Model Fitting: Line and Circle Detection

Alessandro Giusti

Slides in part adapted from Erkut Erdem https://web.cs.hacettepe.edu.tr/~erkut and various other authors credited on individual slides

Edges are a local, **low-level** feature

original

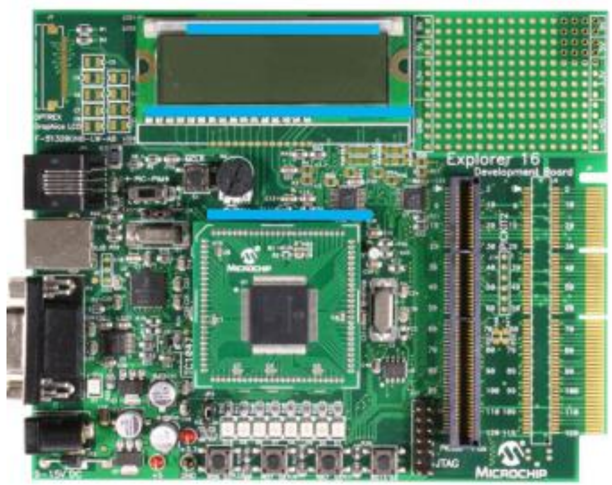
Canny with $\sigma = 1$

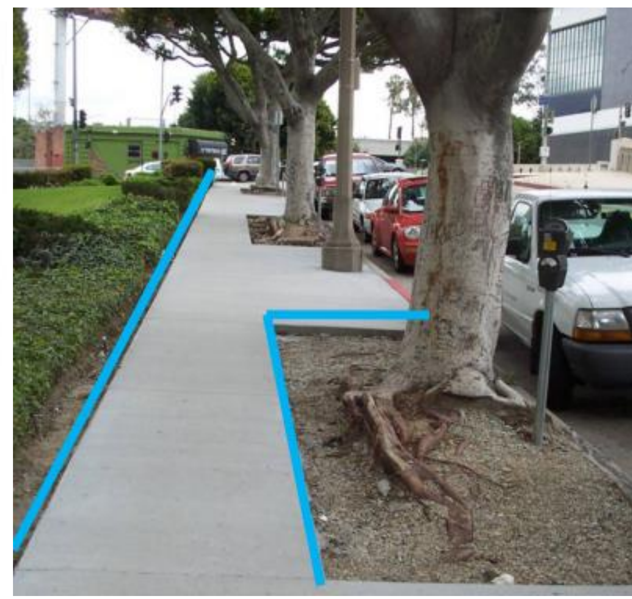
Canny with $\sigma = 2$

Today: how to extract **higher-level** information ("boundaries")

Boundaries are composed by groups of edge pixels

Lines are ubiquitous in man-made scenes





Slide credit: K. Grauman

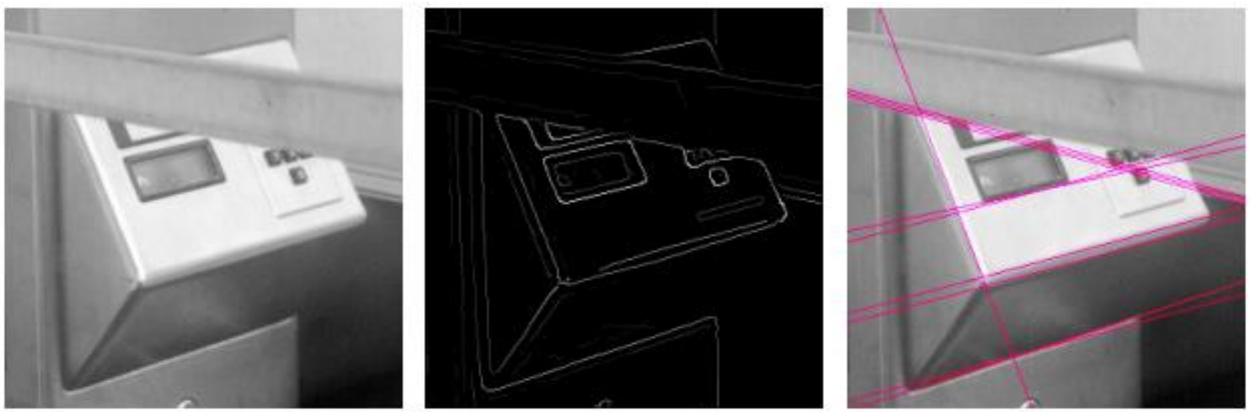
Why is edge detection not enough?

- Usually, more than one line in an image
- Many edge pixels not belonging to any line (clutter)
- Edge pixels belonging to a line might not be perfectly aligned (noise)
- Lines sometimes incomplete

Which pixels go with which line?

What is model fitting?

- Observations: many low-level features
- Model fitting: find (fit) an high-level explanation (model) that well explains the observations



In our case:

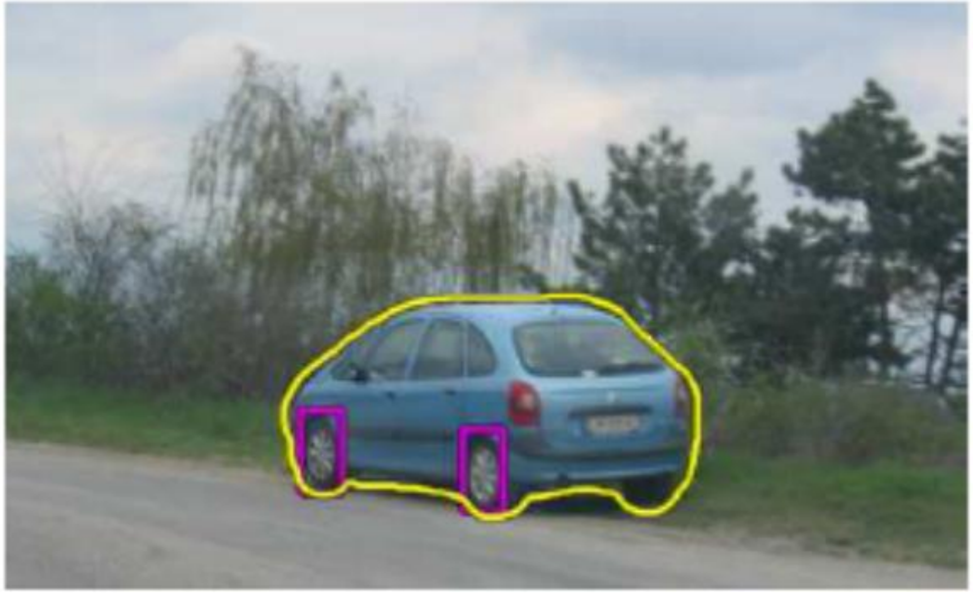
- observations: edges
- model: one or more lines

Slide credit: K. Grauman

Other examples of model fitting

Find edges then identify circles

Find objects with known silhouette



Slide credit: K. Grauman

Voting algorithms
(a general technique)

1. Every **feature** casts votes for all **models** that are compatible with it
2. We choose models that accumulated a lot of votes

What about the votes cast by clutter and noise?

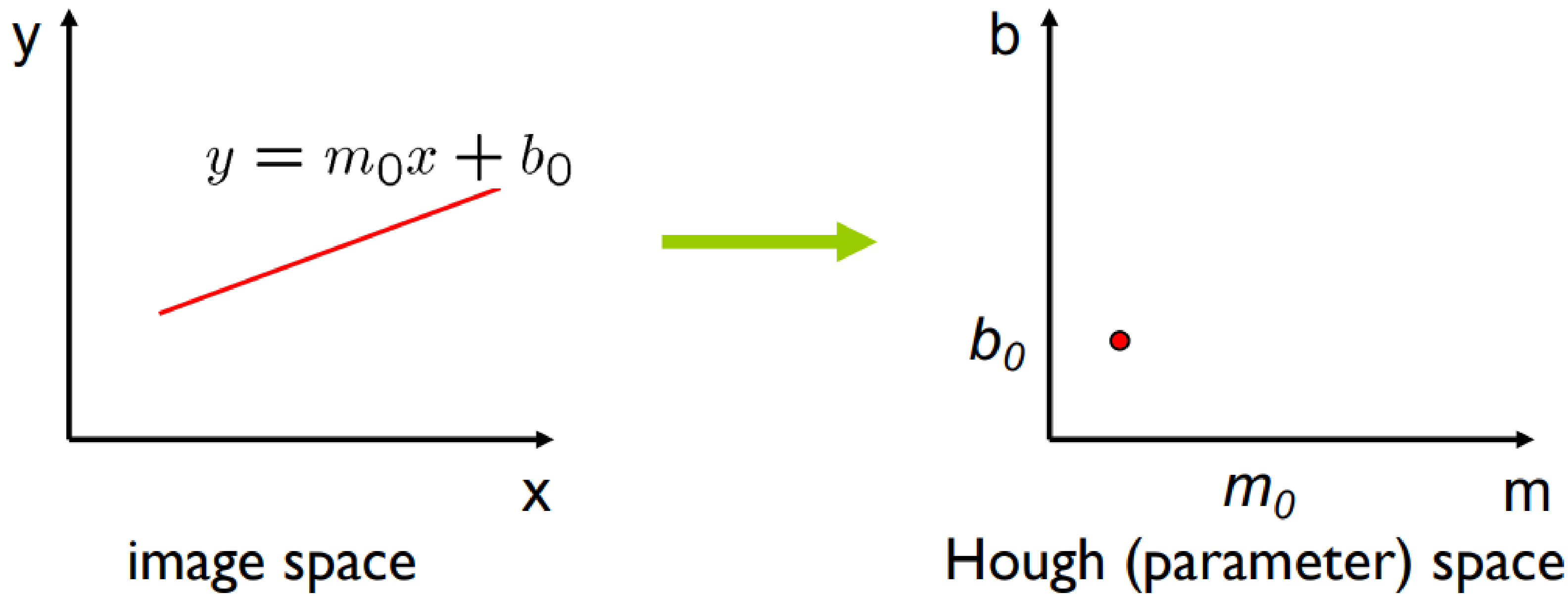
- Their votes might be many but will be **inconsistent**!
- Instead, **features belonging to a model will concentrate a lot of votes for that model**

The Hough Transform: a voting algorithm for finding lines

1. Every **edge point** casts votes for all **lines** that are compatible with it
2. We choose **lines** that accumulated a lot of votes

Part 1: the Hough Algorithm for line fitting

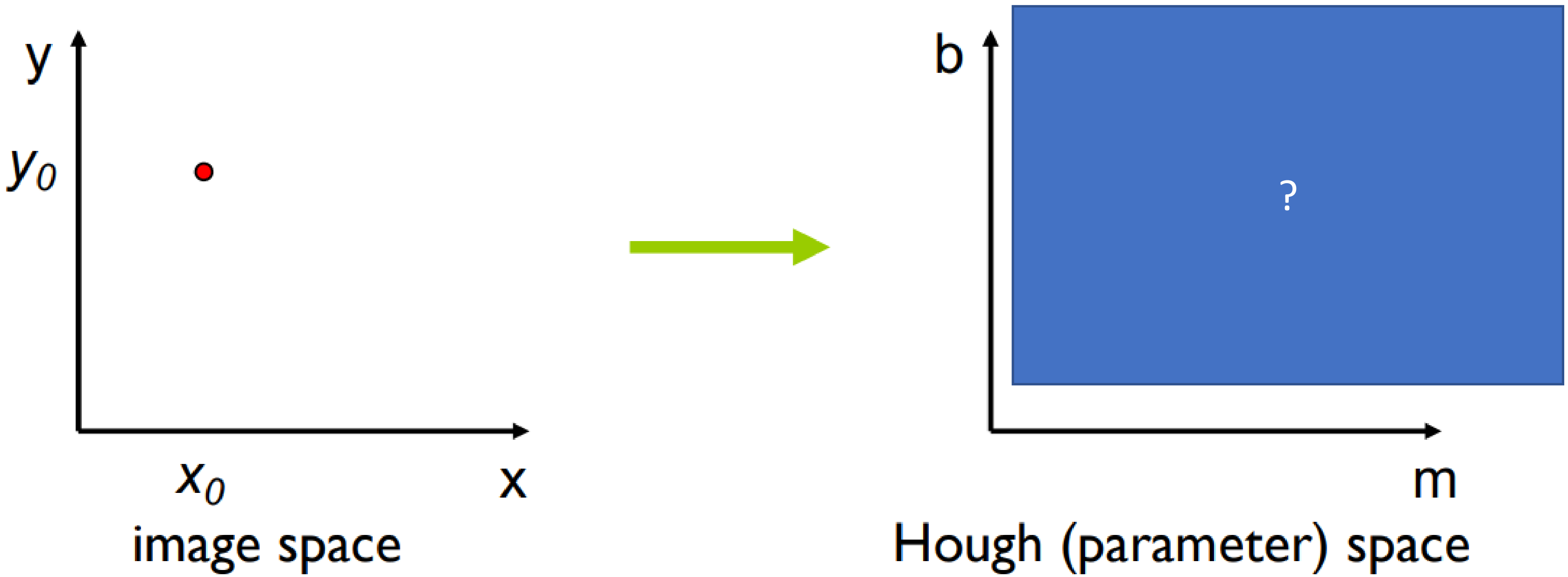
How to represent a line?



