Hough transform
Course announcements

• Homework 1 is now due on **Monday February 11th**!
  - Any questions about the homework?
  - How many of you have looked at/started/finished homework 1?

• Some changes to office hours for this week only:
  - Tuesday’s office hours will be 5:30-7:30 pm and will be covered by Yannis.
  - Friday’s 3-5 pm office hours will be covered by Anshuman.
  - There will be an extra set of office hours this week to make up for the change.
Overview of today’s lecture

Leftover from Lecture 3:

• Frequency-domain filtering.
• Revisiting sampling.

New in lecture 4:

• Finding boundaries.
• Line fitting.
• Line parameterizations.
• Hough transform.
• Hough circles.
• Some applications.
Slide credits

Most of these slides were adapted from:

• Kris Kitani (15-463, Fall 2016).

Some slides were inspired or taken from:

• Fredo Durand (MIT).
• James Hays (Georgia Tech).
Finding boundaries
Where are the object boundaries?
edge detection
Edge strength does not necessarily correspond to our perception of boundaries.
Where are the object boundaries?
Defining boundaries are hard for us too.
Where is the boundary of the mountain top?
Lines are hard to find

Original image

Edge detection

Thresholding

Noisy edge image
Incomplete boundaries
Applications

- Autonomous Vehicles (lane line detection)
- Tissue engineering (blood vessel counting)
- Behavioral genetics (earthworm contours)
- Autonomous Vehicles (semantic scene segmentation)
- Computational Photography (image inpainting)
Line fitting
Line fitting

Given: Many \((x_i, y_i)\) pairs

Find: Parameters \((m, c)\)

Minimize: Average square distance:

\[
E = \sum_{i} \frac{(y_i - mx_i - c)^2}{N}
\]
Line fitting

Given: Many \((x_i, y_i)\) pairs

Find: Parameters \((m, c)\)

Minimize: Average square distance:

\[
E = \frac{\sum (y_i - mx_i - c)^2}{N}
\]

Using:

\[
\frac{\partial E}{\partial m} = 0 \quad \& \quad \frac{\partial E}{\partial c} = 0
\]

Note:

\[
\bar{y} = \frac{\sum y_i}{N} \quad \bar{x} = \frac{\sum x_i}{N}
\]

Note:

\[
c = \bar{y} - m \bar{x}
\]

\[
m = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sum (x_i - \bar{x})^2}
\]

What are some problems with the approach?
Data: \((x_1, y_1), \ldots, (x_n, y_n)\)

Line equation: \(y_i = m x_i + b\)

Find \((m, b)\) to minimize \(E = \sum_{i=1}^{n} (y_i - mx_i - b)^2\)

\[
Y = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}, \quad X = \begin{bmatrix} x_1 & 1 \\ \vdots & \vdots \\ x_n & 1 \end{bmatrix}, \quad B = \begin{bmatrix} m \\ b \end{bmatrix}
\]

\[
E = \|Y - XB\|^2 = (Y - XB)^T (Y - XB) = Y^T Y - 2(XB)^T Y + (XB)^T (XB)
\]

\[
\frac{dE}{dB} = 2X^T XB - 2X^T Y = 0
\]

\[
X^T XB = X^T Y
\]

Normal equations: least squares solution to \(XB = Y\)
Line fitting

Given: Many \((x_i, y_i)\) pairs

Find: Parameters \((m, c)\)

Minimize: Average square distance:

\[
E = \frac{1}{N} \sum_i (y_i - mx_i - c)^2
\]

How can we solve this minimization?
Problems with parameterizations

Where is the line that minimizes $E$?

$$E = \sum_{i=1}^{n} (y_i - mx_i - b)^2$$
Problems with parameterizations

Where is the line that minimizes $E$?

$$E = \sum_{i=1}^{n} (y_i - mx_i - b)^2$$

Huge $E$!
Problems with parameterizations

Where is the line that minimizes $E$?

$$E = \sum_{i=1}^{n} (y_i - mx_i - b)^2$$

Line that minimizes $E$!!
Problems with parameterizations

Where is the line that minimizes $E$?

How can we deal with this?
Line fitting is easily setup as a maximum likelihood problem … but choice of model is important.

\[ E = \sum_{i=1}^{n} (y_i - mx_i - b)^2 \]

What optimization are we solving here?
Problems with noise

Squared error heavily penalizes outliers

Least-squares error fit  Squared error heavily penalizes outliers
Model fitting is difficult because...

- **Extraneous data**: clutter or multiple models
  - We do not know what is part of the model?
  - Can we pull out models with a few parts from much larger amounts of background clutter?

