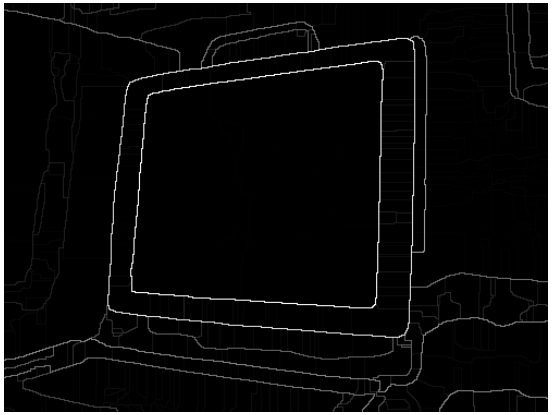
计算机视觉
Computer Vision

-- Structures 2

钟 凡
zhongfan@sdu.edu.cn

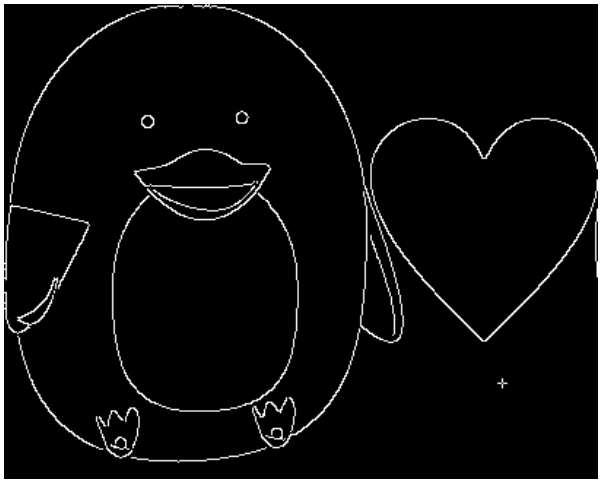
图像结构

- 如何提取非二值图像中的结构信息?
  - 边缘
  - 基元（直线、圆等）

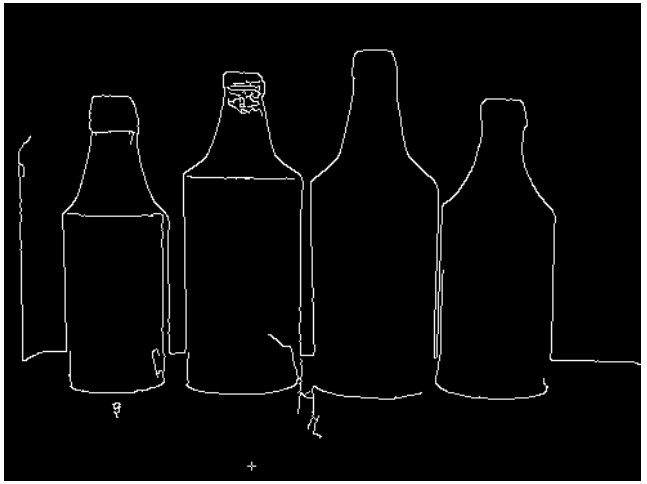


图像边缘检测

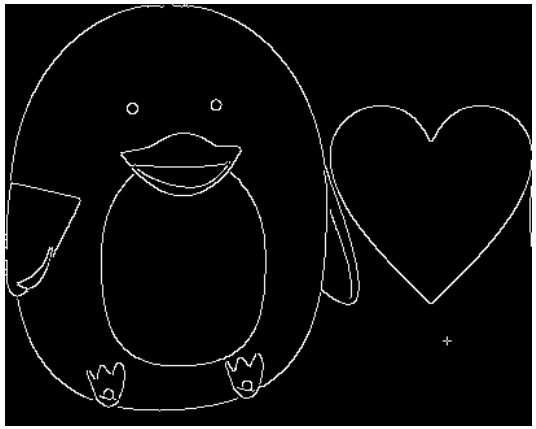
图像边缘

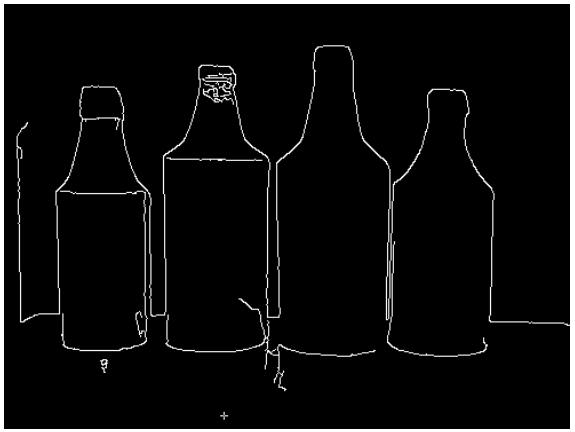


图像边缘



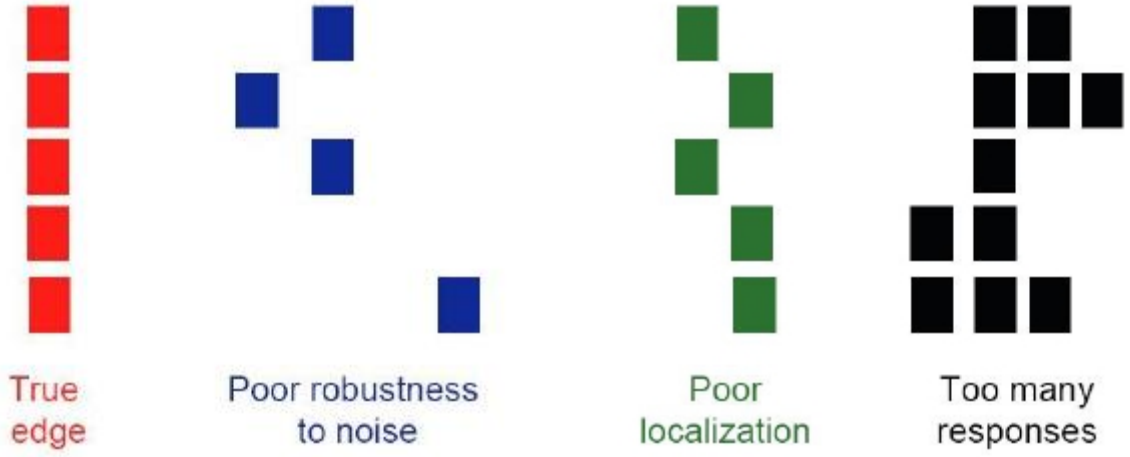
边缘的特点&要求?





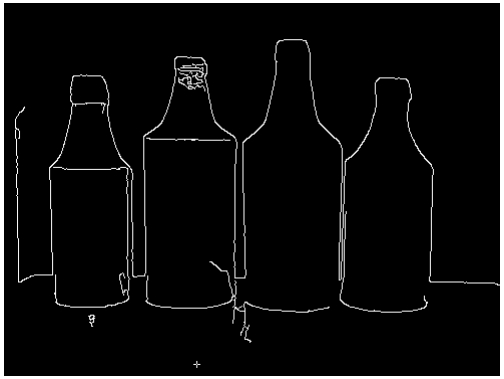
边缘的特点&要求

- **检测准确：** 假阳（false positive)、假阴（false negative)最少
- **定位准确：** 与真实边缘对齐
- **单像素响应：** 边缘宽度为单像素



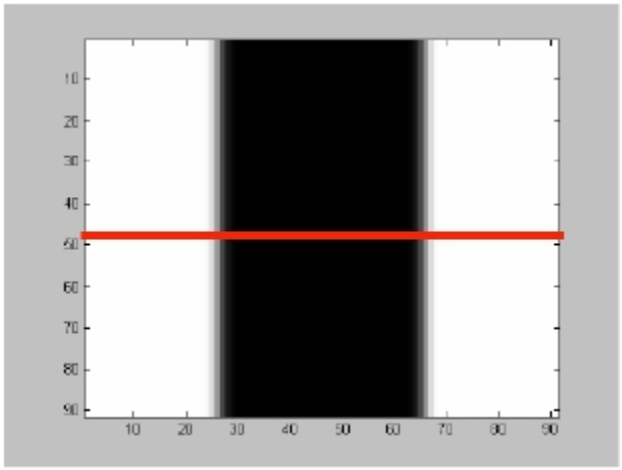
基本步骤

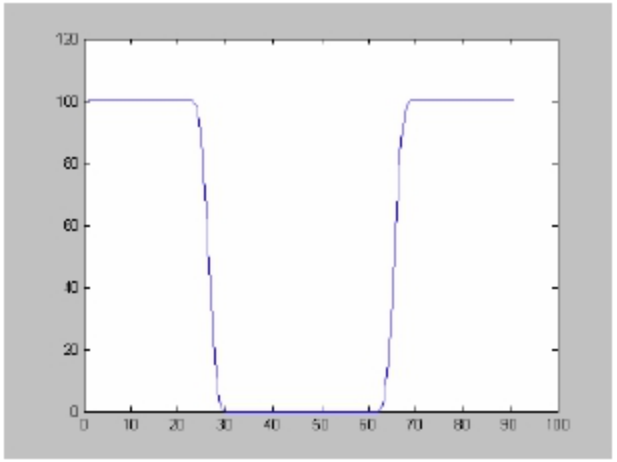
计算边缘响应



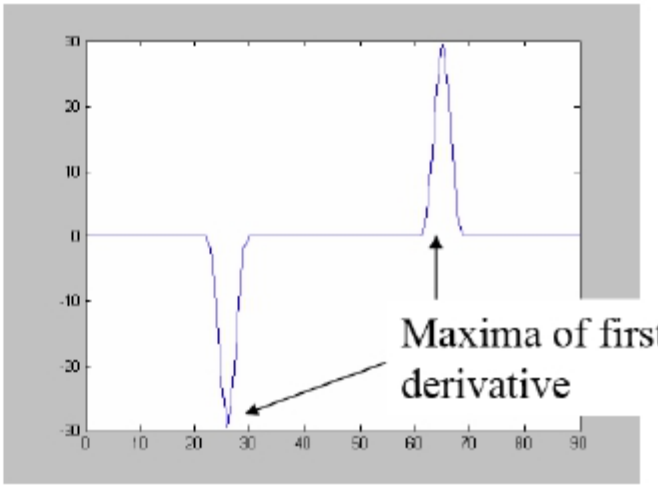
边缘定位

Derivatives and Edges…

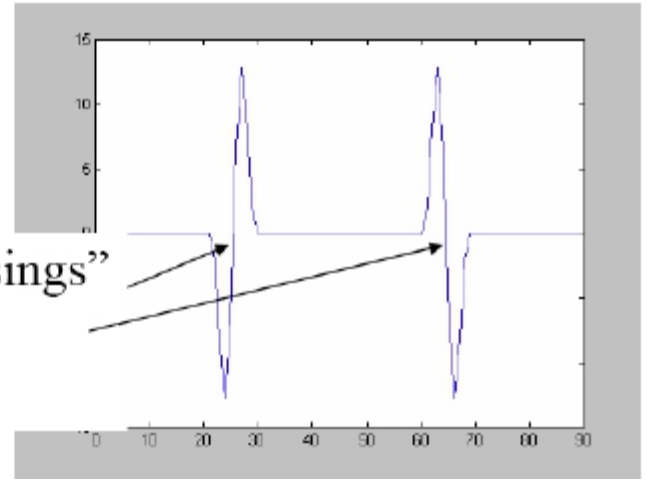




1st derivative



2nd derivative



Differentiation and Convolution

- For the 2D function $f(x,y)$, the partial derivative is:

$$\frac{\partial f(x,y)}{\partial x} = \lim_{\varepsilon \to 0} \frac{f(x+\varepsilon, y) - f(x,y)}{\varepsilon}$$

- For discrete data, we can approximate this using finite differences:

$$\frac{\partial f(x,y)}{\partial x} \approx \frac{f(x+1, y) - f(x,y)}{1}$$

- To implement the above as convolution, what would be the associated filter?

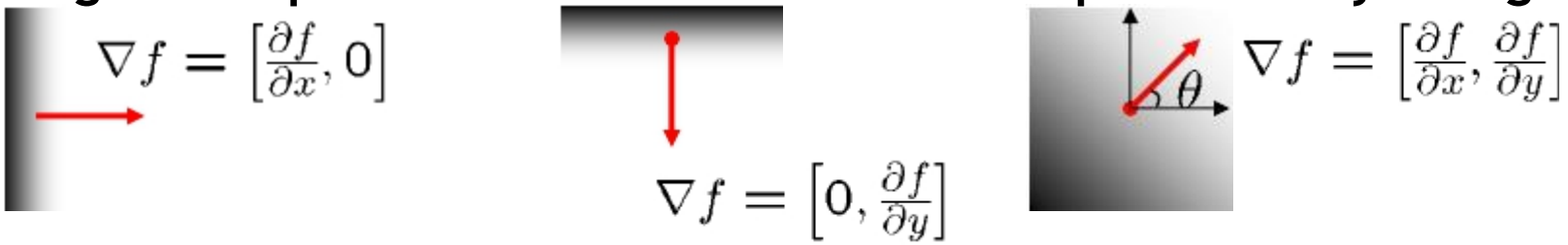
1	-1

Image Gradient

- The gradient of an image:

$$\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]$$

- The gradient points in the direction of most rapid intensity change



- The gradient direction (orientation of edge normal) is given by:

$$\theta = \tan^{-1}\left(\frac{\partial f}{\partial y} / \frac{\partial f}{\partial x}\right)$$

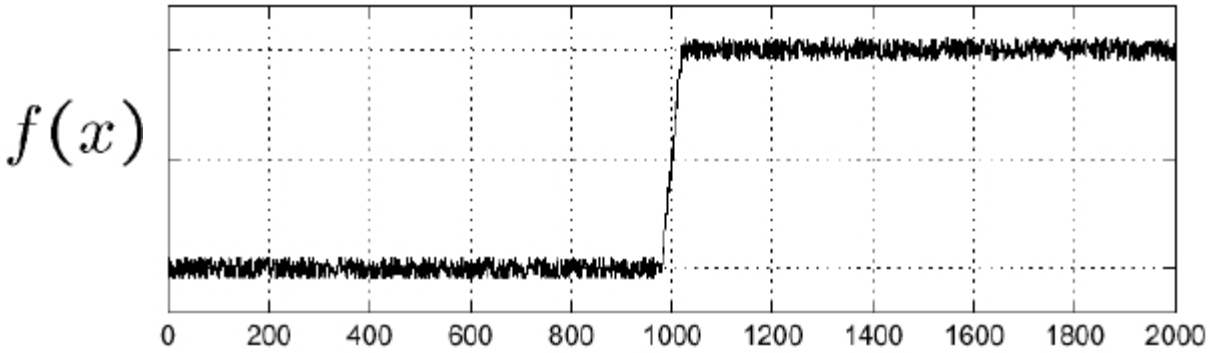
- The edge strength is given by the gradient magnitude

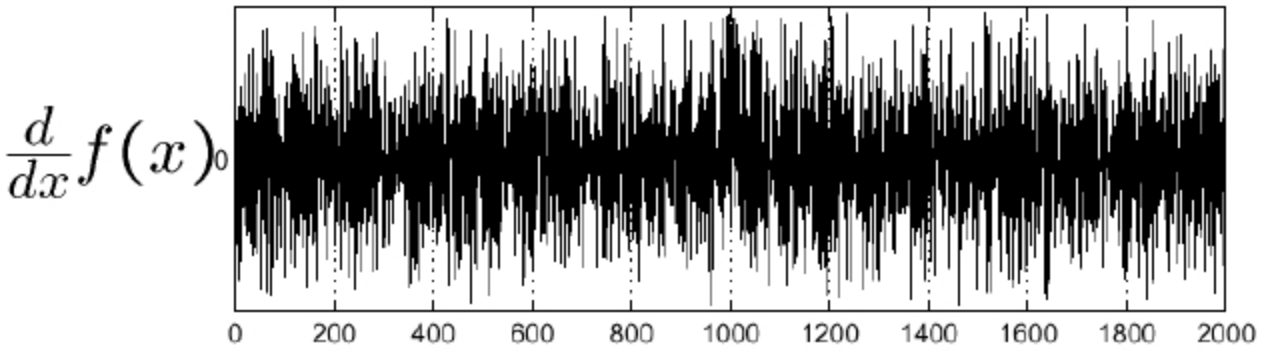
$$\|\nabla f\| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$

Slide credit: Steve Seitz

Effect of Noise

- Consider a single row or column of the image
  - Plotting intensity as a function of position gives a signal

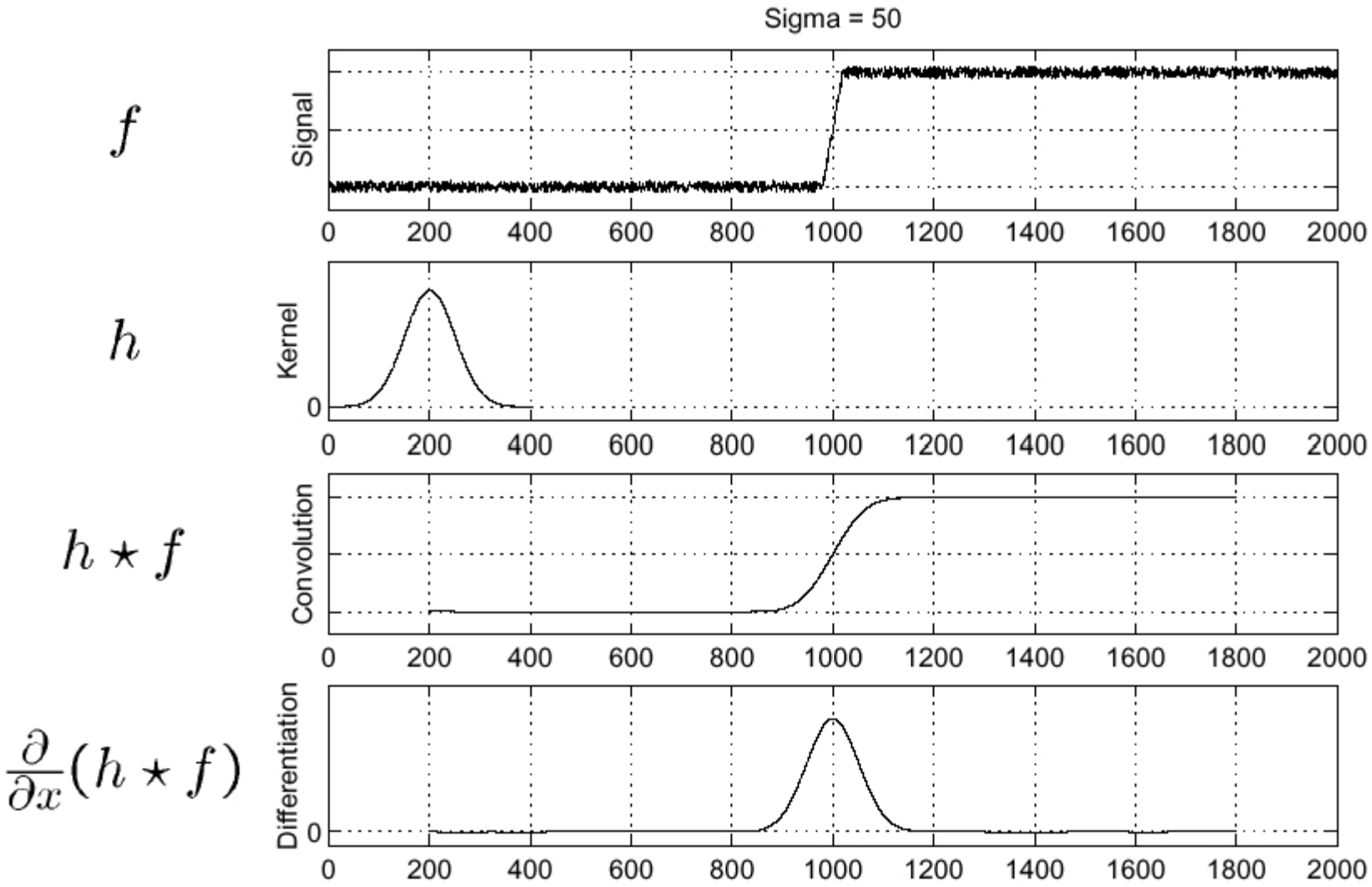




Where is the edge?

