Chapter 10: Grouping + Model Fitting

Consider finding lines in an image:

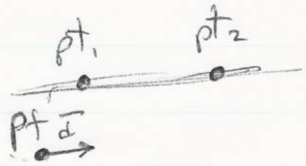
* Find pixels that have an edge reponse
* Find sets of pixels that form a line

Consider mobile robot in hallway
A9/CS5320-lecture 21 Mar

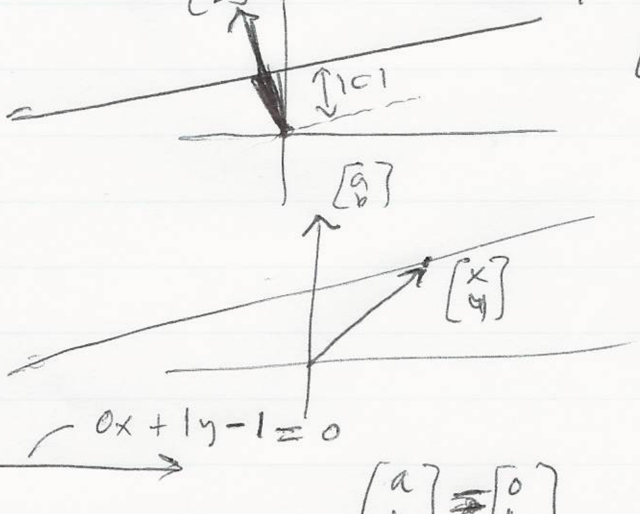
see K_means texture at work
Then look at lines + planes

Line fitting

How to represent a line?
2 points: works in any dimension
pt + direction (unit)
general equation: $ax + by + c = 0$

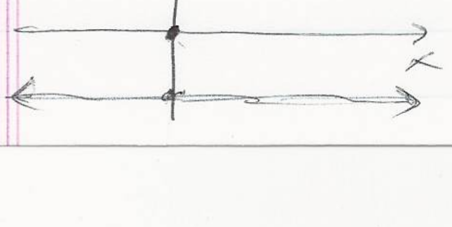


$\begin{bmatrix} a \\ b \end{bmatrix}$ or $x\cos\theta + y\sin\theta + c = 0$



$\begin{bmatrix} a \\ b \end{bmatrix}$ is normal to line

$\begin{bmatrix} a \\ b \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} = -c$



$0x + 1y - 1 = 0$

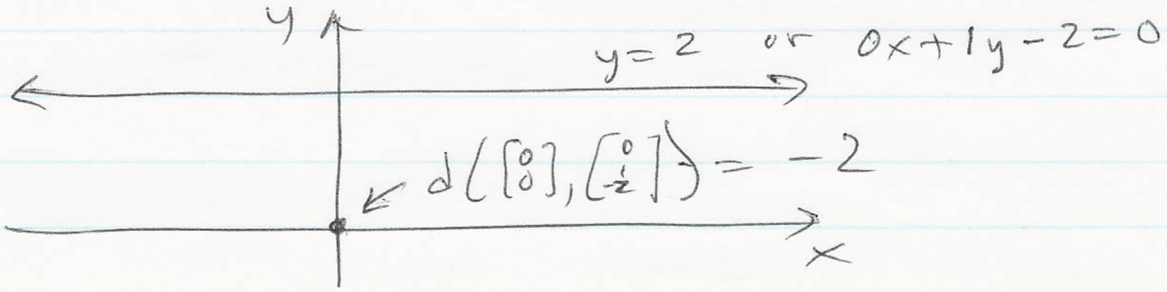
$\begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ $c = -1$

Given a set of points, find parameters.
(so just picking 2 points won't do)

Given any x, y, then

$$d\left(\begin{bmatrix} x \\ y \end{bmatrix}, \text{line} \equiv \begin{bmatrix} a \\ b \\ c \end{bmatrix}\right) = ax + by + c$$

(signed distance)



