

HOUGH

transform

MACHINE ANALYSIS OF BUBBLE CHAMBER PICTURES

P. V. C. Hough

The University of Michigan, Ann Arbor, Mich.

1. AREA ELEMENTS VERSUS LINE SEGMENTS IN PICTURE ANALYSIS

Many people have suggested that a modern digital computer should be able to recognize a fairly complex pattern of tracks in a bubble chamber photograph such as that shown in Fig. 1a (*). Concrete schemes for such recognition generally assume that information is available about the presence or absence of bubbles in area elements covering the pictures and of a size appropriate to the resolution of the chamber. However, rough investigation of the time to read such information into a computer and to conduct a search for linear correlations among bubbles has so far led

important reason, the slope of each line segment provides the computer with a good prediction of the location of the adjoining line segment and so reduces enormously the search time in recognizing a track.

It will be shown below that the tracks in one picture may be recognized in a time of the order of 1-2 s, and therefore a stereo pair of pictures may be analyzed in less than 5 s. It seems that an analysis time of this order of magnitude is a reasonable goal since it matches the cycle time of large accelerators.

Hough
Lines

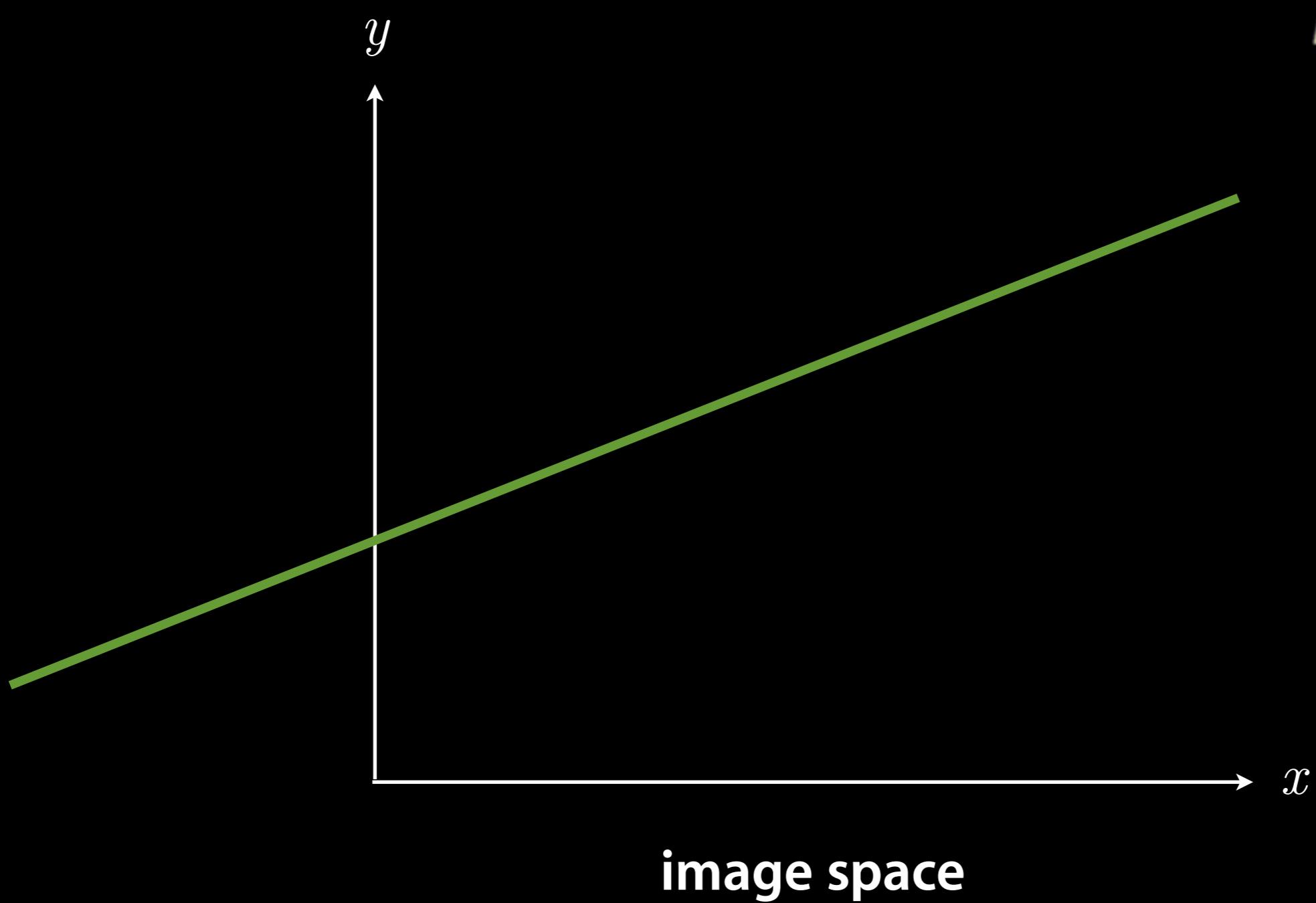
y



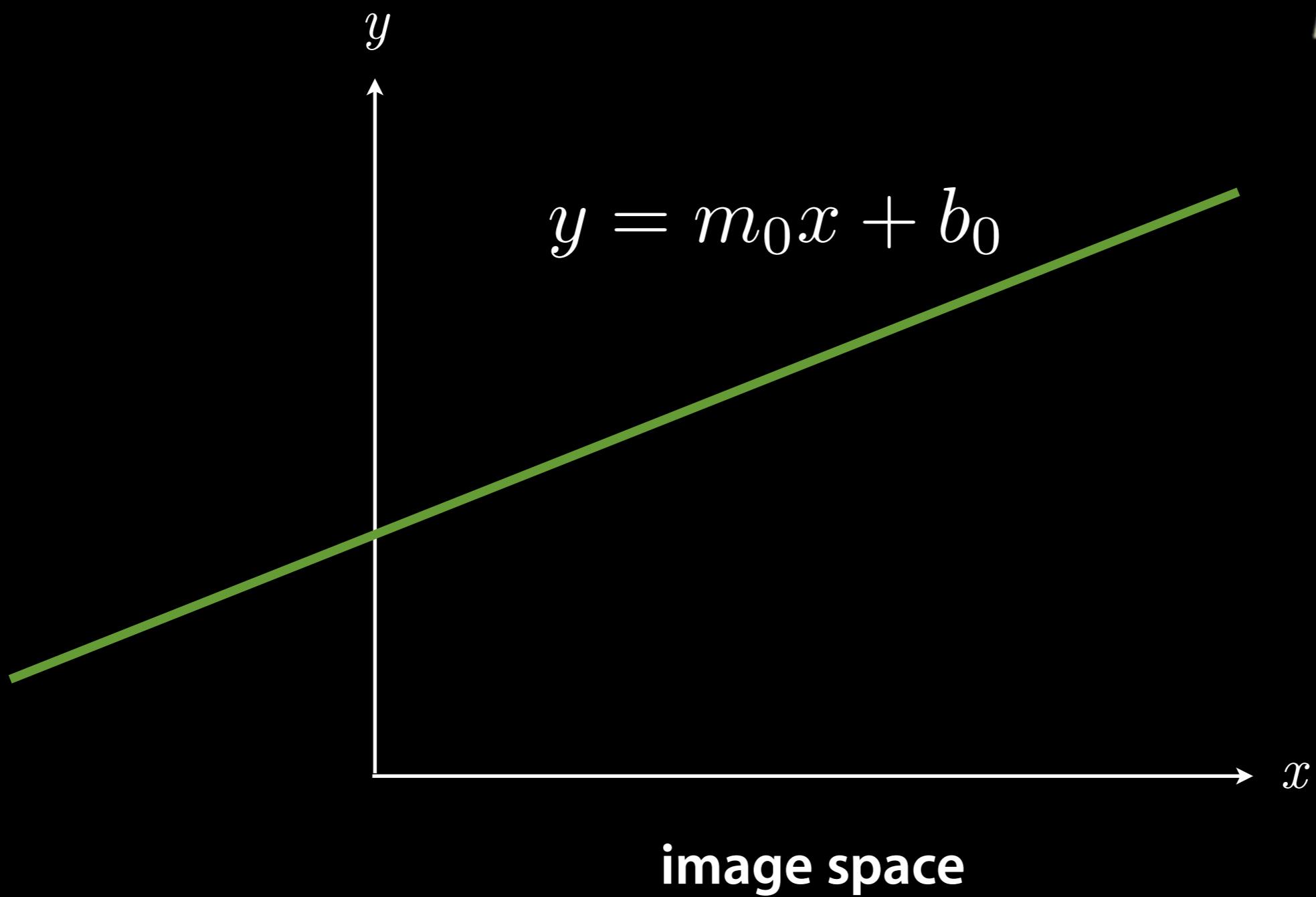
x

image space

Hough
Lines



Hough
Lines



Hough Lines

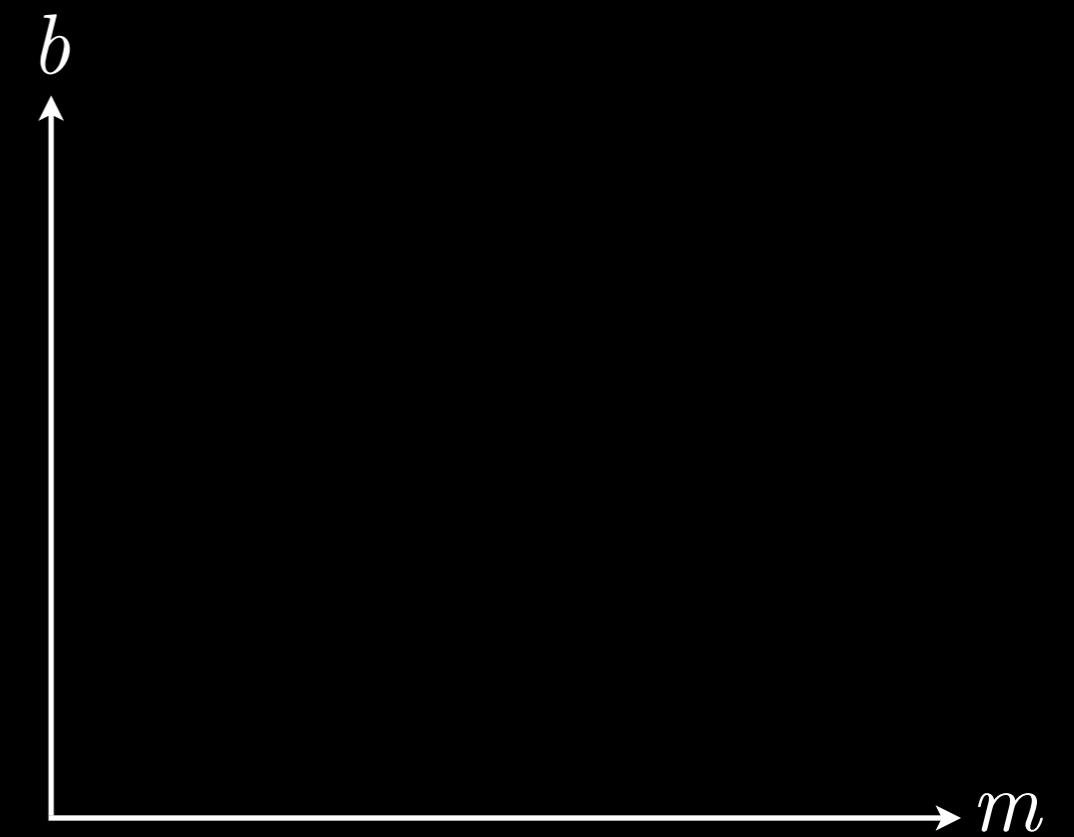
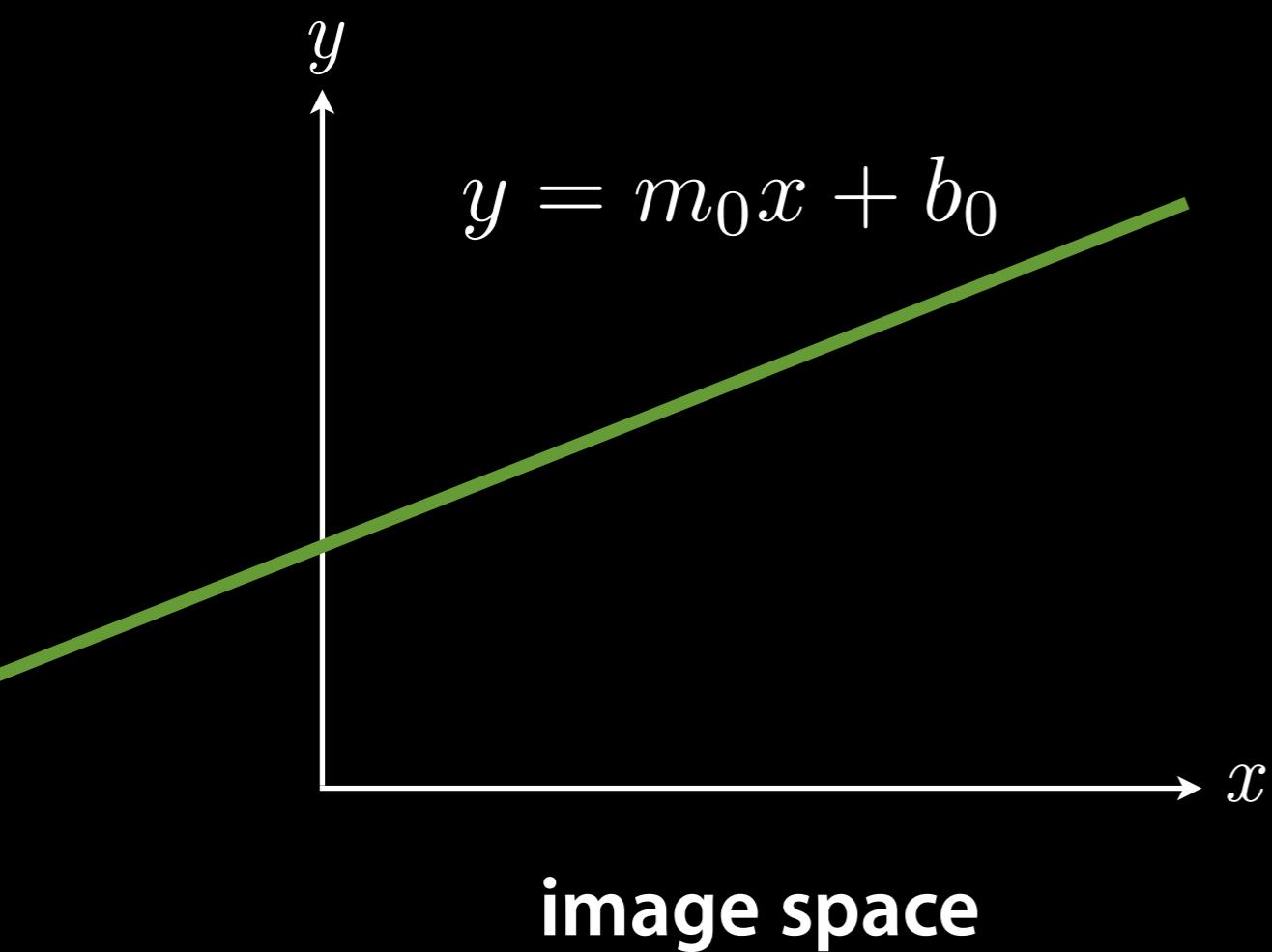
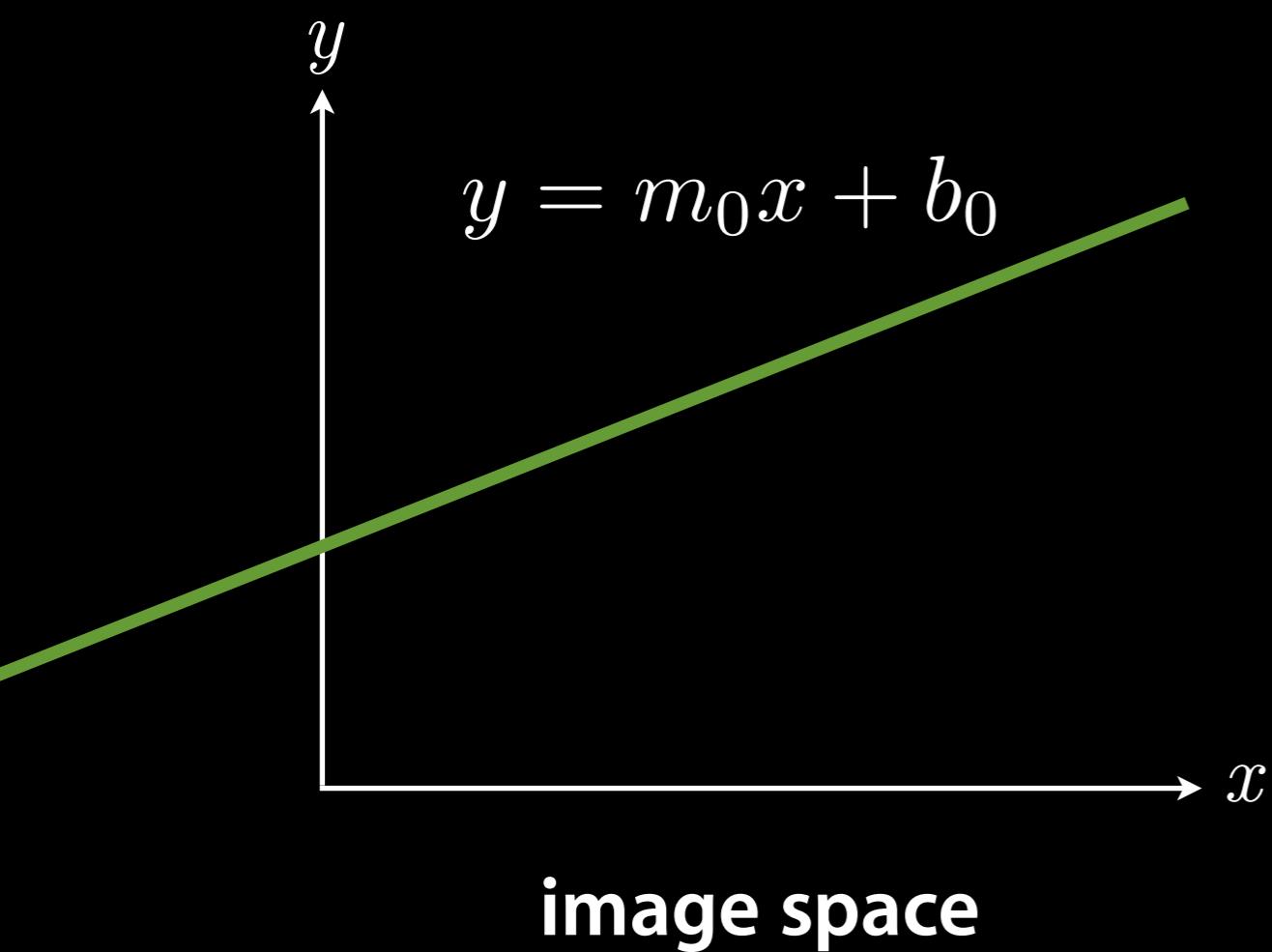


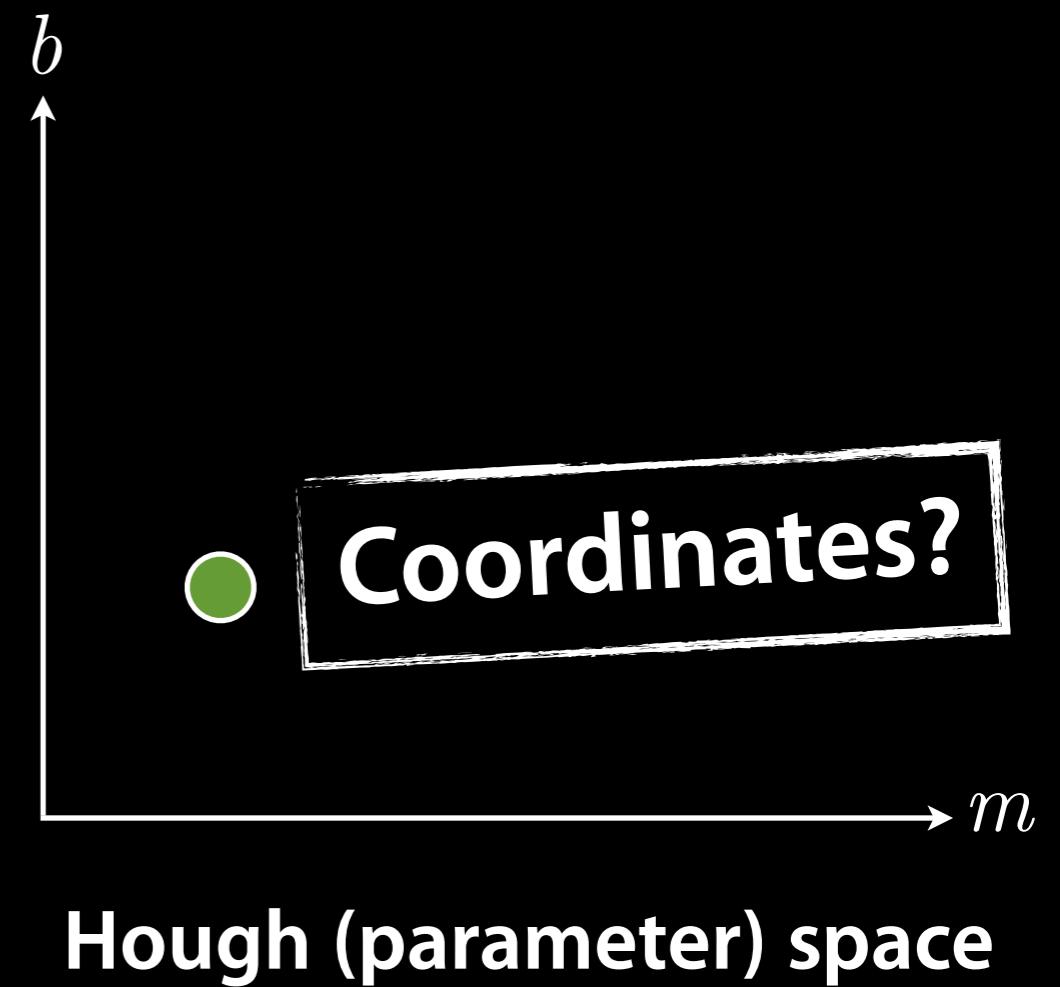
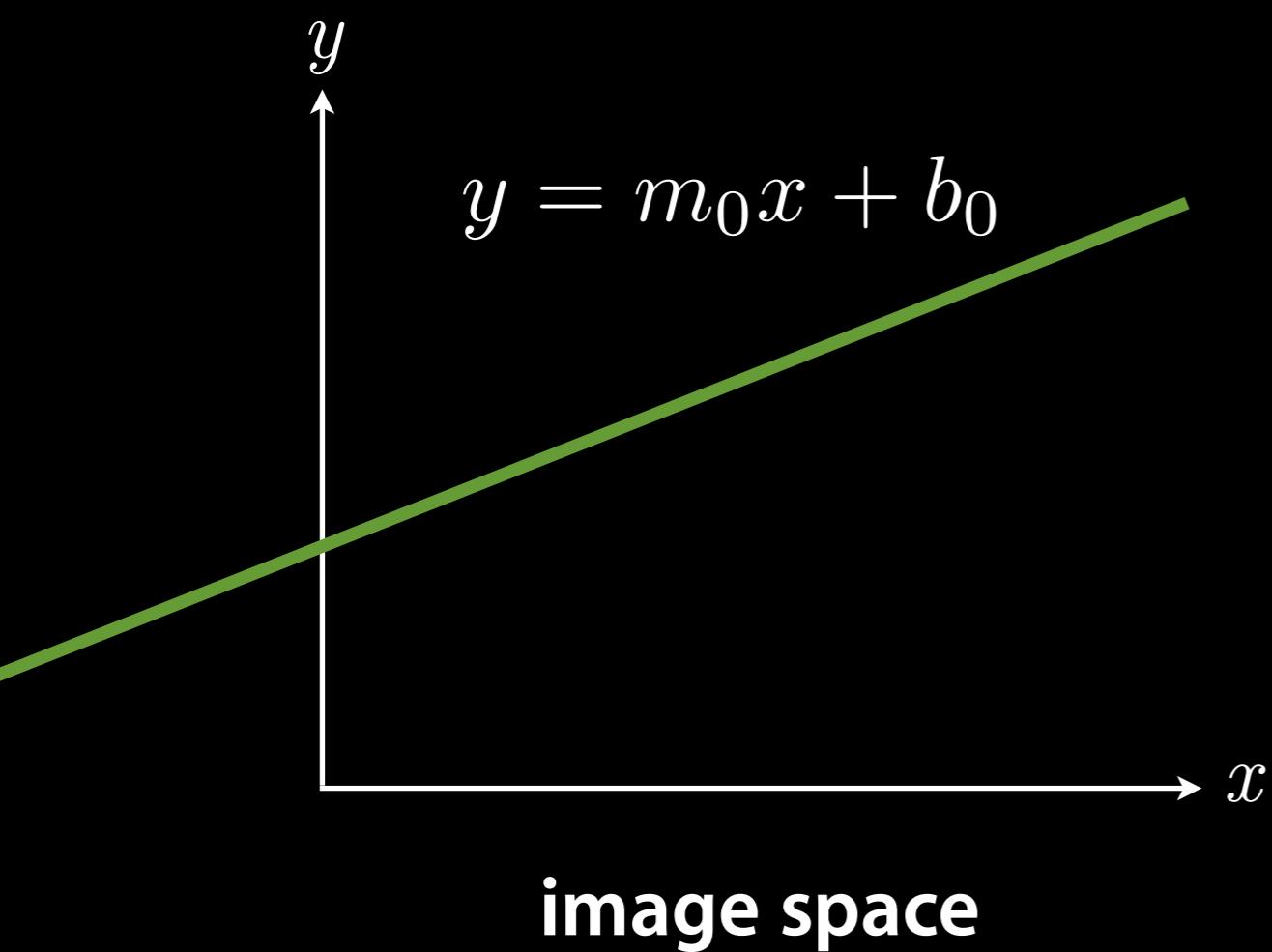
image space

Hough (parameter) space

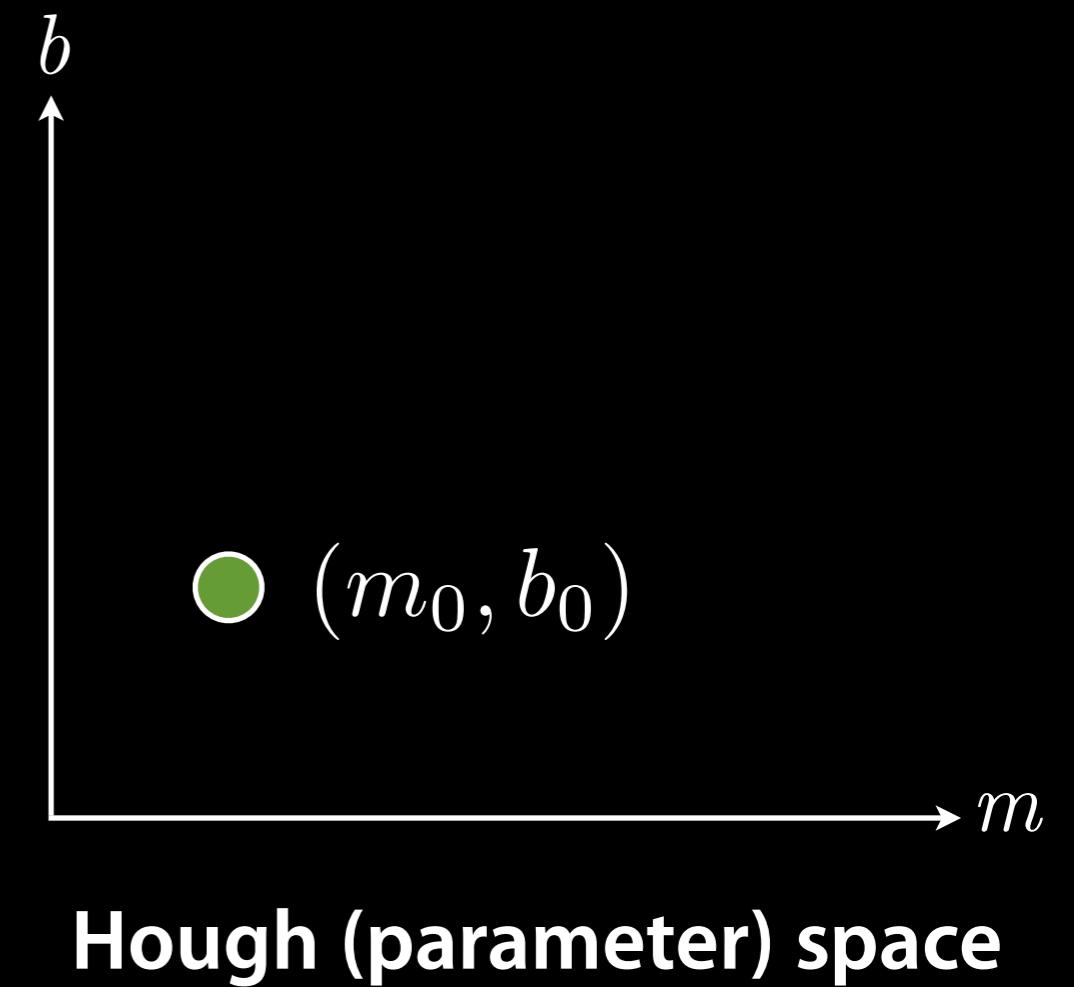
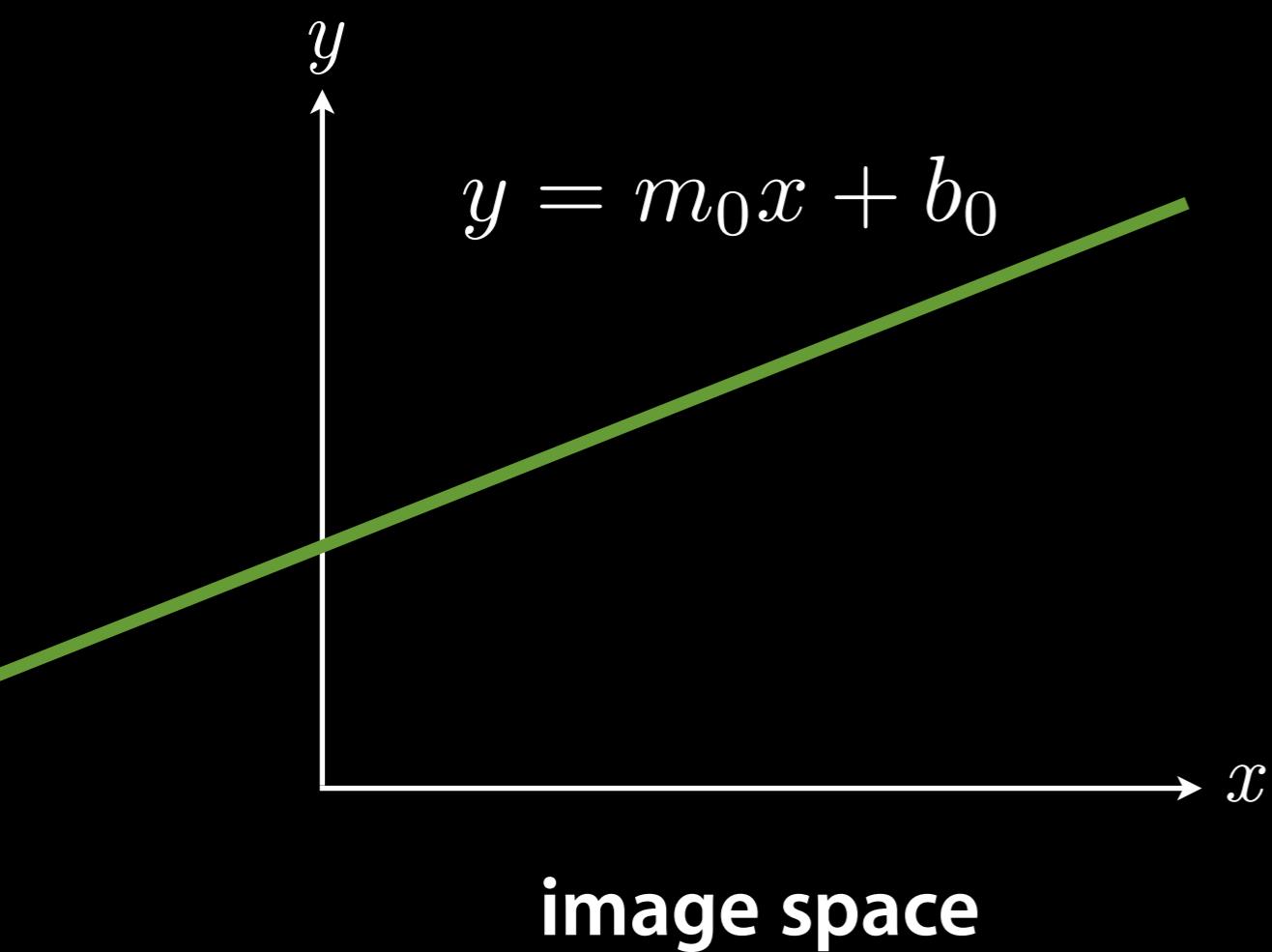
Hough Lines



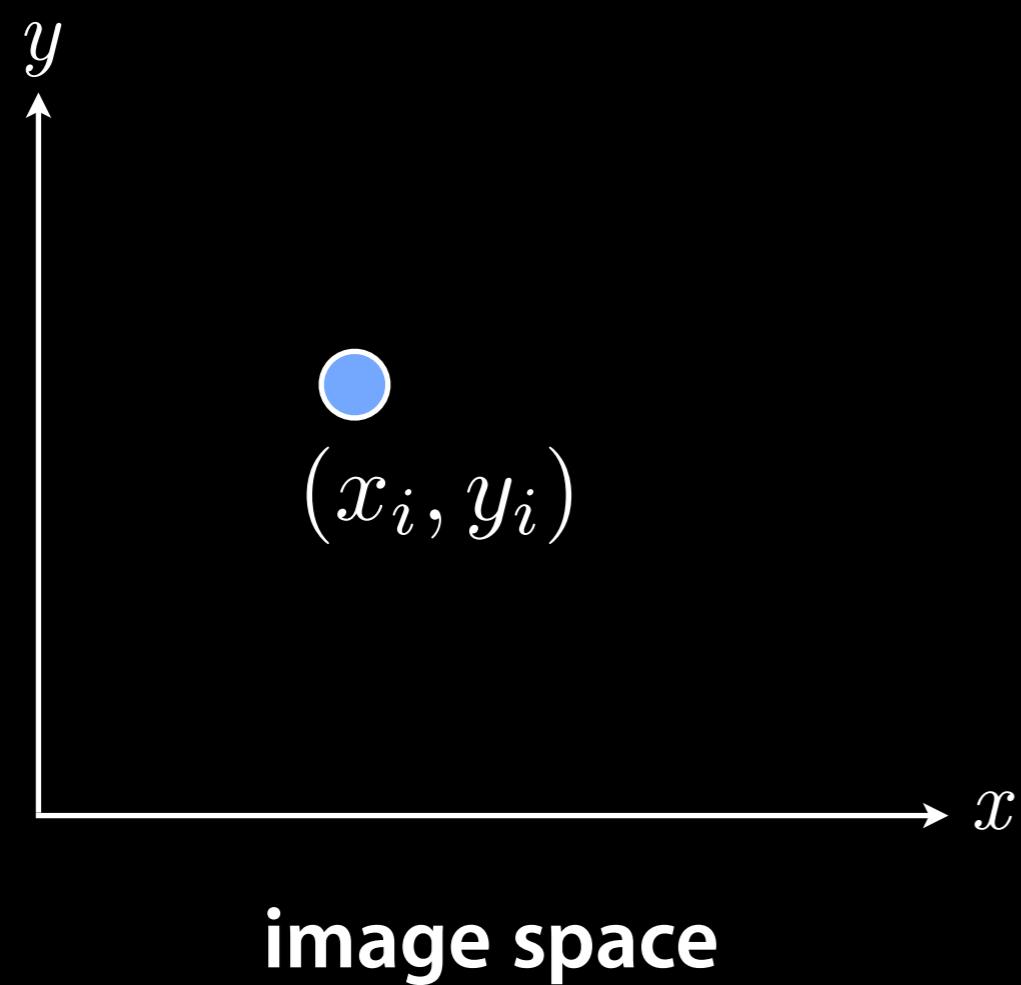
Hough Lines



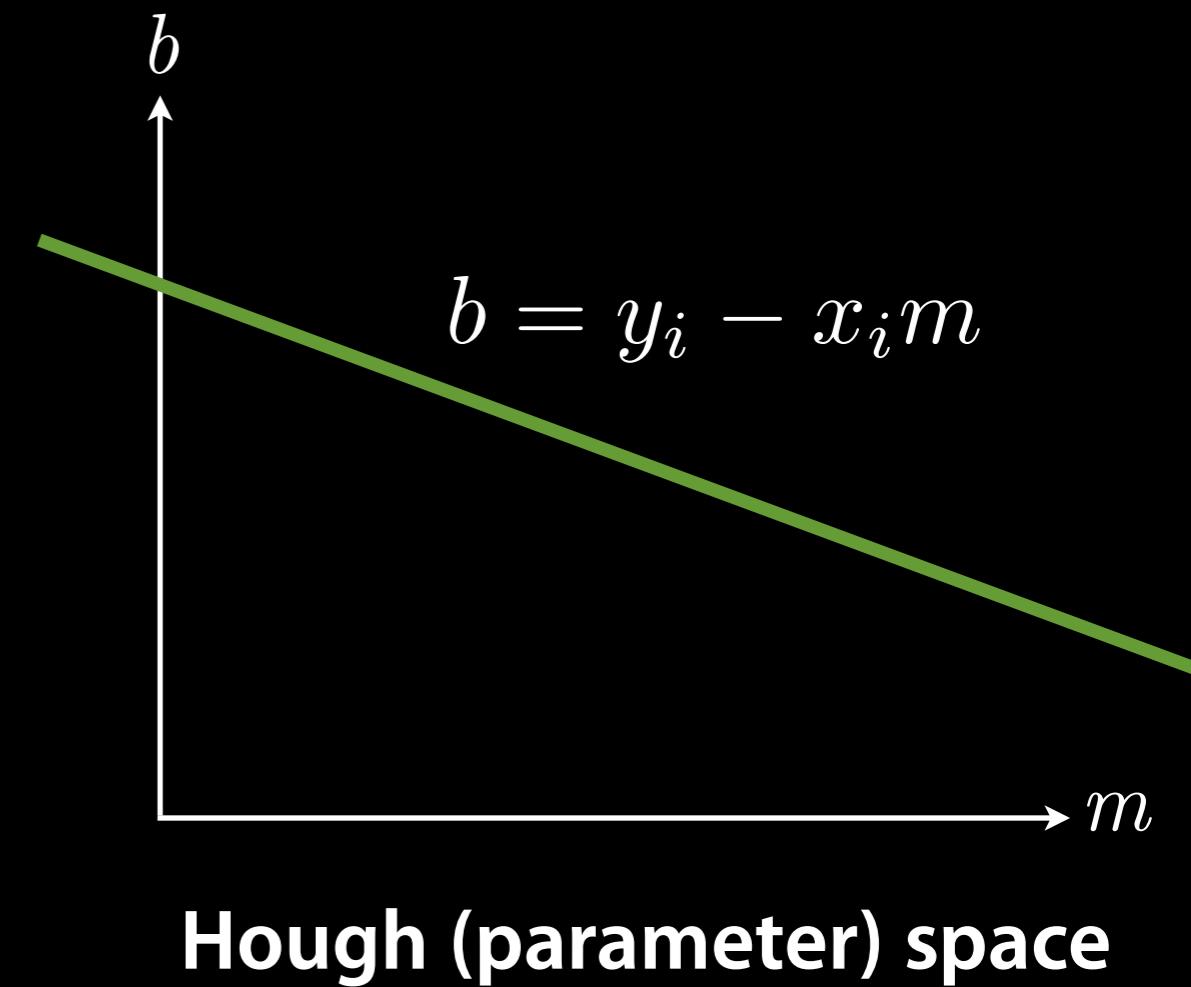
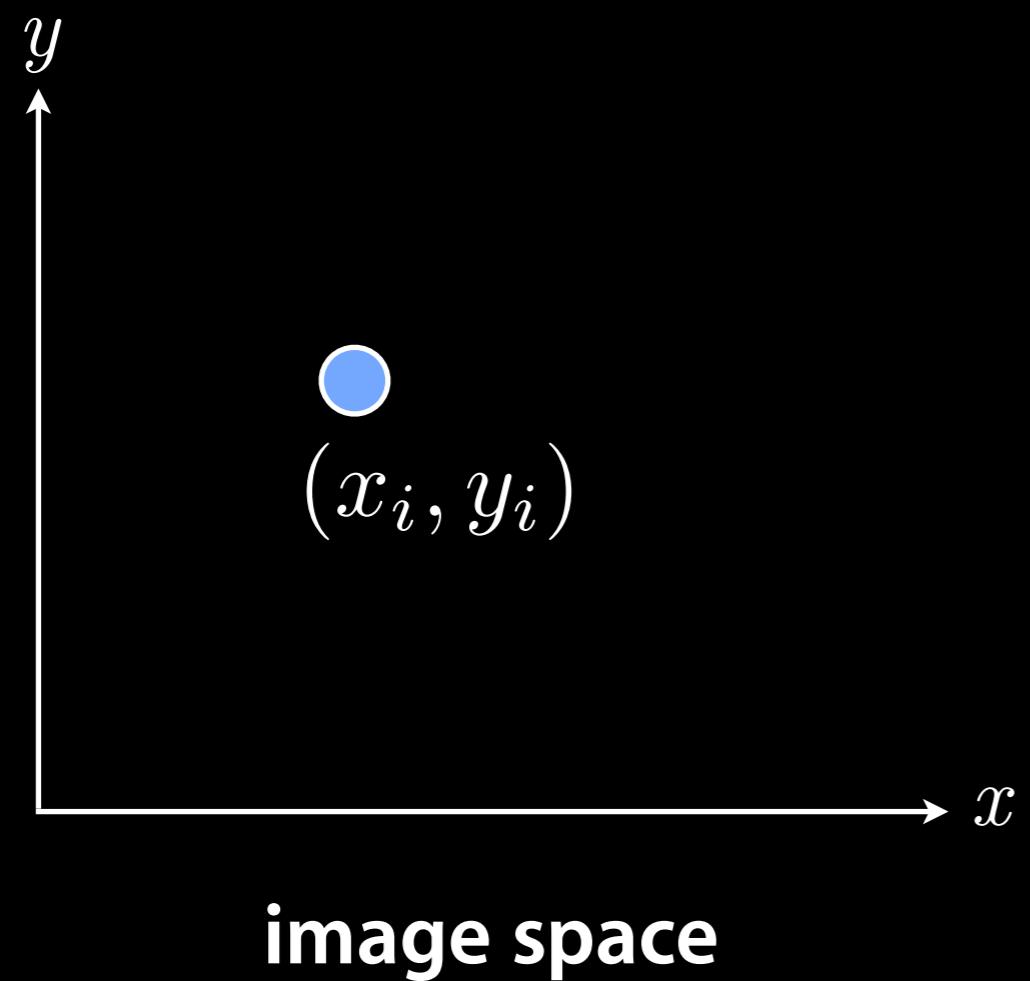
Hough Lines



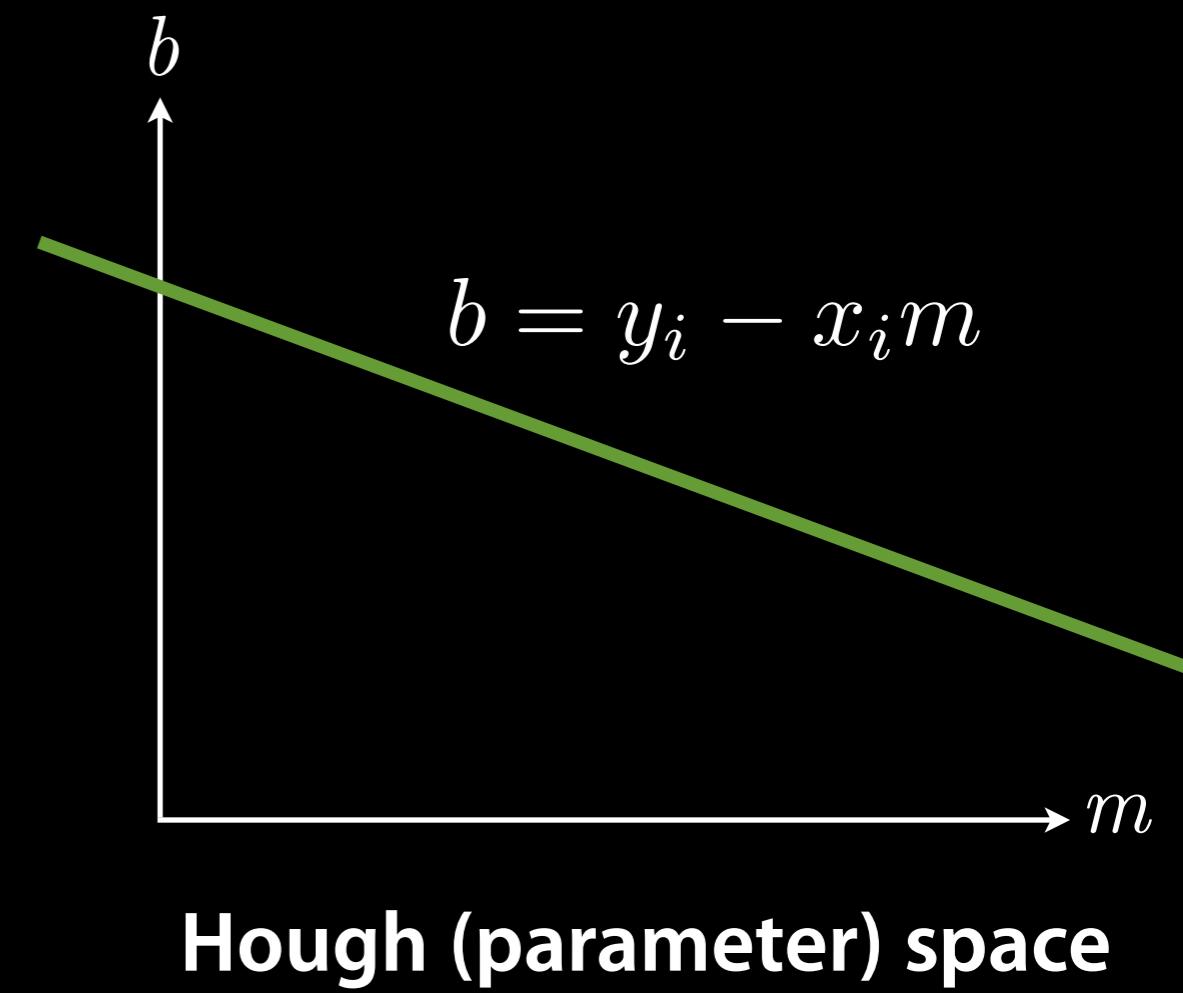
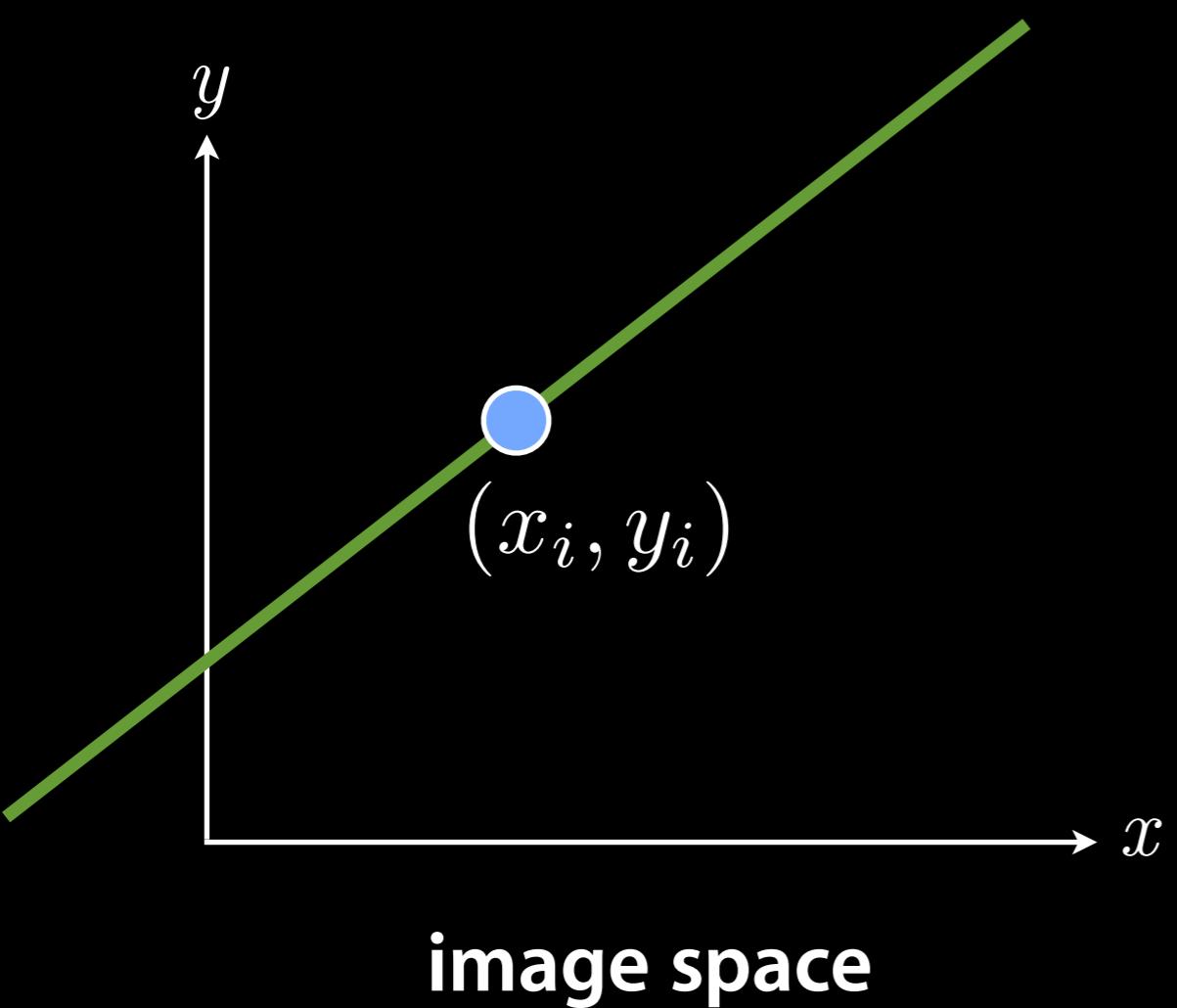
Point-Line
Duality