Slide credit: Steve Seitz

Solution: Smooth First



Where is the edge?

Look for peaks in $\frac{\partial}{\partial x}(h \star f)$

Slide credit: Steve Seitz

Derivative Theorem of Convolution

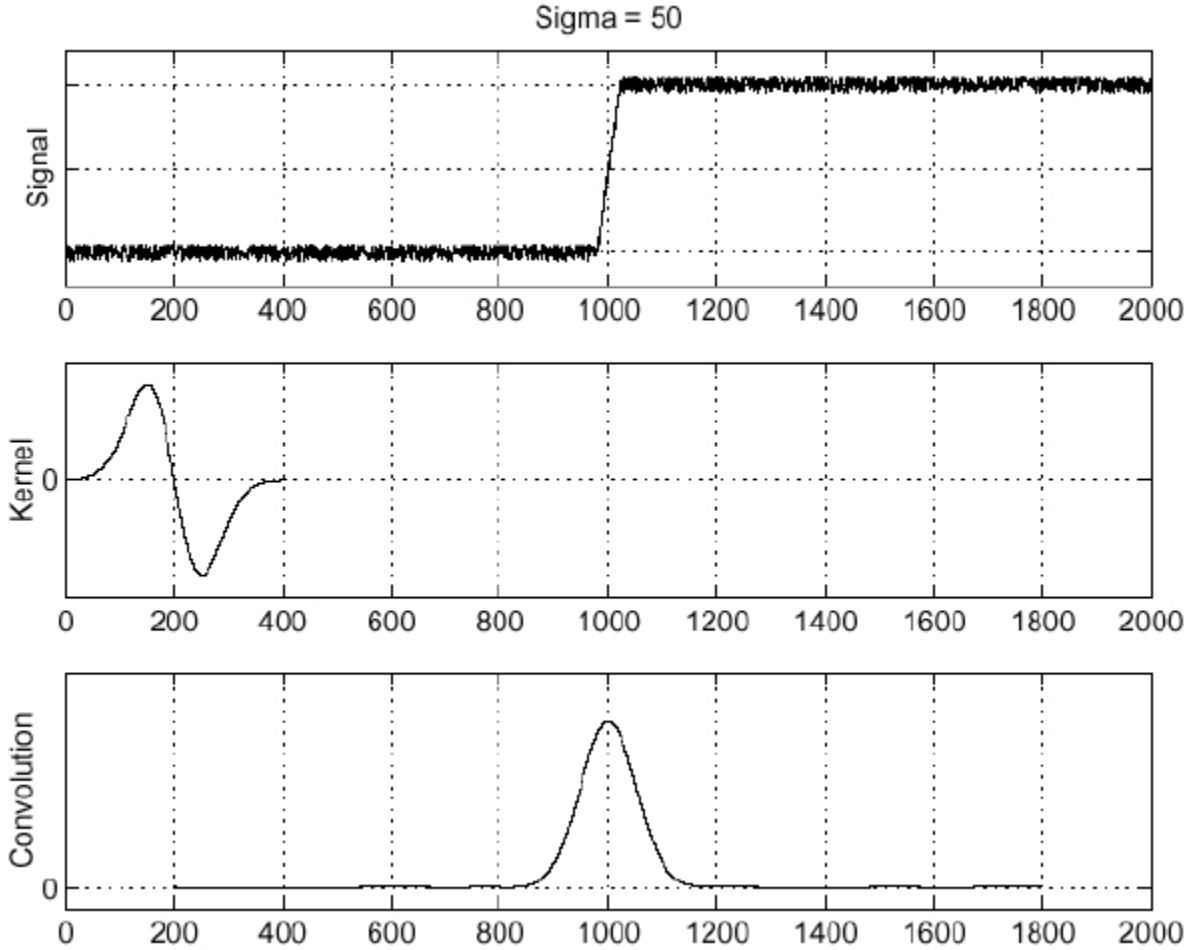
$$\frac{\partial}{\partial x}(h \star f) = (\frac{\partial}{\partial x}h) \star f$$

- **Differentiation property of convolution.**

f

$\frac{\partial}{\partial x}h$

$(\frac{\partial}{\partial x}h) \star f$

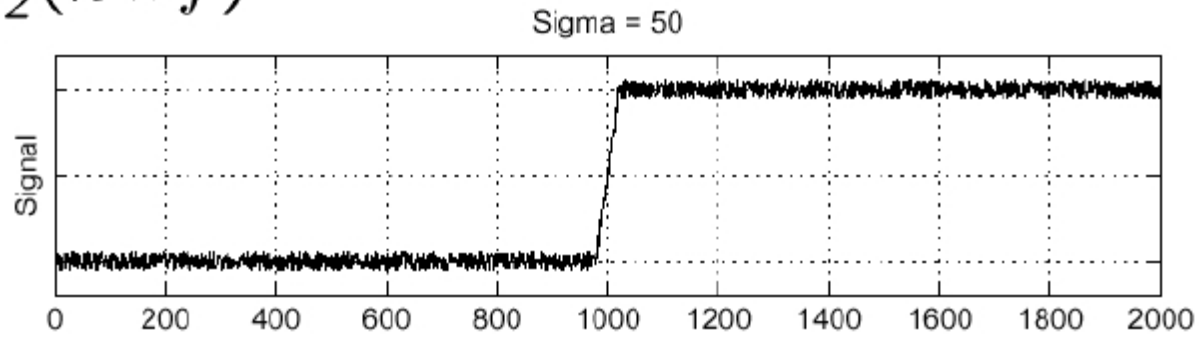


Slide credit: Steve Seitz

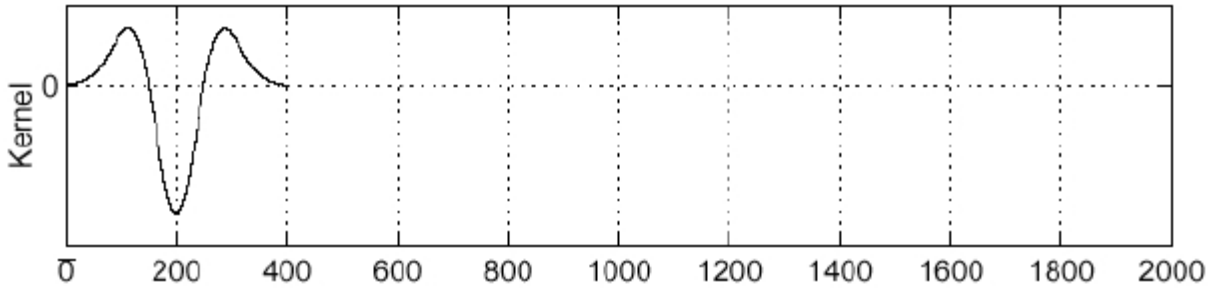
Laplacian of Gaussian (LoG)

- Consider $\frac{\partial^2}{\partial x^2}(h \star f)$

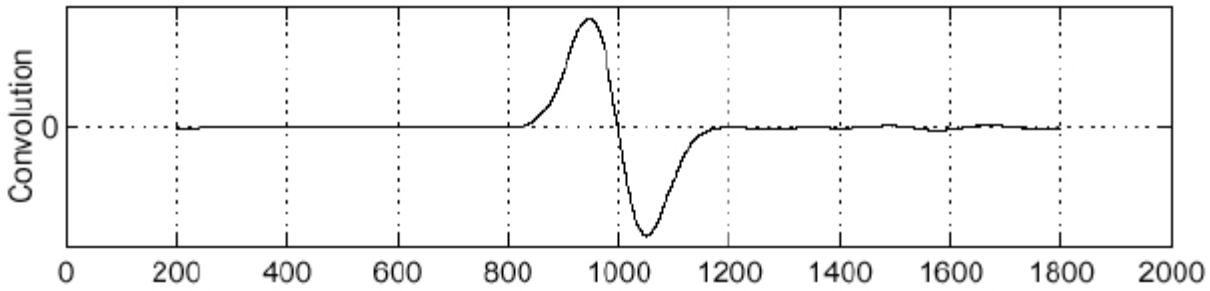
f



$\frac{\partial^2}{\partial x^2}h$



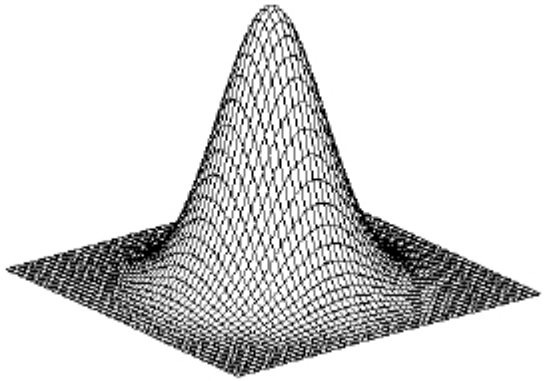
$(\frac{\partial^2}{\partial x^2}h) \star f$



Where is the edge? **Zero-crossings of bottom graph**

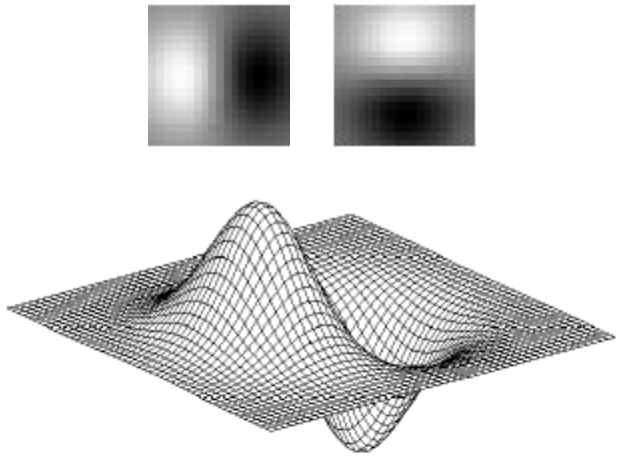
Slide credit: Steve Seitz

Summary: 2D Edge Detection Filters



Gaussian

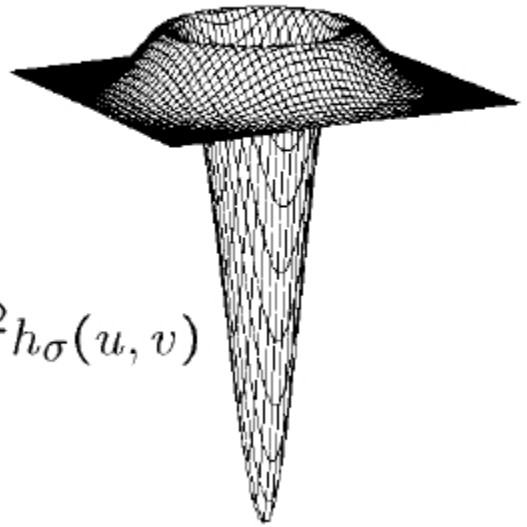
$$h_\sigma(u,v) = \frac{1}{2\pi\sigma^2} e^{-\frac{u^2+v^2}{2\sigma^2}}$$



Derivative of Gaussian

$$\frac{\partial}{\partial x} h_\sigma(u,v)$$

Laplacian of Gaussian



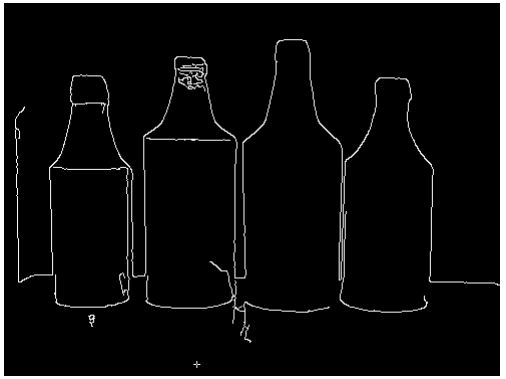
$$\nabla^2 h_\sigma(u,v)$$

- ∇^2 is the Laplacian operator:

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

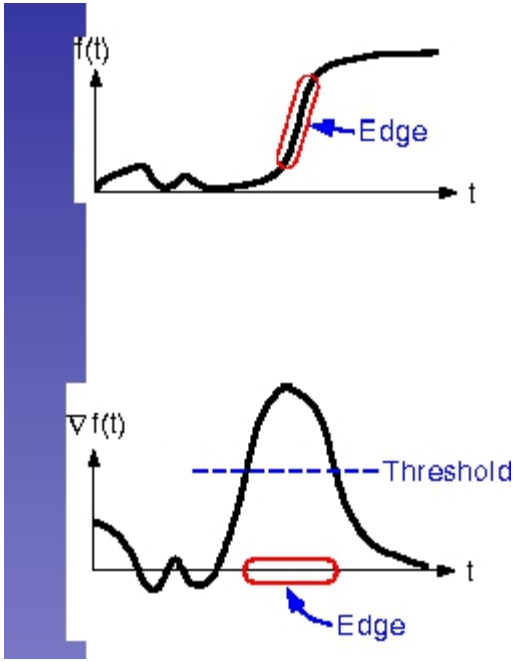
基本步骤

计算边缘响应

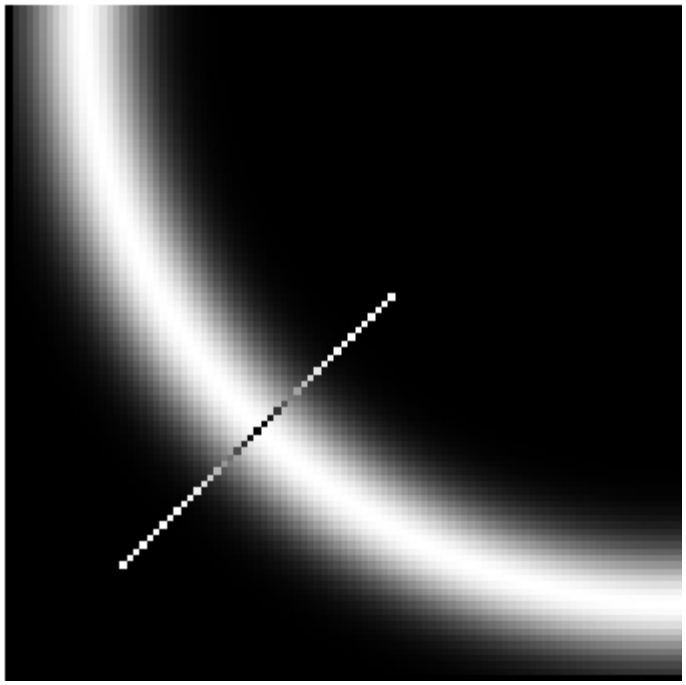


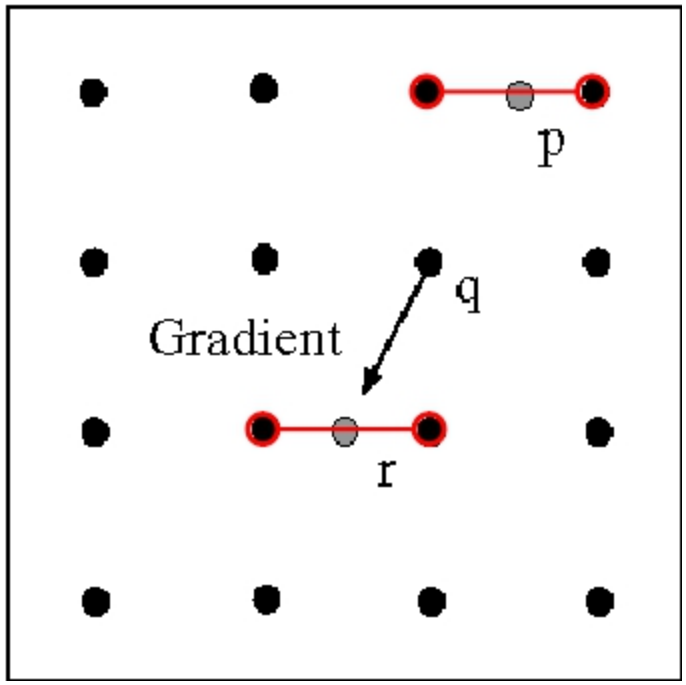
边缘定位?

