Introduction to Features

CS 510

Lecture #13

March 8th, 2013

Colorado State University

What is a Feature?

- A feature is anything that is:
  - Localized
  - Meaningful
  - Detectable & Discrete
- Features are also intermediate
  - a means, not an end

Traditional Hierarchy of Features (e.g. Szeliski's book)

- Edges
- Corners
- Chains
- Line segments
- Parameterized curves
- Regions
- Surface patches
- Closed Polygons

What is an Edge?

- An edge is a description of a localized image pattern
  - We need to know what aspect of the pattern we are measuring

- An edge is a symbolic feature
  - We need to know what it denotes:
    - surface marking, or
    - surface discontinuity, or
    - shadow (illumination discontinuity)
  - These things have precise positions

The Facet Model
Review

- The image can be thought of as a gray level intensity surface
  - piecewise flat (flat facet model)
  - piecewise linear (sloped facet model)
  - piecewise quadratic
  - piecewise cubic
  - Example http://www.mirametrics.com/brief_pro_graphics_2.htm
- Processing implicitly or explicitly estimates the free parameters.

Facet Edge Detection

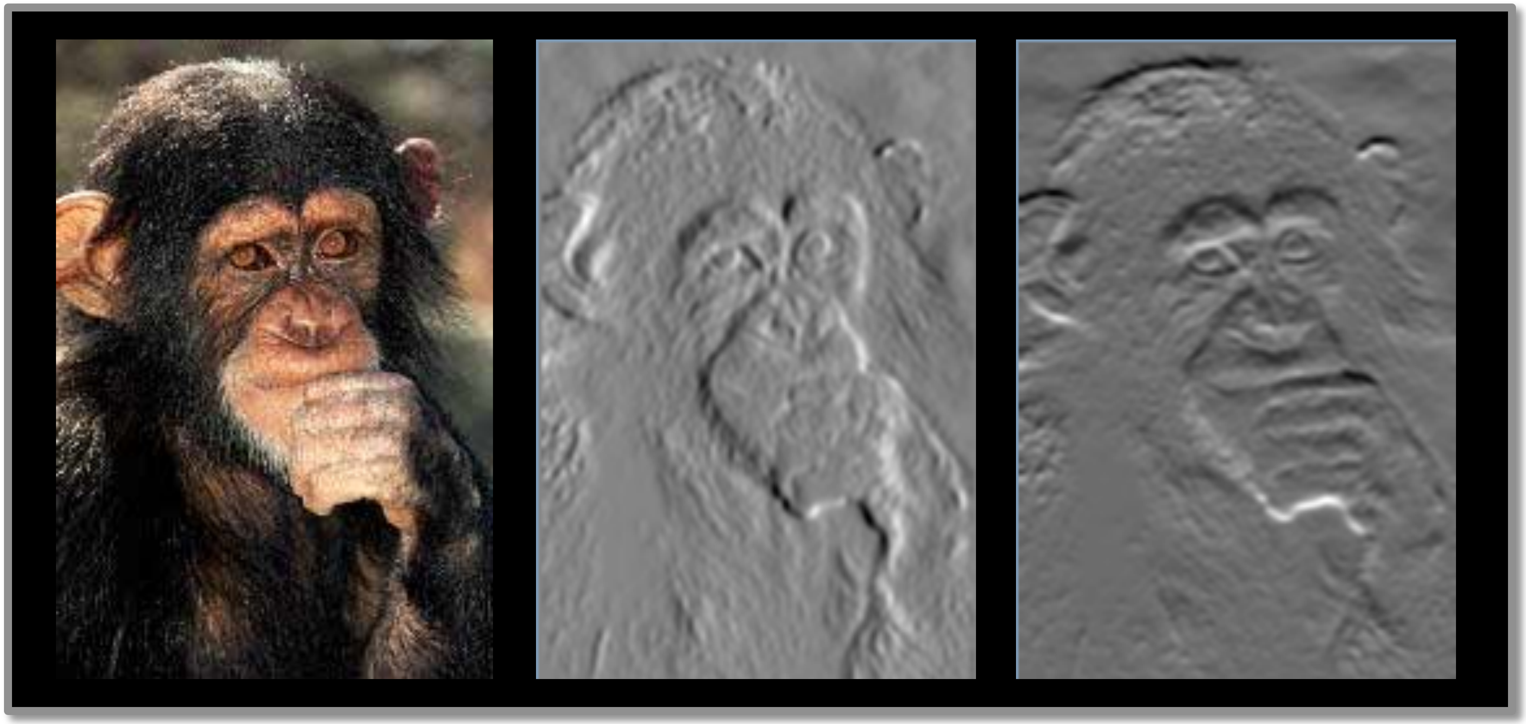
- Facet edge detectors assume a piecewise linear model, and calculate the slope of the planar facet (1st derivative).
  - If we assume that the noise is zero mean, and increases with the square of distance, then convolution with the Sobel Edge Operator is optimal:

$$H = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}, V = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

$$Mag = \sqrt{H^2 + V^2}, \quad \tan\theta = H/V$$

Colorado State University

Examples of Facet Edges



Source Dx Image Dy Image

Properties of Facet Edges/Masks

- Magnitude = $(dx^2 + dy^2)^{1/2}$
- Orientation = $\tan^{-1} dy/dx$
- Dy/Dx responses are signed
- Edges tend to be “thick”
- Edge Masks: sum of weights is zero
- Smoothing masks: sum of weights is one