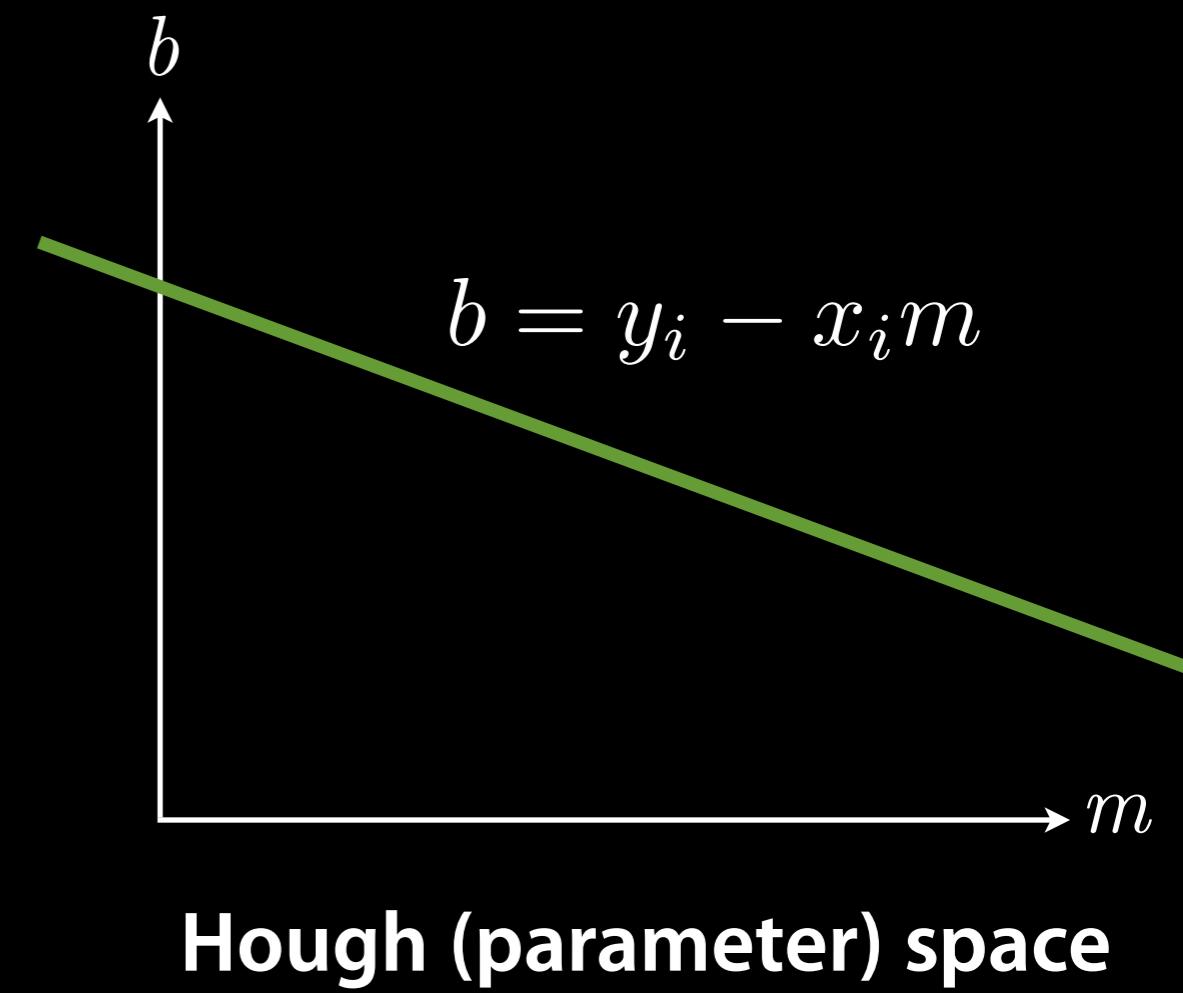
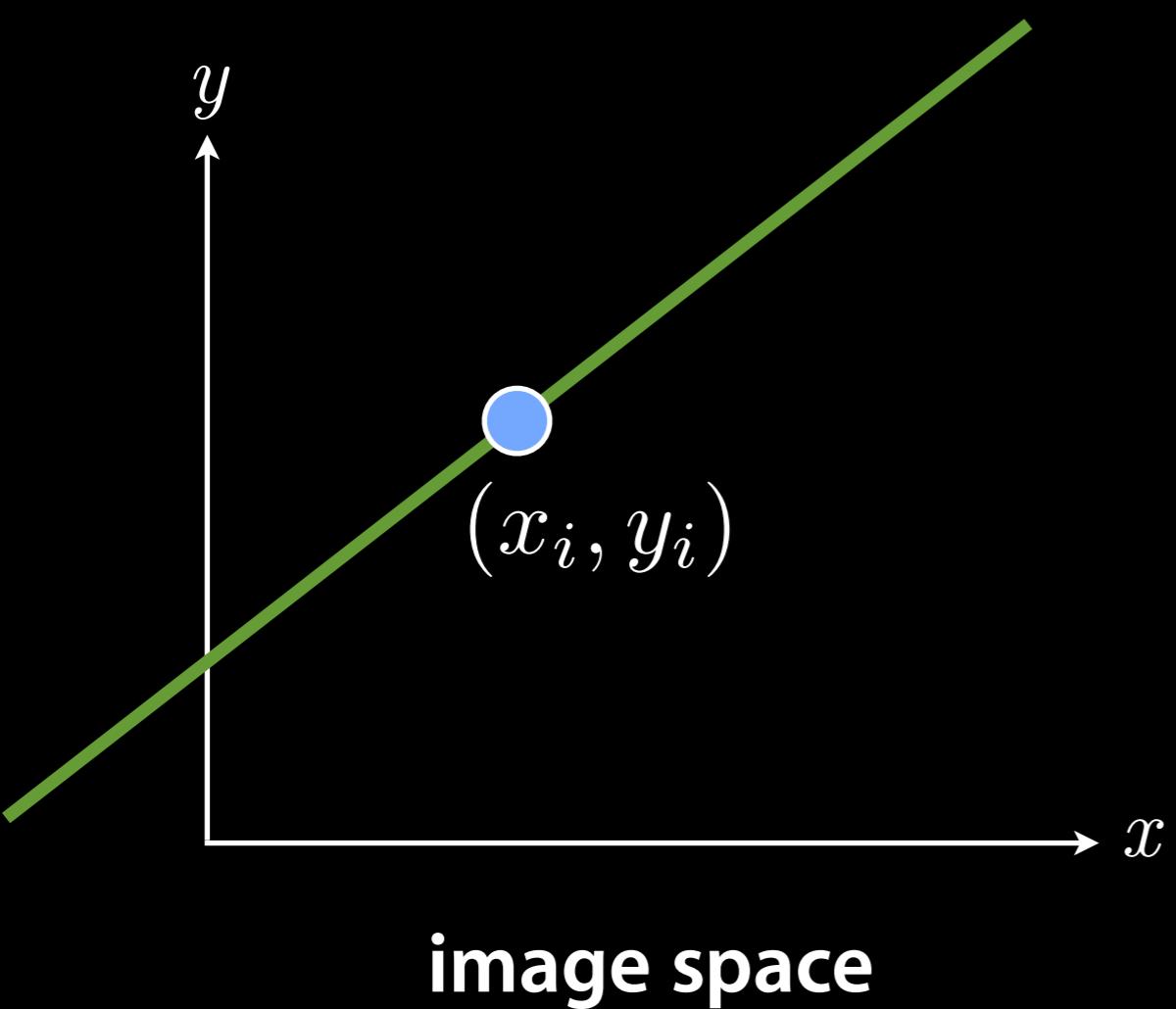
Point-Line
Duality



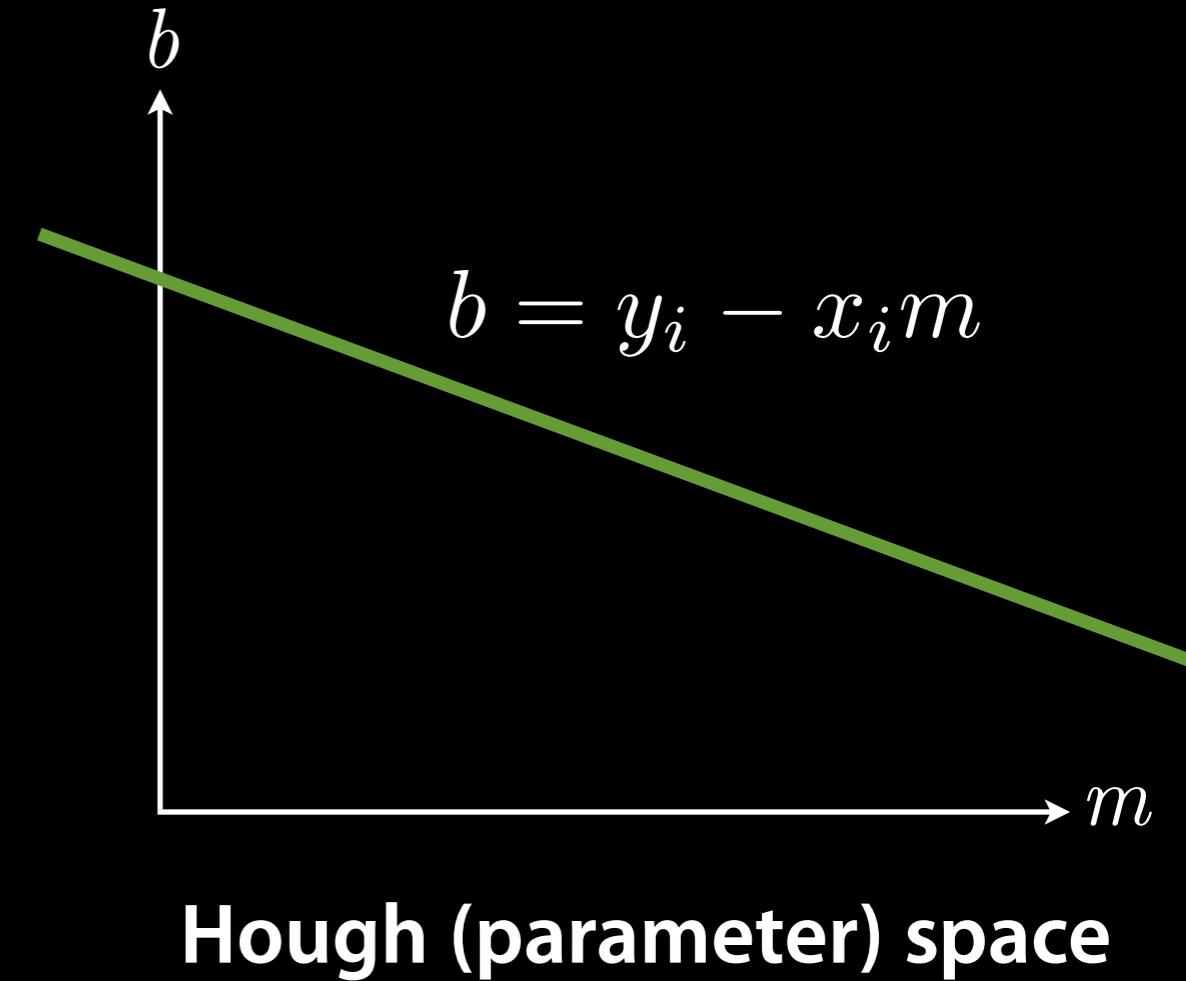
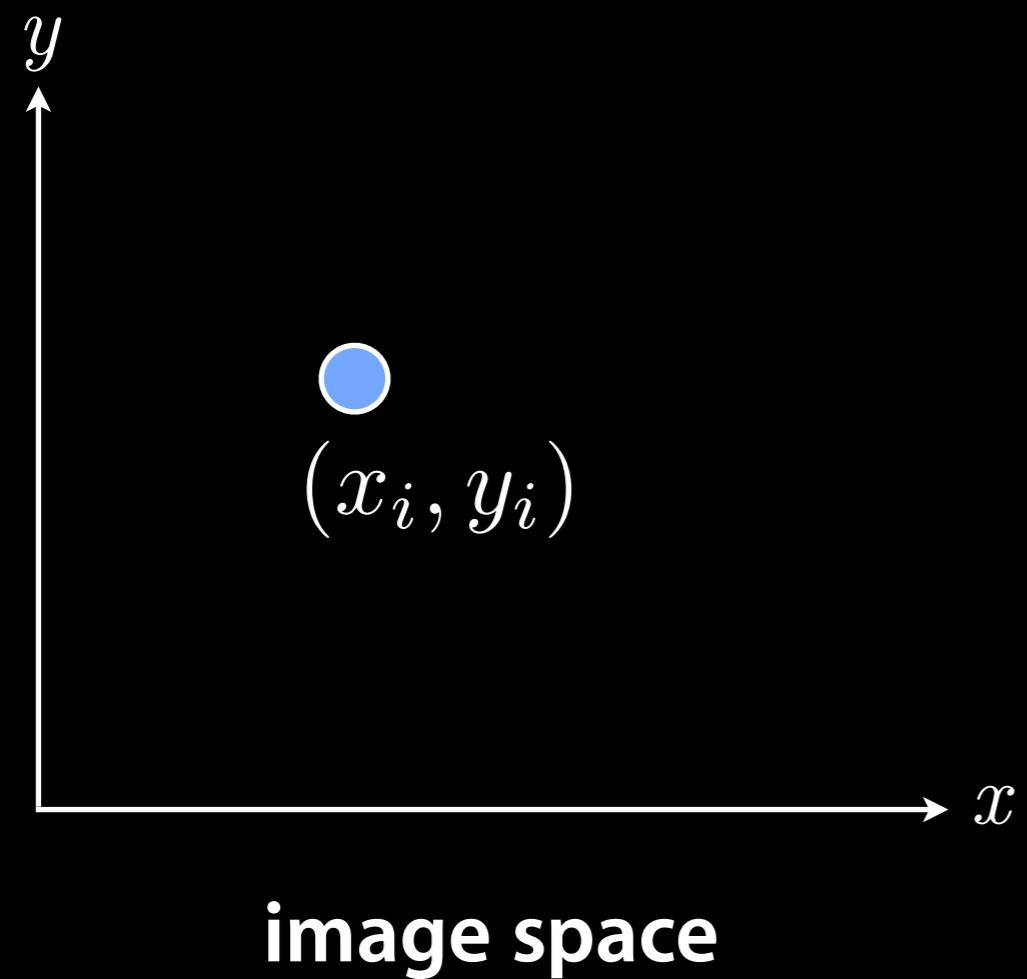
Point-Line Duality



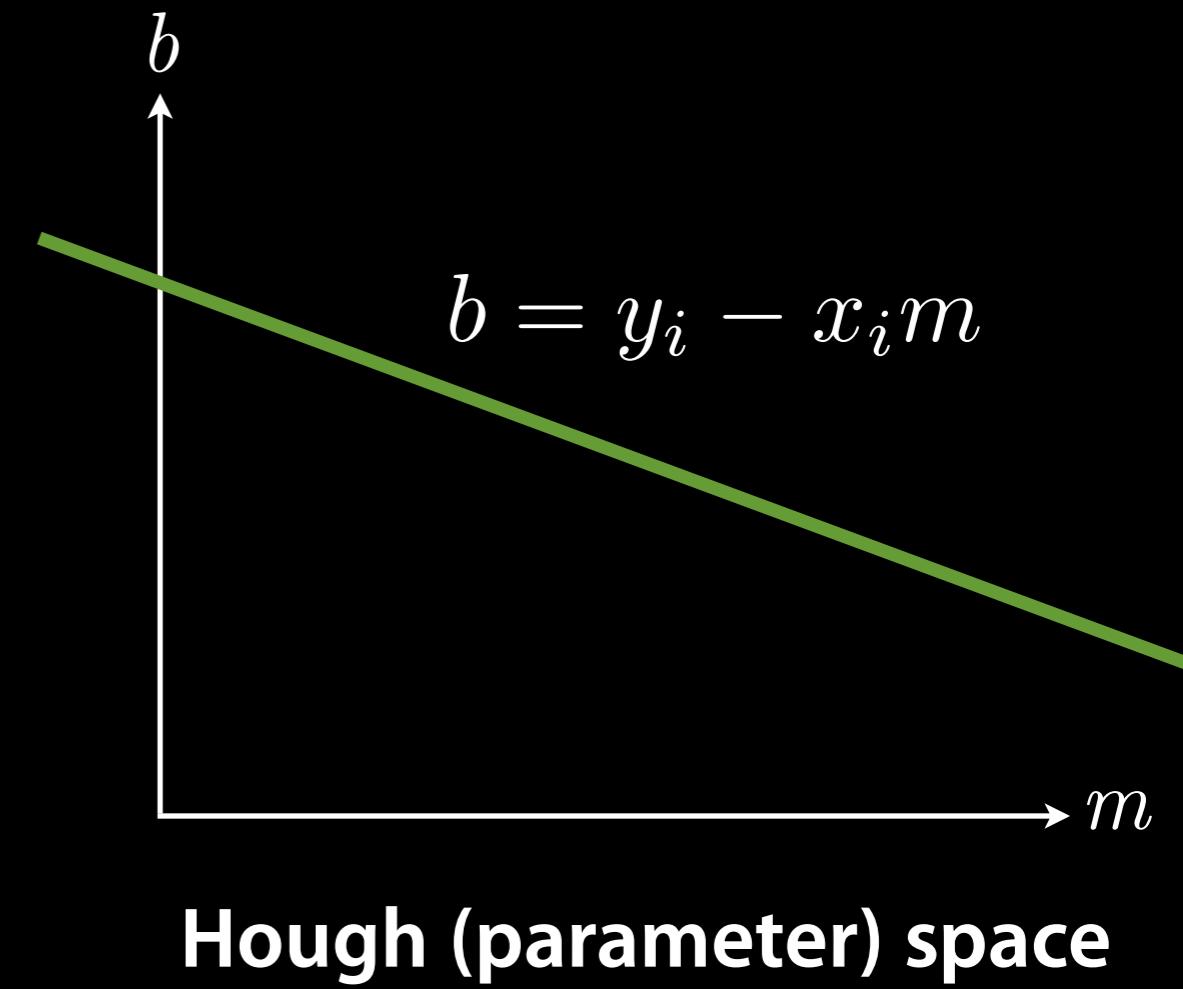
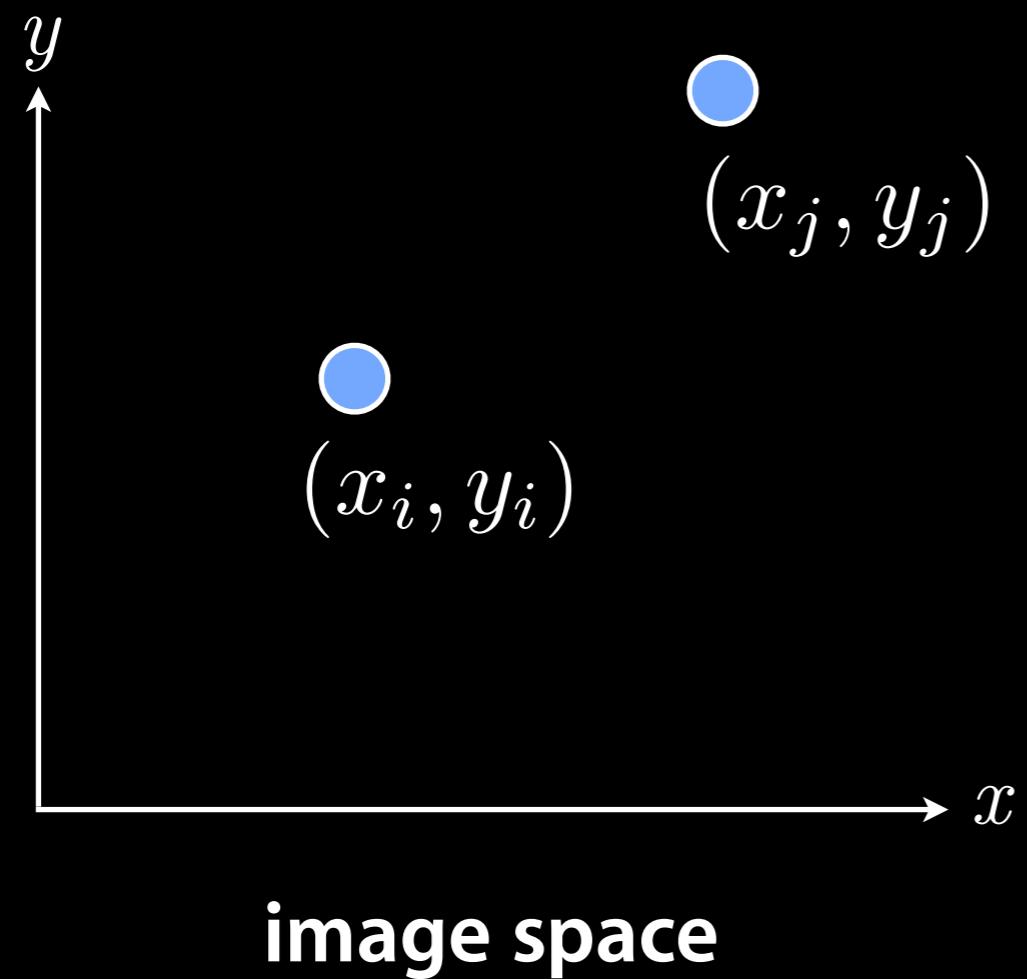
Point-Line Duality



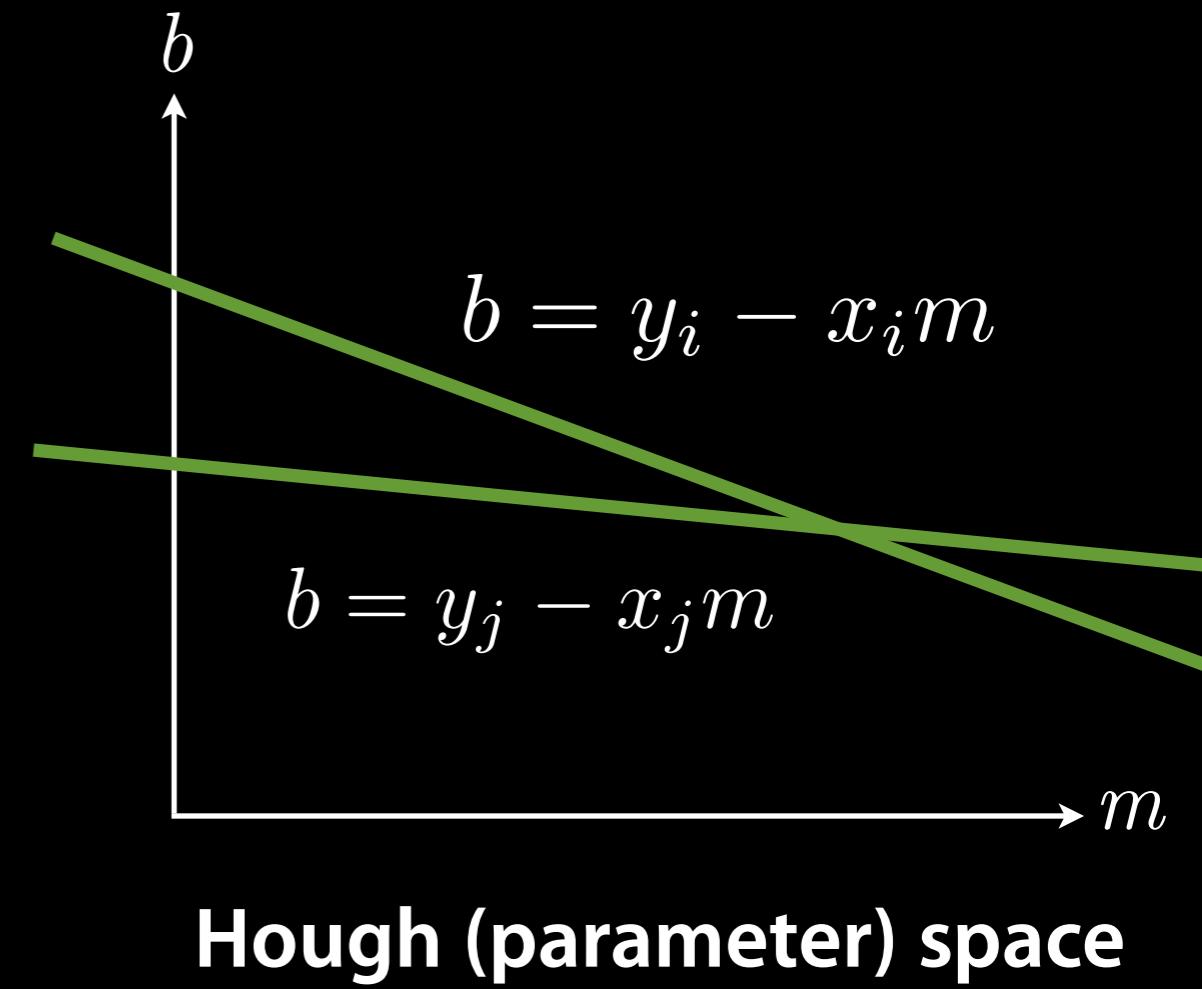
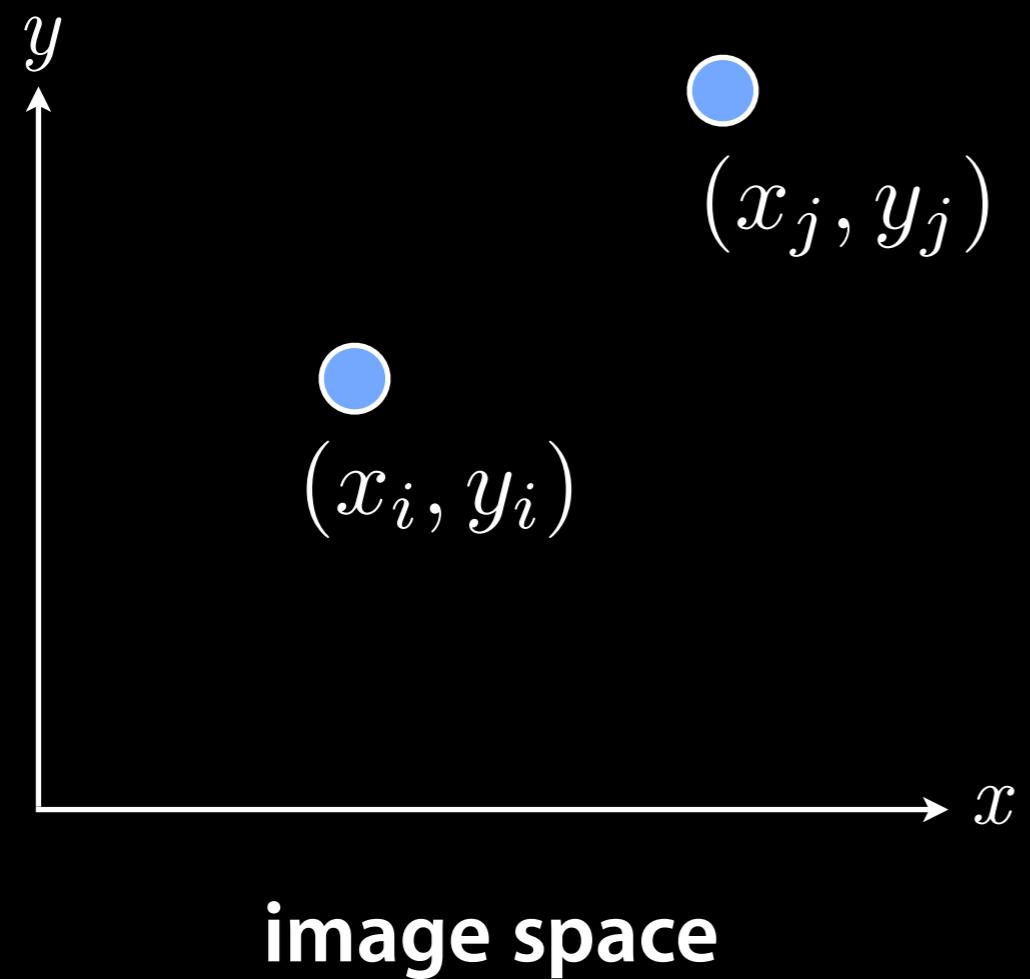
Hough Lines



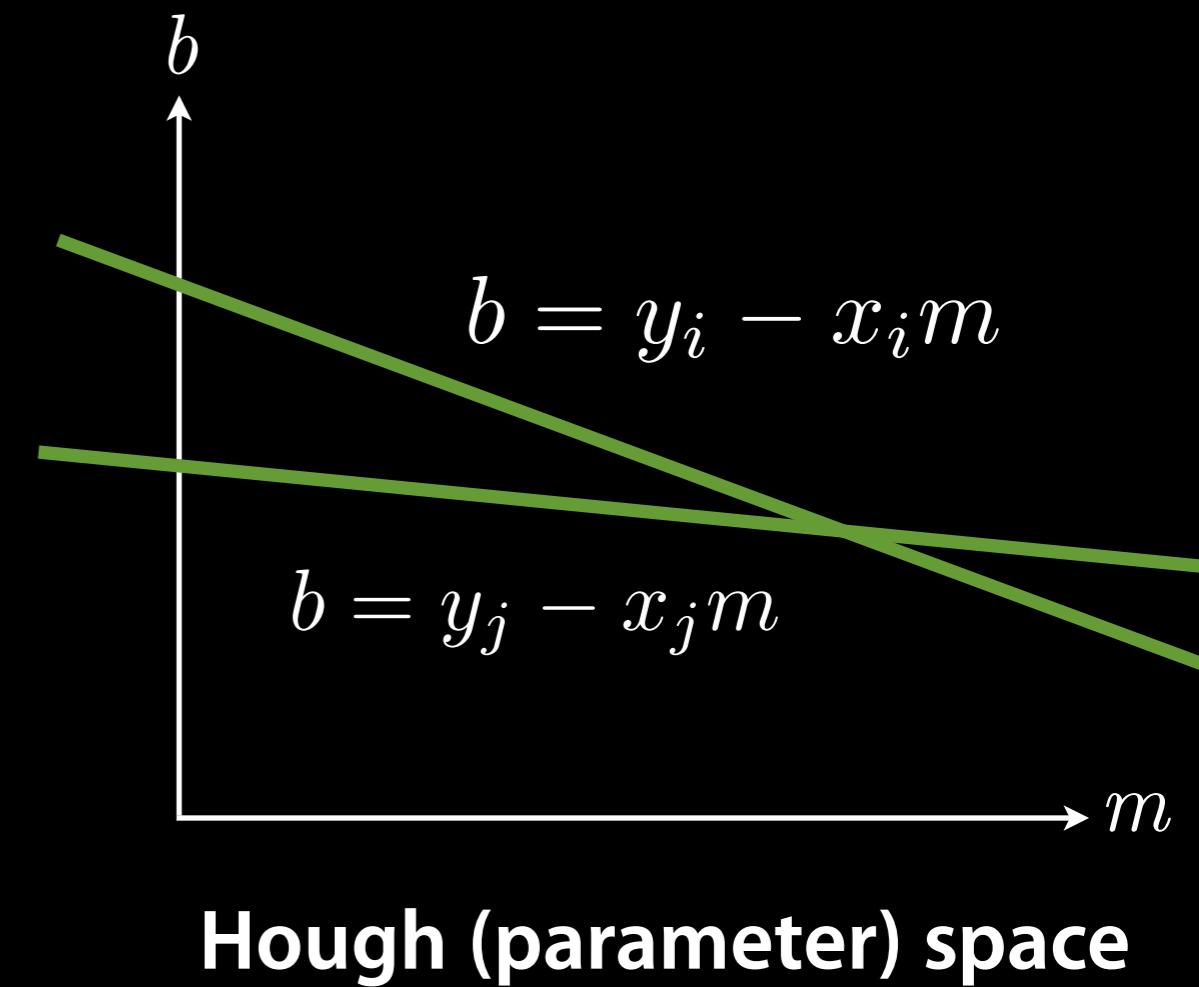
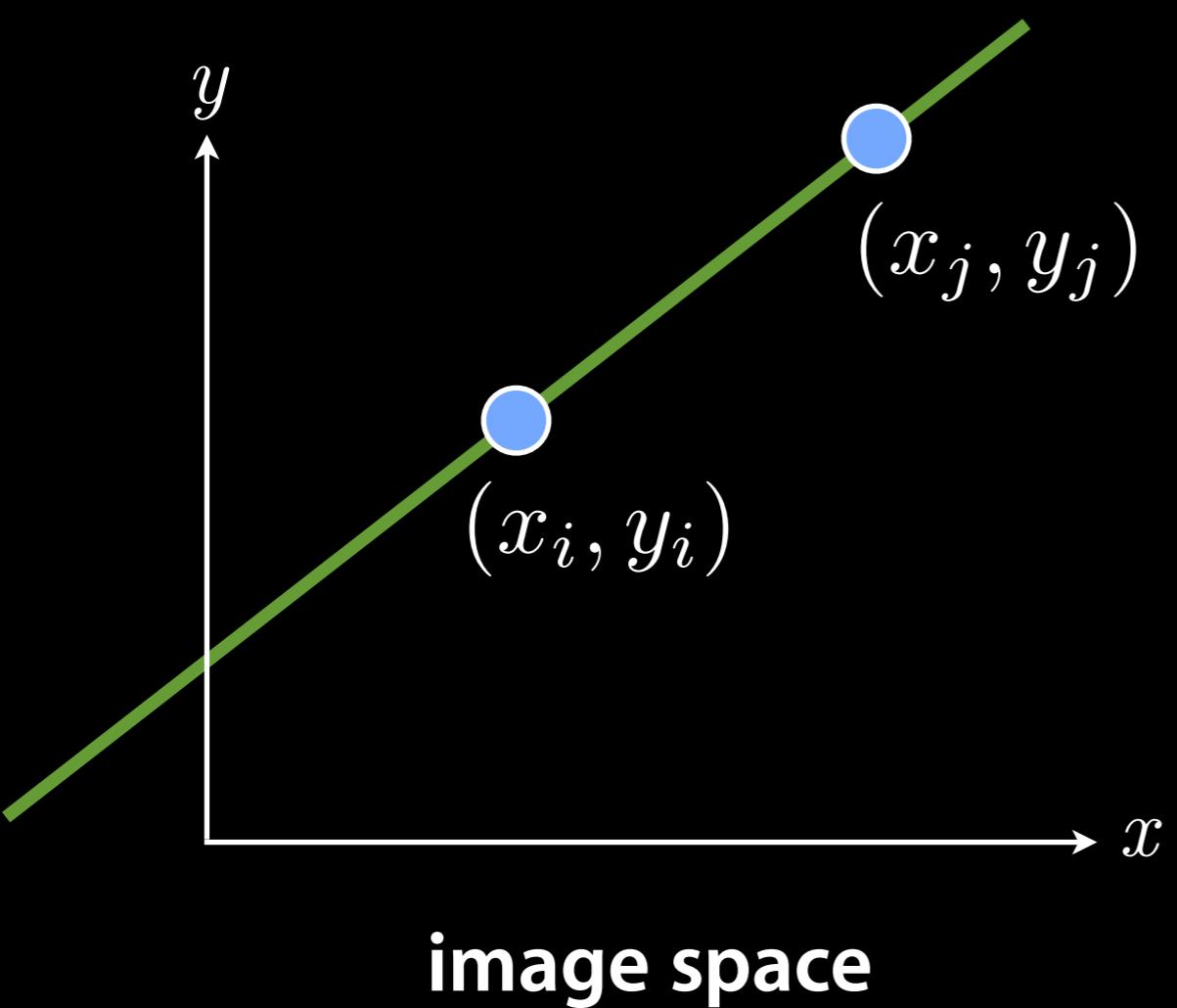
Hough Lines



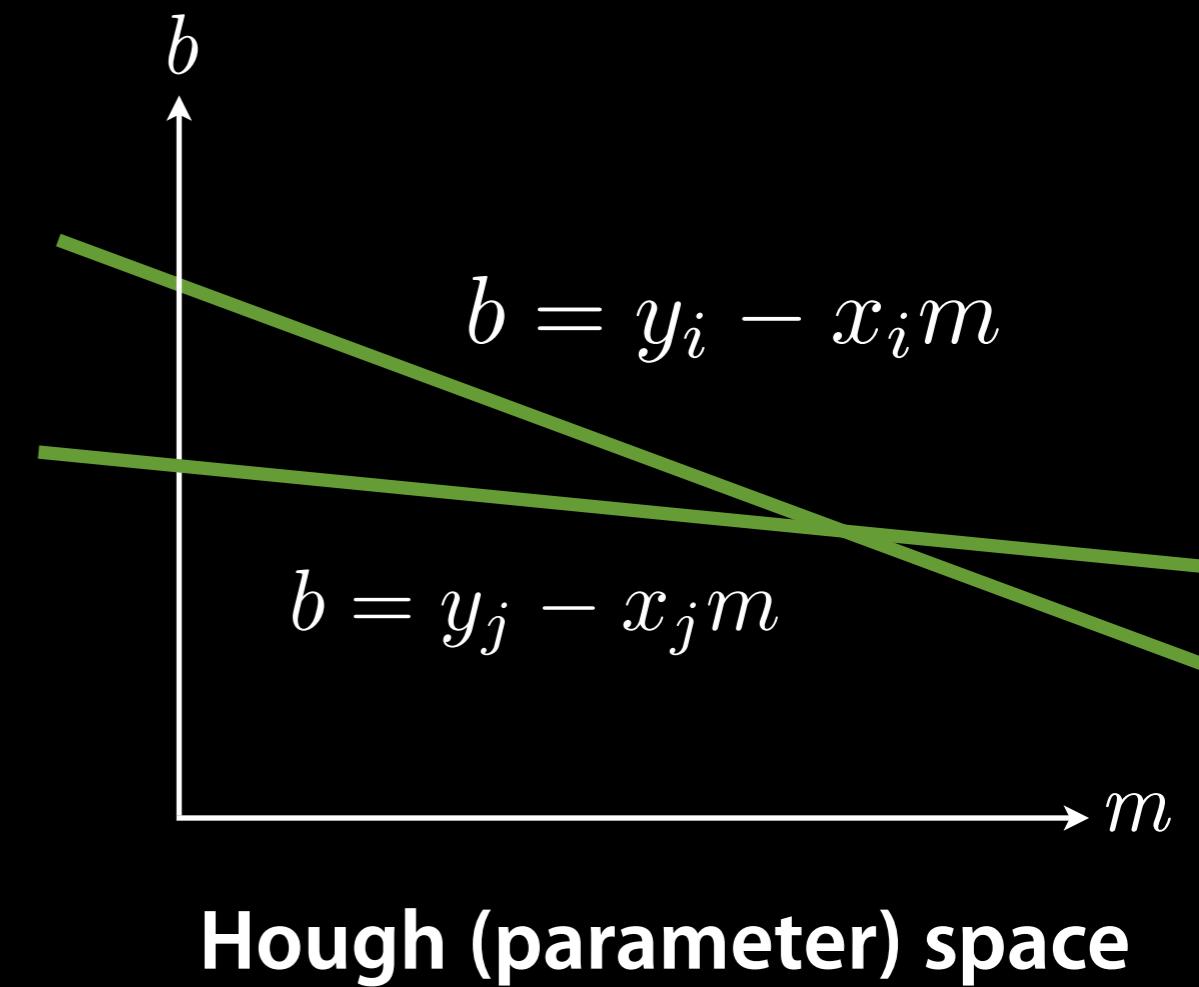
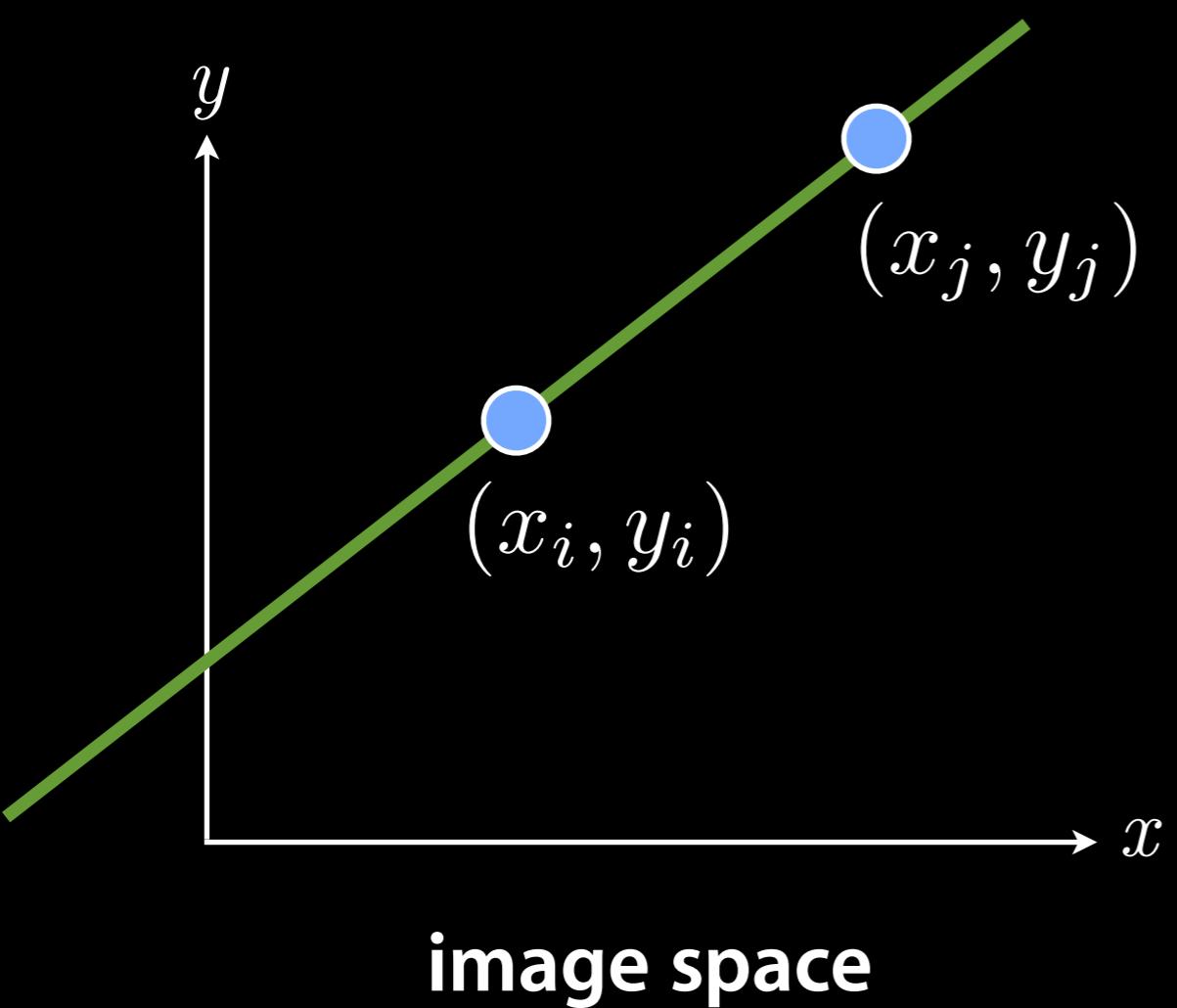
Hough Lines



Hough Lines



Hough Lines



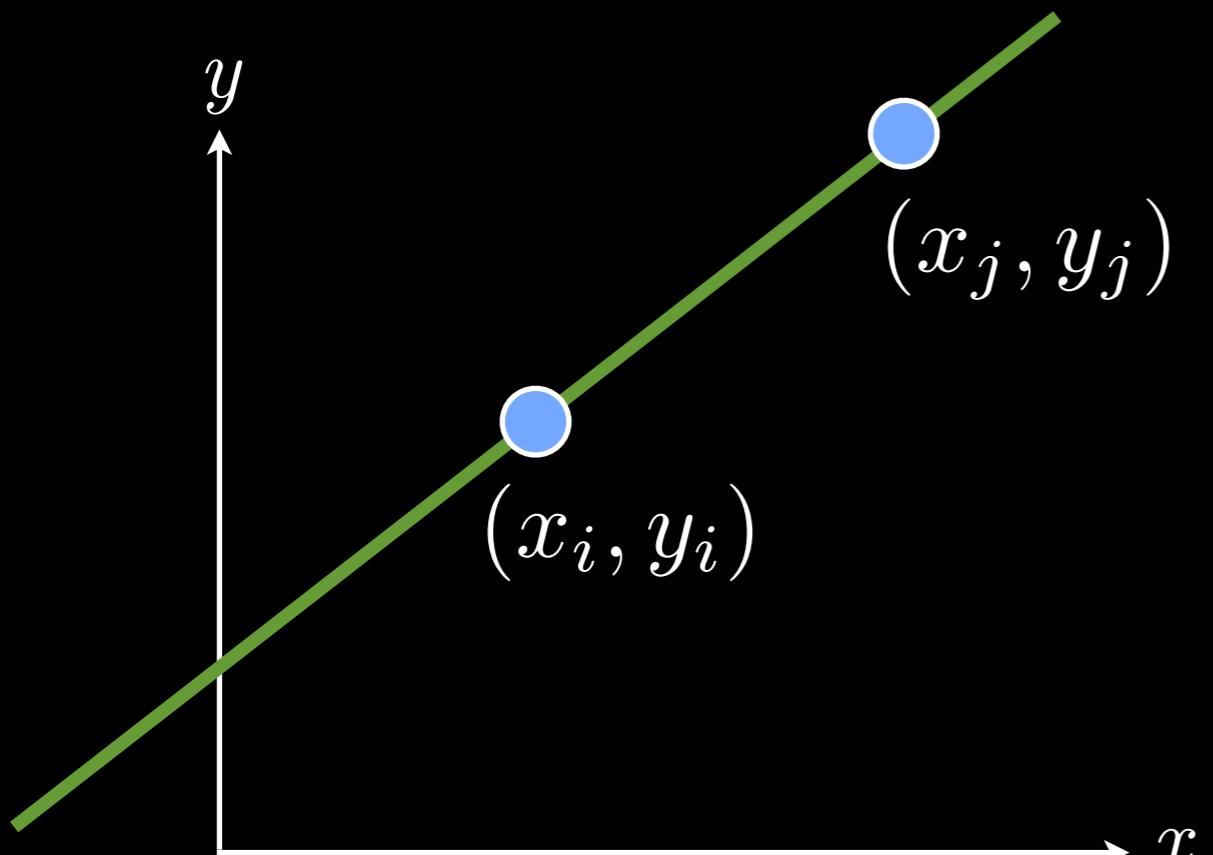
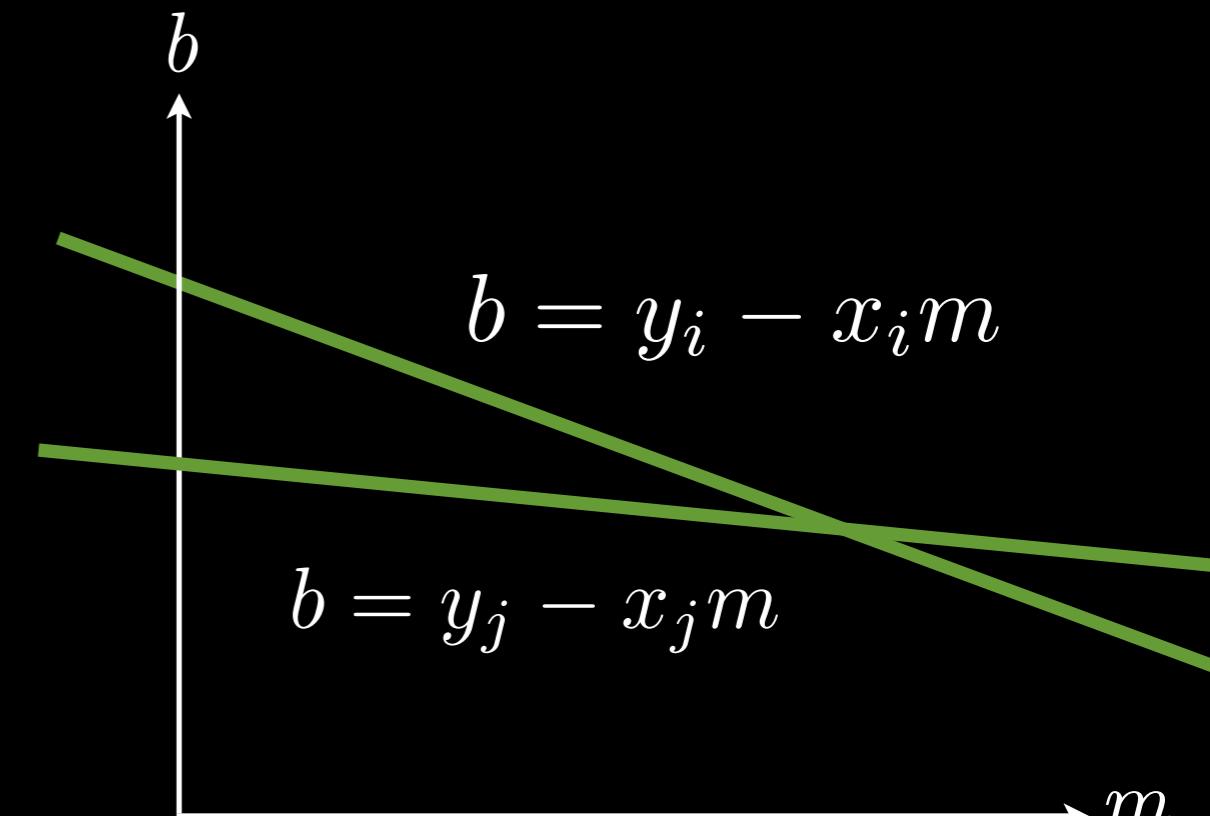


image space



Hough (parameter) space

What are the parameters of the line that contains both points?

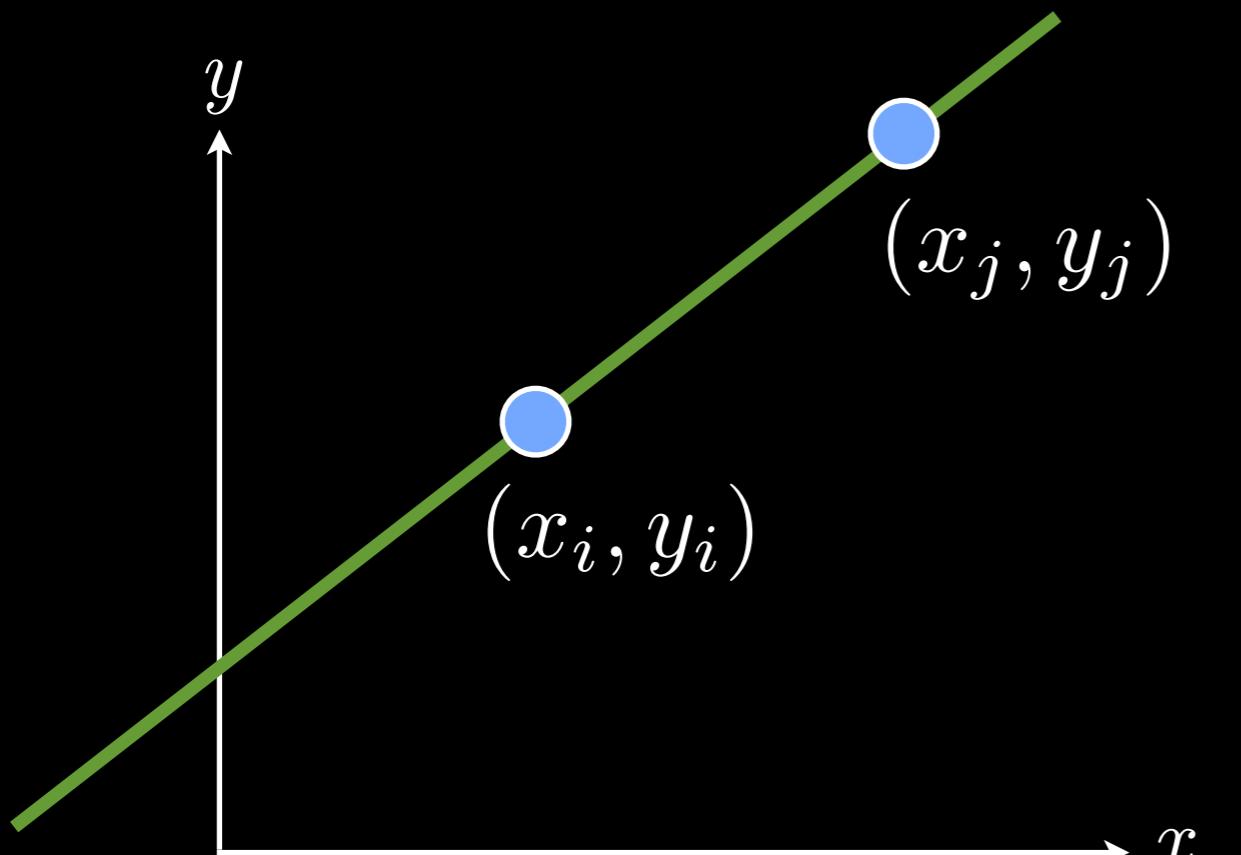
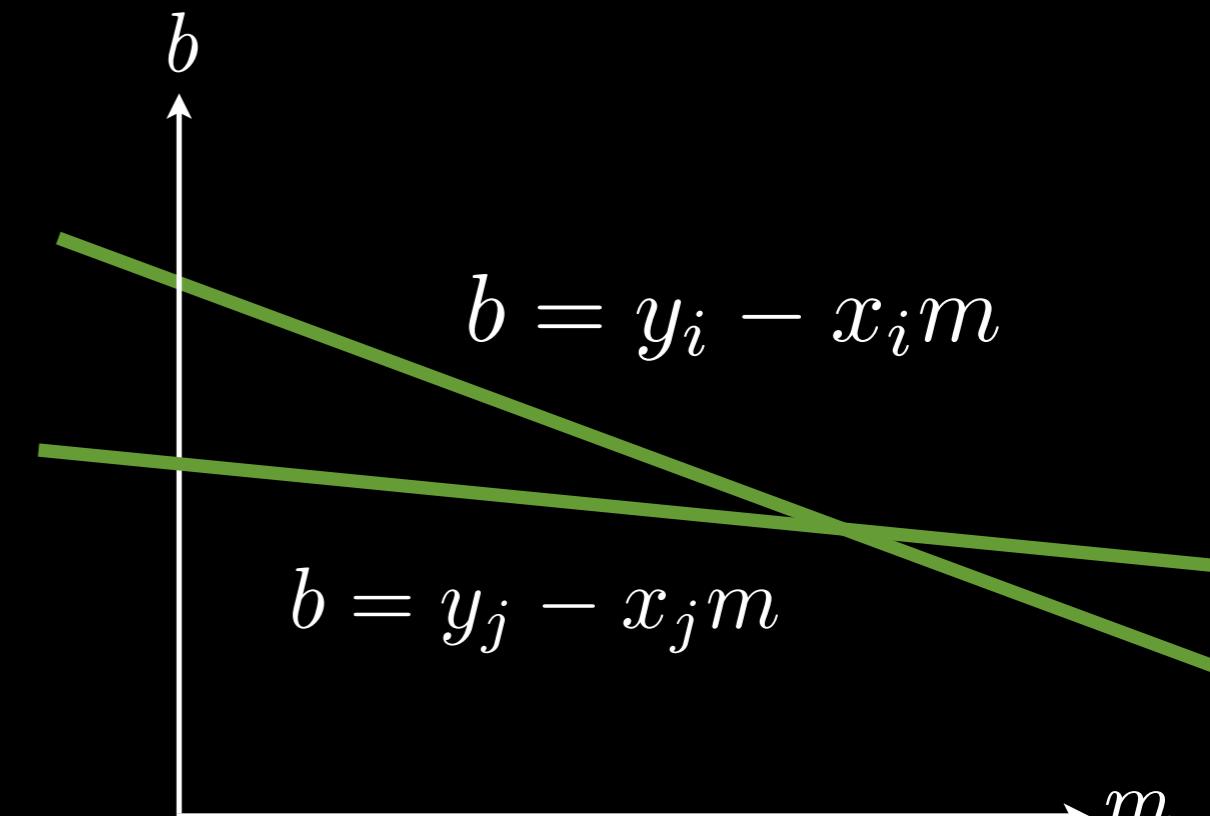


image space



Hough (parameter) space

What are the parameters of the line that contains both points?