响应图->边缘



非极大值抑制

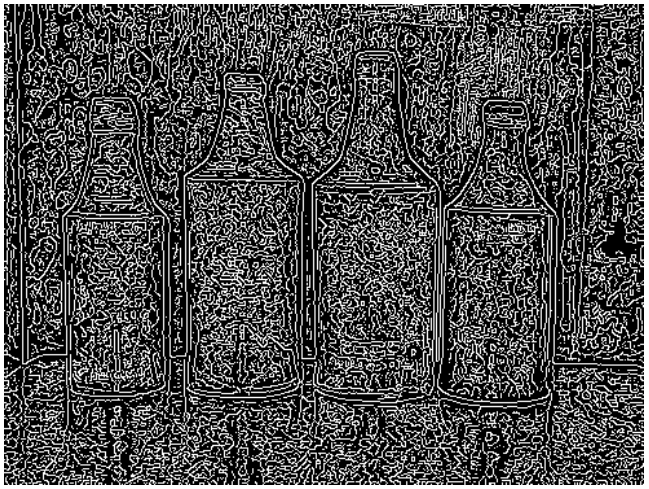




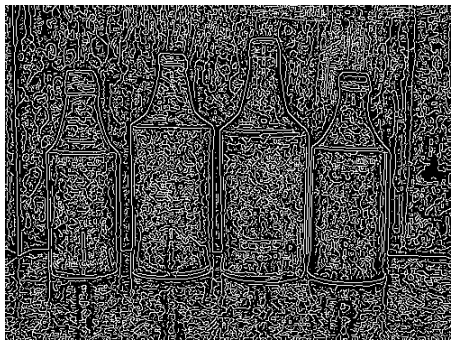
只保留沿梯度方向上响应值的极大值点

实现??

非极大值抑制

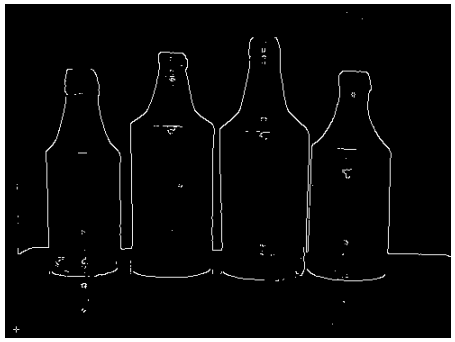


使用阈值



T=0

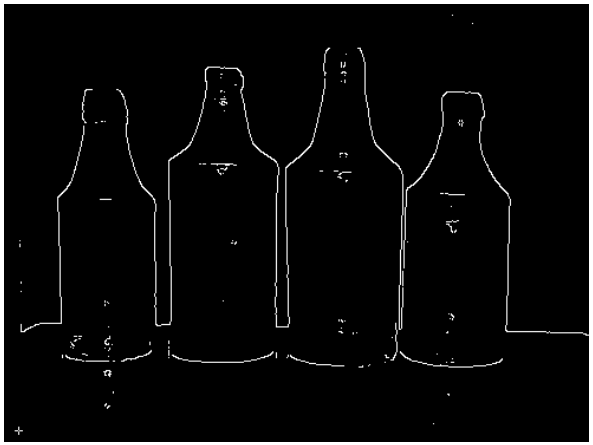
T=50



T=200

阈值选取: 清除杂边&保持弱边

T=50

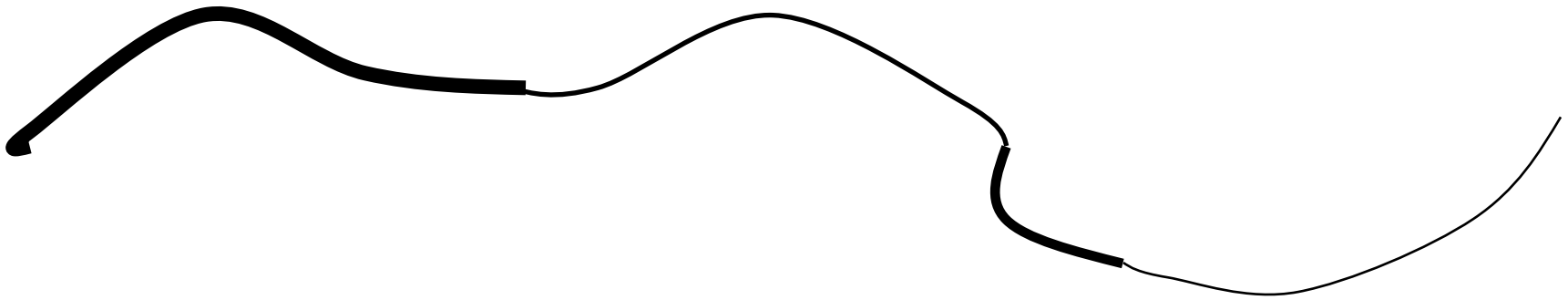


T=200

Solution: Hysteresis Thresholding

- **Hysteresis: A lag or momentum factor**
- **Idea: Maintain two thresholds T_{high} and T_{low}**
  - **Use T_{high} to find strong edges to start edge chain**
  - **Use T_{low} to find weak edges which continue edge chain**
- **Typical ratio of thresholds is roughly**

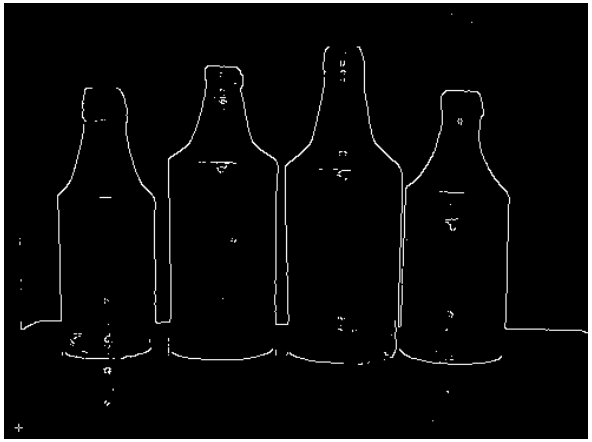
$$T_{high} / T_{low} = 2$$



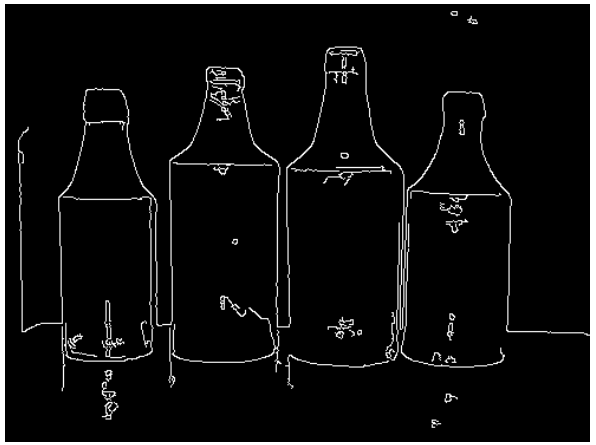
Source: D. Lowe, S. Seitz

阈值选取: 清除杂边&保持弱边

T=50



T=200



T_{high}=200, T_{low}=50

Canny Edge Detector

- Probably the most widely used edge detector in computer vision
- Theoretical model: step-edges corrupted by additive Gaussian noise
- Canny has shown that the first derivative of the Gaussian closely approximates the operator that optimizes the product of *signal-to-noise ratio* and localization.

J. Canny, A Computational Approach To Edge Detection, *IEEE Trans. Pattern Analysis and Machine Intelligence*, 8:679-714, 1986.

Source: Li Fei-Fei

Canny Edge Detector

1. Filter image with derivative of Gaussian
2. Find magnitude and orientation of gradient
3. Non-maximum suppression:
   - Thin multi-pixel wide "ridges" down to single pixel width
4. Linking and thresholding (hysteresis):
   - Define two thresholds: low and high
   - Use the high threshold to start edge curves and the low threshold to continue them

- MATLAB:

```
>> edge(image, 'canny');
>> help edge
```


Source: D. Lowe, L. Fei-Fei

Object Boundaries vs. Edges

Background

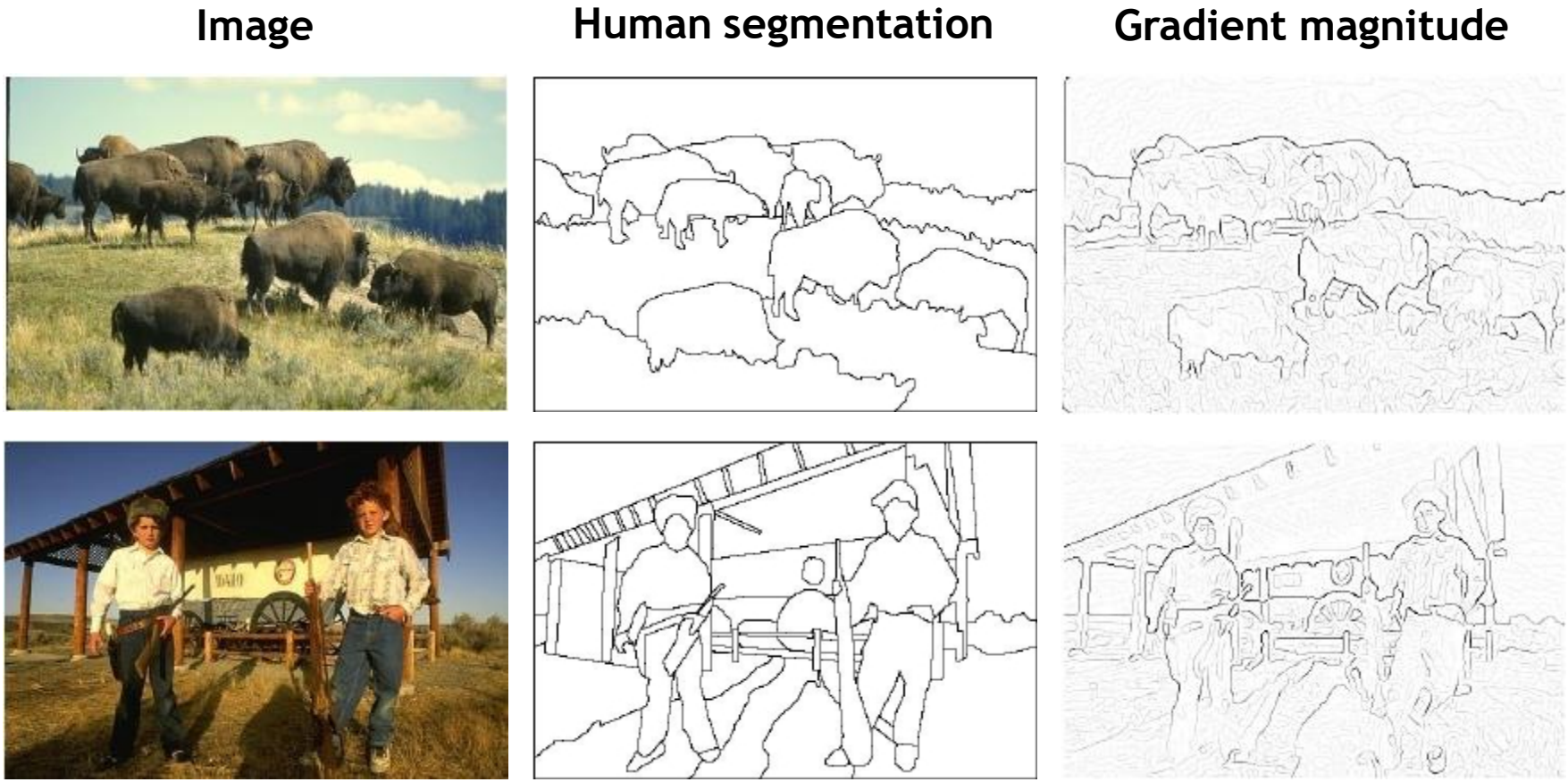
Texture

Shadows

Slide credit: Kristen Grauman

B. Leibe

Edge Detection is Just the Beginning…



- Berkeley segmentation database:
  http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/segbench/

Source: L. Lazebnik

基元检测

基元

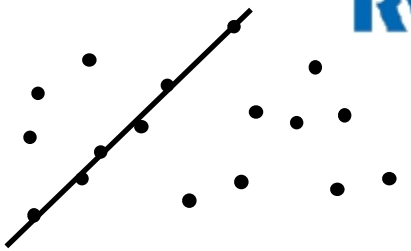
- 直线、圆等基本几何元素

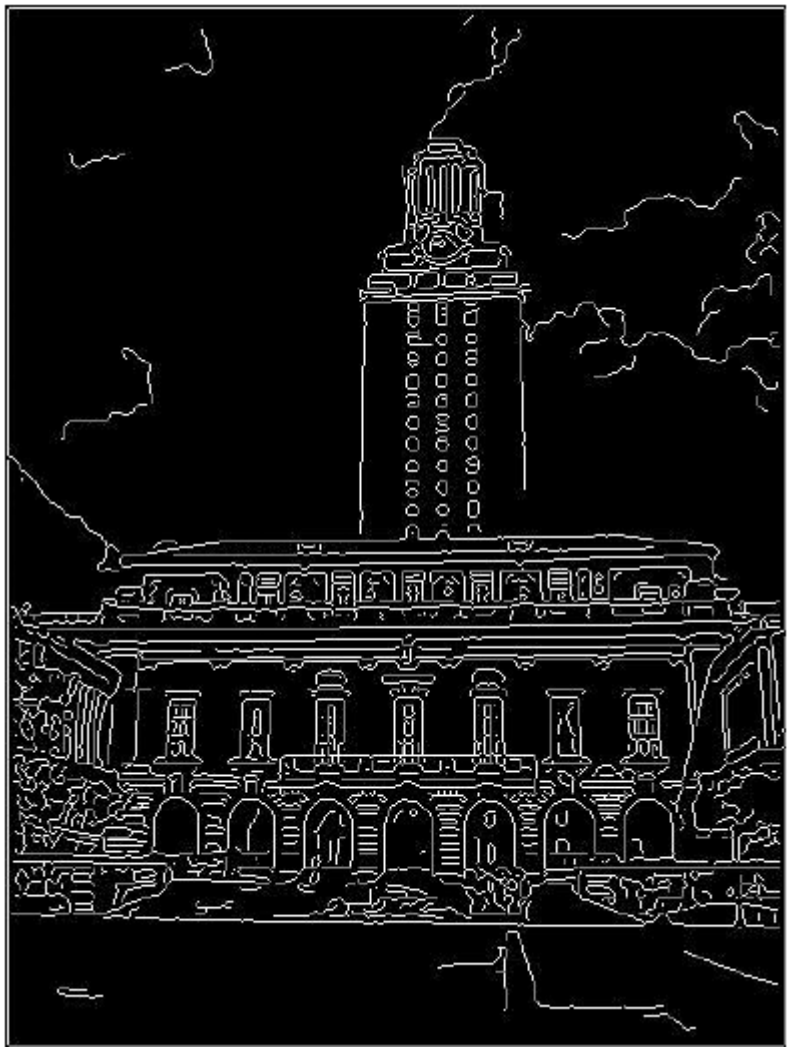
Example: Line Fitting

- **Why fit lines?**
  - Many objects are characterized by presence of straight lines

- **Wait, why aren't we done just by running edge detection?**

Difficulty of Line Fitting





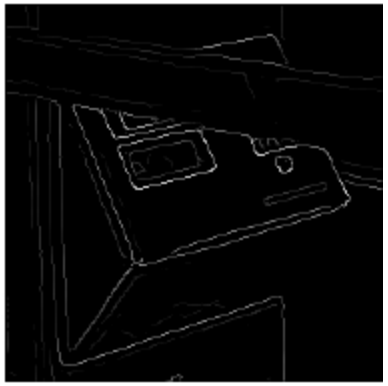
- Extra edge points (clutter), multiple models:
  - Which points go with which line, if any?
- Only some parts of each line detected, and some parts are missing:
  - How to find a line that bridges missing evidence?
- Noise in measured edge points, orientations:
  - How to detect true underlying parameters?