Symbolic Edge Detection

- Although Sobel edges are optimal estimators for the slope of a planar facet, as symbols they:
  - Are continuous; need to be thresholded
  - May be "thick"; need to be localized
  - Are isolated; need to be grouped into longer lines
- If they correspond to scene structure (e.g. discontinuities), we need a model of how scene structures map to images.

Canny Edge Detection (Step 1)

- To maximize the likelihood of finding step-edges,
  1. Smooth image with a Gaussian filter
     - Size is determined by noise model
  2. Compute image gradients over the same size mask

- The bigger the mask, the better detection is but the worse localization is...

Canny Edge Detection (step 2)

- Non-maximal suppression
  - So far, edges are still “thick”
  - For every edge pixel:
    - 1) Calculate direction of edge (gradient)
    - 2) Check neighbors in edge direction
    - If either neighbor is “stronger”, set edge to zero.

Drexel Tutorial -
http://www.pages.drexel.edu/~weg22/can_tut.html

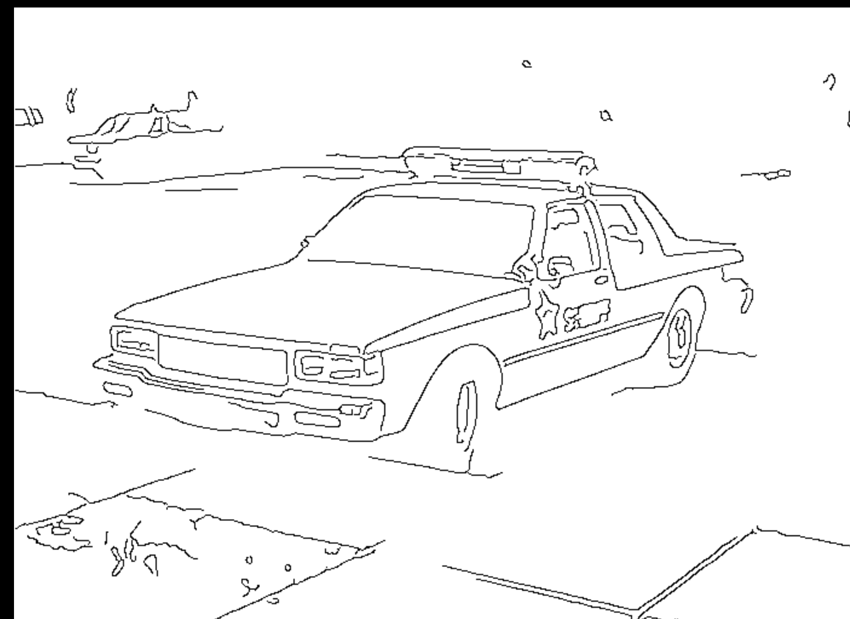
Canny Edge Detection (step 3)

- We still have continuous values that we need to threshold
- Algorithm takes <u>two</u> thresholds: high & low
  - Any pixel with edge strength above the high threshold is an edge
  - Any pixel above the low threshold and next to an edge is an edge
- Iteratively label edges
  - they "grow out" from high points.
  - This is called hysteresis.