Intersection of the two lines in Hough space

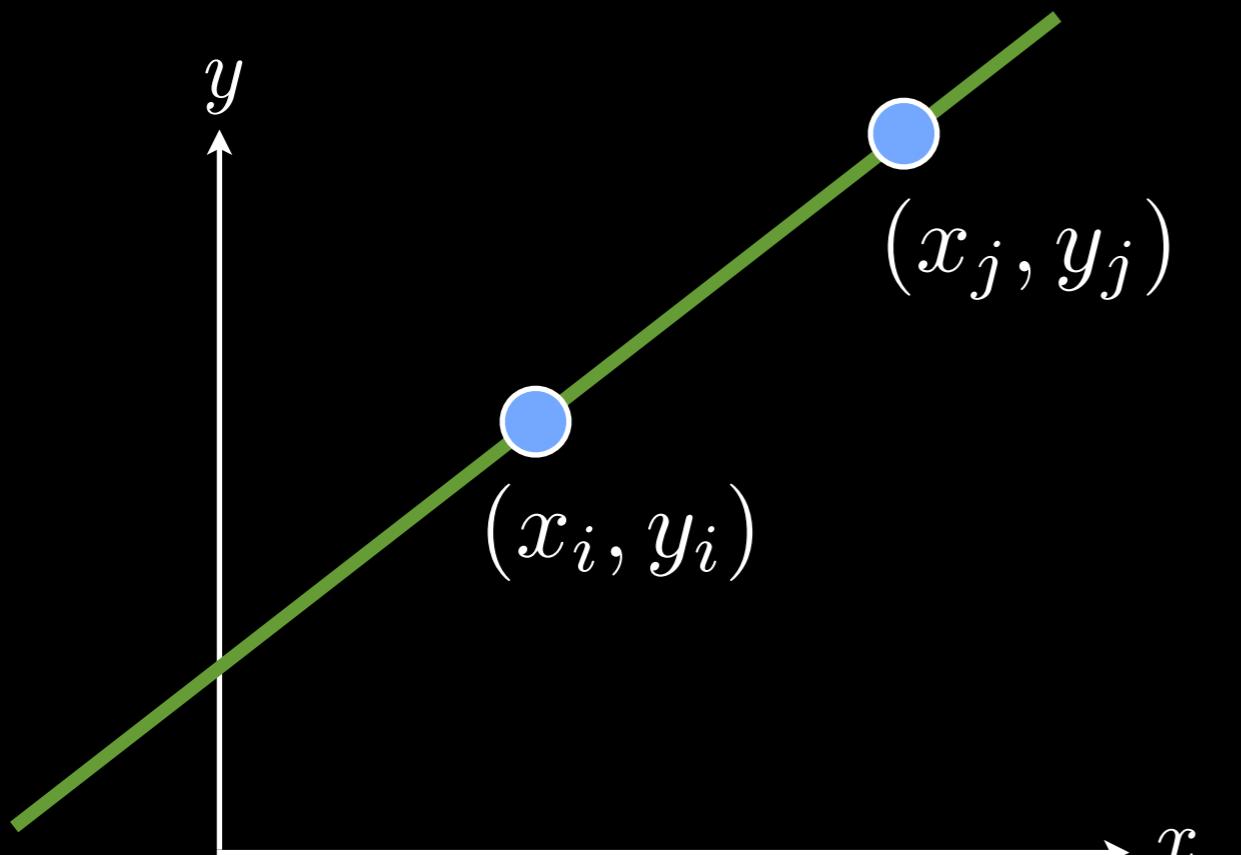
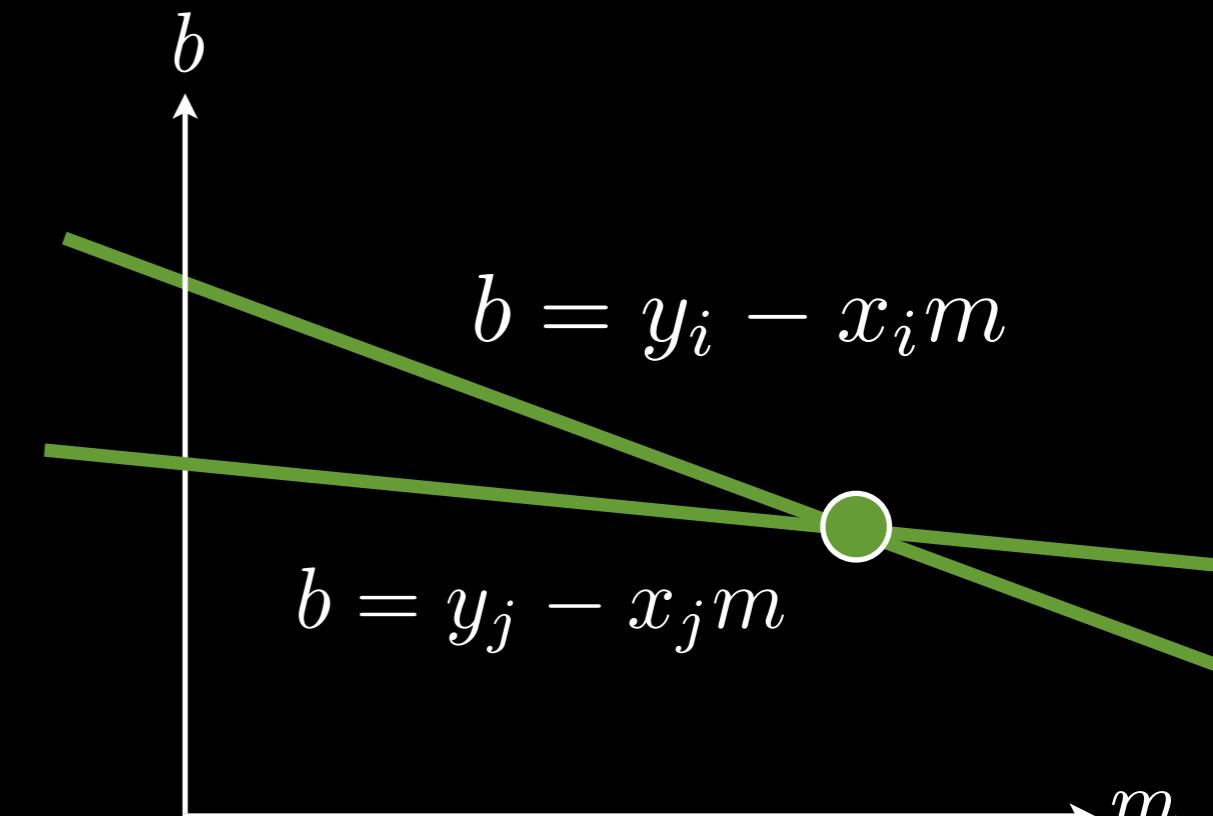


image space



Hough (parameter) space

What are the parameters of the line that contains both points?

Intersection of the two lines in Hough space

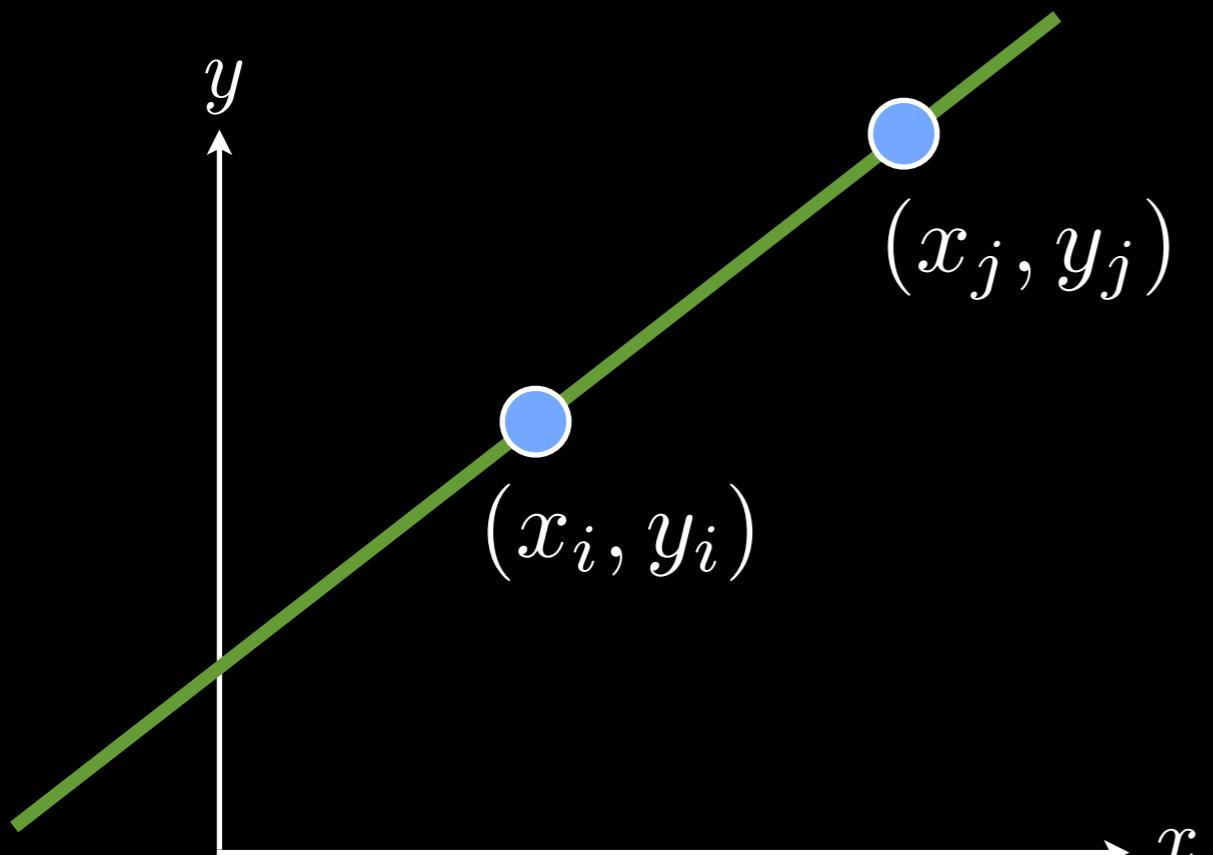
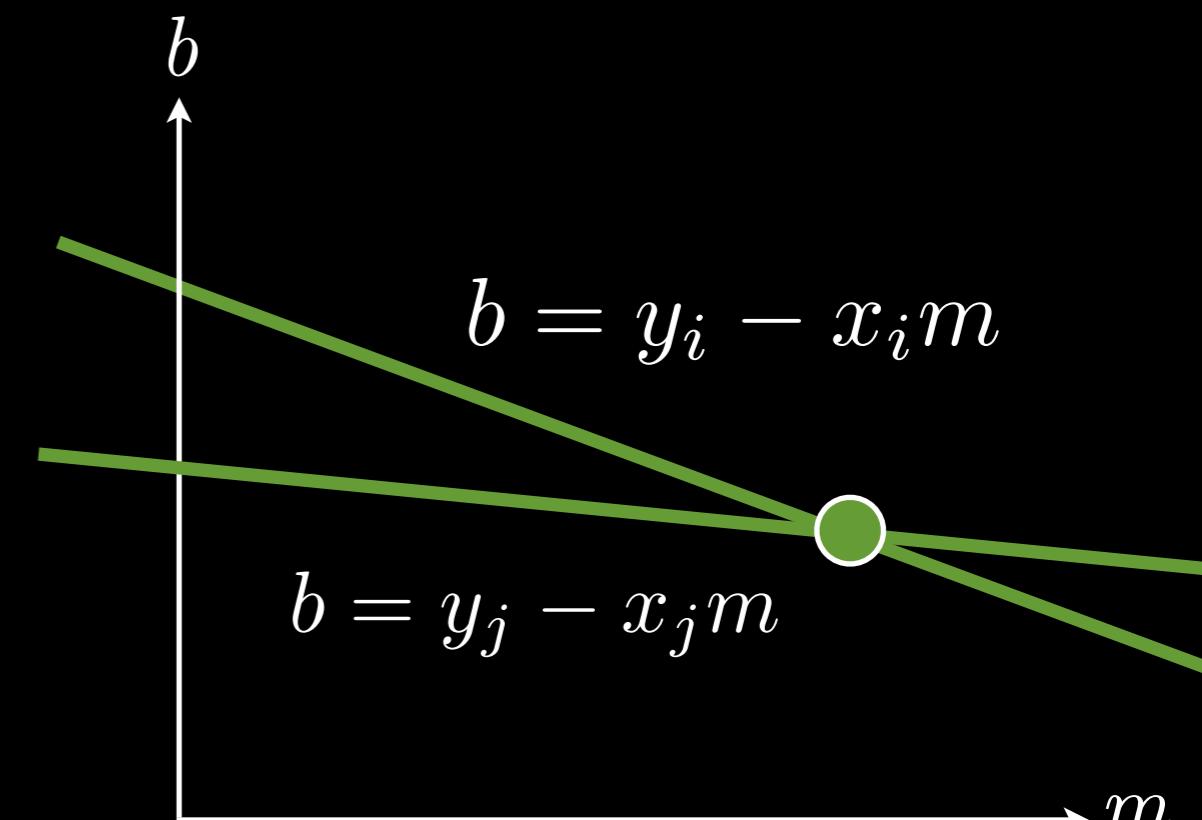


image space



Hough (parameter) space

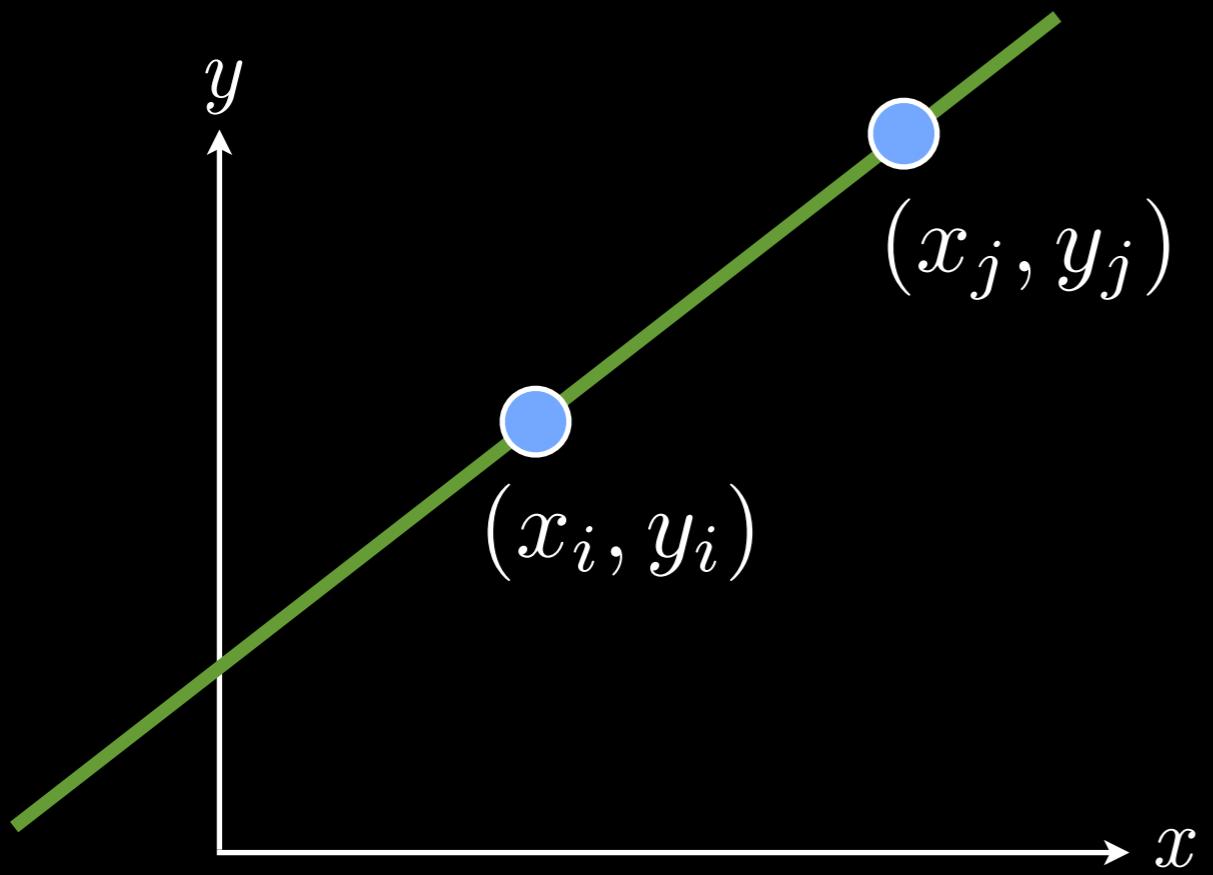
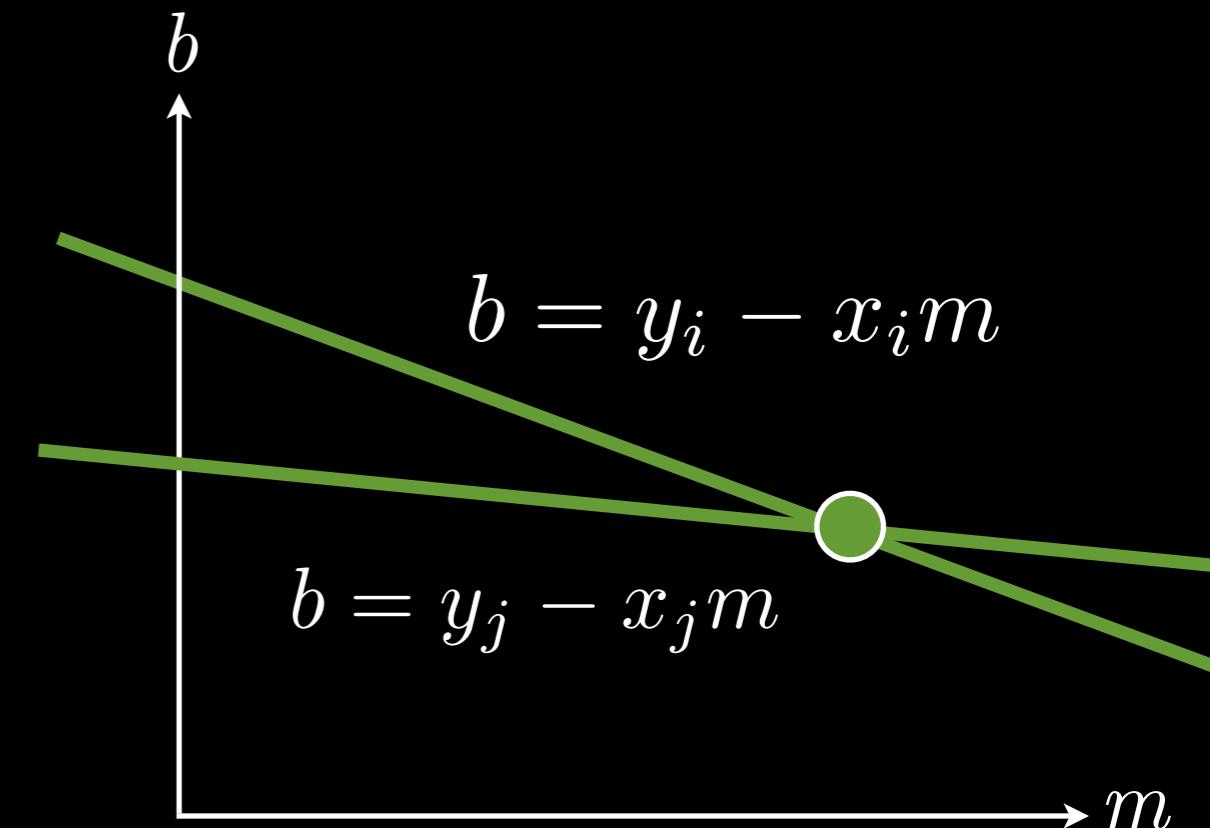


image space



Hough (parameter) space

The parameter space is continuous!
How can we handle this?

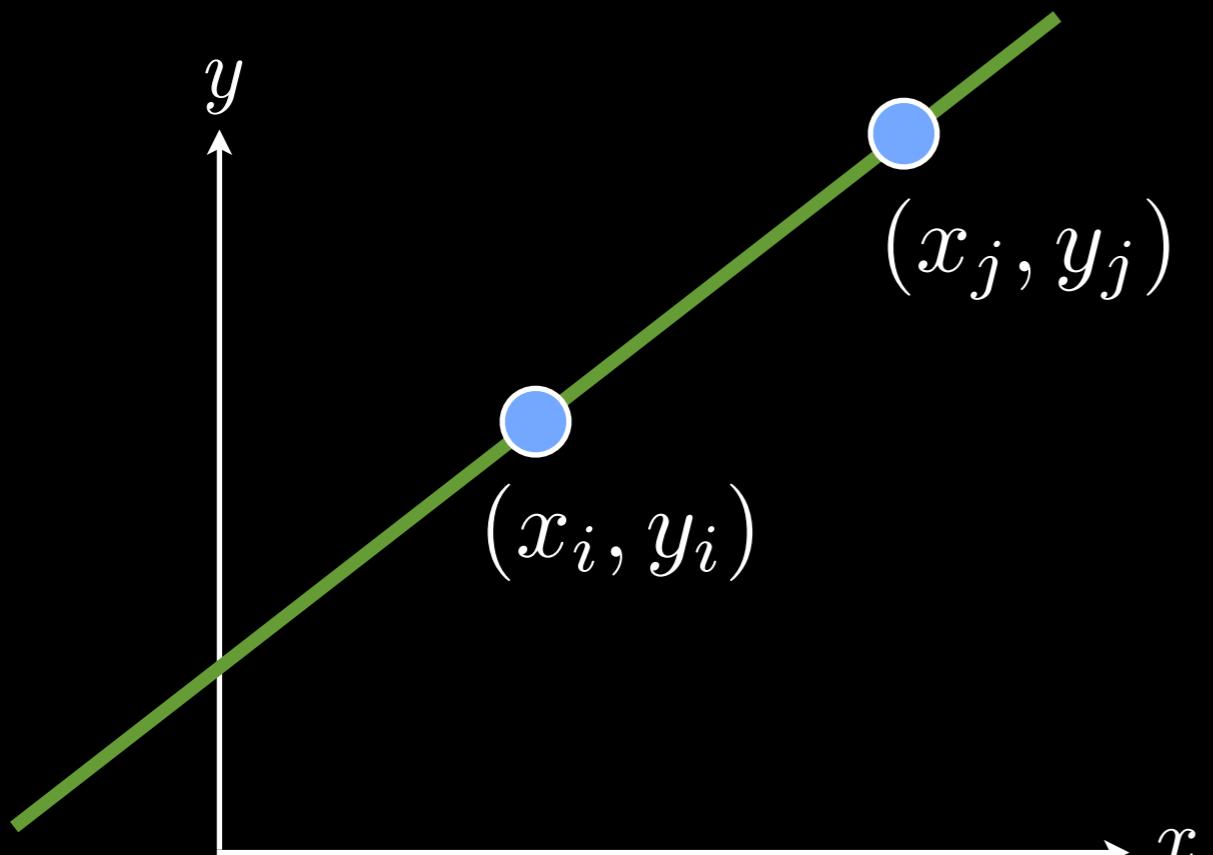
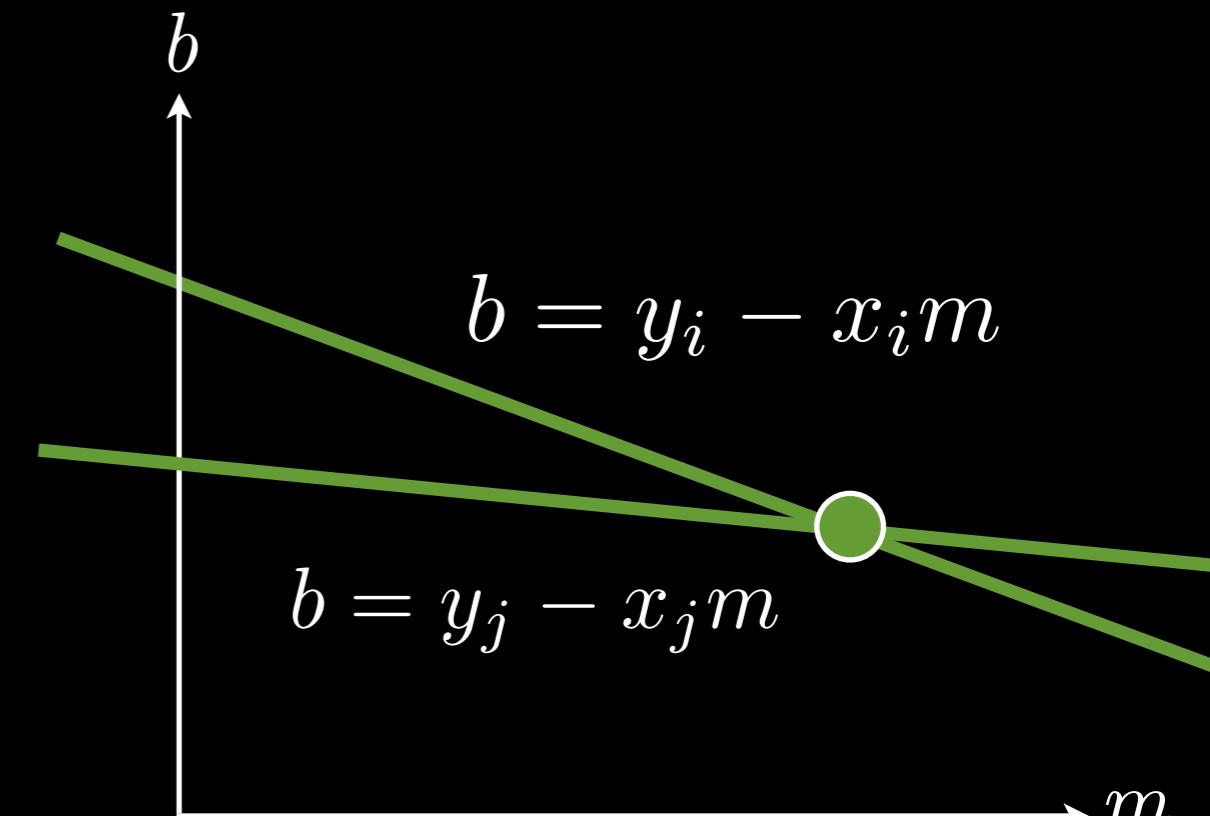


image space



Hough (parameter) space

The parameter space is continuous!
How can we handle this?

Accumulate votes in discrete set of bins

H : accumulator array

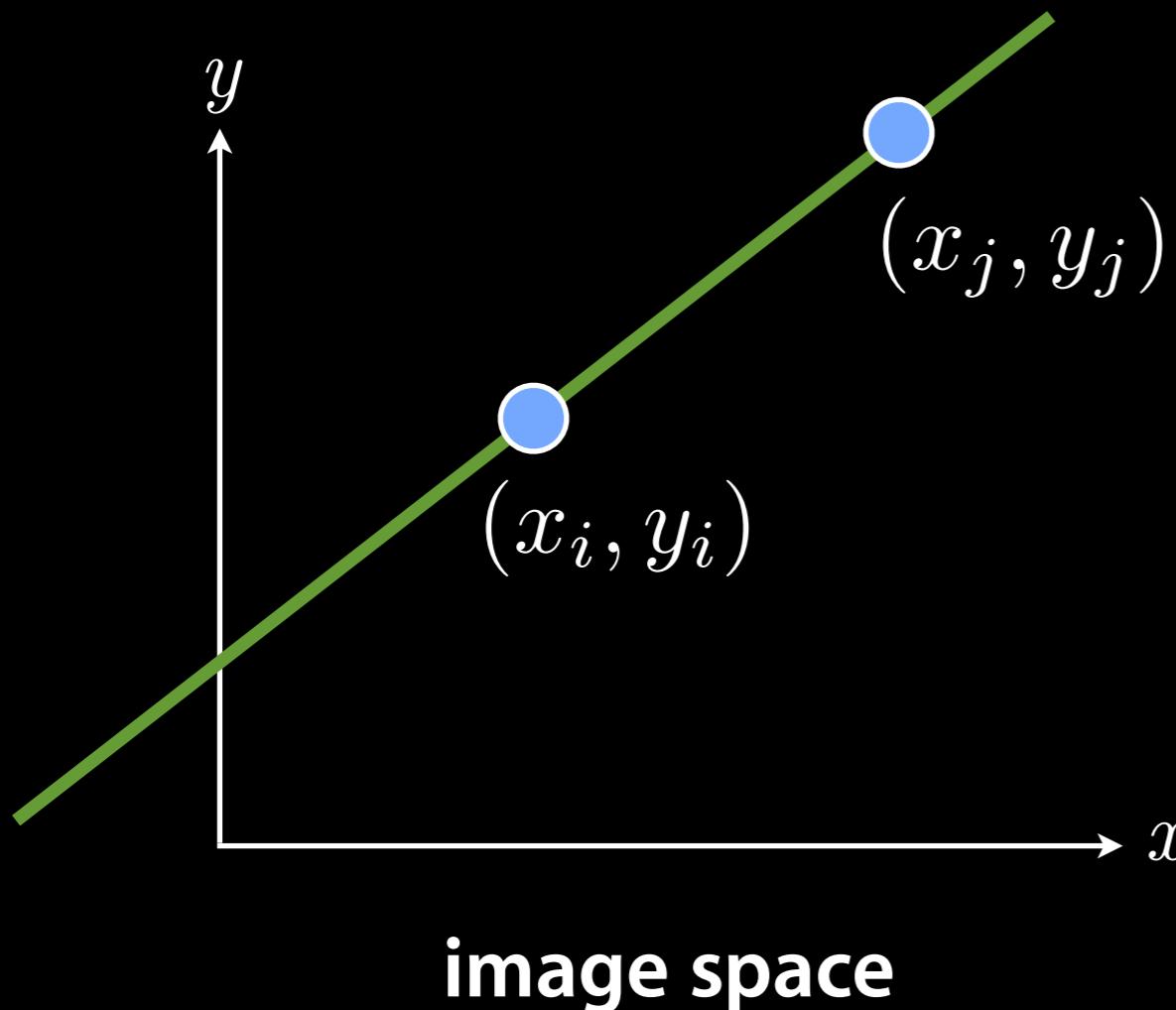
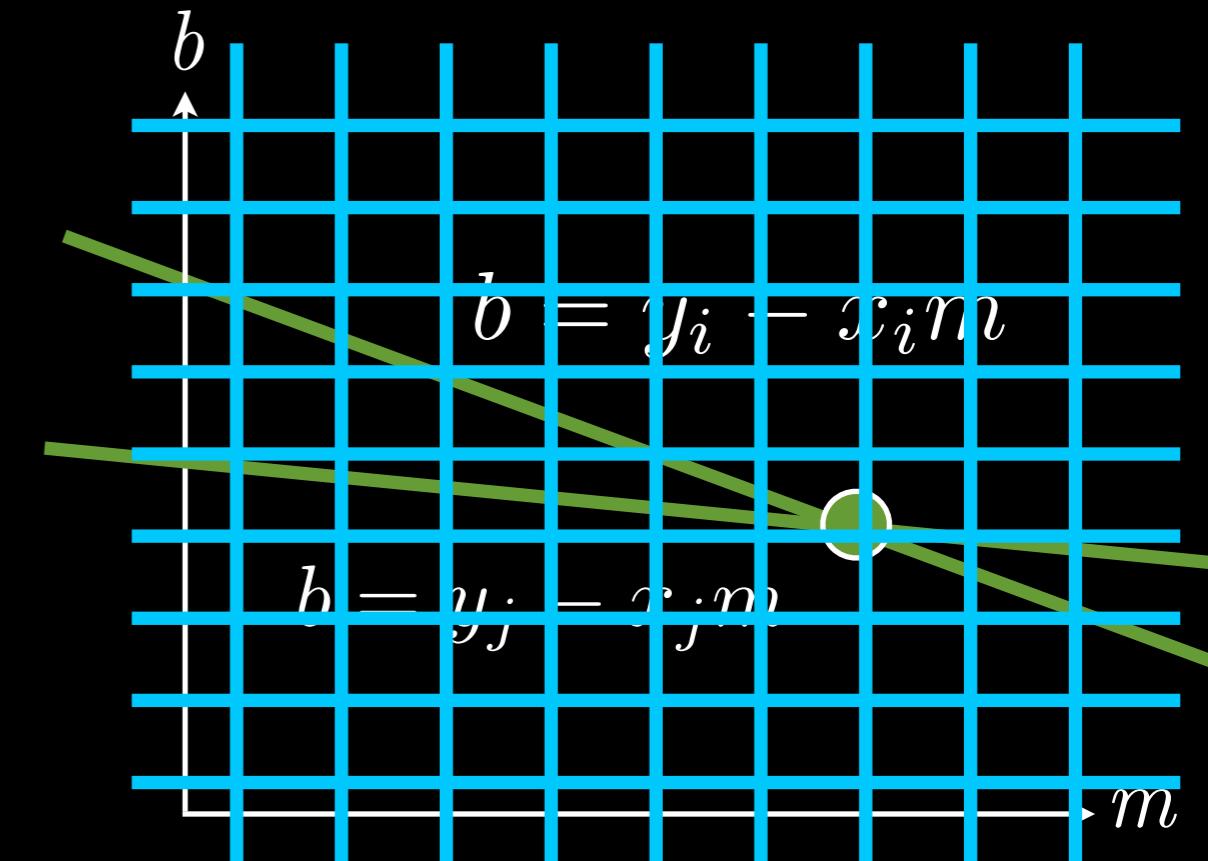


image space



Hough (parameter) space

The parameter space is continuous!
How can we handle this?

Accumulate votes in discrete set of bins

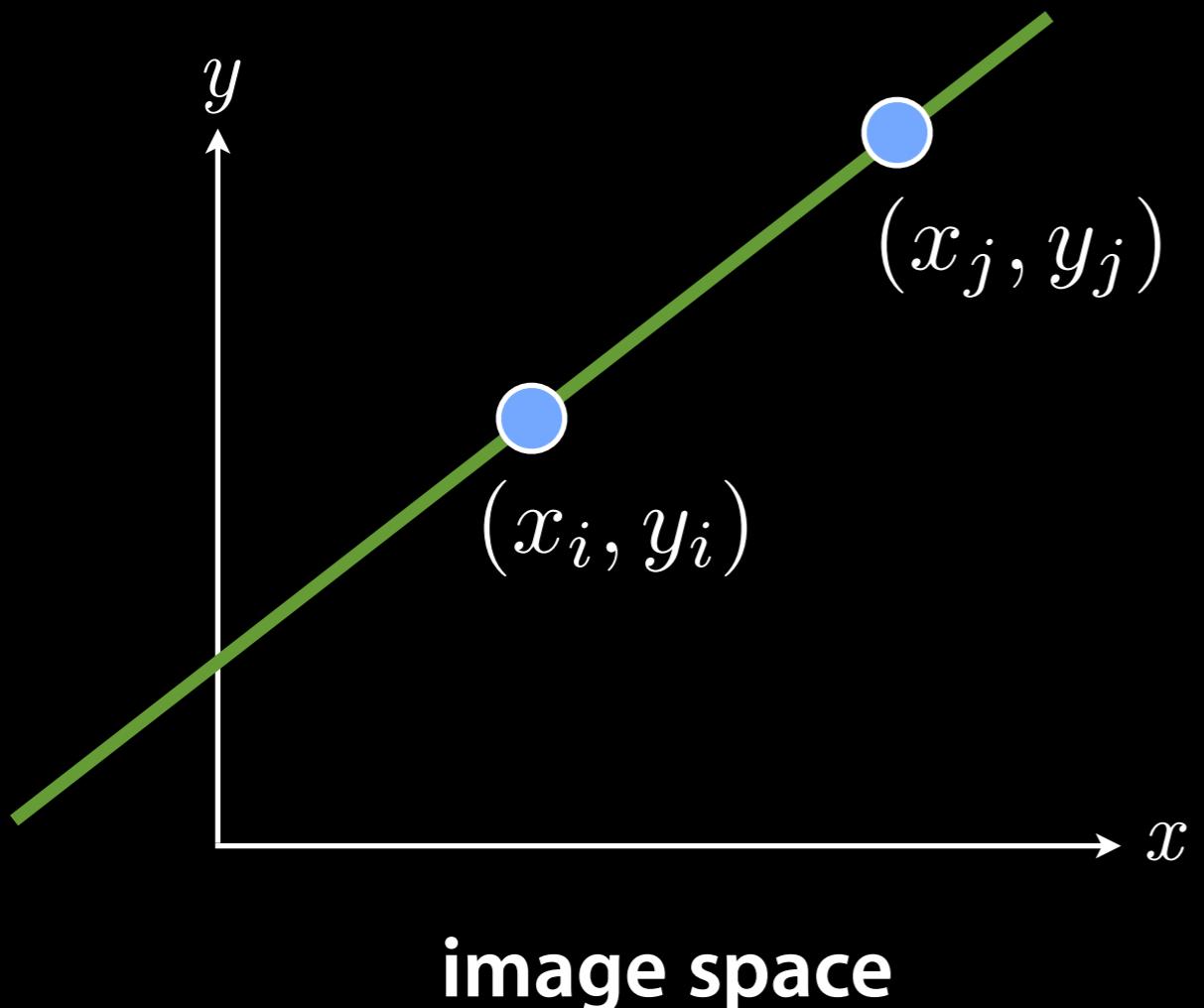
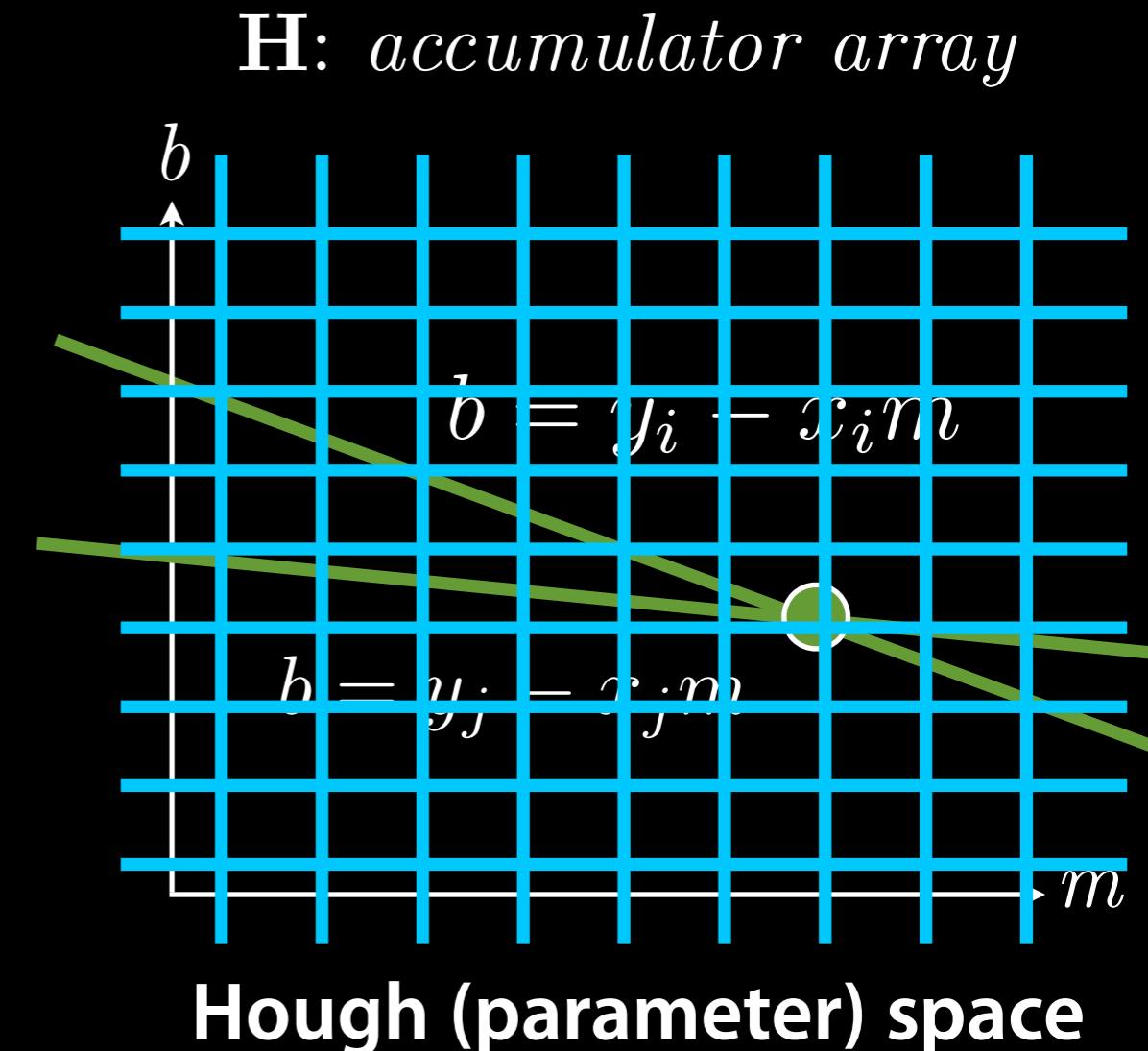


image space



Hough (parameter) space

Local peaks correspond to detected lines

Slope-
Intercept

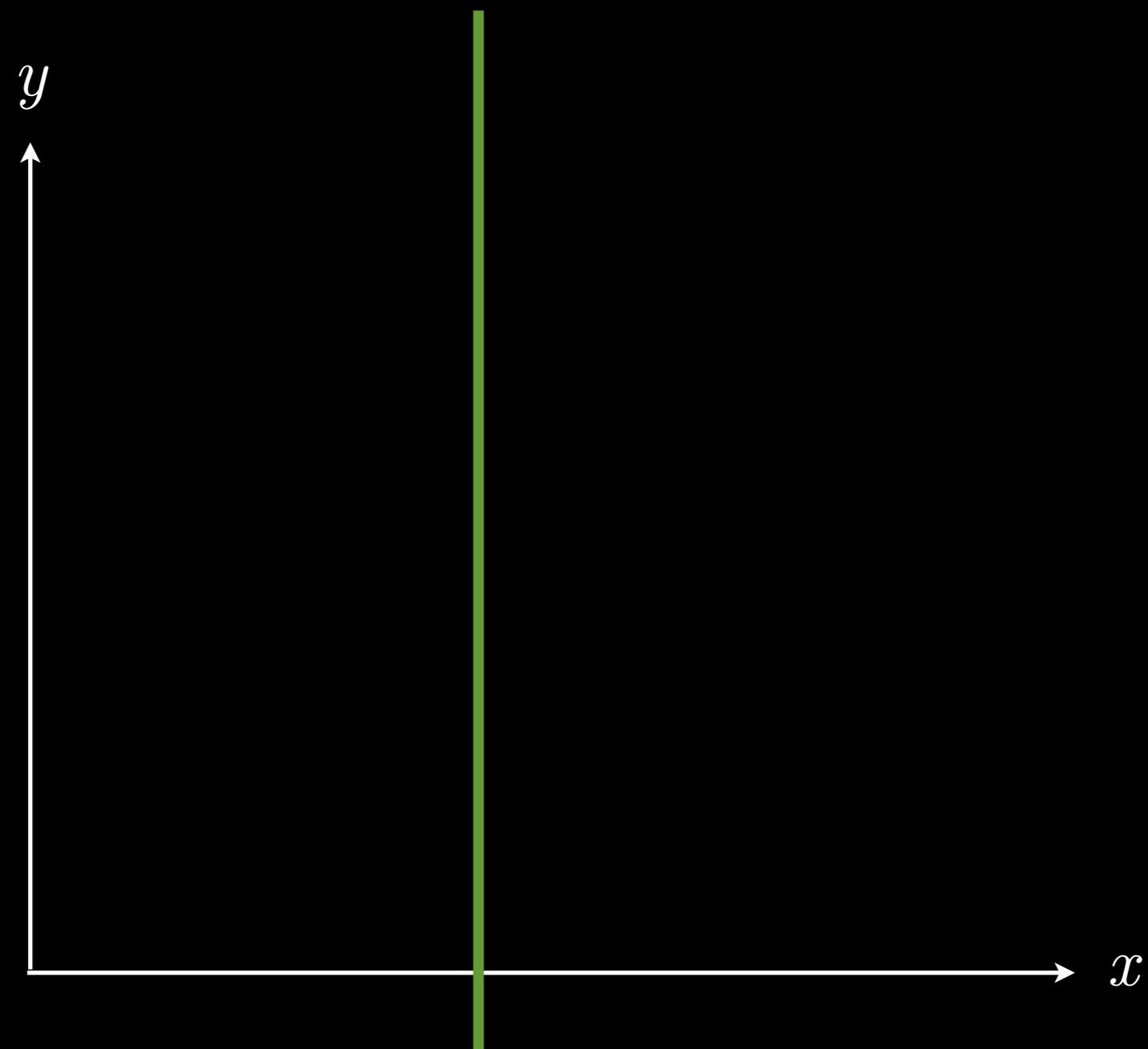


image space

Slope-
Intercept

What is the equation of the vertical line?

