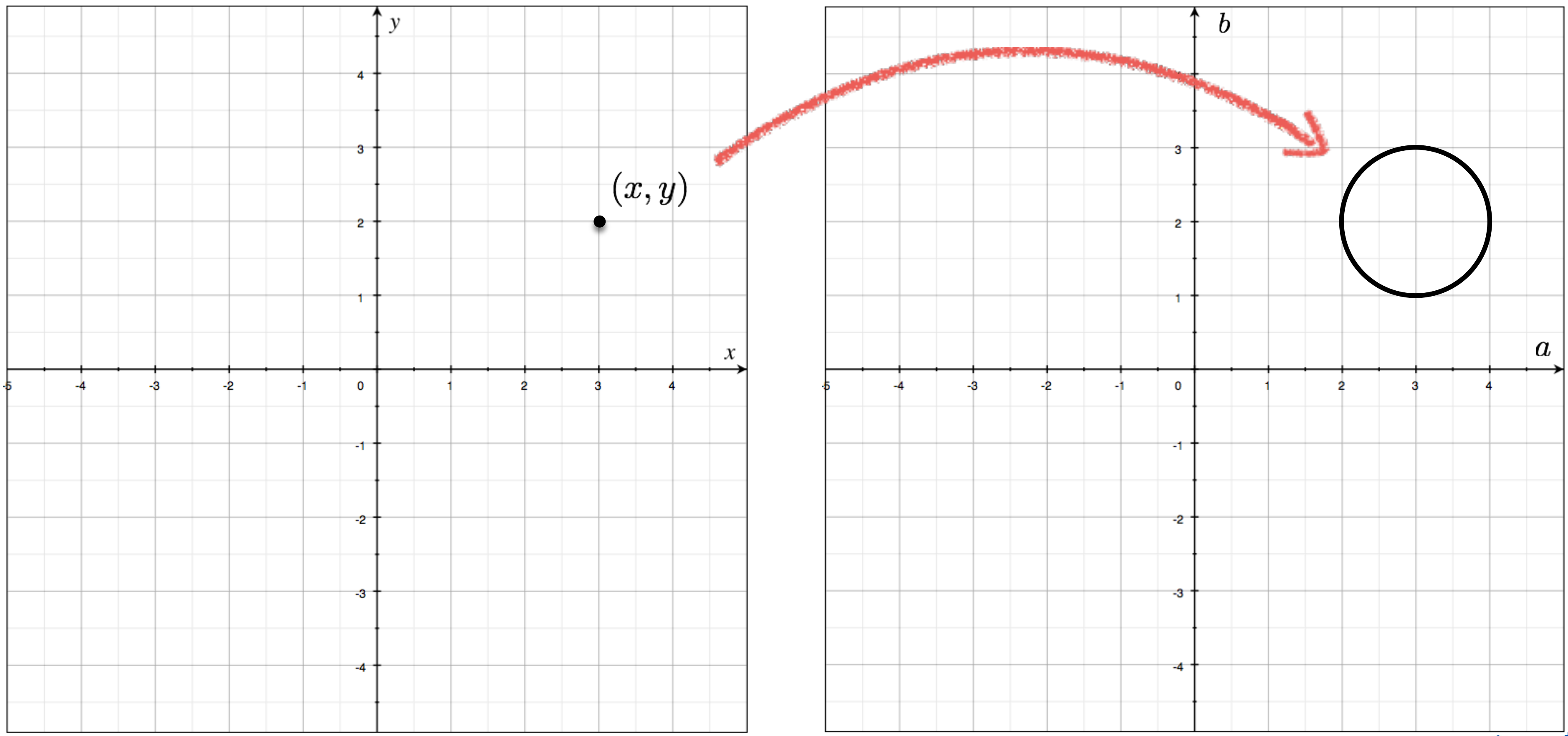
$$(x - a)^2 + (y - b)^2 = r^2$$

variables

parameters

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

variables



Hough Circles

parameters

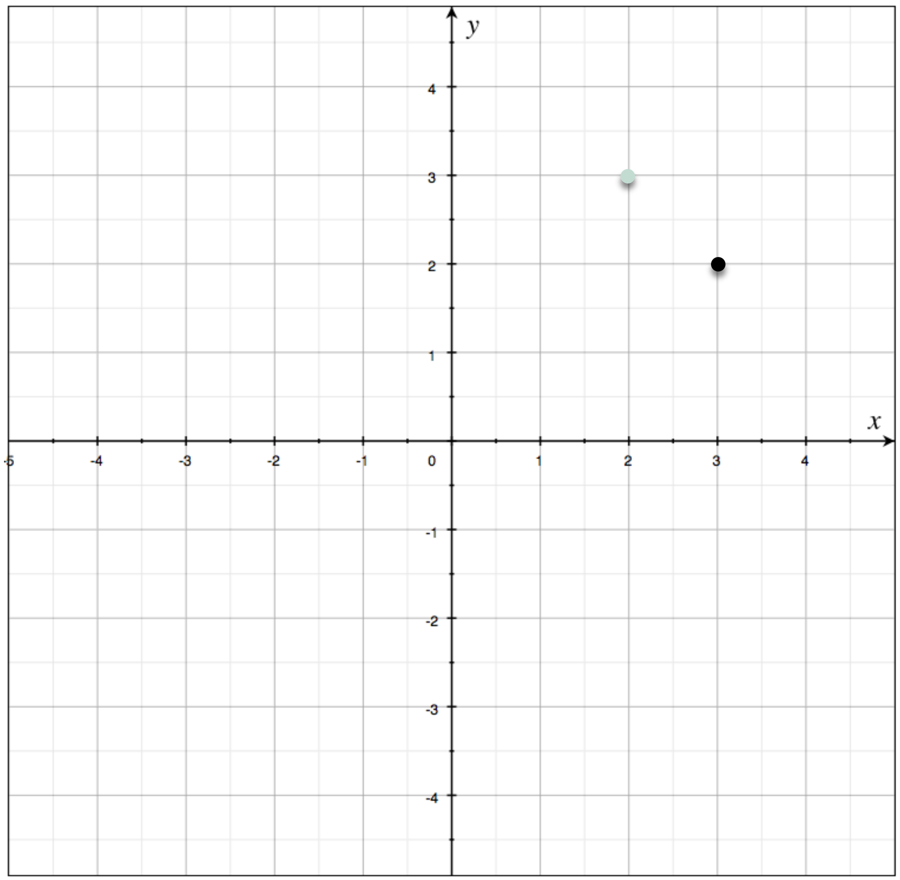
$$(x-a)^2+(y-b)^2=r^2$$

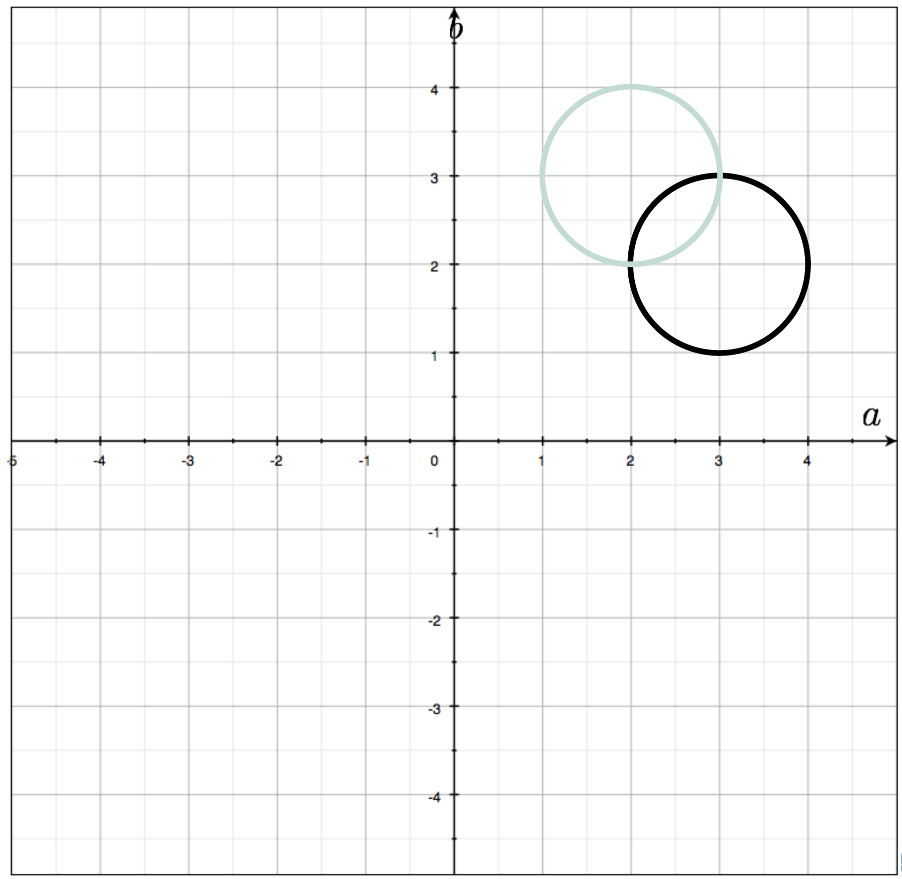
variables

parameters

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

variables





Hough Circles

parameters

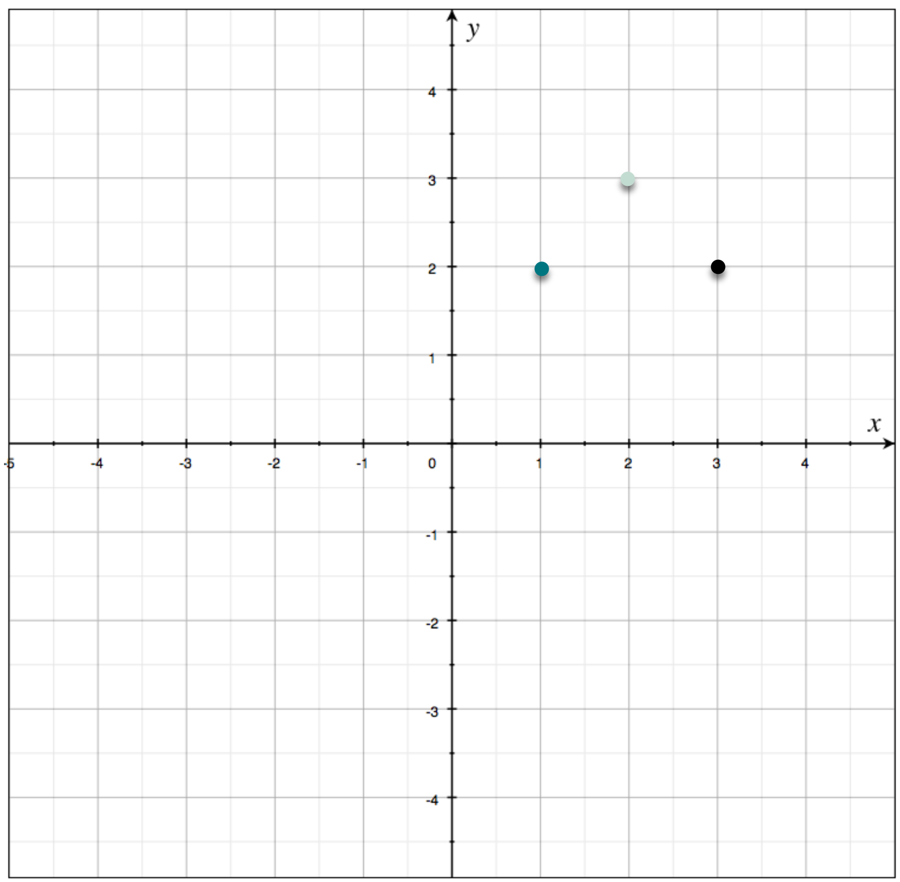
$$(x-a)^2+(y-b)^2=r^2$$

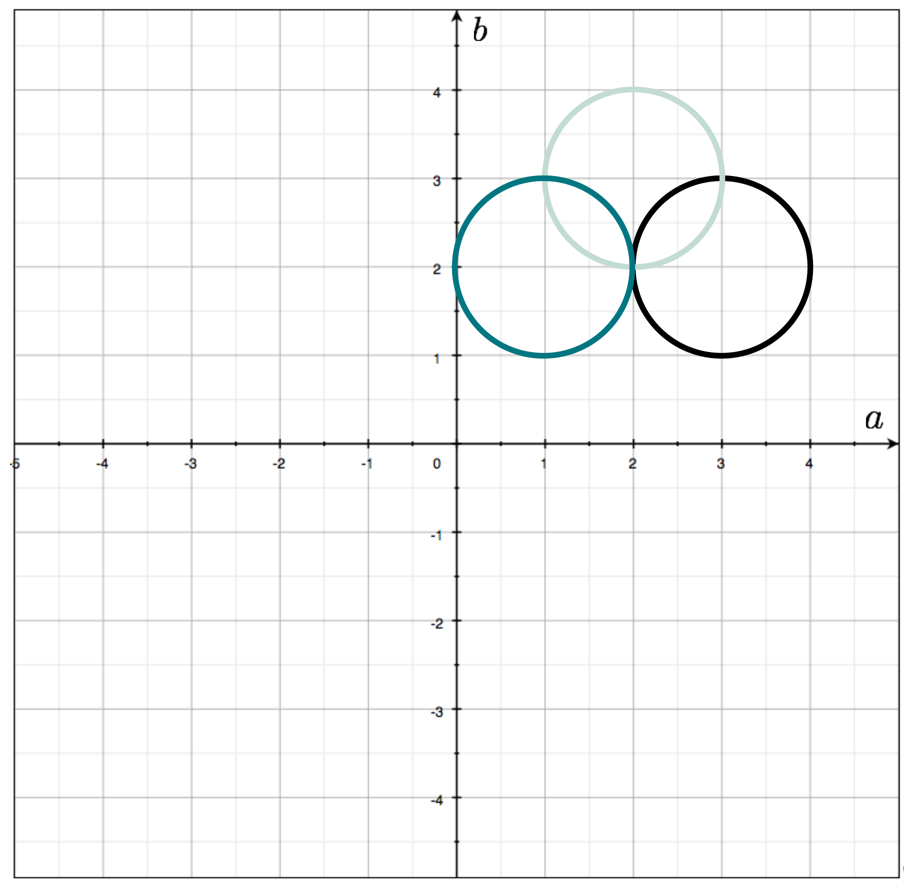
variables

parameters