Try $\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$. $0 \cdot 0 + 1 \cdot 0 - 2 = -2$

<u>Total least squares:</u> (p. 295)

minimize $\sum_i (a x_i + b y_i + c)^2$ subject to $a^2 + b^2 = 1$

Solve: eigenvalue problem:

$$\begin{bmatrix} \overline{x^2} & \overline{xy} & \overline{x} \\ \overline{xy} & \overline{y^2} & \overline{y} \\ \overline{x} & \overline{y} & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \lambda \begin{bmatrix} 2a \\ 2b \\ 0 \end{bmatrix} \quad \text{where } c = -a\overline{x} - b\overline{y}$$

$\Rightarrow$ solve

$$\begin{bmatrix} \overline{x^2} - \overline{x}\,\overline{x} & \overline{xy} - \overline{x}\,\overline{y} \\ \overline{xy} - \overline{x}\,\overline{y} & \overline{y^2} - \overline{y}\,\overline{y} \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \mu \begin{bmatrix} a \\ b \end{bmatrix}$$

10/2a substitute

$$a\overline{x^2} + b\overline{xy} + (-a\bar{x} - b\bar{y})\bar{x} = \lambda 2a$$

$$a\overline{xy} + b\overline{y^2} + (-a\bar{x} - b\bar{y})\bar{y} = \lambda 2b$$

$$a(\overline{x^2} - \bar{x}\bar{x}) + b(\overline{xy} - \bar{x}\bar{y}) = \lambda 2a$$

$$a(\overline{xy} - \bar{x}\bar{y}) + b(\overline{y^2} - \bar{y}\bar{y}) = \lambda 2b$$

$$\begin{bmatrix} \overline{x^2} - \bar{x}\bar{x} & \overline{xy} - \bar{x}\bar{y} \\ \overline{xy} - \bar{x}\bar{y} & \overline{y^2} - \bar{y}\bar{y} \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \mu \begin{bmatrix} a \\ b \end{bmatrix}$$

plane

$$\sum_i (a x_i + b y_i + c z_i + d)^2$$

$$\sum_i \Big[a^2 x_i^2 + 2ab\, x_i y_i + 2ac\, x_i z_i + 2ad\, x_i$$
$$+ b^2 y_i^2 + 2bc\, y_i z_i + 2bd\, y_i$$
$$+ c^2 z_i^2 + 2cd\, z_i$$
$$+ d^2 \Big]$$

$$\begin{bmatrix} \overline{x^2} & \overline{xy} & \overline{xz} & \overline{x} \\ \overline{xy} & \overline{y^2} & \overline{yz} & \overline{y} \\ \overline{xz} & \overline{yz} & \overline{z^2} & \overline{z} \\ \overline{x} & \overline{y} & \overline{z} & 1 \end{bmatrix} \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} = \lambda \begin{bmatrix} 2a \\ 2b \\ 2c \\ 0 \end{bmatrix}$$

$$d = -a\overline{x} - b\overline{y} - cz$$

10/3

see CS 5320_lecture - 21 Mar

<u>Problem for us</u>: given image, find lines

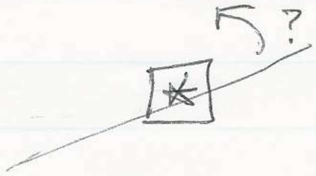
<u>Issues</u>: * edge points (may not be lines)
* multiple lines

(see hall image)

<u>Hough transform</u>

Suppose every edge pixel is part of some line

but we don't know which one:

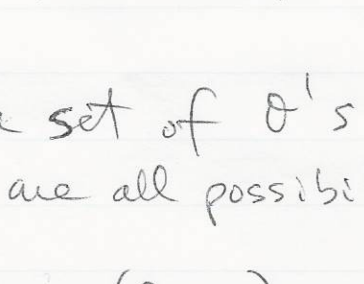


rewrite line equation:

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

Try a set of θ's + compute r's
These are all possibilities.

