Hough Transform
Edges vs Boundaries

• Edges
  – Local intensity discontinuities
  – Points
  – Not dependent on models

• Boundaries
  – Extensive
  – Composed of many points
  – May be dependent on models

• Typically our goal is to reconstruct the boundary from local edge elements

Local edge point or edge element
Object boundary
May need domain or object model
Knowledge about Boundary

- Can extract the complete closed contour of the object if its shape is known

- Can extract pieces of the boundary using general line or curve models

- Can just try to find a connected series of edge elements, using only heuristics on say, edge curvature
Finding Lines via Hough Transform

- Useful for detecting any parametric curves (e.g., lines, conics)
- Relatively unaffected by gaps in curves, and noise
- Given a set of edge points, find the line(s) which best explain the data
Hough Transform (continued)

- A line has two parameters \((m, b)\)

  \[
  y = mx + b
  \]

- Given a point \((x_0, y_0)\), the lines that could pass through this point are all \((m, b)\) satisfying

  \[
  y_0 = mx_0 + b
  \]

  Or

  \[
  b = -x_0m + y_0
  \]

  The equation \(b = -x_0m + y_0\) is a line in \((m, b)\) space

\(\text{Diagram:} \quad \text{An} \quad \text{Hough} \quad \text{transform} \quad \text{representation} \quad \text{of} \quad \text{lines} \quad \text{in} \quad \text{an} \quad \text{image} \)
Hough Transform (continued)

• All points on a line in image space, yield lines in parameter space which intersect at a common point
  – This point is the \((m,b)\) of the line in image space

\[\begin{align*}
  y &= mx + b \\
  b &= -x_2 m + y_2 \\
  b &= -x_1 m + y_1
\end{align*}\]
Hough Transform Algorithm

- Initialize an accumulator array $A(m,b)$ to zero
- For each edge element $(x,y)$, increment all cells that satisfy $b = -x \cdot m + y$
- Local maxima in $A(m,b)$ correspond to lines

10 points voted for this line $(m,b)$
Polar Coordinate Representation of Line

- $\rho = x \cos \theta + y \sin \theta$
  - Avoids infinite slope
  - Constant resolution

The parameter space transform of a point is a sinusoidal curve
Hough Transform

- Angle, axis conventions
  - angle range is -90°..+89°
  - rho range is –dmax..+dmax
    - dmax is the largest possible distance

- Example of a point at \((x,y) = (50,100)\)

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

\[
\rho = 50 \cos(-45) + 100 \sin(-45)
\]

\[
= 50 \sqrt{2}/2 - 100 \sqrt{2}/2 = -25\sqrt{2}
\]

\[
\rho = 50 \cos(45) + 100 \sin(45)
\]

\[
= 50 \sqrt{2}/2 + 100 \sqrt{2}/2 = 75\sqrt{2}
\]
Example

Image of size 100x100, containing 5 points (at the corners)
Example

Image of size 100x100, containing 5 points (at the corners)
FIGURE 10.34 (a) A 502 × 564 aerial image of an airport. (b) Edge image obtained using Canny’s algorithm. (c) Hough parameter space (the boxes highlight the points associated with long vertical lines). (d) Lines in the image plane corresponding to the points highlighted by the boxes. (e) Lines superimposed on the original image.
Pseudo Code

for all x
    for all y
        if edge point at (x,y)
            for all theta
                \[
                \rho = x \cos(\theta) + y \sin(\theta)
                \]
                increment (add 1 to) the cell in H corresponding to \((\theta, \rho)\)
            end
        end
    end
end
Matlab Hough Transform Functions

- \([H, \theta, \rho] = \text{hough}(bw)\)
  - Output Hough array, size NRho x NTheta
  - Vectors of theta and rho values
  - Input binary image of edge points

- \(\text{peaks} = \text{houghpeaks}(H, \text{numpeaks})\)
  - Output array (row,col) of peaks (up to numpeaks)
  - The (rho, theta) values for the ith peak are
    \(r = \rho(\text{peaks}(i,1));\)
    \(t = \theta(\text{peaks}(i,2));\)

- \(\text{lines} = \text{houghlines}(bw, \theta, \rho, \text{peaks})\)
  - Output structure array of lines. Each line has fields:
    (endpoint1, endpoint2, rho, theta)
clear all
close all
I = imread('gantrycrane.png');

G = rgb2gray(I);
E = edge(G, 'canny');
imshow(E)

[H,theta,rho] = hough(E);
figure, imshow(H,[],);

peaks = houghpeaks(H,50,'Threshold',30);
figure, imshow(G,[],), hold on

lines =
houghlines(E,theta,rho,peaks,'FillGap',5,'MinLength',15);

for k = 1:length(lines)
    xy = [lines(k).point1; lines(k).point2];
    plot(xy(:,1),xy(:,2),'LineWidth',1,'Color','r');
end
Other Shapes

• Hough transform can be used to find any parameterized curve
• Example - circles \((x_0, y_0, r)\)
Other Shapes

- Hough transform can be used to find any parameterized curve
- Example - circles \((x_0, y_0, r)\)

\[
(\alpha - x_0)^2 + (\beta - y_0)^2 = r^2
\]

\[
r = \sqrt{(x - x_0)^2 + (y - y_0)^2}
\]

INC H at cell corresp to \(x_0, y_0, r\)
Hough Transform

• Is an efficient way to implement a matched filter
• Very tolerant to noise and missing data

• Problems:
  – Complexity
    • For N parameters, you have N-1 nested “for” loops to increment accumulator array, at each edge point
  – Choosing bin size
    • Smaller bin size yields more precision
    • But smaller bin size takes more running time; also fewer counts in each bin
  – Finding peaks
    • Noise in input image may cause counts to be spread across multiple bins