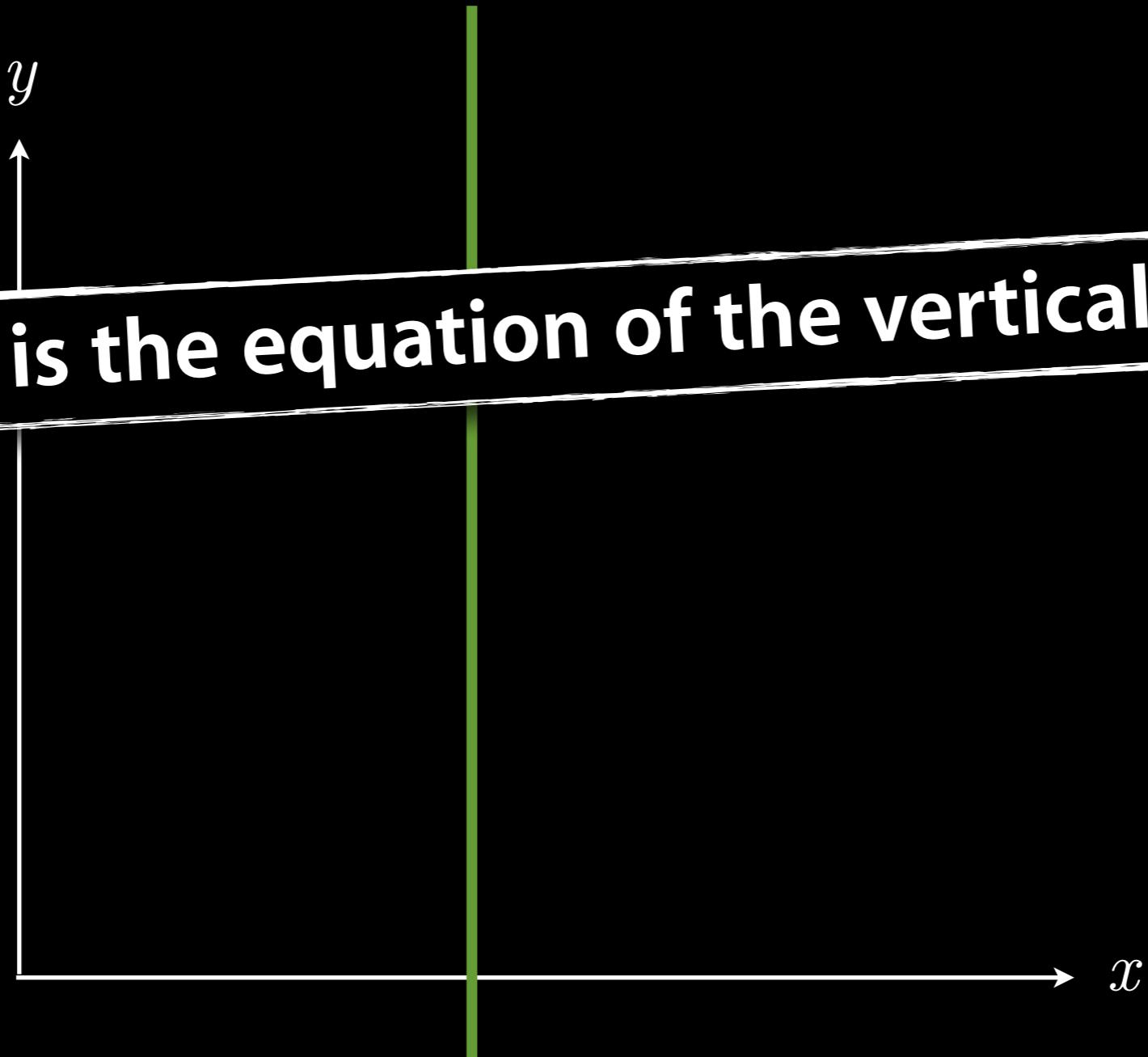
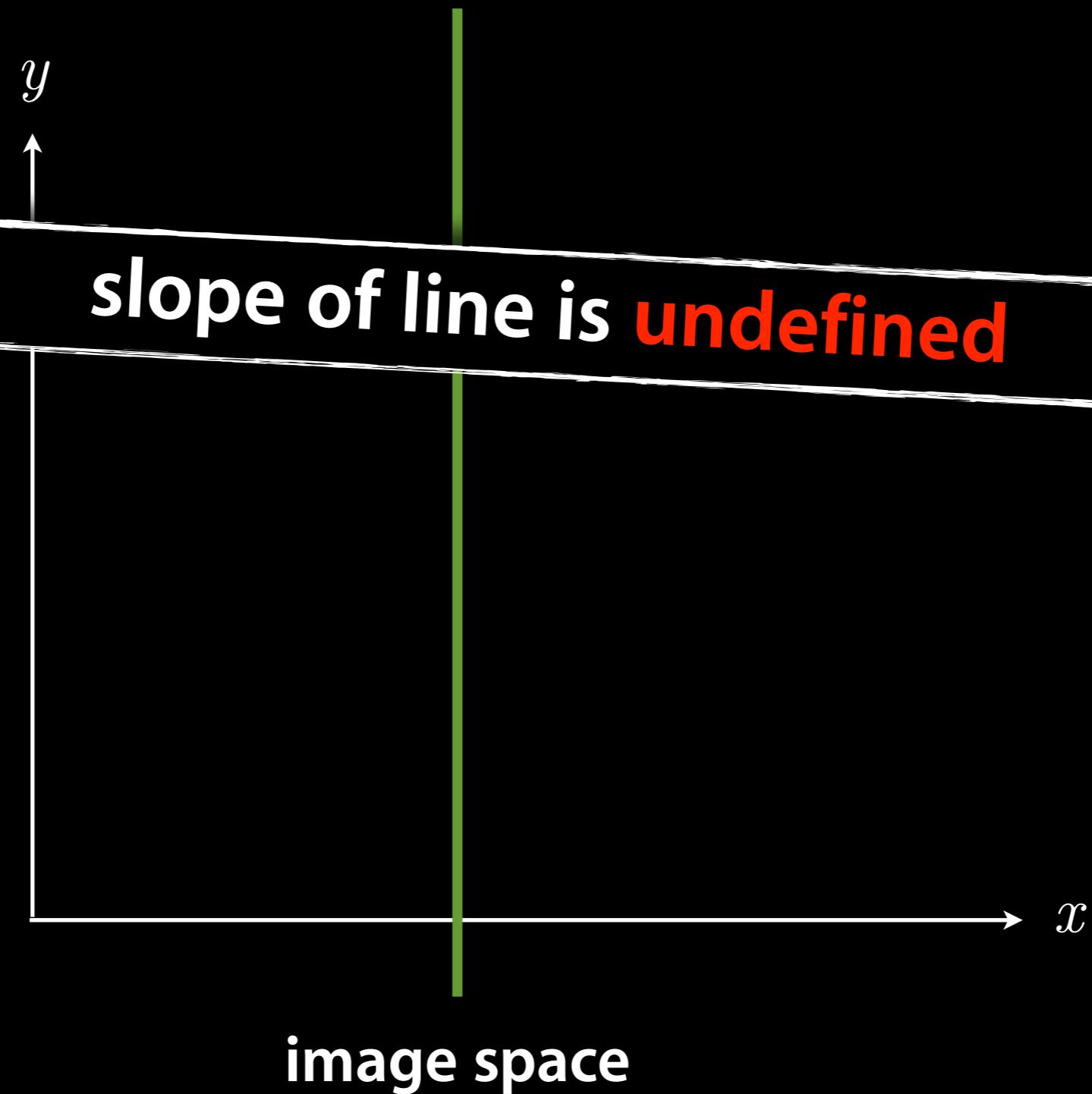
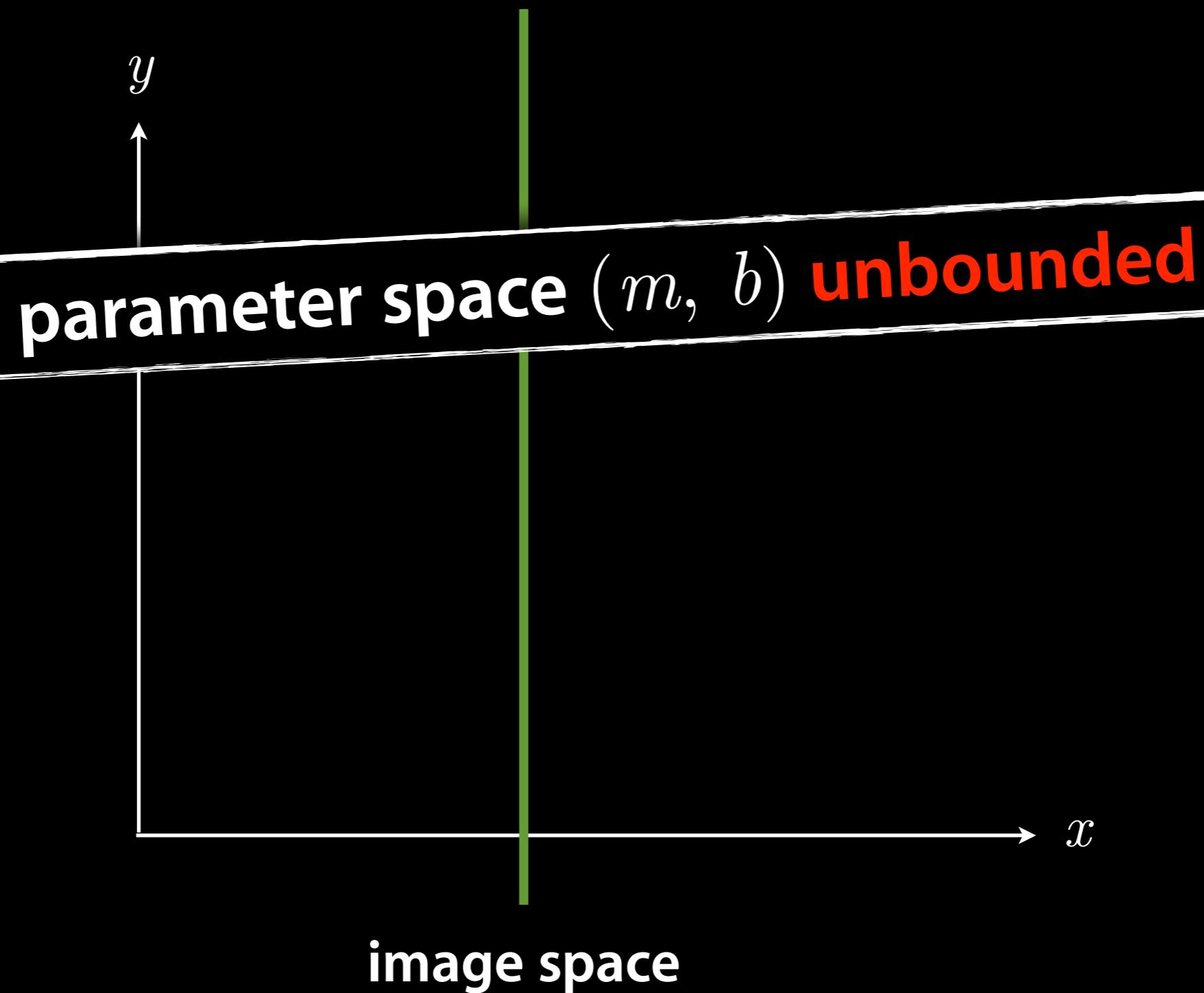


image space

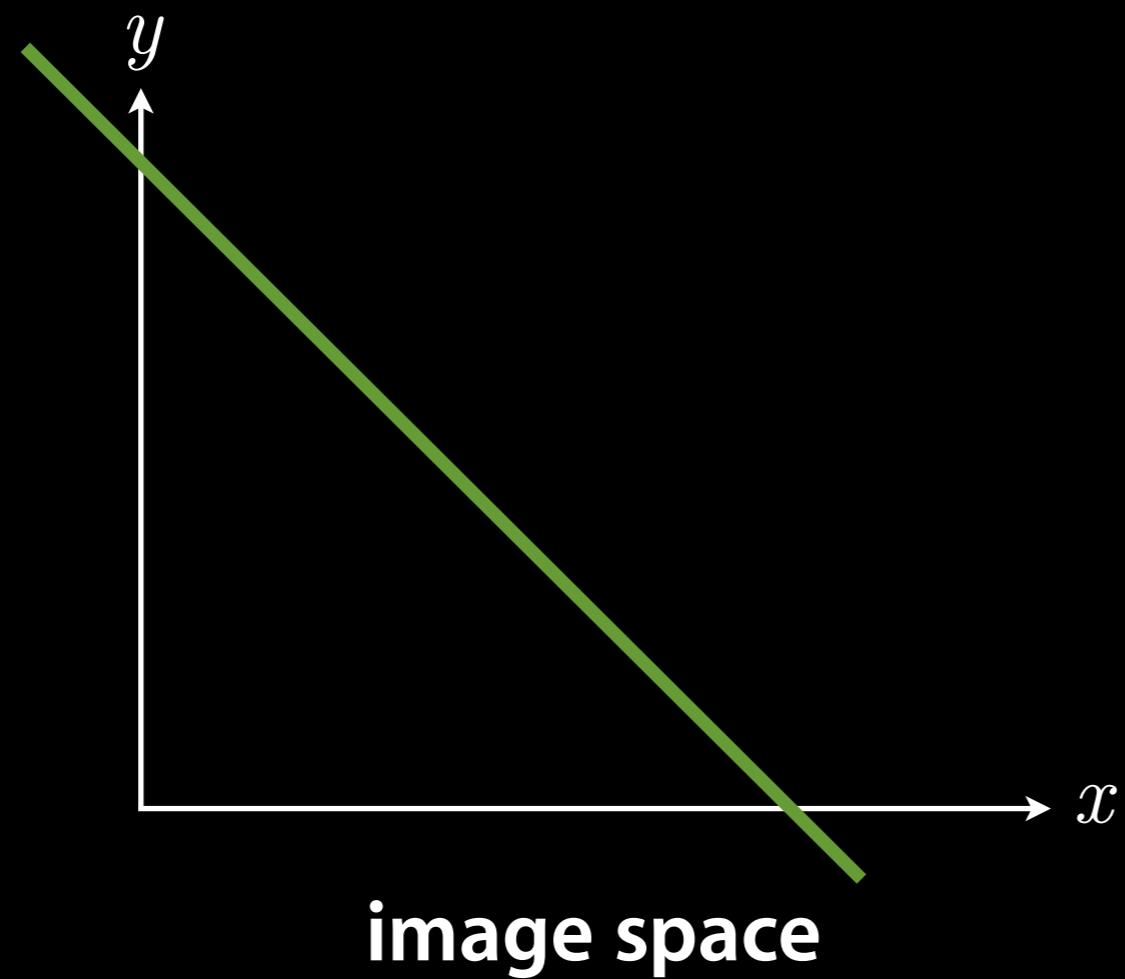
Slope-
Intercept



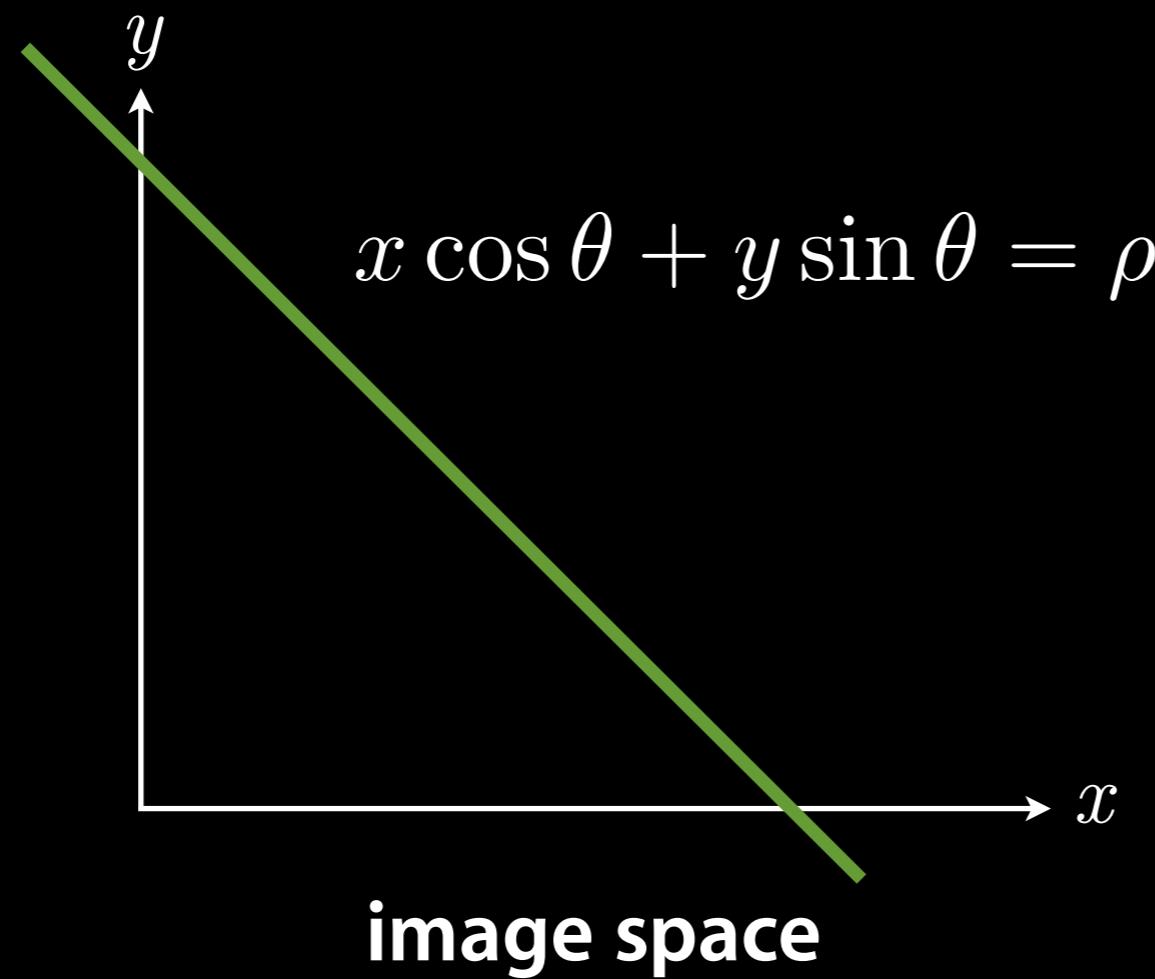
Slope-
Intercept



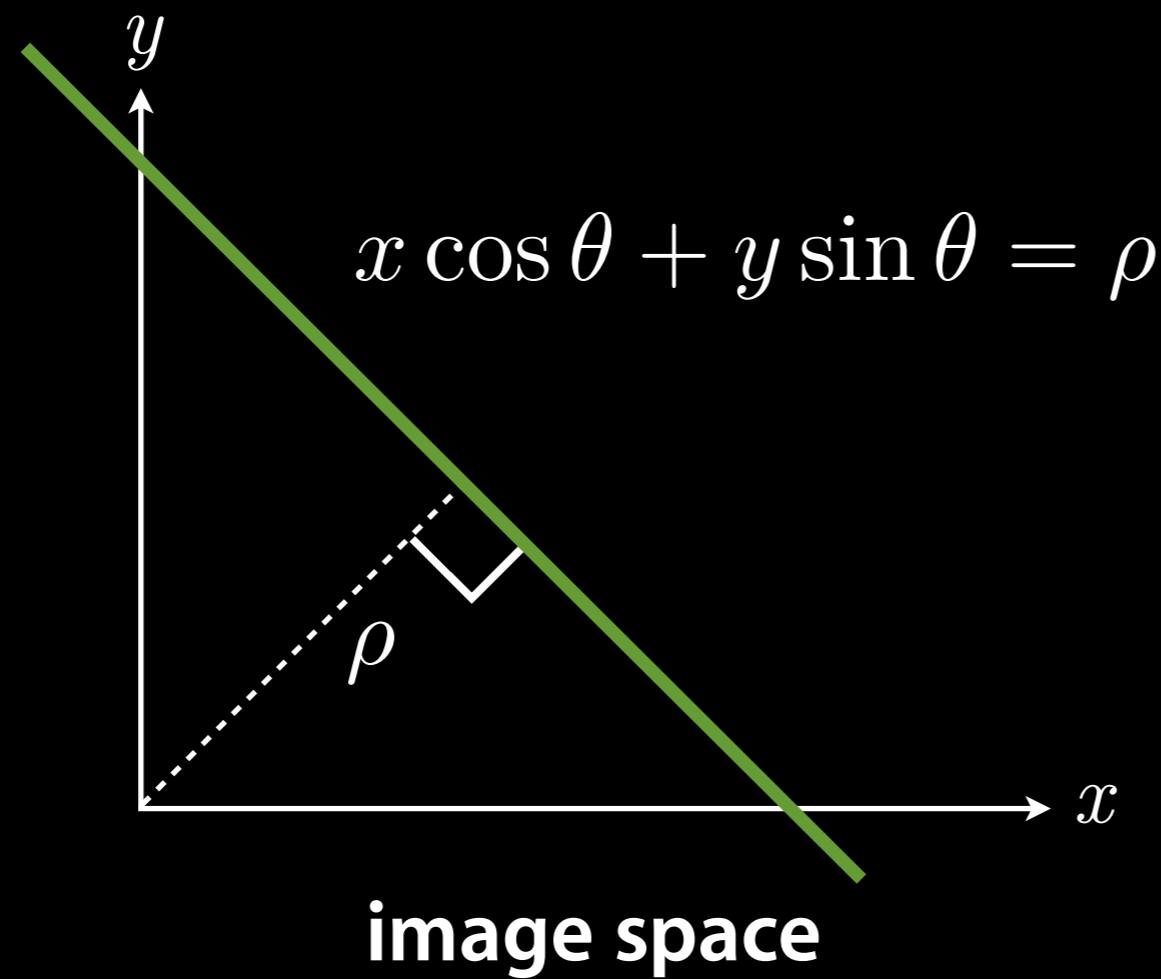
Polar
Representation



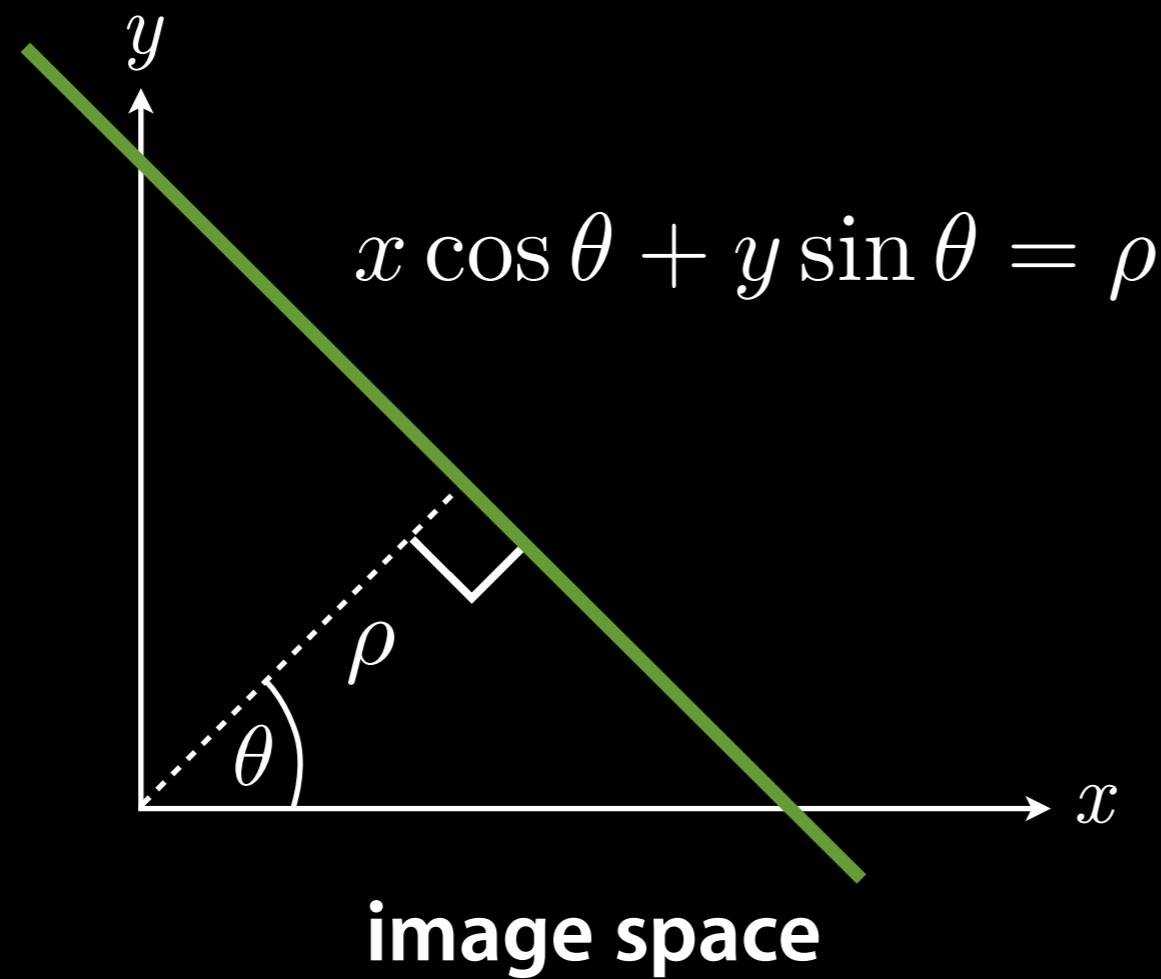
Polar Representation



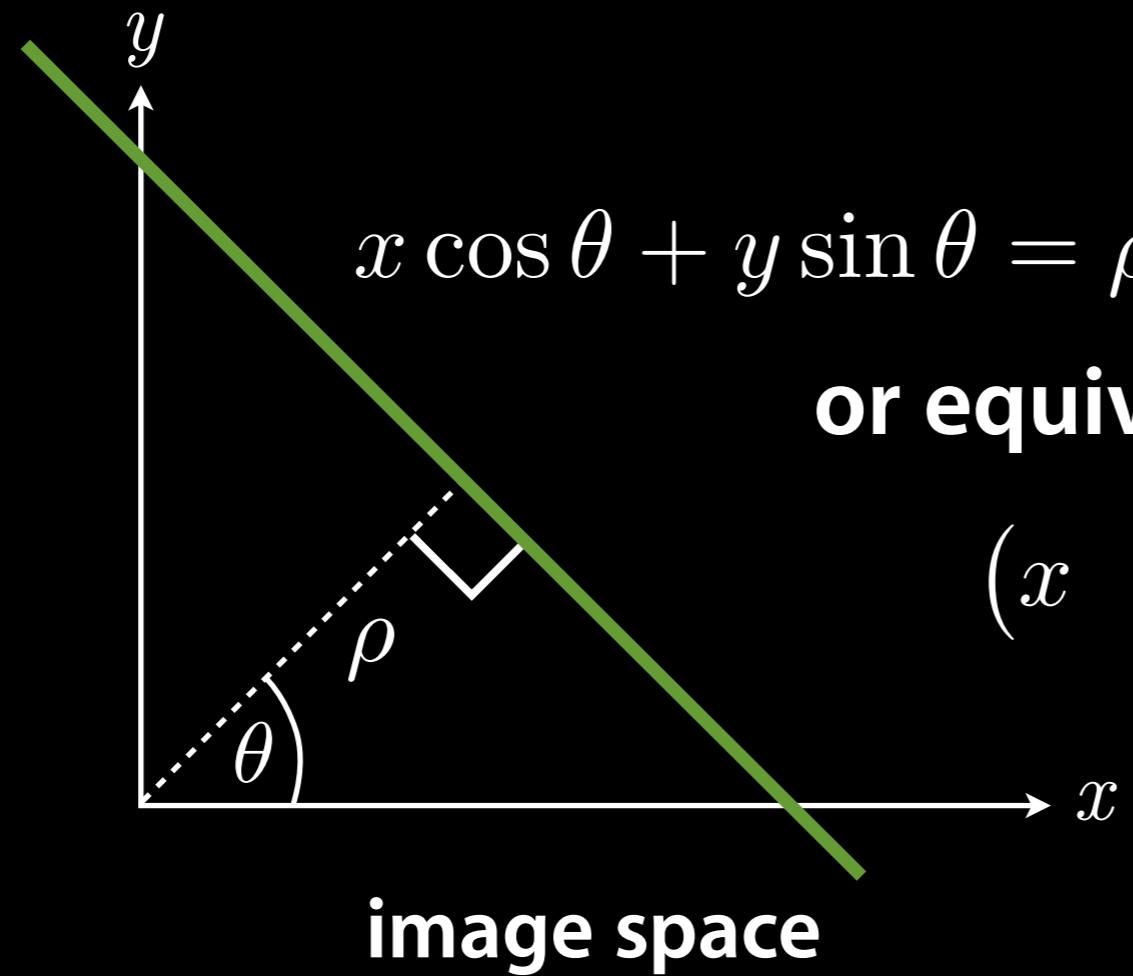
Polar Representation



Polar Representation



Polar Representation



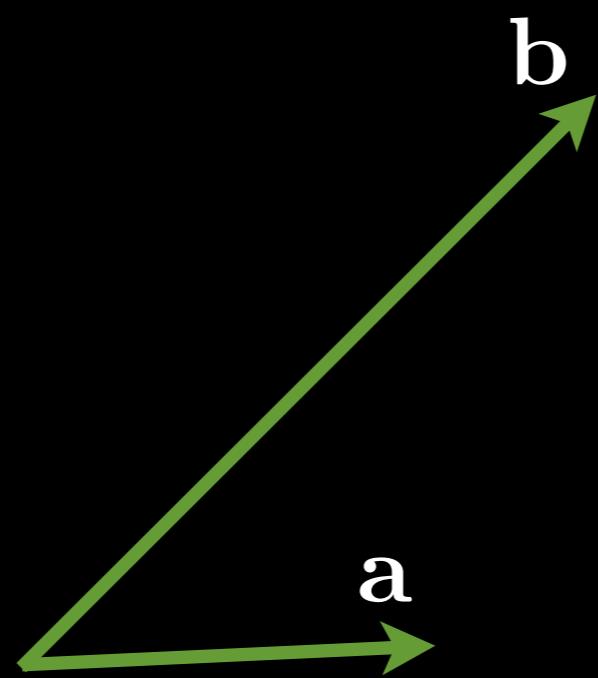
LINEAR ALGEBRA
AND ITS APPLICATIONS
GILBERT STRANG

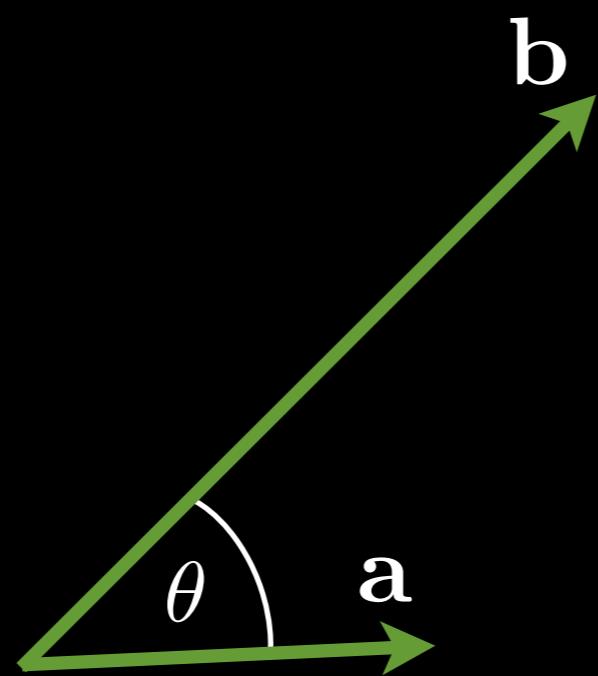
Linear Algebra Review

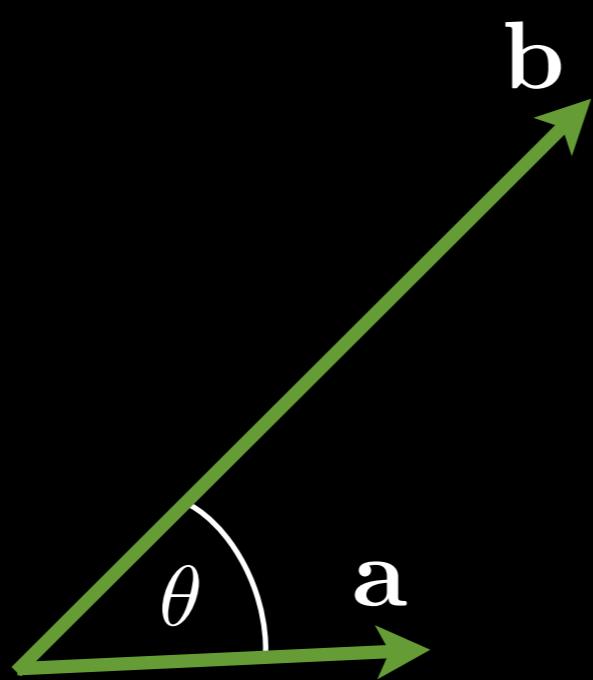
LINEAR ALGEBRA
AND ITS APPLICATIONS
GILBERT STRANG

Linear Algebra Review

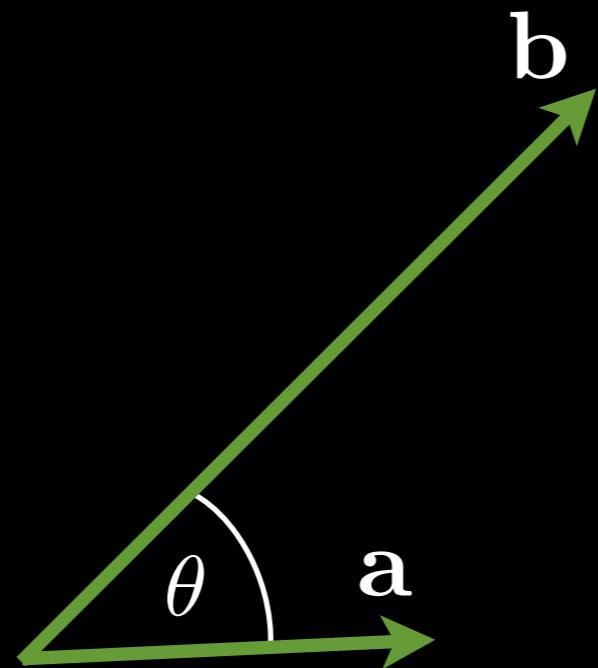




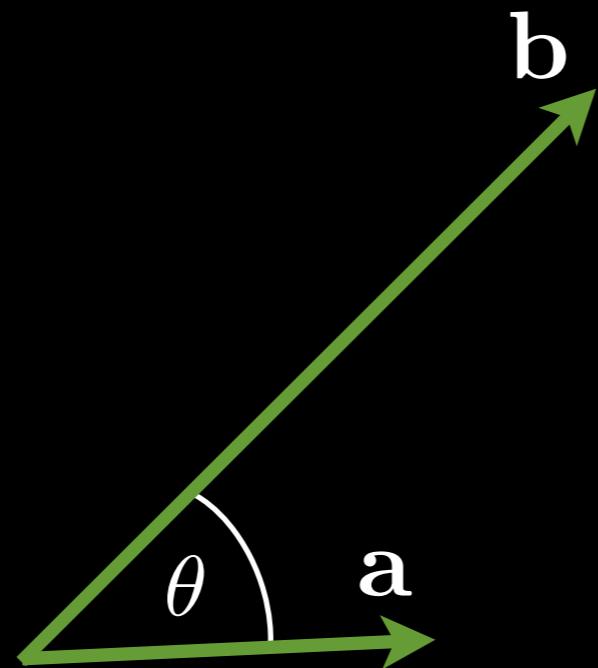




$$\mathbf{a} \cdot \mathbf{b}$$

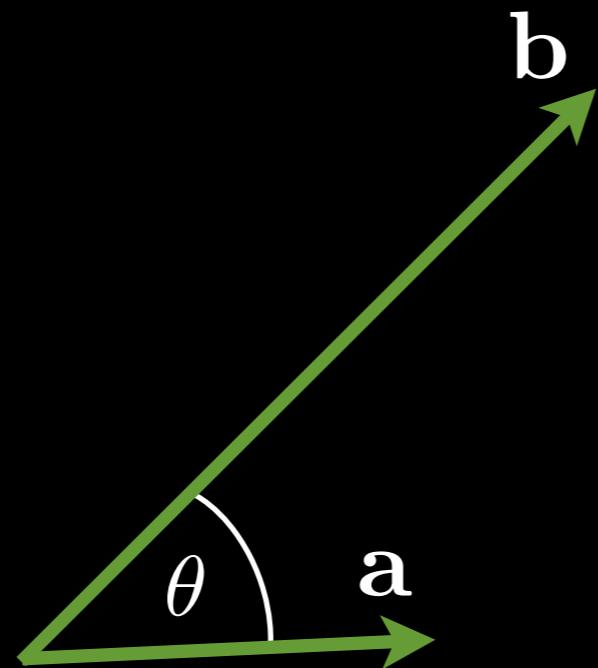


$$\mathbf{a} \cdot \mathbf{b} = \|\mathbf{a}\| \|\mathbf{b}\| \cos \theta$$



$$\mathbf{a} \cdot \mathbf{b} = \|\mathbf{a}\| \|\mathbf{b}\| \cos \theta$$

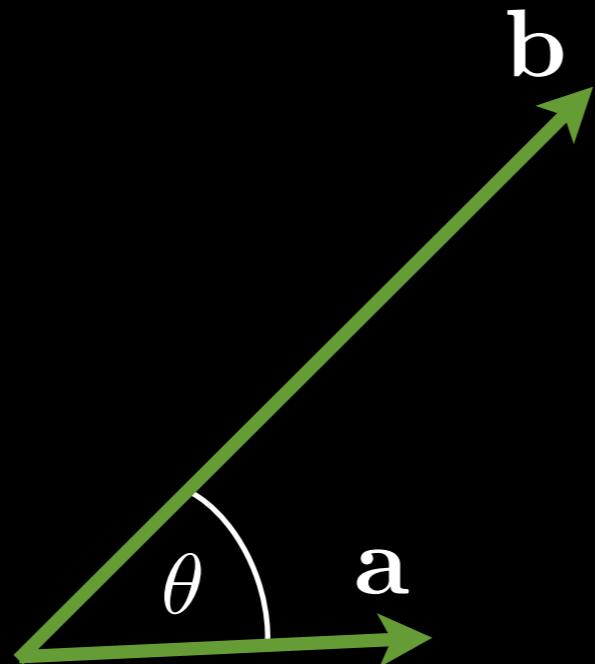
assume $\|\mathbf{a}\| = 1$



$$\mathbf{a} \cdot \mathbf{b} = \|\mathbf{a}\| \|\mathbf{b}\| \cos \theta$$

$$= \|\mathbf{b}\| \cos \theta$$

assume $\|\mathbf{a}\| = 1$

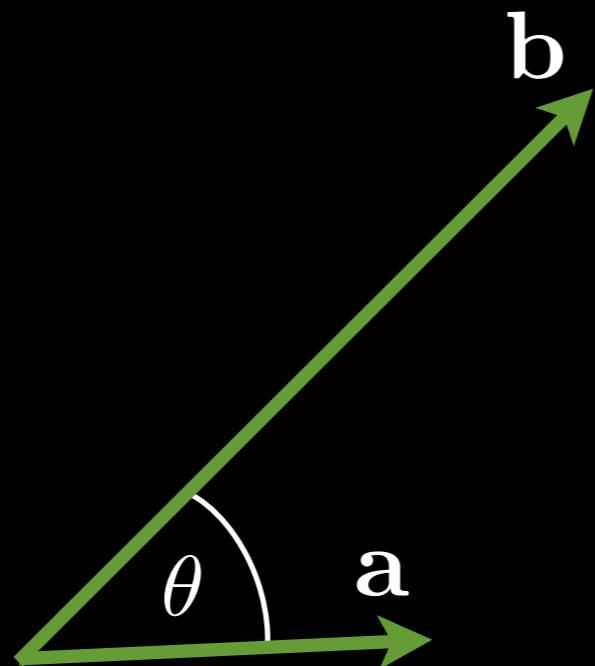


$$\mathbf{a} \cdot \mathbf{b} = \|\mathbf{a}\| \|\mathbf{b}\| \cos \theta$$

$$= \|\mathbf{b}\| \cos \theta$$

assume $\|\mathbf{a}\| = 1$

What is the geometric interpretation?

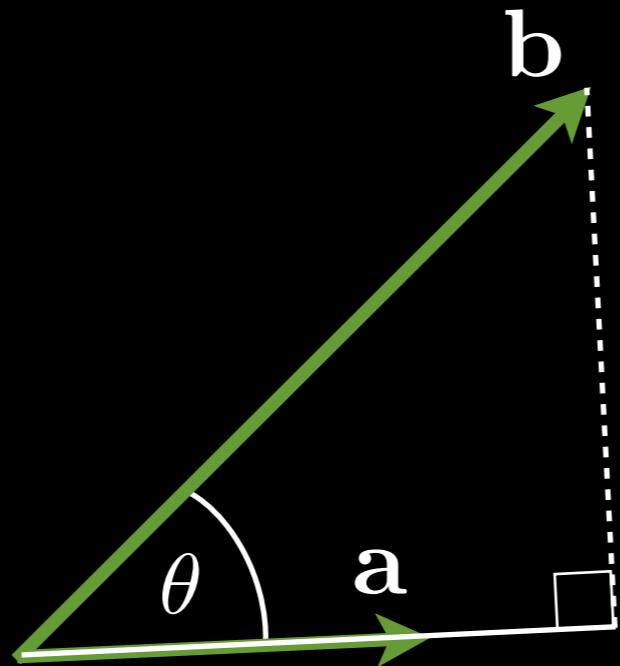


$$\mathbf{a} \cdot \mathbf{b} = \|\mathbf{a}\| \|\mathbf{b}\| \cos \theta$$

assume $\|\mathbf{a}\| = 1$

$$= \|\mathbf{b}\| \cos \theta$$

Length of projection of \mathbf{b} onto \mathbf{a}

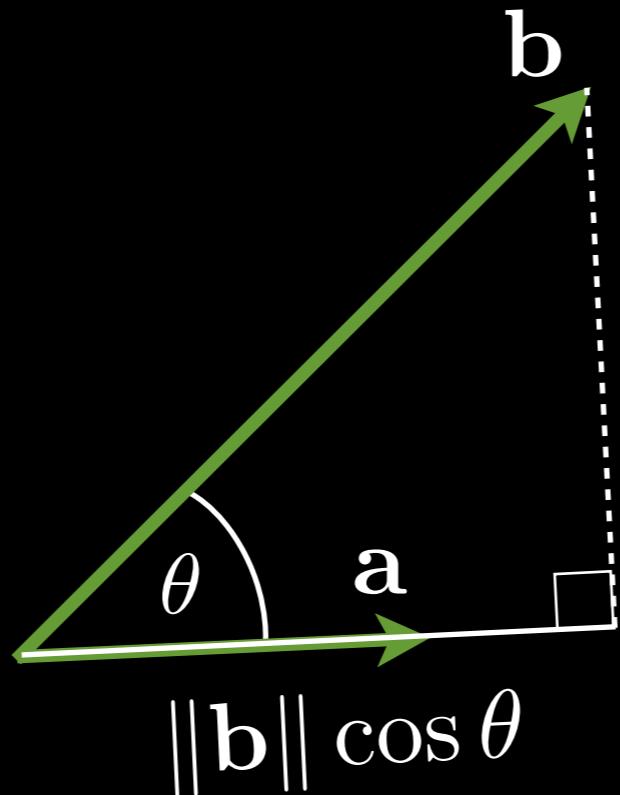


$$\mathbf{a} \cdot \mathbf{b} = \|\mathbf{a}\| \|\mathbf{b}\| \cos \theta$$

assume $\|\mathbf{a}\| = 1$

$$= \|\mathbf{b}\| \cos \theta$$

Length of projection of b onto a



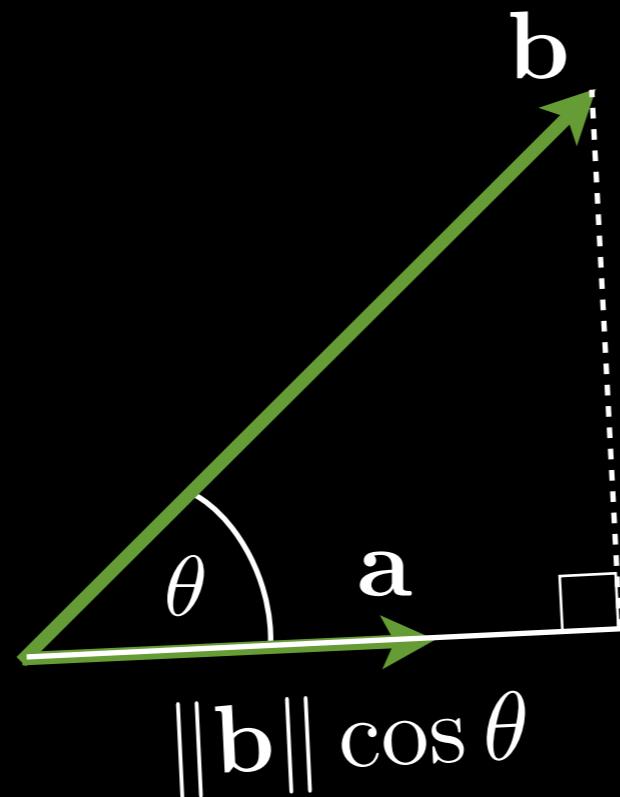
$$\mathbf{a} \cdot \mathbf{b} = \|\mathbf{a}\| \|\mathbf{b}\| \cos \theta$$

$$= \|\mathbf{b}\| \cos \theta$$

assume $\|\mathbf{a}\| = 1$

Length of projection of b onto a

Vector Projection



$$\mathbf{a} \cdot \mathbf{b} = \|\mathbf{a}\| \|\mathbf{b}\| \cos \theta$$

$$= \|\mathbf{b}\| \cos \theta$$

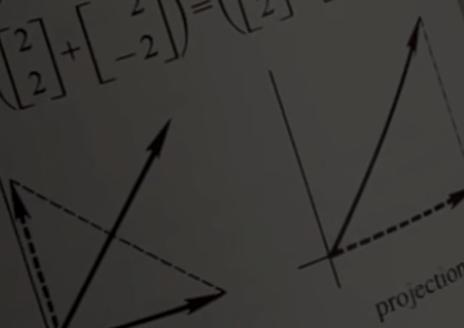
assume $\|\mathbf{a}\| = 1$

Length of projection of \mathbf{b} onto \mathbf{a}

Now how a matrix moves subspaces around. The nullspace goes or, when we multiply by A . All vectors go into the column space, cases a combination of the columns. You will soon see something A takes its row space into its column space, and on those spaces it is 100% invertible. That is the real action of a matrix. It is partly spaces and left nullspaces, which lie at right angles and go their own zero—but when A is square and invertible those are insignificant. is what happens inside the space—which means inside n -dimensional forming that vector into a user look. When A multiplies x , we can think of it x is an n -dimensional vector. This happens at every point x in the n -dimensional space \mathbf{R}^n . The n -dimensional space \mathbf{R}^n is formed, or “mapped into” by the matrix A . We give four examples.

1. A multiple of the identity matrix, $A = cI$, vector by the same factor c . The whole space expands or contracts (or somehow goes through the origin and out the opposite side, when c is negative).
2. A rotation matrix turns the whole space around the origin. This example turns all vectors through 90° , transforming $(1, 0)$ on the x -axis to $(0, 1)$, and sending $(0, 1)$ on the y -axis to $(-1, 0)$.
3. A reflection matrix transforms every vector into its image on the opposite side of a mirror. In this example the mirror is the 45° line $y = x$, and a point like $(2, 2)$ is unchanged. A point like $(2, -2)$ is reversed to $(-2, 2)$. On a combination like $(2, 2) + (2, -2) = (4, 0)$, the matrix leaves one part and reverses the other part. The result is to exchange y and x , and produce $(0, 4)$:

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 4 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 4 \end{bmatrix}, \text{ or } A \left(\begin{bmatrix} 2 \\ 2 \end{bmatrix} + \begin{bmatrix} 2 \\ -2 \end{bmatrix} \right) = \left(\begin{bmatrix} 2 \\ 2 \end{bmatrix} + \begin{bmatrix} -2 \\ 2 \end{bmatrix} \right).$$



4. A projection matrix takes the whole space onto a lower-dimensional subspace (and therefore fails to be invertible). The example transforms each vector (x, y) in the plane to the nearest point $(x, 0)$ on the horizontal axis. That axis is the column space of A , and the vertical axis (which projects onto the origin) is the nullspace.

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$$

That reflection matrix is also a permutation matrix! It is algebraically so simple, sending (x, y) to (y, x) , that the geometric picture was concealed. The fourth example is simple in both respects:

4. A projection matrix takes the whole space onto a lower-dimensional subspace (and therefore fails to be invertible). The example transforms each vector (x, y) in the plane to the nearest point $(x, 0)$ on the horizontal axis. That axis is the column space of A , and the vertical axis (which projects onto the origin) is the nullspace.

Those examples could easily be lifted into three dimensions. There are matrices to stretch the earth or spin it or reflect it across the plane of the equator (north pole transforming to south pole). There is a matrix that projects everything onto that plane (both poles to the center). Other examples are certainly possible and necessary. But it is also important to recognize that matrices cannot do everything, and some transformations are not possible with matrices:

It is impossible to move the origin, since $A0 = 0$ for every matrix. If a vector x goes to x' , then $2x$ must go to $2x'$. In general cx must go to $(cx)'$. If $x = Ax$, then $x + y$ must go to $x' + y'$, then their sum $x + y$ must go to $x' + y'$. Matrix multiplication imposes those rules on the transformation of the space. The first two rules are easy, and the second one contains the third: $c(Ax) = A(cx) = A(x) = 0$. We saw rule (iii) in action when the vector $(4, 0)$ was reflected across the y -axis. The same could be done for projections: split, project separately, and add. These rules apply to *any transformation that comes from a matrix*. Their importance has earned them a name: Transformations that obey rules (i)-(iii) are called *linear transformations*.

Those conditions can be combined into a single requirement:

2T For all numbers c and d and all vectors x and y , matrix multiplication satisfies the rule of linearity:

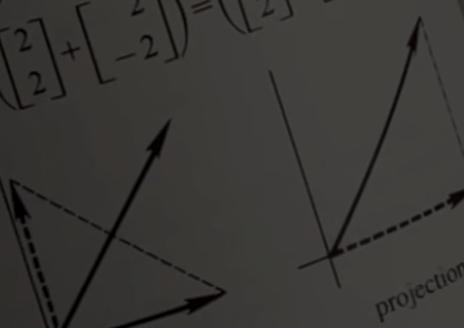
$$A(cx + dy) = c(Ax) + d(Ay). \quad (1)$$

Every transformation that meets this requirement is a linear transformation. Any matrix leads immediately to a linear transformation. The more interesting question is in the opposite direction: Does every linear transformation lead to a matrix? The object of this section is to answer that question (affirmatively, in n dimensions). This theory is the foundation of an approach to linear algebra—starting with property (1) and developing its consequences—which is much more abstract

Now how a matrix moves subspaces around. The nullspace goes or, when we multiply by A . All vectors go into the column space, cases a combination of the columns. You will soon see something A takes its row space into its column space, and on those spaces it is 100% invertible. That is the real action of a matrix. It is partly spaces and left nullspaces, which lie at right angles and go their own zero—but when A is square and invertible those are insignificant. is what happens inside the space—which means inside n -dimensional forming that vector into a user look. When A multiplies x , we can think of it x is an n -dimensional vector. This happens at every point x in the n -dimensional space \mathbf{R}^n . The n -dimensional space \mathbf{R}^n is formed, or “mapped into” by the matrix A . We give four examples.

1. A multiple of the identity matrix, $A = cI$, vector by the same factor c . The whole space expands or contracts (or somehow goes through the origin and out the opposite side, when c is negative).
2. A rotation matrix turns the whole space around the origin. This example turns all vectors through 90° , transforming $(1, 0)$ on the x -axis to $(0, 1)$, and sending $(0, 1)$ on the y -axis to $(-1, 0)$.
3. A reflection matrix transforms every vector into its image on the opposite side of a mirror. In this example the mirror is the 45° line $y = x$, and a point like $(2, 2)$ is unchanged. A point like $(2, -2)$ is reversed to $(-2, 2)$. On a combination like $(2, 2) + (2, -2) = (4, 0)$, the matrix leaves one part and reverses the other part. The result is to exchange y and x , and produce $(0, 4)$:

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 4 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 4 \end{bmatrix}, \text{ or } A \left(\begin{bmatrix} 2 \\ 2 \end{bmatrix} + \begin{bmatrix} 2 \\ -2 \end{bmatrix} \right) = \left(\begin{bmatrix} 2 \\ 2 \end{bmatrix} + \begin{bmatrix} -2 \\ 2 \end{bmatrix} \right).$$



$$A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

$$A = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

$$A = \begin{bmatrix} c & 0 \\ 0 & c \end{bmatrix}$$

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$$

That reflection matrix is also a permutation matrix! It is algebraically so simple, sending (x, y) to (y, x) , that the geometric picture was concealed. The fourth example is simple in both respects:

4. A projection matrix takes the whole space onto a lower-dimensional subspace (and therefore fails to be invertible). The example transforms each vector (x, y) in the plane to the nearest point $(x, 0)$ on the horizontal axis. That axis is the column space of A , and the vertical axis (which projects onto the origin) is the nullspace.

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$$

Those examples could easily be lifted into three dimensions. There are matrices to stretch the earth or spin it or reflect it across the plane of the equator (north pole transforming to south pole). There is a matrix that projects everything onto that plane (both poles to the center). Other examples are certainly possible and necessary. But it is also important to recognize that matrices cannot do everything, and some transformations are not possible with matrices:

It is impossible to move the origin, since $A0 = 0$ for every matrix. If a vector x goes to x' , then $2x$ must go to $2x'$. In general cx must go to $(cx)'$. If $x = Ax$, then $x + y$ must go to $x' + y'$, then their sum $x + y$ must go to $x' + y'$.

Matrix multiplication imposes those rules on the transformation of the space. The first two rules are easy, and the second one contains the rule of linearity: if $x = Ax$, then $x + y$ must go to $x' + y'$, then their sum $x + y$ must go to $x' + y'$.

We saw rule (iii) in action when the vector $(4, 0)$ was reflected across the y -axis. The same could be done for projections: split, project separately, and add. These rules apply to *any transformation that comes from a matrix*. Their importance has earned them a name: Transformations that obey rules (i)-(iii) are called *linear transformations*.

Those conditions can be combined into a single requirement:

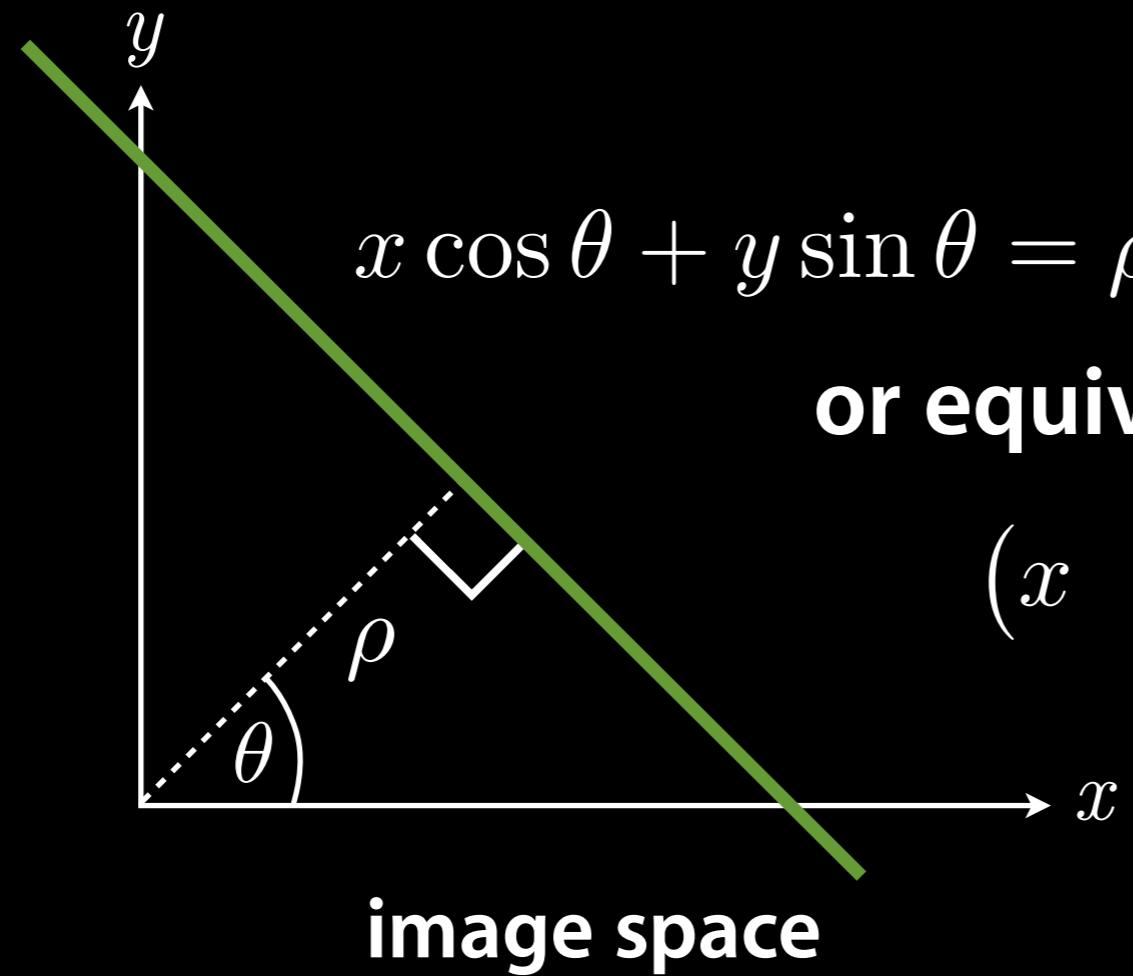
2T For all numbers c and d and all vectors x and y , matrix multiplication satisfies the rule of linearity:

$$A(cx + dy) = c(Ax) + d(Ay). \quad (1)$$

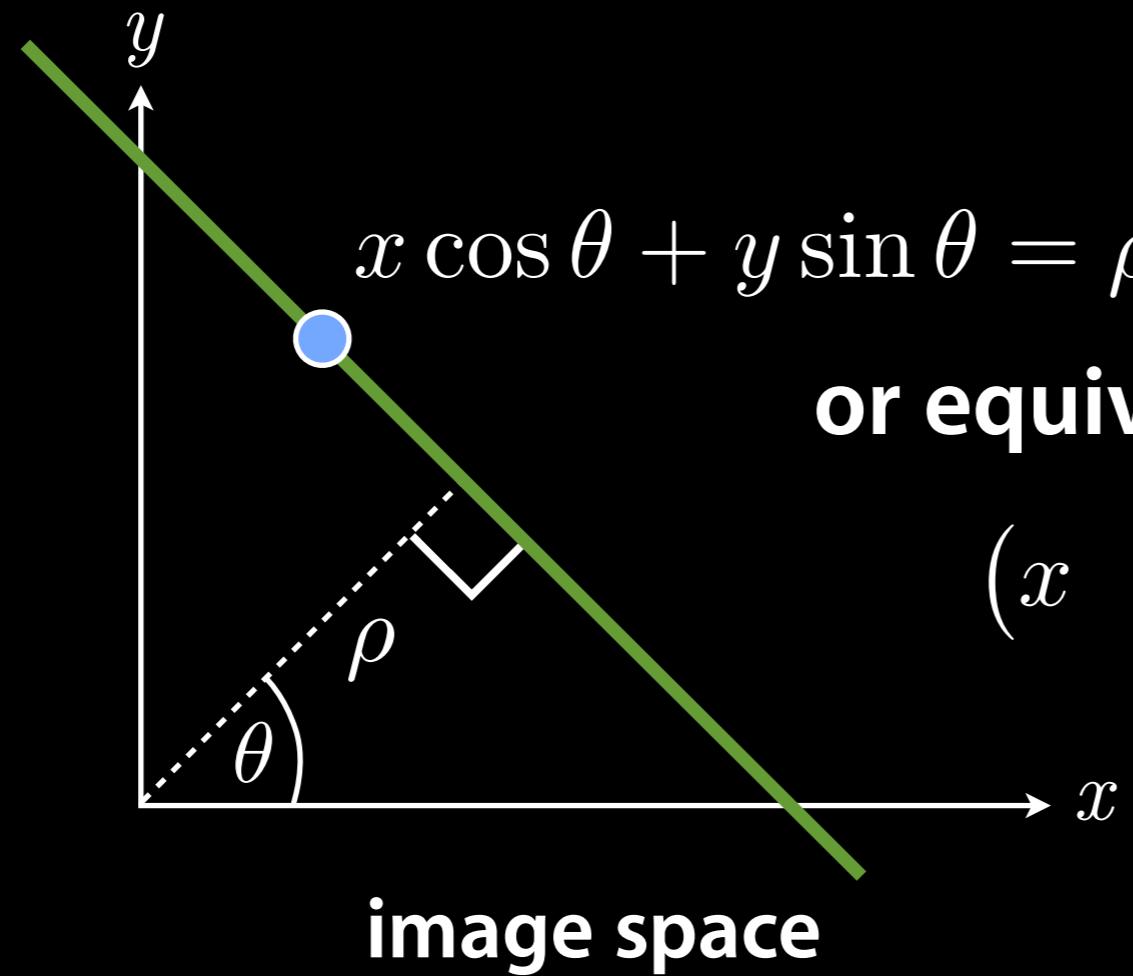
Every transformation that meets this requirement is a linear transformation.