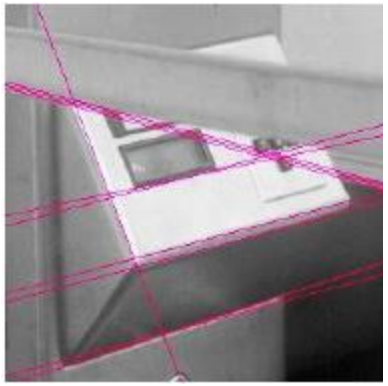
Slide credit: Kristen Grauman

霍夫变换（Hough Transform)

- 给定点集，检测包含的直线；
- 可能存在多条直线；

- 基于投票（Voting)的思想：
  - 每个点投票所有可能经过它的直线
  - 在直线参数空间取票数较多的点





Finding Lines in an Image: Hough Space

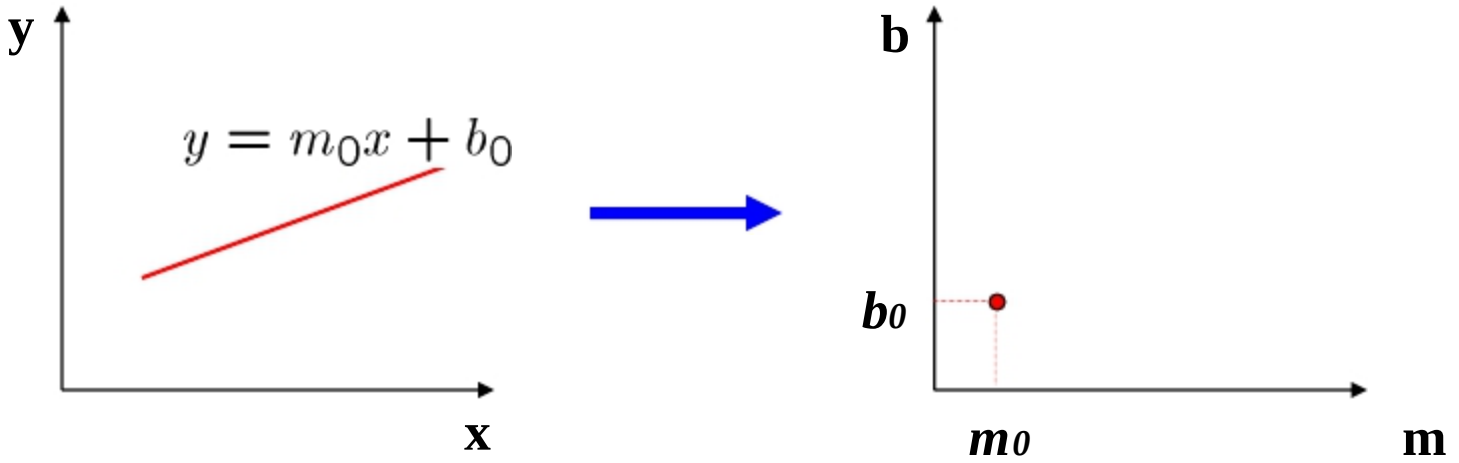


Image space

Hough (parameter) space

- Connection between image (x,y) and Hough (m,b) spaces
  - A line in the image corresponds to a point in Hough space.
  - To go from image space to Hough space:
    - Given a set of points (x,y), find all (m,b) such that $y = mx + b$

Finding Lines in an Image: Hough Space

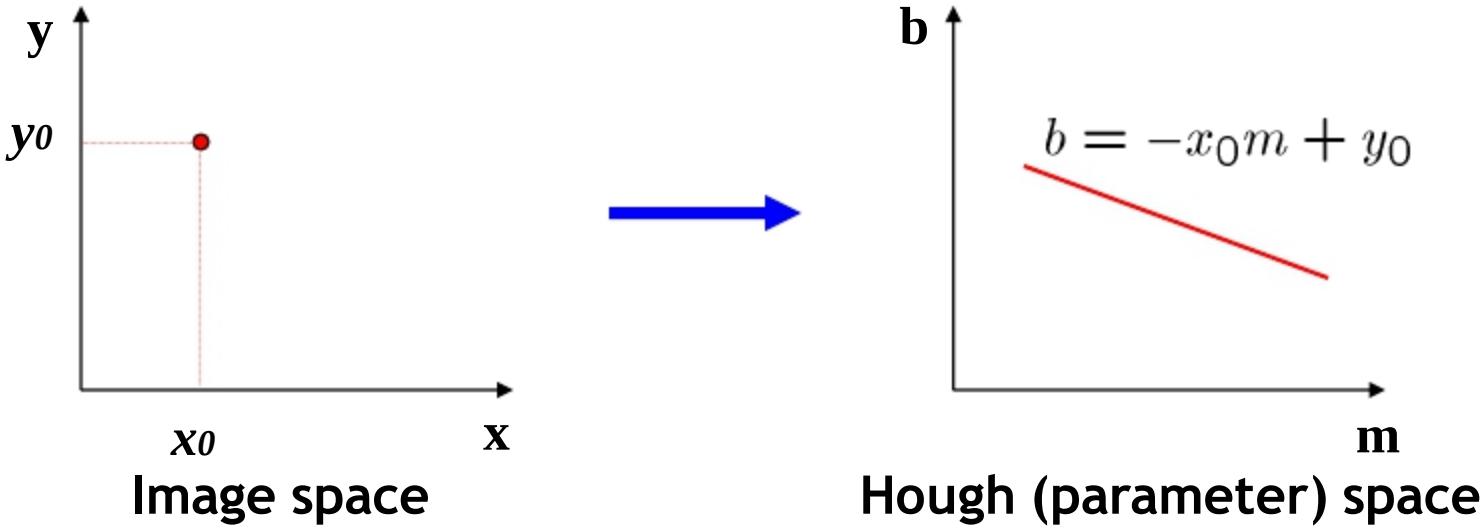
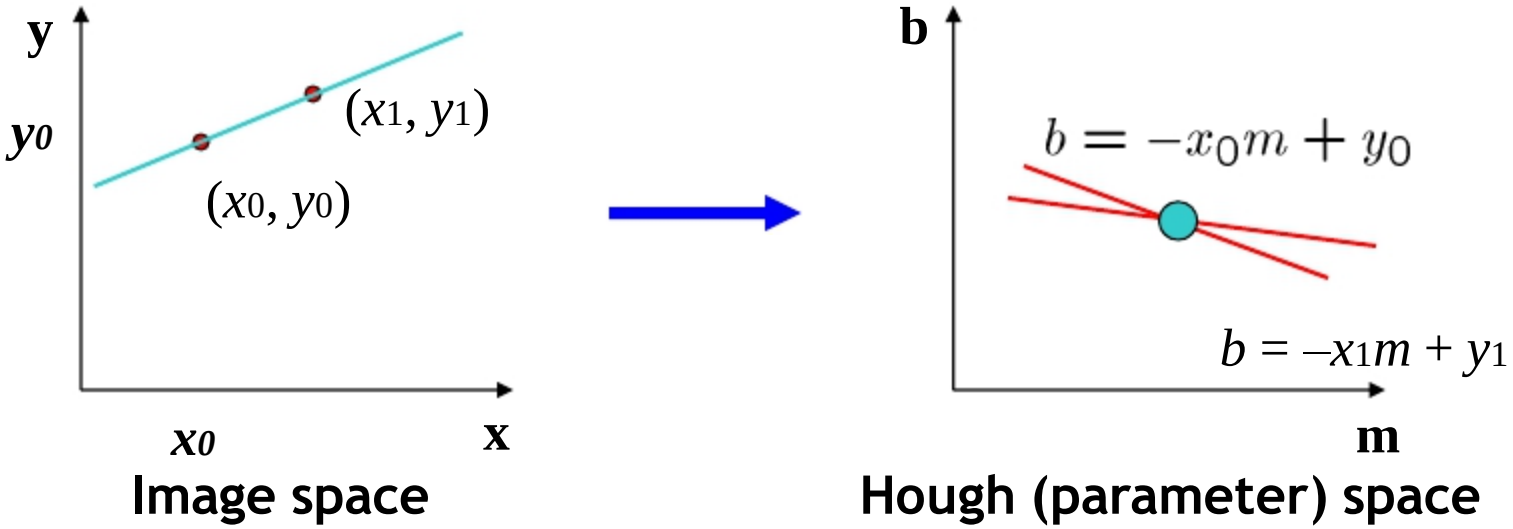


Image space

Hough (parameter) space

- Connection between image (x,y) and Hough (m,b) spaces
  - A line in the image corresponds to a point in Hough space.
  - To go from image space to Hough space:
    - Given a set of points (x,y), find all (m,b) such that $y = mx + b$
  - What does a point (x_0, y_0) in the image space map to?
    - Answer: the solutions of $b = -x_0 m + y_0$
    - This is a line in Hough space.

Finding Lines in an Image: Hough Space



- **What are the line parameters for the line that contains both (x_0, y_0) and (x_1, y_1)?**
  - **It is the intersection of the lines**

    $b = -x_0 m + y_0$ **and**

    $b = -x_1 m + y_1$

Finding Lines in an Image: Hough Space

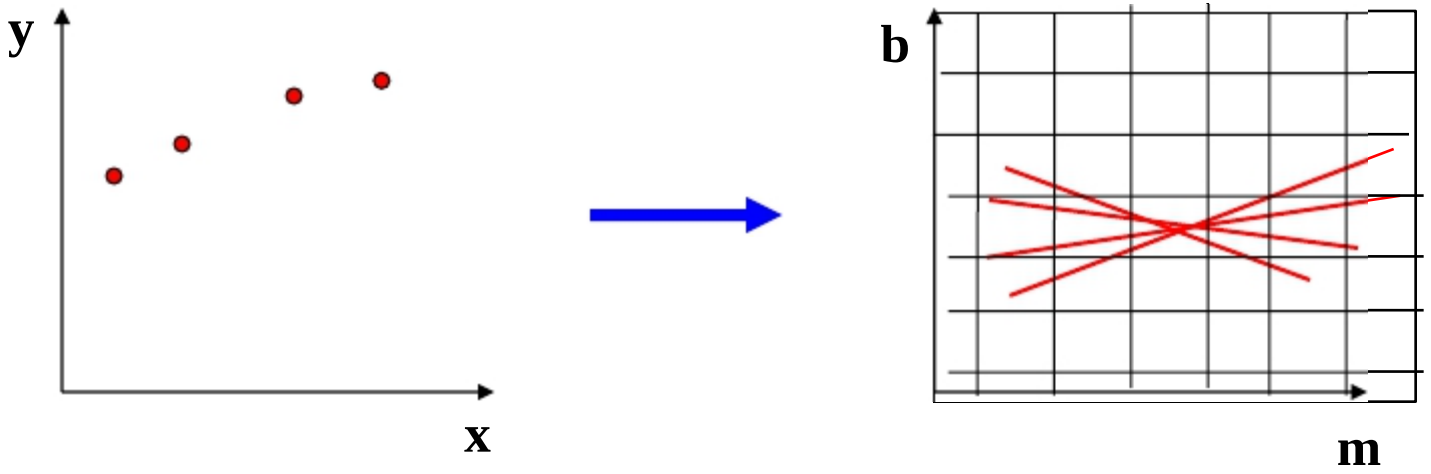


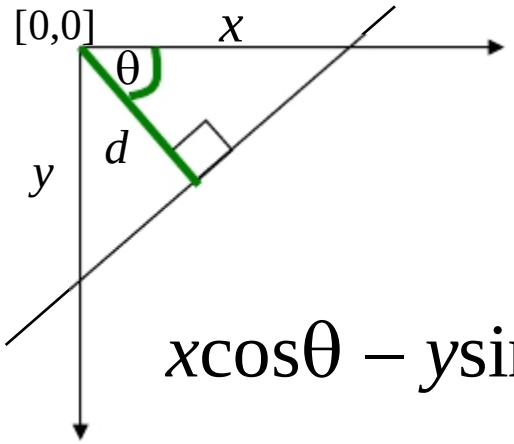
Image space

Hough (parameter) space

- How can we use this to find the most likely parameters (m,b) for the most prominent line in the image space?
  - Let each edge point in image space *vote* for a set of possible parameters in Hough space.
  - Accumulate votes in discrete set of bins; parameters with the most votes indicate line in image space.

Polar Representation for Lines

- Issues with usual (m,b) parameter space: can take on infinite values, undefined for vertical lines.

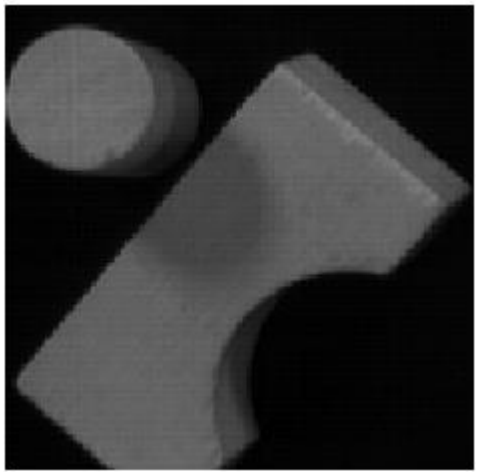


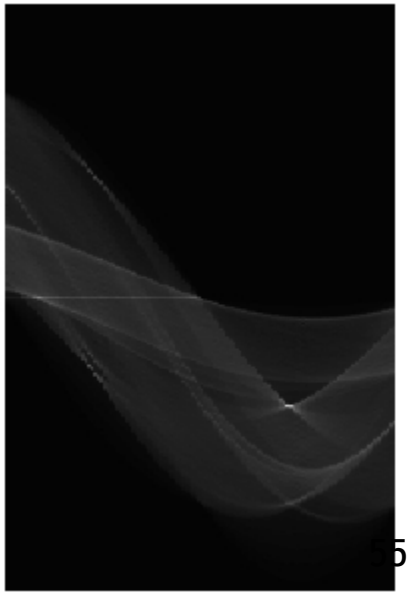
$$x\cos\theta - y\sin\theta = d$$

d : perpendicular distance from line to origin

θ : angle the perpendicular makes with the x-axis

- Point in image space
  ⇒ Sinusoid segment in Hough space





Slide adapted from Steve Seitz

Hough Transform Algorithm

Using the polar parameterization:

$$x\cos\theta - y\sin\theta = d$$

Basic Hough transform algorithm

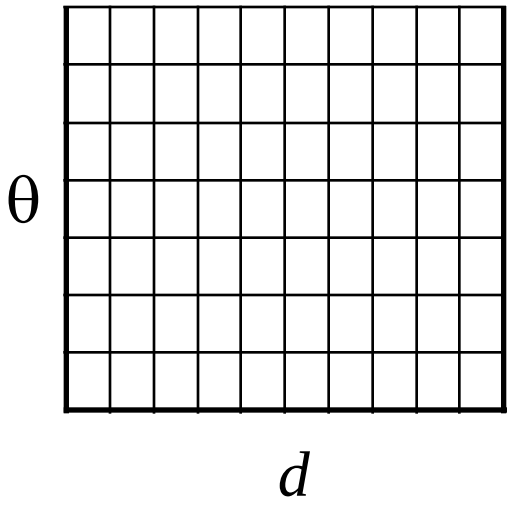
1. Initialize $H[d,\theta] = 0$.
2. For each edge point (x,y) in the image

   for $\theta = 0$ to 180 // some quantization

   $d = x\cos\theta - y\sin\theta$

   H[d, θ] += 1
3. Find the value(s) of (d,θ) where $H[d,\theta]$ is maximal.
4. The detected line in the image is given by $d = x\cos\theta - y\sin\theta$

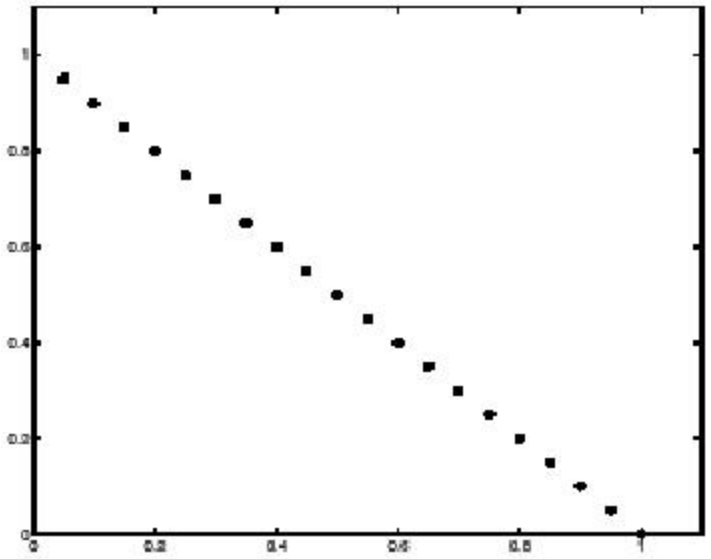
H: accumulator array (votes)



Hough line demo

- Time complexity (in terms of number of votes)?

