Hough Transform

Reading

- Watt, 10.3-10.4

An edge is not a line...

How can we detect ***lines*** ?

Finding lines in an image

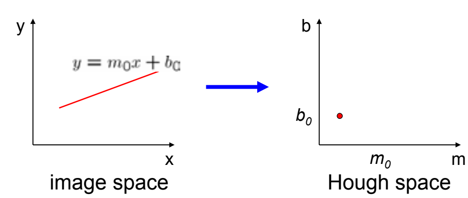
Option 1:

- Search for the line at every possible position/orientation
- What is the cost of this operation?

Option 2:

- Use a voting scheme: Hough transform

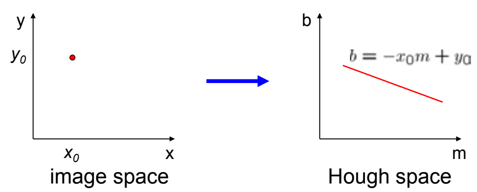
Finding lines in an image



Connection between image (x,y) and Hough (m,b) spaces

- A line in the image corresponds to a point in Hough space
- To go from image space to Hough space:
  - given a set of points (x,y), find all (m,b) such that y = mx + b

Finding lines in an image



Connection between image (x,y) and Hough (m,b) spaces

- A line in the image corresponds to a point in Hough space
- To go from image space to Hough space:
  - given a set of points (x,y), find all (m,b) such that y = mx + b
- What does a point (x_0, y_0) in the image space map to?
  - A: the solutions of $b = -x_0 m + y_0$
  - this is a line in Hough space

Hough transform algorithm

Typically use a different parameterization

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

- d is the perpendicular distance from the line to the origin
- θ is the angle this perpendicular makes with the x axis
- Why?

Hough transform algorithm

Typically use a different parameterization

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

- d is the perpendicular distance from the line to the origin
- θ is the angle this perpendicular makes with the x axis
- Why?

Basic Hough transform algorithm

1. Initialize H[d, θ]=0
2. for each edge point I[x,y] in the image

   for θ = 0 to 180

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

   H[d, θ] += 1
3. Find the value(s) of (d, θ) where H[d, θ] is maximum
4. The detected line in the image is given by $d = x cos\theta + y sin\theta$

What's the running time (measured in # votes)?

Hough line demo

Extensions

Extension 1: Use the image gradient

1. same
2. for each edge point I[x,y] in the image

   compute unique (d, θ) based on image gradient at (x,y)

   H[d, θ] += 1
3. same
4. same

What's the running time measured in votes?

Extensions

Extension 1: Use the image gradient

1. same
2. for each edge point I[x,y] in the image
   compute unique (d, θ) based on image gradient at (x,y)
   H[d, θ] += 1
3. same
4. same

What's the running time measured in votes?

Extension 2

- give more votes for stronger edges

Extension 3

- change the sampling of (d, θ) to give more/less resolution

Extension 4

- The same procedure can be used with circles, squares, or any other shape

Hough circle demo