- **Missing data**: only some parts of model are present

- **Noise**

- **Cost**:
  - It is not feasible to check all combinations of features by fitting a model to each possible subset

*So what can we do?*
Line parameterizations
Slope intercept form

\[ y = mx + b \]

What are \( m \) and \( b \)?
Slope intercept form

\[ y = mx + b \]

slope  
\[ b \]  
y-intercept
Double intercept form

\[ \frac{x}{a} + \frac{y}{b} = 1 \]

What are \( x \) and \( y \)?
Double intercept form

\[
\frac{x}{a} + \frac{y}{b} = 1
\]

Derivation:

(Similar slope) \[\frac{y - b}{x - 0} = \frac{0 - y}{a - x}\]

\[ya + yx - ba + bx = -yx\]

\[ya + bx = ba\]

\[\frac{y}{b} + \frac{x}{a} = 1\]
Normal Form

\[ x \cos \theta + y \sin \theta = \rho \]

*What are rho and theta?*
Normal Form

\[ x \cos \theta + y \sin \theta = \rho \]

Derivation:

\[
\begin{align*}
\cos \theta &= \frac{\rho}{a} \rightarrow a &= \frac{\rho}{\cos \theta} \\
\sin \theta &= \frac{\rho}{b} \rightarrow b &= \frac{\rho}{\sin \theta}
\end{align*}
\]

plug into:

\[
\frac{x}{a} + \frac{y}{b} = 1
\]

\[ x \cos \theta + y \sin \theta = \rho \]
Hough transform
Hough transform

- Generic framework for detecting a parametric model
- Edges don’t have to be connected
- Lines can be occluded
- Key idea: edges vote for the possible models
Image and parameter space

\[ y = mx + b \]
Image and parameter space

$$y = mx + b$$

variables

parameters

Image space

Parameter space

$$y - mx = b$$

variables

parameters

a line becomes a point

(1, 1)
Image and parameter space

\[ y = mx + b \]

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

\[ y = mx + b \]

variables
parameters

\[ y - mx = b \]

variables
parameters

Image space

Parameter space

a point becomes a line
Image and parameter space

\[ y = mx + b \]

variables
parameters

\[ y - mx = b \]

variables
parameters

Image space

Parameter space

two points become ?
Image and parameter space

\[ y = mx + b \]

Image space

Parameter space

\[ y - mx = b \]
Image and parameter space

\[ y = mx + b \]

variables

\[ y - mx = b \]

variables

three points become ?

Image space

Parameter space
Image and parameter space

\[ y = mx + b \]

variables

parameters

\[ y - mx = b \]

variables

parameters

three points become ?

Image space

Parameter space
Image and parameter space

\[ y = mx + b \]

\[ y - mx = b \]

Image space

Parameter space

four points become ?
Image and parameter space

\[ y = mx + b \]

variables

parameters

\[ y - mx = b \]

variables

parameters

Image space

Parameter space

four points become ?
How would you find the best fitting line?

Is this method robust to measurement noise?

Is this method robust to outliers?
Line Detection by Hough Transform

**Algorithm:**

1. Quantize Parameter Space \((m,c)\)

2. Create Accumulator Array \(A(m,c)\)

3. Set \(A(m,c) = 0\) \(\forall m,c\)

4. For each image edge \((x_i,y_i)\)
   - For each element in \(A(m,c)\)
   - If \((m,c)\) lies on the line: \(c = -x_i m + y_i\)
     - Increment \(A(m,c) = A(m,c) + 1\)

5. Find local maxima in \(A(m,c)\)
Problems with parameterization

*How big does the accumulator need to be for the parameterization \((m, c)\)?*
Problems with parameterization

How big does the accumulator need to be for the parameterization \((m, c)\)?

The space of \(m\) is huge! The space of \(c\) is huge!

\[-\infty \leq m \leq \infty\]

\[-\infty \leq c \leq \infty\]
Better Parameterization

Use normal form:

\[ x \cos \theta + y \sin \theta = \rho \]

Given points \((x_i, y_i)\) find \((\rho, \theta)\)

Hough Space Sinusoid

\[ 0 \leq \theta \leq 2\pi \]
\[ 0 \leq \rho \leq \rho_{\text{max}} \]

(Finite Accumulator Array Size)
Image and parameter space

\[ y = mx + b \]

variables
parameters

\[ x \cos \theta + y \sin \theta = \rho \]

parameters
variables

a point becomes?

Image space

Parameter space
Image and parameter space

\[ y = mx + b \]

\[ x \cos \theta + y \sin \theta = \rho \]

Image space

Parameter space

a point becomes a wave
Image and parameter space

\[ y = mx + b \]

\[ x \cos \theta + y \sin \theta = \rho \]

Image space

Parameter space

(a line becomes?)
Image and parameter space

\[ y = mx + b \]

\[ x \cos \theta + y \sin \theta = \rho \]

Image space

Parameter space

(a line becomes a point)
Image and parameter space

\[ y = mx + b \]

\[ x \cos \theta + y \sin \theta = \rho \]
Image and parameter space

\[ y = mx + b \]

\[ x \cos \theta + y \sin \theta = \rho \]

Image space

Parameter space

a line becomes a point
Image and parameter space

Variables

\[ y = mx + b \]

Parameters

\[ x \cos \theta + y \sin \theta = \rho \]

Image space

Parameter space

a line becomes a point
Image and parameter space

\[ y = mx + b \]

\[ x \cos \theta + y \sin \theta = \rho \]

Image space

Parameter space

variables

parameters

a line becomes a point
Image and parameter space

\[ y = mx + b \]

\[ x \cos \theta + y \sin \theta = \rho \]

Image space

Parameter space

a line becomes a point
Image and parameter space

Image space

Parameter space

\[ y = mx + b \]

\[ x \cos \theta + y \sin \theta = \rho \]
Image and parameter space

\[ y = mx + b \]

\[ x \cos \theta + y \sin \theta = \rho \]

Wait…why is \( \rho \) negative?