Example: HT for Straight Lines



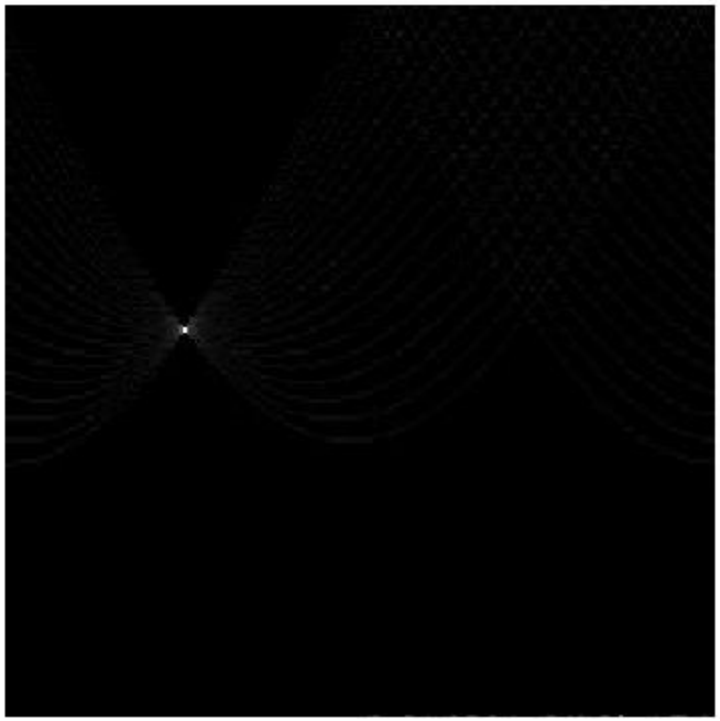


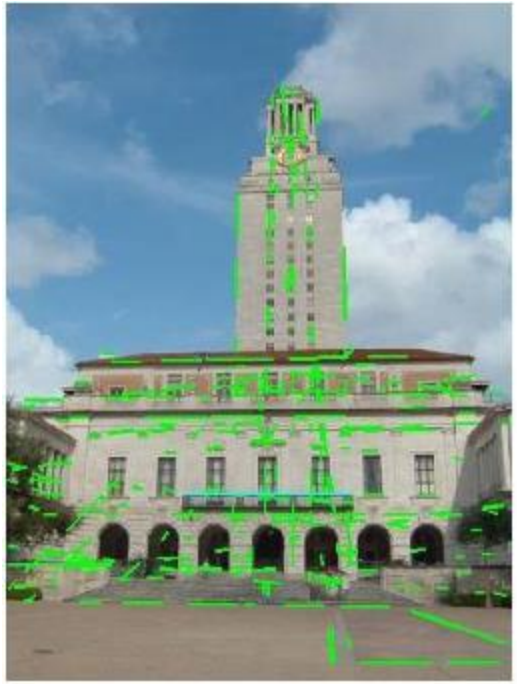
Image space
edge coordinates

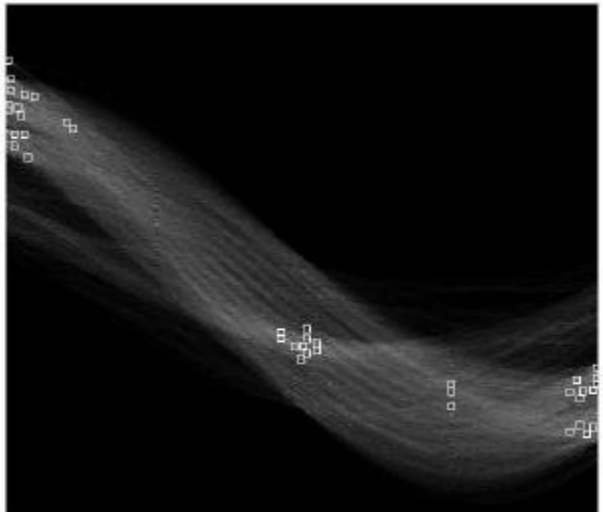
Votes

Bright value = high vote count
Black = no votes

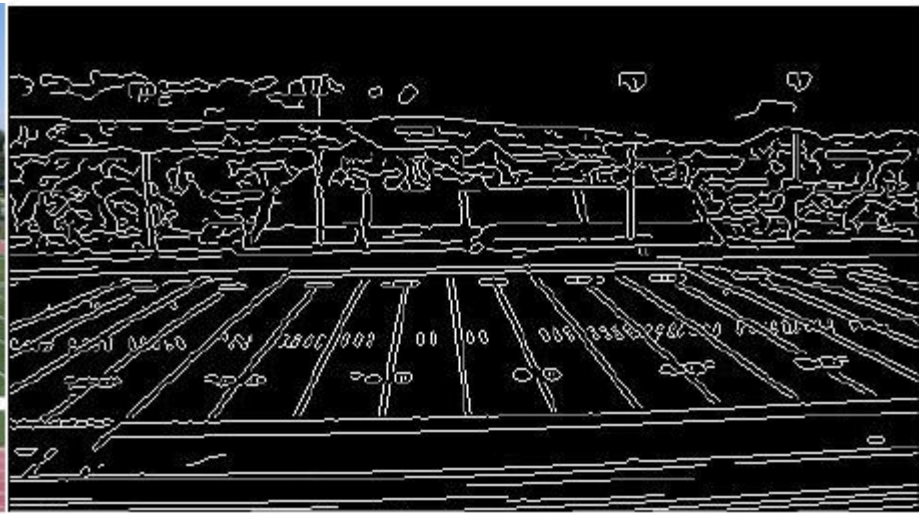
Slide credit: David Lowe

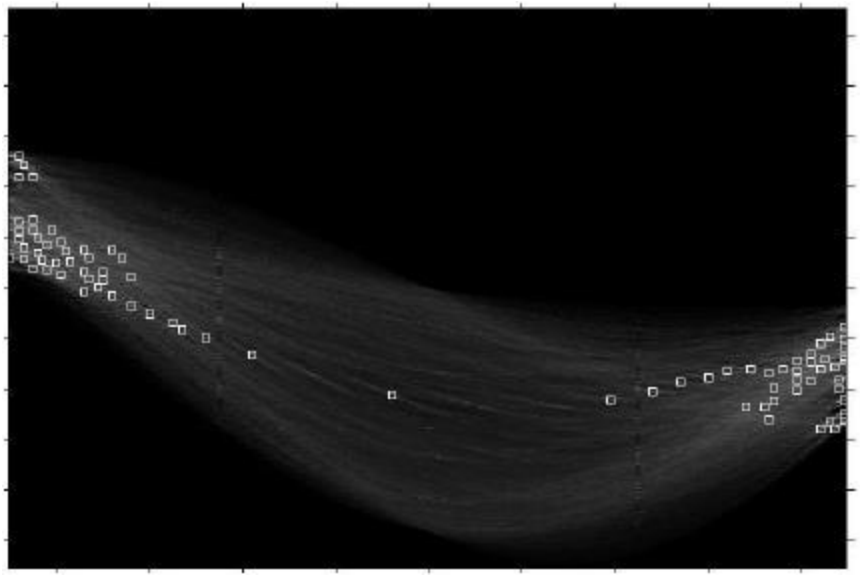
Real-World Examples

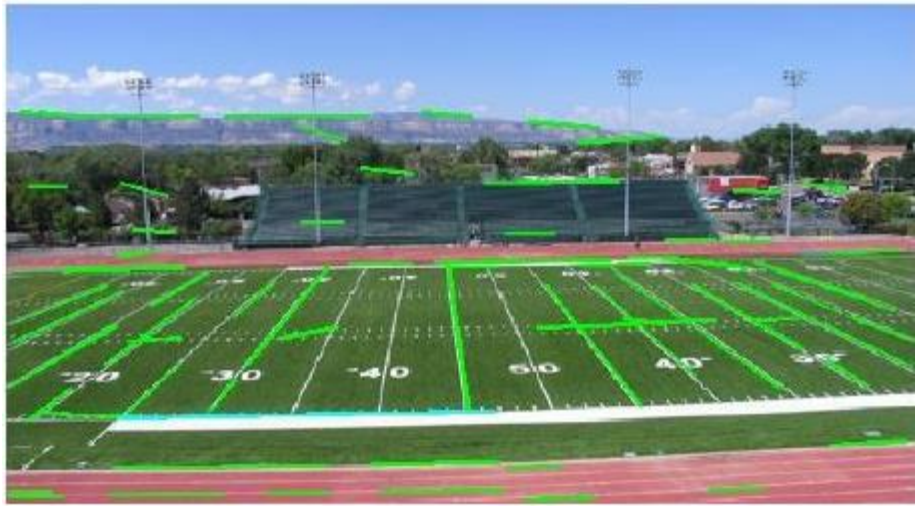




Slide credit: Kristen Grauman







Showing longest segments found

Slide credit: Kristen Grauman

Impact of Noise on Hough Transform

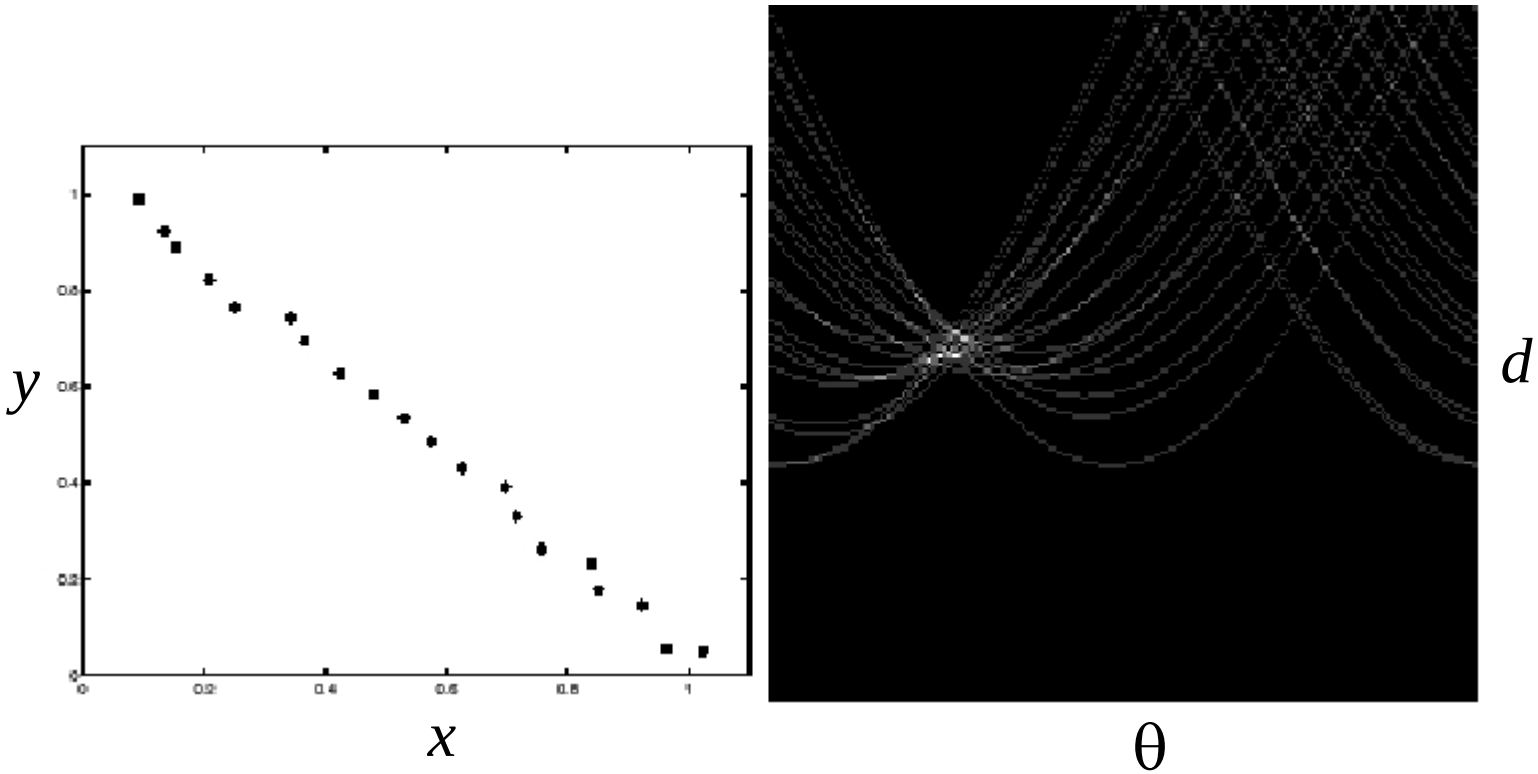
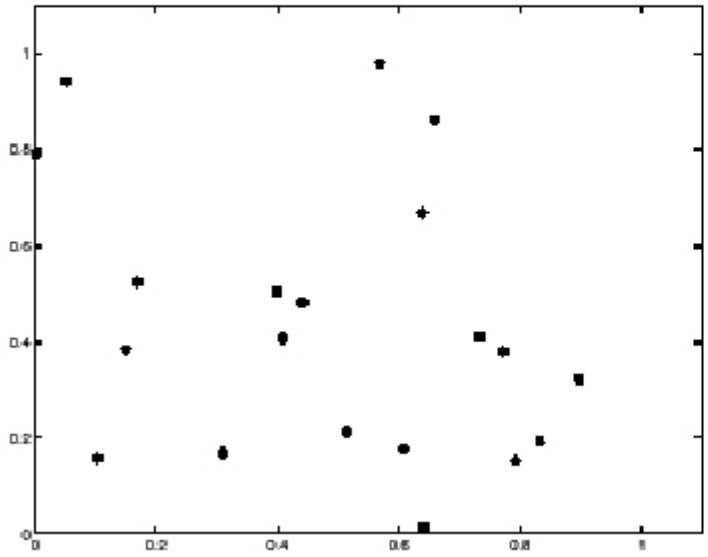


Image space
edge coordinates

Votes

What difficulty does this present for an implementation?

Impact of Noise on Hough Transform



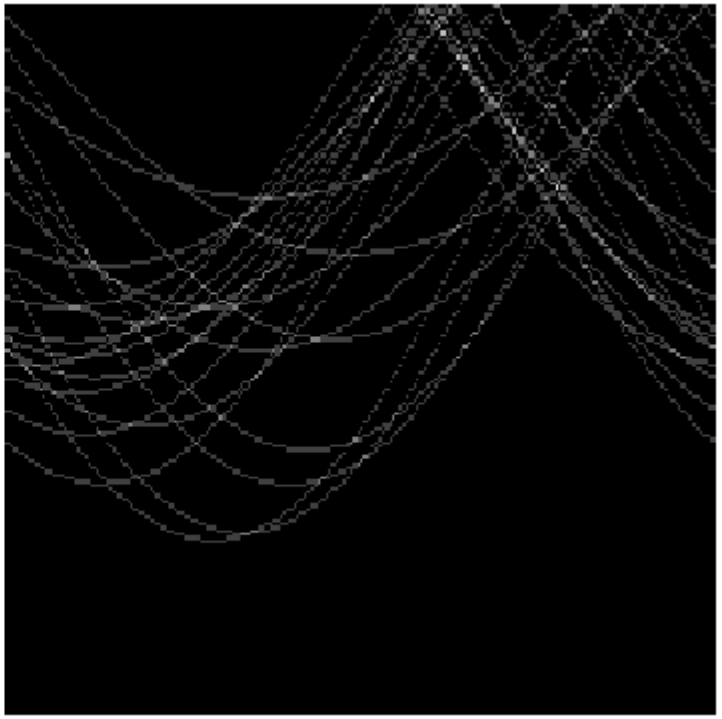


Image space
edge coordinates

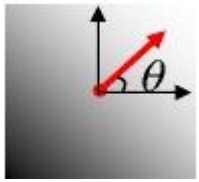
Votes

Here, everything appears to be “noise”, or random edge points, but we still see peaks in the vote space.

Slide credit: David Lowe

Extensions

Extension 1: Use the image gradient



$$\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]$$

$$\theta = \tan^{-1}\left(\frac{\partial f}{\partial y} / \frac{\partial f}{\partial x}\right)$$

1. same
2. for each edge point $I[x,y]$ in the image

   θ = gradient at (x,y)

   $d = x\cos\theta - y\sin\theta$

   $H[d,\theta]$ += 1
3. same
4. same

(Reduces degrees of freedom)

Slide credit: Kristen Grauman

Extensions

Extension 1: Use the image gradient

1. same
2. for each edge point $\mathrm{I}[x,y]$ in the image

   compute unique (d,θ) based on image gradient at (x,y)

   $\mathrm{H}[d,\theta]$ += 1
3. same
4. same

(Reduces degrees of freedom)

Extension 2

- Give more votes for stronger edges (use magnitude of gradient)

Extension 3

- Change the sampling of (d,θ) to give more/less resolution

Extension 4

- The same procedure can be used with circles, squares, or any other shape…

Slide credit: Kristen Grauman

推广到圆的检测?