A line in the image corresponds to a point in Hough space

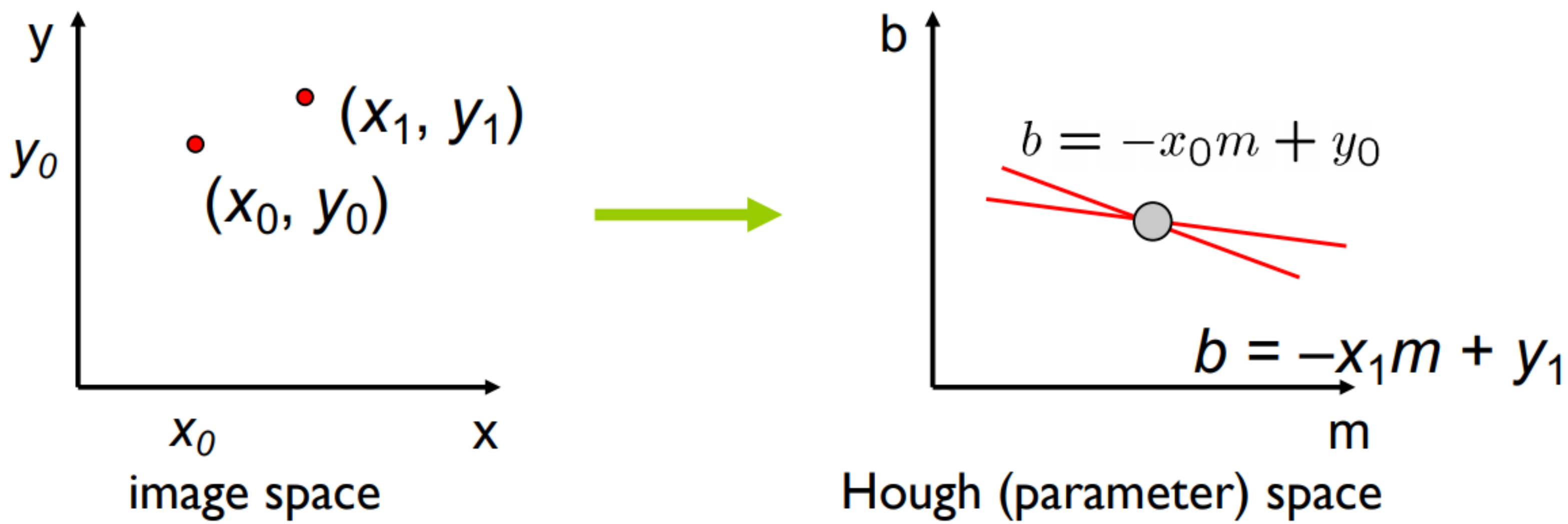
Slide credit: S. Seitz

What does a point correspond to?



A point in the image corresponds to a line in Hough space

Finding a line through two points

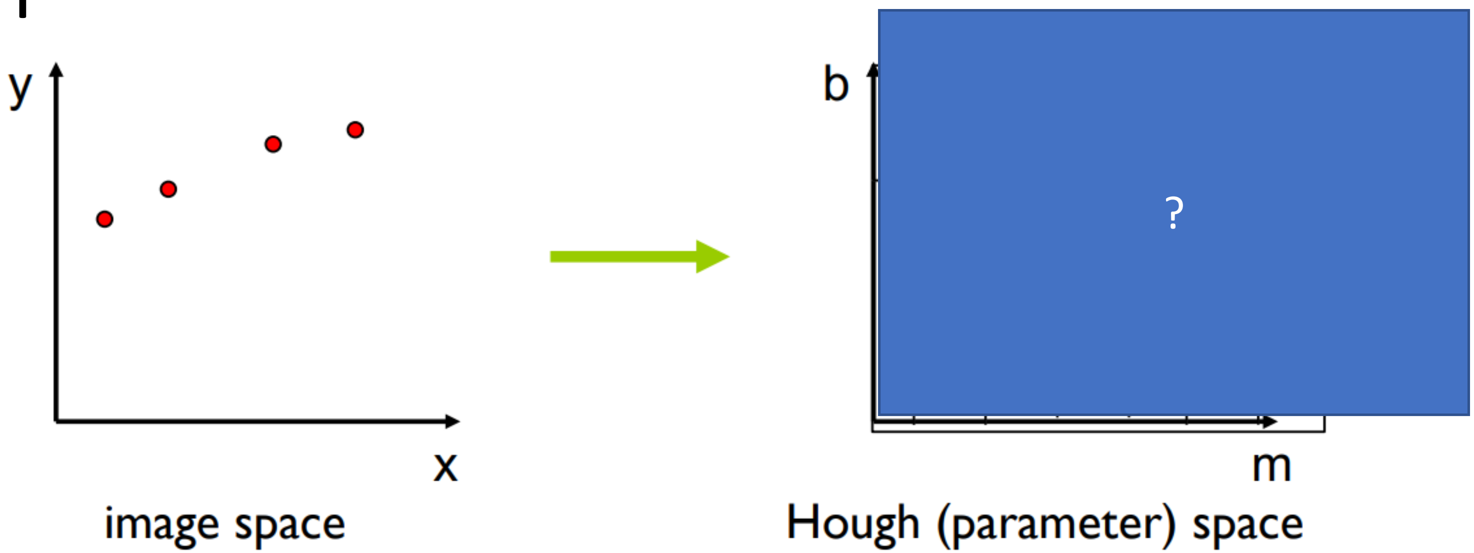


What are the line parameters for the line that contains both (x0, y0) and (x1, y1)?

The (m,b) coordinates of the intersection of the hough-space lines

$b = -x_0 m + y_0$
and
$b = -x_1 m + y_1$

Finding a line through many points



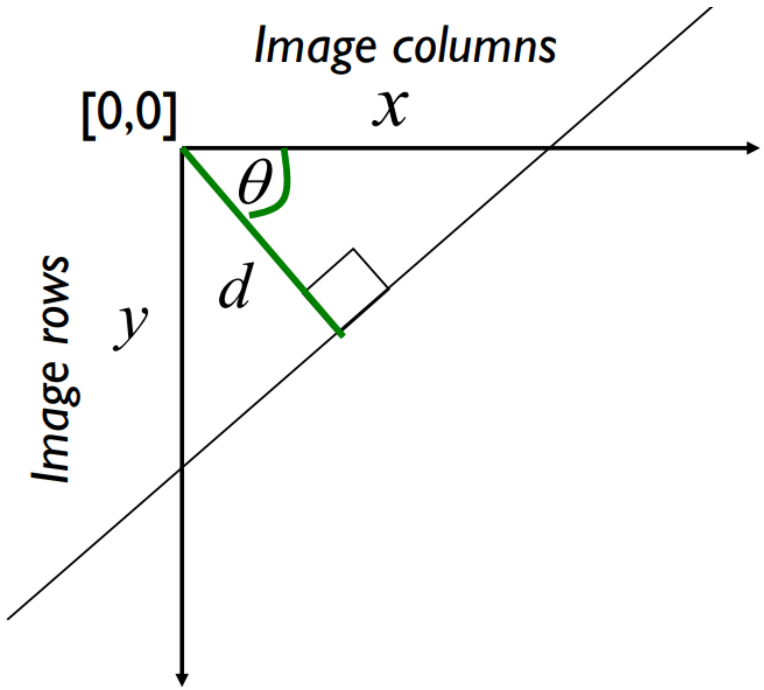
Note that the 4 points are not exactly aligned: in Hough space, the lines corresponding to each point will almost intersect.

Now, let each edge point in image space vote for a **set of possible parameters** in Hough space.

Accumulate votes in a discrete set of bins; parameters with the most votes indicate a line in image space.

Where is the problem with this approach?

Solution: represent lines in polar coordinates



d : perpendicular distance from line to origin

θ : angle the perpendicular makes with the x-axis

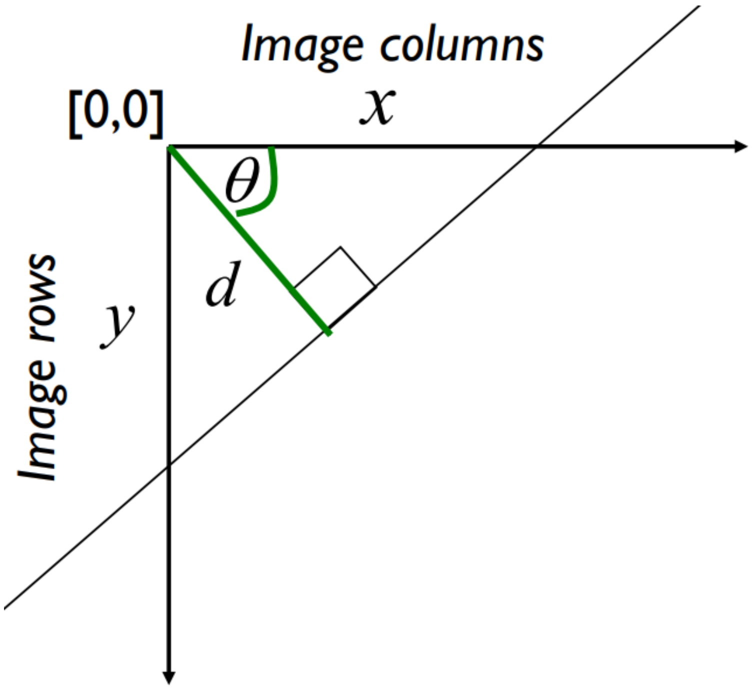
$$x\cos\theta - y\sin\theta = d$$

The hough space is now (d,θ) instead of (m,b)

Same as before: a line in image space is a point in Hough space.

Slide credit: K. Grauman

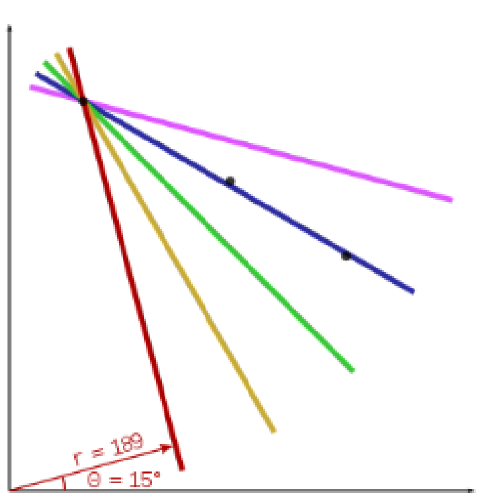
Pause and ponder

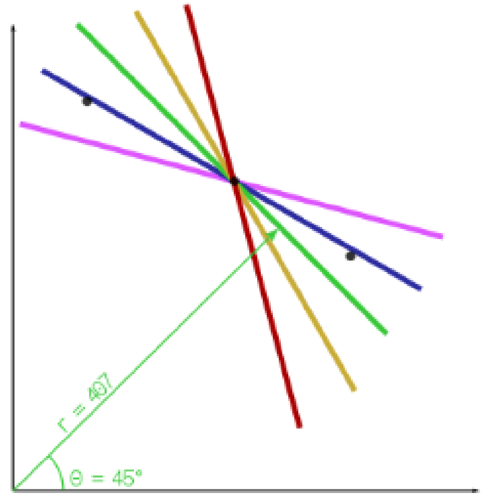


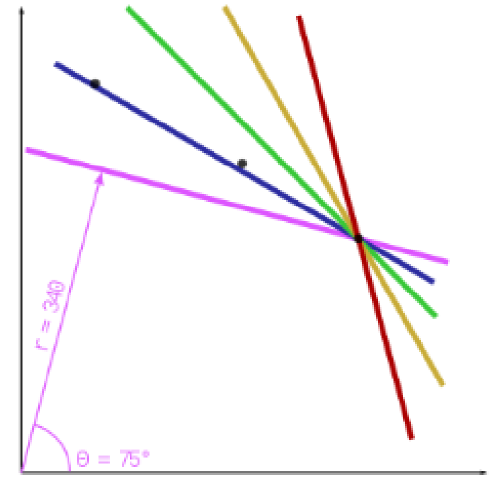
- For an image of size (width, height). What are the boundaries of the Hough space such that we can represent any line overlapping the image?
- Given a point in image space, what is the corresponding shape in Hough space?

(solution: a sinusoid)

An example







θ	r
15	189.0
30	282.0
45	355.7
60	407.3
75	429.4

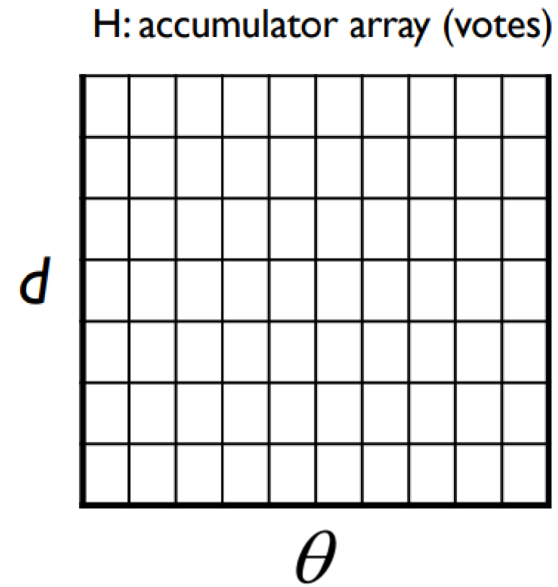
θ	r
15	318.5
30	376.8
45	407.3
60	409.8
75	385.3

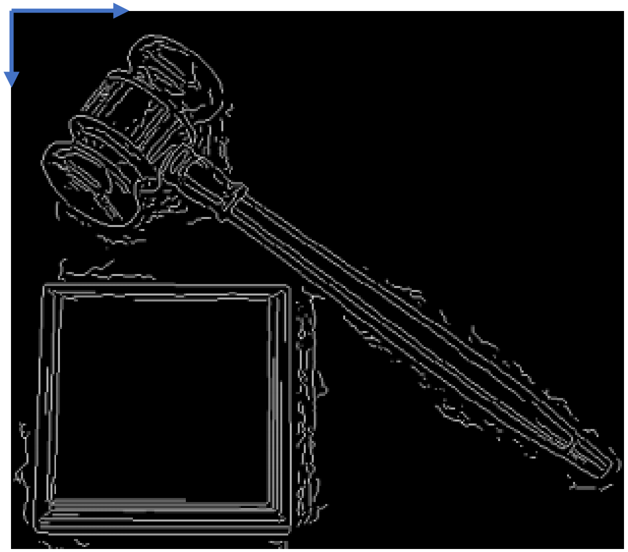
θ	r
15	419.0
30	443.6
45	438.4
60	402.9
75	340.1