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

variables





Hough Circles

parameters

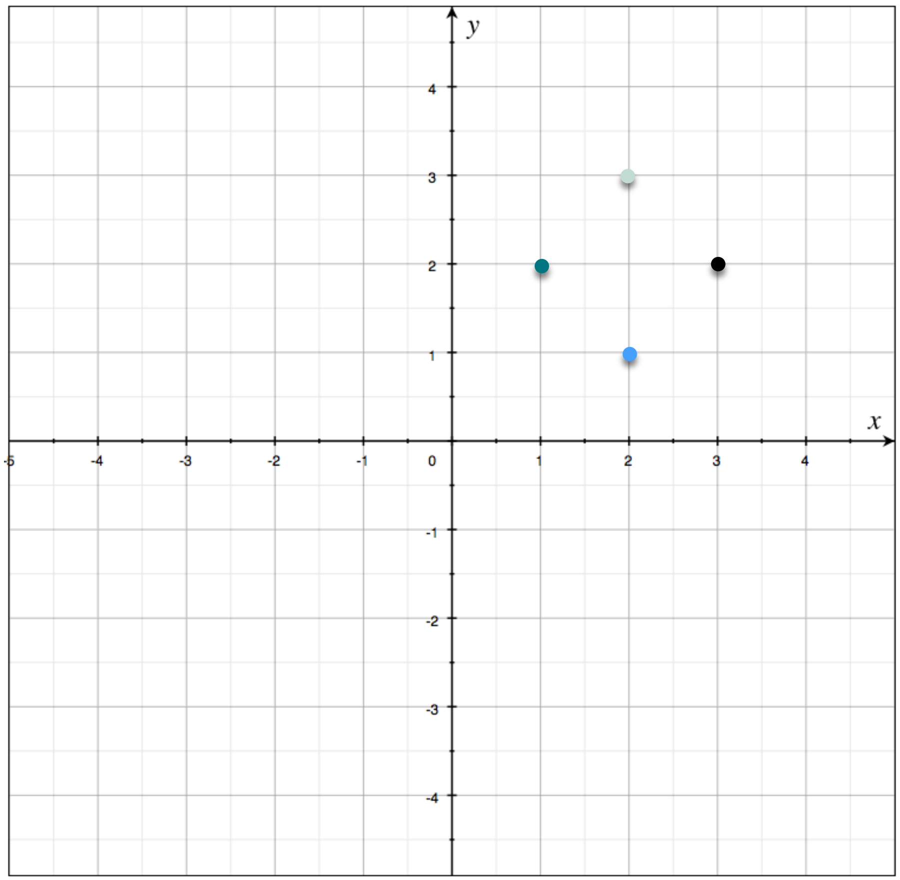
$$(x-a)^2+(y-b)^2=r^2$$

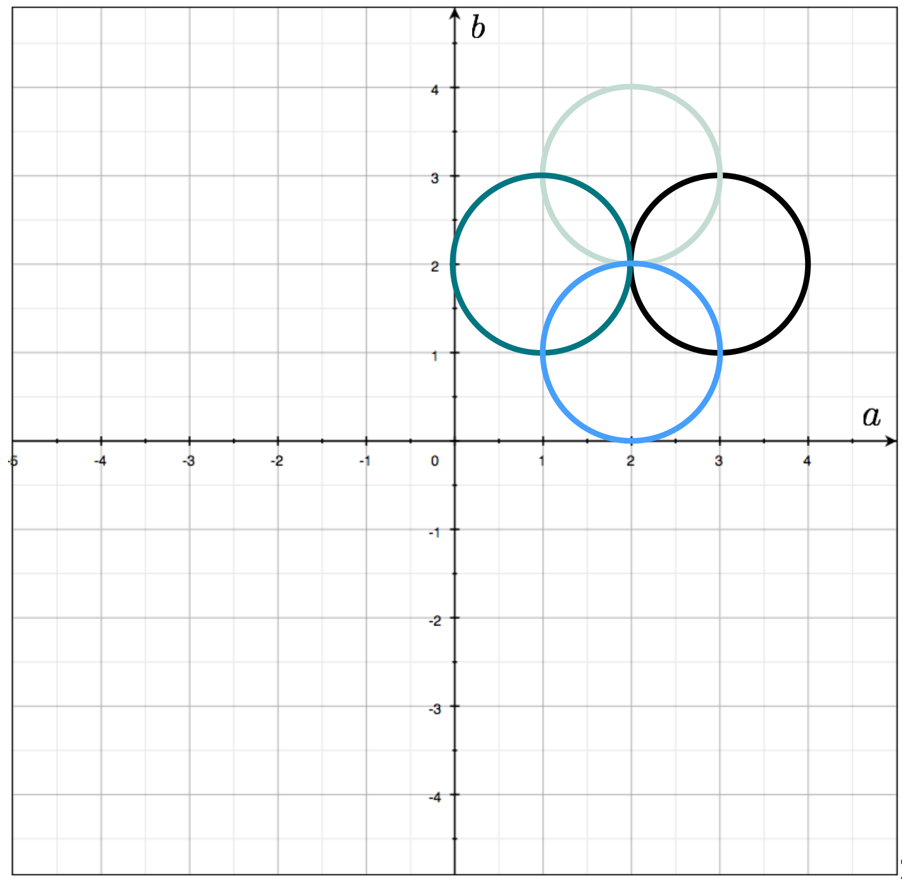
variables

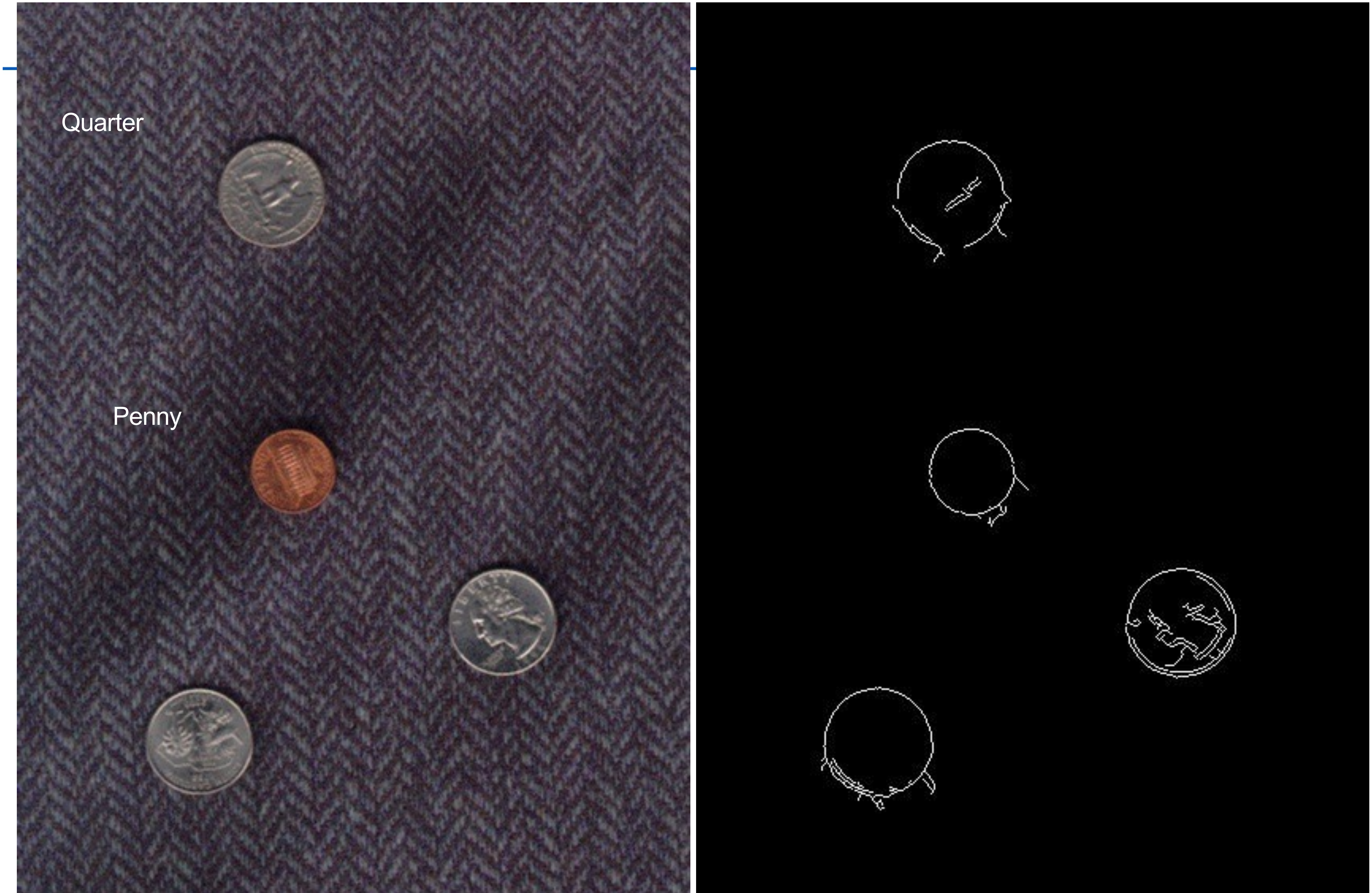
parameters

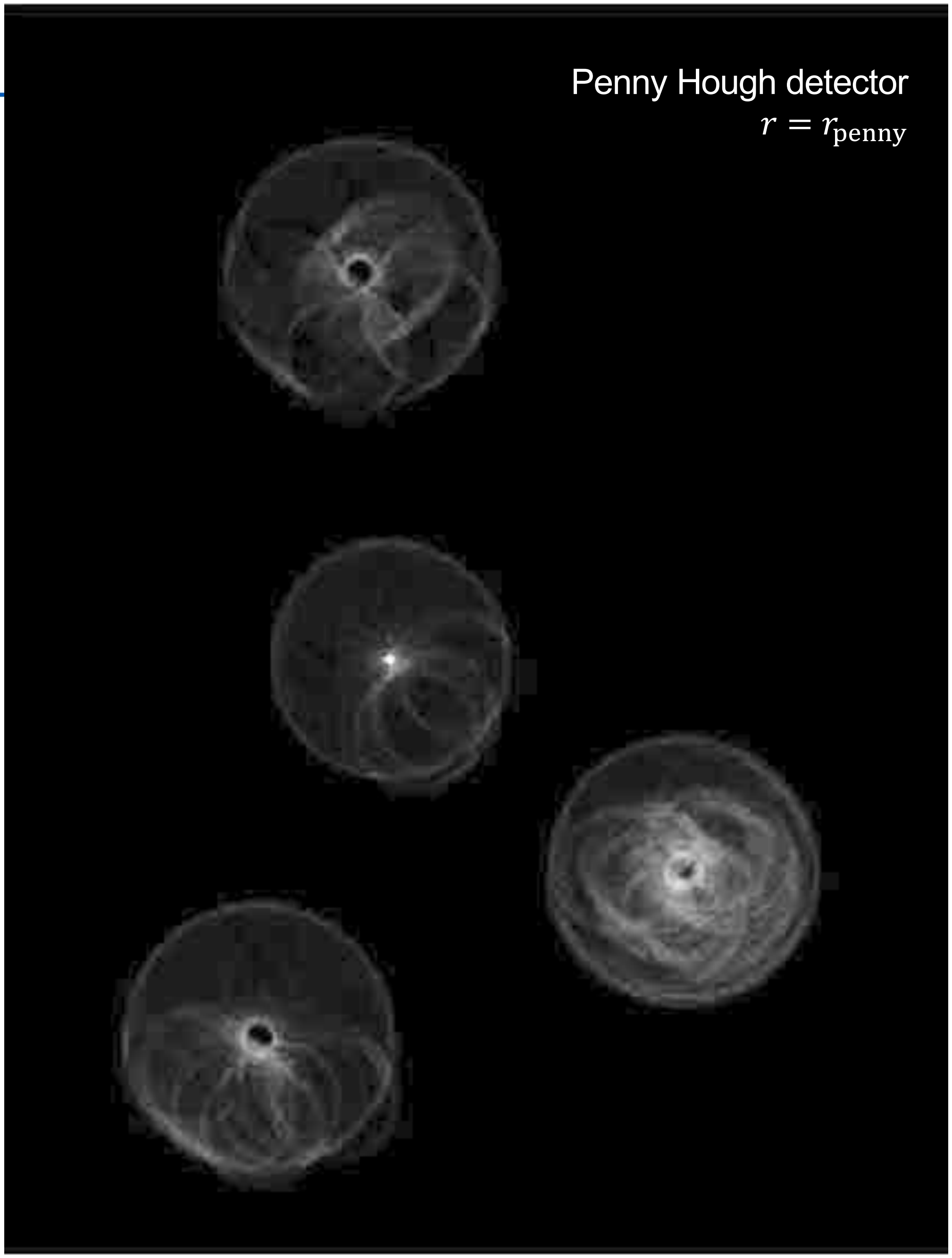
$$(x-a)^2+(y-b)^2=r^2$$

variables

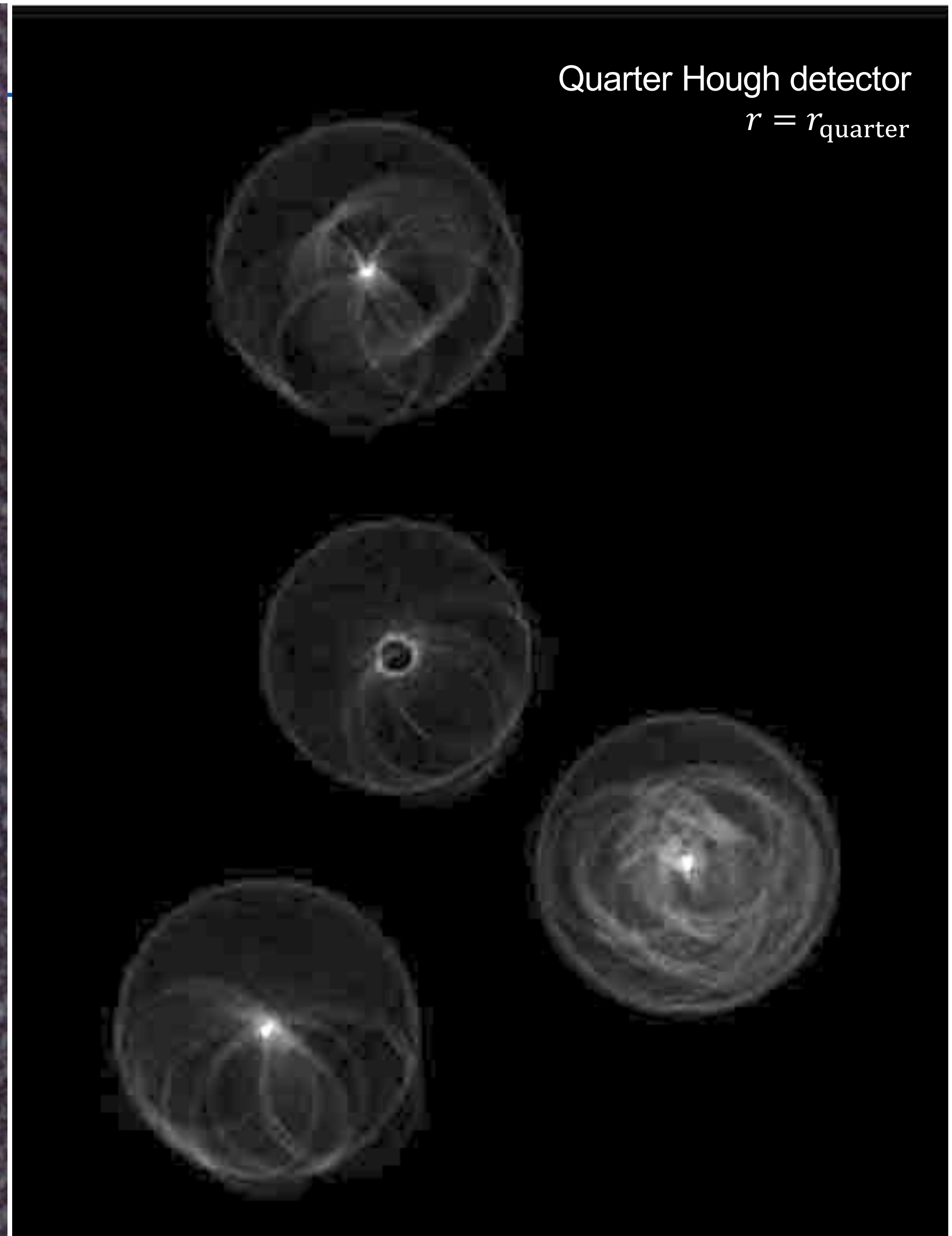


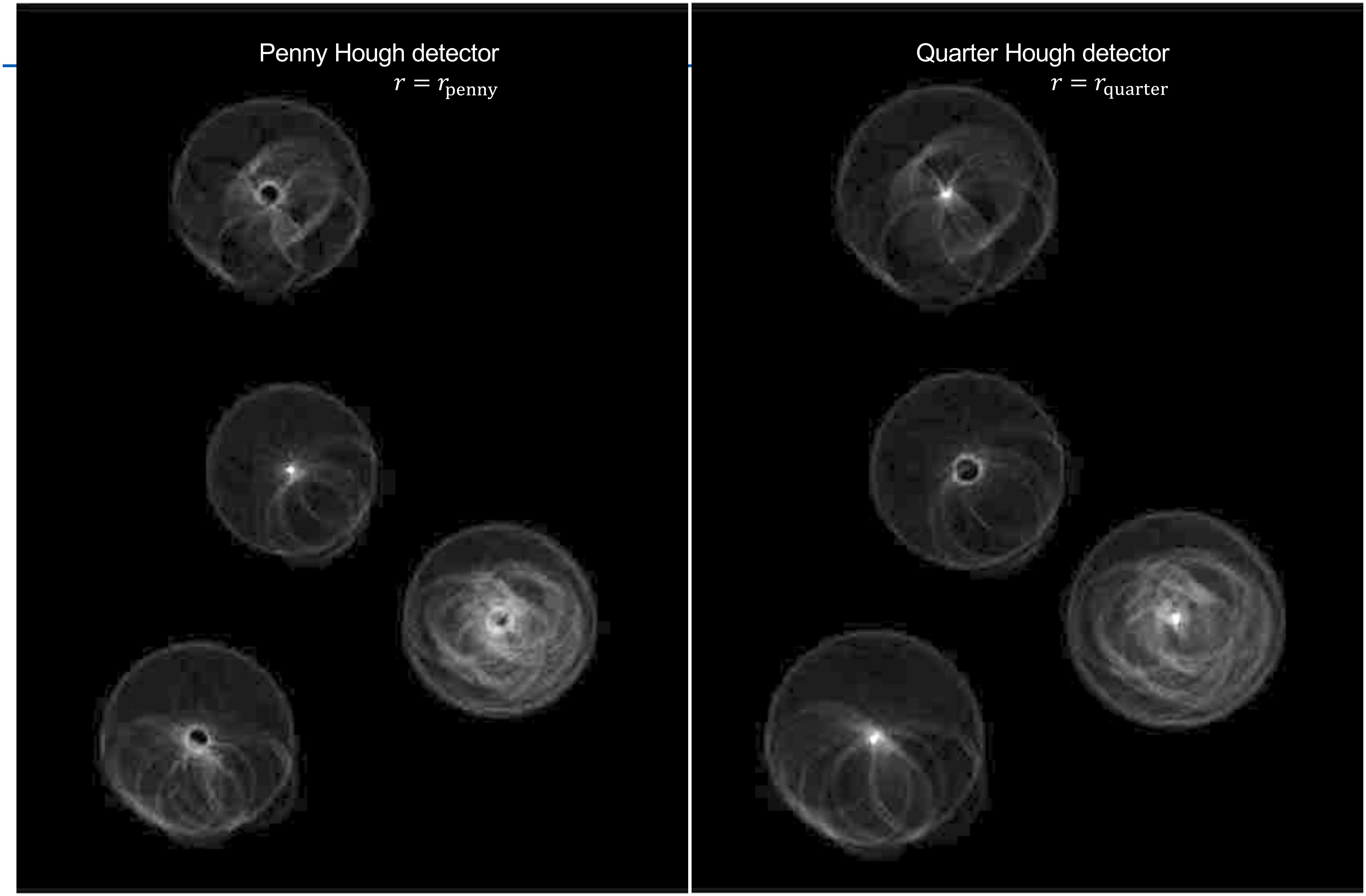


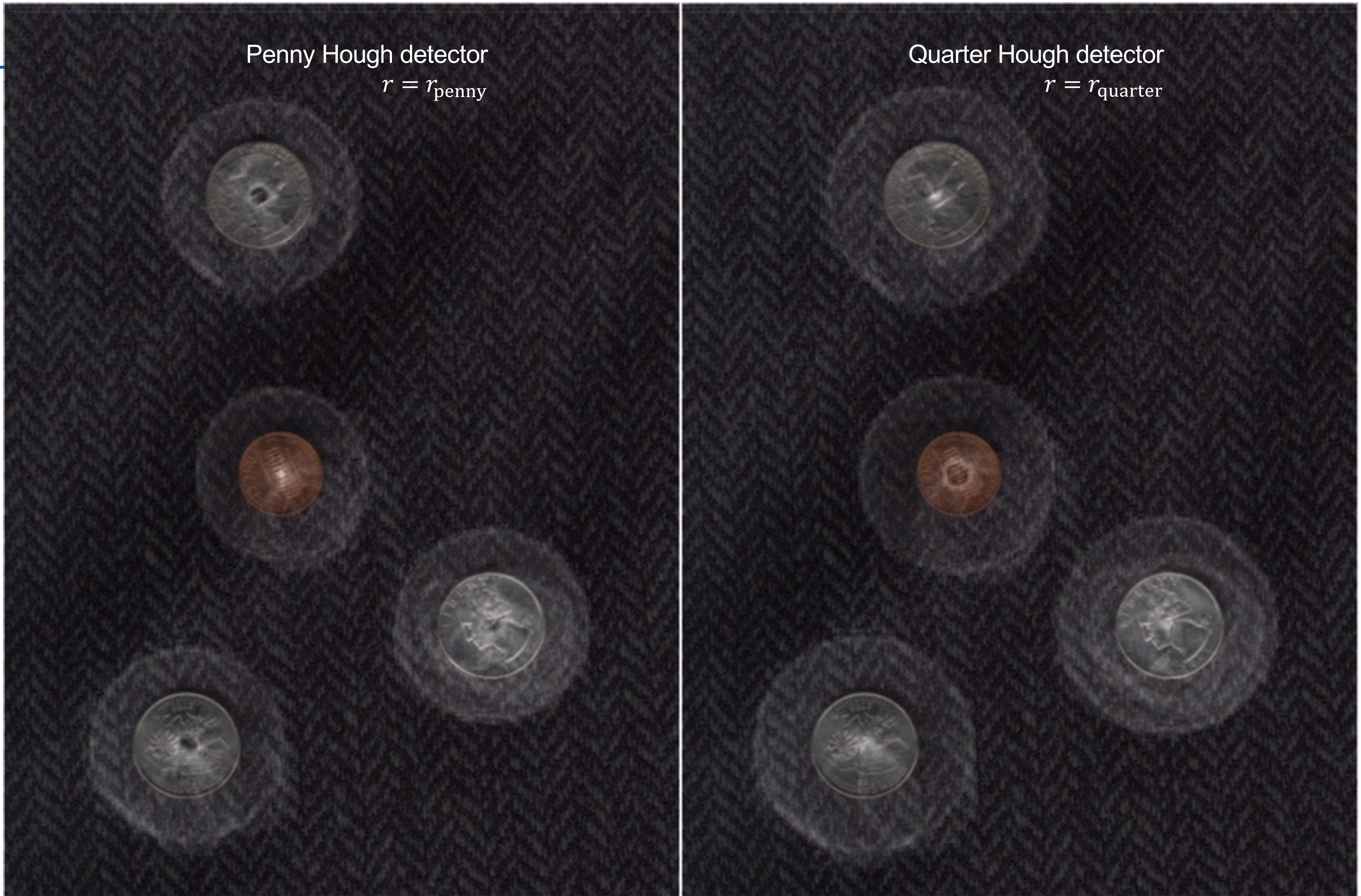

Quarter
Penny



Penny Hough detector
$r = r_{\text{penny}}$


Quarter Hough detector
$r = r_{\text{quarter}}$


Penny Hough detector
$r = r_{\text{penny}}$
Quarter Hough detector
$r = r_{\text{quarter}}$


Penny Hough detector
$r = r_{\text{penny}}$
Quarter Hough detector
$r = r_{\text{quarter}}$

What if radius is unknown?

parameters

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

variables

parameters

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

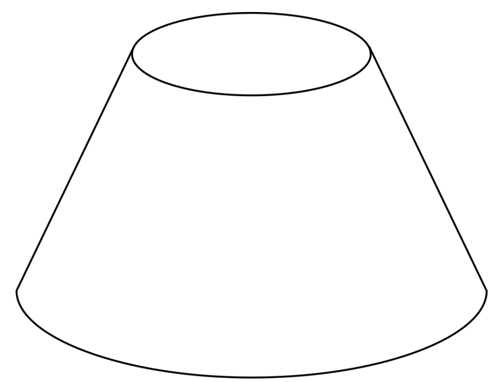
variables

If radius is unknown:

3D Hough Space!

Use Hough array $H(a,b,r)$.

Surface shape in Hough space is complicated.



Frustum of cone

SAIR Spatial AI & Robotics Lab

Other Shapes?