Hough Transform for Circles

- **Circle: center (a,b) and radius r**

$$(x_i - a)^2 + (y_i - b)^2 = r^2$$

- **For a fixed radius r, unknown gradient direction**

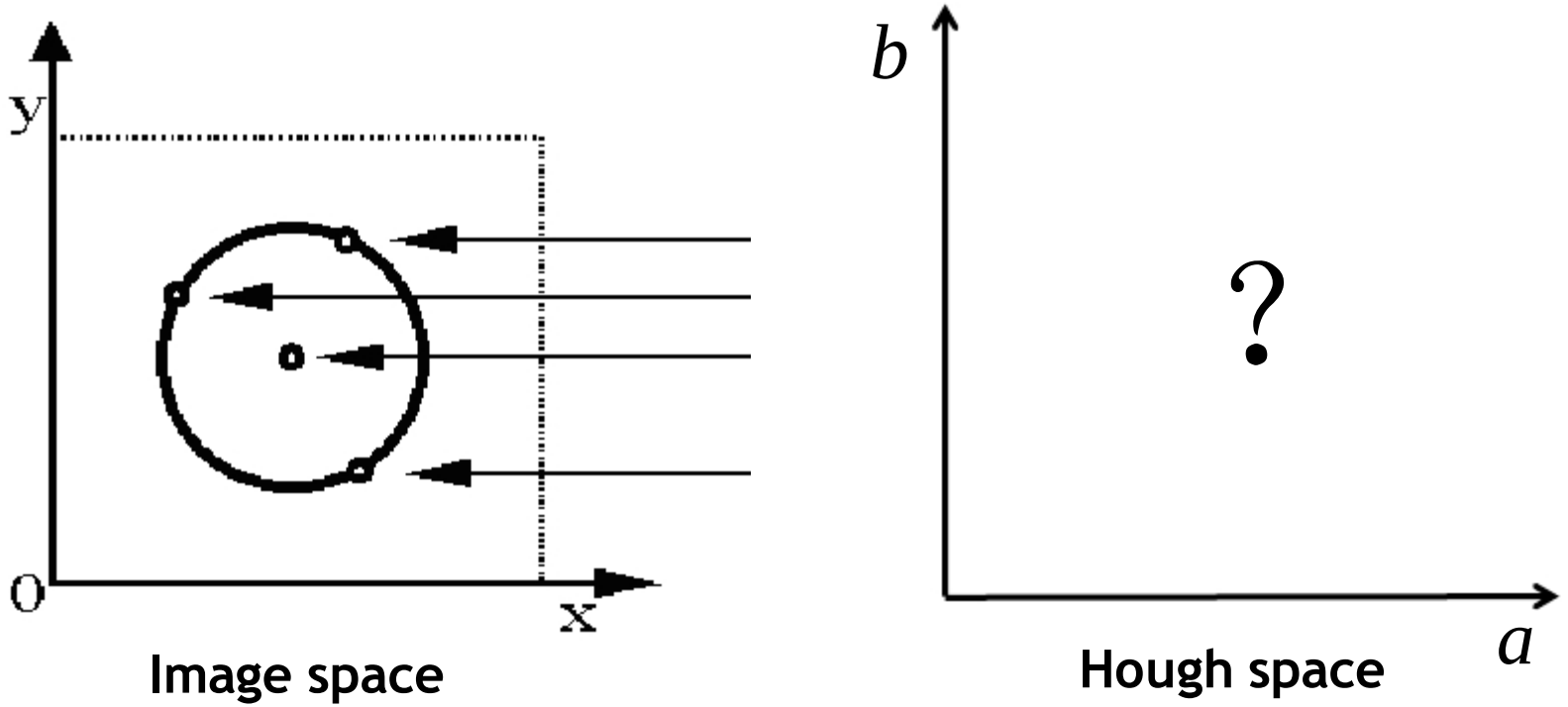


Image space

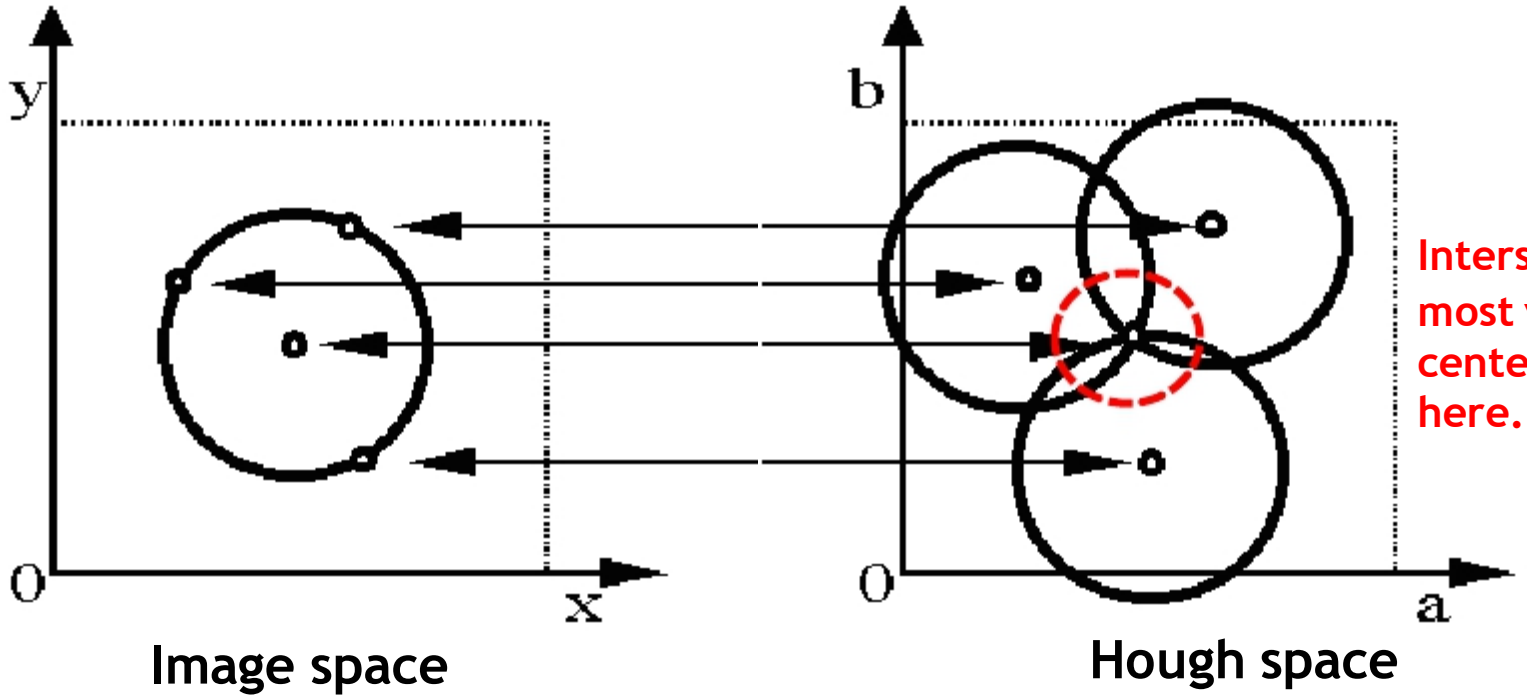
Hough space

Hough Transform for Circles

- **Circle: center (a,b) and radius r**

$$(x_i - a)^2 + (y_i - b)^2 = r^2$$

- **For a fixed radius r, unknown gradient direction**



Hough Transform for Circles

- **Circle: center (a,b) and radius r**

$$(x_i - a)^2 + (y_i - b)^2 = r^2$$

- **For an unknown radius r, unknown gradient direction**

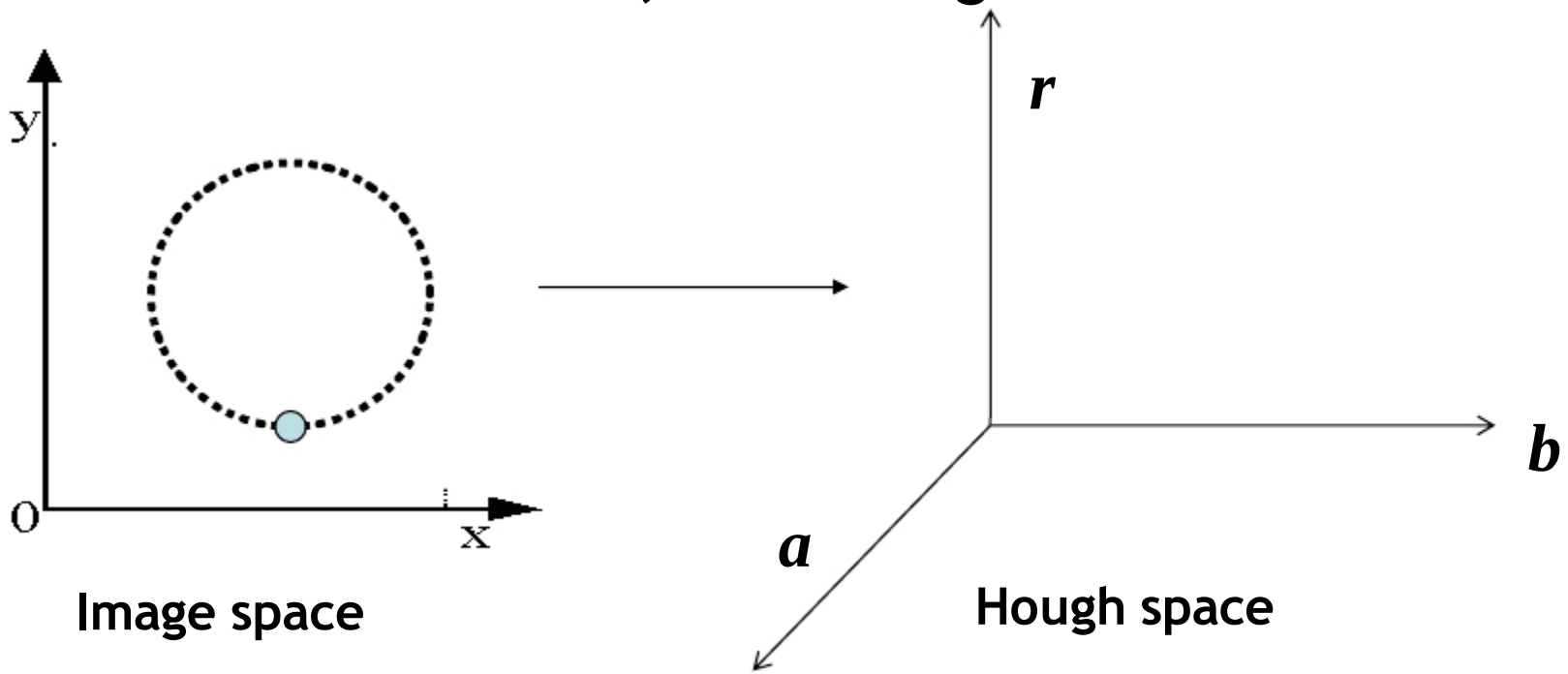


Image space

Hough space

Hough Transform for Circles

- **Circle: center (a,b) and radius r**

$$(x_i - a)^2 + (y_i - b)^2 = r^2$$

- **For an unknown radius r, unknown gradient direction**

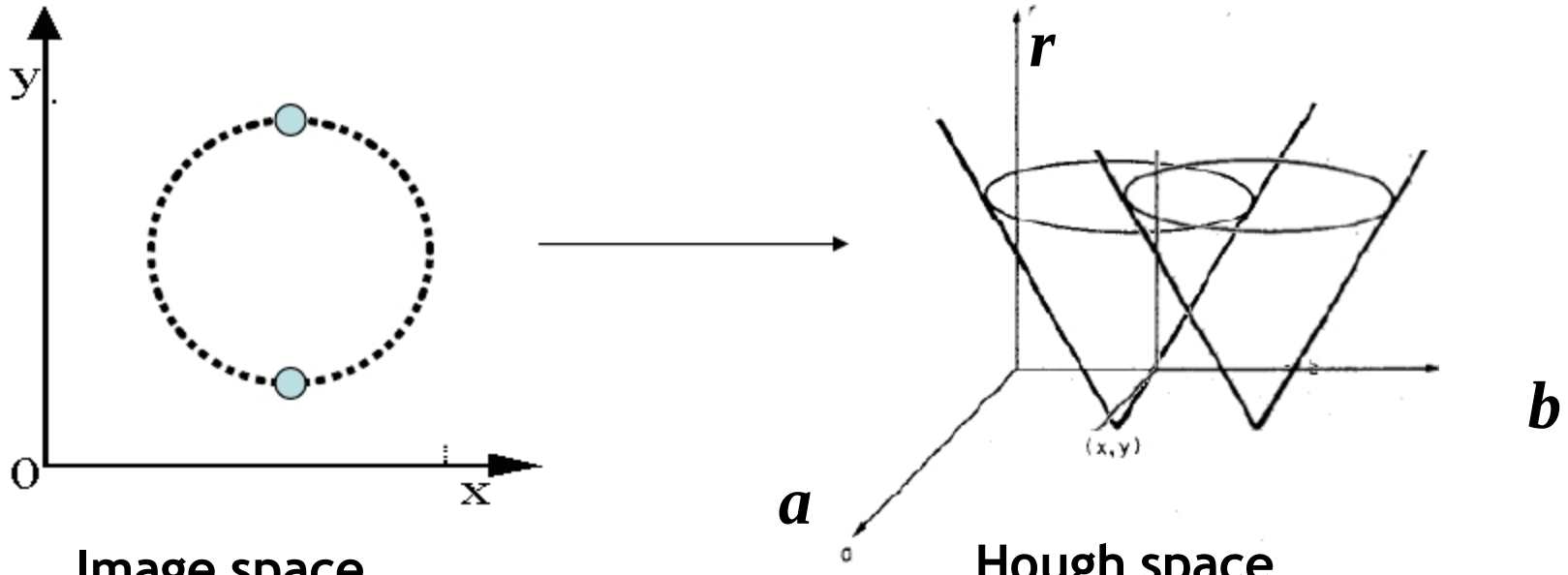


Image space

Hough space

Hough Transform for Circles

- **Circle: center (a,b) and radius r**

$$(x_i - a)^2 + (y_i - b)^2 = r^2$$

- **For an unknown radius r, *known* gradient direction**

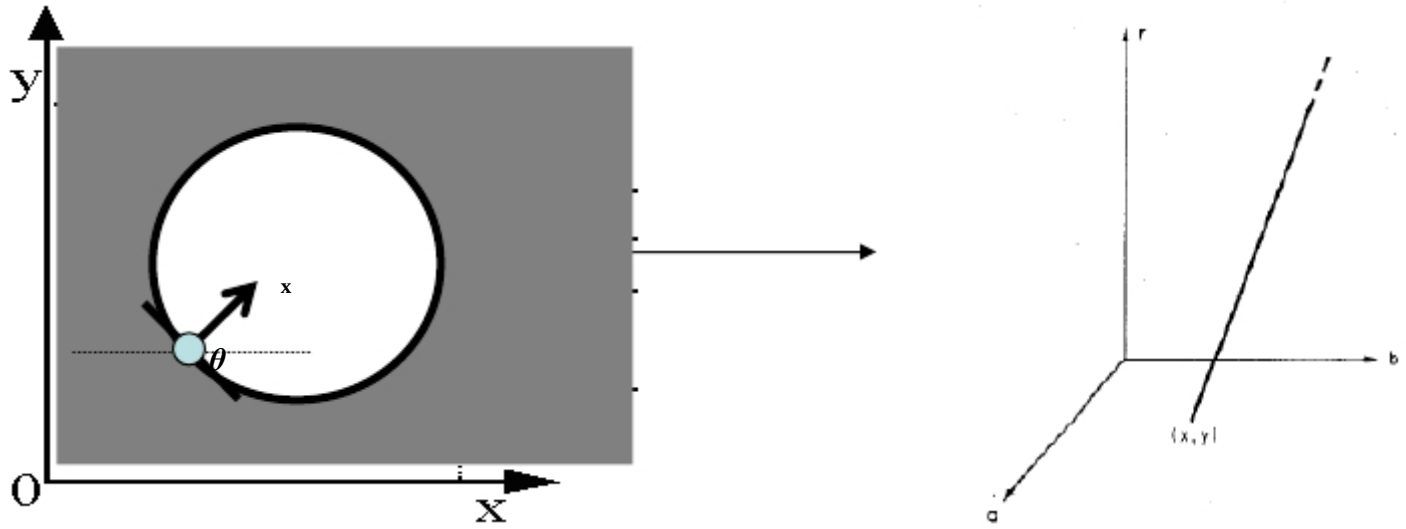


Image space

Hough space

Slide credit: Kristen Grauman

Hough Transform for Circles

For every edge pixel (x,y) **:**

  For each possible radius value r**:**

    For each possible gradient direction θ**:**

      // or use estimated gradient

      $a = x - r\cos(\theta)$

      $b = y + r\sin(\theta)$

      $H[a,b,r] \mathrel{+}= 1$

  end

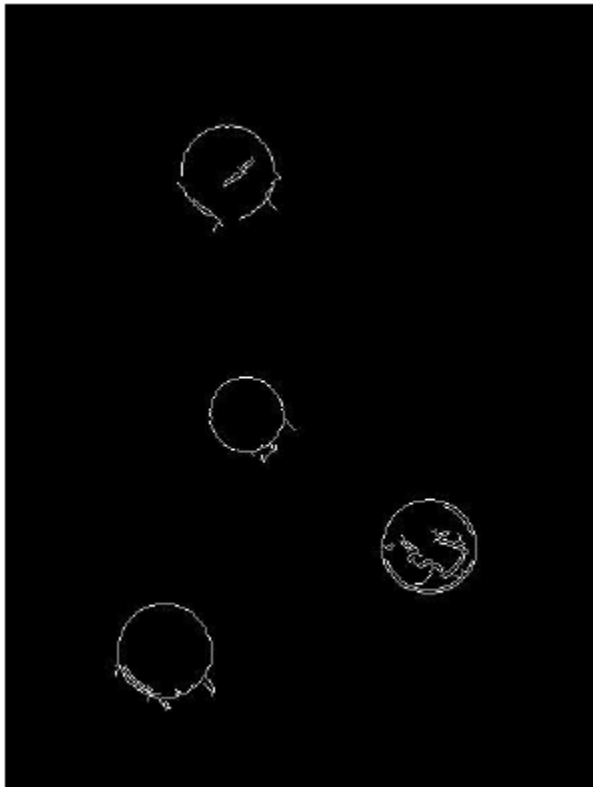
end

Slide credit: Kristen Grauman

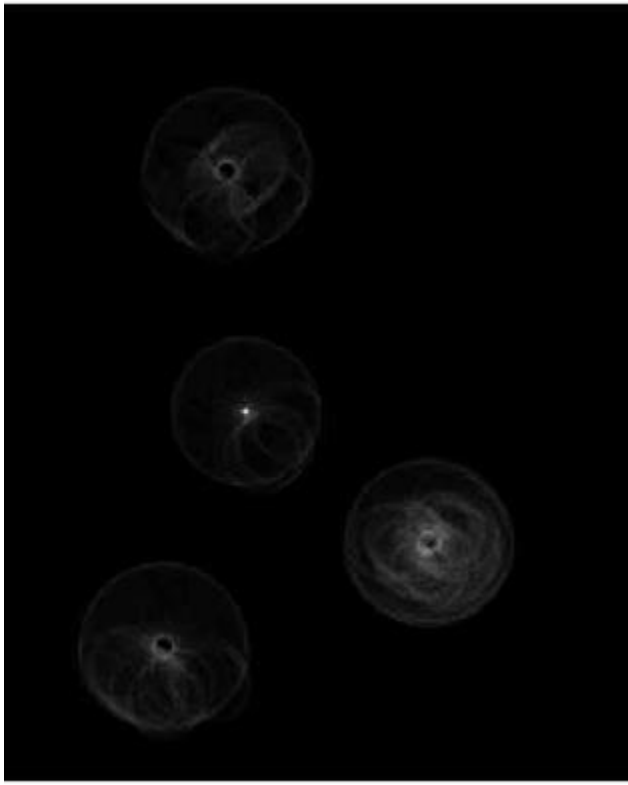
Example: Detecting Circles with Hough

Original

Edges



Votes: Penny



Note: a different Hough transform (with separate accumulators) was used for each circle radius (quarters vs. penny).

Slide credit: Kristen Grauman

Coin finding sample images from: Vivek Kwatra

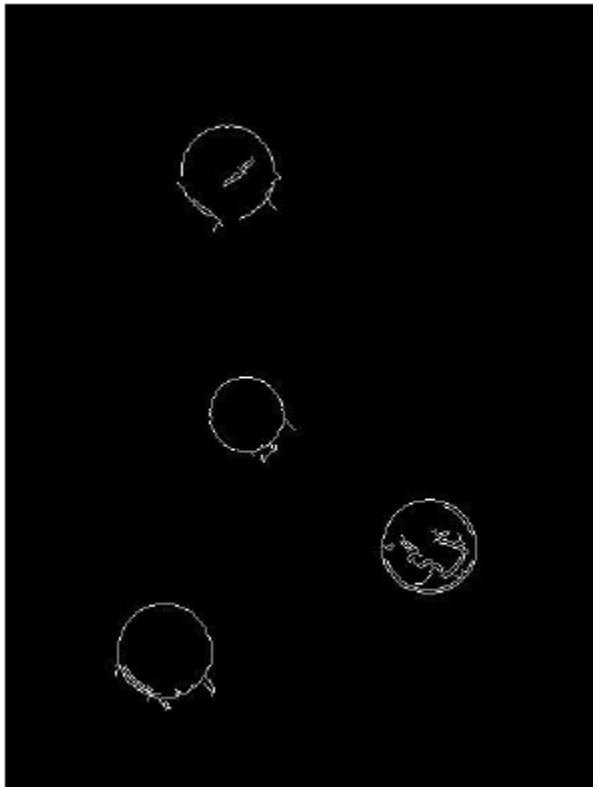
Example: Detecting Circles with Hough

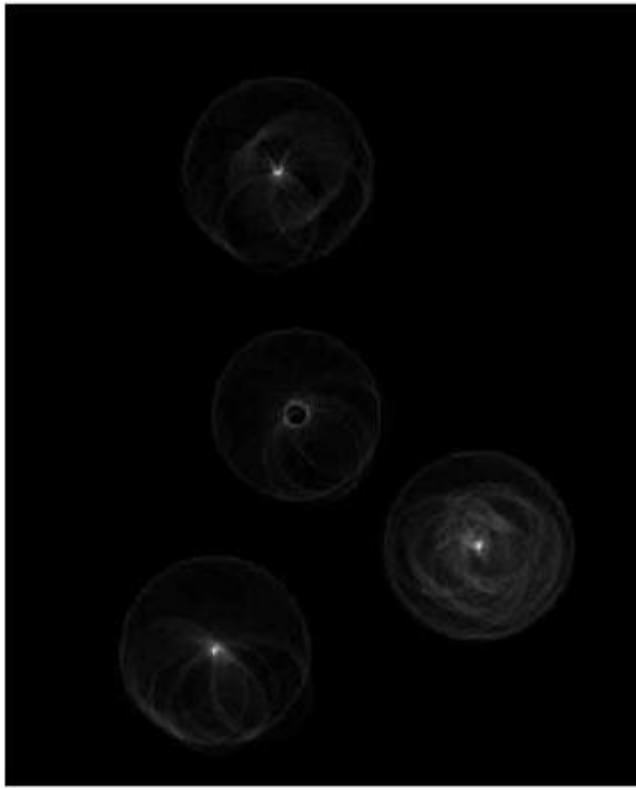
Original

Combined detections

Edges

Votes: Quarter





Voting: Practical Tips

- Minimize irrelevant tokens first (take edge points with significant gradient magnitude)
- Choose a good grid / discretization
  - Too coarse: large votes obtained when too many different lines correspond to a single bucket
  - Too fine: miss lines because some points that are not exactly collinear cast votes for different buckets
- Vote for neighbors, also (smoothing in accumulator array)
- Utilize direction of edge to reduce free parameters by 1
- To read back which points voted for “winning” peaks, keep tags on the votes.