Given 3 points in image space, here are some of the corresponding points in hough space

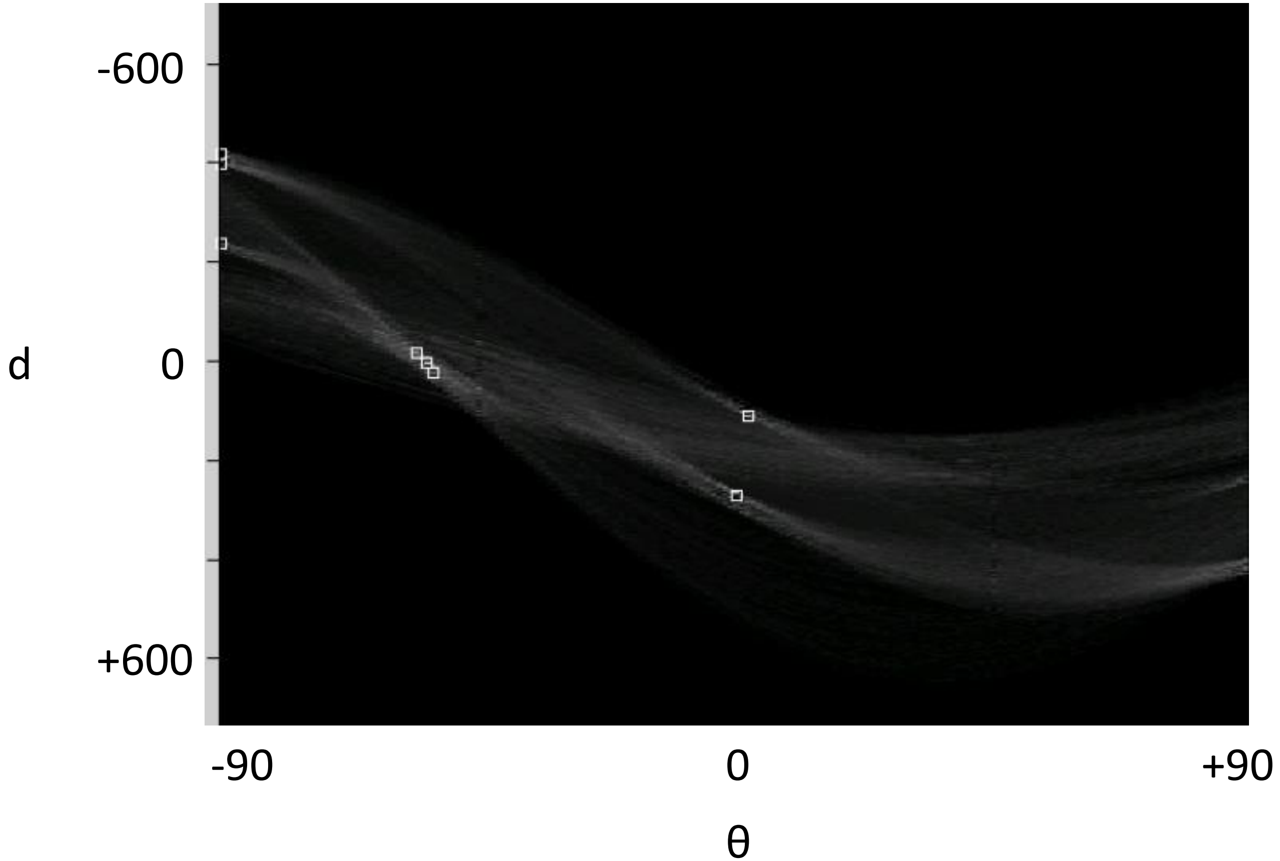
Hough Transform: the algorithm

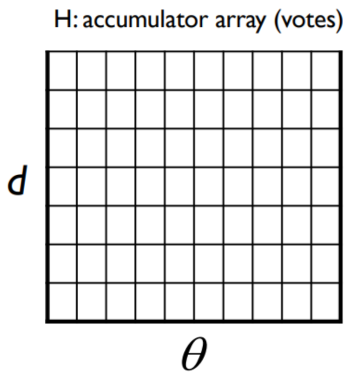
```
Initialize H[d,θ]=0

for each edge point (x,y) in the image:

  for θ in range(θmin,θmax):

    d = x cos(θ) - y sin(θ)

    H[d,θ]+=1

Find the value(s) of (d,θ) where H[d,θ] is
maximum

The detected line in the image is given by
d = x cos(θ) - y sin(θ)
```





Quiz: identify which maxima represent the four lines of the square object!

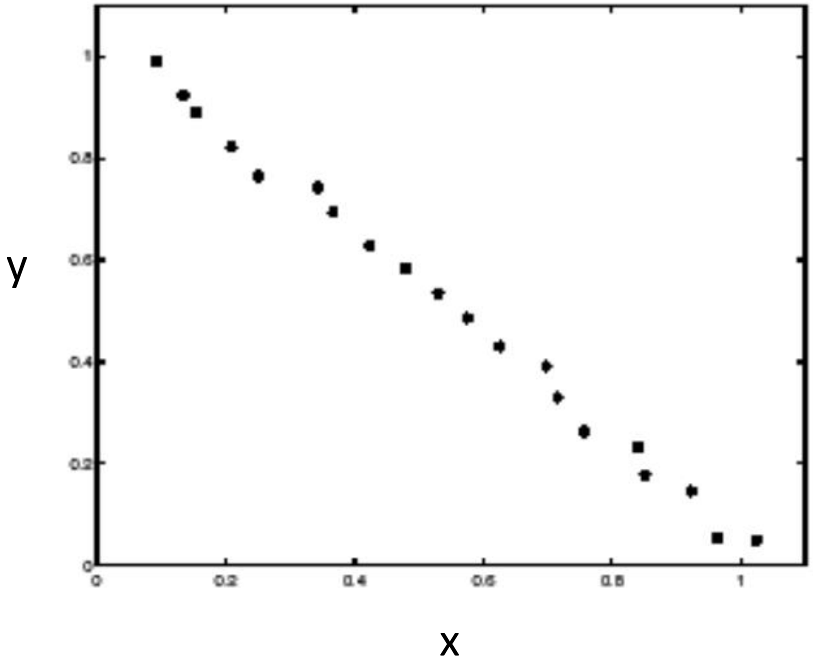




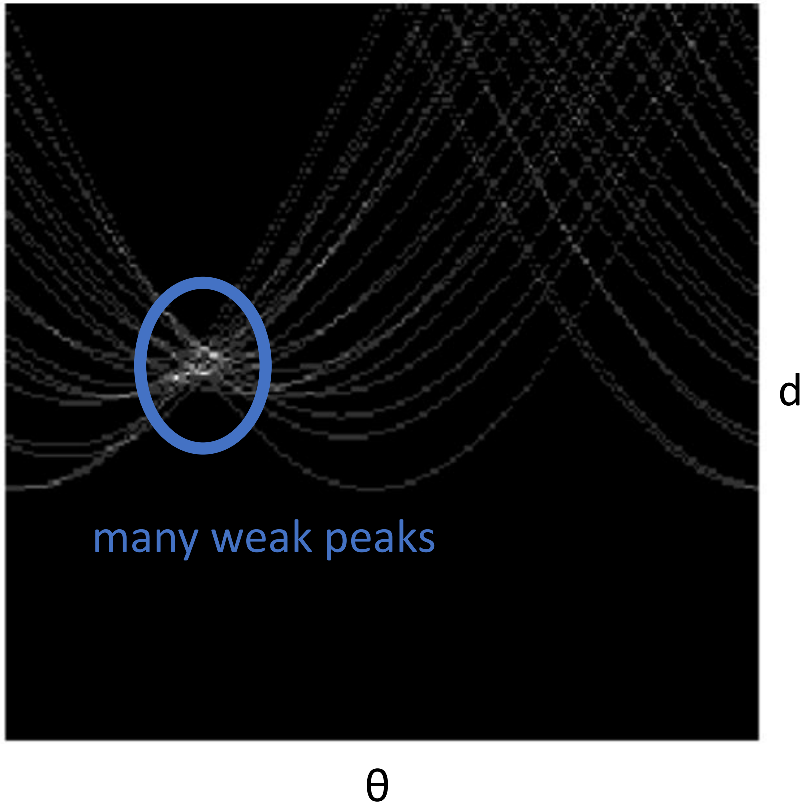
Slide credit: K. Grauman

What’s the effect of noise?

Image space



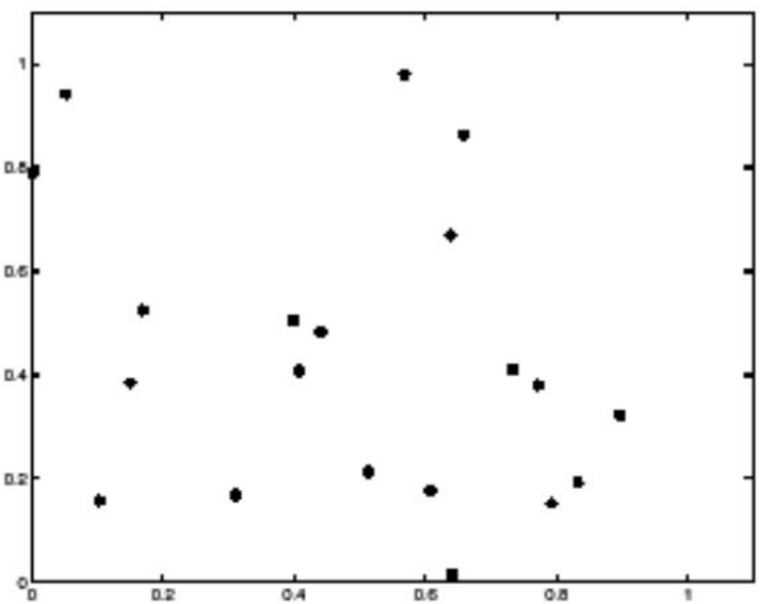
H: accumulator array



Slide credit: K. Grauman

What’s the effect of noise?

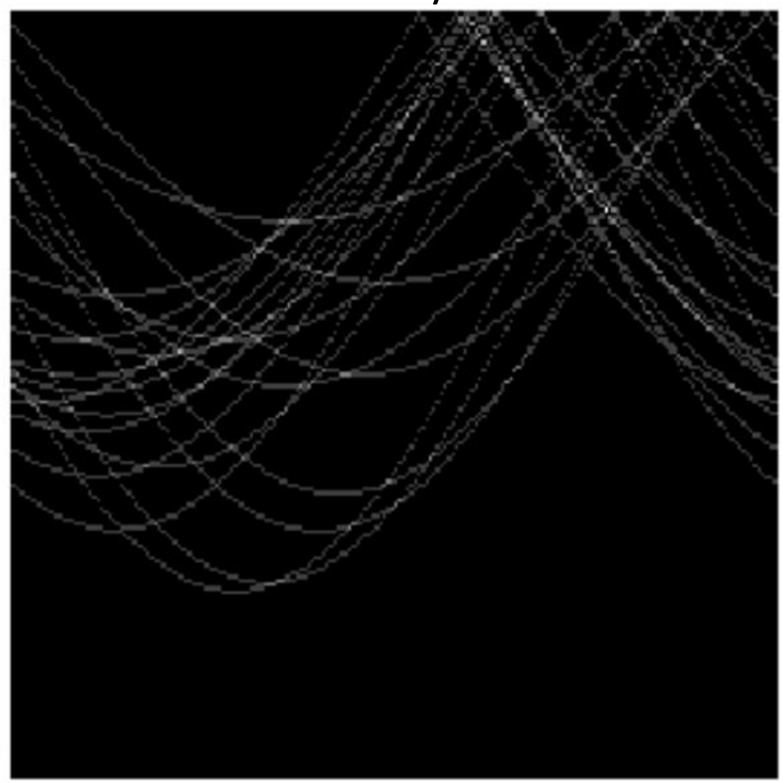
Image space



y

x

H: accumulator array



d

θ

Bin size in the accumulator: an important parameter

How large are the bins in the accumulator?

- Too small:
  - many weak peaks due to noise
- Just right:
  - one strong peak per line, despite noise
- Too large:
  - poor accuracy in locating the line
  - many votes from clutter might end up in the same bin

A solution:

keep bin size small but also vote for neighbors in the accumulator

(this is the same as “smoothing” the accumulator image)

Extension 1

From the edge detection algorithm, we know the direction of the gradient for each edge pixel

- Remember how that was related to edge direction?

  Edge direction is orthogonal to gradient direction

- We can make sure an edge pixel only votes for lines that have (almost) the direction of the edge!

→reduces the computation time

→reduces the number of useless votes (better visibility of maxima corresponding to real lines)

Extension 2

Keep track of edge points that voted for each line.

If a line is chosen, look for groups of edge points that voted for that line → find start point and end point of segment.