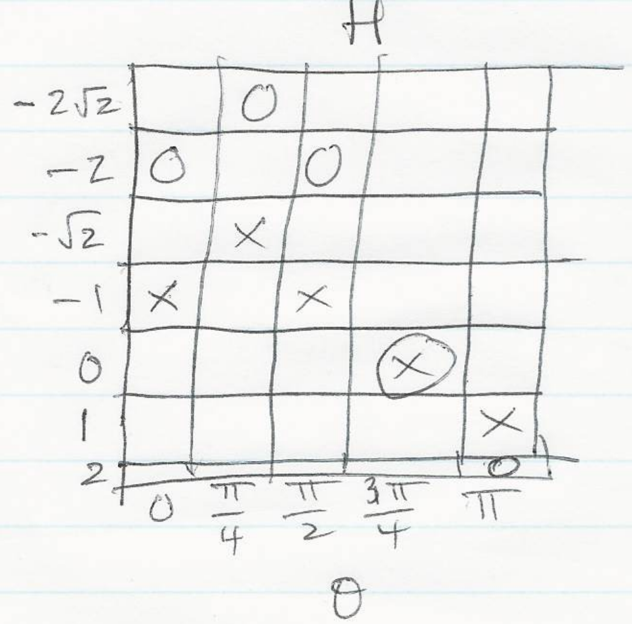
Let each (θ, r) combo get 1 vote from this edge

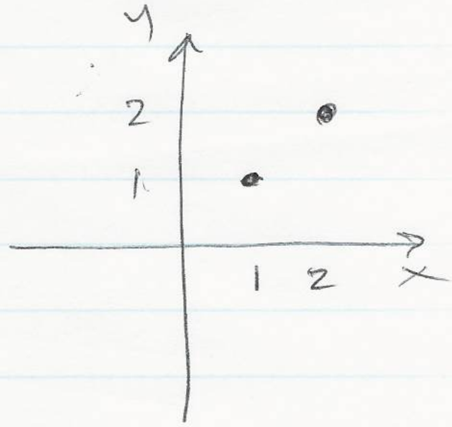


see Matlab lecture

10\4

Consider an accumulator array, discretized in r + θ





for $\begin{bmatrix}1\\1\end{bmatrix}$ use X

for $\begin{bmatrix}2\\2\end{bmatrix}$ use O

$$H\left(0, \frac{3\pi}{4}\right) = 2$$ max

$$\frac{3*\pi}{4} = 2.3562$$ (see graph for 1 pt)

Issues: * setting up indexes of array
* limiting range of r

lines are maxima in H

see Hough output

Matlab

10 x 10 diag = 15

#r's = 2* diag = 30

16 − 15 = 1

10/5

Hough algorithm

use ρ in 1 pixel increments ; [1 : 2* diag]
θ in 1 degree increments from 0 to 179

input: imo : gray level
output: H, Hpts

```
im ← edge detector        (edge, 'canny')
initialize H, Hpts
∀ edge pixel (r,c)
   x,y ← convert from (r,c)
   ∀ θ
      ρ ← compute ρ
      ρ_index ← compute ρ index
      increment H (ρ_index, θ_index)
        add r,c point to Hpts (ρ_index, θ_index)
   end
end
```

10/6

<u>Hough_lines</u>

input: im, H, thresh (absolute)
output: lines (image with labled lines)

Matlab

<u>Hough_draw_pts</u>

input: im, H
output: lines

Matlab

<u>plot_line</u>

input: ρ, x_1, x_2, y_1, y_2
← [a, b, c]

plot from x_1 to x_2 unless vertical, then from y_1 to y_2

<u>line_segs</u>

input: ime, Hpts, min_len
output: segments .pts .rho .theta .endpt .endpt2

<u>shapes</u>

10\7

Fitting Curved Shapes

Hough can be used for circles, ellipses

General Hough shape method

Other: Ransac

(1) Try (random) small sample
(2) fit line
(3) see how many pts fit
(4) continue till likely got line

p. 305 Matlab