Any matrix leads immediately to a linear transformation. The more interesting question is in the opposite direction: Does every linear transformation lead to a matrix? The object of this section is to answer that question (affirmatively, in n dimensions). This theory is the foundation of an approach to linear algebra—starting with property (1) and developing its consequences—which is much more abstract

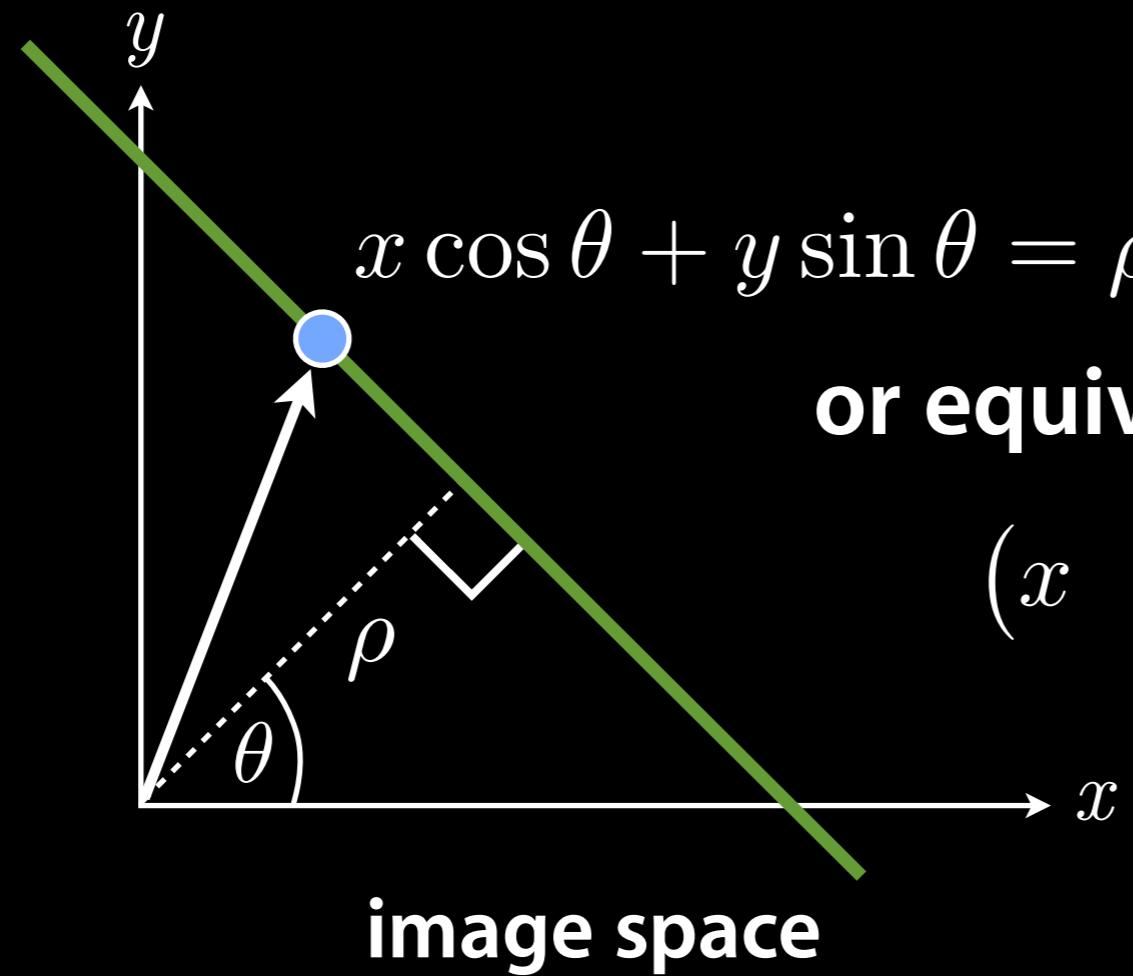
Polar Representation



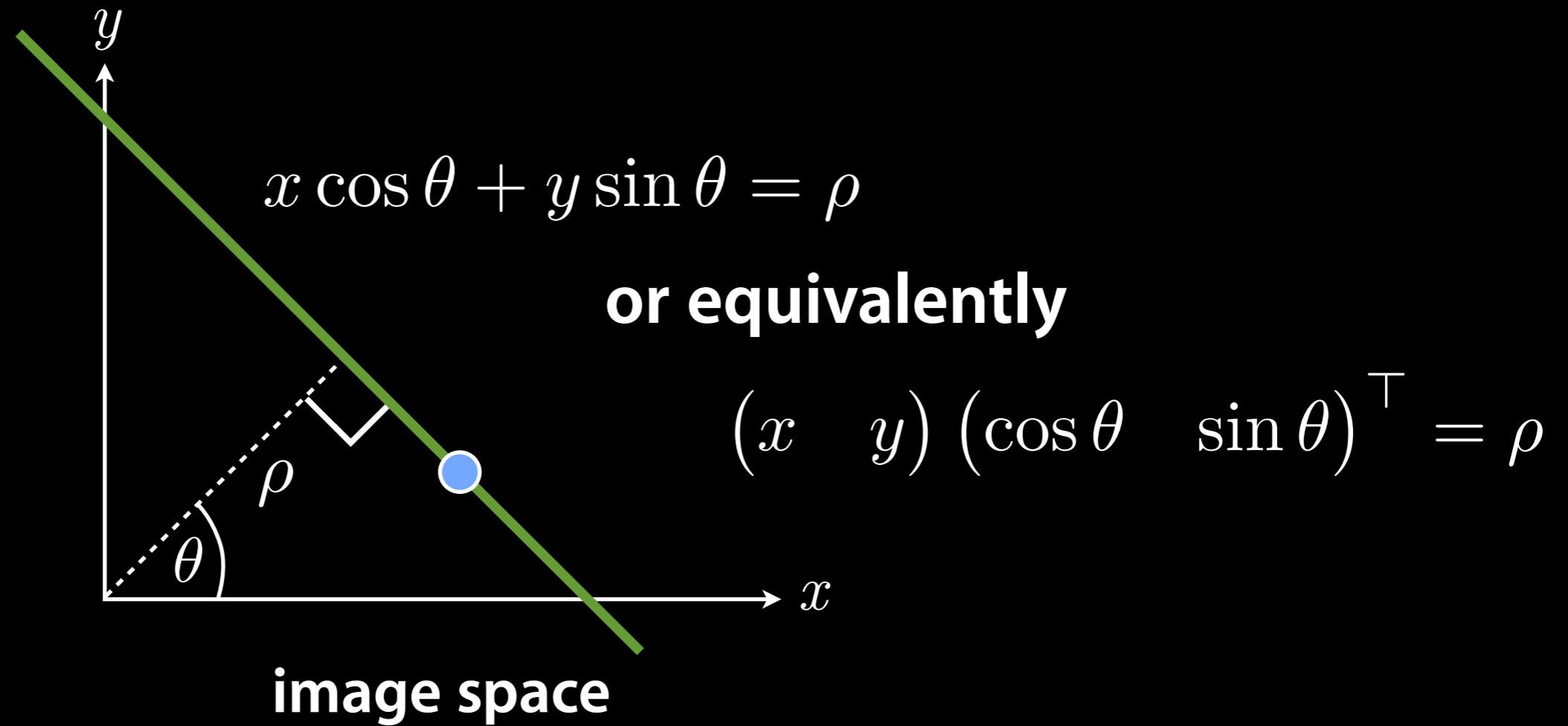
Polar Representation



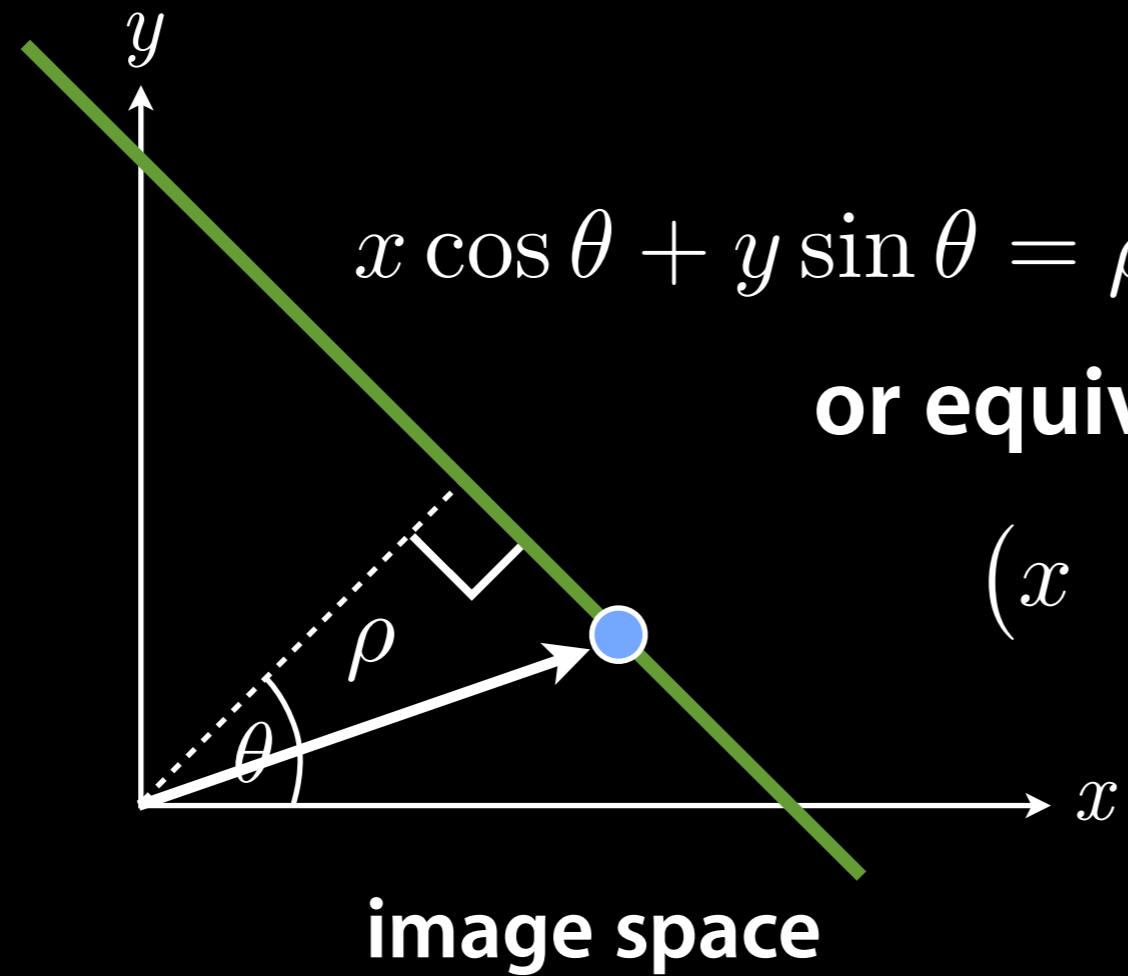
Polar Representation



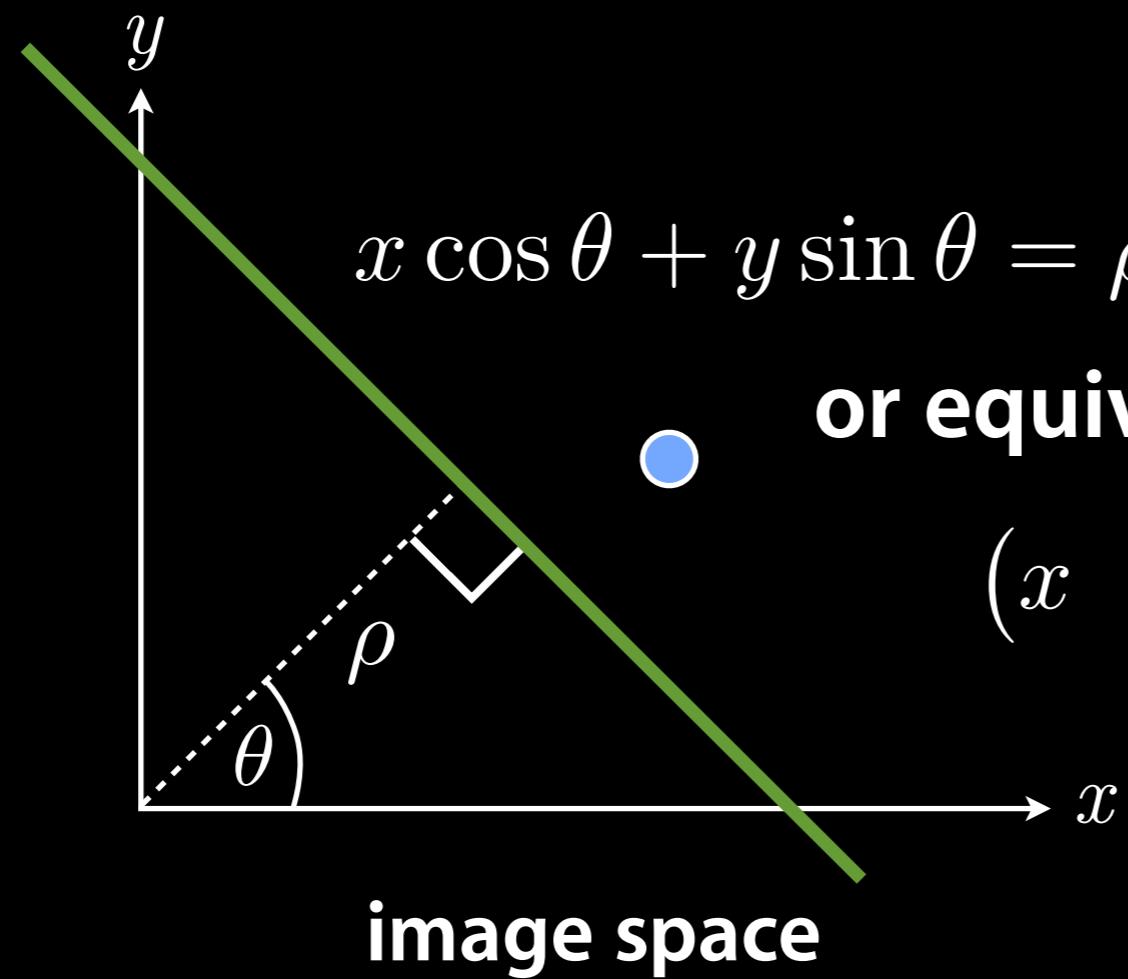
Polar Representation



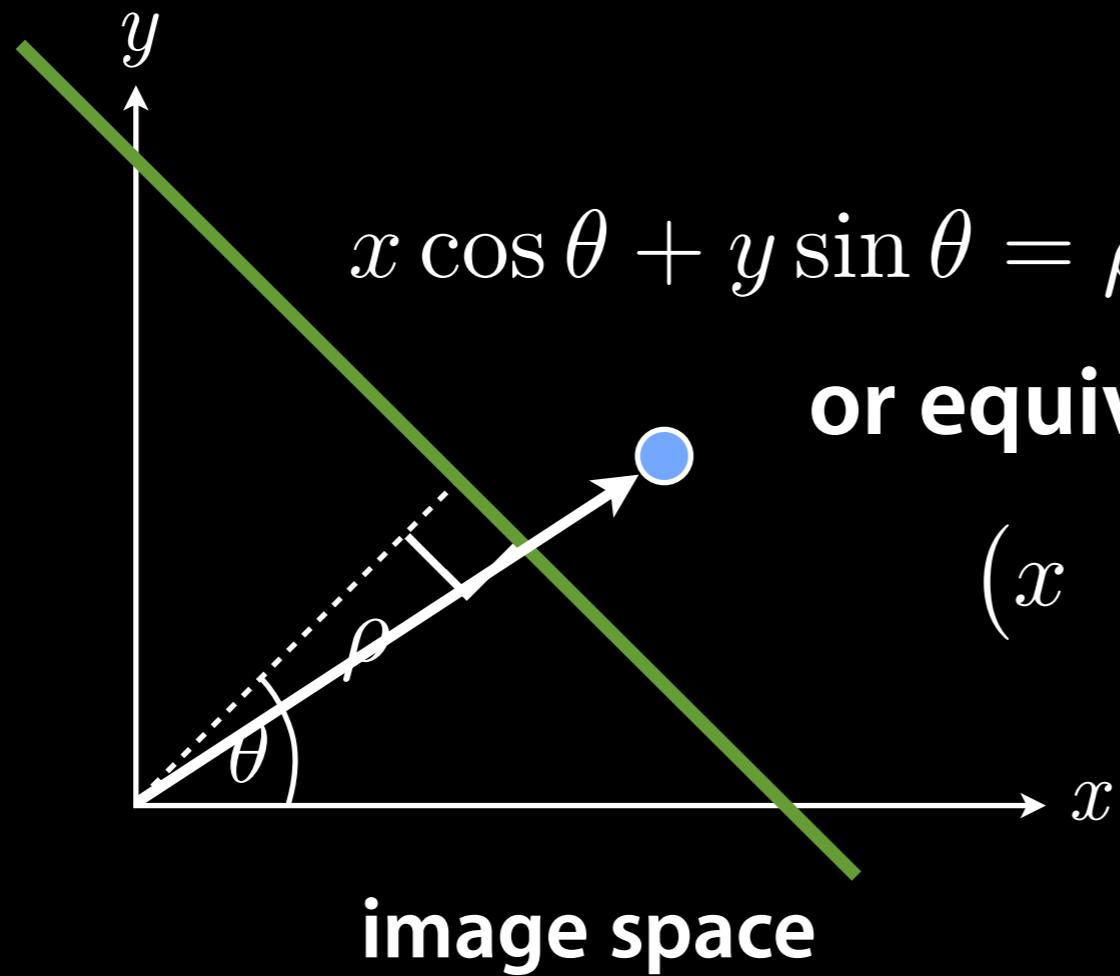
Polar Representation



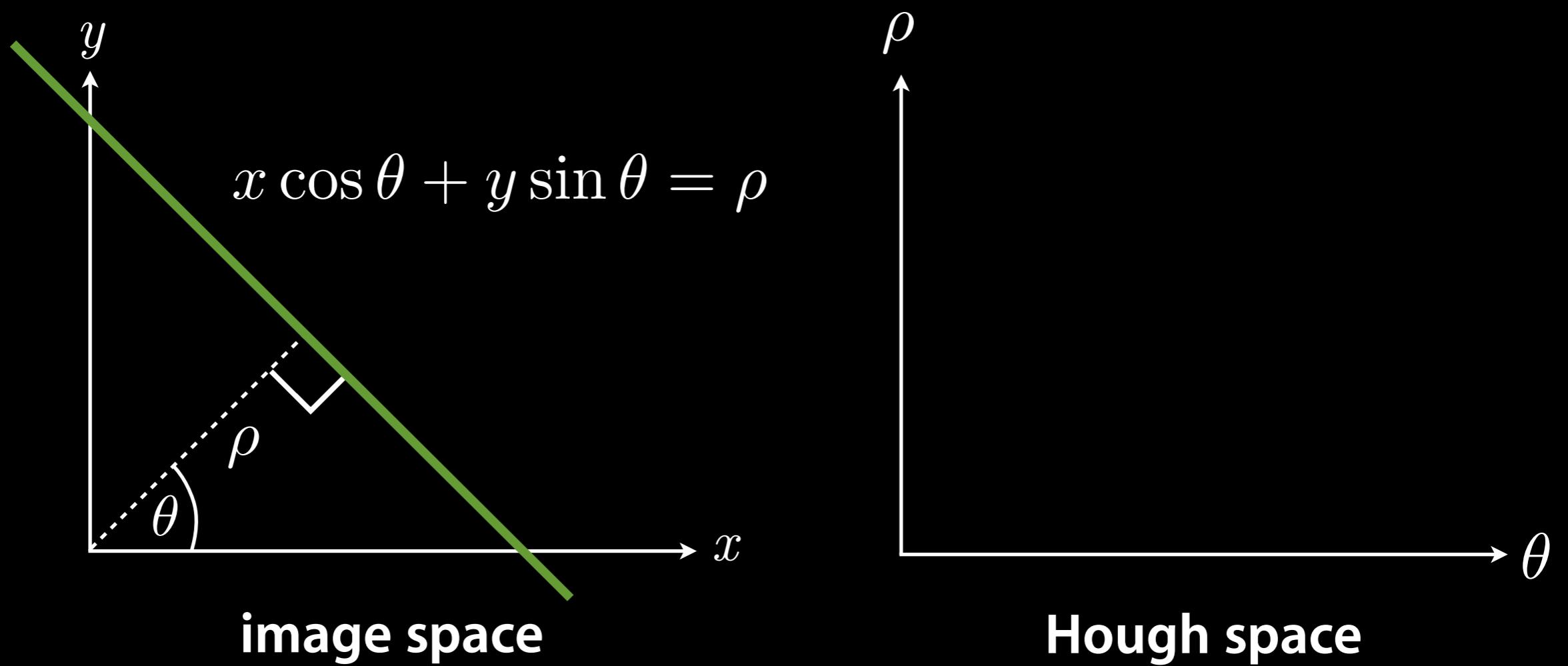
Polar Representation



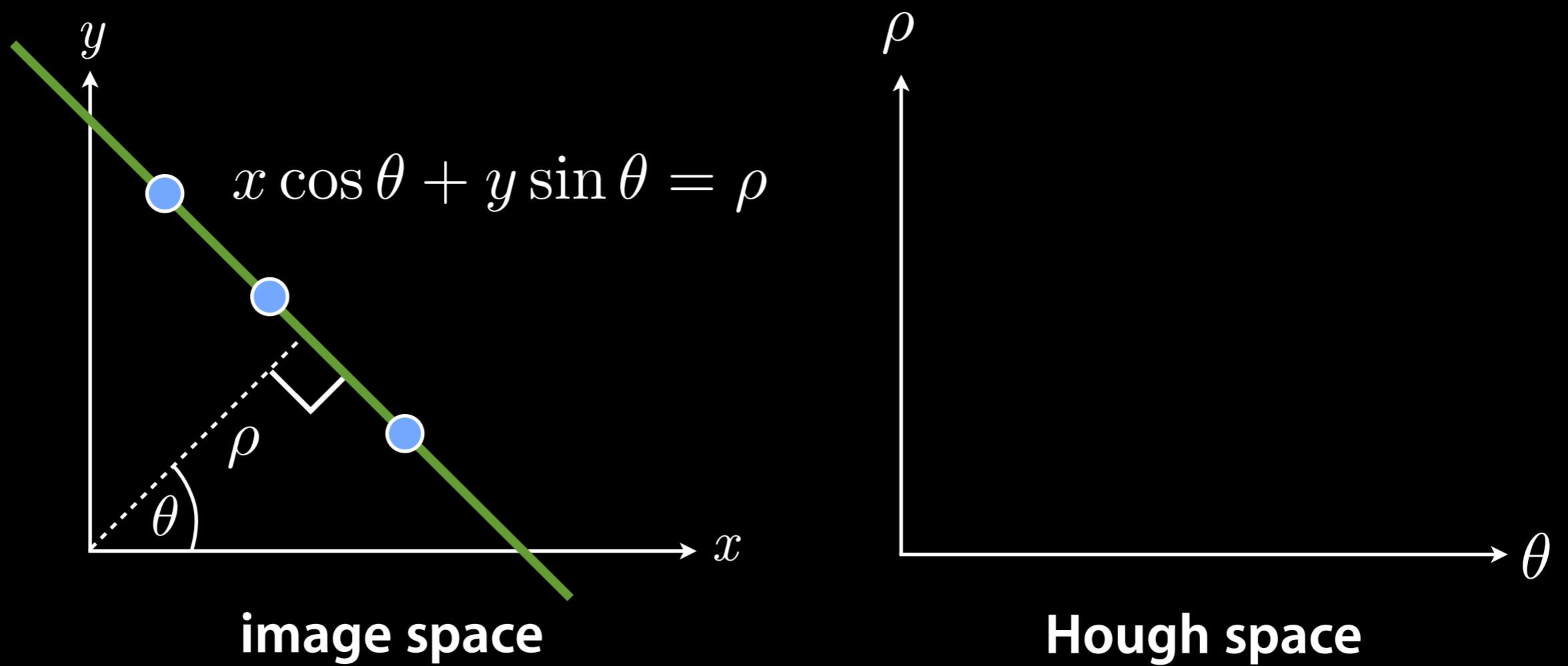
Polar Representation



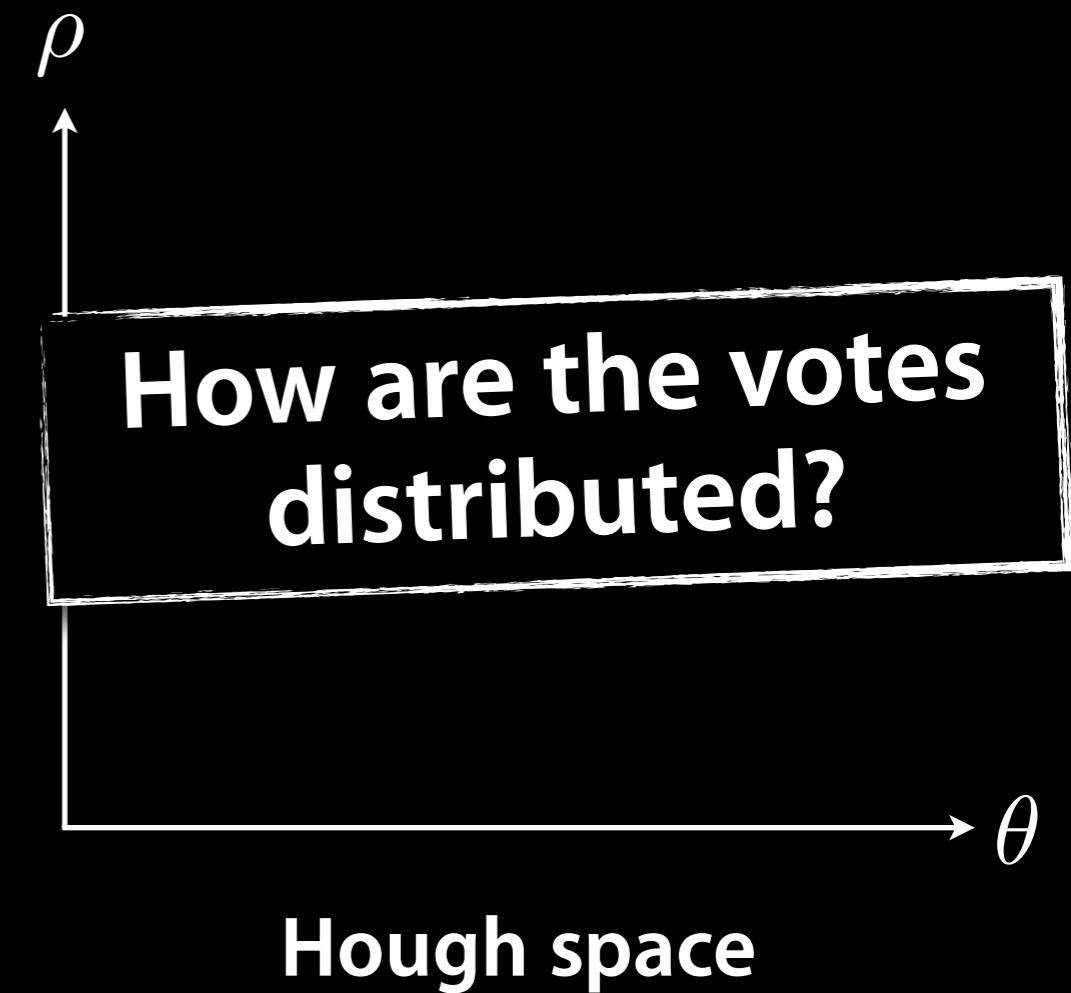
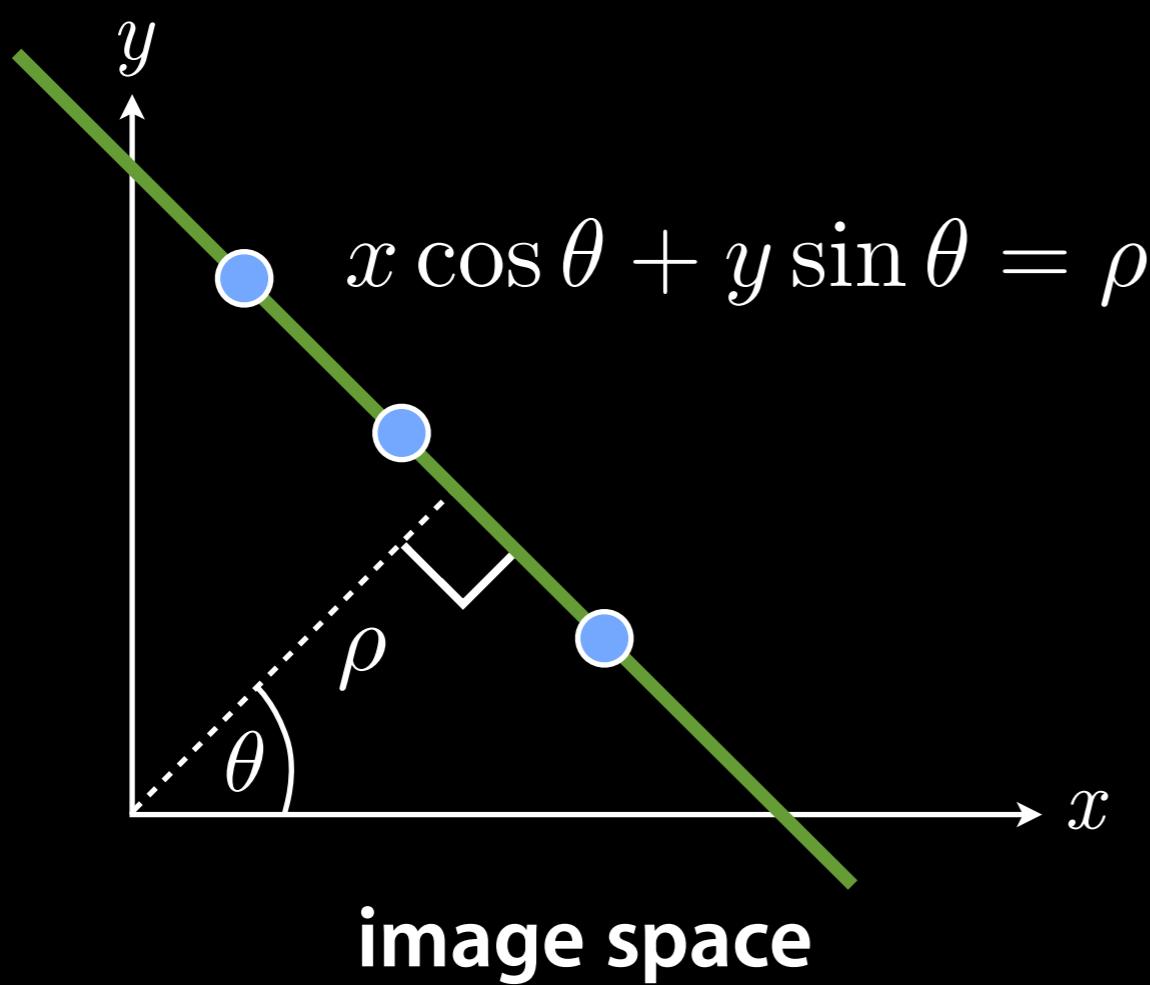
Polar Representation



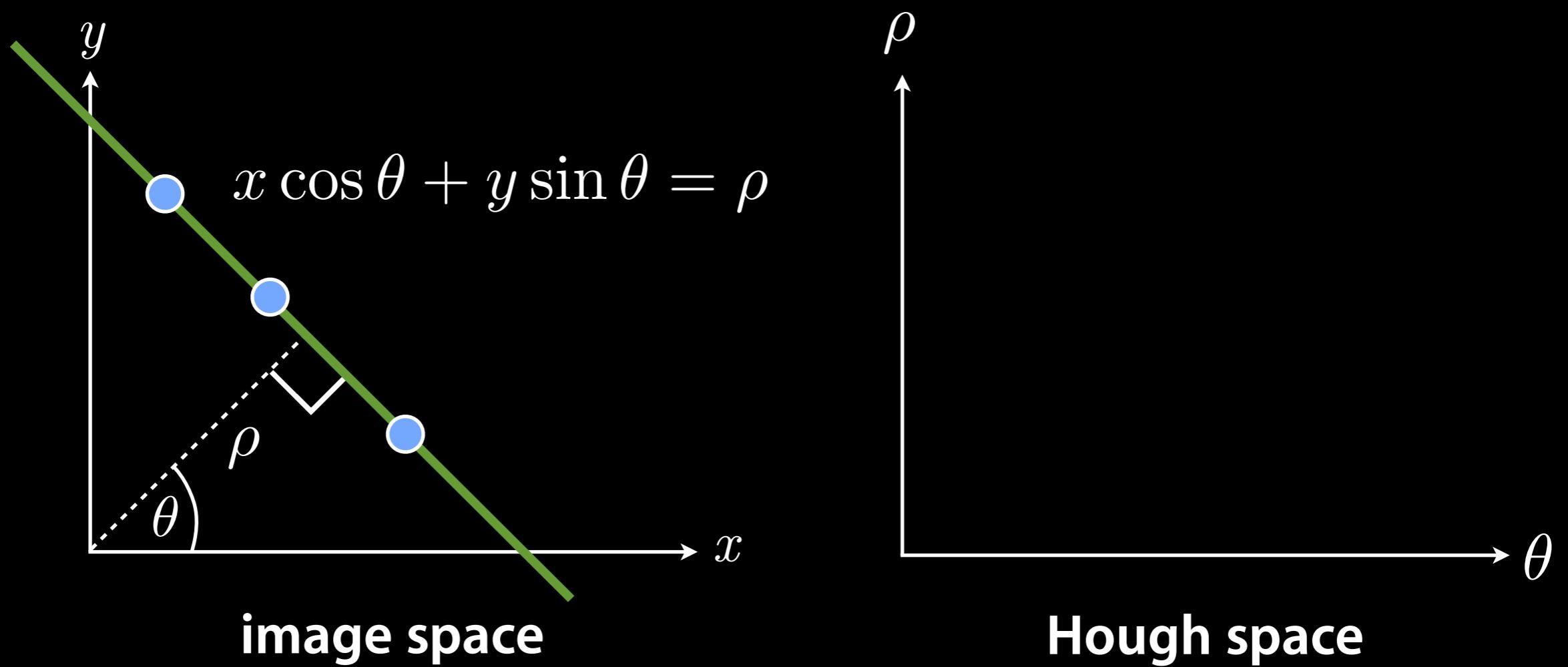
Polar Representation



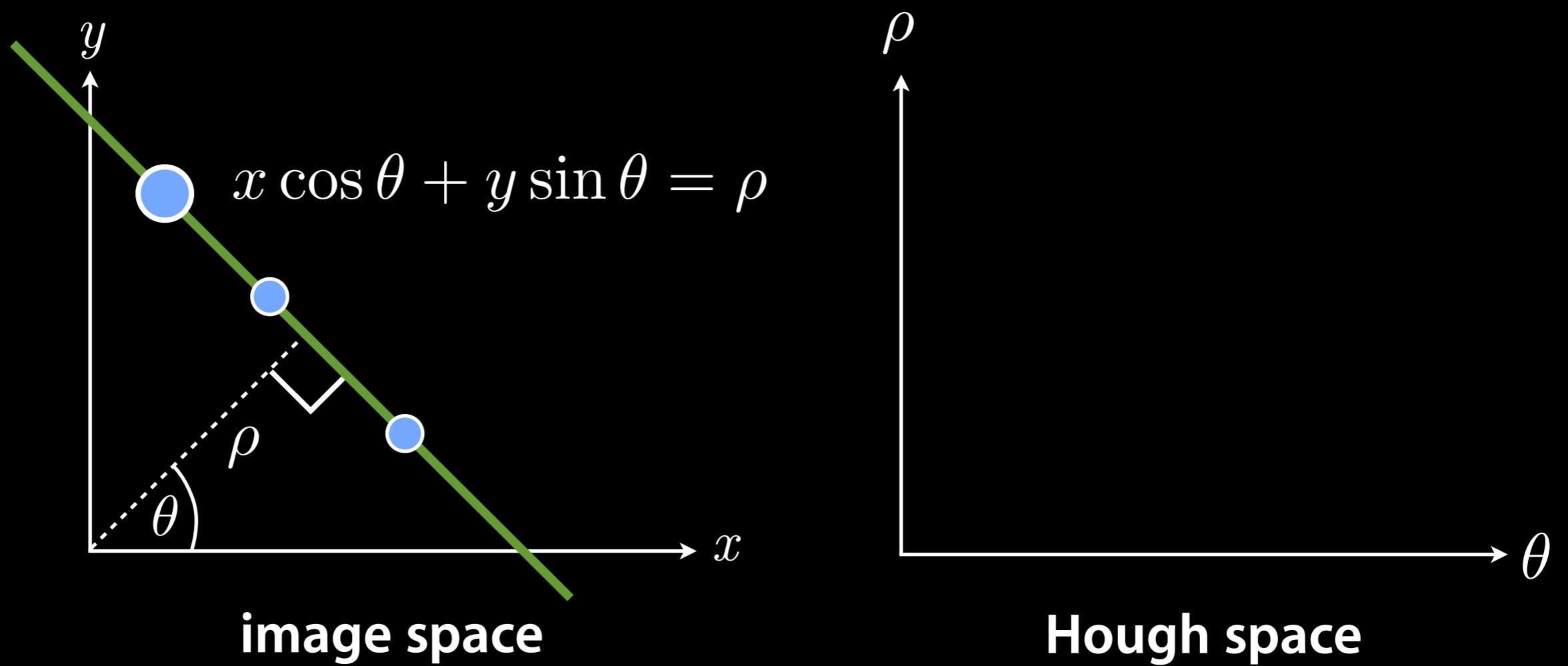
Polar Representation



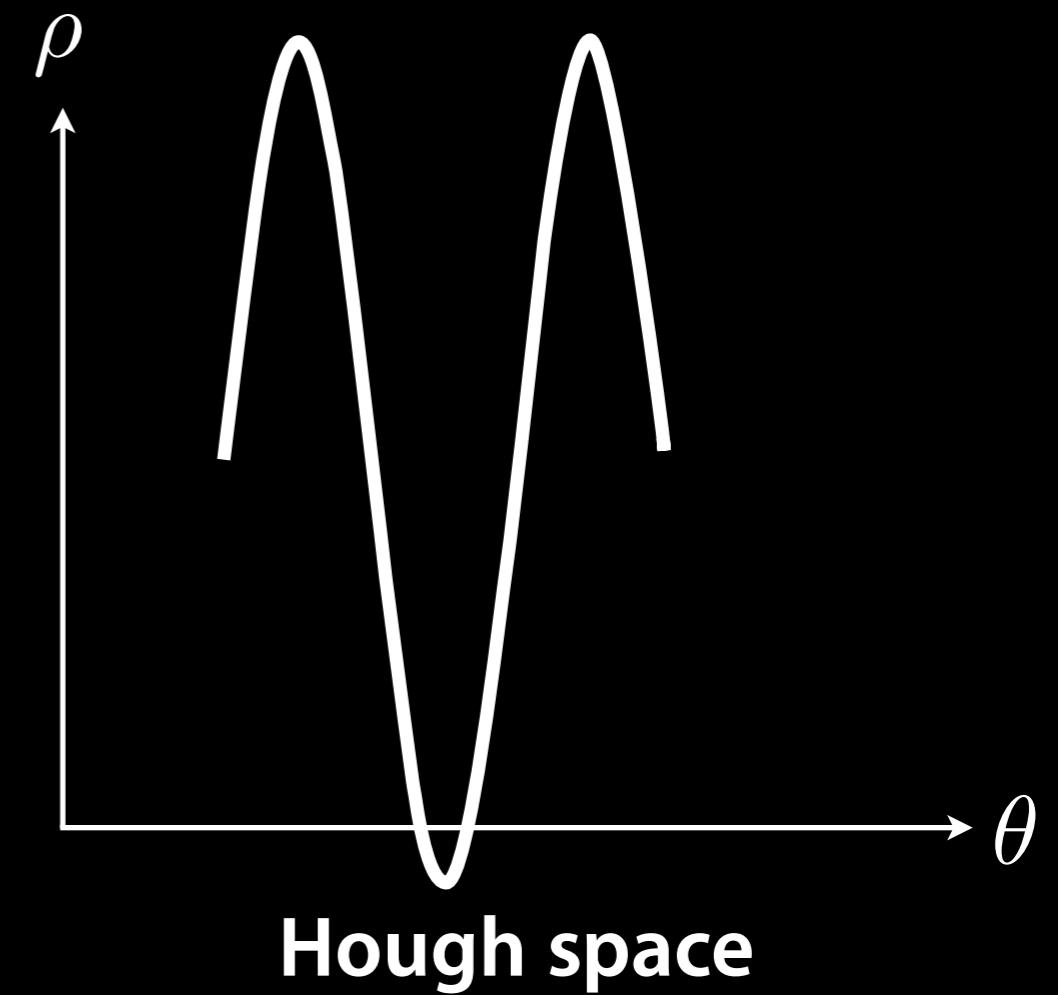
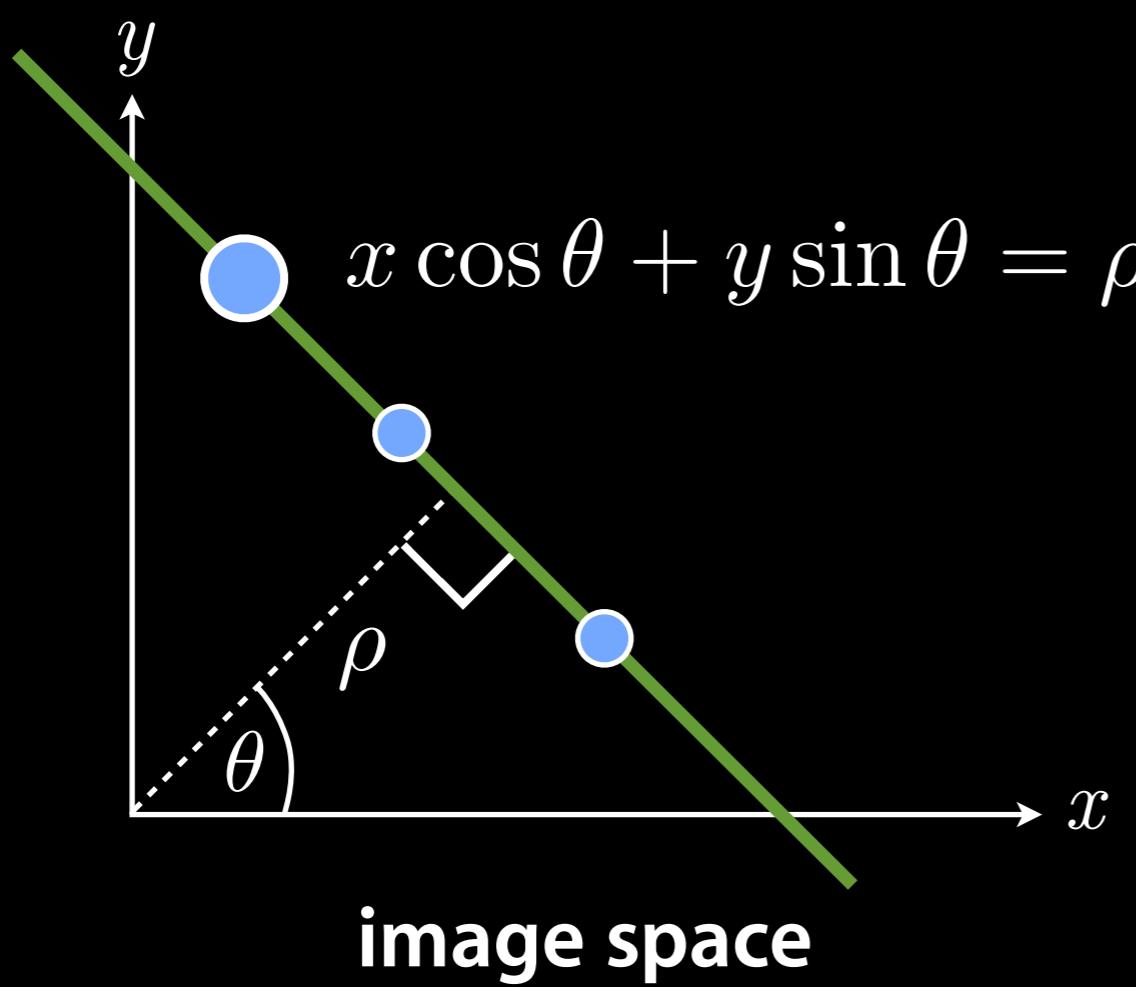
Polar Representation



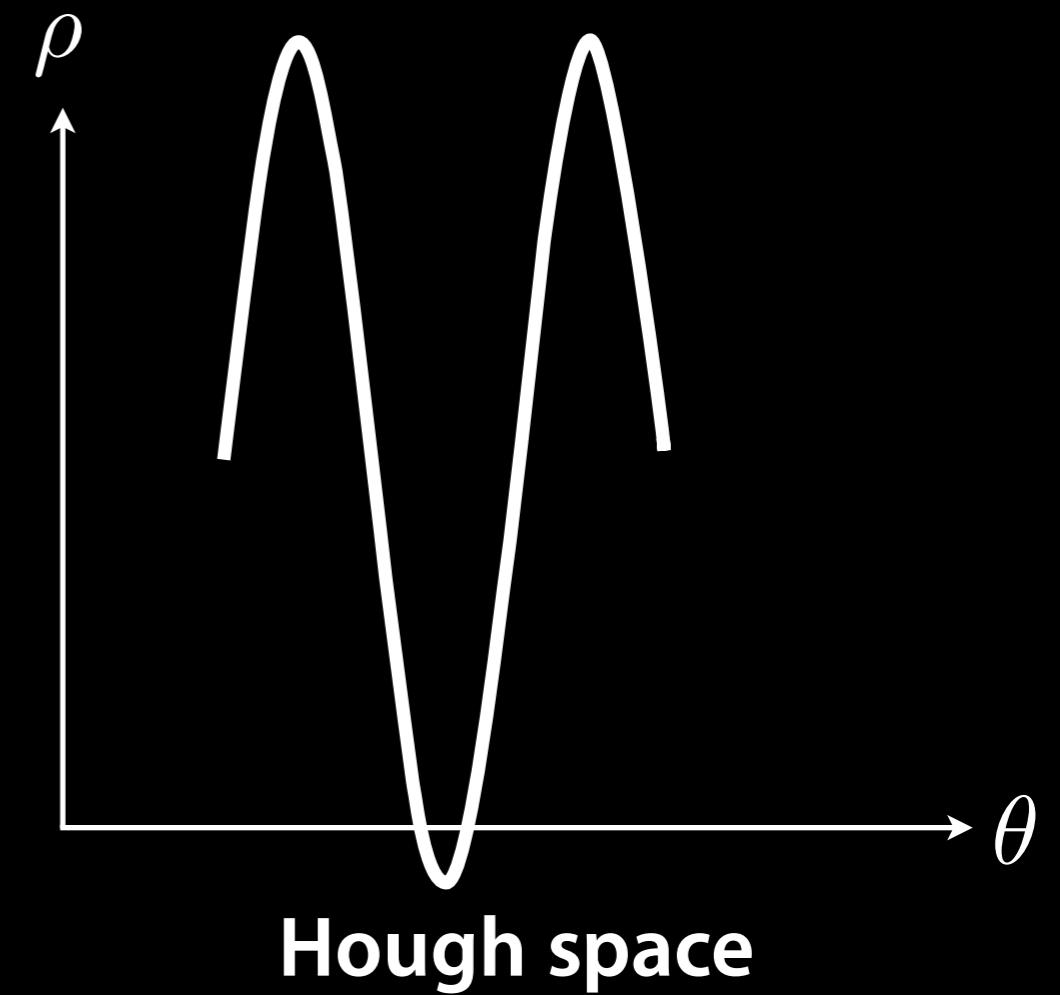
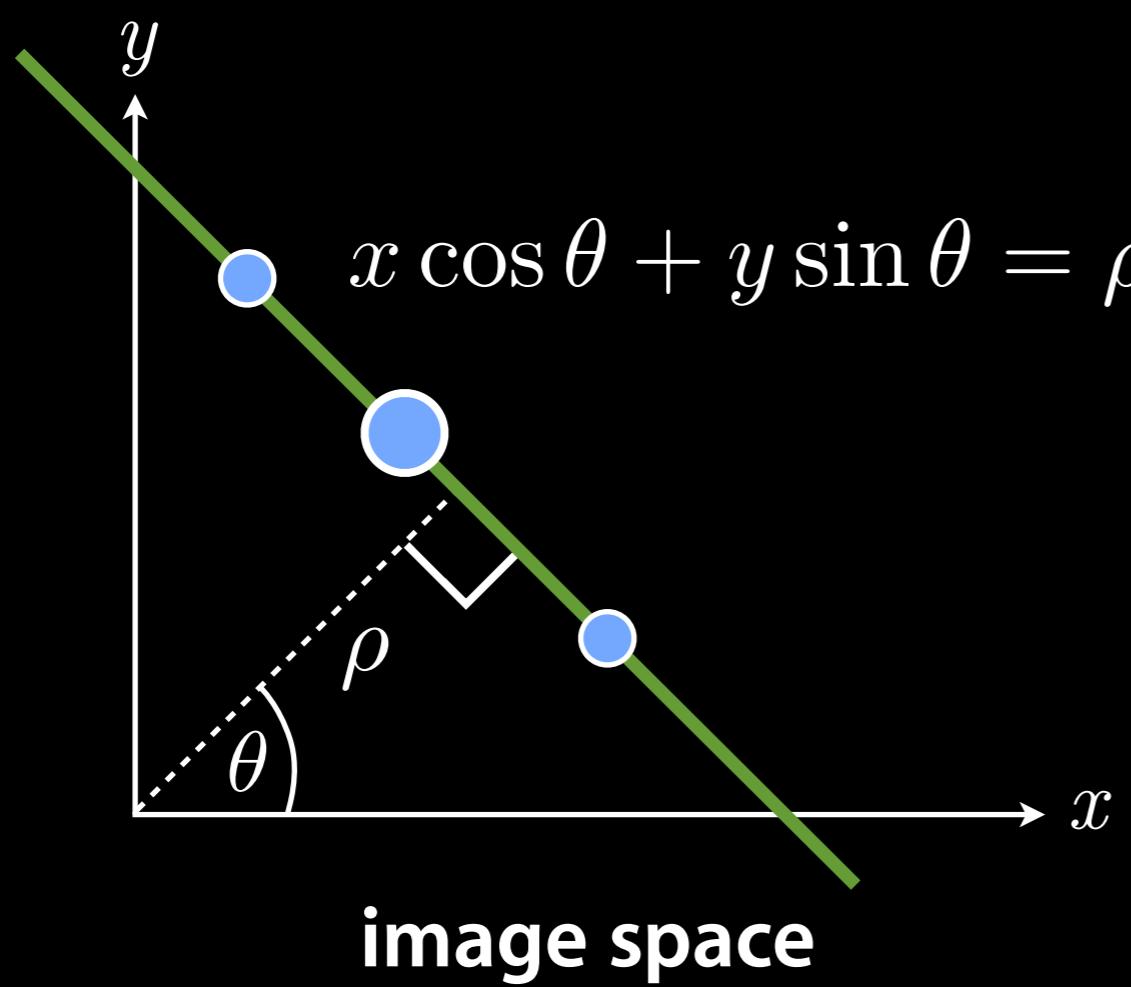
Polar Representation