Part 2: the Hough algorithm for circle detection

The Hough Transform: a voting algorithm…

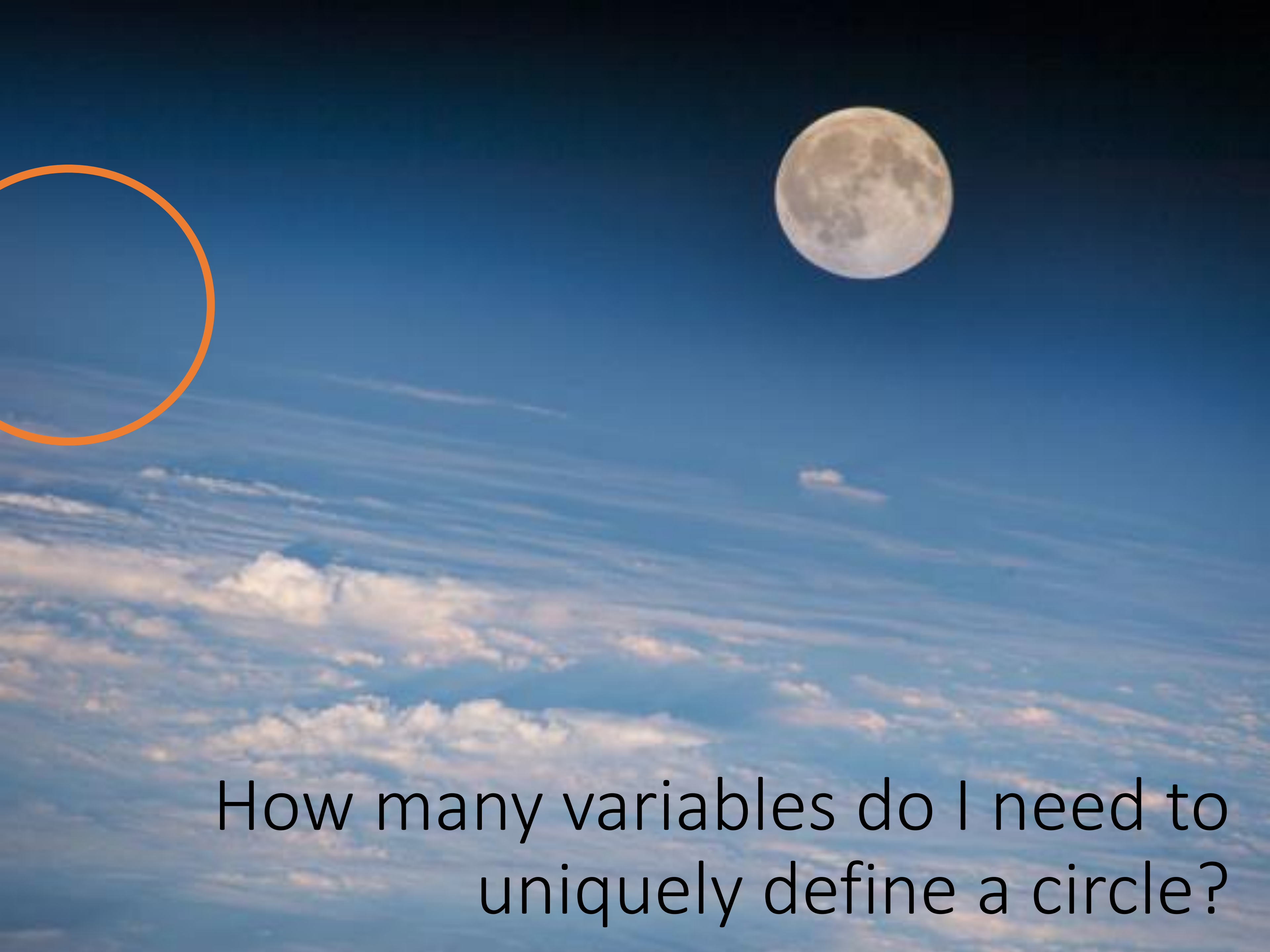
1. Every **feature** casts votes for all **models** that are compatible with it
2. We choose **models** that accumulated a lot of votes

The Hough Transform: a voting algorithm… for finding **lines**

1. Every **edge point** casts votes for all **lines** that are compatible with it
2. We choose **lines** that accumulated a lot of votes

The Hough Transform: a voting algorithm… for finding **circles**

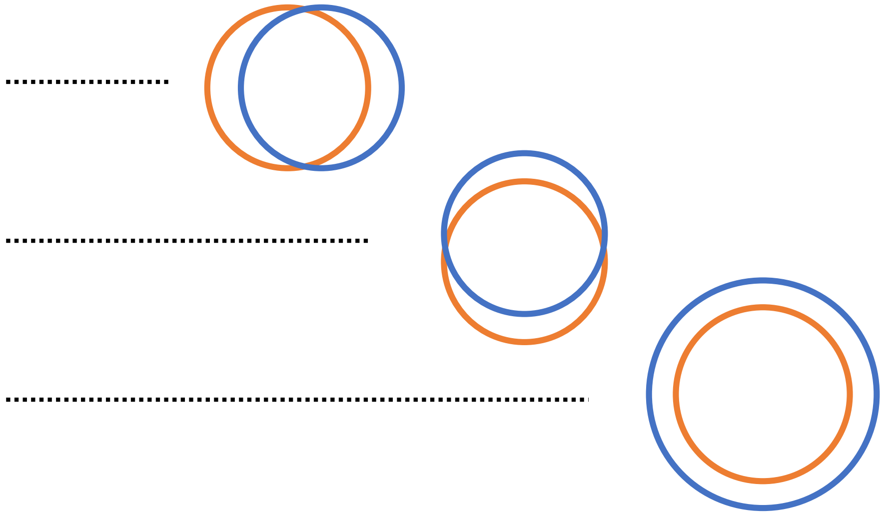
1. Every **edge point** casts votes for all **circles** that are compatible with it
2. We choose **circles** that accumulated a lot of votes


How many variables do I need to uniquely define a circle?

3 parameters uniquely define a circle

→ You use a 3-dimensional Hough Space

- Center x
- Center y
- Radius



How do we parametrize circles?

$$(x_i - a)^2 + (y_i - b)^2 = r^2$$

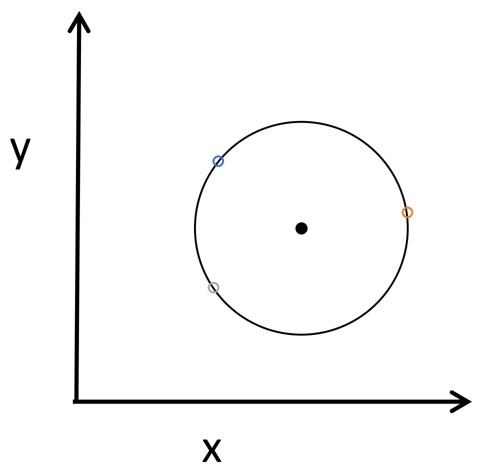
- Center (x=**a**,y=**b**) and radius **r**

  3 degrees of freedom (a, b and r)

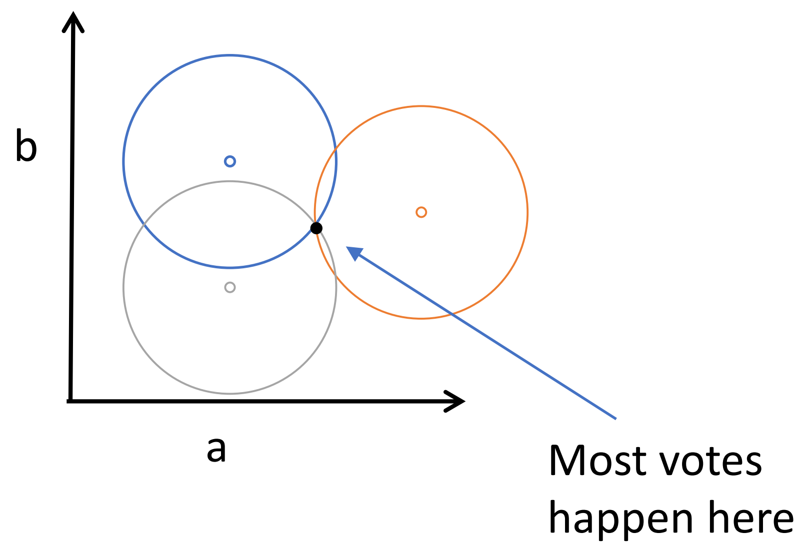
- **If we assume known radius (fix r)**
  - Hough space is 2D:
    - a: x coordinate of circle center
    - b: y coordinate of circle center
  - One point in image space maps to...
    a circle in hough space

Hough space for circles with **known** radius

Image space

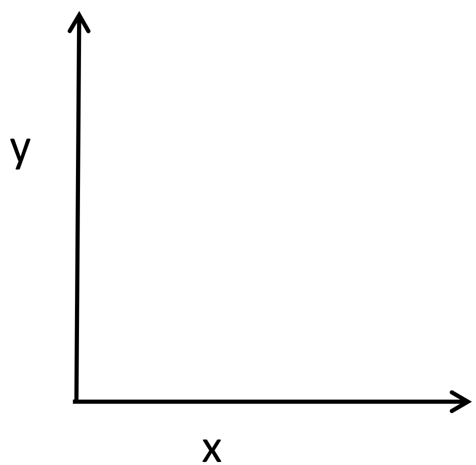


Hough space (2D)

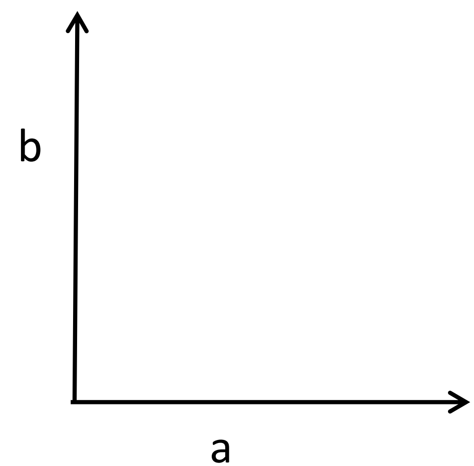


Hough space for circles with **unknown** radius

Image space

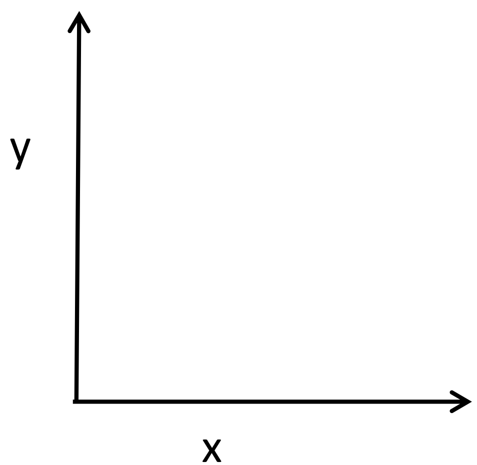


Hough space

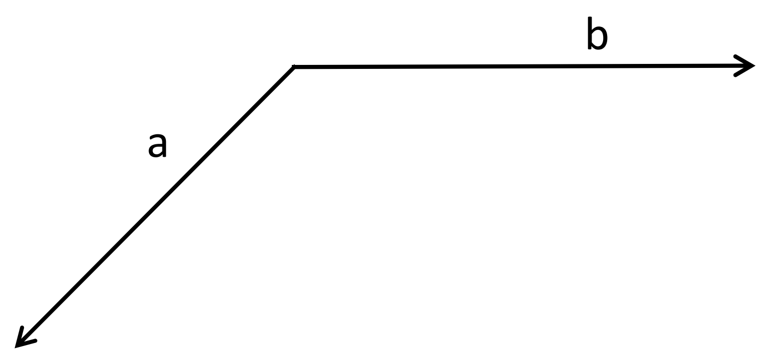


Hough space for circles with unknown radius

Image space

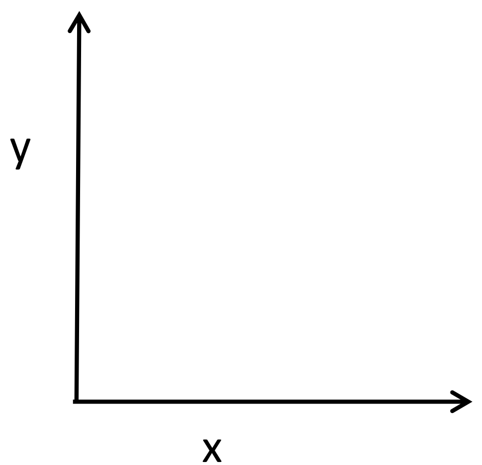


Hough space

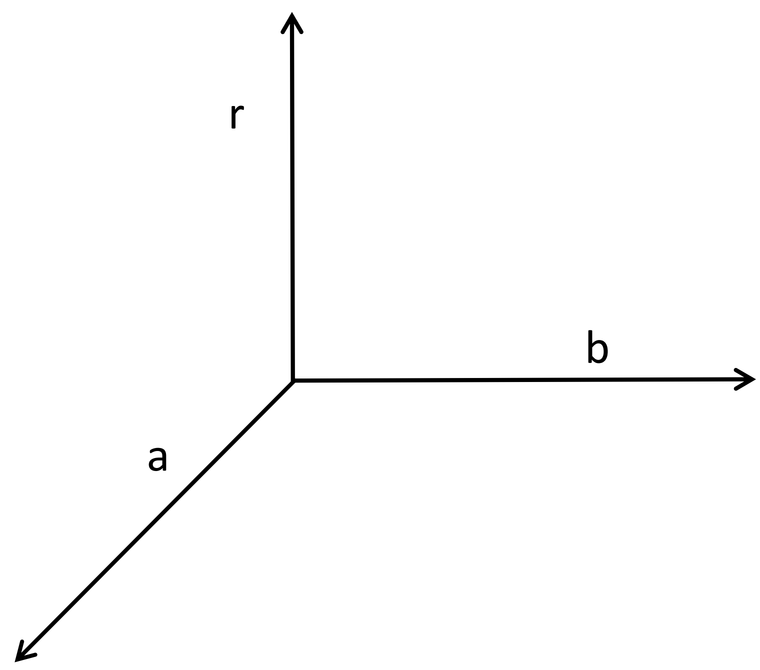


Hough space for circles with unknown radius

Image space

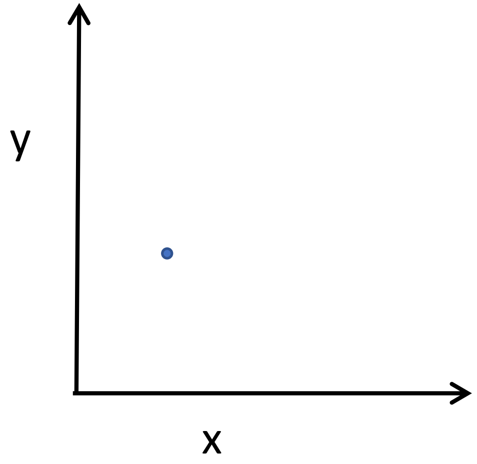


Hough space (3D)