Colorado State University

Canny Example

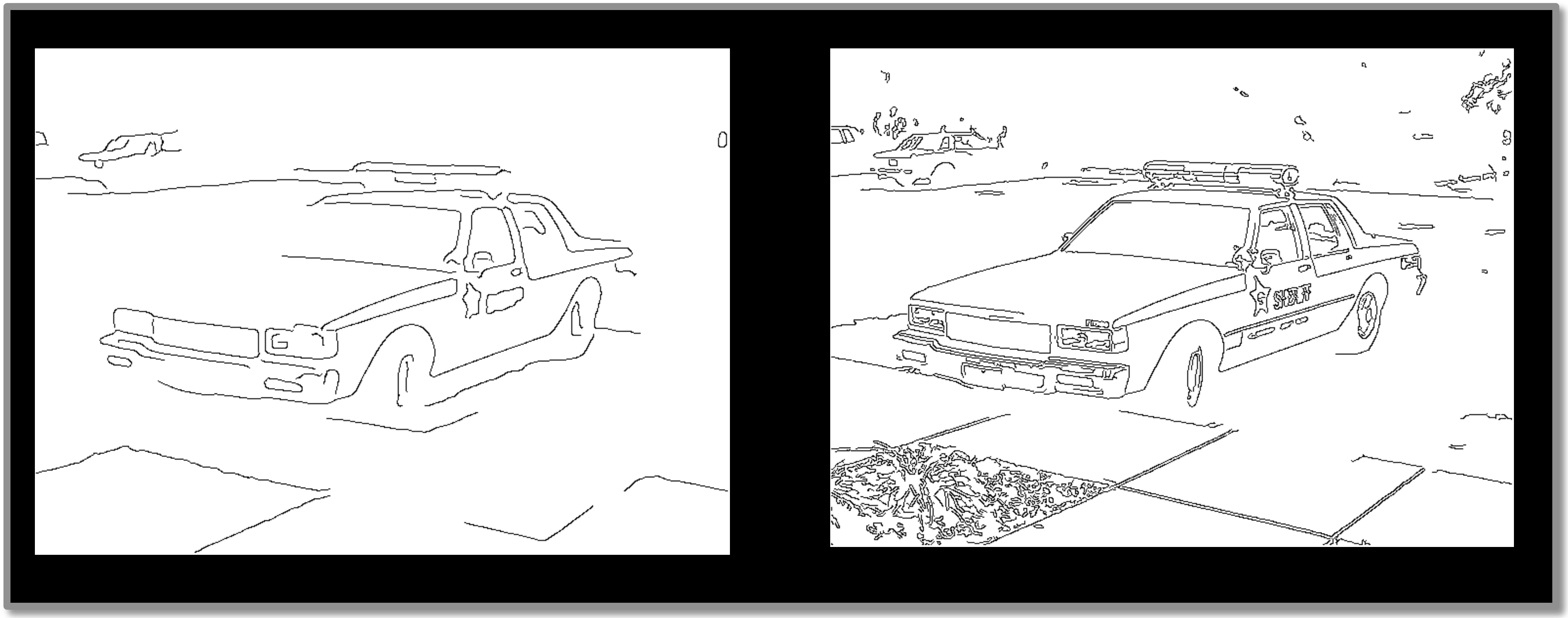
Source image



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

Colorado State University

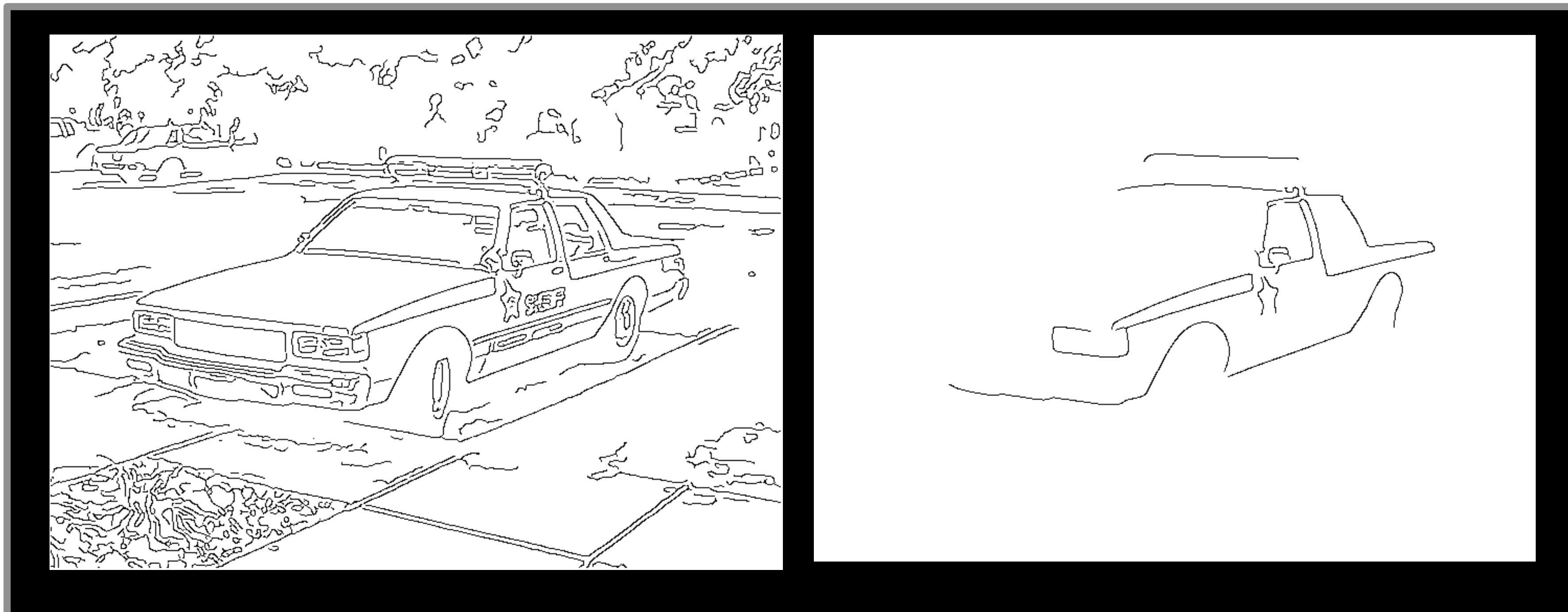
Canny Example (cont.)