Vertical Ellipse:

parameters

$$\frac{(x-x_0)^2}{a^2}+\frac{(y-y_0)^2}{b^2}=1$$

$$H(x_0, y_0, a, b)$$

Ellipse:

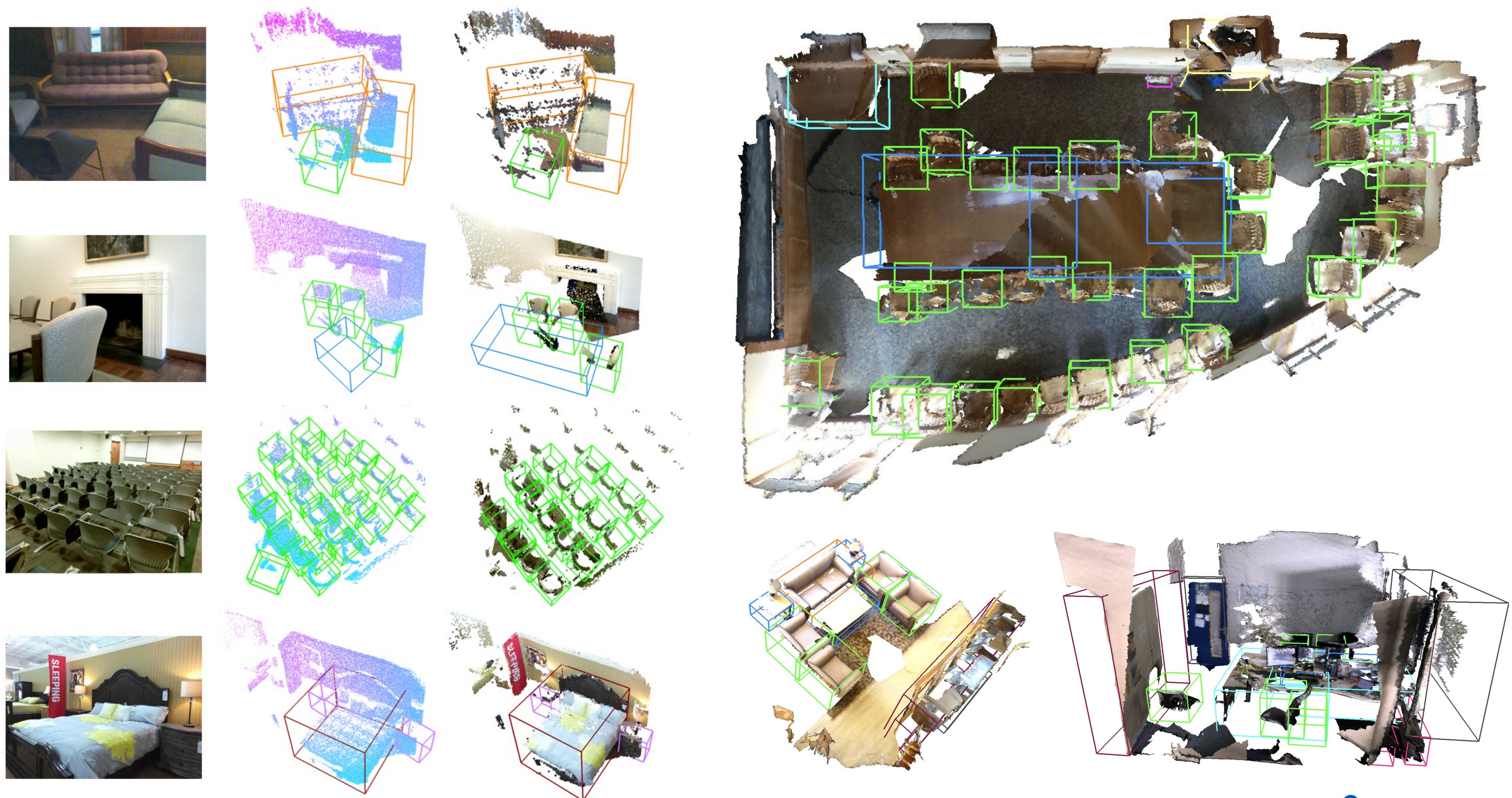
$$\frac{[(x-x_0)\cos\theta+(y-y_0)\sin\theta]^2}{a^2}+\frac{[-(x-x_0)\sin\theta+(y-y_0)\cos\theta]^2}{b^2}=1$$

$$H(x_0, y_0, a, b, \theta)$$

S A I R
Spatial AI & Robotics Lab

Applications of Hough Voting

Scenes Prediction Ground Truth



Qi, Charles R., et al. "Deep **Hough Voting** for 3D Object Detection in Point Clouds." *IEEE/CVF International Conference on Computer Vision (CVPR)*, 2019

Conclusion

Is the following correct about Hough transform …

- Detects multiple instances (lines/circles)?
- Robust to noise?
- Can be used for other shapes beyond lines/circles?
- Good computational complexity?
- Deals with occlusion well?

SAIR

Spatial AI & Robotics Lab

IMAGE PROCESSING

Alignment and Fitting

What are Fitting and Alignment?

- Fitting
  - find the parameters of a model that best fit the data
- Alignment
  - find the parameters of the transformation that best align matched points

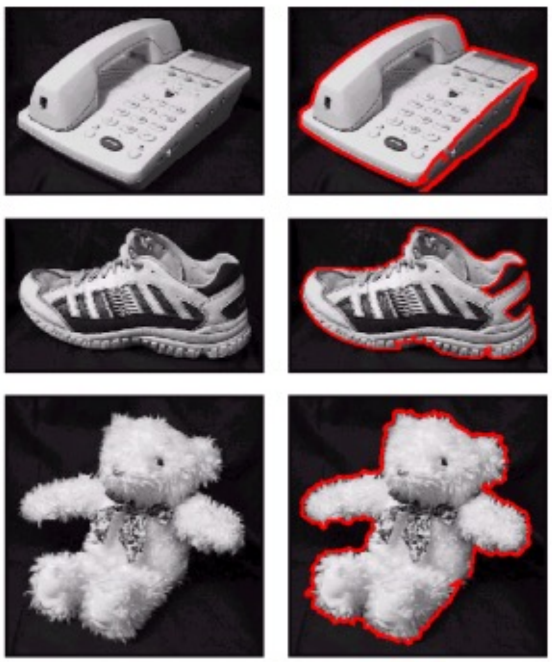
SAIR
Spatial AI & Robotics Lab

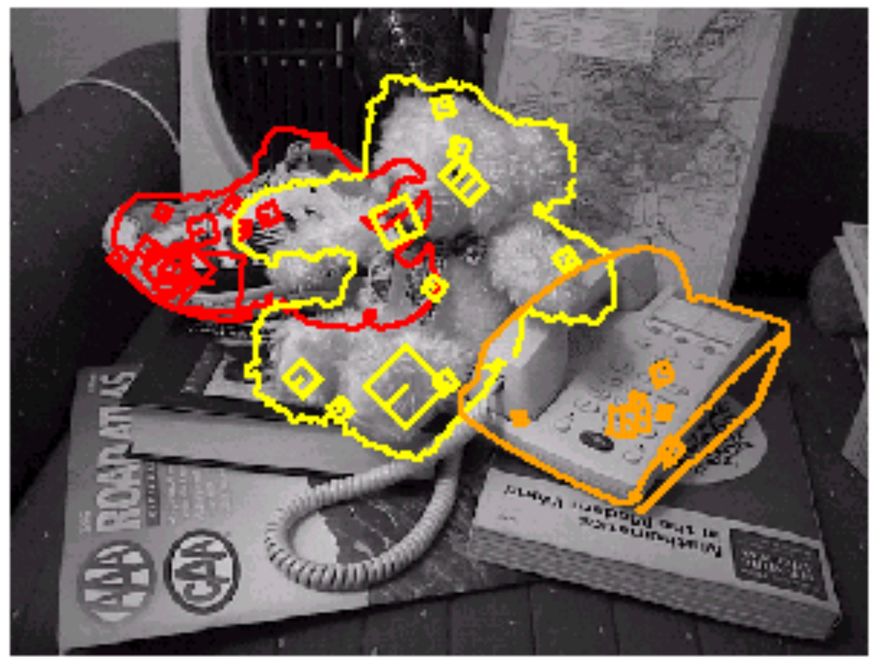
Fitting and Alignment: Methods

- Hypothesize and test
  - Generalized Hough transform
- General Alignment
  - Homographies
  - Rotational Panoramas
  - RANSAC
  - Global alignment
  - Warping
  - Blending
- Global optimization / Search for parameters
  - Least squares fit
  - Robust least squares
  - Other parameter search methods

SAIR
Spatial AI & Robotics Lab

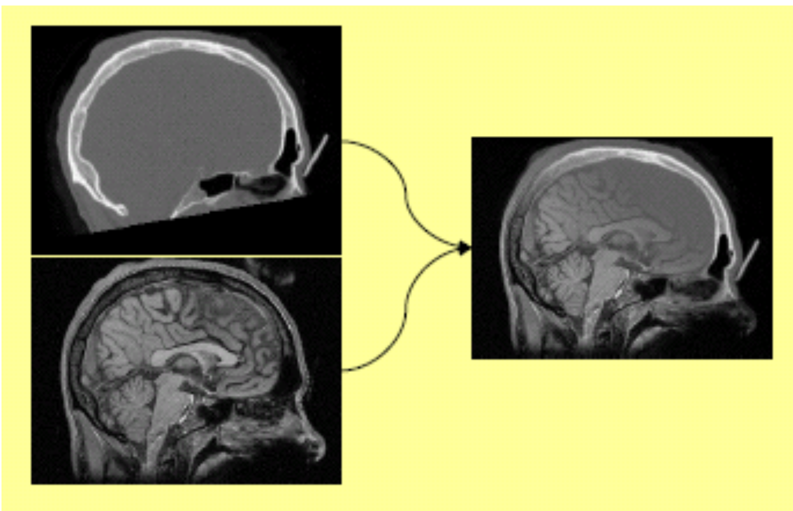
Motivation: Recognition

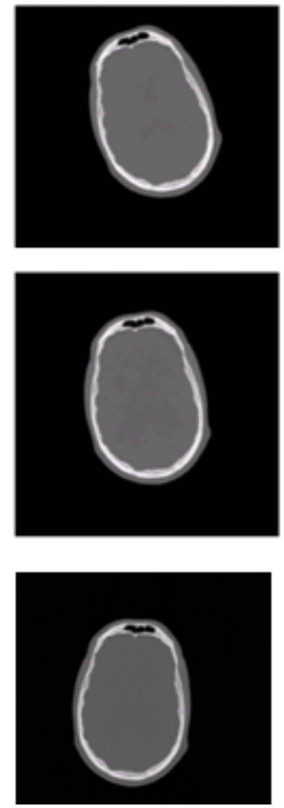




Figures from David Lowe

Motivation: Medical image registration





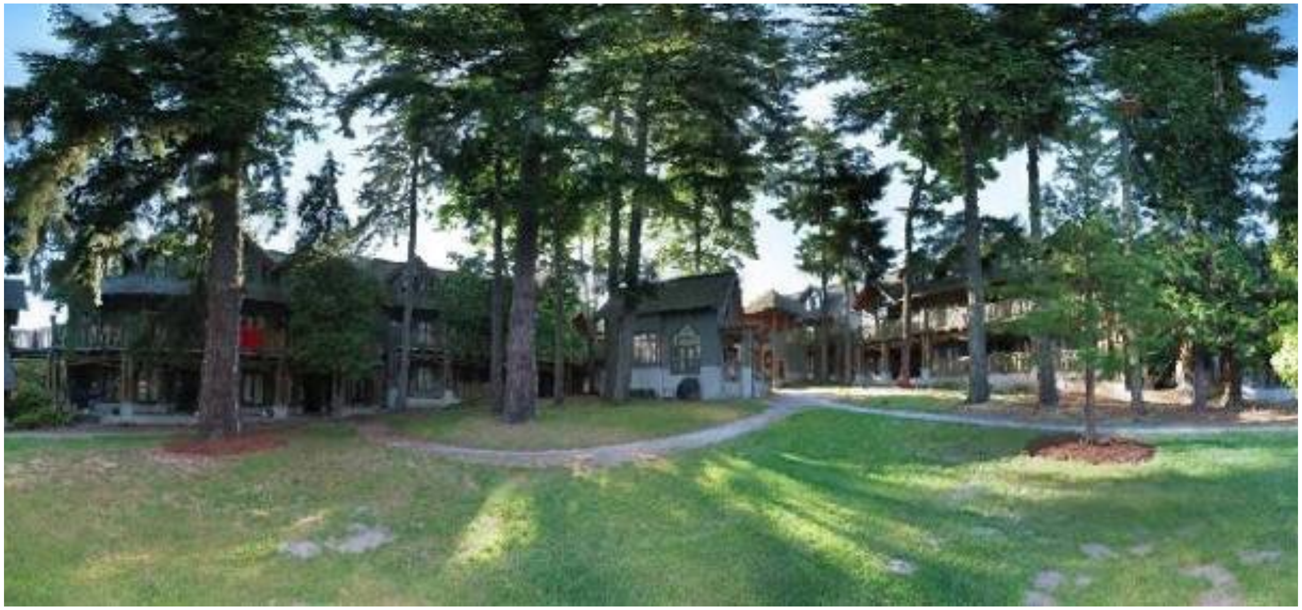
Motivation: Mosaics

- Getting the whole picture
  - Typical camera: 50° x 35°

Brown & Lowe 2003

Motivation: Mosaics

- Getting the whole picture
  - Typical camera: 50° x 35°
  - Human Vision: 176° x 135°



Brown & Lowe 2003

Motivation: Mosaics

- Getting the whole picture
  - Typical camera: 50° x 35°
  - Human Vision: 176° x 135°

Brown & Lowe 2003

Alignment

- Homography
- Rotational Panoramas
- RANSAC
- Global alignment
- Warping
- Blending

(a) (b)

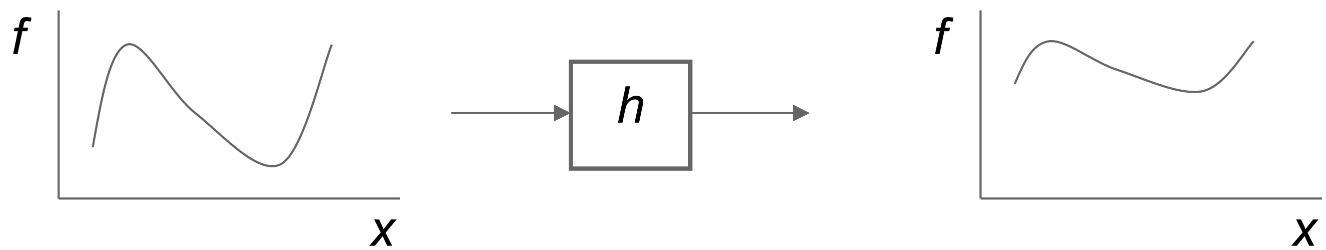
Motion models

- What happens when we take two images with a camera and try to align them?
- translation?
- rotation?
- scale?
- affine?
- perspective?