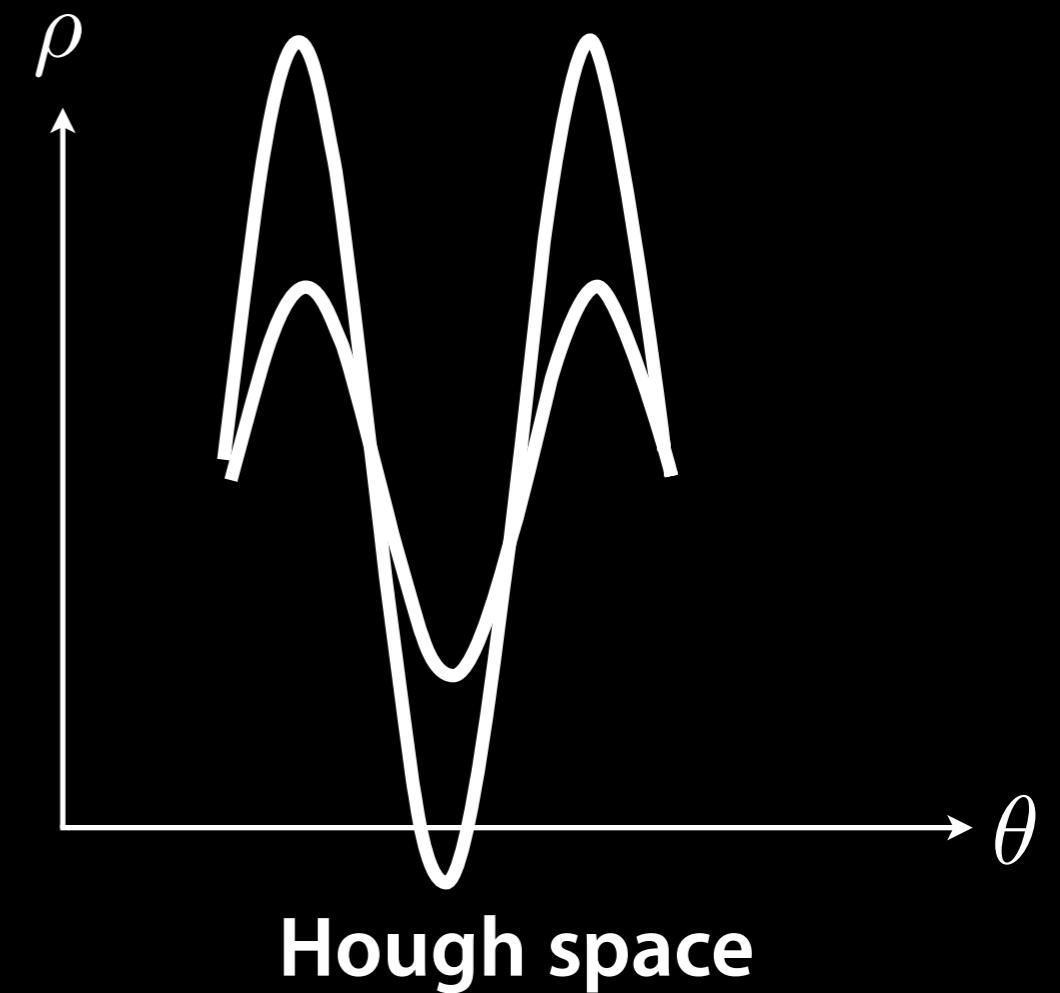
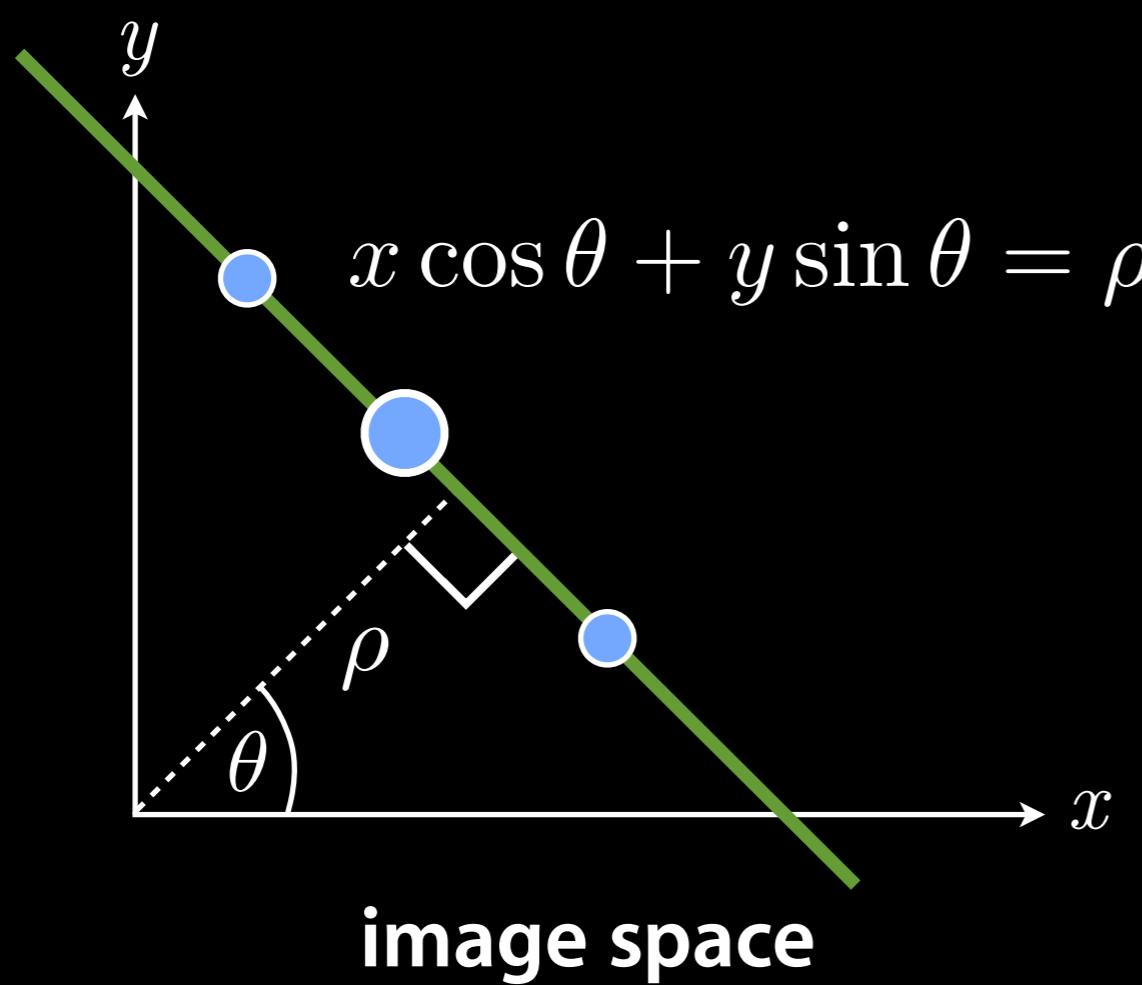
Polar Representation



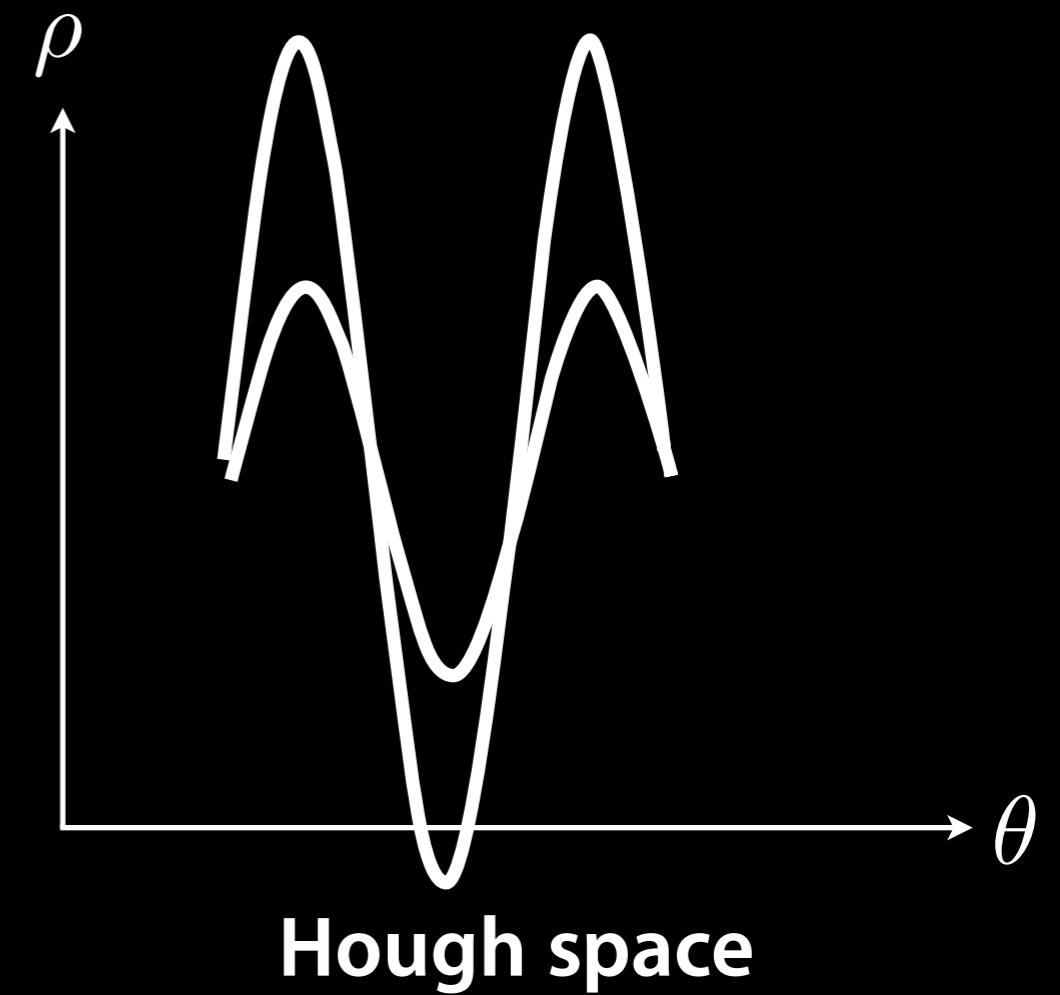
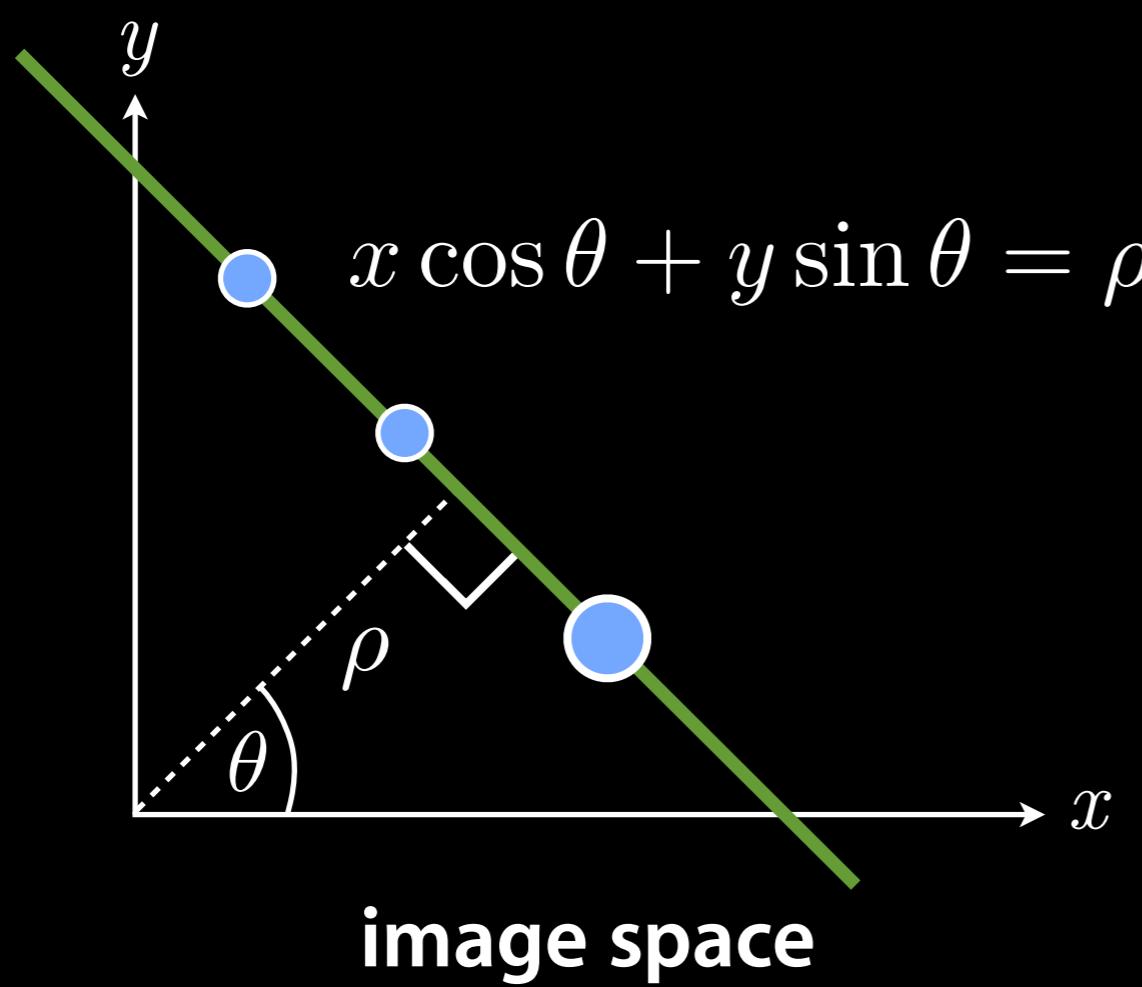
Polar Representation



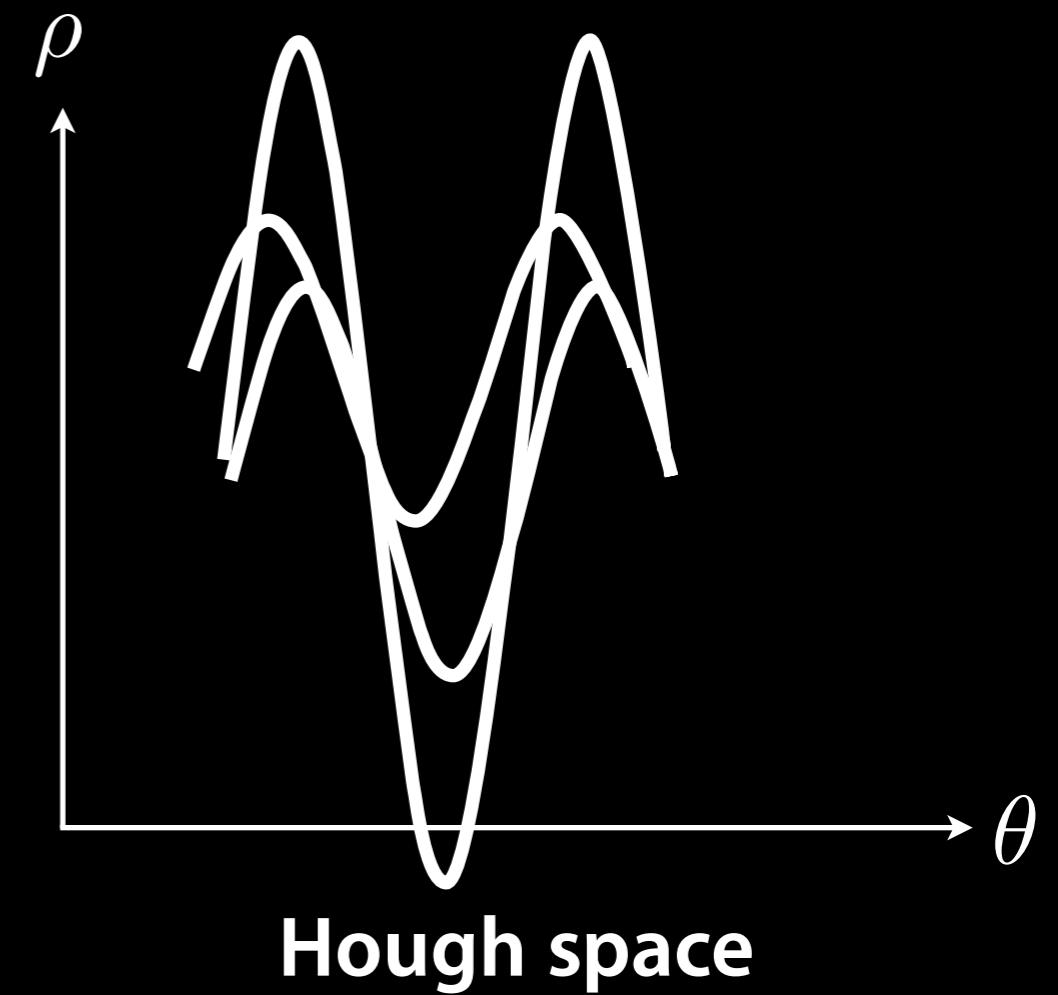
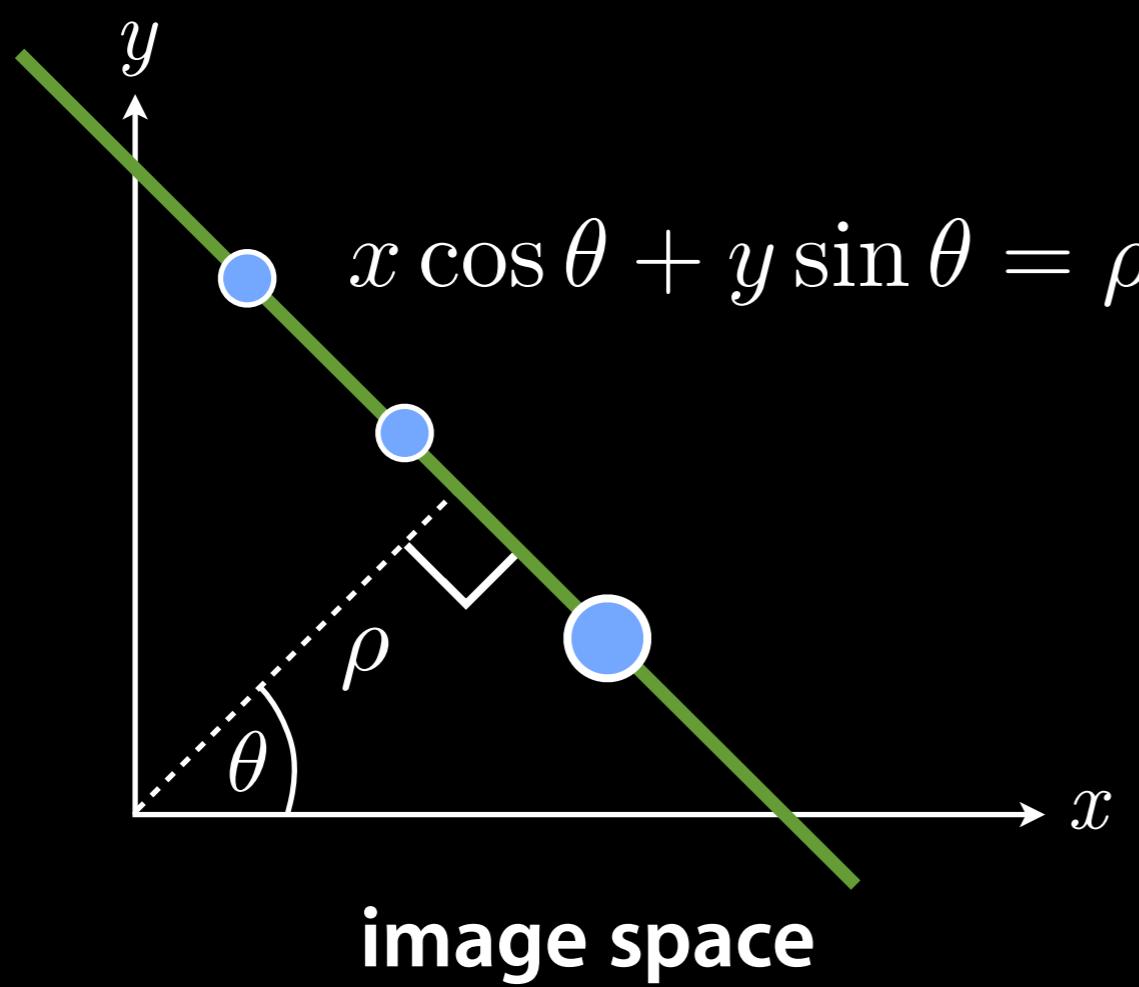
Polar Representation



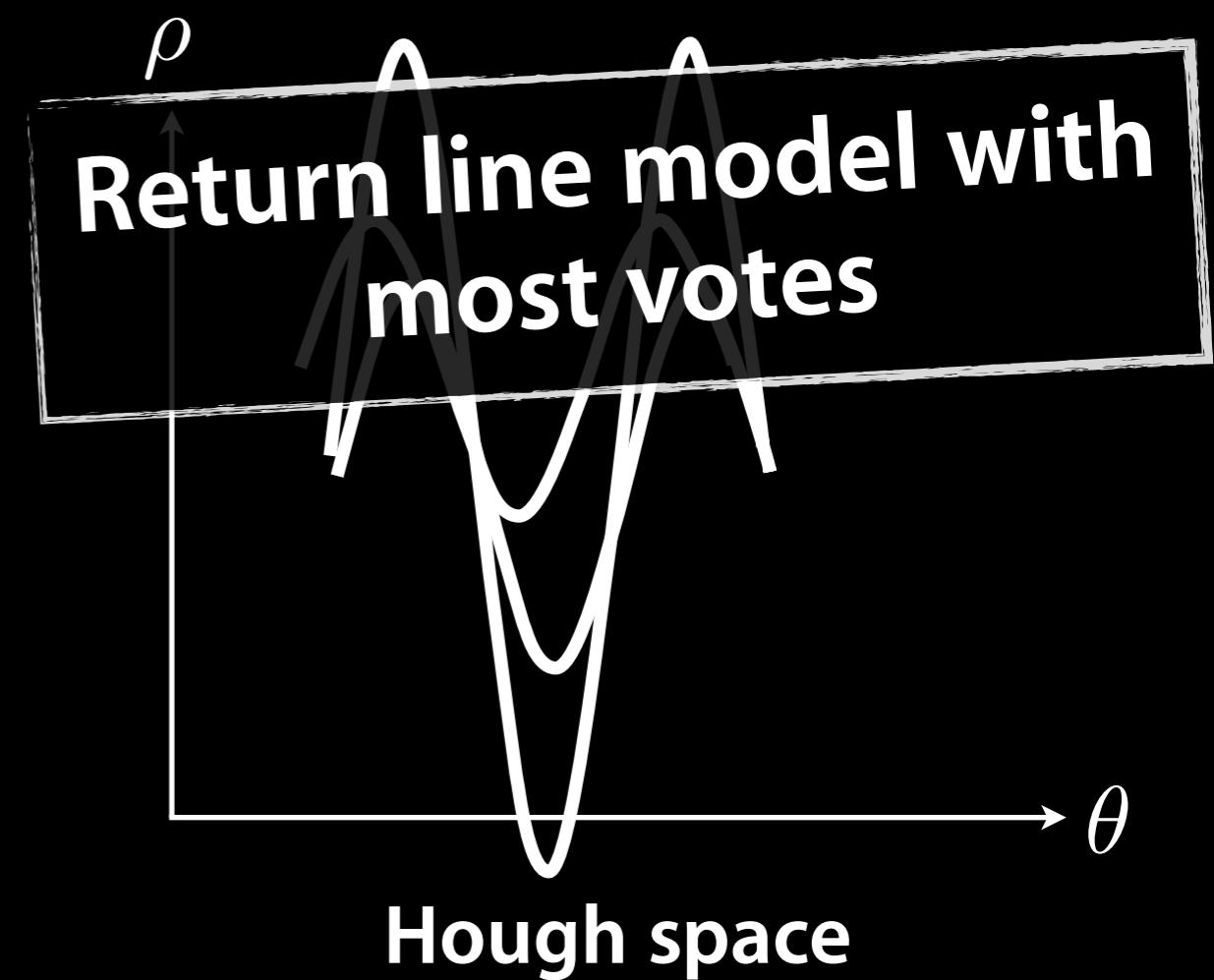
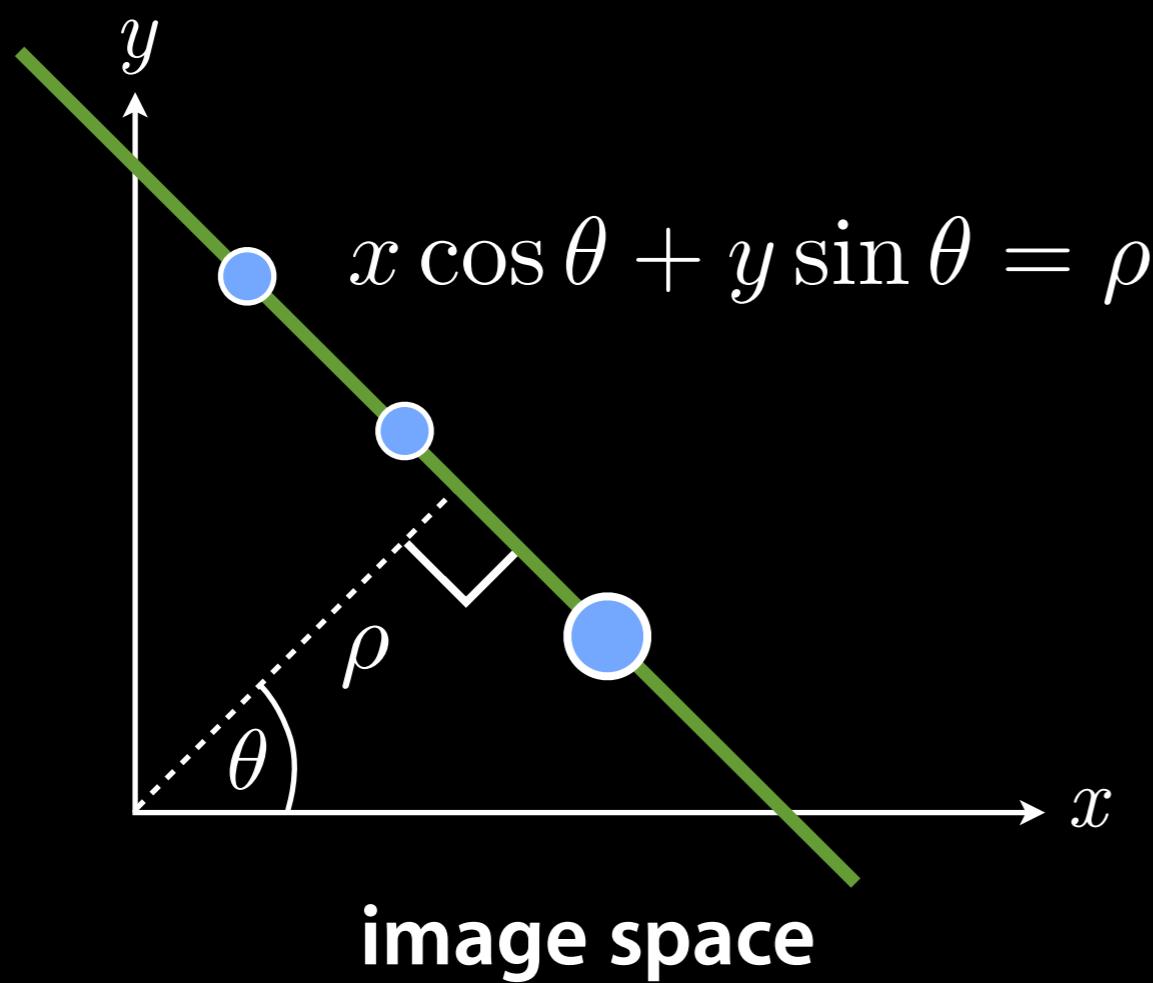
Polar Representation



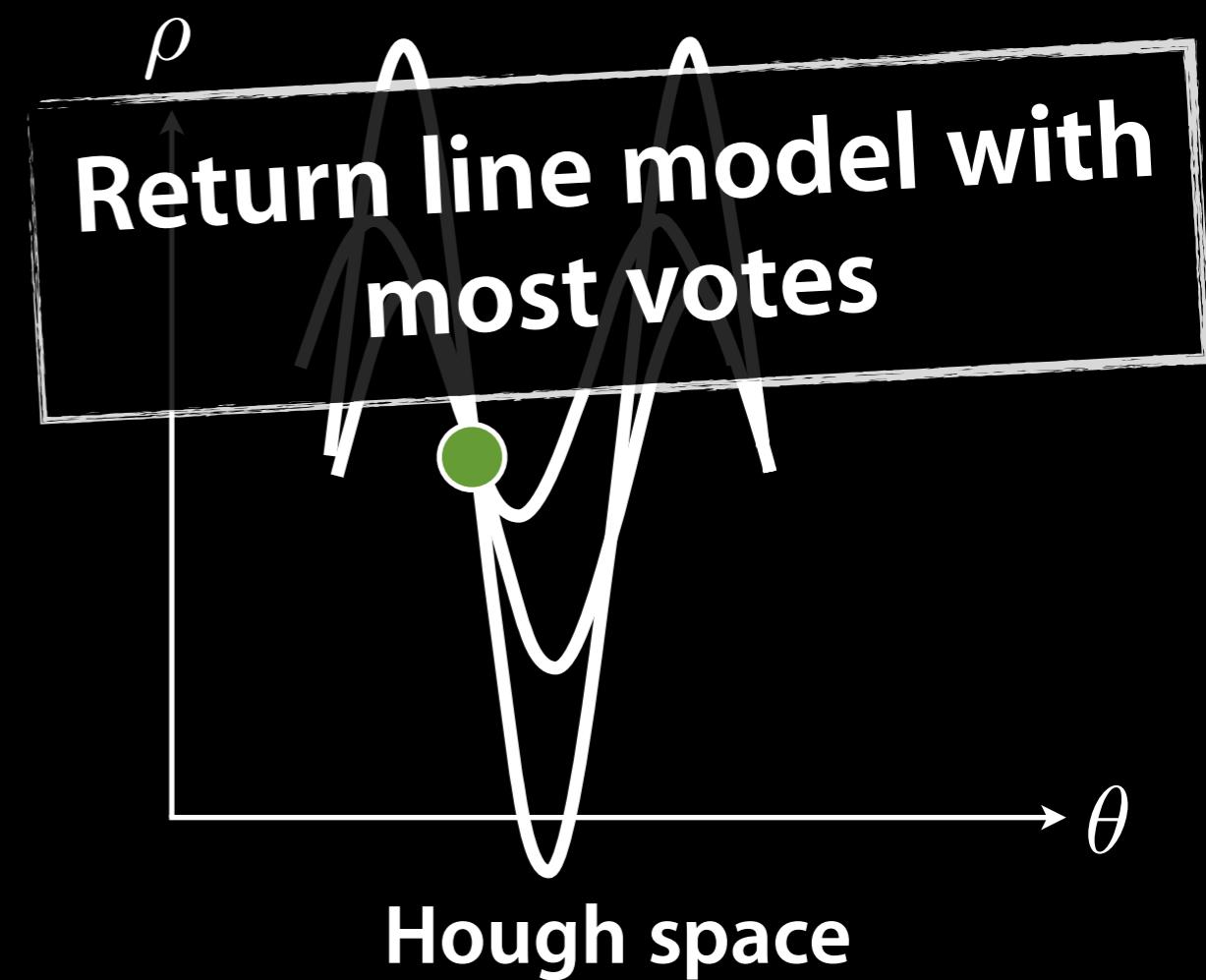
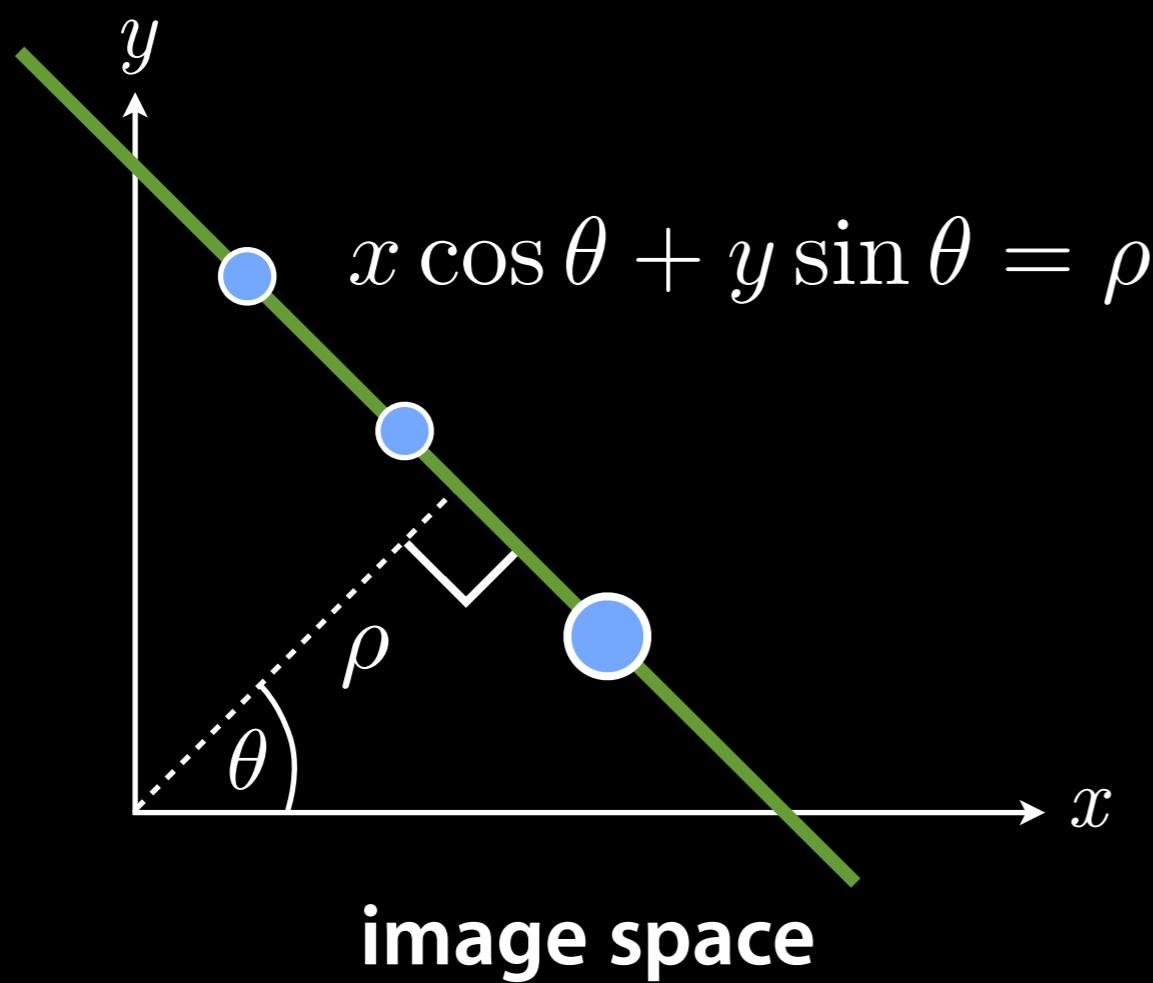
Polar Representation



Polar Representation



Polar Representation



Hough Lines

```
1: Initialize  $H[\rho, \theta] = 0$ 
2: foreach edgel  $e \in I[x, y]$  do
3:   foreach  $\theta = \theta_{\min}$  to  $\theta_{\max}$  do
4:      $\rho = x \cos \theta + y \sin \theta$ 
5:      $H[\rho, \theta] += 1$ 
6:   end
7: end
8:  $\rho^*, \theta^* = \underset{\rho, \theta}{\operatorname{argmax}} H[\rho, \theta]$ 
```

Hough Lines

```
1: Initialize  $H[\rho, \theta] = 0$ 
2: foreach edgel  $e \in I[x, y]$  do
3:   foreach  $\theta = \theta_{\min}$  to  $\theta_{\max}$  do
4:      $\rho = x \cos \theta + y \sin \theta$ 
5:      $H[\rho, \theta] += 1$ 
6:   end
7: end
8:  $\rho^*, \theta^* = \operatorname{argmax}_{\rho, \theta} H[\rho, \theta]$ 
```

Time complexity in terms of number of votes per point?

Hough Lines

```
1: Initialize  $H[\rho, \theta] = 0$ 
2: foreach edgel  $e \in I[x, y]$  do
3:   foreach  $\theta = \theta_{\min}$  to  $\theta_{\max}$  do
4:      $\rho = x \cos \theta + y \sin \theta$ 
5:      $H[\rho, \theta] += 1$ 
6:   end
7: end
8:  $\rho^*, \theta^* = \underset{\rho, \theta}{\operatorname{argmax}} H[\rho, \theta]$ 
```

Hough Lines

```
1: Initialize  $H[\rho, \theta] = 0$ 
2: foreach edgel  $e \in I[x, y]$  do
3:   | foreach  $\theta = \theta_{\min}$  to  $\theta_{\max}$  do
4:     |   |  $\rho = x \cos \theta + y \sin \theta$ 
5:     |   |  $H[\rho, \theta] += 1$ 
6:   | end
7: end
8:  $\rho^*, \theta^* = \underset{\rho, \theta}{\operatorname{argmax}} H[\rho, \theta]$ 
```

extension
#1

Hough Lines

```
1: Initialize  $H[\rho, \theta] = 0$ 
2: foreach edgel  $e \in I[x, y]$  do
3:   |
4:      $\rho = x \cos \theta + y \sin \theta$ 
5:      $H[\rho, \theta] += 1$ 
6:   |
7: end
8:  $\rho^*, \theta^* = \underset{\rho, \theta}{\operatorname{argmax}} H[\rho, \theta]$ 
```

extension
#1

Hough Lines

```
1: Initialize  $H[\rho, \theta] = 0$ 
2: foreach edgel  $e \in I[x, y]$  do
3:    $\theta = \text{angle}(\nabla(I[x, y]))$ 
4:    $\rho = x \cos \theta + y \sin \theta$ 
5:    $H[\rho, \theta] += 1$ 
6:
7: end
8:  $\rho^*, \theta^* = \underset{\rho, \theta}{\operatorname{argmax}} H[\rho, \theta]$ 
```

extension
#1

Hough Lines

```
1: Initialize  $H[\rho, \theta] = 0$ 
2: foreach edgel  $e \in I[x, y]$  do
3:    $\theta = \text{angle}(\nabla(I[x, y]))$ 
4:    $\rho = x \cos \theta + y \sin \theta$ 
5:    $H[\rho, \theta] += 1$ 
6:
7: end
8:  $\rho^*, \theta^* = \underset{\rho, \theta}{\operatorname{argmax}} H[\rho, \theta]$ 
```

extension
#1

Hough Lines

```
1: Initialize  $H[\rho, \theta] = 0$ 
2: foreach edgel  $e \in I[x, y]$  do
3:    $\theta = \text{angle}(\nabla(I[x, y]))$ 
4:    $\rho = x \cos \theta + y \sin \theta$ 
5:    $H[\rho, \theta] += 1$ 
6:
7: end
8:  $\rho^*, \theta^* = \underset{\rho, \theta}{\operatorname{argmax}} H[\rho, \theta]$ 
```

extension

#2

Hough Lines

```
1: Initialize  $H[\rho, \theta] = 0$ 
2: foreach edgel  $e \in I[x, y]$  do
3:    $\theta = \text{angle}(\nabla(I[x, y]))$ 
4:    $\rho = x \cos \theta + y \sin \theta$ 
5:    $H[\rho, \theta] += 1$ 
6:
7: end
8:  $\rho^*, \theta^* = \underset{\rho, \theta}{\operatorname{argmax}} H[\rho, \theta]$ 
```

extension

#2

Hough Lines

```
1: Initialize  $H[\rho, \theta] = 0$ 
2: foreach edgel  $e \in I[x, y]$  do
3:    $\theta = \text{angle}(\nabla(I[x, y]))$ 
4:    $\rho = x \cos \theta + y \sin \theta$ 
5:    $H[\rho, \theta] +=$ 
6:
7: end
8:  $\rho^*, \theta^* = \underset{\rho, \theta}{\operatorname{argmax}} H[\rho, \theta]$ 
```

extension

#2

Hough Lines

```
1: Initialize  $H[\rho, \theta] = 0$ 
2: foreach edgel  $e \in I[x, y]$  do
3:    $\theta = \text{angle}(\nabla(I[x, y]))$ 
4:    $\rho = x \cos \theta + y \sin \theta$ 
5:    $H[\rho, \theta] += \|\nabla(I[x, y])\|$ 
6:
7: end
8:  $\rho^*, \theta^* = \underset{\rho, \theta}{\operatorname{argmax}} H[\rho, \theta]$ 
```

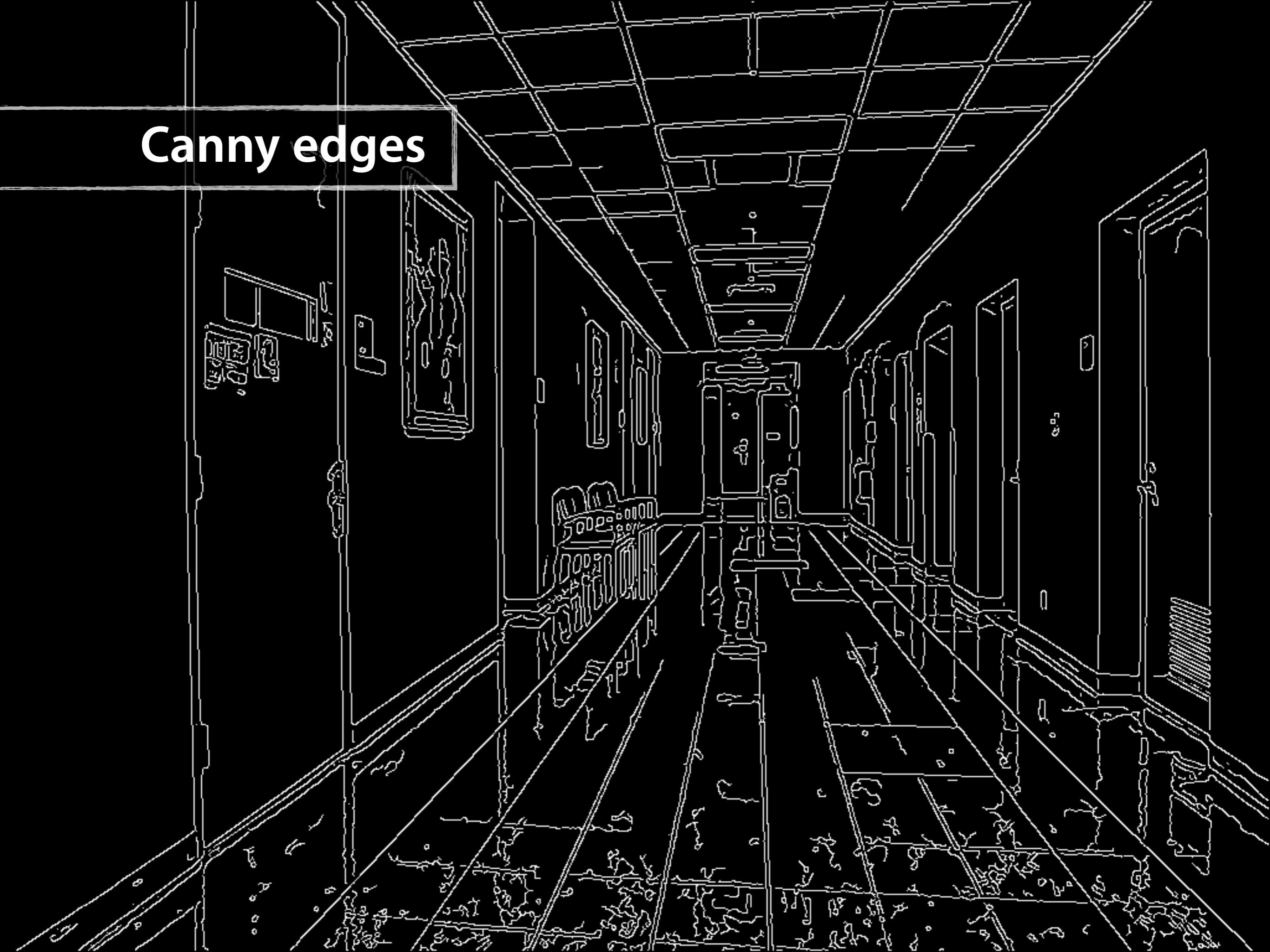
extension

#2

input image



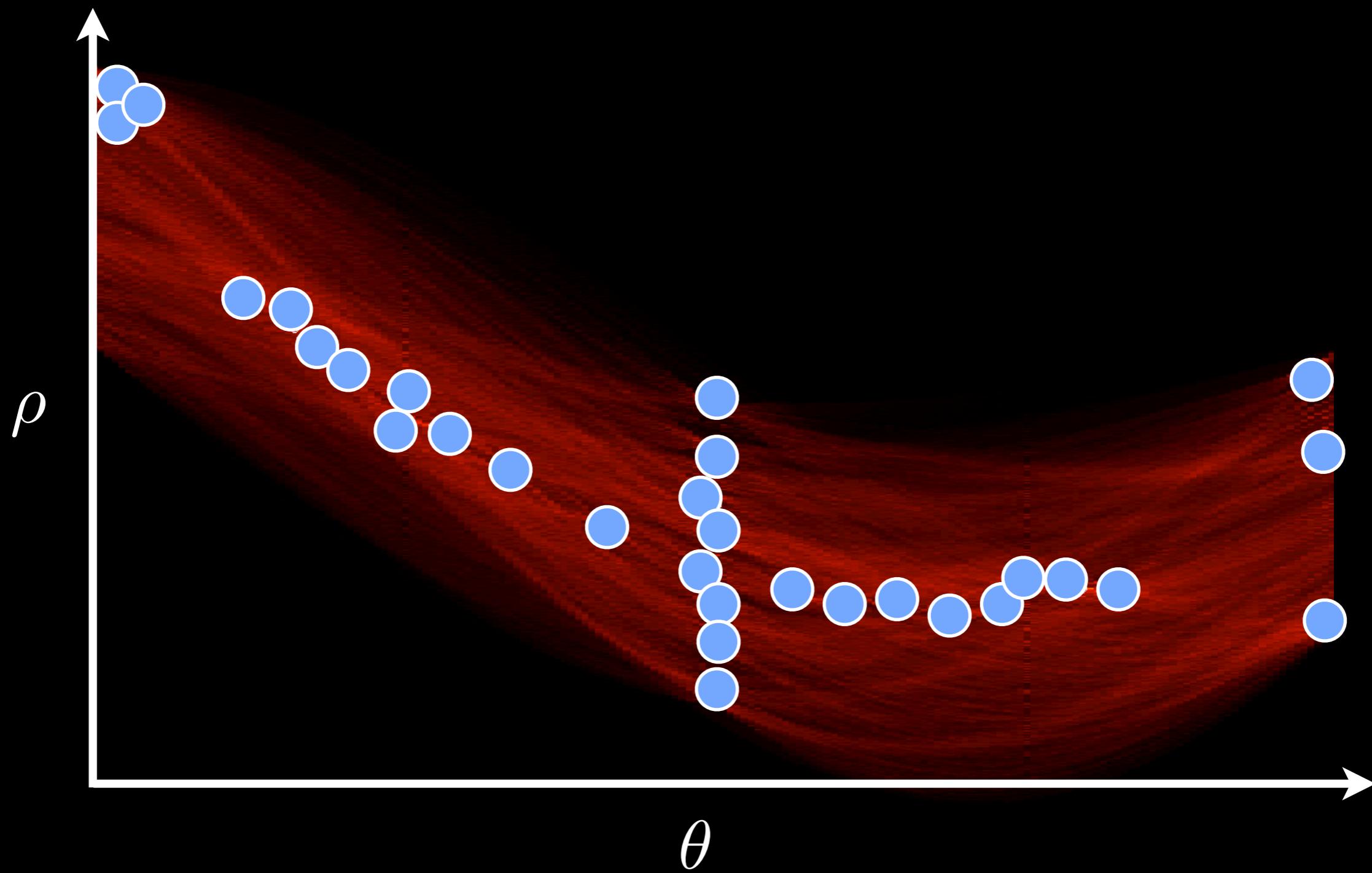
Canny edges



Hough lines



Hough space





Lane detection



Lane detection

A photograph of a street scene used for demonstrating lane detection. A thick orange line is drawn across the image, starting from the bottom left and curving upwards towards the center, indicating the detected lane boundary. In the background, there's a road, some trees, a building, and utility poles.

Lane detection

A photograph of a street scene used for demonstrating lane detection. A thick orange line is drawn across the image, starting from the bottom left and curving upwards towards the center, indicating the detected lane boundary. In the background, there's a road, some trees, a building, and utility poles.