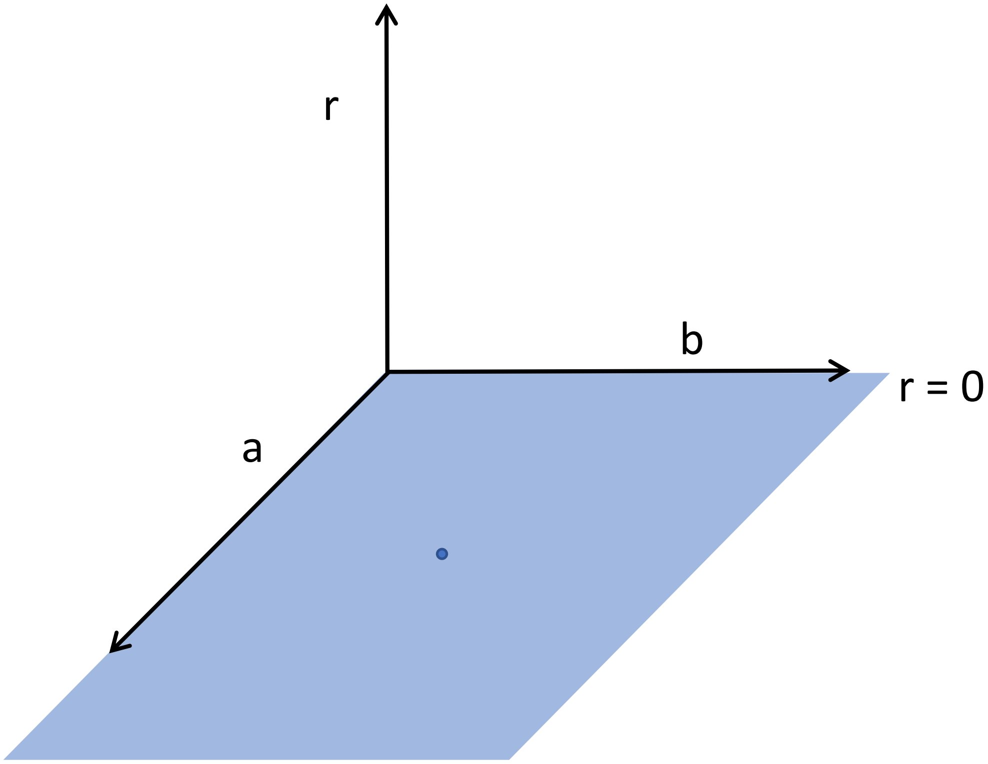


Hough space for circles with unknown radius

Image space

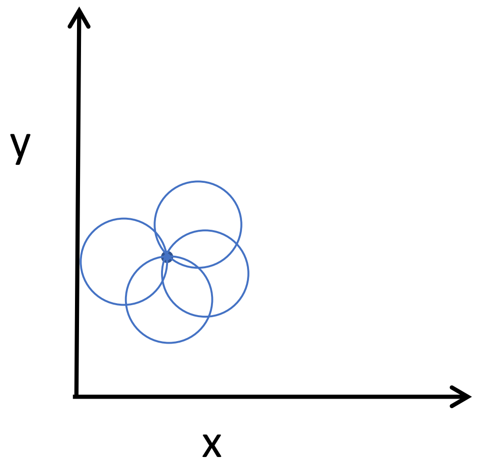


Hough space

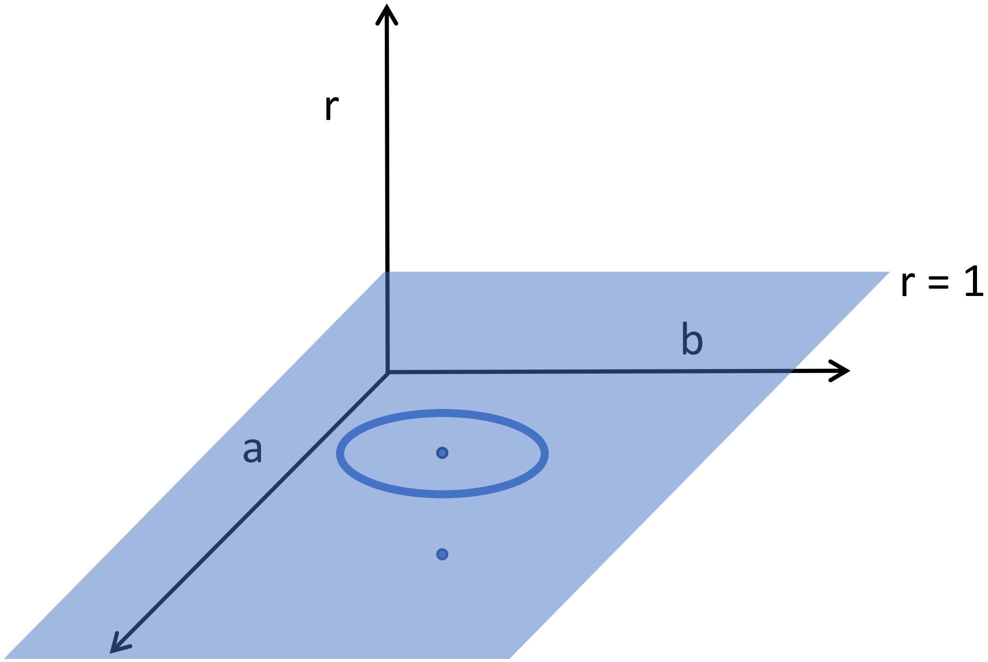


Hough space for circles with unknown radius

Image space

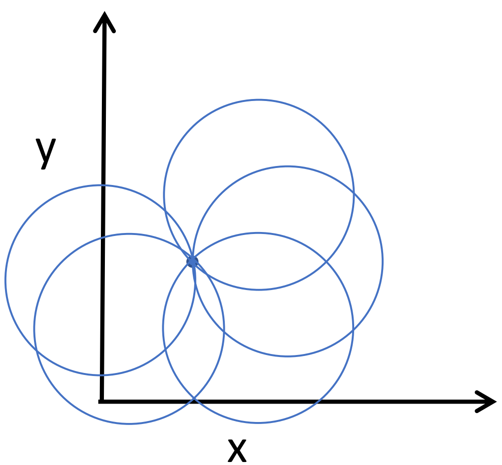


Hough space

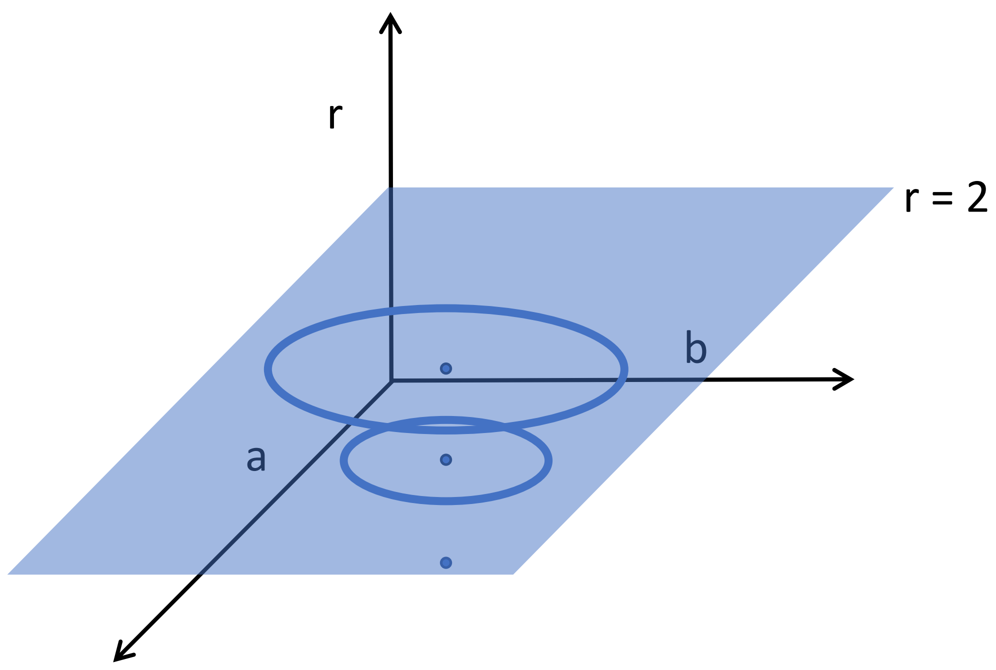


Hough space for circles with unknown radius

Image space

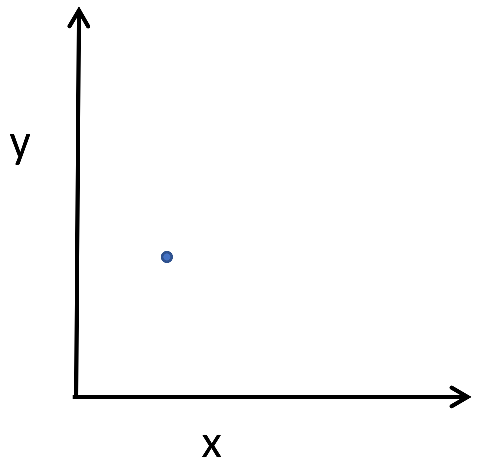


Hough space

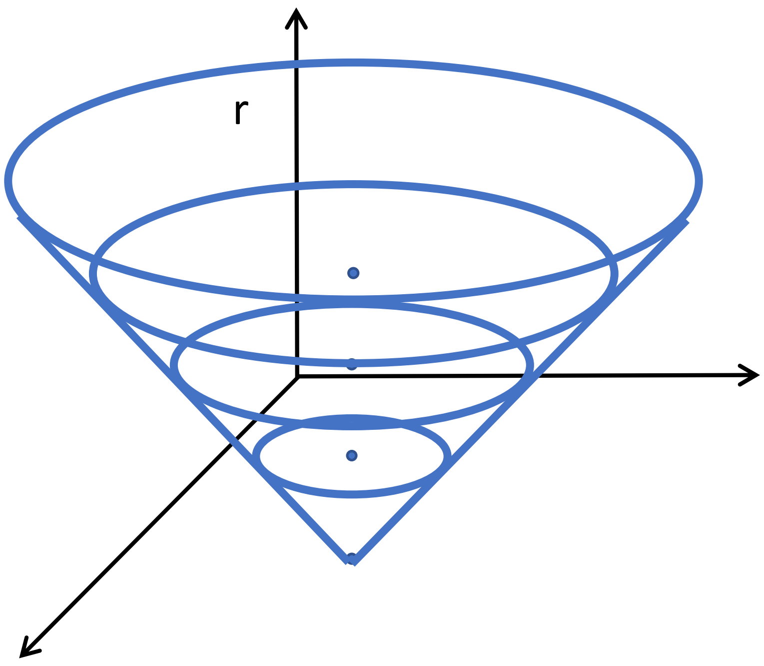


Hough space for circles with unknown radius

Image space

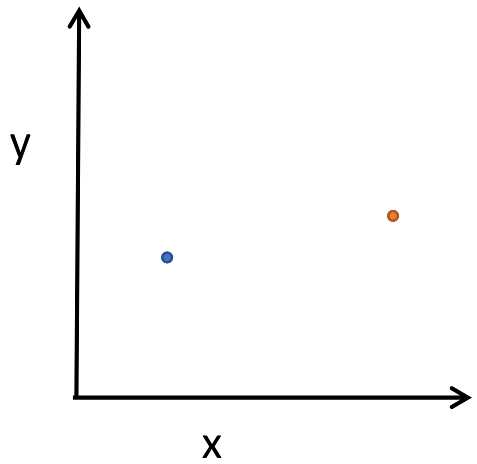


Hough space

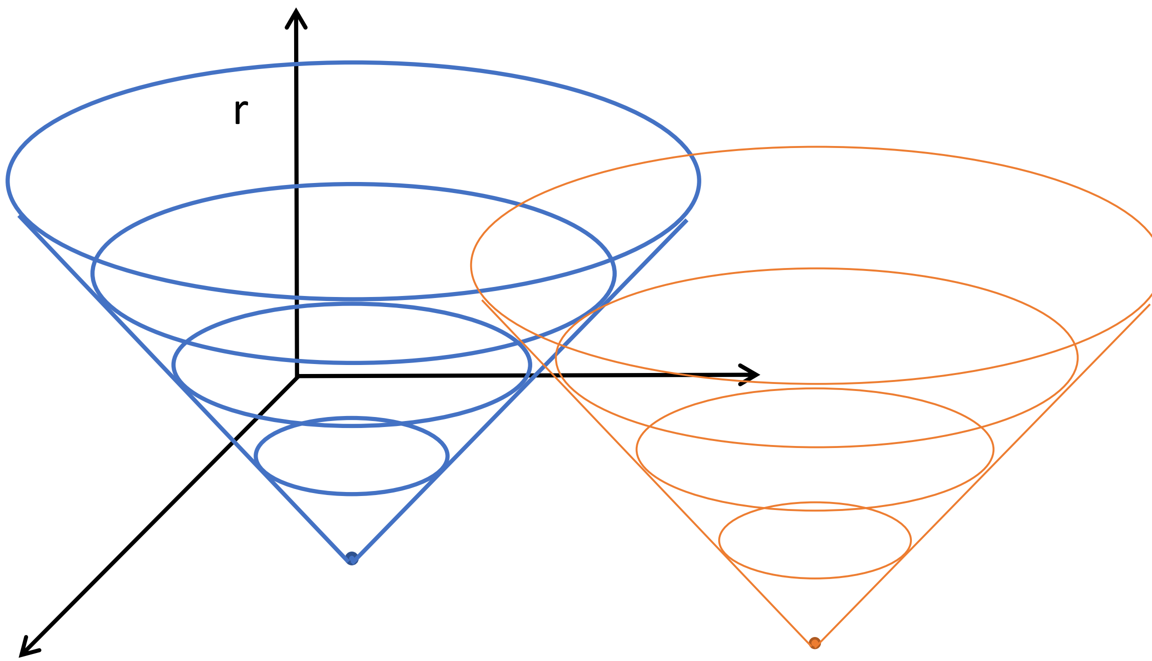


Hough space for circles with unknown radius

Image space

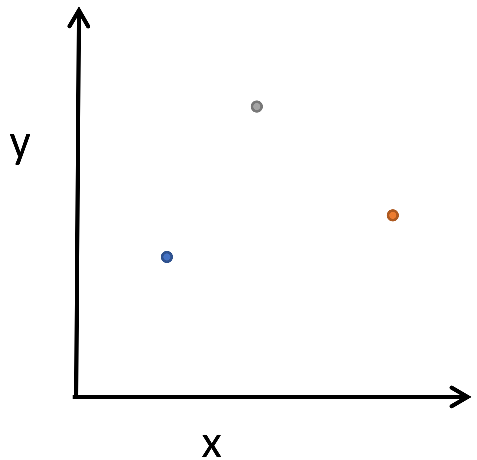


Hough space

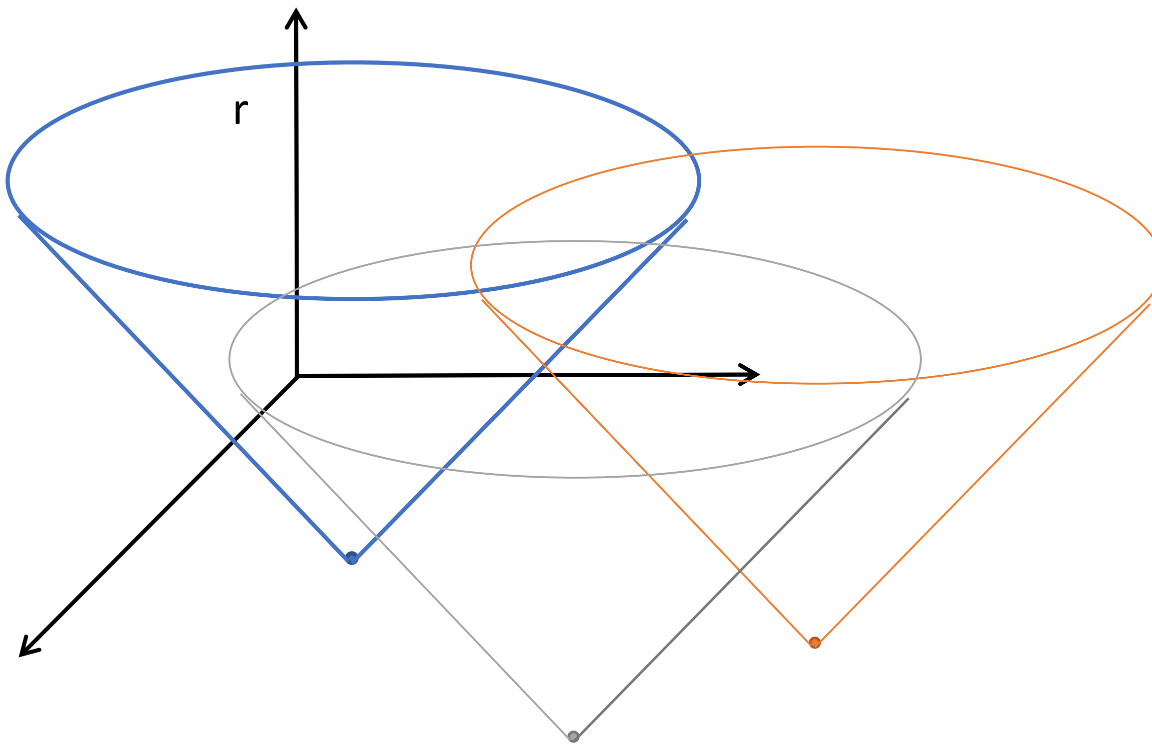


Hough space for circles with unknown radius

Image space



Hough space



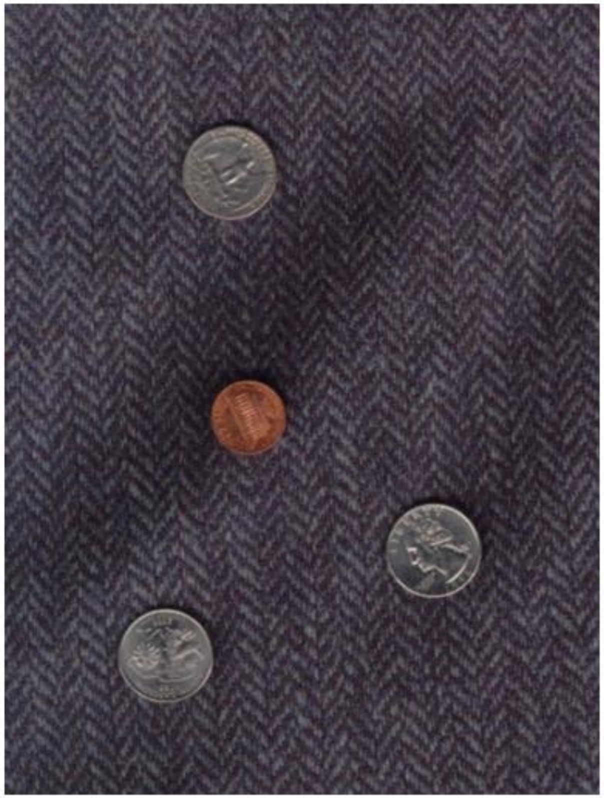
Algorithm

```
Initialize H accumulator to zeros
For every edge pixel (x,y):
  For each possible radius value r:
    For each possible direction θ:
      a = x – r cos(θ) // column
      b = y + r sin(θ) // row
      H[a,b,r] += 1
```