Sigma = 3.0
low = 0.4, high = 0.9

Sigma = 1.0
low = 0.4, high = 0.9

Canny Example (III)



Sigma = 2.0
low = 0.4, high = 0.6

Sigma = 2.0
low = 0.4, high = 0.99

Canny Example (IV)

Sigma = 2.0
low = 0.2, high = 0.9

Sigma = 2.0
low = 0.6, high = 0.9

2nd Order Edge - Laplacians

- Alternative approach is to look for zero crossings of the (approximation to) the second derivative.

$$\begin{vmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{vmatrix}$$

- Nice overview http://homepages.inf.ed.ac.uk/rbf/HIPR2/log.htm

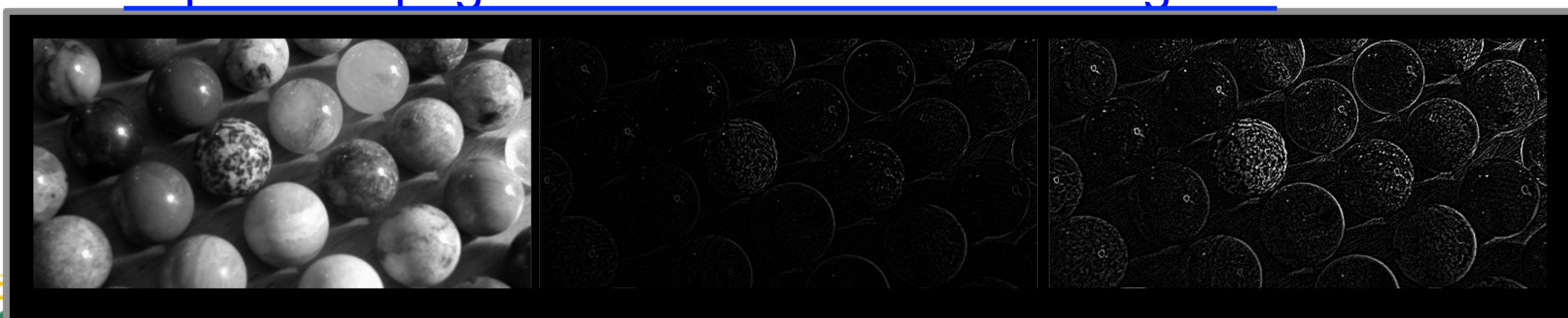
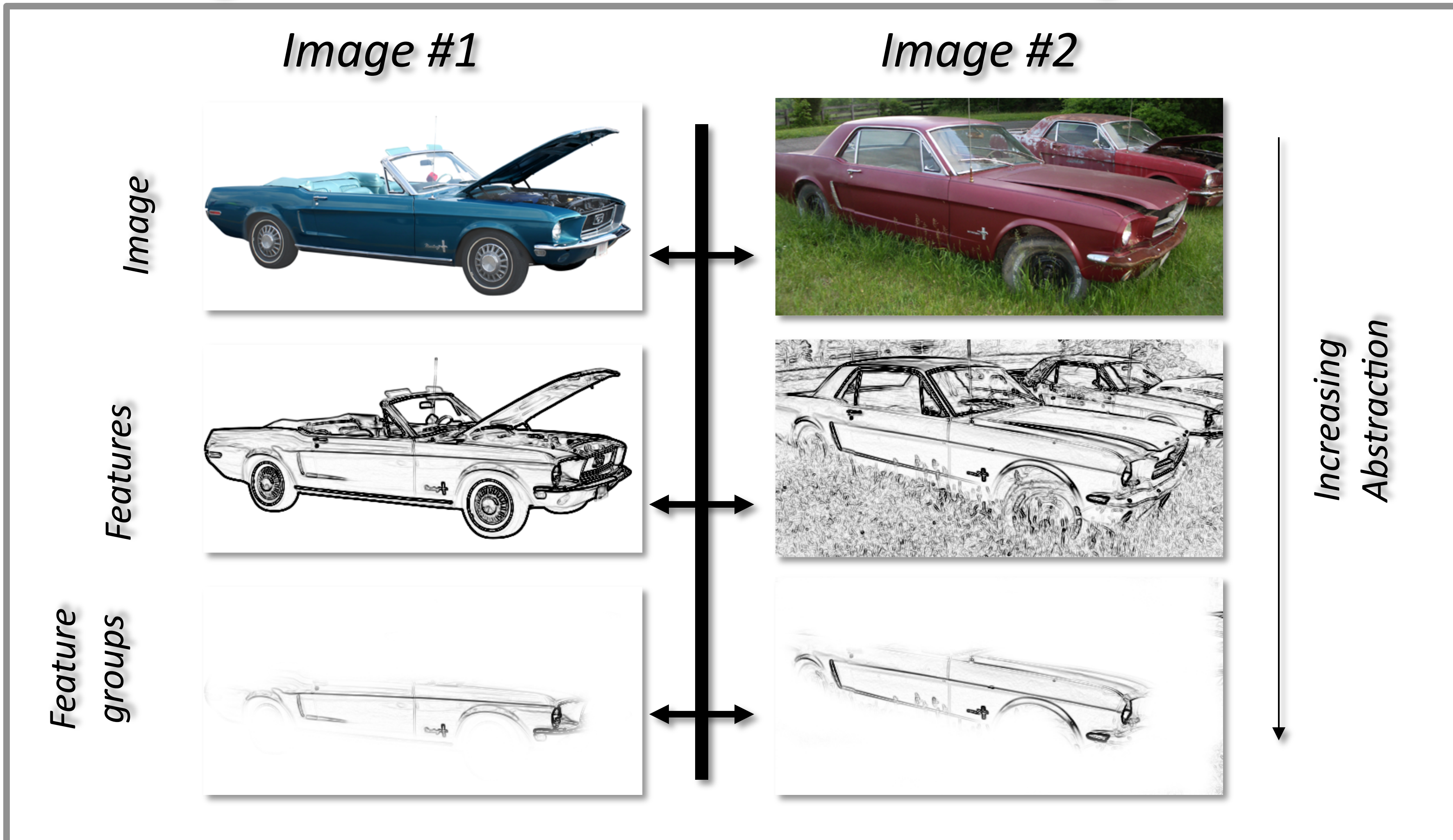


