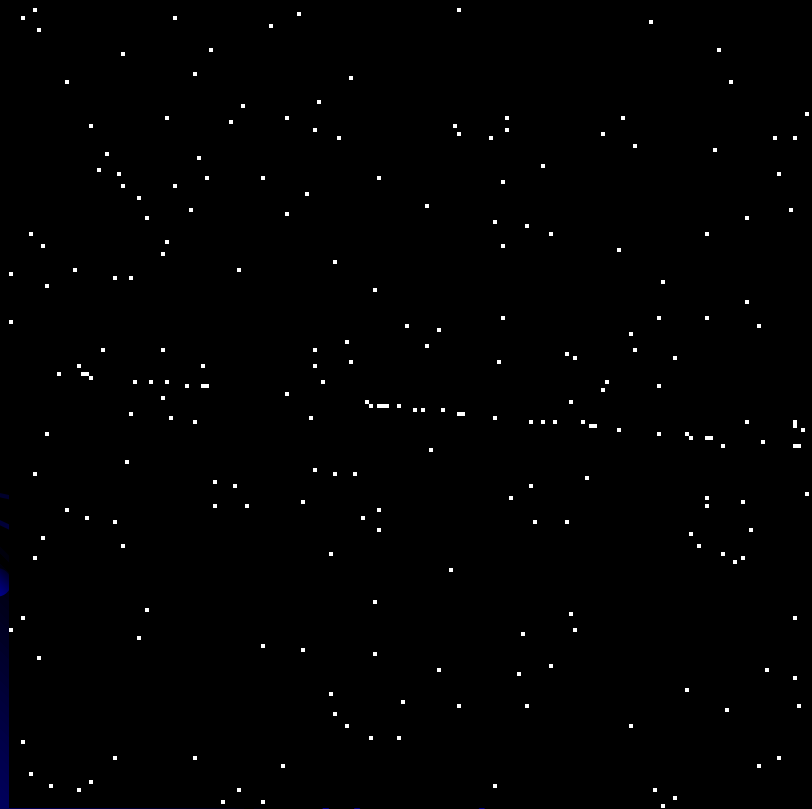


- This week
 - Today: Finding lines and circles using the Hough transform (Sonka 6.26)
 - Please fill out ANGEL evaluation of Sunset partner using "Term Project Partner Evaluation"
 - Next class: Applications of PCA
 - Sunday night: Project plans and preliminary work due. See rubric
- Questions?

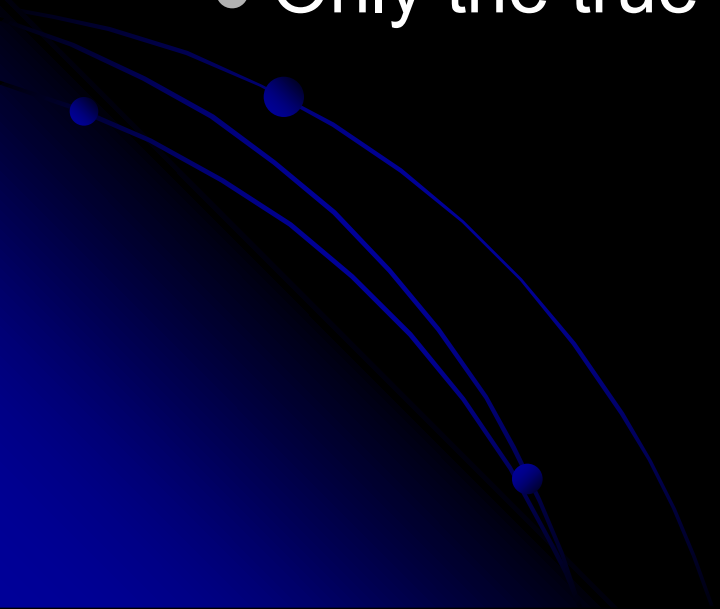
Finding lines in real images



- Input: set of edge points
- Output: the equation of a line containing them
- Methods:
 - Least-squares (if you know which points belong to the line...)
 - Hough transform (today)

Hough transform

- Idea (Sonka 6.2.6; Forsyth and Ponce, ch 15):
 - Represent a line using parameters
 - Each edge point in the image casts a vote for all lines of which it could be part.
 - Only the true line receives lots of votes