Variables

Parameters

a line becomes a point

Image space

Parameter space
Image and parameter space

\[ y = mx + b \]

variables

parameters

\[ x \cos \theta + y \sin \theta = \rho \]

same line through the point

a line becomes a point

Image space

Parameter space
There are two ways to write the same line:

Positive rho version:

\[ x \cos \theta + y \sin \theta = \rho \]

Negative rho version:

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

Recall:

\[ \sin(\theta) = -\sin(\theta + \pi) \]
\[ \cos(\theta) = -\cos(\theta + \pi) \]
Image and parameter space

\[ y = mx + b \]

\[ x \cos \theta + y \sin \theta = \rho \]

Image space

Parameter space

variables

parameters

(1, 1)

same line through the point

a line becomes a point
Image and parameter space

\[ y = mx + b \]

variables

parameters

two points become ?

Image space

Parameter space
Image and parameter space

$y = mx + b$

variables

parameters

three points become ?
Image and parameter space

\[ y = mx + b \]

variables
parameters

four points become ?

Image space
Parameter space
Implementation

1. Initialize accumulator $H$ to all zeros

2. For each edge point $(x, y)$ in the image
   For $\theta = 0$ to 180
     $\rho = x \cos \theta + y \sin \theta$
     $H(\theta, \rho) = H(\theta, \rho) + 1$
   end
end

3. Find the value(s) of $(\theta, \rho)$ where $H(\theta, \rho)$ is a local maximum

4. The detected line in the image is given by
   $\rho = x \cos \theta + y \sin \theta$

NOTE: Watch your coordinates. Image origin is top left!
Basic shapes
(in parameter space)

*can you guess the shape?*
Basic shapes
(in parameter space)

line
Basic shapes
(in parameter space)

line
rectangle
Basic shapes
(in parameter space)

line
rectangle
circle
Basic Shapes
More complex image
In practice, measurements are noisy…
Too much noise …
Effects of noise level

More noise, less votes (in the right bin)
Effect of noise points

More noise, more votes (in the wrong bin)
Real-world example

Original

Edges

parameter space

Hough Lines
Hough Circles
Let's assume radius known.

\[(x - a)^2 + (y - b)^2 = r^2\]

What is the dimension of the parameter space?
What does a point in image space correspond to in parameter space?
\[(x - a)^2 + (y - b)^2 = r^2\]
\[(x - a)^2 + (y - b)^2 = r^2\]

parameters

variables
\[(x - a)^2 + (y - b)^2 = r^2\]
$$(x - a)^2 + (y - b)^2 = r^2$$
What if radius is unknown?

\[(x - a)^2 + (y - b)^2 = r^2\]

\[(x - a)^2 + (y - b)^2 = r^2\]
What if radius is unknown?

If radius is not known: 3D Hough Space!

Use Accumulator array $A(a, b, r)$

Surface shape in Hough space is complicated
Using Gradient Information

Gradient information can save lot of computation:

- Edge Location: \((x_i, y_i)\)
- Edge Direction: \(\phi_i\)

Assume radius is known:

\[
\begin{align*}
a &= x - r \cos \phi \\
b &= y - r \sin \phi
\end{align*}
\]

Need to increment only one point in accumulator!
\[(x - a)^2 + (y - b)^2 = r^2\]
\[(x - a)^2 + (y - b)^2 = r^2\]
Pennie Hough detector  

Quarter Hough detector
Can you use Hough Transforms for other objects, beyond lines and circles?

Do you have to use edge detectors to vote in Hough Space?
The Hough transform ...

Deals with occlusion well?
Detects multiple instances?
Robust to noise?
Good computational complexity?
Easy to set parameters?
Application of Hough transforms
Detecting shape features

F. Jurie and C. Schmid, Scale-invariant shape features for recognition of object categories, CVPR 2004
Original images

A

Laplacian circles

B

Hough-like circles

D

Which feature detector is more consistent?
Robustness to scale and clutter
Object detection

Index displacements by “visual codeword”

B. Leibe, A. Leonardis, and B. Schiele, Combined Object Categorization and Segmentation with an Implicit Shape Model, ECCV Workshop on Statistical Learning in Computer Vision 2004
References

Basic reading:
• Szeliski textbook, Sections 4.2, 4.3.