Lane detection

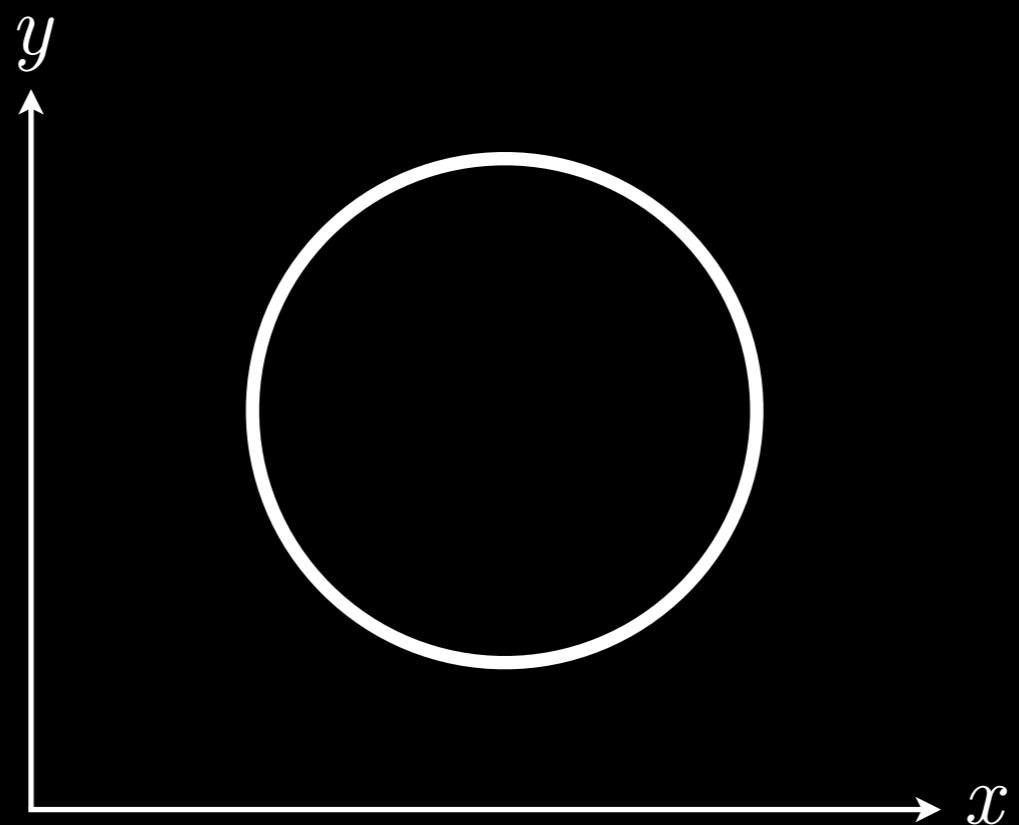
Hough Circle Transform

Assume
Fixed Radius

$$(x_i - \alpha)^2 + (y_i - \beta)^2 = r^2$$

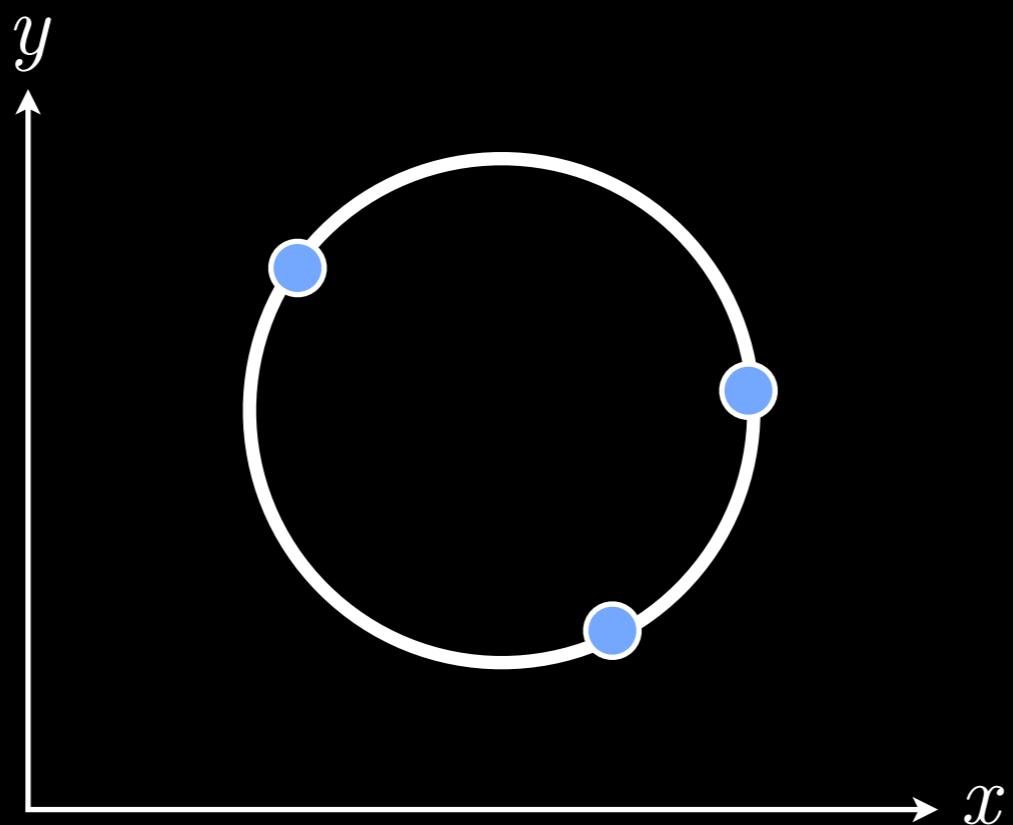
Assume
Fixed Radius

$$(x_i - \alpha)^2 + (y_i - \beta)^2 = r^2$$



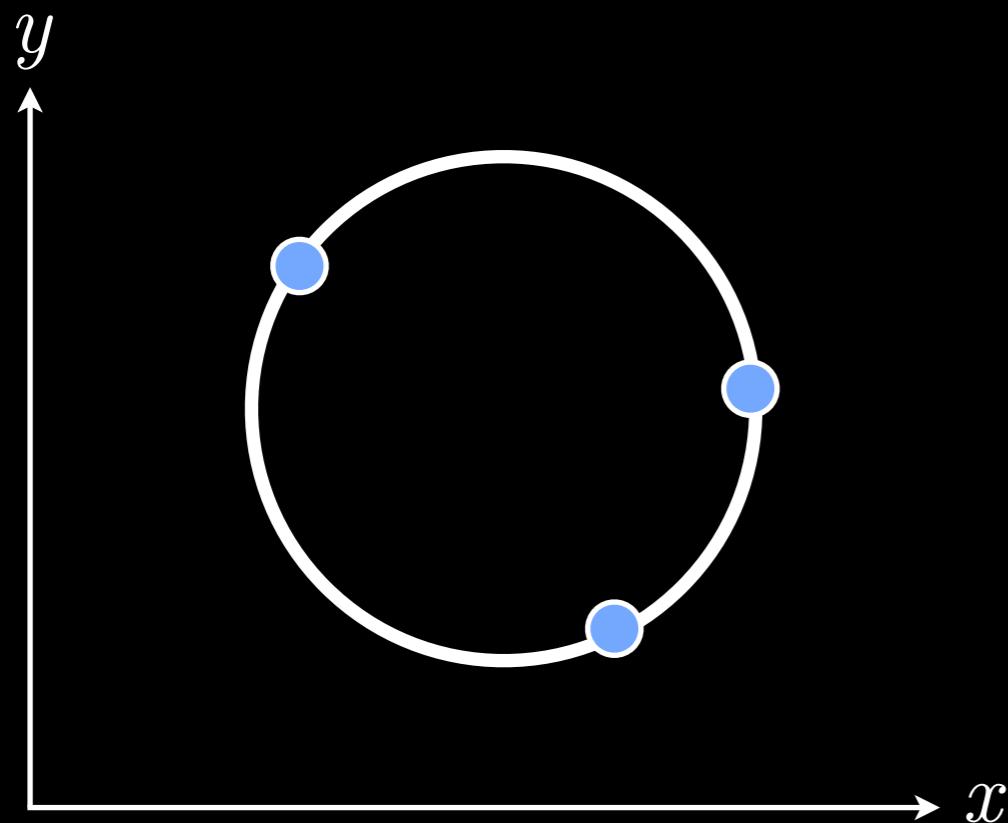
Assume
Fixed Radius

$$(x_i - \alpha)^2 + (y_i - \beta)^2 = r^2$$



Assume
Fixed Radius

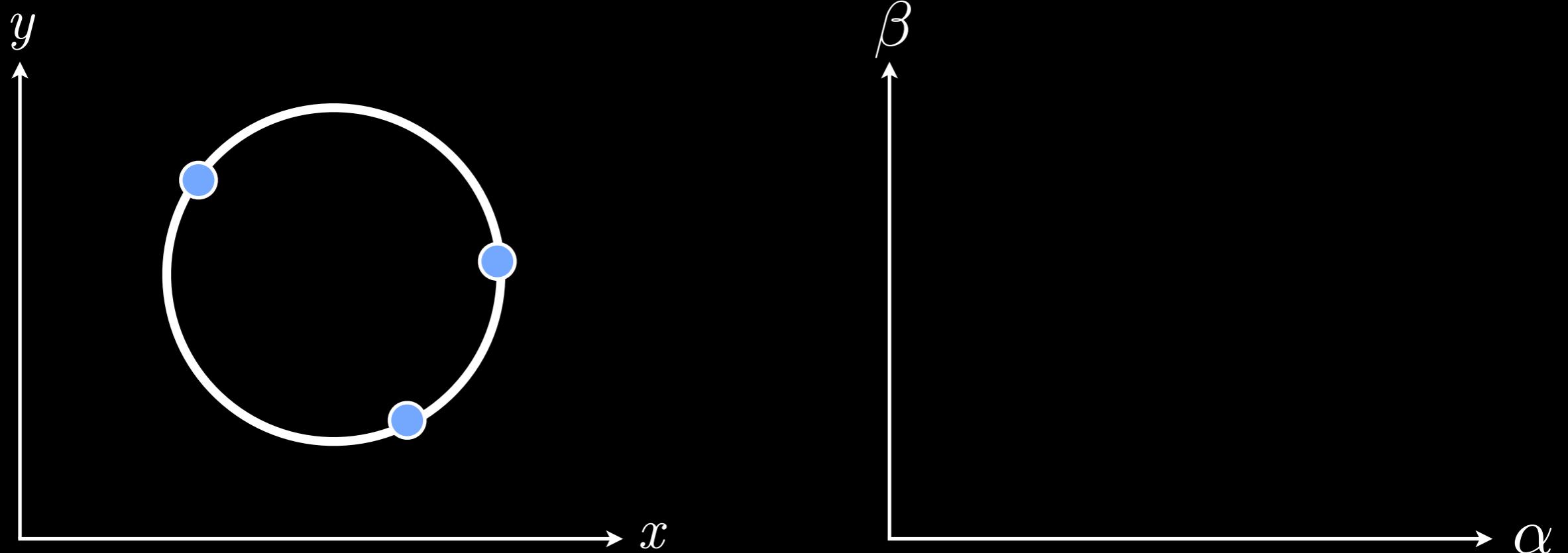
$$(x_i - \alpha)^2 + (y_i - \beta)^2 = r^2$$



Parameter space?

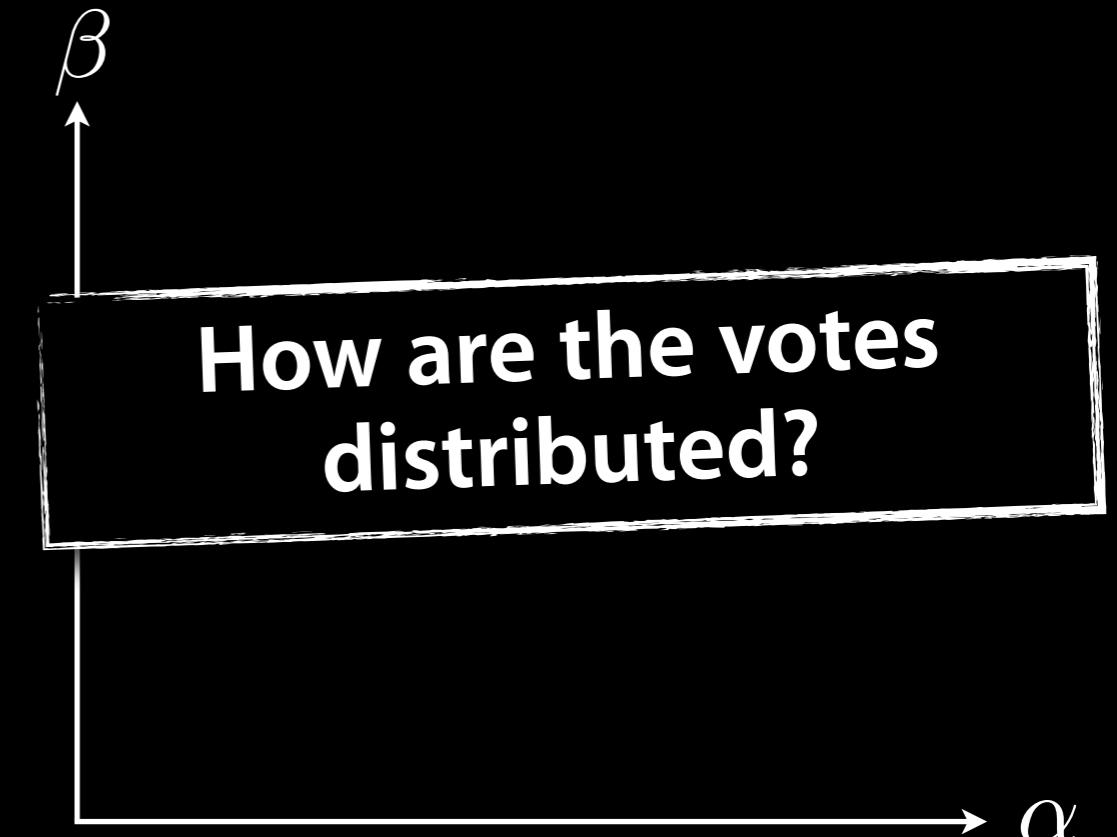
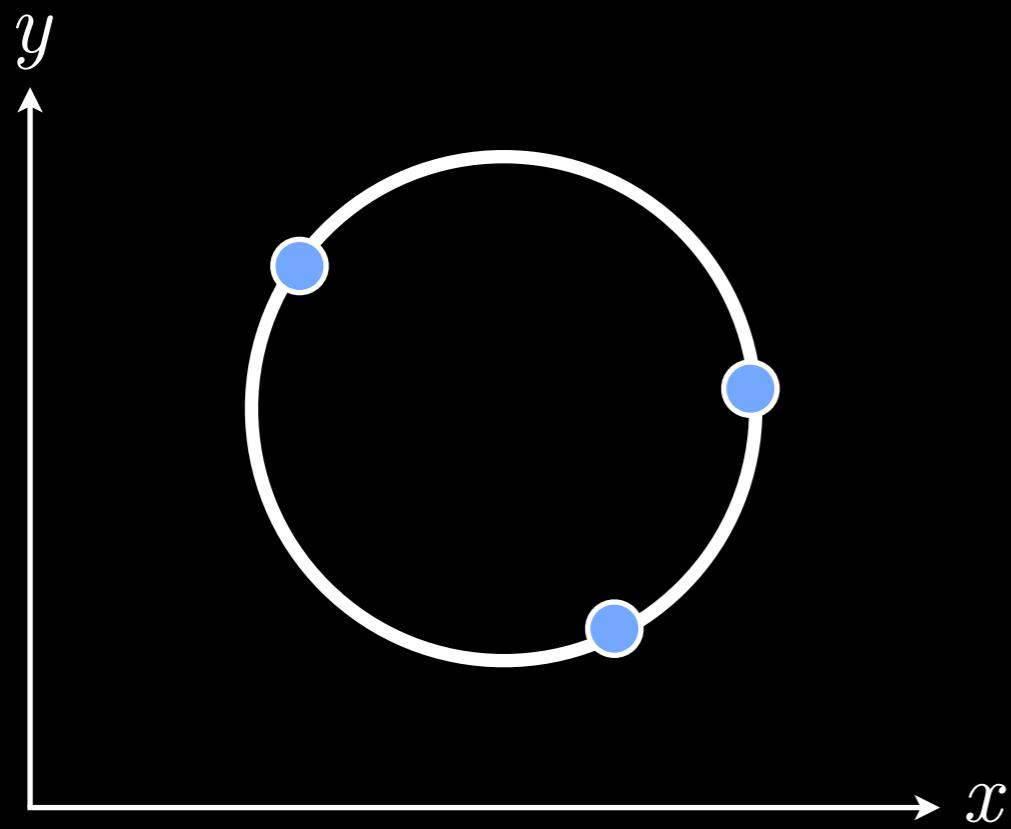
Assume
Fixed Radius

$$(x_i - \alpha)^2 + (y_i - \beta)^2 = r^2$$



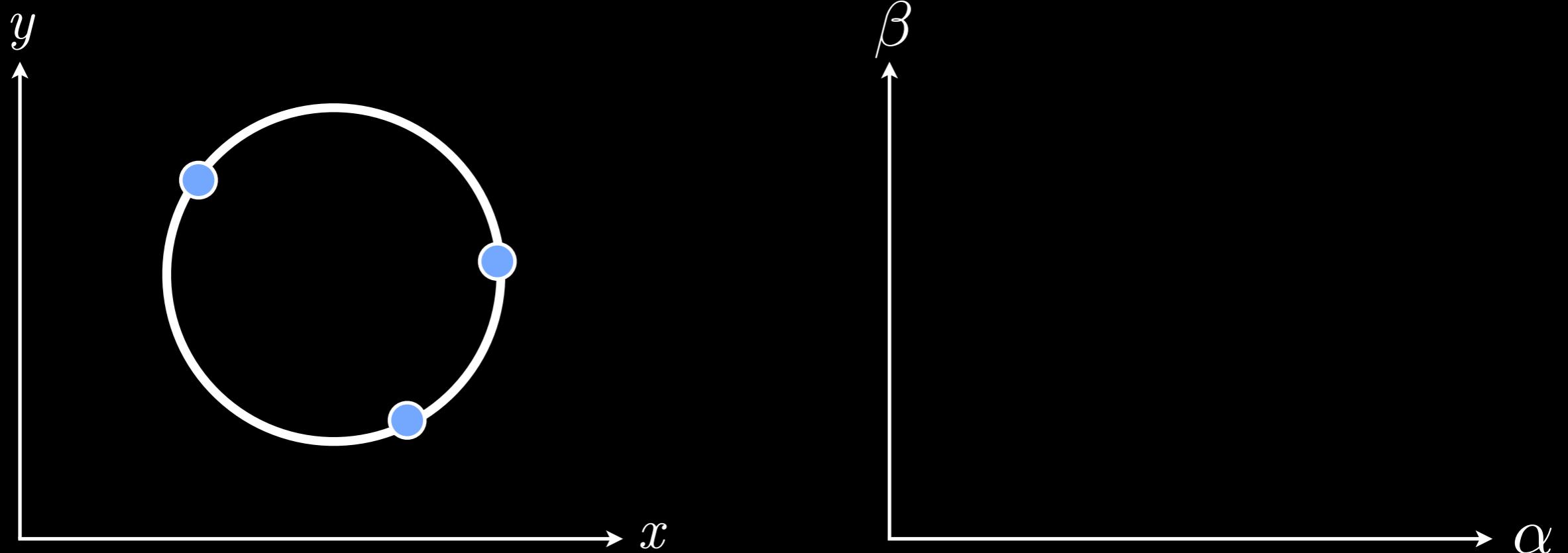
Assume
Fixed Radius

$$(x_i - \alpha)^2 + (y_i - \beta)^2 = r^2$$



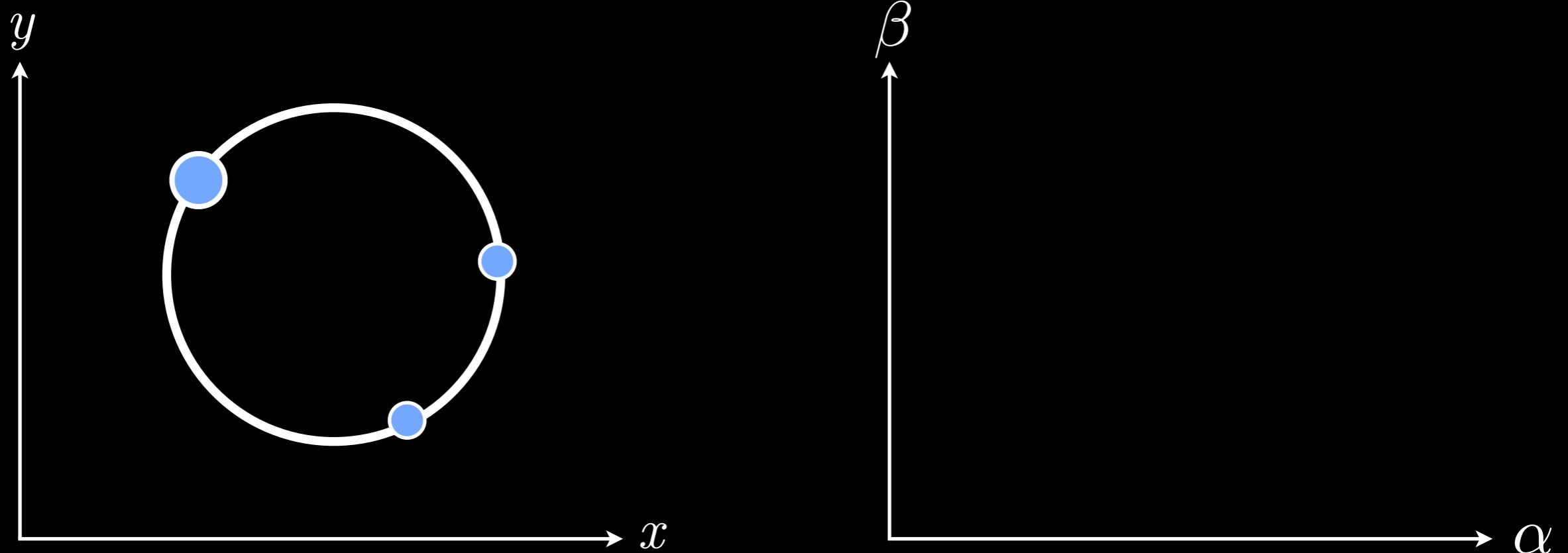
Assume
Fixed Radius

$$(x_i - \alpha)^2 + (y_i - \beta)^2 = r^2$$



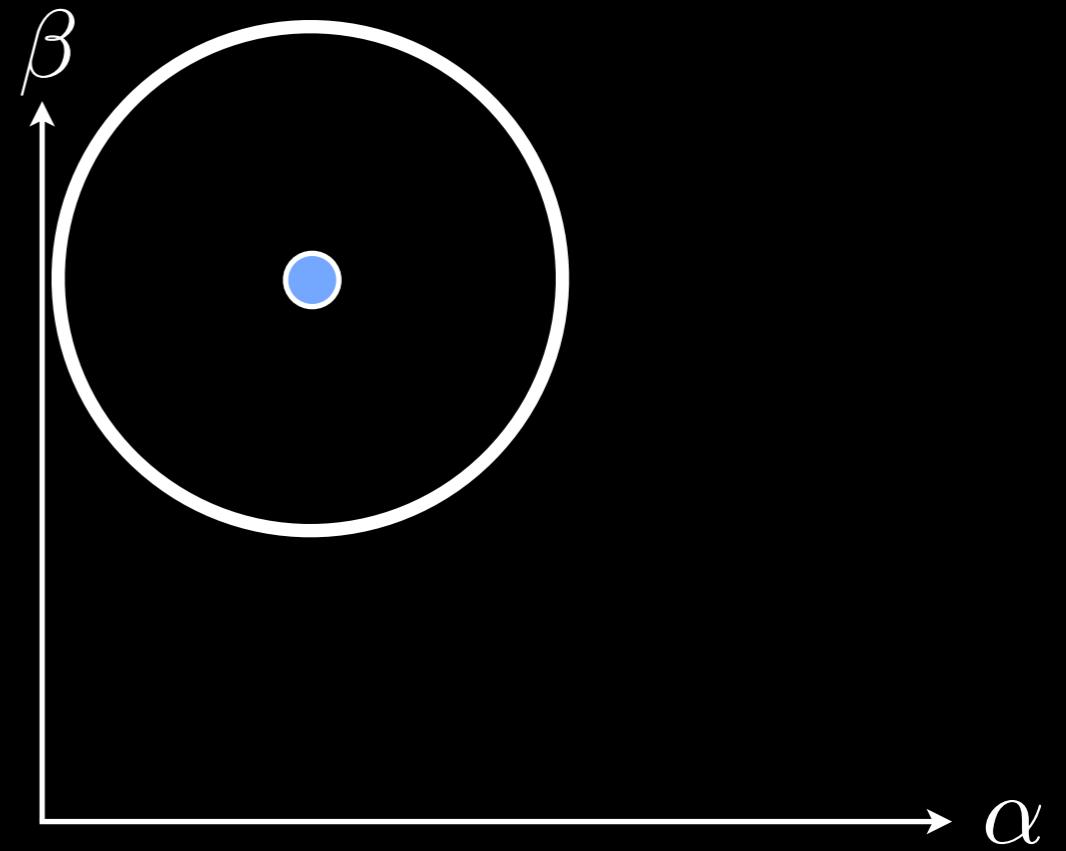
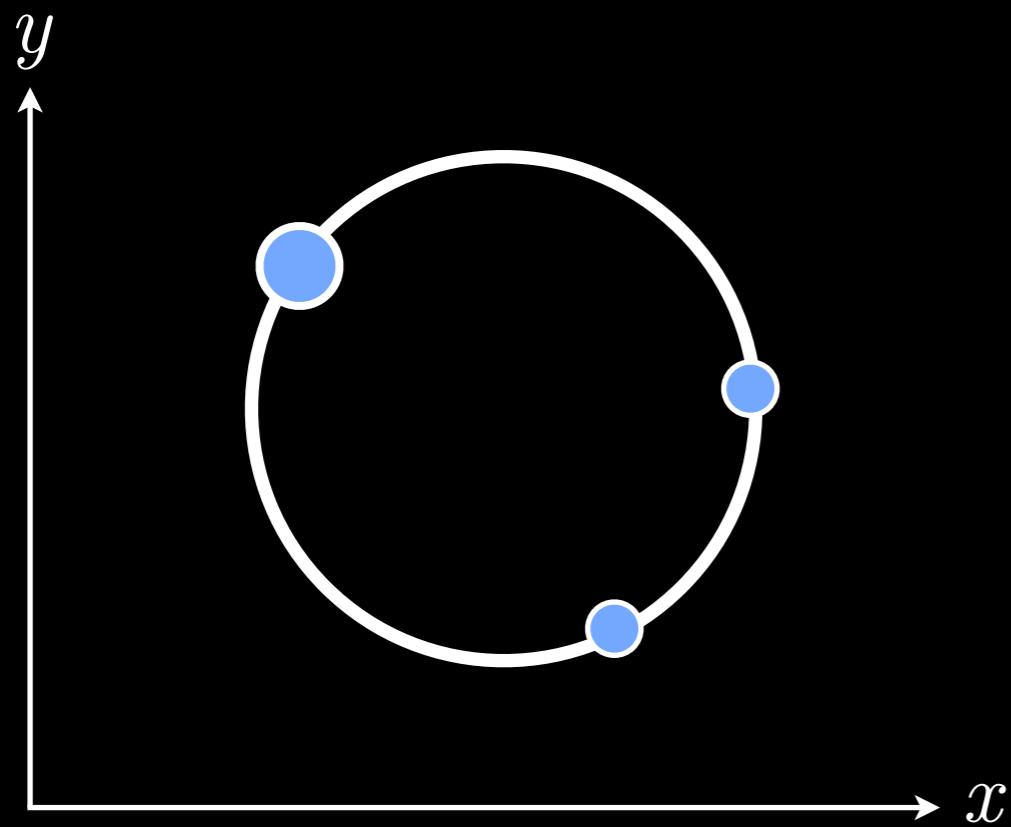
Assume
Fixed Radius

$$(x_i - \alpha)^2 + (y_i - \beta)^2 = r^2$$



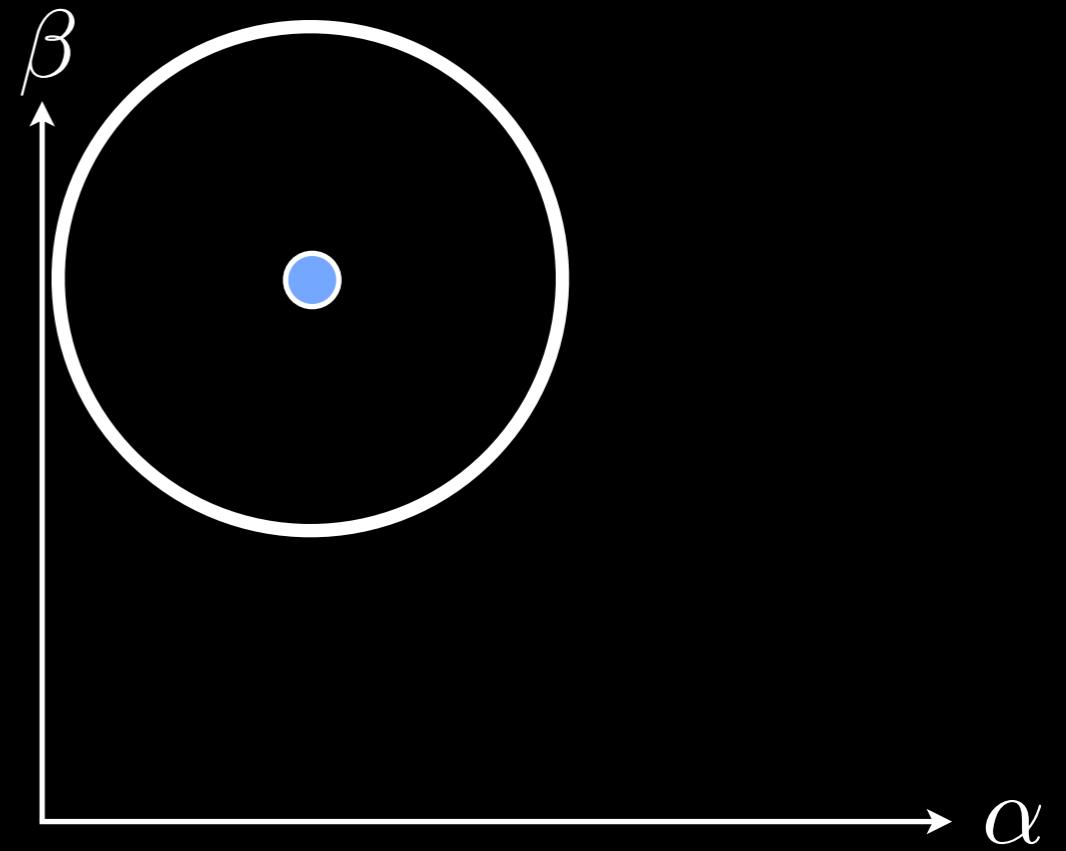
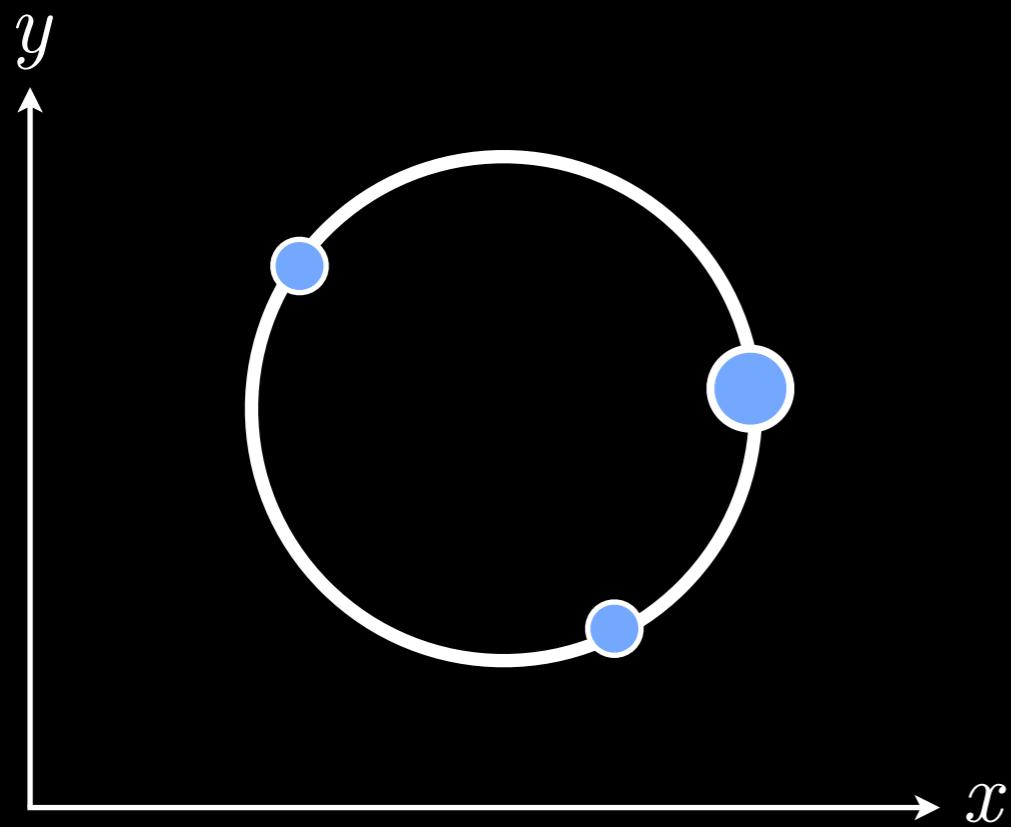
Assume
Fixed Radius

$$(x_i - \alpha)^2 + (y_i - \beta)^2 = r^2$$



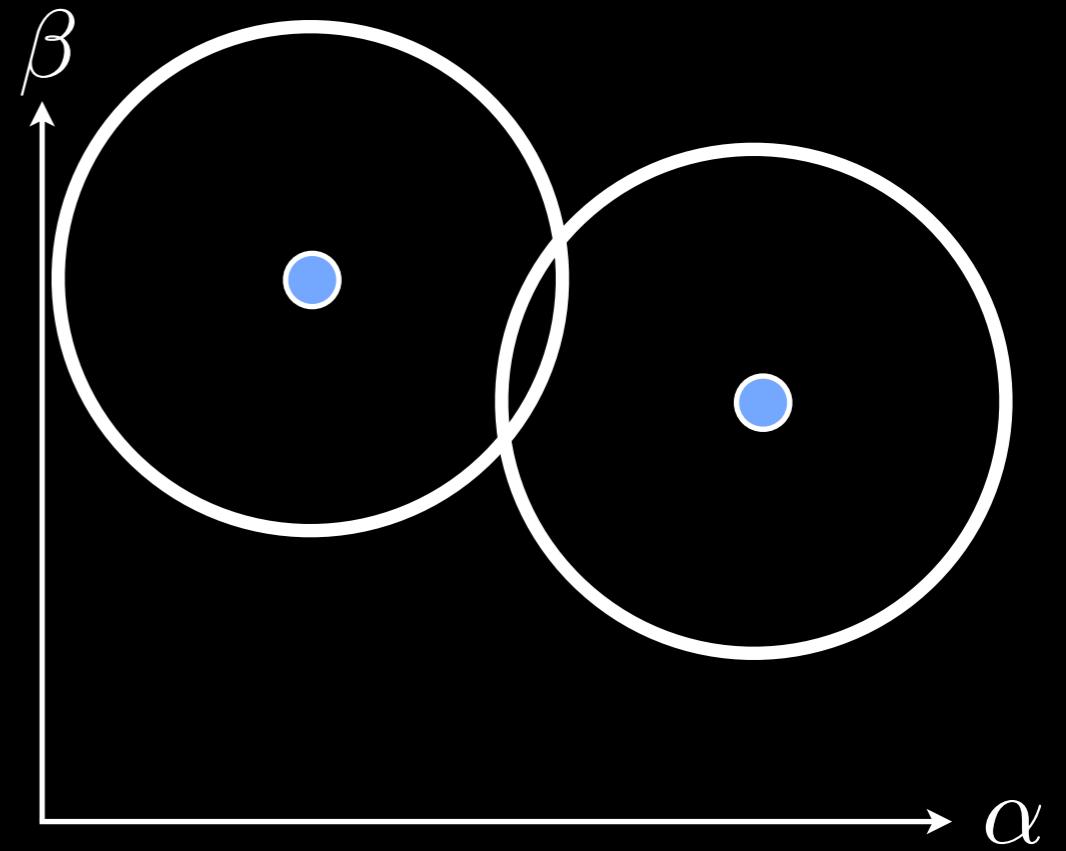
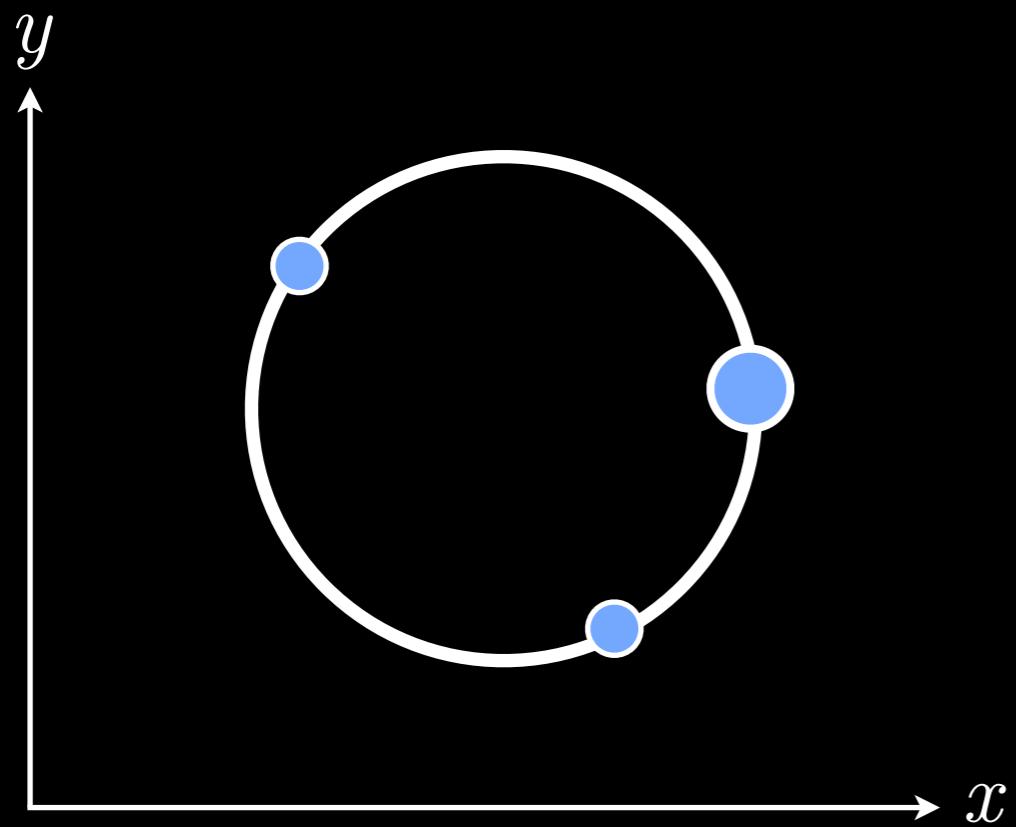
Assume
Fixed Radius

$$(x_i - \alpha)^2 + (y_i - \beta)^2 = r^2$$



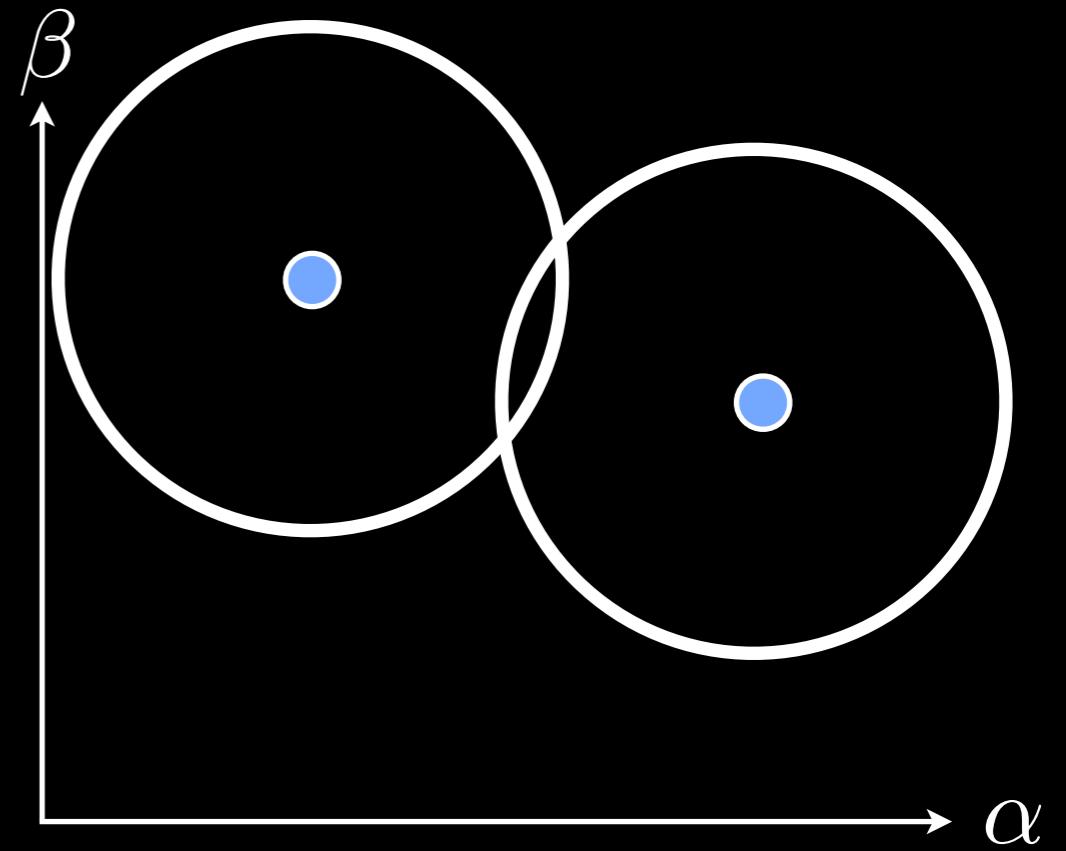
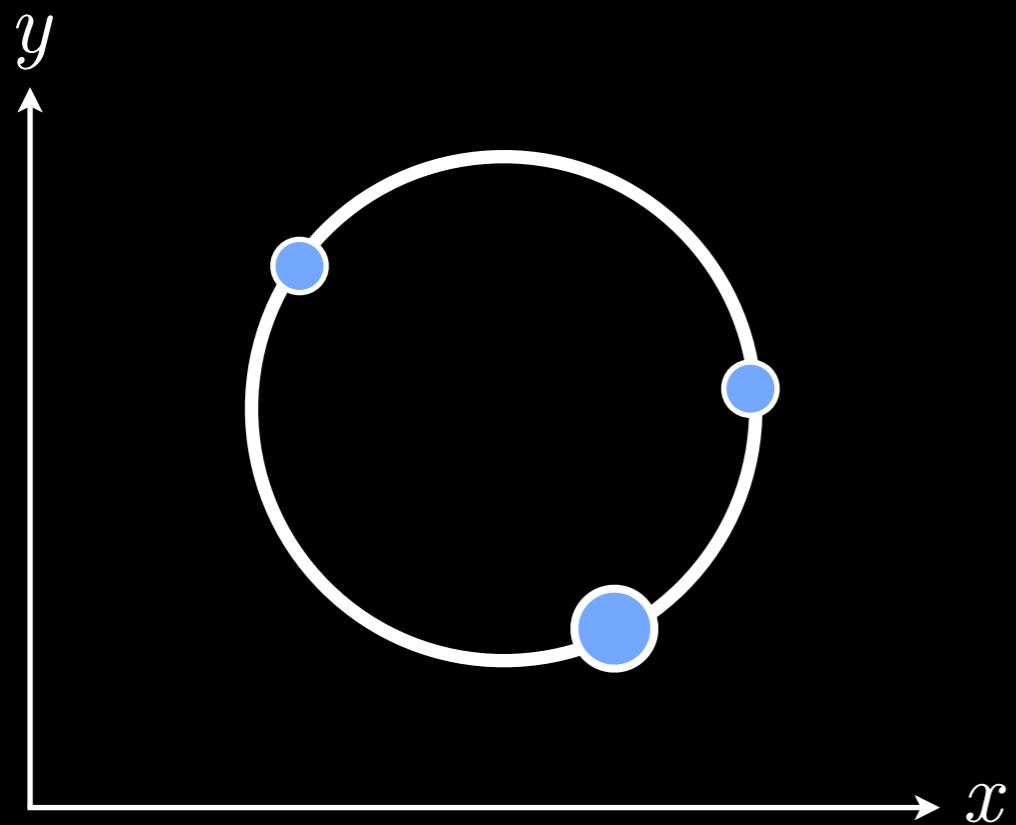
Assume
Fixed Radius

$$(x_i - \alpha)^2 + (y_i - \beta)^2 = r^2$$



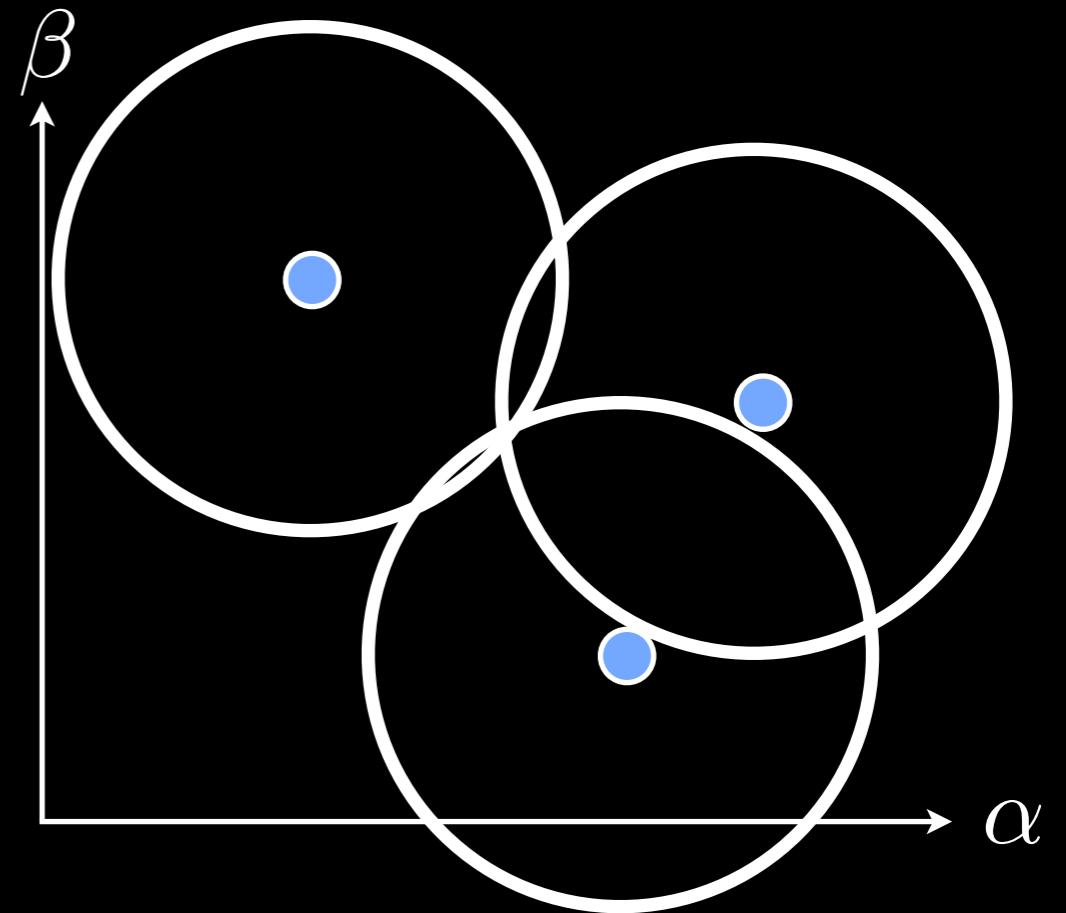
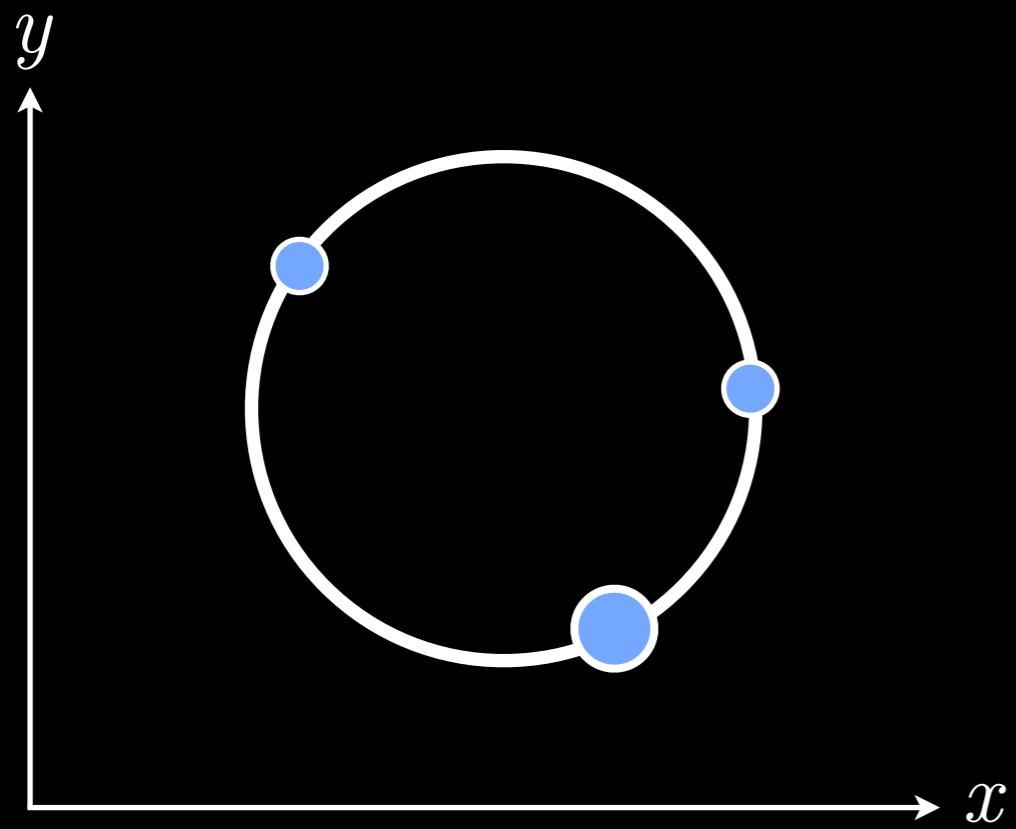
Assume
Fixed Radius