Parametric Equation of a Line

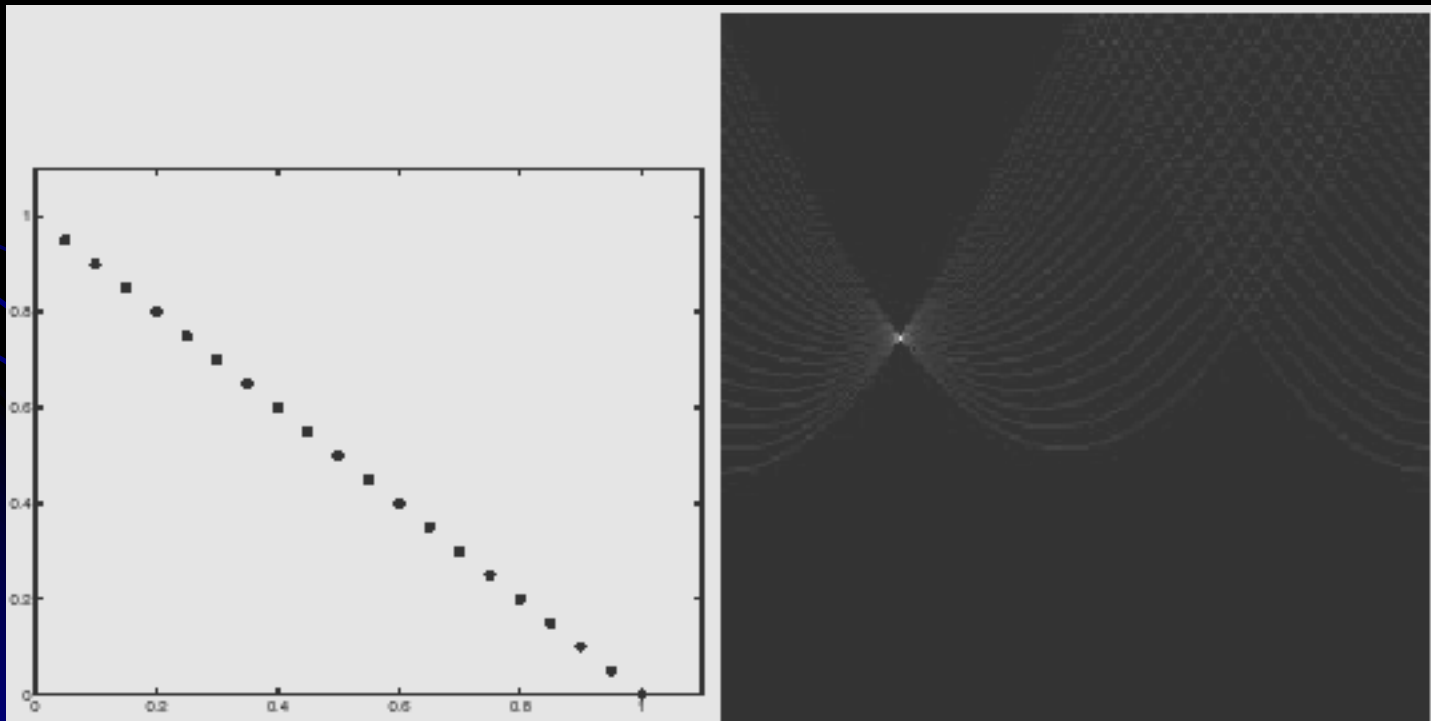
- Represent a line using 2 parameters
- $y = mx + b$
 - Problem?
- $Ax + By + C = 0$?
 - 3 parameters; but A, B, and C are related...we only need 2
- ρ and θ
 - ρ is distance from line to origin
 - θ is the angle the distance segment makes with x-axis
 - $x \cos\theta + y \sin\theta = \rho$

Voting

- Each point in image votes for all lines of which it could be part.
- Only “true” line receives lots of votes.
- Quiz question: show $(4,4)$, $(2,2)$, and $(0,0)$ voting for a line in $y = mx+b$ space (for simplicity)