Hough Transform: Pros and Cons

Pros

- All points are processed independently, so can cope with occlusion
- Some robustness to noise: noise points unlikely to contribute consistently to any single bin
- Can detect multiple instances of a model in a single pass

Cons

- Complexity of search time increases exponentially with the number of model parameters
- Non-target shapes can produce spurious peaks in parameter space
- Quantization: hard to pick a good grid size

推广到任意形状?

Generalized Hough Transform

- What if want to detect arbitrary shapes defined by boundary points and a reference point?

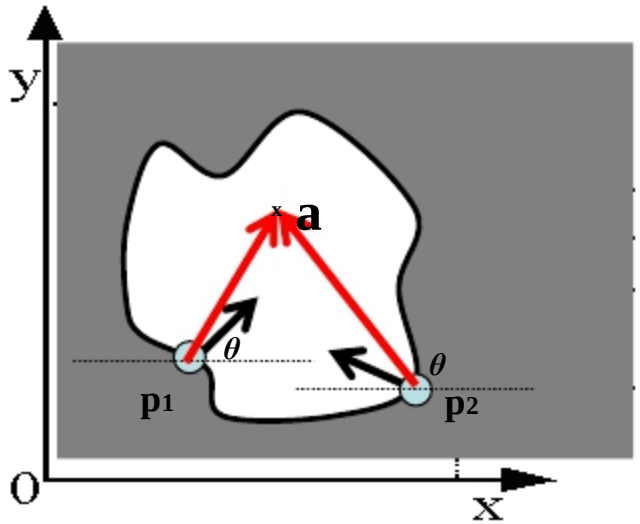


Image space

At each boundary point, compute displacement vector: $r = a - p_i$.

For a given model shape: store these vectors in a table indexed by gradient orientation θ.

[Dana H. Ballard, Generalizing the Hough Transform to Detect Arbitrary Shapes, 1980]

Generalized Hough Transform

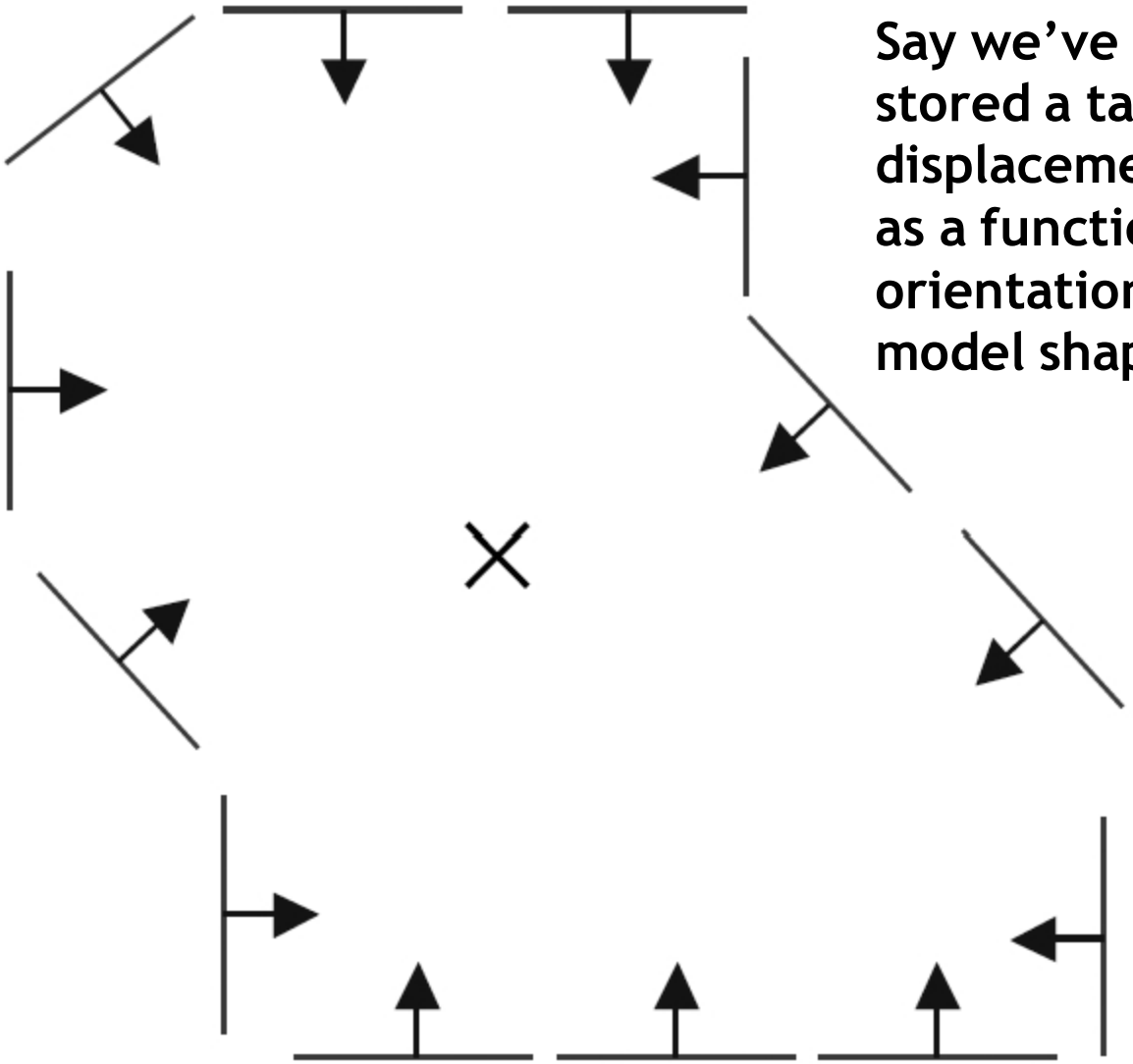
To *detect* the model shape in a new image:

- For each edge point
  - Index into table with its gradient orientation θ
  - Use retrieved r vectors to vote for position of reference point
- Peak in this Hough space is reference point with most supporting edges

Assuming translation is the only transformation here, i.e., orientation and scale are fixed.

Slide credit: Kristen Grauman

Example: Generalized Hough Transform

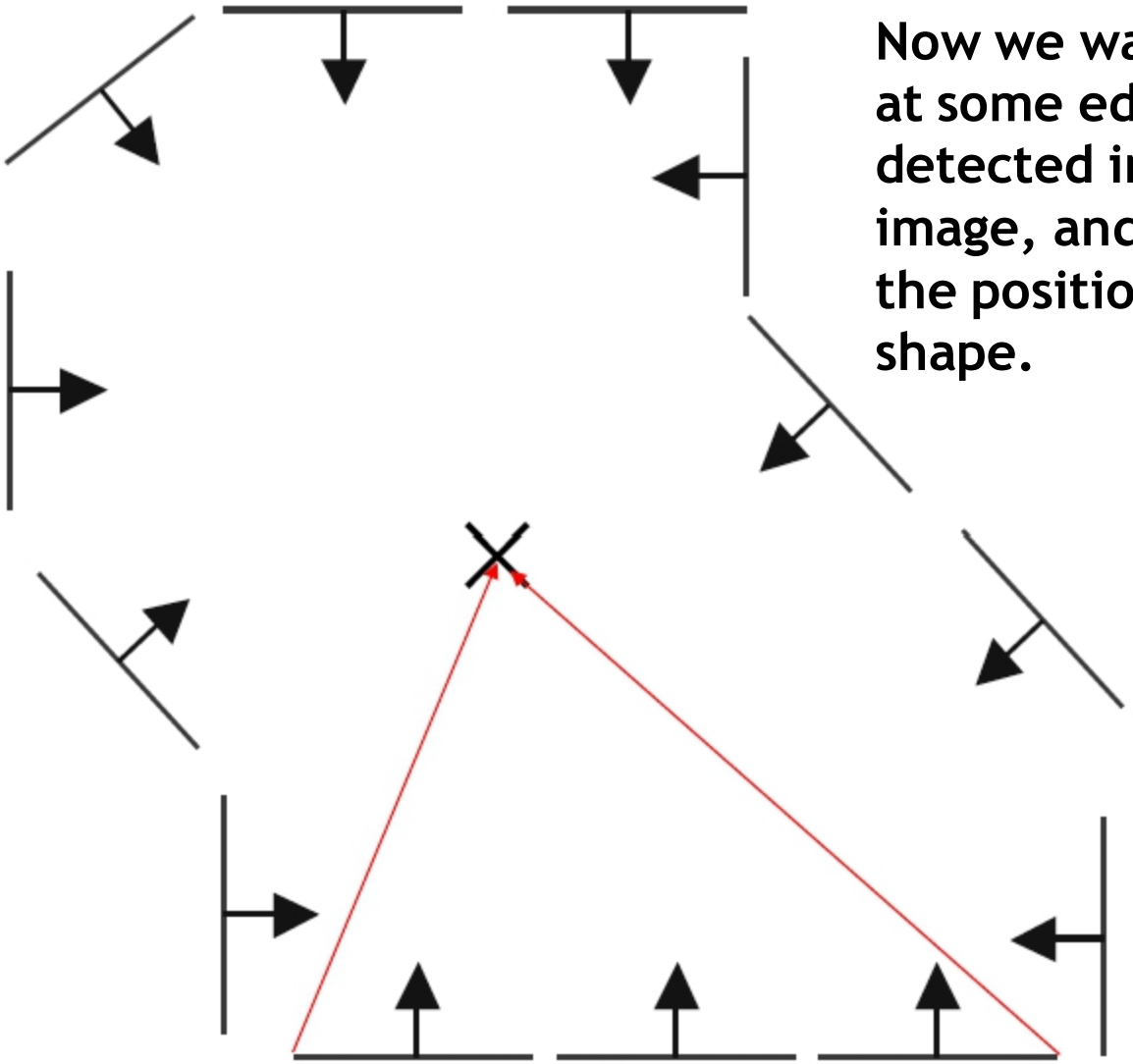


Model shape

Say we've already stored a table of displacement vectors as a function of edge orientation for this model shape.

Slide credit: Svetlana Lazebnik

Example: Generalized Hough Transform

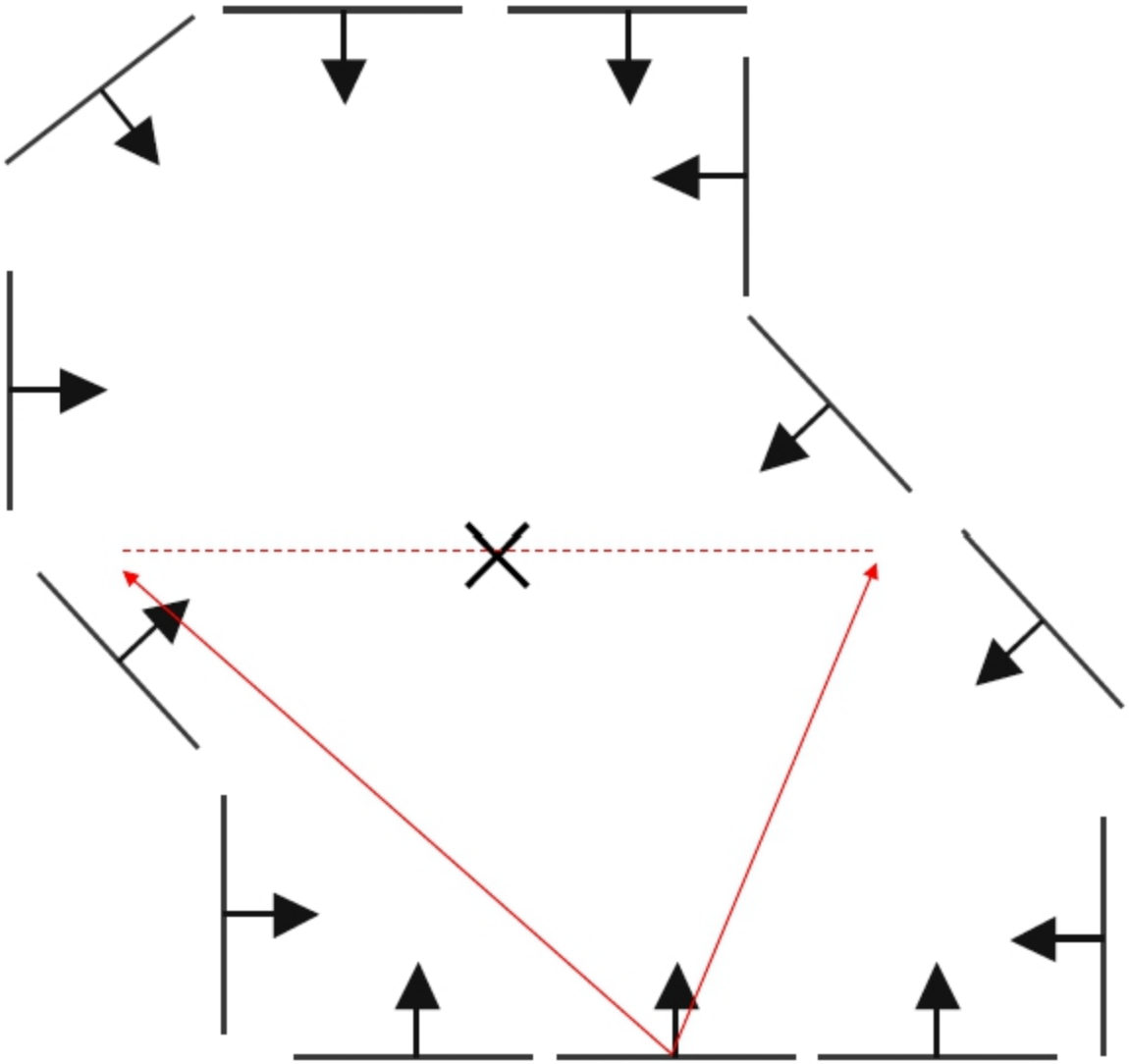


Now we want to look at some edge points detected in a *new* image, and vote on the position of that shape.