Image Contents Matching



Some Photo Shop liberties have been taken to illustrate the larger point ☺

Hierarchical Feature Extraction

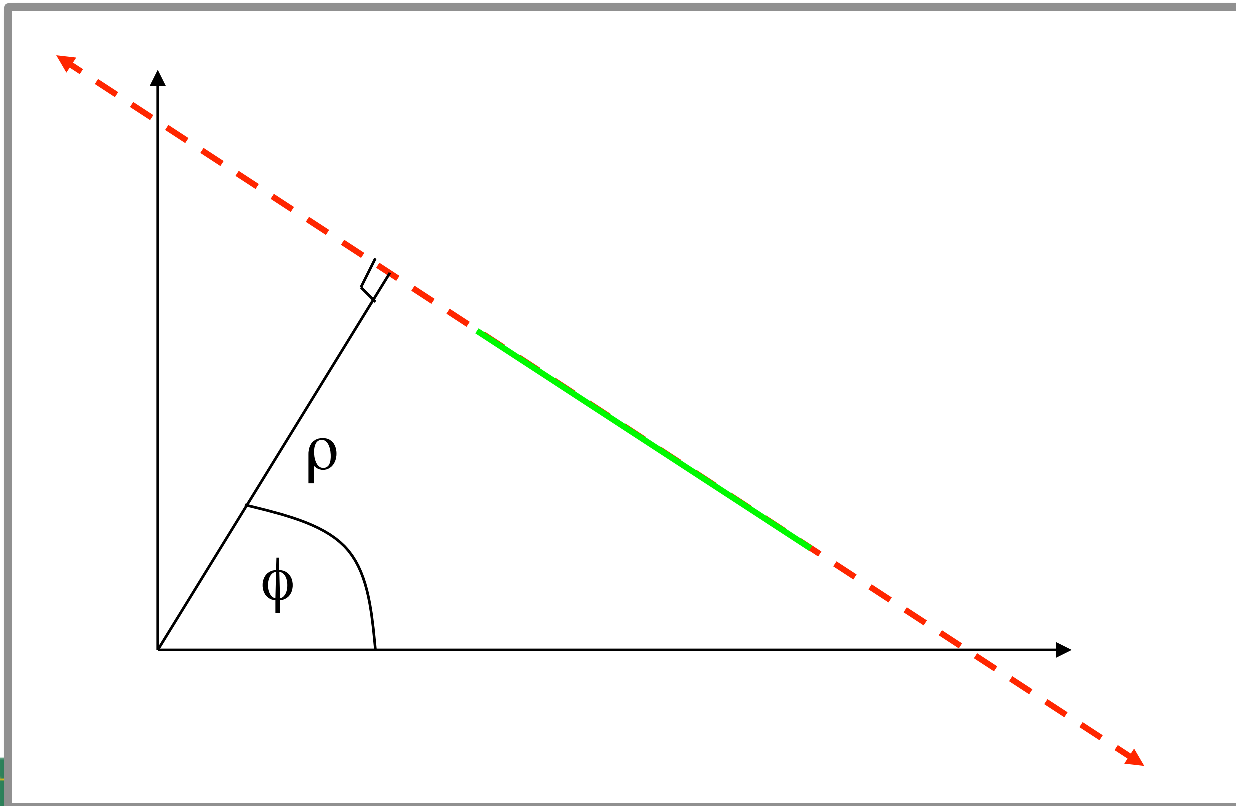
- Most features are extracted by combining a small set of primitive features (edges, corners, regions)
  - Grouping: which pixels form an edges/corners/curves group?
  - Model Fitting: what structure best describes the group?
- Simple example: The Hough Transform
  - Groups points into lines
  - (patented 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 -- *compact, but problems with vertical lines*
  - $(x_0, y_0) + t(x_1, y_1)$ -- *your raytracer used this form, but it is highly redundant (4 free parameters)*
  - ax + by + c = 0 -- *Bresenham's uses this form. Still redundant (3 free parameters)*
- How else might you represent a line?