$$(x_i - \alpha)^2 + (y_i - \beta)^2 = r^2$$



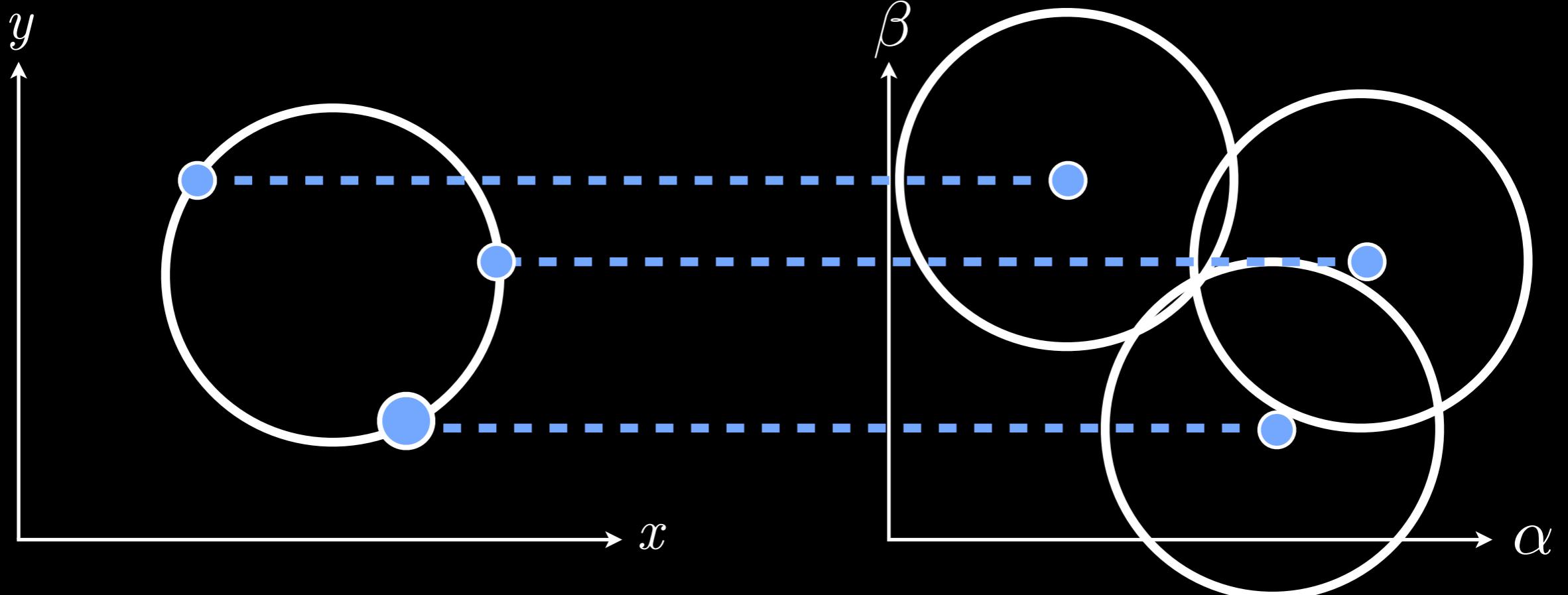
Assume
Fixed Radius

$$(x_i - \alpha)^2 + (y_i - \beta)^2 = r^2$$



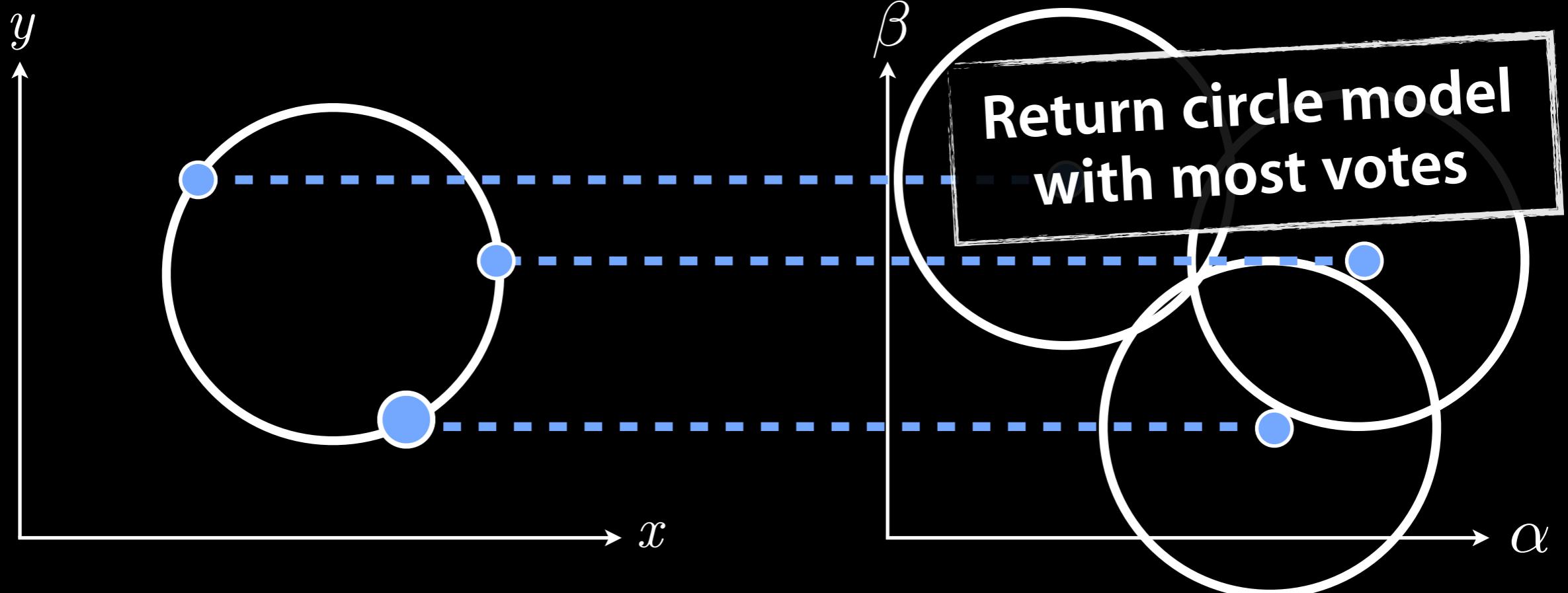
Assume
Fixed Radius

$$(x_i - \alpha)^2 + (y_i - \beta)^2 = r^2$$



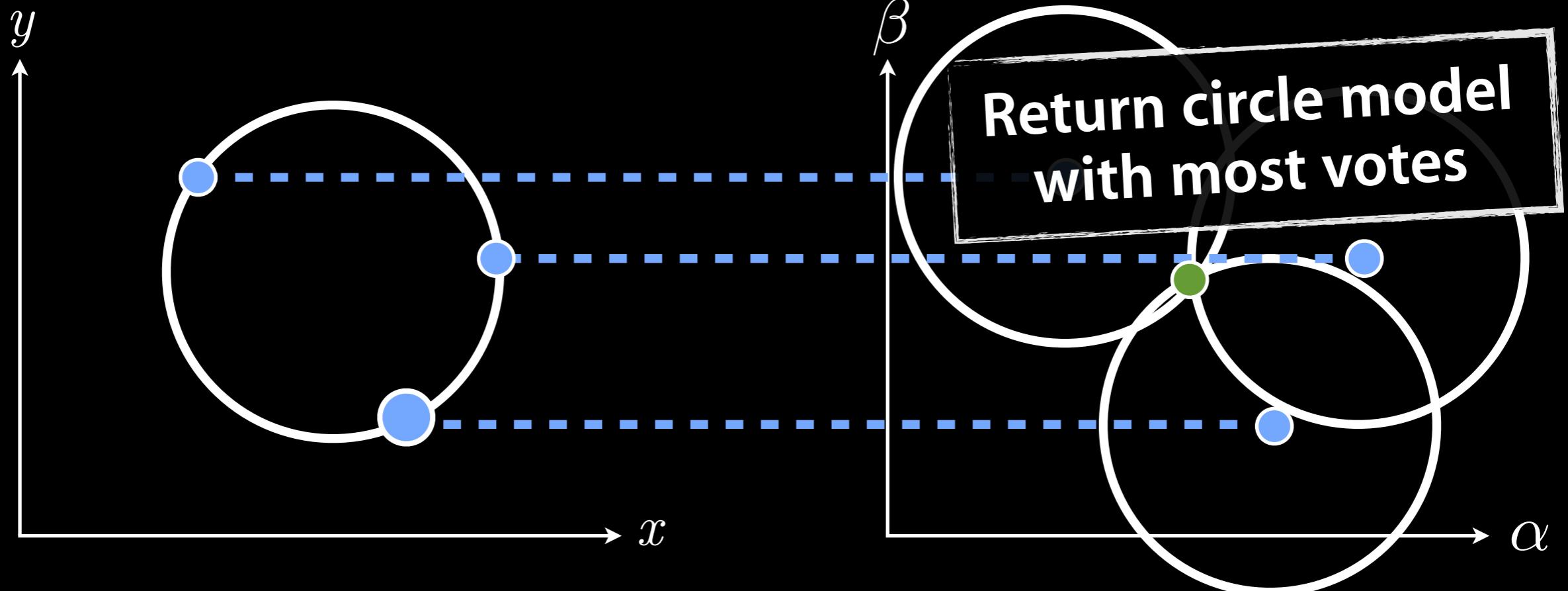
Assume
Fixed Radius

$$(x_i - \alpha)^2 + (y_i - \beta)^2 = r^2$$



Assume
Fixed Radius

$$(x_i - \alpha)^2 + (y_i - \beta)^2 = r^2$$

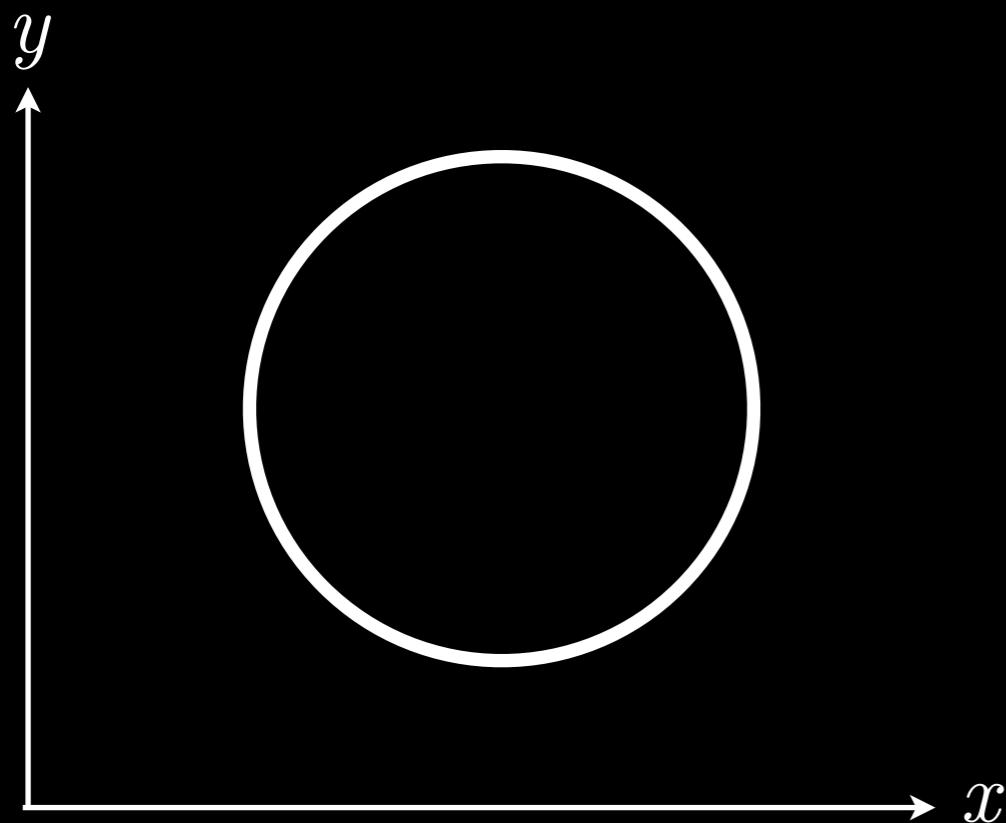


Assume
Unknown Radius

$$(x_i - \alpha)^2 + (y_i - \beta)^2 = r^2$$

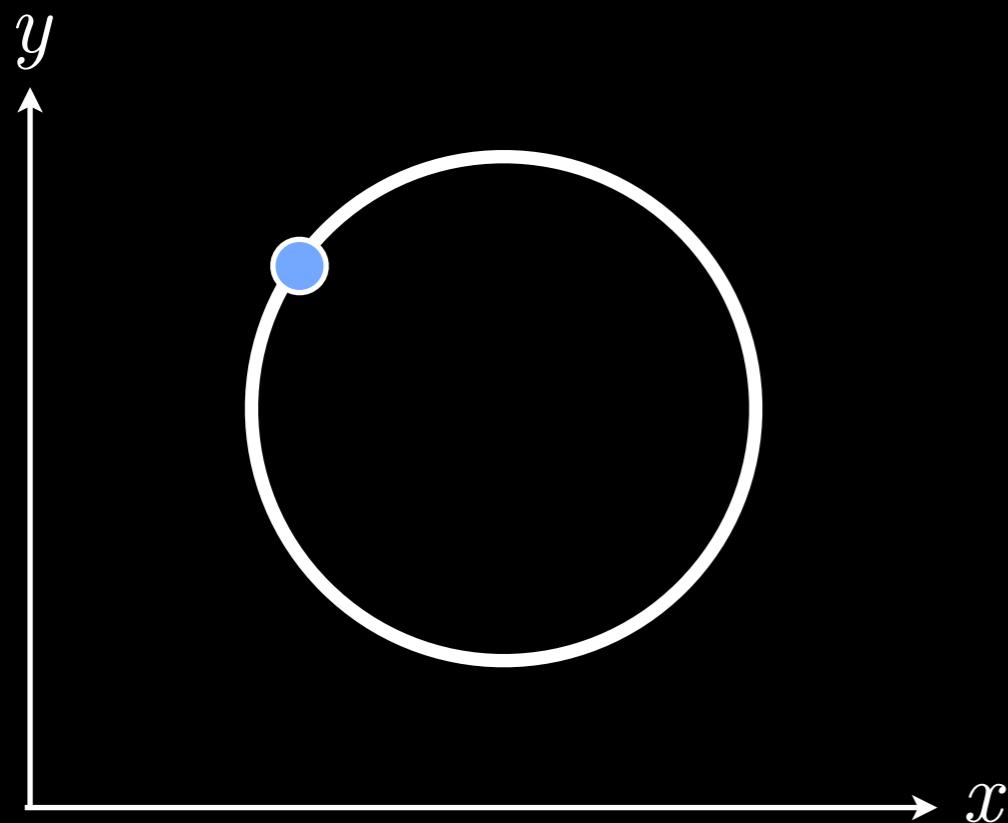
Assume
Unknown Radius

$$(x_i - \alpha)^2 + (y_i - \beta)^2 = r^2$$



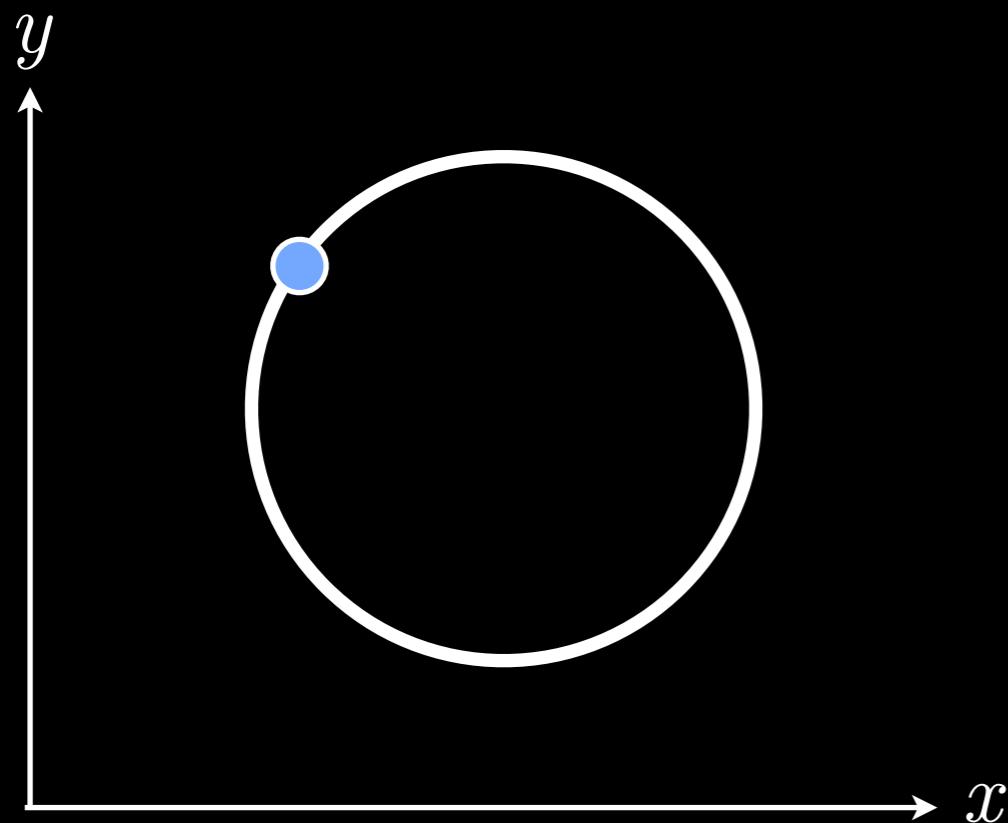
Assume
Unknown Radius

$$(x_i - \alpha)^2 + (y_i - \beta)^2 = r^2$$



Assume
Unknown Radius

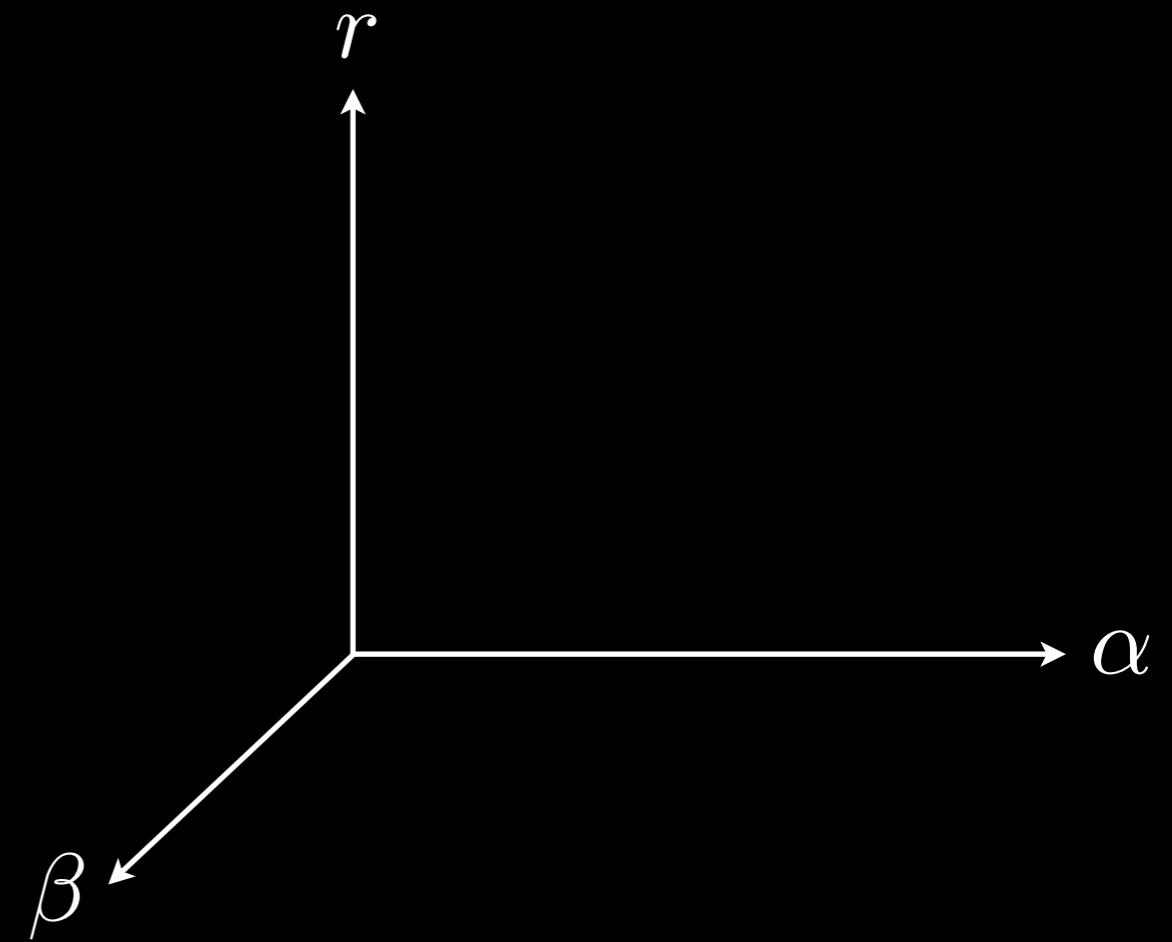
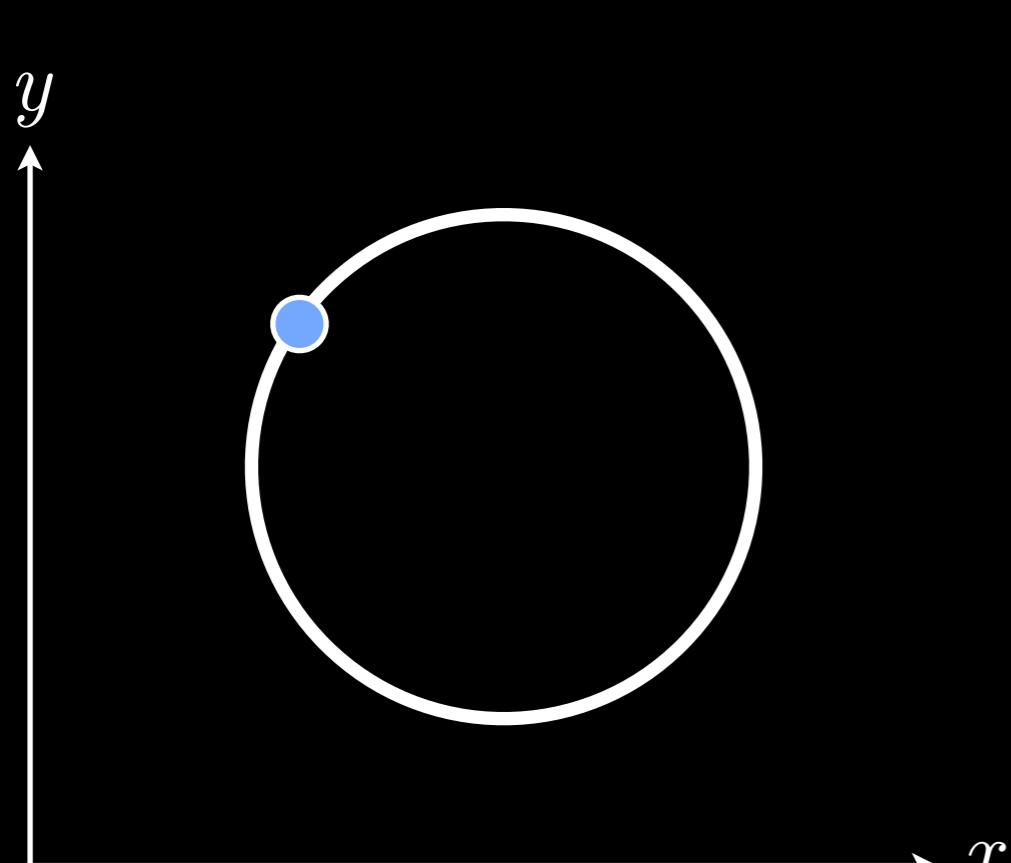
$$(x_i - \alpha)^2 + (y_i - \beta)^2 = r^2$$



Parameter space?

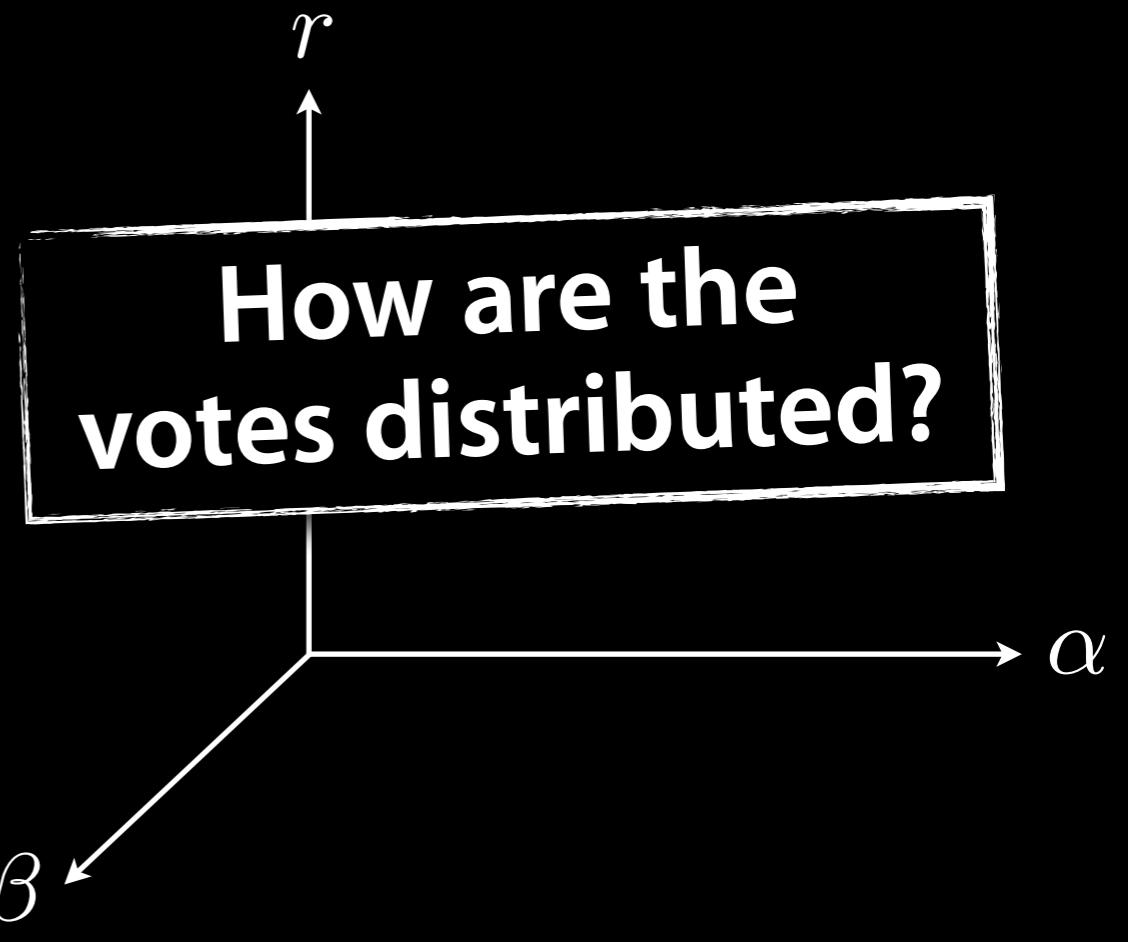
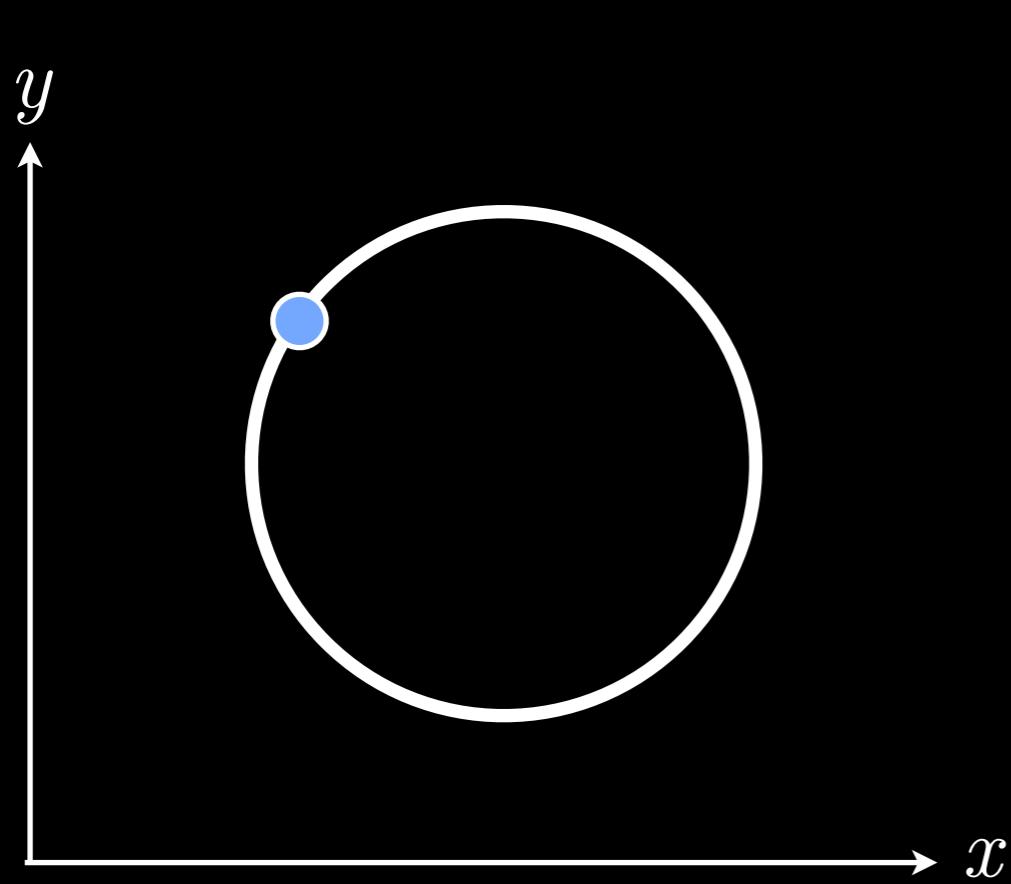
Assume
Unknown Radius

$$(x_i - \alpha)^2 + (y_i - \beta)^2 = r^2$$



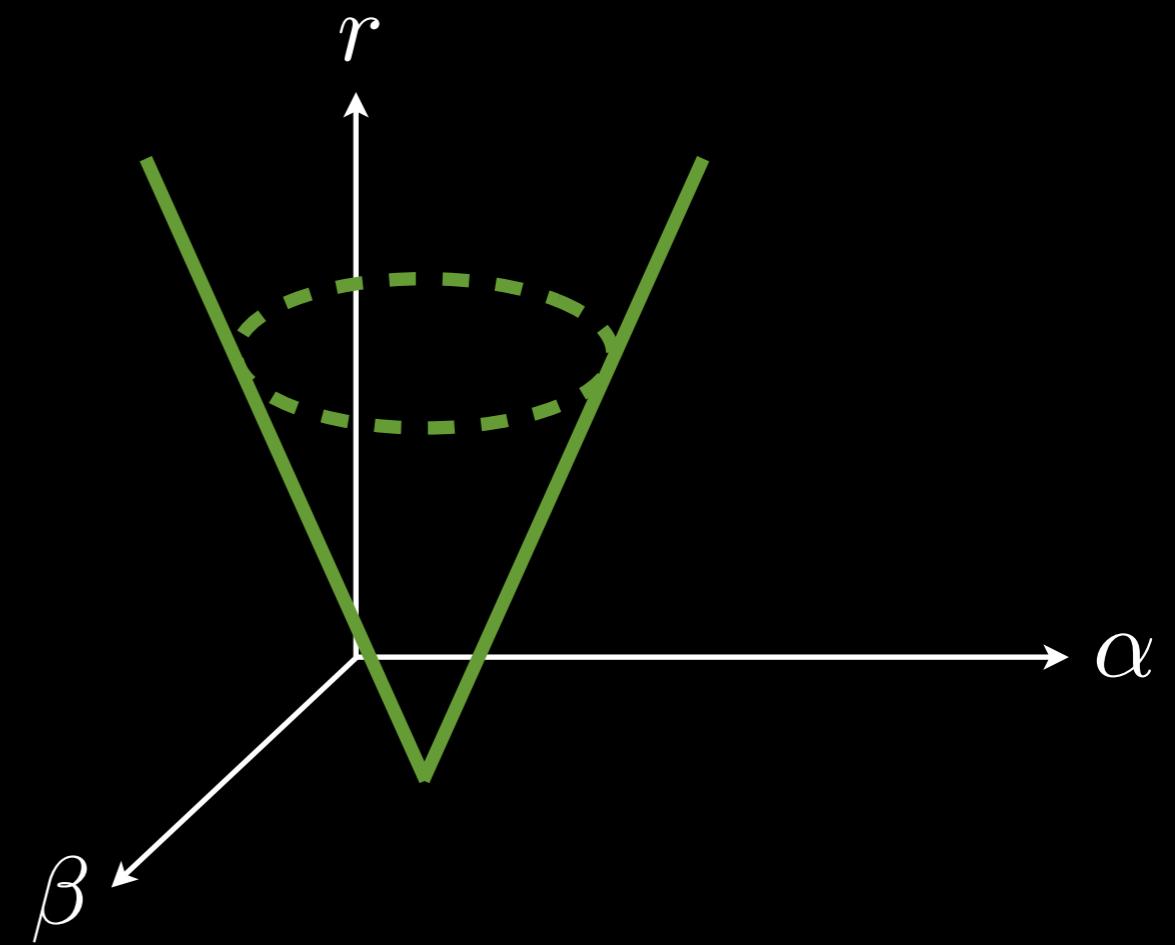
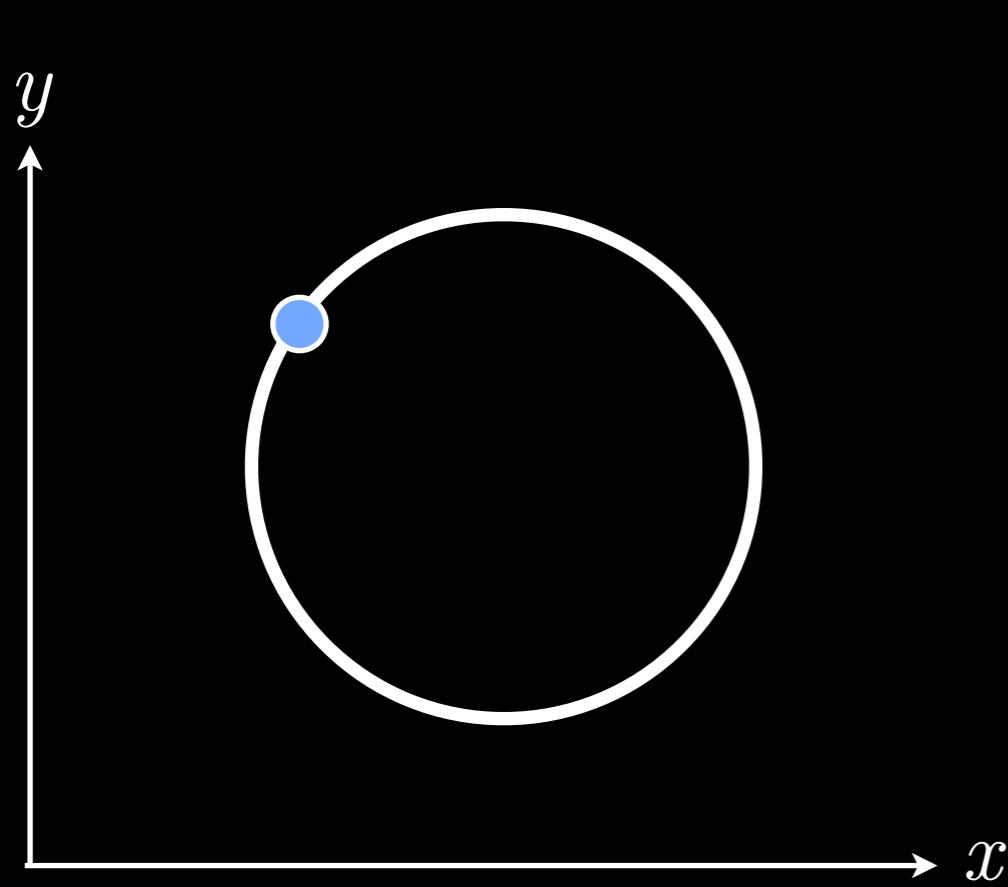
Assume
Unknown Radius

$$(x_i - \alpha)^2 + (y_i - \beta)^2 = r^2$$



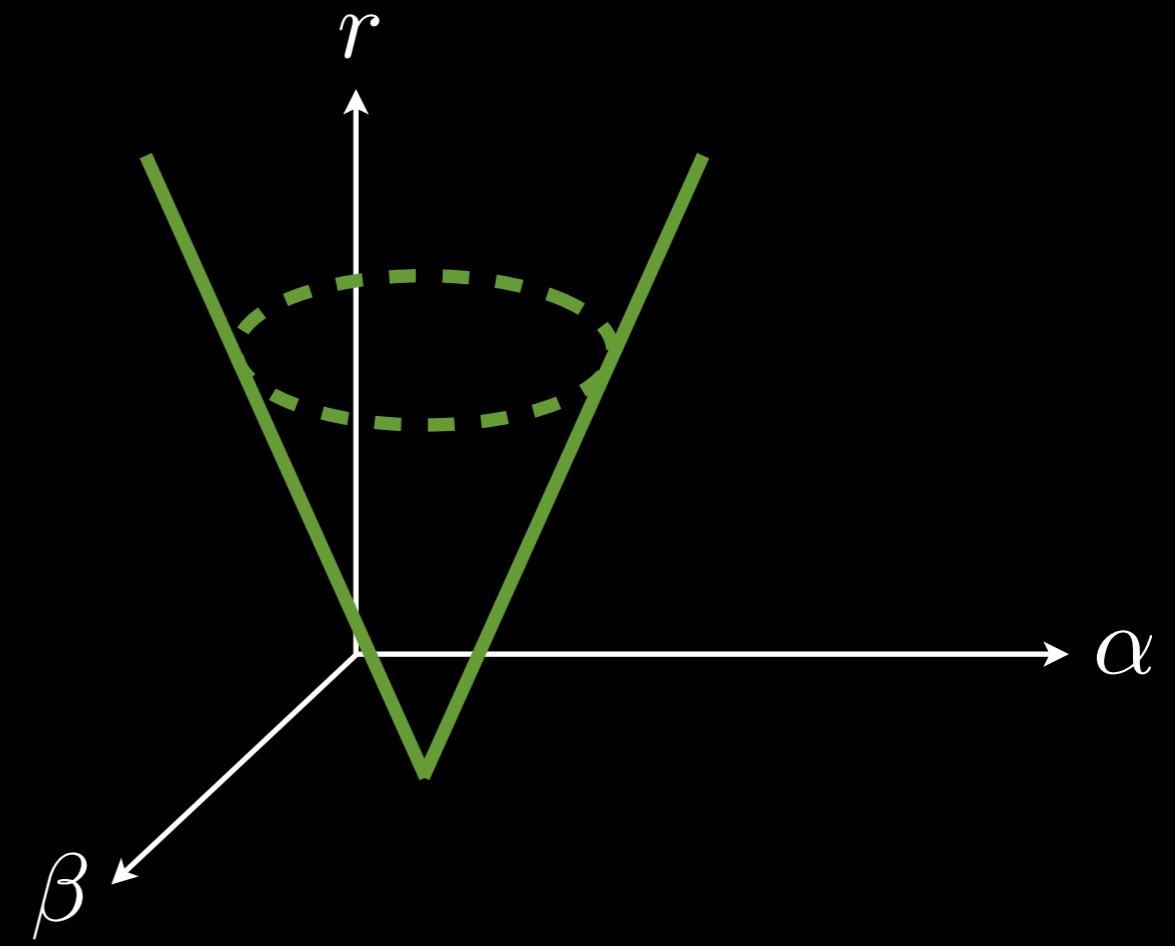
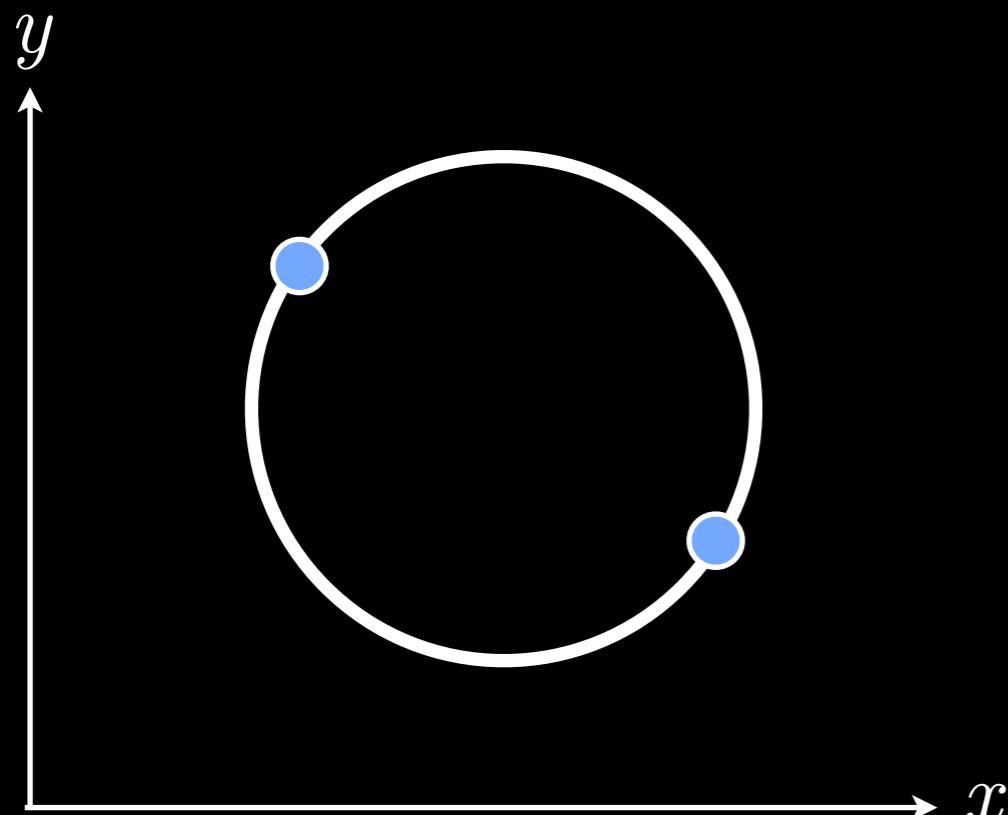
Assume
Unknown Radius

$$(x_i - \alpha)^2 + (y_i - \beta)^2 = r^2$$



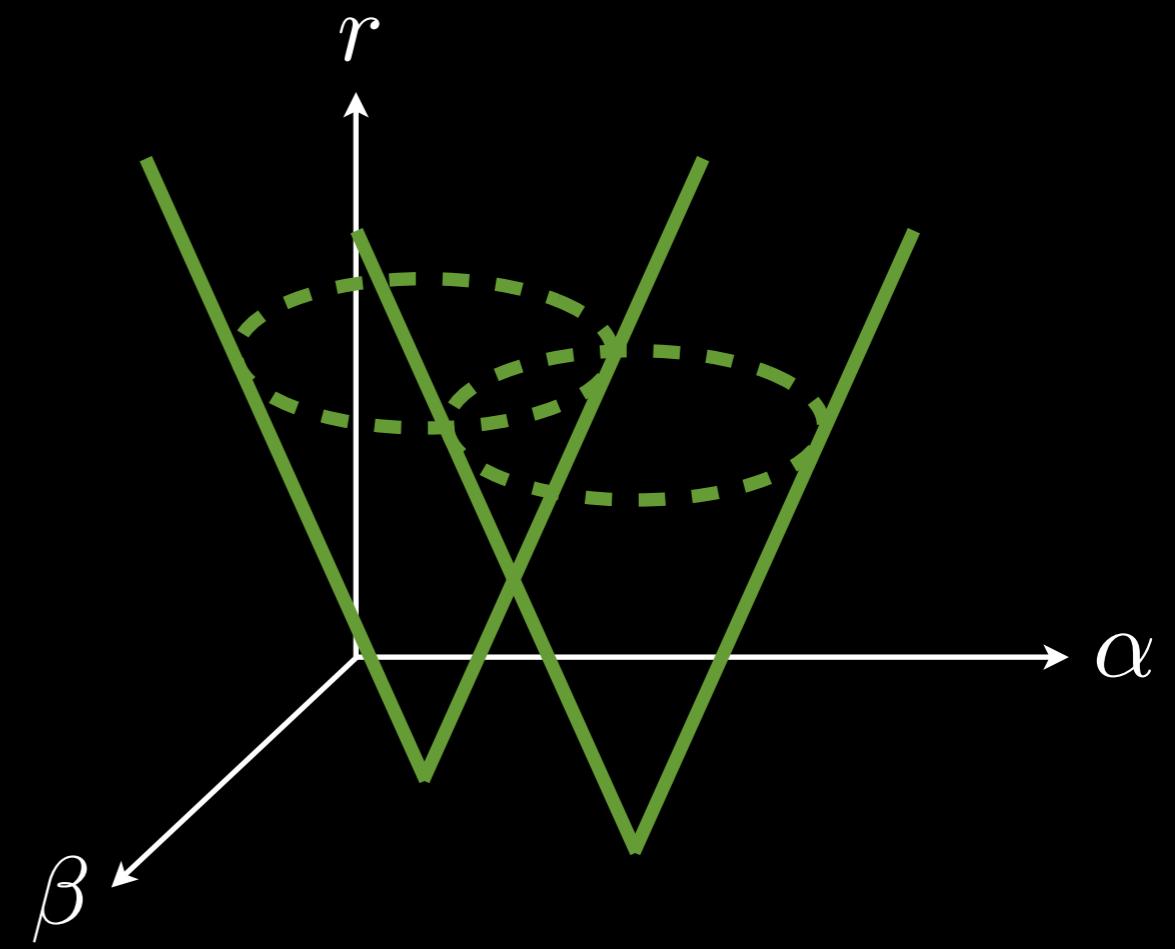
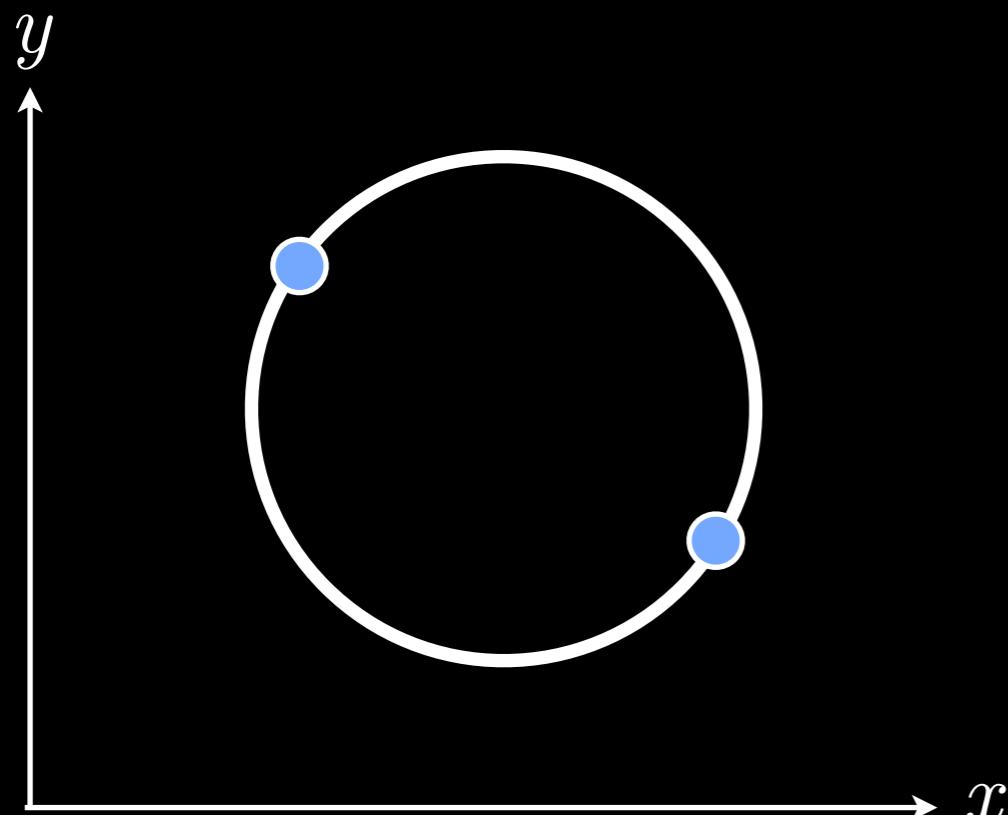
Assume
Unknown Radius

$$(x_i - \alpha)^2 + (y_i - \beta)^2 = r^2$$



Assume
Unknown Radius

$$(x_i - \alpha)^2 + (y_i - \beta)^2 = r^2$$



Unknown Radius
Known Gradient
Direction

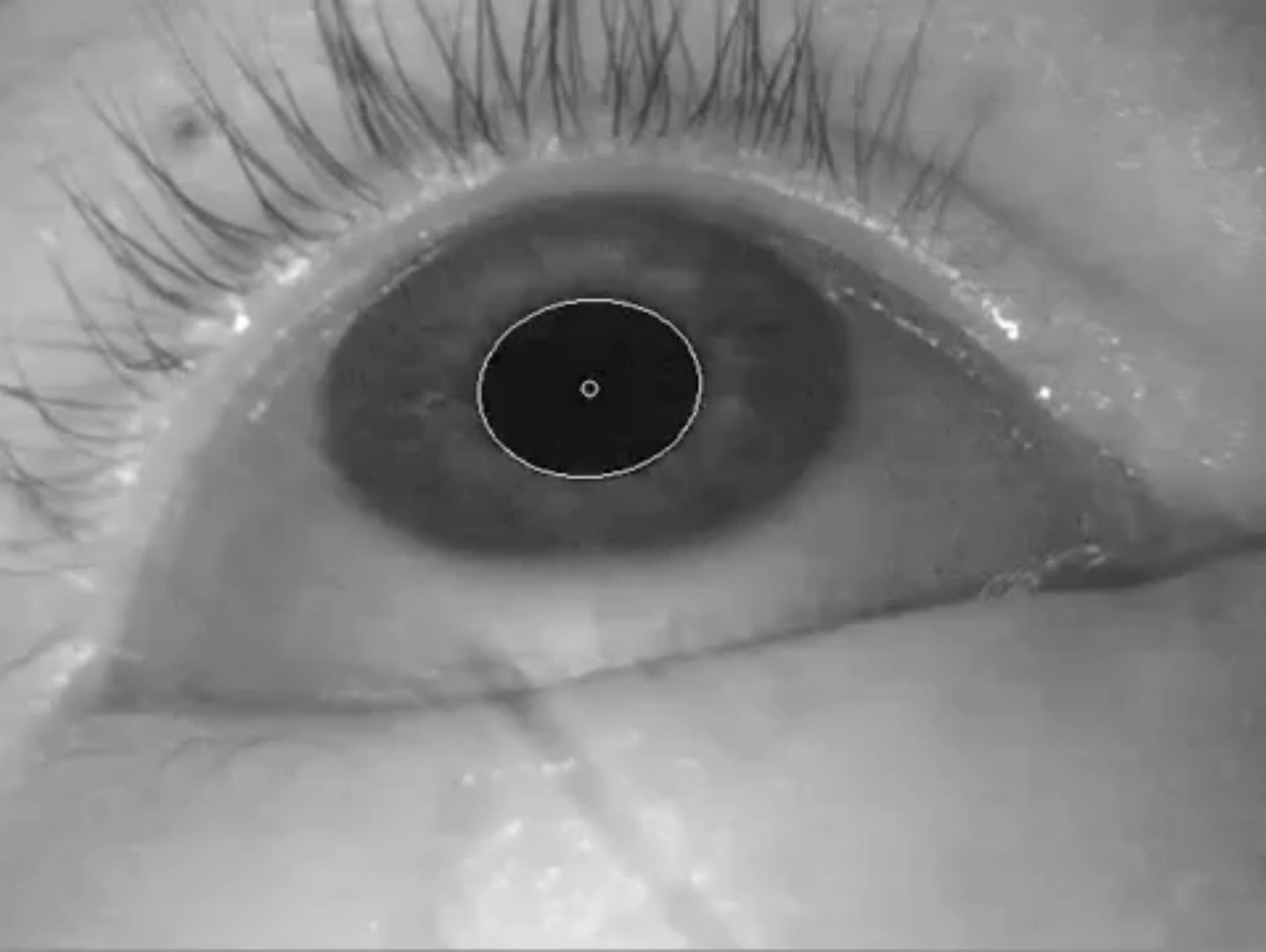
$$(x_i - \alpha)^2 + (y_i - \beta)^2 = r^2$$

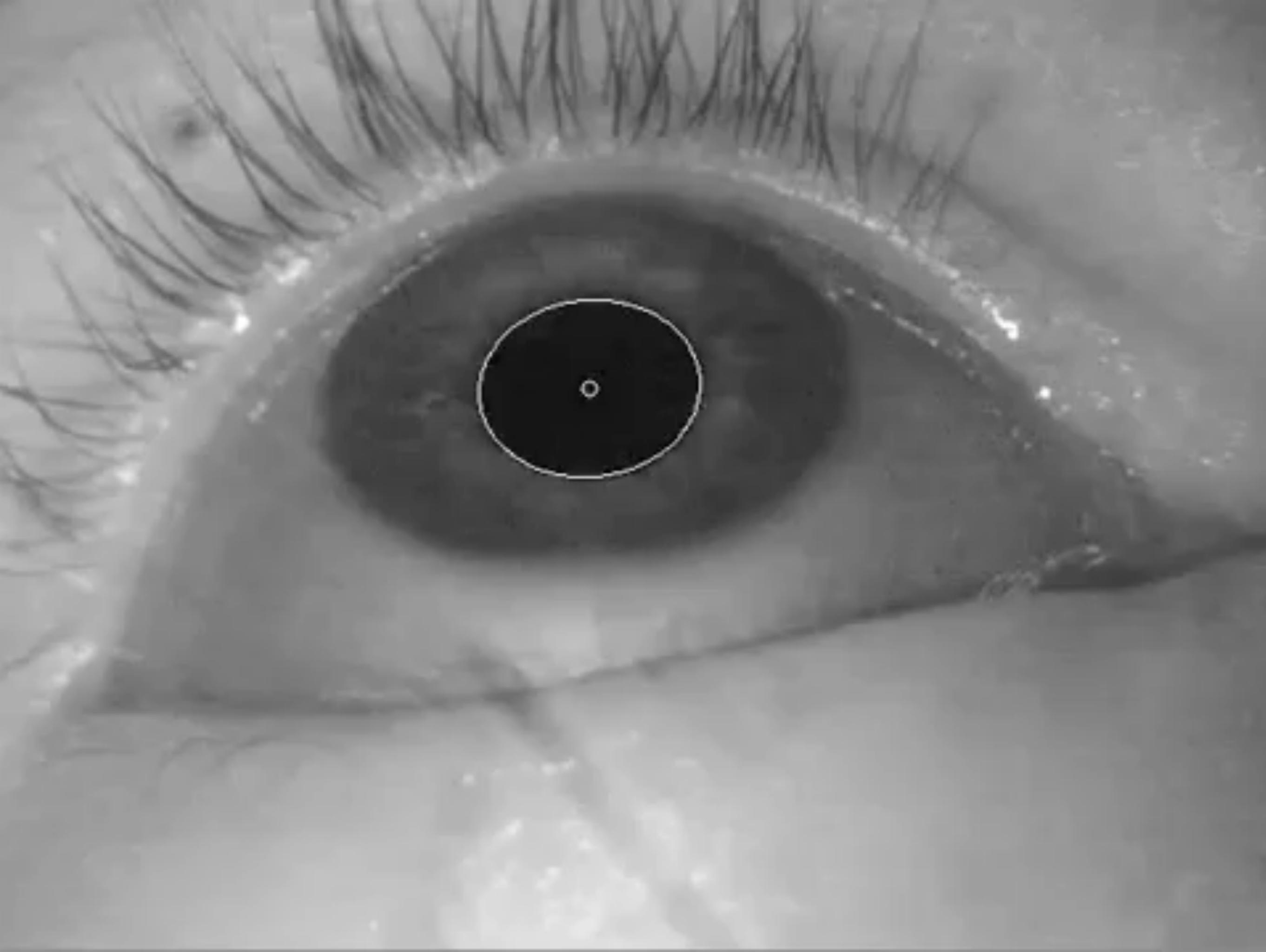


Yellow



Yellow





PROS

PROS

All points processed independently

PROS

All points processed independently

Robustness to **OUTLIERS**

PROS

All points processed independently

Robustness to **OUTLIERS**

Can detect multiple model instances in
a single pass

CONS

Search time complexity increases exponentially with number of parameters

CONS

Search time complexity increases exponentially with number of parameters

Non-target shapes can produce spurious peaks in parameter space

CONS

Search time complexity increases exponentially with number of parameters

Non-target shapes can produce spurious peaks in parameter space

Difficult to select an optimal quantization