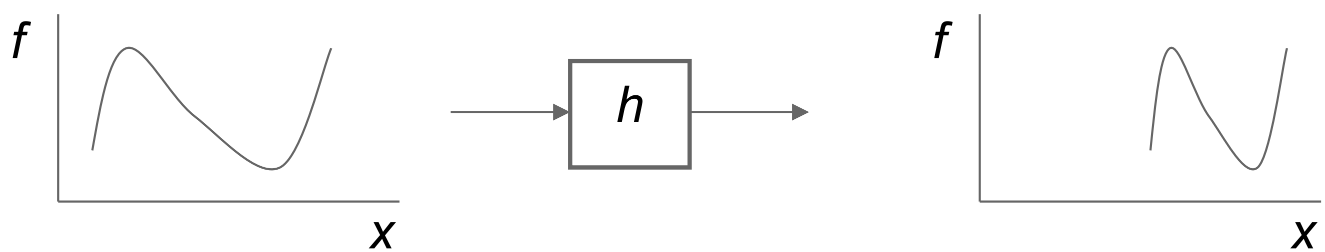
Szeliski

Image Warping (Recap)

- image filtering: change *range* of image
  - $g(x) = h(f(x))$



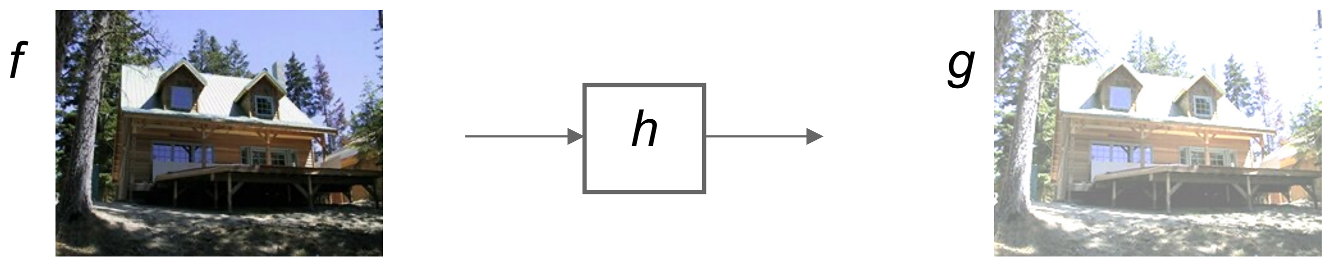
- image warping: change *domain* of image
  - $g(x) = f(h(x))$



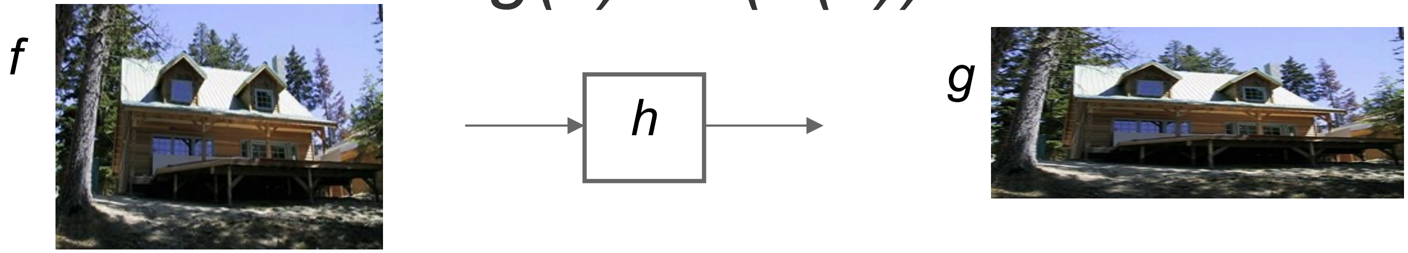
Szeliski

Image Warping

- image filtering: change *range* of image
  - $g(x) = h(f(x))$



- image warping: change *domain* of image
  - $g(x) = f(h(x))$



Parametric (global) warping

- Examples of parametric warps:

translation

rotation

aspect

affine

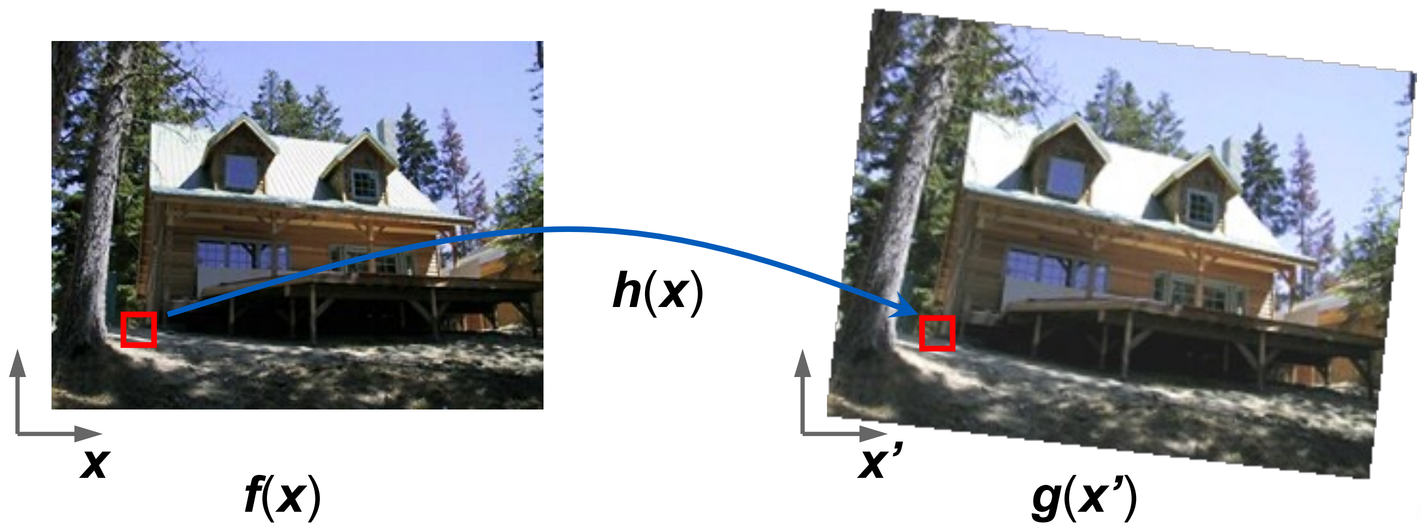
perspective

cylindrical

Szeliski

Image Warping

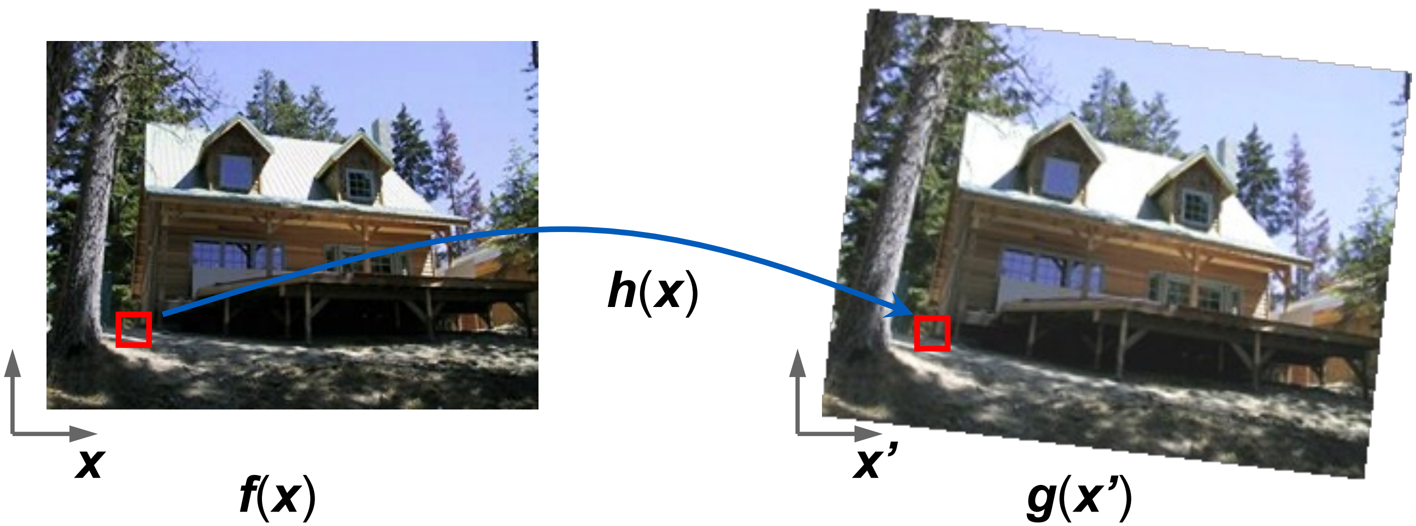
- Given a coordinate transform $\boldsymbol{x'} = \boldsymbol{h}(\boldsymbol{x})$ and a source image $\boldsymbol{f}(\boldsymbol{x})$, how do we compute a transformed image $\boldsymbol{g}(\boldsymbol{x'}) = \boldsymbol{f}(\boldsymbol{h}(\boldsymbol{x}))$?



Forward Warping

- Send each pixel $\boldsymbol{f}(\boldsymbol{x})$ to its corresponding location $\boldsymbol{x}' = \boldsymbol{h}(\boldsymbol{x})$ in $\boldsymbol{g}(\boldsymbol{x}')$

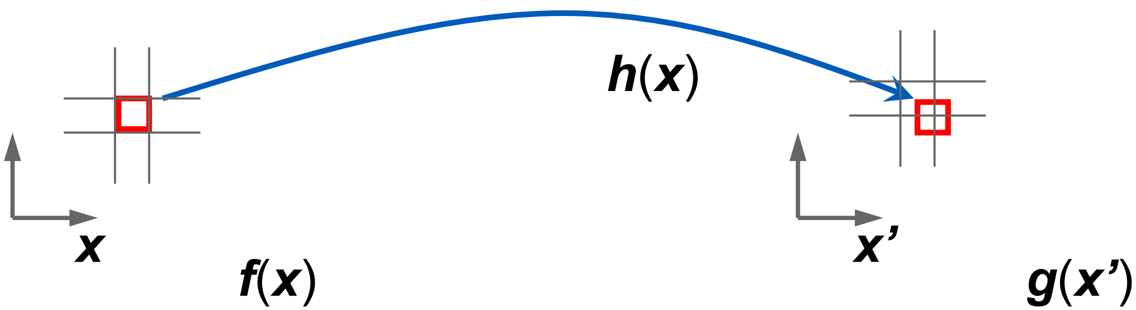
- What if pixel lands "between" two pixels?