Hough Grouping (cont.)

- Represent infinite lines as (ϕ,ρ):

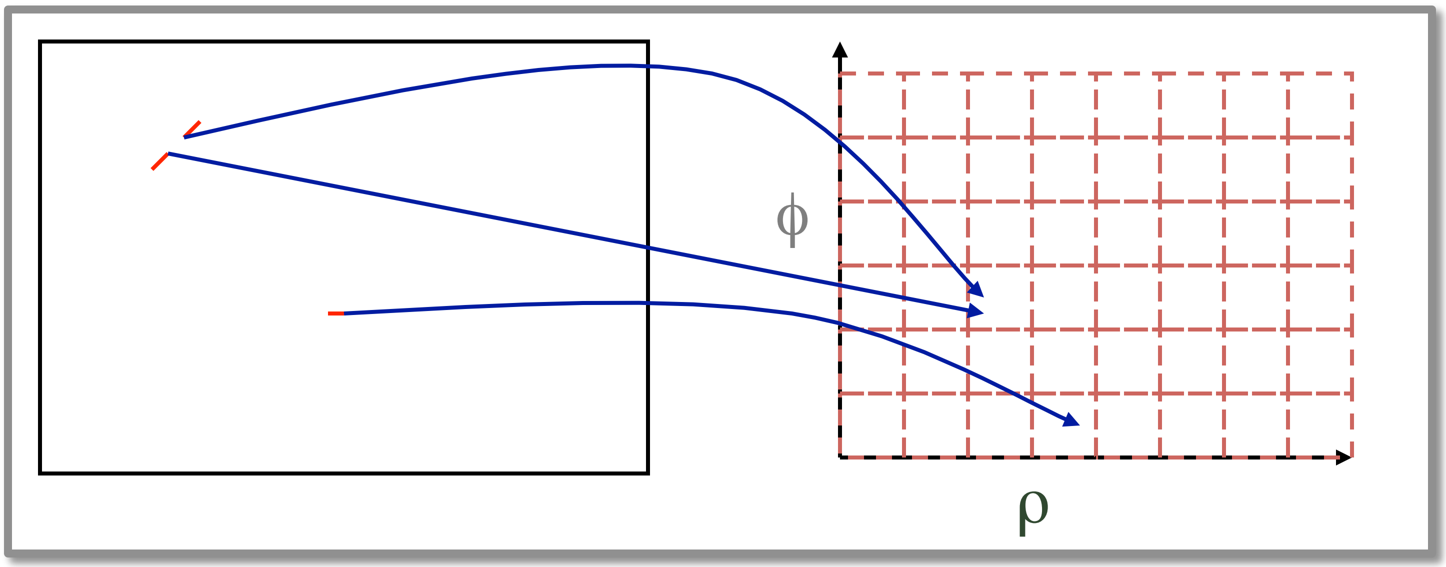


Hough Grouping (III)

- Why? This representation is:
  - Small: only two free parameters (like y=mx+b)
  - Finite in all parameters : $0 <= \rho < \sqrt{(row^2+col^2)}$, $0 <= \phi < 2\pi$
  - Unique: only one representation per line
- General Idea:
  - The Hough space (ϕ,ρ) represents every possible line segment
  - Next step - use discrete 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.



- Co-linear edges map to one bucket in Hough space.

Hough Grouping: Edges

- Reduces line grouping to peak detection
  - Each edge votes for a bucket (line)
  - # of votes equates to support
    - The # of participating edges.
  - 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 Fitting

- After finding the peaks in the Hough Transform - still two potential problems:
  - – Resolution limited by bucket size.
  - – Infinite lines, not line segments
- Both of these problems can be fixed,
  - – If you kept a linked list of edges (not just #)
  - – 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

