The Hough Transform: A Trick with Many Applications

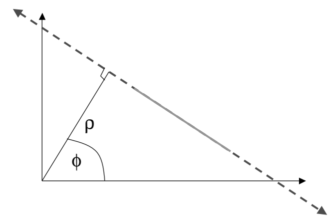
CS 510

Lecture #23

4/17/02

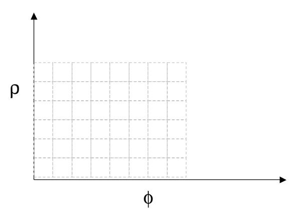
Hough Grouping (review)

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



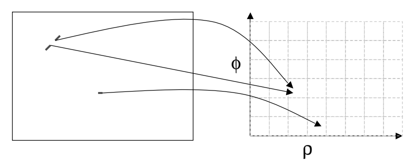
The Hough (Line) Space

- The idea is to parameterize lines, digitize the parameter space, and have edges or points “vote”



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

“Vote Early and Often”

Underconstrained Cases

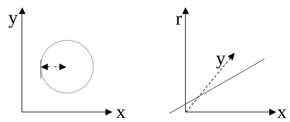
- In the case of points (rather than edges)
  - a point is consistent with infinitely many lines
  - it is not consistent with all lines, however.
- So points vote for every line they are consistent with
  - more likely to find accidental mismatches
  - higher threshold for peaks in Hough space.

Finding Circles

- This same trick (an underconstrained Hough space) can be used to find circles
  - Circles have three parameters:
    - Their center (x,y)
    - Their radius r
  - Create a 3D digitized Hough space (x,y,r)
- Every edge (with a direction) implies a line that the center must lie along.
- The radius is determined by the position of the edge & center.

Circles (cont.)

- So, every edge is consistent with an infinite number of circles.
  - These circles lie on a line in Hough space
  - Vote for all of them.
    - Note that this is scan line conversion -- Bresenham!

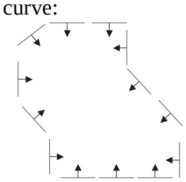


Ellipses $b^2x^2 + a^2y^2 = b^2a^2$

- Circles project to ellipses under perspective, so can we find ellipses?
- Limit ourselves to center at the origin, major axis along X:
  - So for every value of a, there is a legal value of b;
  - Vote for all [a,b] pairs in the (2D) Hough space
  - Note that the orientation of the edge is not being used
- In the fully general form, ellipses have six parameters. Can we apply a Hough space for this?

The Generalized Hough Transform *Ballard, 1981*

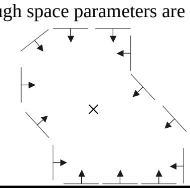
- Suppose that we have edges extracted from an arbitrary 2D curve:



- Can we use a Hough space to determine if the curve is in another image?

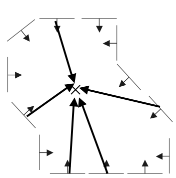
Generalized Hough (cont.)

- Under what transformation? Assume translation only
- Then select a reference point
  - Can be center of mass, doesn't have to be
  - The hough space parameters are the position of this point



Generalized Hough (III)

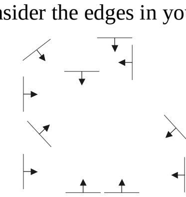
- For every edge in the model curve, there is a vector from the edge to the reference point:



- Store this set of vectors

Generalized Hough (IV)

- Now consider the edges in your test data:

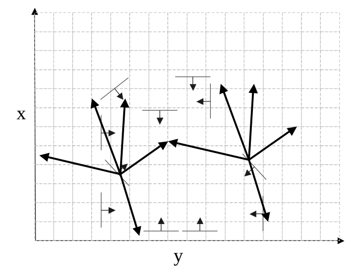


  - If the edge is part of the curve, where could the reference point be?
  - It must be one of the edges in the curve, so *the offset from the edge to the reference point must be one of the stored vectors! (vote often!)*

Generalized Hough (V)

- Create a Hough space of (x,y) reference point positions
- If there are n points in the model curve, then each edge votes n times
  - An edge votes for the (x,y) positions that can be reached by adding one of the stored vectors to it.
- The peak in this Hough space is the reference point with the most supporting edges.

Visualizing the Generalized Hough



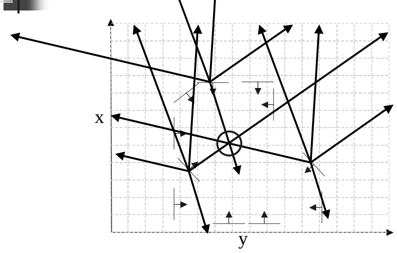
Making it Still More General...

- What if one curve might be rotated relative to another?
  - You know the orientation of the model edge
  - So you know the relative orientation of the displacement vector and the edge
  - Rotate the displacement vector prior to voting
    - This is easiest if you store displacement vectors as angle/length
- The result is rotation invariant 2D curve matching

… and Even More General

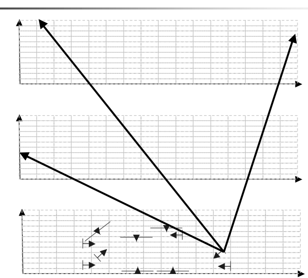
- What if the curve might vary in scale, as well as rotation and translation?
- Two options
  - The length of the displacement vectors is unknown; each edge point votes for a set of lines in Hough space
    - See next slide
  - Extend the parameterization to (x,y,scale); each edge votes for a line in this 3D Hough space
    - Larger Hough space → slower
    - Fewer accidental intersections → more robust

Method #1 Illustrated



Blue circle notes point of maximum intersection

Method #2 Illustrated



dimension is scale