Line Detection

CS 510
Lecture #22
4/15/00

Canny Example

Source image

Canny: sigma = 2.0, low = 0.40, high = 0.90

Hierarchical Feature Extraction

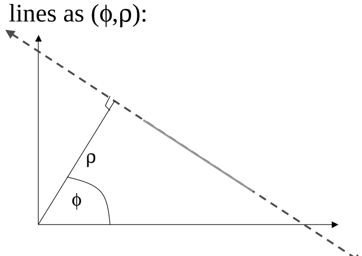
- Most features are extracted by combining a small set of primitive features (edges, corners, regions)
  - Grouping: which edges/corners/curves form a group?
  - Model Fitting: what structure best describes the group?
- Simple example: The Hough Transform
  - Groups points into lines
  - (developed in 1962)

Hough Transform: Grouping

- The idea of the Hough transform is that a change in representation converts a point grouping problem into a peak detection problem
- Standard line representations:
  - y = mx + b -- *problems with vertical lines*
  - $(x_0, y_0) + t(x_1, y_1)$ -- *your raytracer used this form*
  - ax + by + c = 0 -- *Bresenham's uses this form*
- How else might you represent a line?

Hough Grouping (cont.)

- The Hough Transform represents infinite lines as (ϕ,ρ):

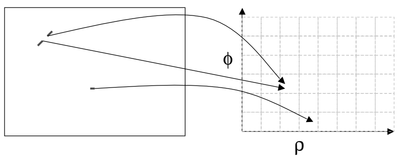


Hough Grouping (III)

- Why? This representation is:
  - Small: only two parameters (like y=mx+b)
  - Finite: $0 \le \rho \le \sqrt{(row^2+col^2)}$, $0 \le \phi \le 2\pi$
  - Unique: only one representation per line
- General Idea:
  - The Hough space (ϕ,ρ) is a representation of every possible line segment
  - Discretize the Hough space
  - Let every point "vote for" any line is might belong to.

Hough Grouping: Directed Edges

- Every edge has a location and position, so it can be part of only one (infinitely extended) line.



- Colinear edges point to a single bucket in Hough space

Hough Grouping: Edges

- This reduces line grouping to peak detection
  - # of edges/bucket is the number of edges that support that line segment
  - Position of bucket provides the ϕ, ρ parameters
- Problem: if “true” line parameters are on the boundary of a bucket, supporting data may be split
- Solution: smooth the histogram (Hough image) before selecting peaks.

Hough Grouping: Points

- Depending on noise, you may not trust the orientation data associated with your edges, or…
- You may have extracted directionless points from the image, not edges (e.g. corners)
- In this case, one point is consistent with an infinite set of line segments
  - Any line passing through the designated point
  - So vote for every line the point is consistent with

Hough Fitting

- After finding the peaks in the Hough Transform, there are still two potential problems:
  - You know the line parameters to the resolution of your buckets.
  - You have infinite lines, not line segments
- Both of these problems can be fixed, IF you kept a linked list of edges (not just a count)
  - Of course, this is more expensive...

Hough Fitting (II)

- Sort your edges
  - rotate edge points according to ρ
  - sort them by (rotated) x coordinate
- Look for gaps
  - have the user provide a “max gap” threshold
  - if two edges (in the sorted list) are more than max gap apart, break the line into segments
  - if there are enough edges in a given segment, fit a straight line to the points

Sidebar:
Fitting Straight Lines to Points

- In n dimensions, compute the eigenvalues & eigenvectors and take the eigenvector associated with the largest eigenvalue.
- In 2 dimensions, its simpler: for p points (x,y),

$$a = \sum_p x^2, b = \sum_p xy, c = \sum_p y^2$$

$$\sin 2\phi = \pm \frac{b}{\sqrt{b^2 + (a-c)^2}}$$

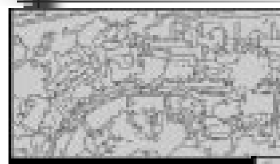
$$\cos 2\phi = \pm \frac{a-c}{\sqrt{b^2 + (a-c)^2}}$$

Hough Example

Source Image

Hough Space

Hough Example (II)



Edge data

Line data