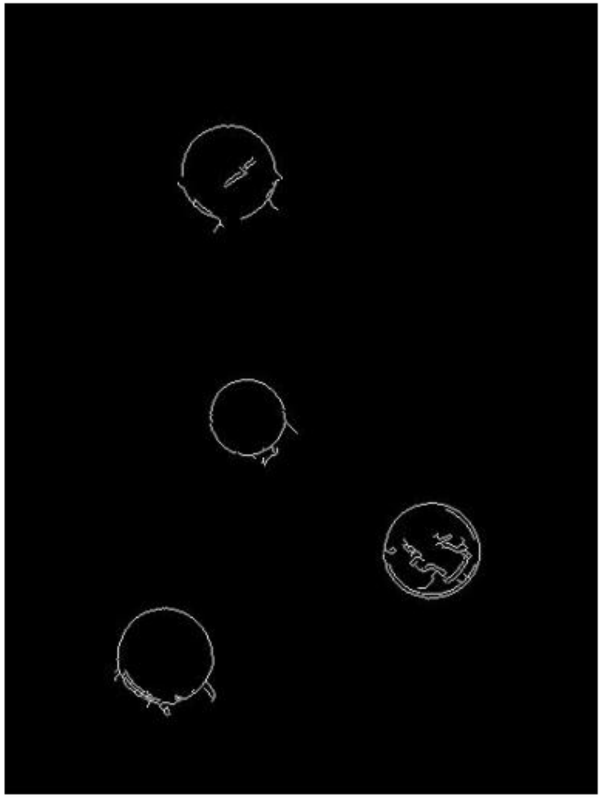
An example

Accumulator for radius equal to radius of a penny

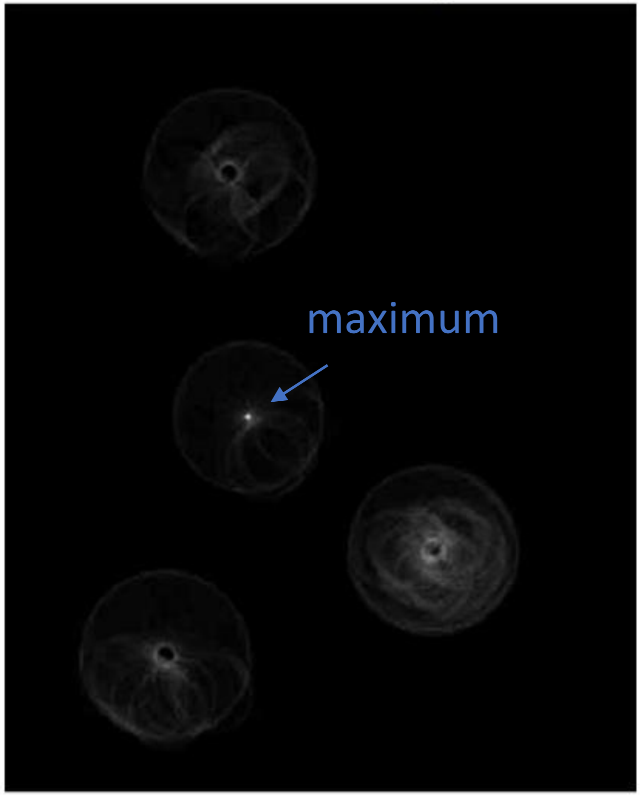
Image



Edges



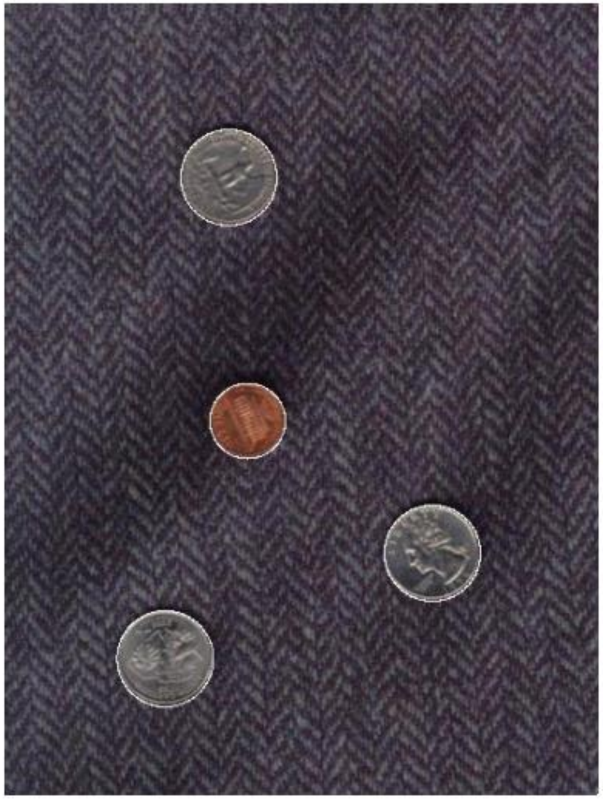
Accumulator for radius=penny



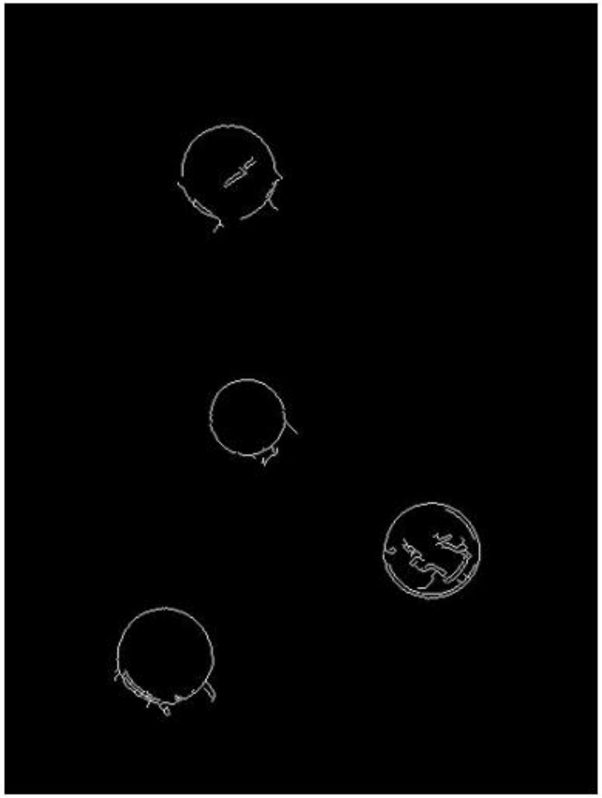
An example

Accumulator for radius equal to radius of a quarter

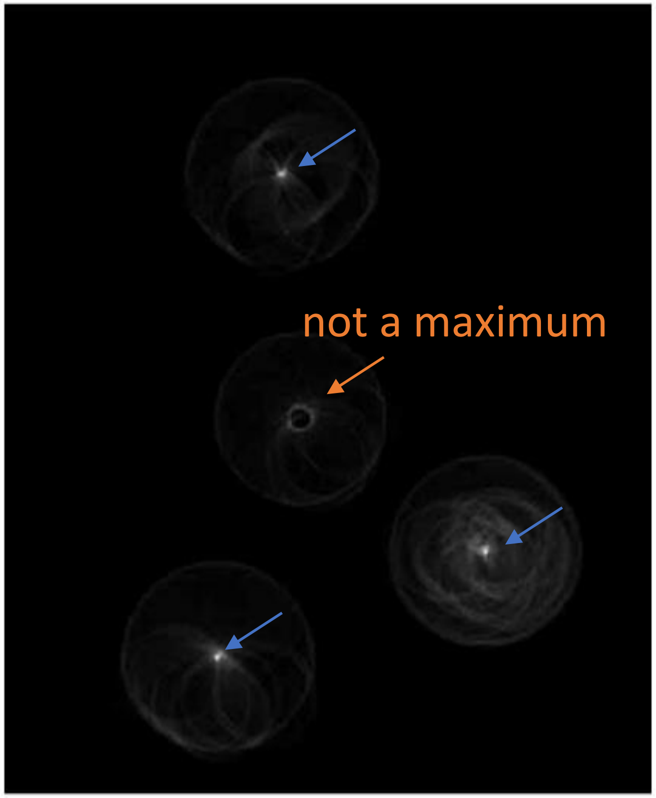
Image



Edges



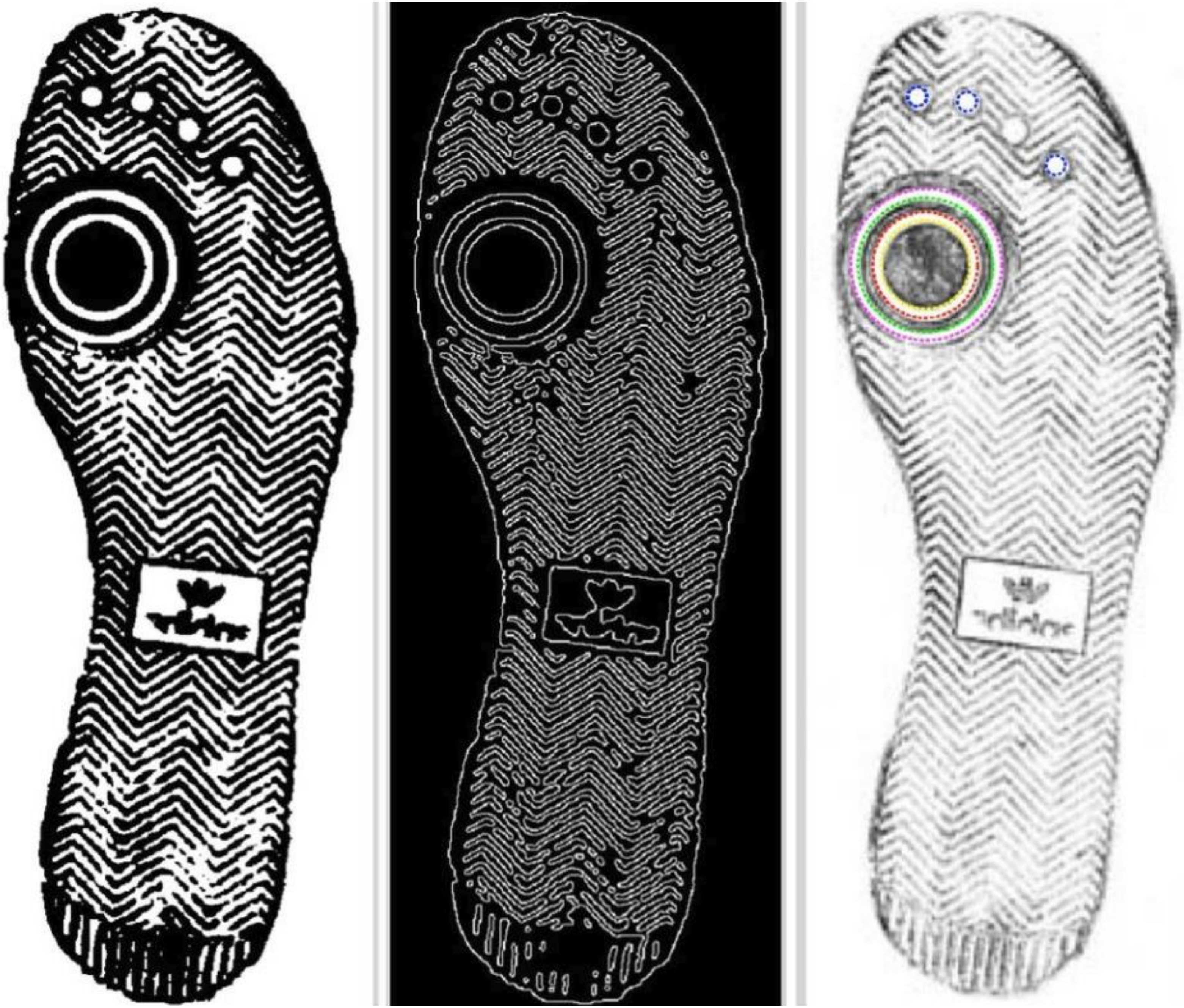
Accumulator for radius=quarter