Forward Warping

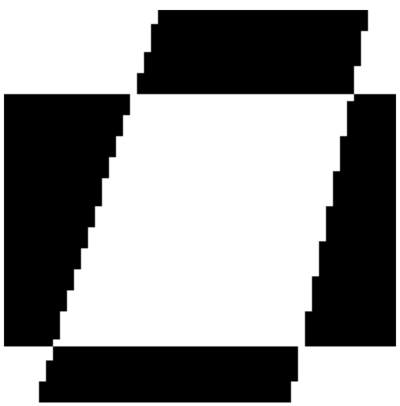
- Send each pixel ***f***(***x***) to its corresponding location ***x'*** = ***h***(***x***) in ***g***(***x'***)

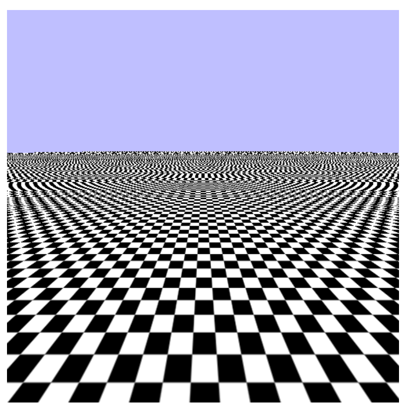
- What if pixel lands "between" two pixels?
- Answer: add "contribution" to several pixels, normalize later



Interpolation

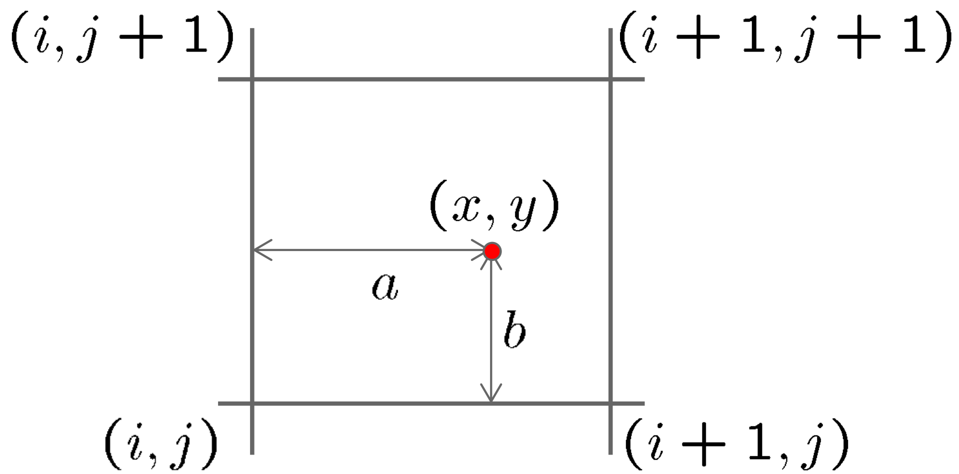
- Possible interpolation filters:
  - **Nearest Neighbor**
  - **Bilinear**
  - **Bicubic**
  - Sinc / FIR (band-limited)
- Needed to prevent "jaggies" and "texture crawl"





Bilinear interpolation

Sampling at *f(x,y):*



$$\begin{aligned} f(x,y) = \; & (1-a)(1-b) & f[i,j] \\ & +a(1-b) & f[i+1,j] \\ & +ab & f[i+1,j+1] \\ & +(1-a)b & f[i,j+1] \end{aligned}$$

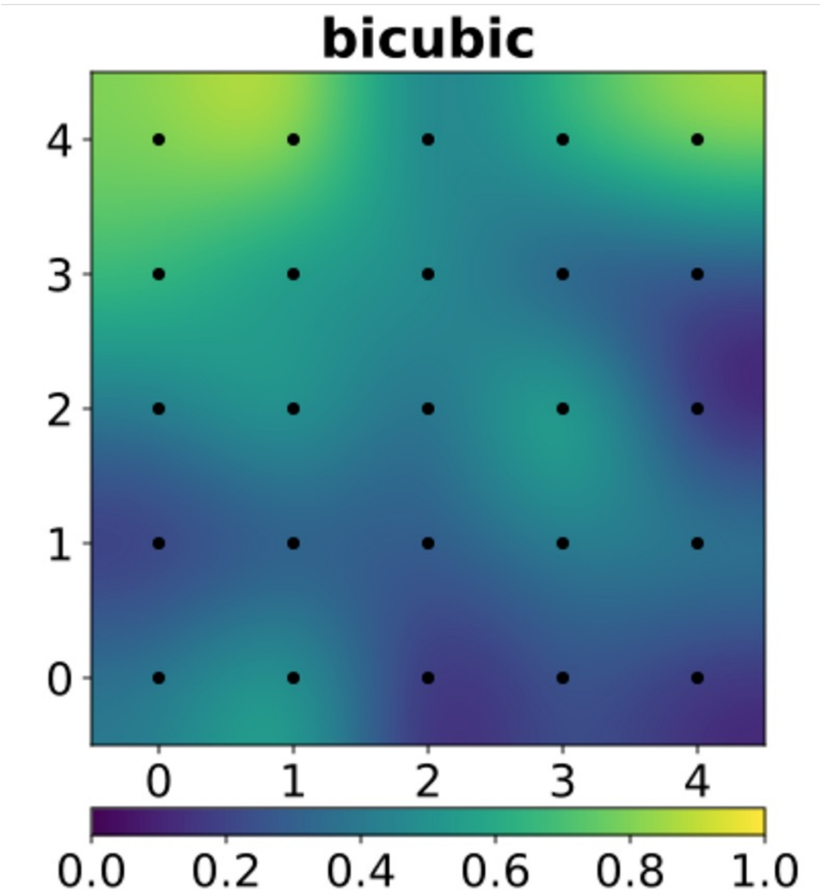
SAIR Spatial AI & Robotics Lab

Slide from Alyosha Efros

Reading more

Bicubic interpolation

- Bilinear interpolation processes 2x2 (4 pixels) squares
- Bicubic interpolation processes 4x4 (16 pixels) squares.



$$p(x,y)=\sum_{i=0}^{3}\sum_{j=0}^{3}a_{ij}x^iy^j.$$

This requires determining the 16 coefficients.

Consider 4 corners of the unit square.

(0, 0) (1, 0) (0, 1) (1, 1)

1. $f(0,0)=p(0,0)=a_{00},$
2. $f(1,0)=p(1,0)=a_{00}+a_{10}+a_{20}+a_{30},$
3. $f(0,1)=p(0,1)=a_{00}+a_{01}+a_{02}+a_{03},$
4. $f(1,1)=p(1,1)=\sum_{i=0}^{3}\sum_{j=0}^{3}a_{ij}.$

If the quality is of concern, bicubic would be the best choice.

2D coordinate transformations

- translation: $\mathbf{x}' = \mathbf{x} + \mathbf{t}$ $\qquad \mathbf{x} = (x, y)$
- rotation: $\mathbf{x}' = \mathbf{R}\,\mathbf{x} + \mathbf{t}$
- similarity: $\mathbf{x}' = s\,\mathbf{R}\,\mathbf{x} + \mathbf{t}$
- affine: $\mathbf{x}' = \mathbf{A}\,\mathbf{x} + \mathbf{t}$
- perspective: $\underline{\mathbf{x}}' \cong \mathbf{H}\,\underline{\mathbf{x}}$ $\qquad \underline{\mathbf{x}} = (x, y, 1)$

  ($\underline{\mathbf{x}}$ is a *homogeneous* coordinate)

- These all form a nested *group* (closed w/ inv.)

SAIR Spatial AI & Robotics Lab

Szeliski

Basic 2D Transformations

- Basic 2D transformations as 3x3 matrices

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Translate

$$\begin{bmatrix} \boldsymbol{x}' \\ \boldsymbol{y}' \\ 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{s}_{\boldsymbol{x}} & 0 & 0 \\ 0 & \boldsymbol{s}_{\boldsymbol{y}} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{y} \\ 1 \end{bmatrix}$$

Scale

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\Theta & -\sin\Theta & 0 \\ \sin\Theta & \cos\Theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Rotate

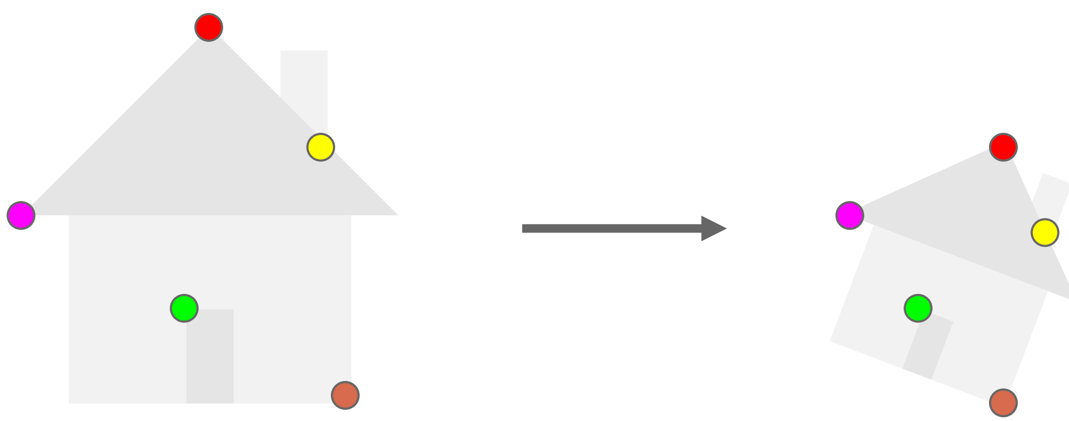
$$\begin{bmatrix} \boldsymbol{x}' \\ \boldsymbol{y}' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & \boldsymbol{sh}_{\boldsymbol{x}} & 0 \\ \boldsymbol{sh}_{\boldsymbol{y}} & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{y} \\ 1 \end{bmatrix}$$

Shear (Skew)

Source: Alyosha Efros

Image alignment

- Two broad approaches:
  - Direct (pixel-based) alignment
    - Search for alignment where most pixels agree
  - Feature-based alignment
    - Search for alignment where *extracted features* agree
    - Can be verified using pixel-based alignment



Source: L. Lazebnik

Fitting an affine transformation

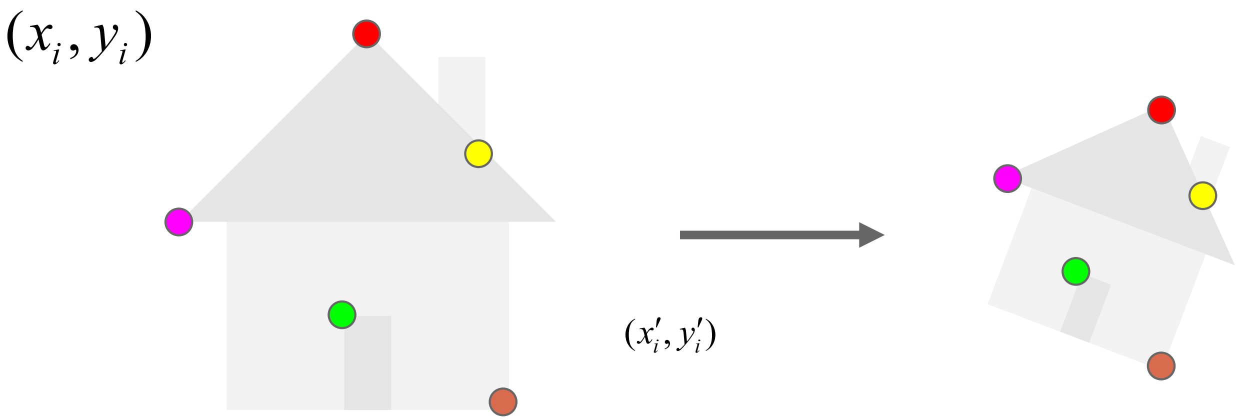
Affine model approximates perspective projection of planar objects.

Figures from David Lowe, ICCV 1999

SAIR
Spatial AI & Robotics Lab

Fitting an affine transformation

- Assuming we know the correspondences, how do we get the transformation?

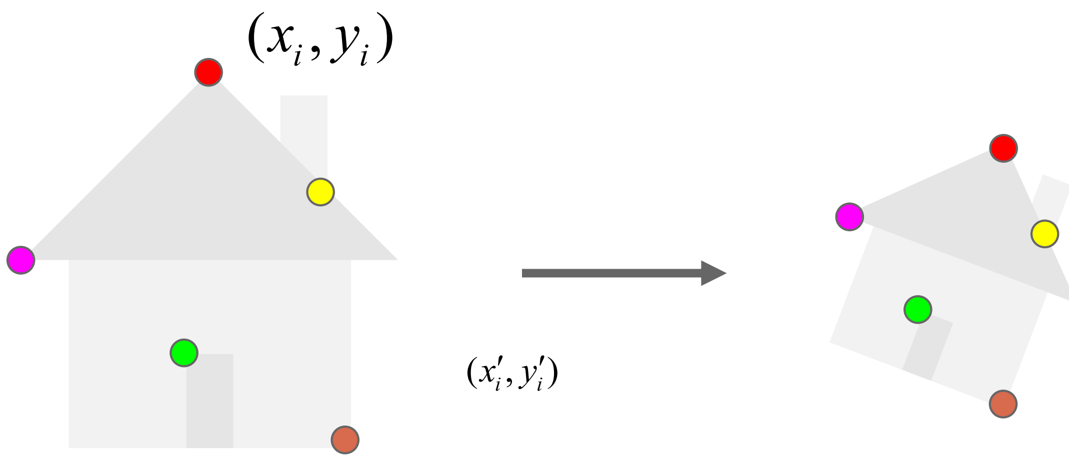


$$\begin{bmatrix} x_i' \\ y_i' \end{bmatrix} = \begin{bmatrix} m_1 & m_2 \\ m_3 & m_4 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix} + \begin{bmatrix} t_1 \\ t_2 \end{bmatrix}$$

Grauman

Fitting an affine transformation

- Assuming we know the correspondences, how do we get the transformation?



$$\begin{bmatrix} x_i' \\ y_i' \end{bmatrix} = \begin{bmatrix} m_1 & m_2 \\ m_3 & m_4 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix} + \begin{bmatrix} t_1 \\ t_2 \end{bmatrix}$$

$$\begin{bmatrix} \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \\ t_1 \\ t_2 \end{bmatrix} = \begin{bmatrix} \end{bmatrix}$$

SAIR
Spatial AI & Robotics Lab

Fitting an affine transformation

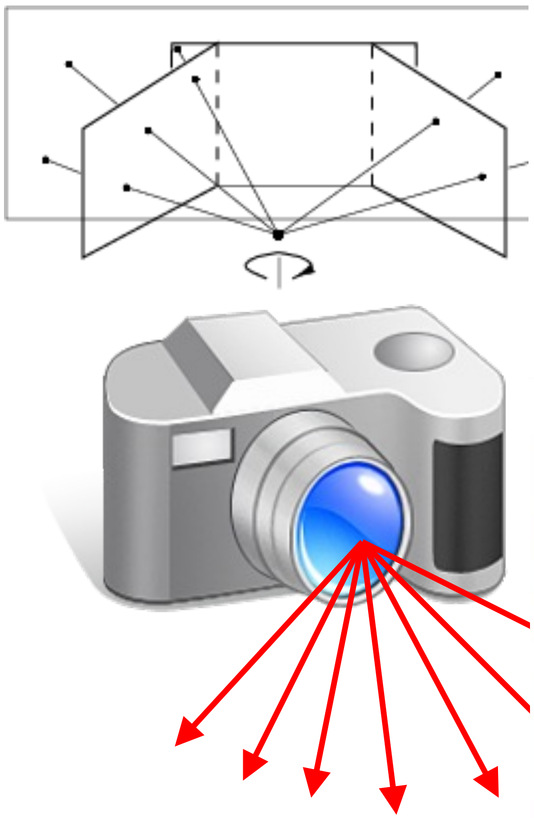
$$\begin{bmatrix} & & \cdots & & & \\ x_i & y_i & 0 & 0 & 1 & 0 \\ 0 & 0 & x_i & y_i & 0 & 1 \\ & & \cdots & & & \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \\ t_1 \\ t_2 \end{bmatrix} = \begin{bmatrix} \cdots \\ x_i' \\ y_i' \\ \cdots \end{bmatrix}$$

- How many matches (correspondence pairs) do we need to solve for the transformation parameters?
- Once we have solved for the parameters, how do we compute the coordinates of the corresponding point for (x_{new}, y_{new})?