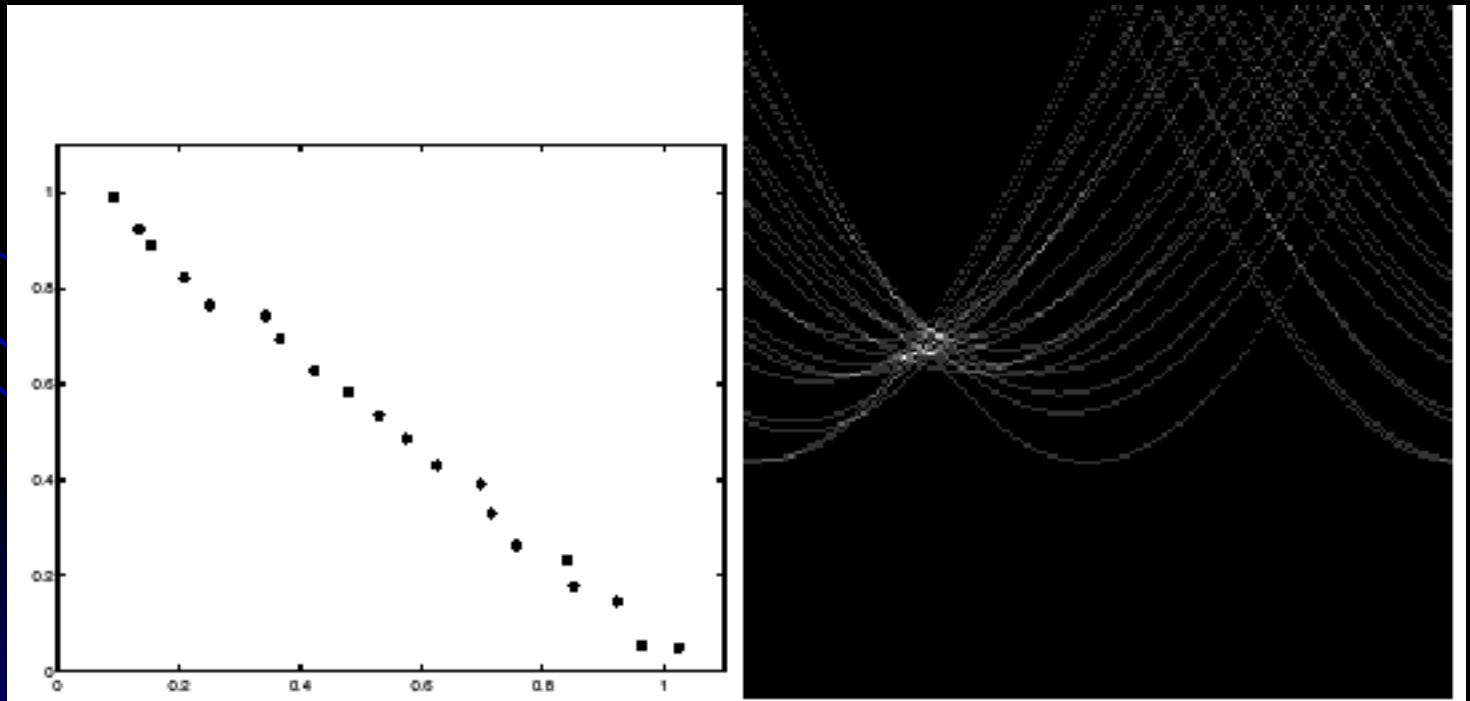
Perfect line

- Notice sharp peak in voting space
- (next 3 images from Forsyth and Ponce, ch 15)