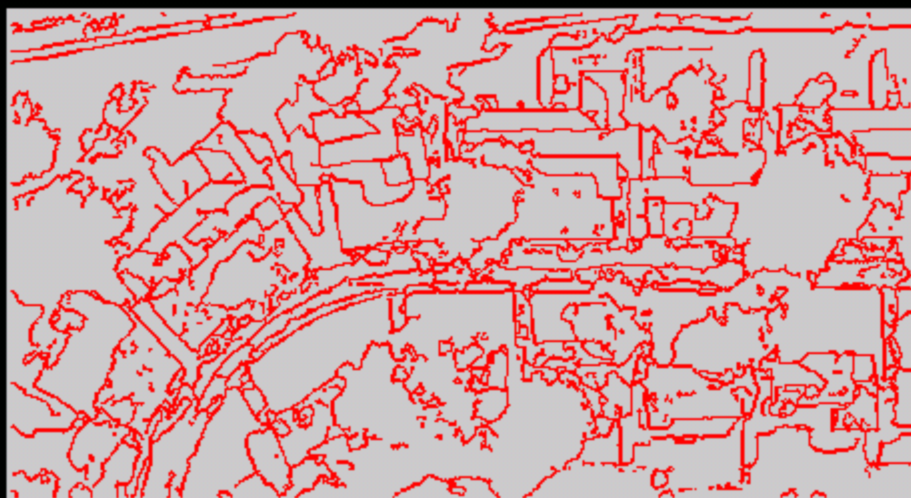
- In *n* dimensions, compute the Eigenvalues & Eigenvectors and take the Eigenvector associated with the largest Eigenvalue.
- In 2 dimensions, its simpler:
  - for p points (x,y),

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

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

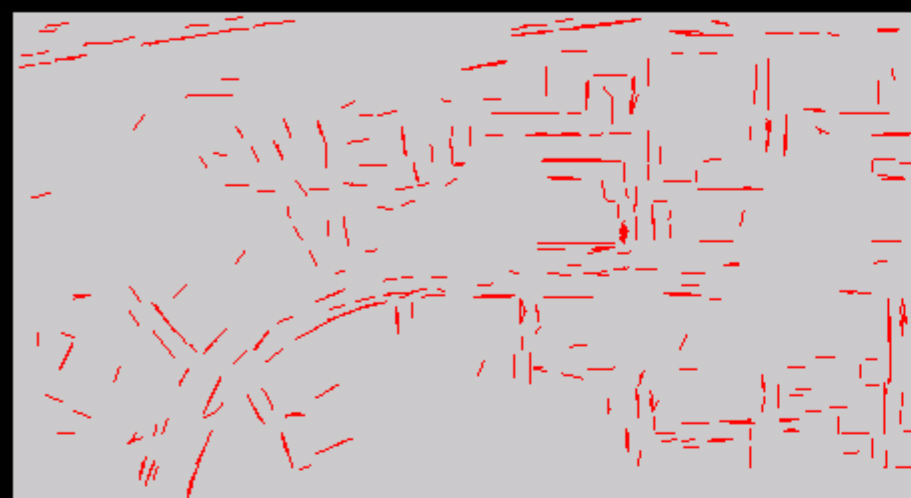
Hough Example

Hough Example (II)



Edge data

Line data



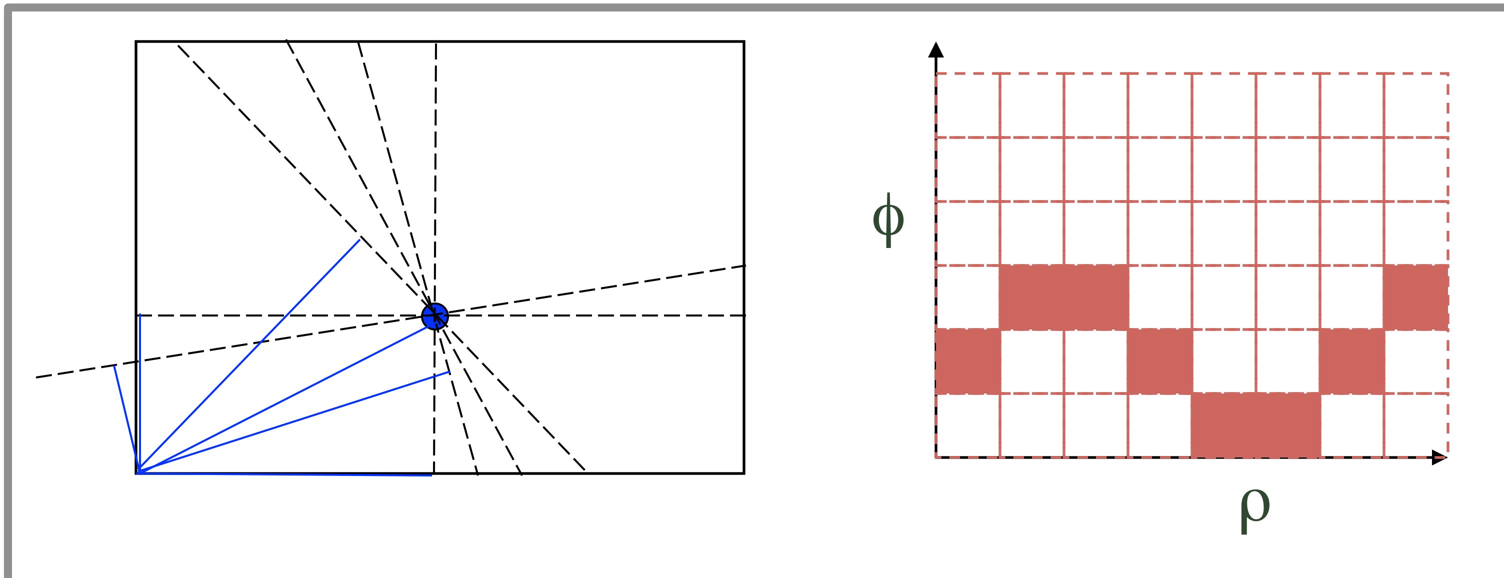
"Vote Early and Often"