SAIR Spatial AI & Robotics Lab

Grauman

Panoramas



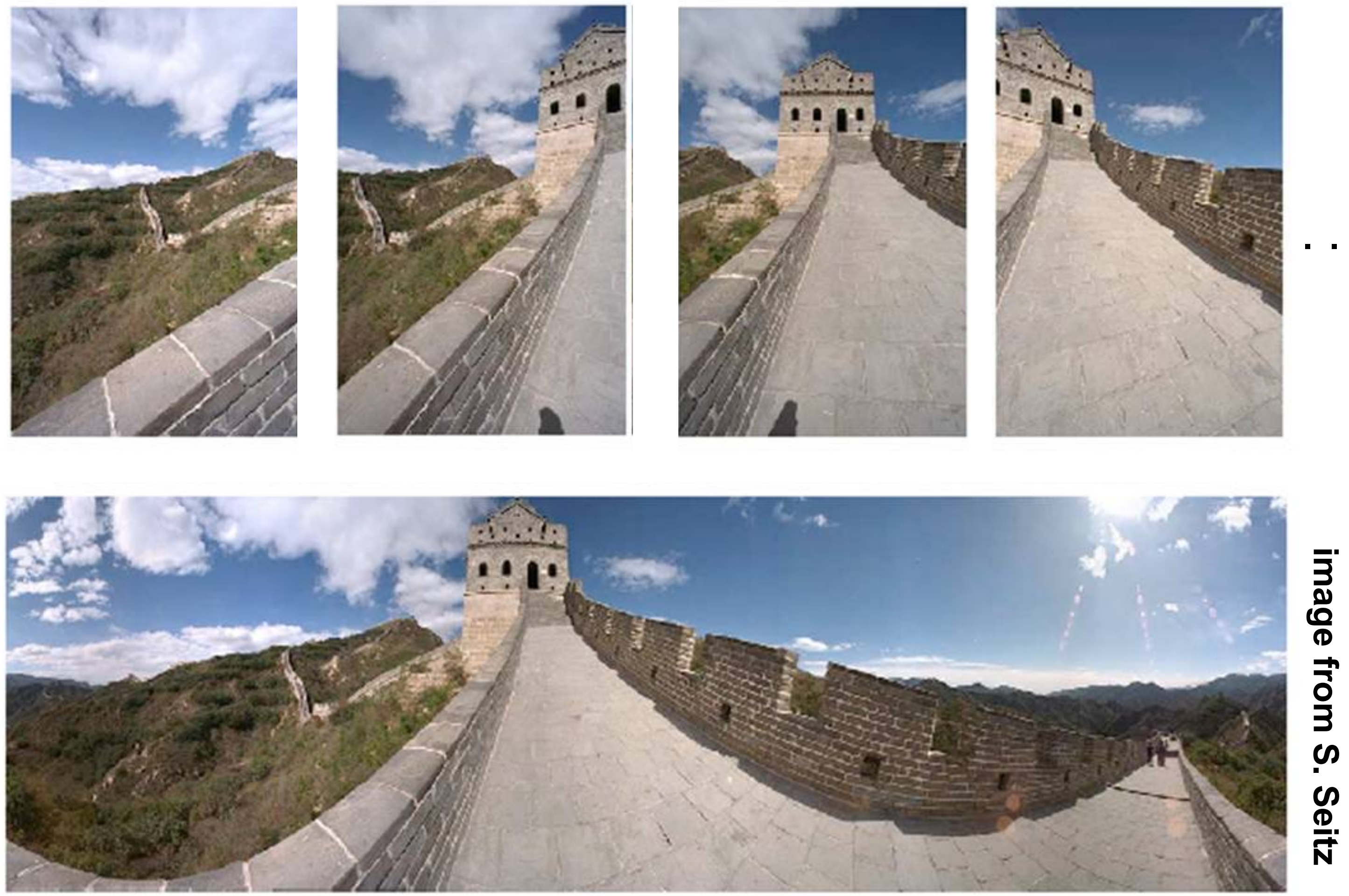


image from S. Seitz

Obtain a wide angle view by combining multiple images.

Grauman

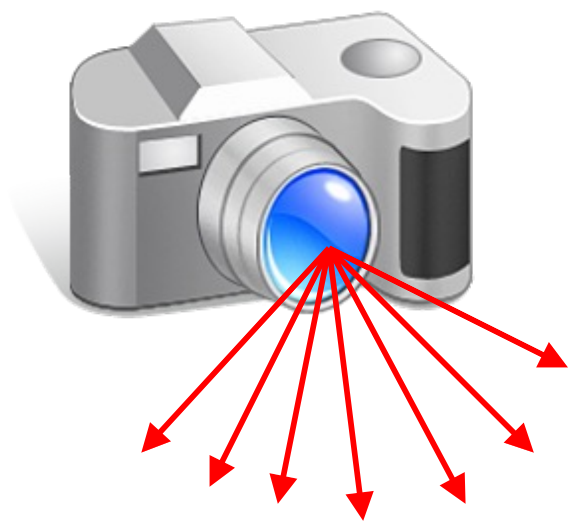
How to stitch together a panorama?

- Basic Procedure
  - Take a sequence of images from the same position
    - Rotate the camera about its **optical center.**
  - Compute transformation between 2nd and 1st image
  - Transform the 2nd image to overlap with the 1st
  - Blend the two together to create a mosaic
  - (If there are more images, repeat)
- Why should this work at all?
  - What about the 3D geometry of the scene?
  - Why aren't we using it?

Source: Steve Seitz

Correspondence

- Allows us to map image back to some real space



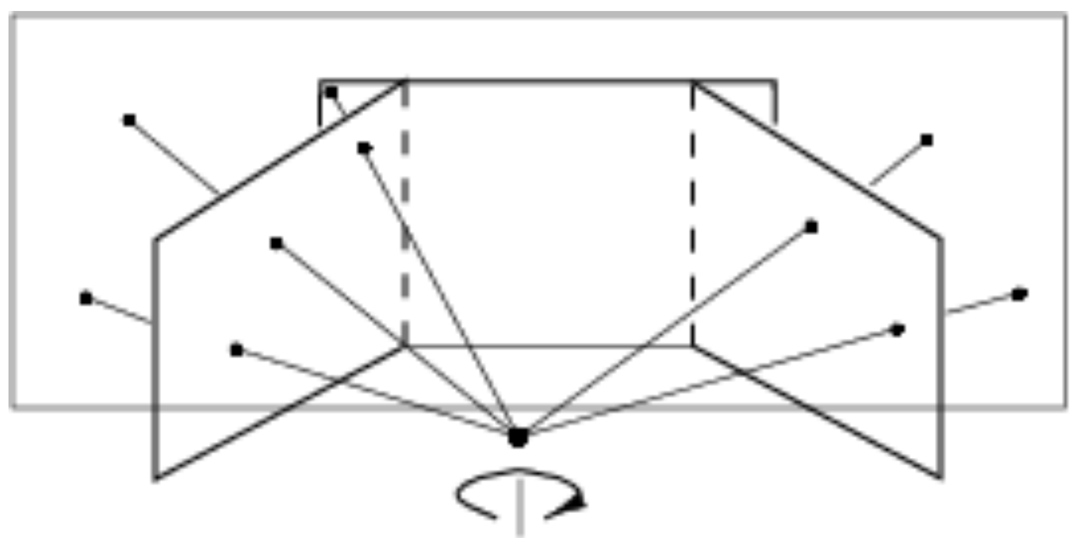
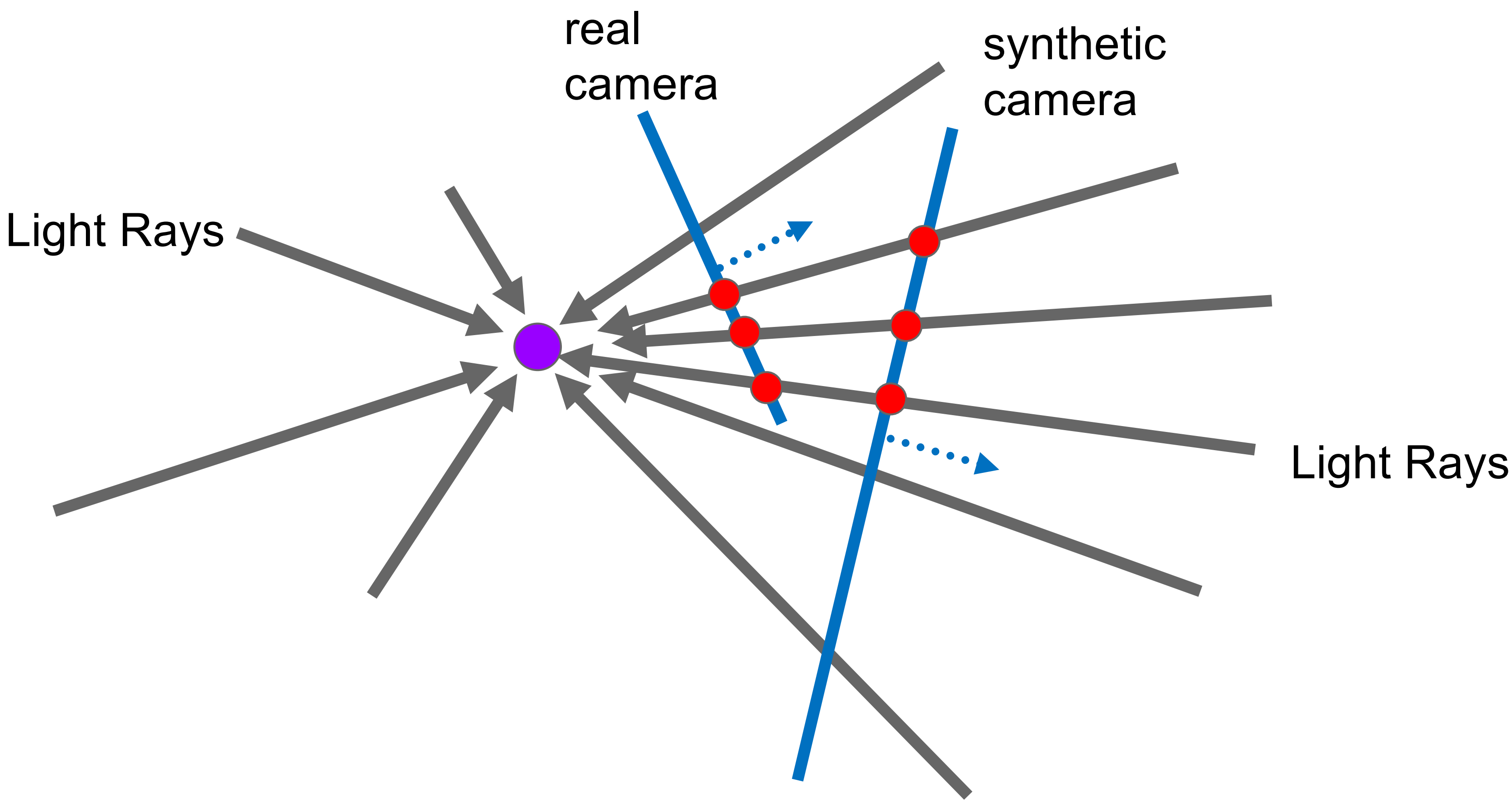