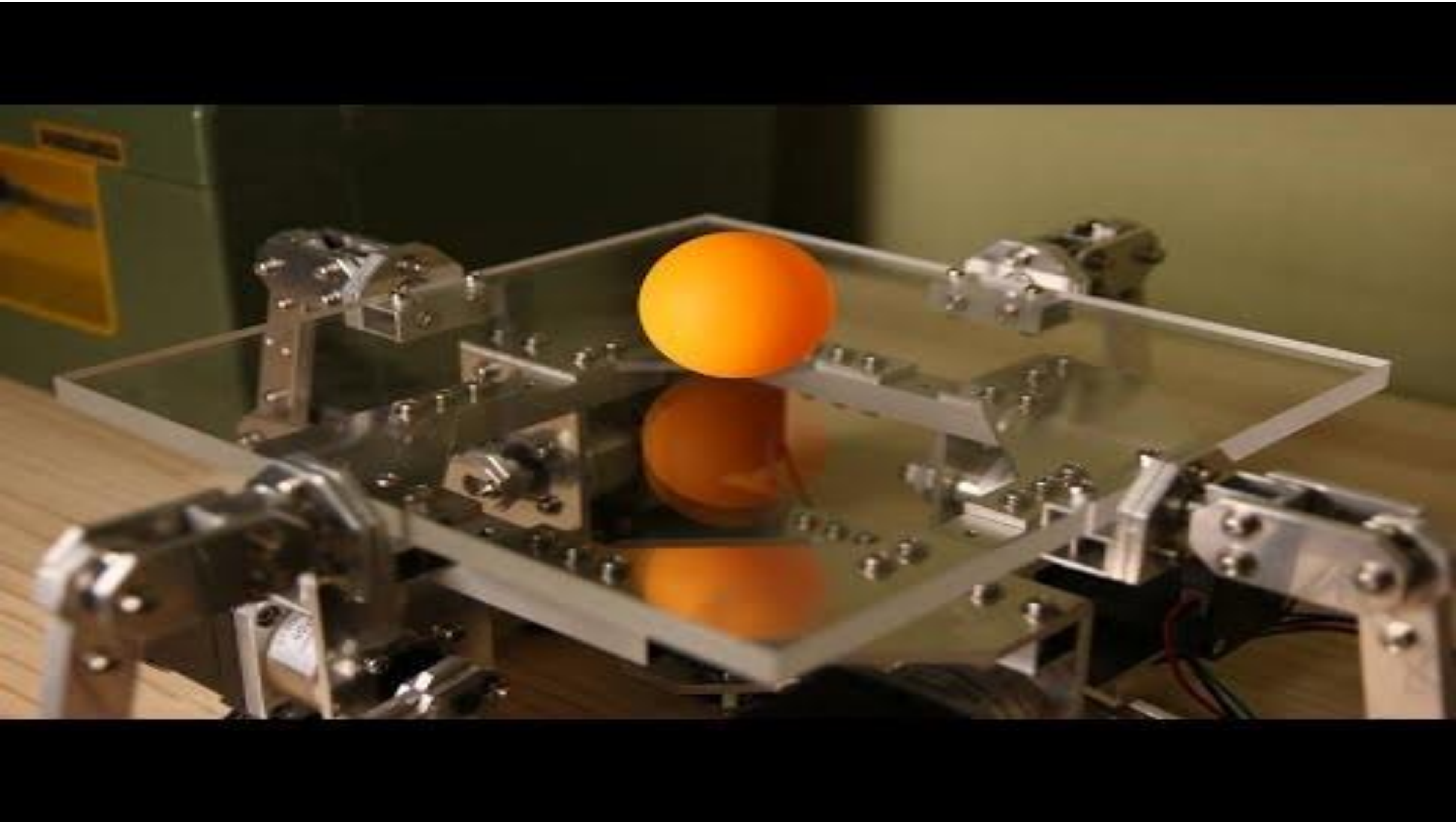
Iris detection

Gradient+threshold | Hough space (fixed radius) | Max detections

Hemerson Pistori and Eduardo Rocha Costa http://rsbweb.nih.gov/ij/plugins/hough-circles.html



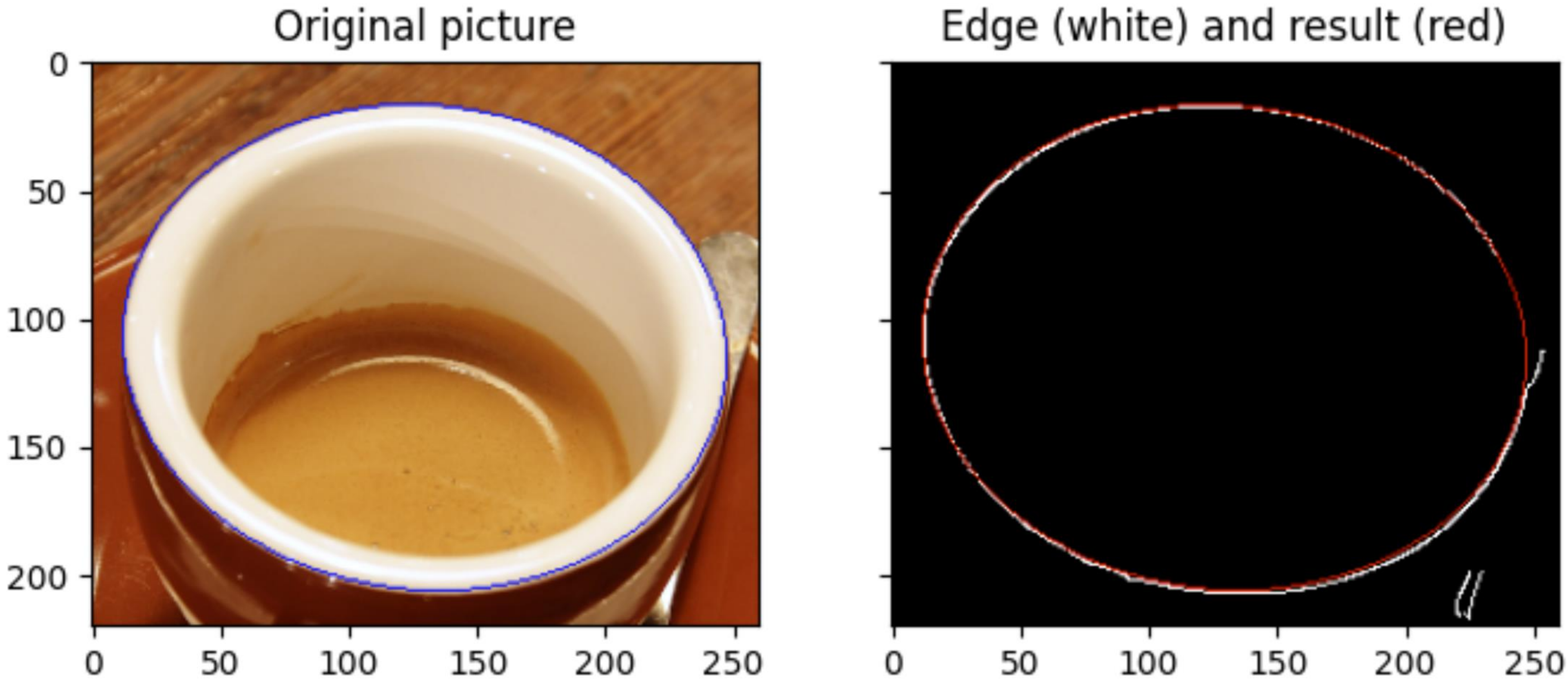
An application to robotics



https://www.youtube.com/watch?v=lYyAMDYzJQM

Part 3: Extensions and conclusions

Detecting ellipses



https://scikit-image.org/docs/dev/auto_examples/edges/plot_circular_elliptical_hough_transform.html