Displacement vectors for model points

Slide credit: Svetlana Lazebnik

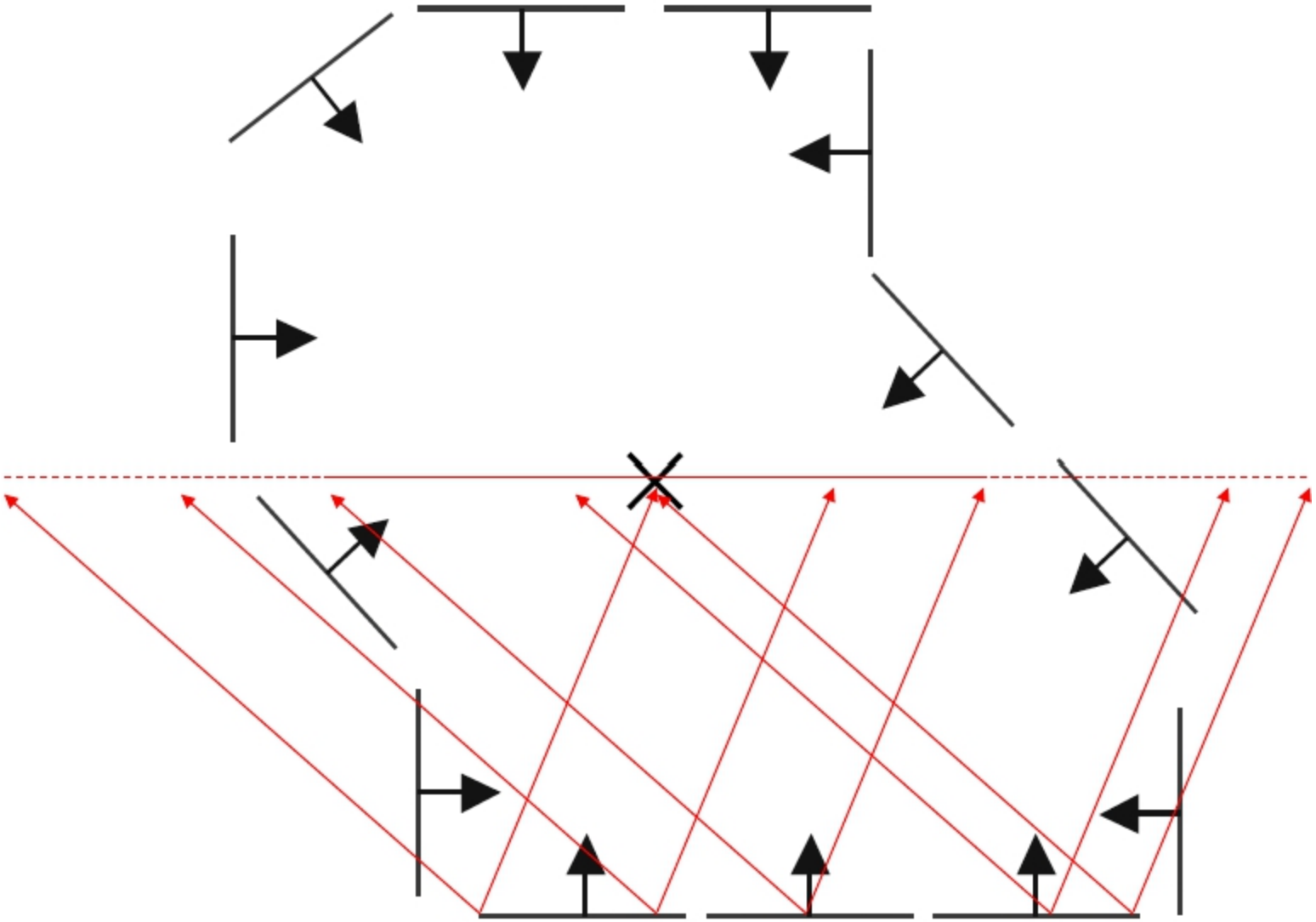
Example: Generalized Hough Transform



Range of voting locations for test point

Slide credit: Svetlana Lazebnik

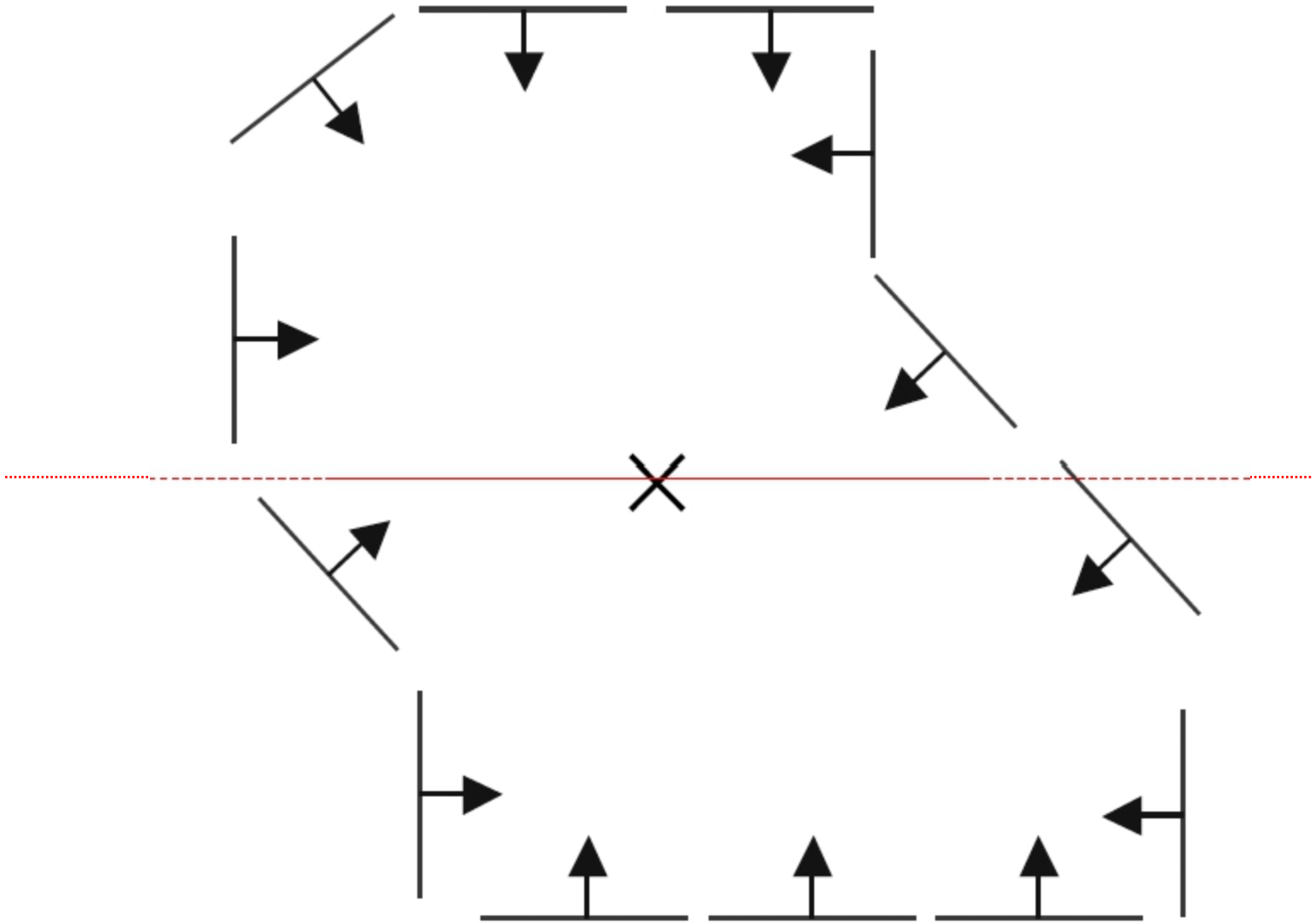
Example: Generalized Hough Transform



Range of voting locations for test point

Slide credit: Svetlana Lazebnik

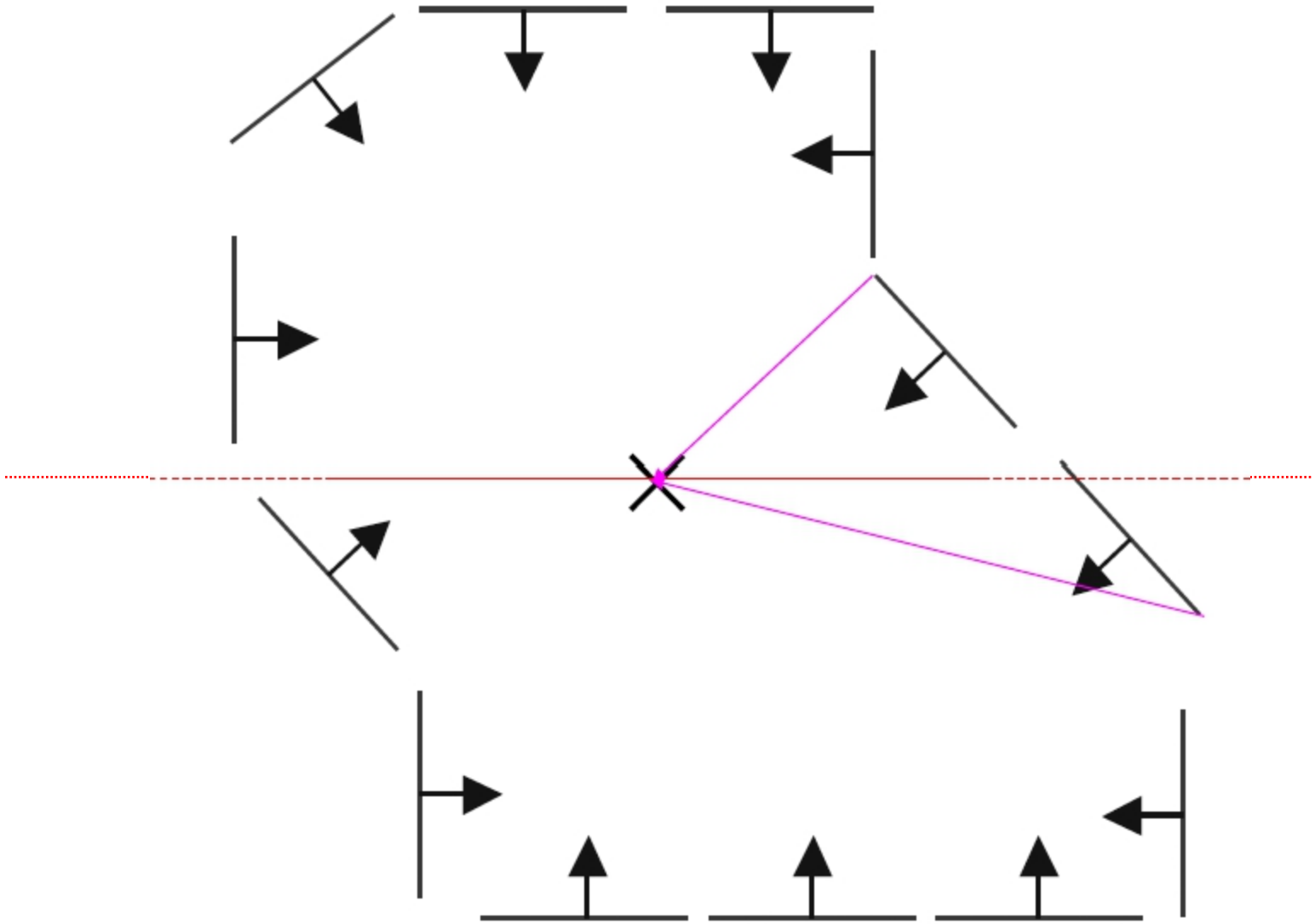
Example: Generalized Hough Transform



Votes for points with θ = ↑

Slide credit: Svetlana Lazebnik

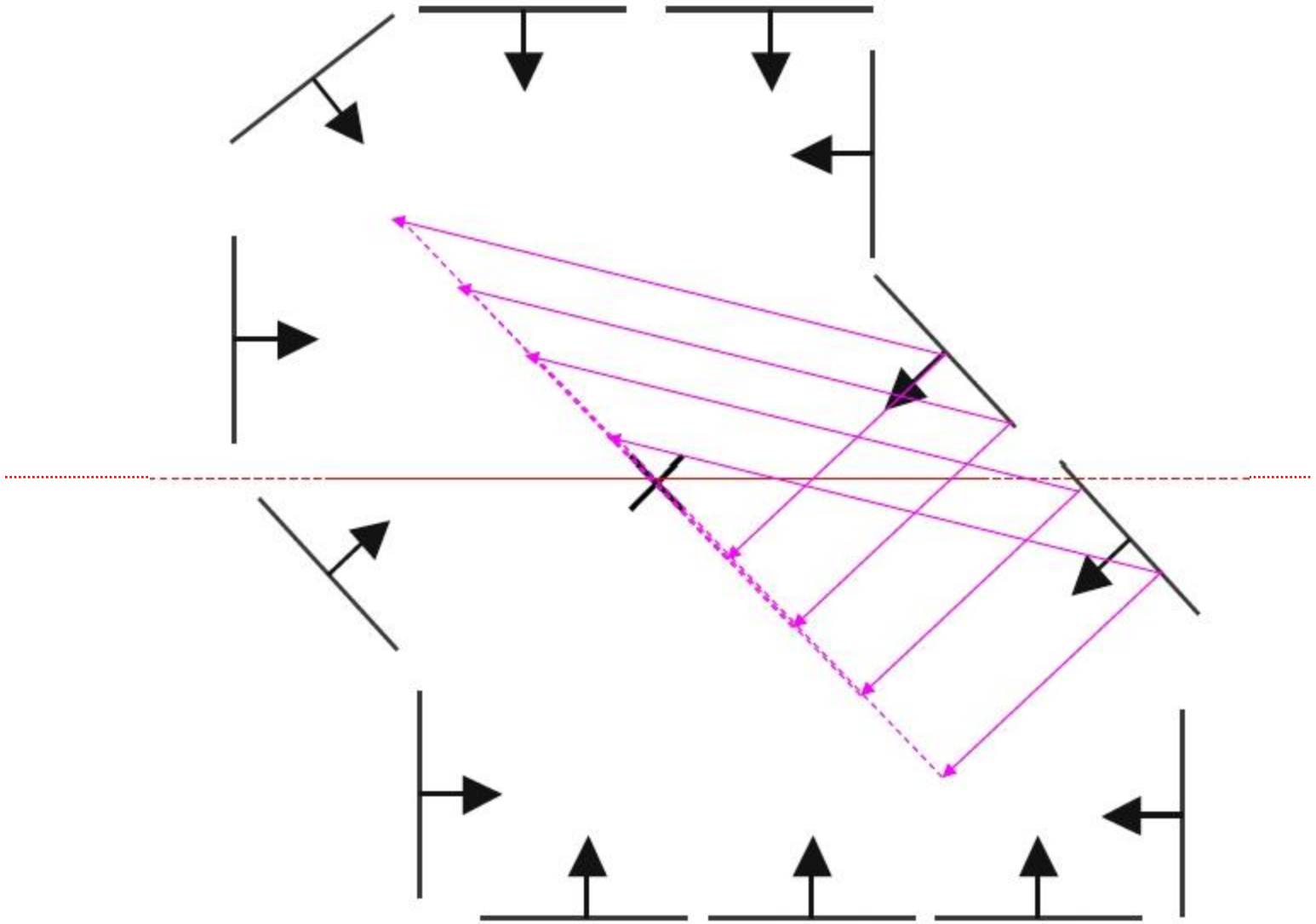
Example: Generalized Hough Transform



Displacement vectors for model points

Slide credit: Svetlana Lazebnik

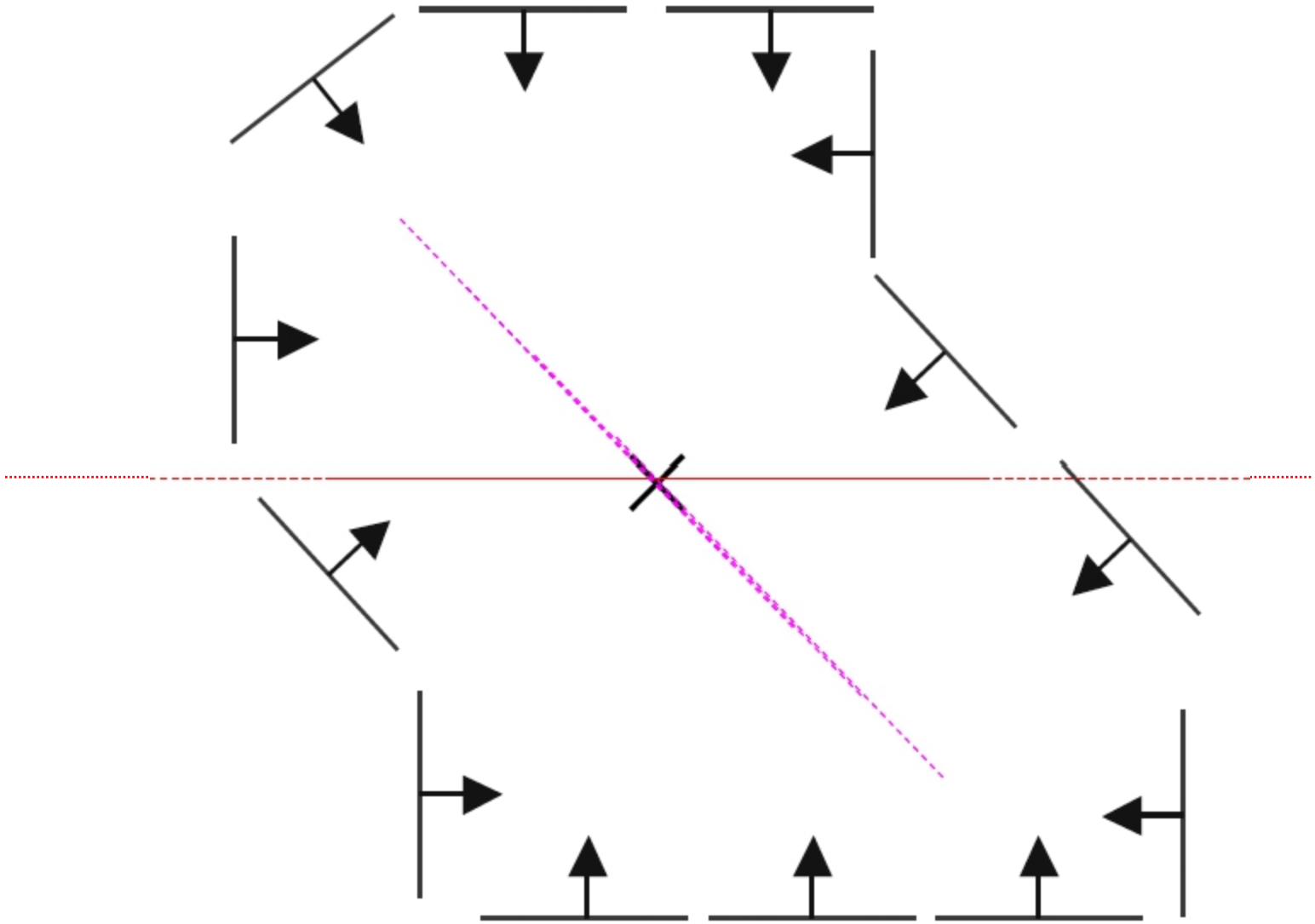
Example: Generalized Hough Transform



Range of voting locations for test point

Slide credit: Svetlana Lazebnik

Example: Generalized Hough Transform

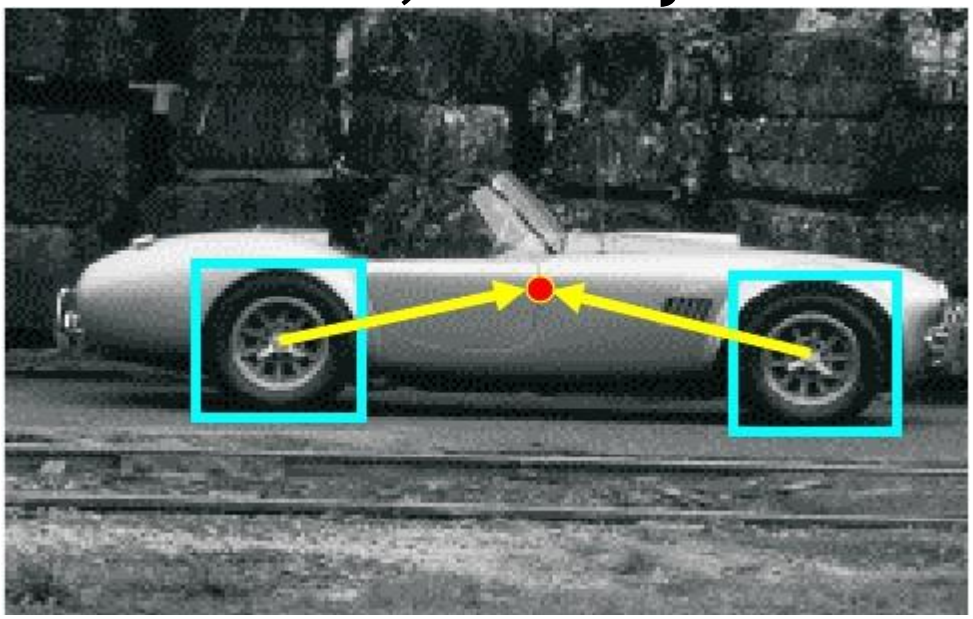


Votes for points with θ = ↙

Slide credit: Svetlana Lazebnik

Application in Recognition

- Instead of indexing displacements by gradient orientation, index by "visual codeword".



Training image

Visual codeword with displacement vectors

B. Leibe, A. Leonardis, and B. Schiele, Robust Object Detection with Interleaved Categorization and Segmentation, International Journal of Computer Vision, Vol. 77(1-3), 2008.

Application in Recognition

- Instead of indexing displacements by gradient orientation, index by "visual codeword".

Test image

- We'll hear more about this method in lecture 14…