image from S. Seitz

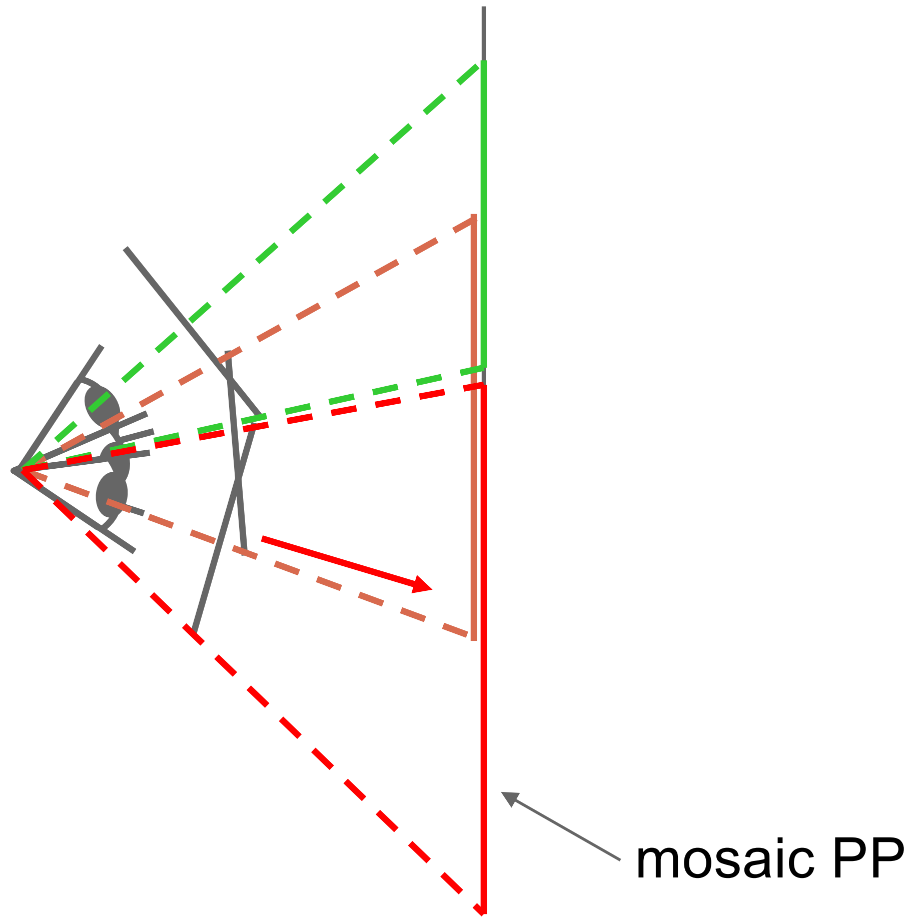
Panoramas: generating synthetic views



Can generate any synthetic camera view
as long as it has **the same center of projection**!

Source: Alyosha Efros

Image reprojection



- The mosaic has a natural interpretation in 3D
  - The images are reprojected onto a common plane
  - The mosaic is formed on this plane
  - Mosaic is a *synthetic wide-angle camera*

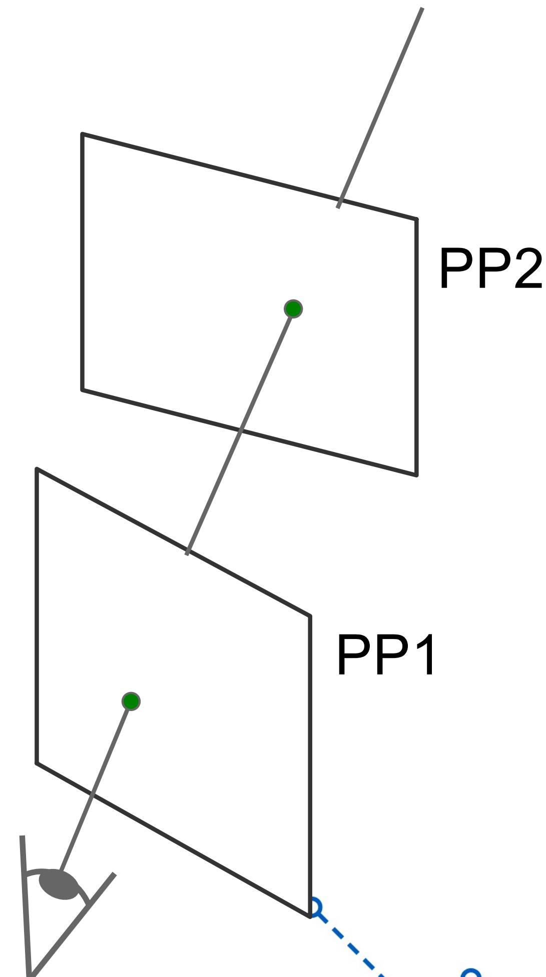
SAIR
Spatial AI & Robotics Lab

Source: Steve Seitz

Homography

- How to relate two images from the same camera center?
    - how to map a pixel from PP1 to PP2?
- Take as a 2D **image warp** using projective transform.
- A **projective transform** is a mapping between any two PPs with the same center of projection
    - rectangle should map to arbitrary quadrilateral
    - parallel lines aren't preserved.
    - but straight lines are preserved.
- Called **Homography**

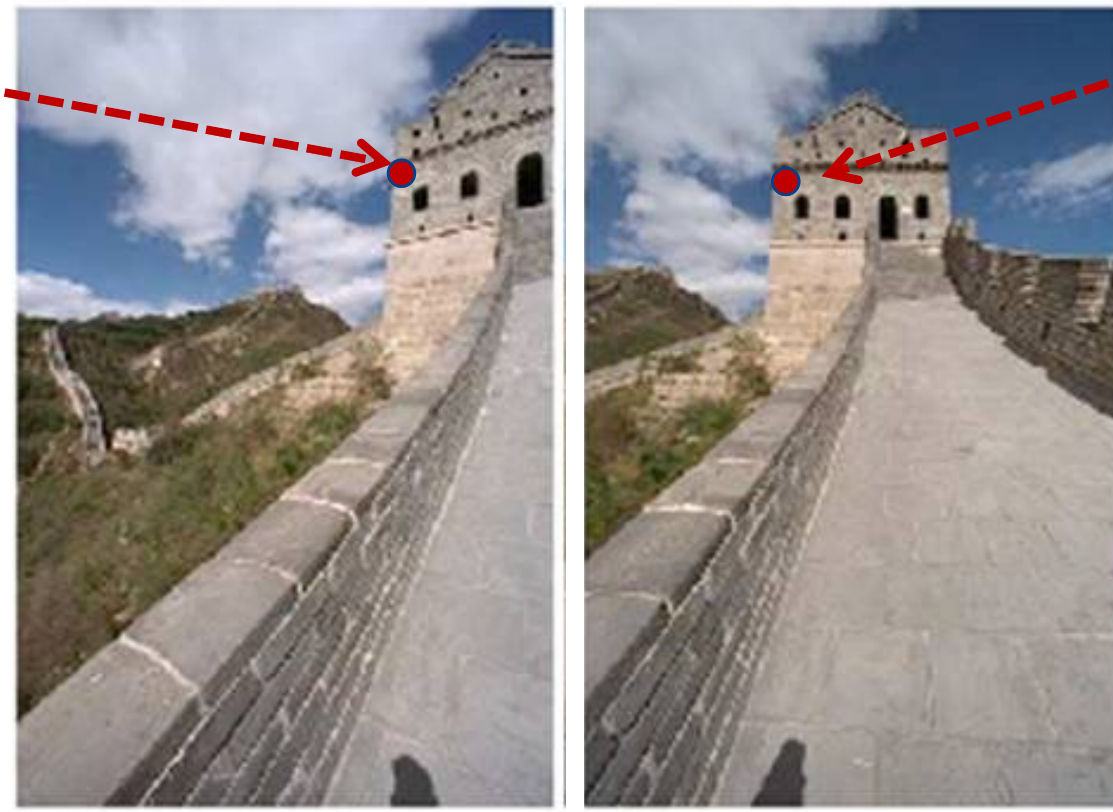
$$\underbrace{\begin{bmatrix} wx' \\ wy' \\ w \end{bmatrix}}_{\mathbf{p'}} = \underbrace{\begin{bmatrix} * & * & * \\ * & * & * \\ * & * & * \end{bmatrix}}_{\mathbf{H}} \underbrace{\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}}_{\mathbf{p}}$$



Source: Alyosha Efros

Solving for homographies

$$(x, y)$$



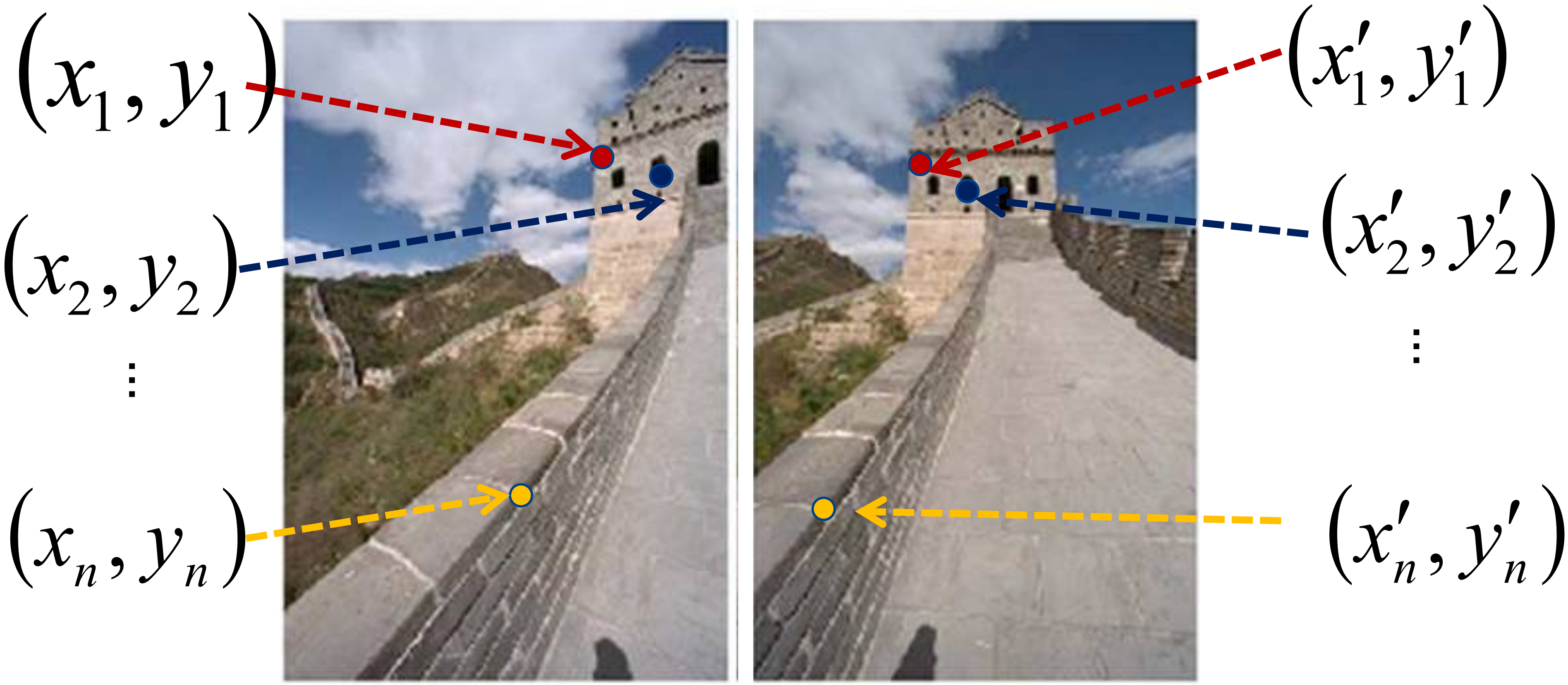
$$\left({}^{wx'}\!/\!{}_{w}, {}^{wy'}\!/\!{}_{w} \right) = (x', y')$$

To **apply** a given homography **H**

- Compute **p’** = **Hp**   (regular matrix multiply)
- Convert **p’** from homogeneous to image coordinates

$$\underbrace{\begin{bmatrix} wx' \\ wy' \\ w \end{bmatrix}}_{\mathbf{p}'} = \underbrace{\begin{bmatrix} * & * & * \\ * & * & * \\ * & * & * \end{bmatrix}}_{\mathbf{H}} \underbrace{\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}}_{\mathbf{p}}$$

Solving for homographies



To **compute** the homography given pairs of corresponding points, we need to set up an equation where the parameters of **H** are the unknowns…

S A I R
Spatial AI & Robotics Lab

Grauman

Solving for homographies

p’ = Hp

$$\begin{bmatrix} wx' \\ wy' \\ w \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

- Can set scale factor *i*=1 or ||H|| = 1. So, there are 8 unknowns.
- Set up a system of linear equations:
  - **Ah = b**

  where vector of unknowns h = [a, b, c, d, e, f, g, h]^T
- Need at least 8 equations, but the more the better…
- Solve for H. If over-constrained, solve using least-squares:

$$\min \|Ah - b\|^2$$

$$h = (A^T A)^{-1} A^T b$$

Grauman

Proof of least squares

- $F(h) = ||Ah - b||^2 = (Ah - b)^T \ (Ah - b)$
- $F(h) = h^T A^T Ah - h^T A^T b - b^T Ah + b^T b$
- $\frac{\partial}{\partial h} F(h) = 2A^T Ah - A^T b - (b^T A)\hat{}T$
- Setting derivative to 0: $\frac{\partial}{\partial h} F(h)$=0
- $A^T Ah = A^T b$
- $h = (A^T A)^{-1} A^T b$

SAIR Spatial AI & Robotics Lab

How to stitch together a panorama?

- Take a sequence of images from the same position
    - Rotate the camera about its optical center
- Compute transformation between second image and first
- Transform the second image to overlap with the first
- Blend the two together to create a mosaic
- If there are more images, repeat

Source: Steve Seitz