Quiz

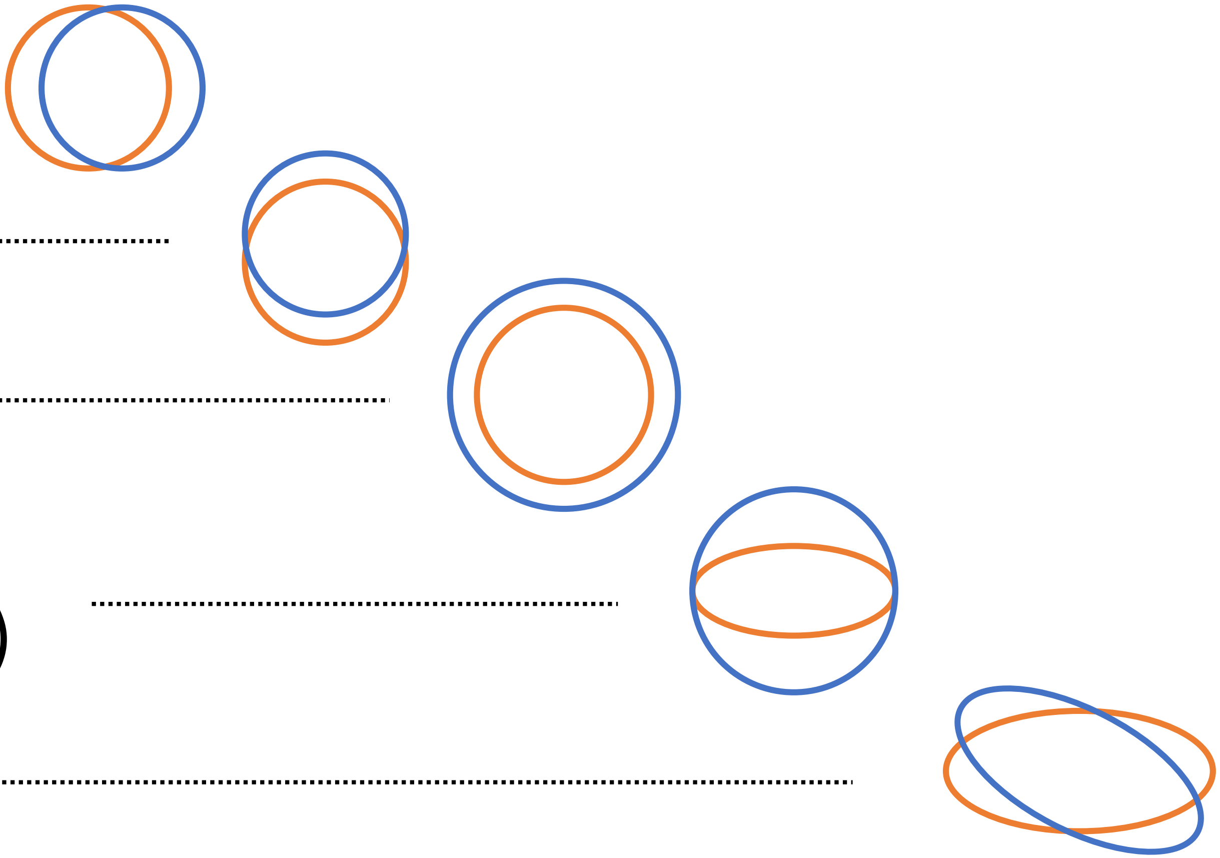
How many parameters are needed to define a generic ellipse in 2D?

Try to fit an ellipse in powerpoint

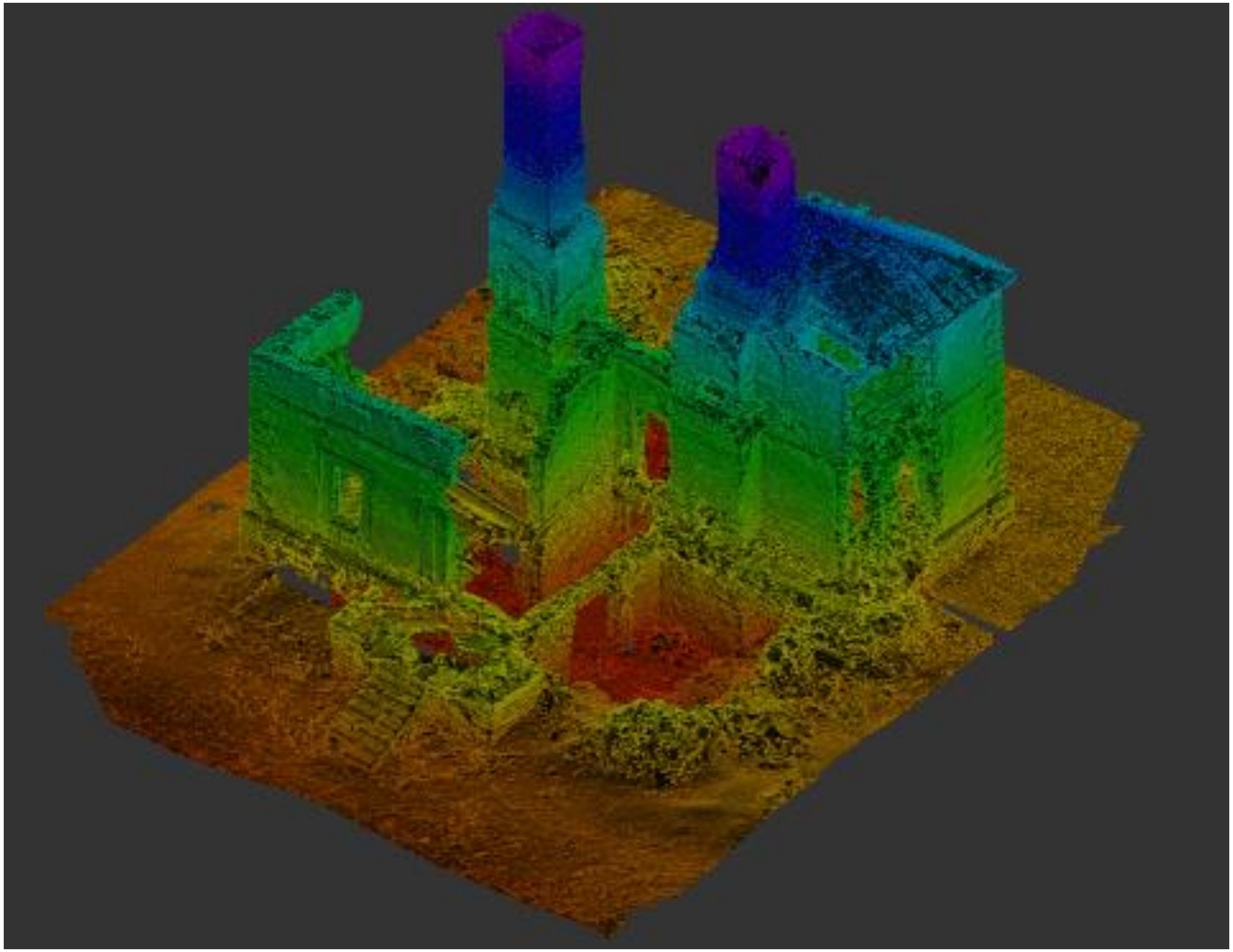
5 parameters uniquely define an ellipse

→ You use a 5-dimensional Hough Space

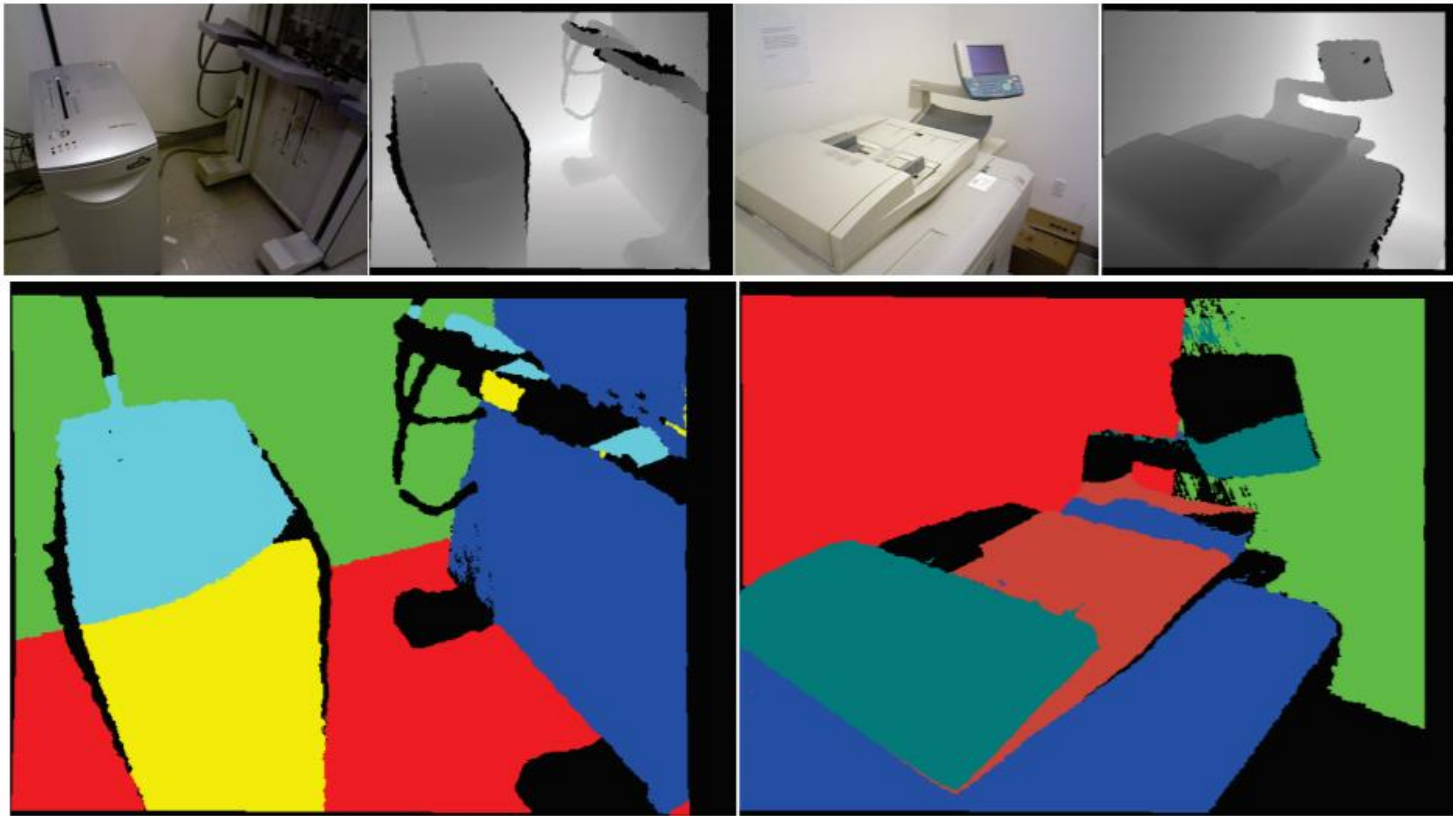
- Center x
- Center y
- Radius
- Eccentricity (how squished?)
- Angle



Finding 3D planes in point clouds



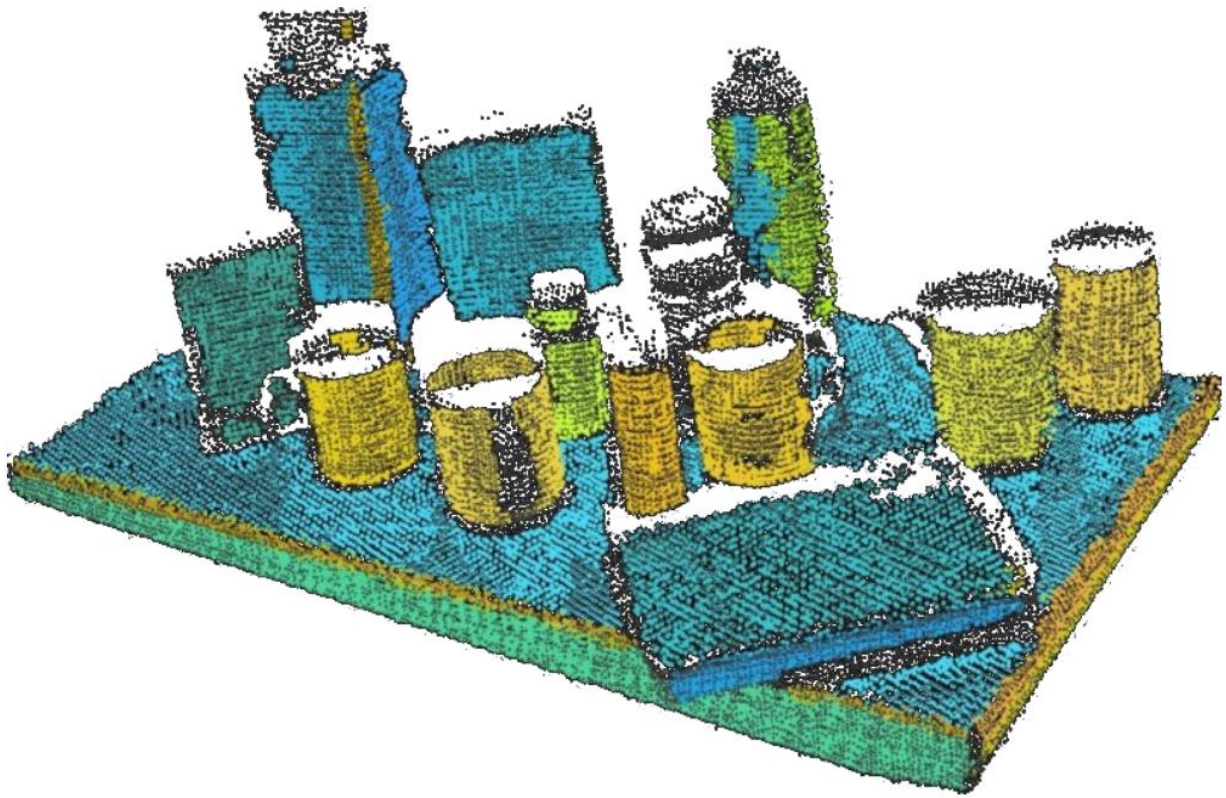
Finding 3D planes in point clouds



https://www.visgraf.impa.br/Data/RefBib/PS_PDF/prl2018/Vera2018.pdf

Finding planes and cylinders in point cloud data

How many parameters are needed to define a plane in 3D?



Voting Algorithms for Model Fitting - Conclusions

A family of powerful algorithms based on a simple concept

Advantages

- All points are processed independently, so **the algorithm can cope with occlusions and gaps**
- Voting algorithms are **robust to clutter**, because points not corresponding to any model are unlikely to contribute consistently to any single bin
- Can detect **multiple instances of a model** in a single pass

Disadvantages

- Only suitable for models with **few parameters**
- Must filter out spurious peaks in hough accumulator
- Quantization of Hough space is tricky