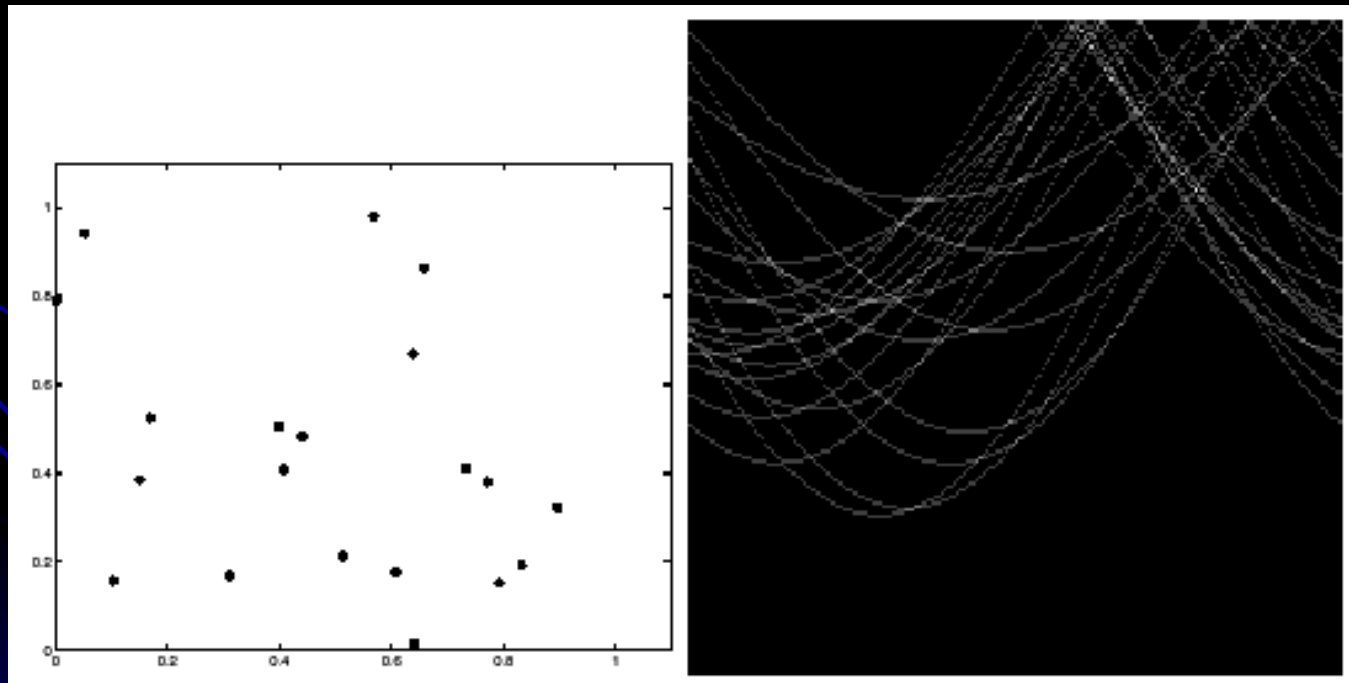
Approximate line

- Notice the broader peak. Can we detect it?
- Could smooth or use a coarser quantization?
- Accumulator array: bin size? Range?



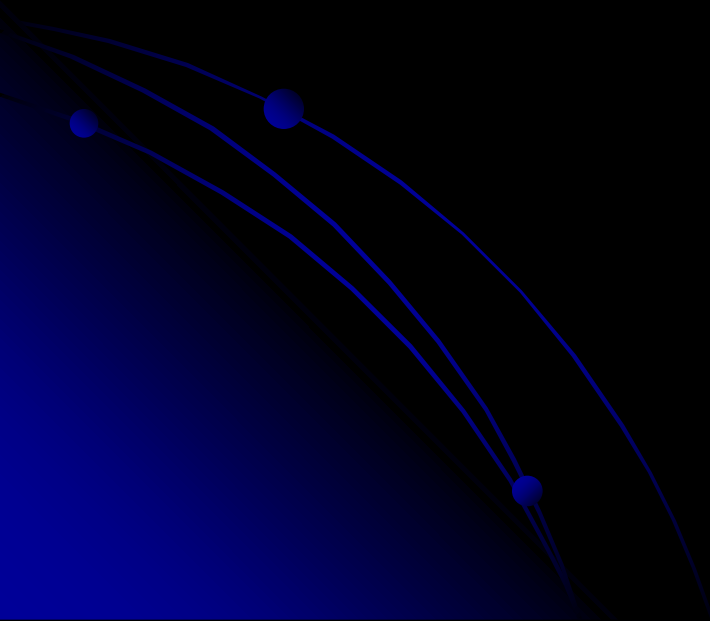
Random noise

- Votes spread all over the place: no line
- Too much noise creates “phantom lines”
 - Smoothing can sometimes help

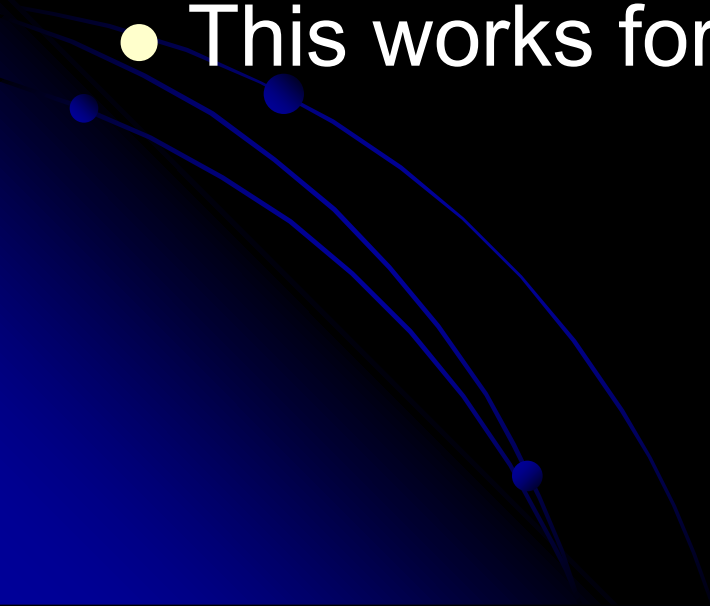


Limitations

- Finding the right grid size in parameter space may be tricky
 - Trial and error