Underconstrained Cases

- In the case of points (rather than edges)
  - Points have locations but not orientations
  - A point is consistent with infinitely many lines
    - Every line that passes through the point
  - 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.

Under constrained point voting

- Edge points are consistent with many lines.



They map to many buckets in Hough space

Applet:

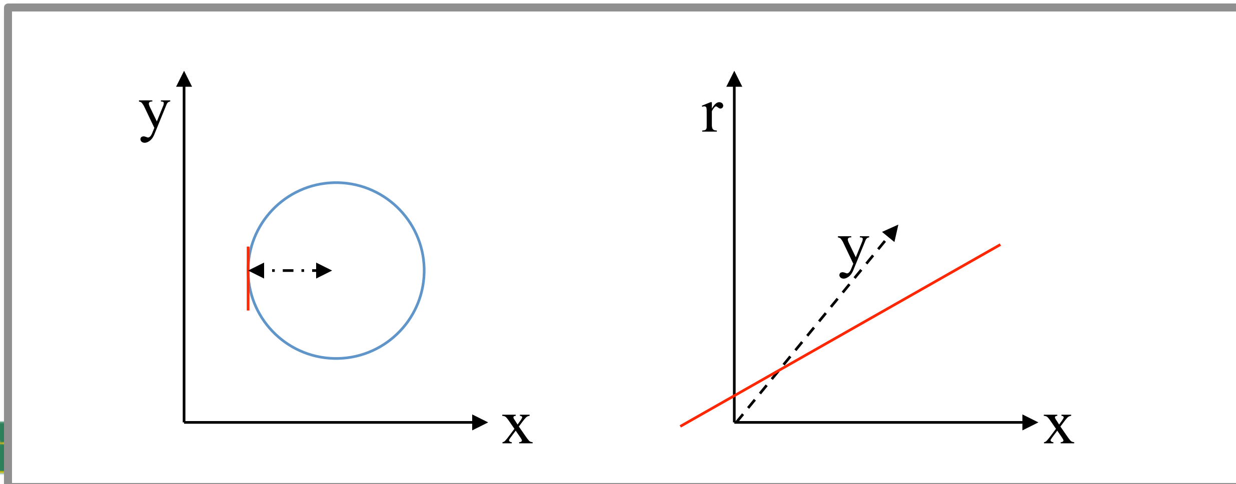
http://www.dis.uniroma1.it/~iocchi/slides/icra2002/java/hough.html

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.

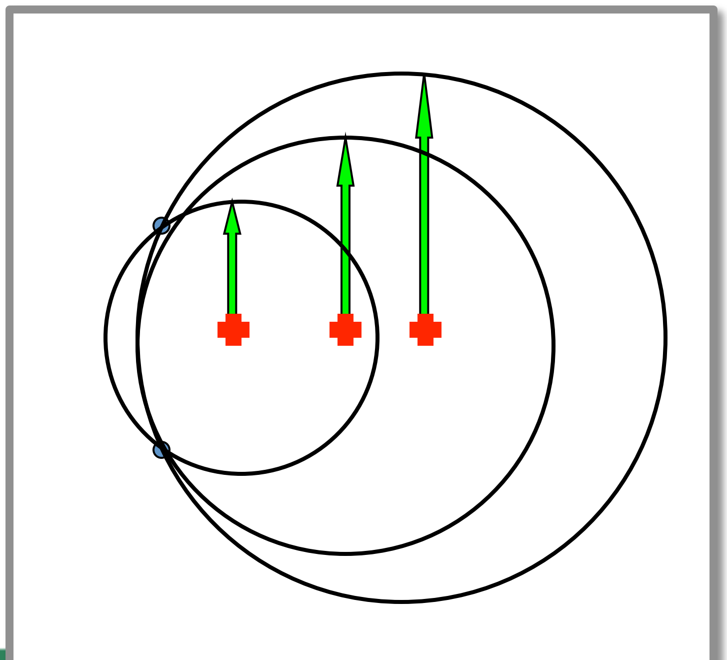
Circles (cont.)

- So, every edge is consistent with an infinite number of circles.
- These circles lie on a line in 3D parameter space - Vote for all of them.
  - This is 3D scan line conversion -- Bresenham!



Circles - Two Point Method

- Consider all pairs of edge points
  - In practice, enforce a minimum separation.



Pairs of edges vote for combinations of radius and image centers