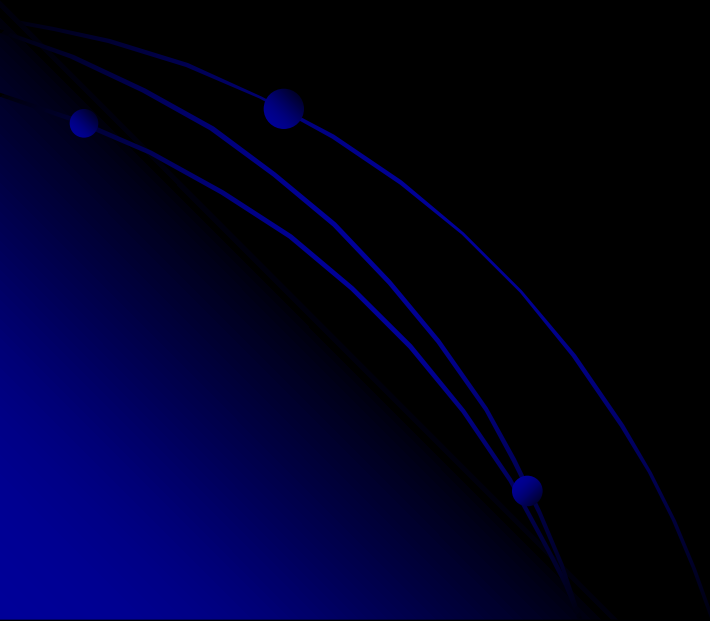


Matlab

- Run an edge detector first to find points that are voting
 - `[H, theta, rho] = hough(edgeImg);`
 - `peaks = houghpeaks(H,nPeaks);`
 - This works for lines only
- 

Another demo

<http://www.rob.cs.tu-bs.de/content/04-teaching/06-interactive/HNF.html>

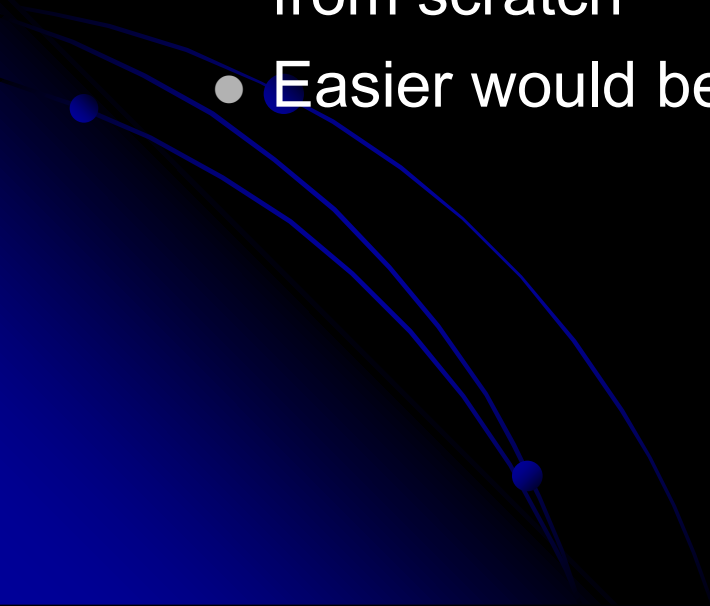


Generalizations

- Finding circles with fixed radius...
- Finding circles with arbitrary radius...
- Finding line segments
- Finding arbitrary shapes...
 - Ballard, Dana. 1981. Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2):111-122.
 - Dana was a long-time member of Rochester's computer vision group.

My Circle Finder

- Demo
- Wouldn't this be a great lab? 😊
 - Like Matlab's `hough` and `houghpeaks` (for lines), but from scratch
 - Easier would be to find circles of fixed radius.



Reducing the number of votes

- Use the edge gradient information as well
 - Only need to cast votes for centers along the gradient
 - I've done this; it works really well
 - Use partial curves. If you had a way of grouping relating points, you could use curvature.
 - I haven't tried this.
- 