THE UNIVERSITY OF BRITISH COLUMBIA

CPSC 425: Computer Vision

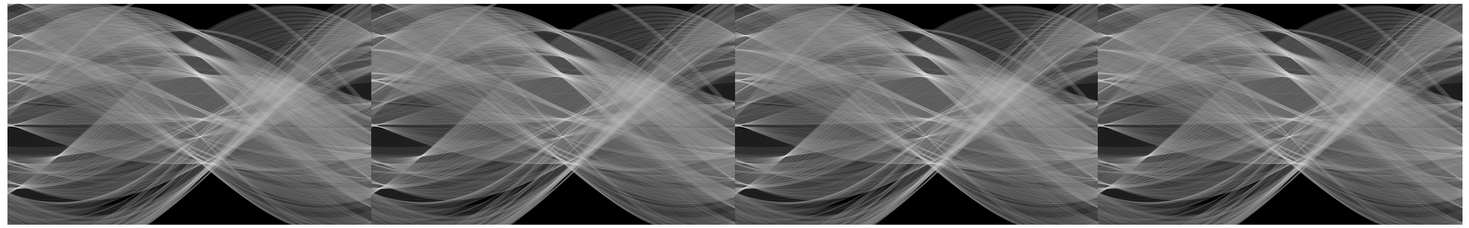


Image Credit: Ioannis (Yannis) Gkioulekas (CMU)

Lecture 22: Hough Transform cont.

Menu for Today (**October 29, 2018**)

Topics:

- Hough Transform
- iClicker Quiz
- Stereo Vision

Redings:

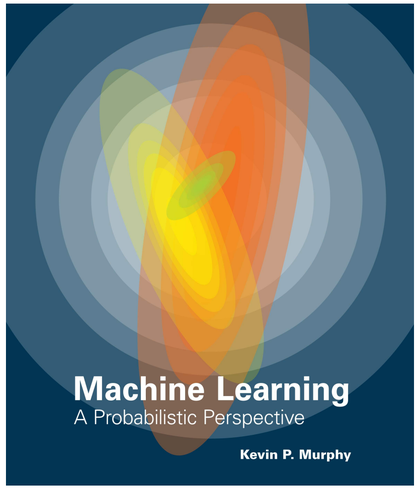
- **Today's** Lecture: Forsyth & Ponce (2nd ed.) 10.1, 10.2
- **Next** Lecture: Forsyth & Ponce (2nd ed.) 7.1.1, 7.2.1, 7.4, 7.6

Reminders:

- **Assignment 3**: Texture Syntheis is **out,** due on **October 29th**
- **Assignment 4**: Local Invariant Features and RANSAC **will be out** later today, due on **November 14th**

Today's "**fun**" Example: Im2Calories

ICCV 2015 paper by **Kevin Murphy**
(UBC's former faculty)



Coincidently Kevin is also author of one of the most prominent ML books

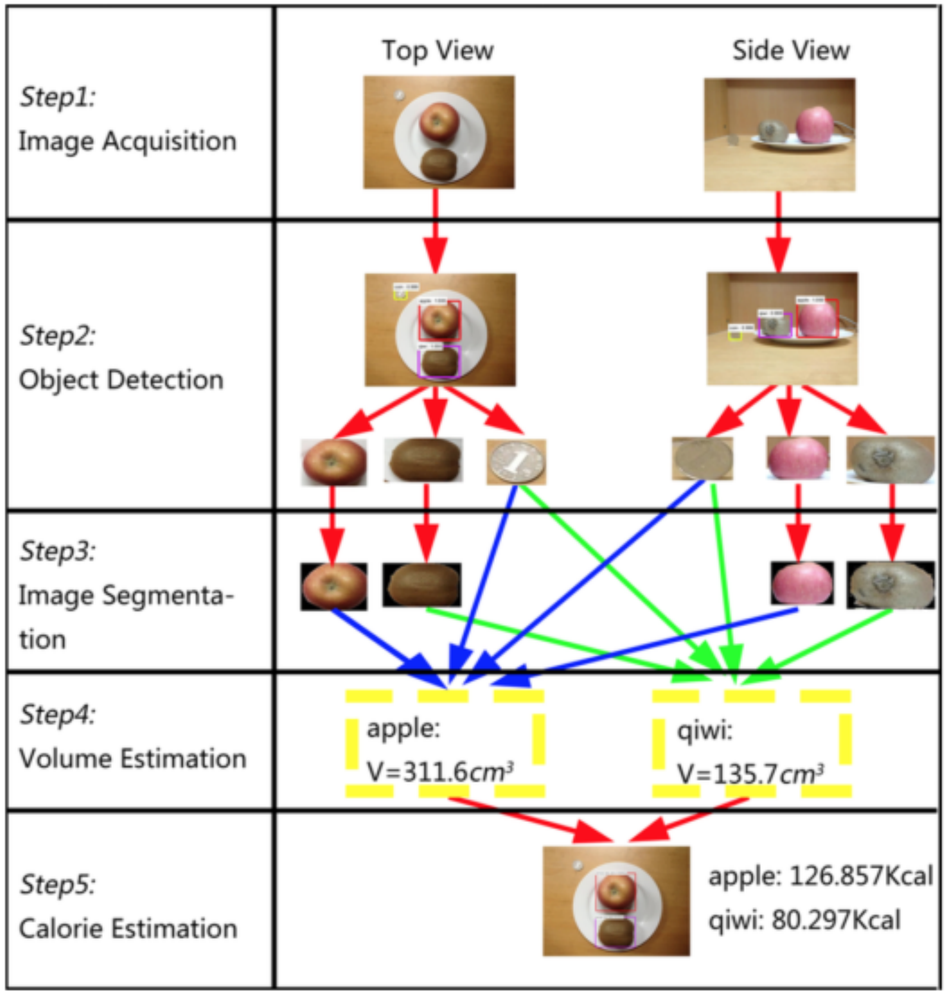


Figure 1: Calorie Estimation Flowchart

Today's "**fun**" Example: Im2Calories

Im2Calories: towards an automated mobile vision food diary

Austin Myers, Nick Johnston, Vivek Rathod, Anoop Korattikara, Alex Gorban Nathan Silberman, Sergio Guadarrama, George Papandreou, Jonathan Huang, Kevin Murphy amyers@umd.edu, (nickj, rathodv, kbanoop, gorban)@google.com (nsilberman, sguada, gpapan, jonathanhuang, kpmurphy)@google.com

Today's "**fun**" Example: Im2Calories

Fun **on-line demo**: http://www.caloriemama.ai/api

Lecture 21: Re-cap

Idea of **Hough transform**:

— For each token vote for all models to which the token could belong

— Return models that get many votes

Example: For each point, vote for all lines that could pass through it; the true lines will pass through many points and so receive many votes

Hough Transform for Lines (switching to books notation)

Idea: Each point votes for the lines that pass through it

— A line is the set of points, (x, y), such that

$$x \sin\theta - y \cos\theta + r = 0$$

— Different choices of θ, r give different lines

Hough Transform for Lines (switching to books notation)

Idea: Each point votes for the lines that pass through it

— A line is the set of points, (x, y), such that

$$x \sin\theta + y \cos\theta + r = 0$$

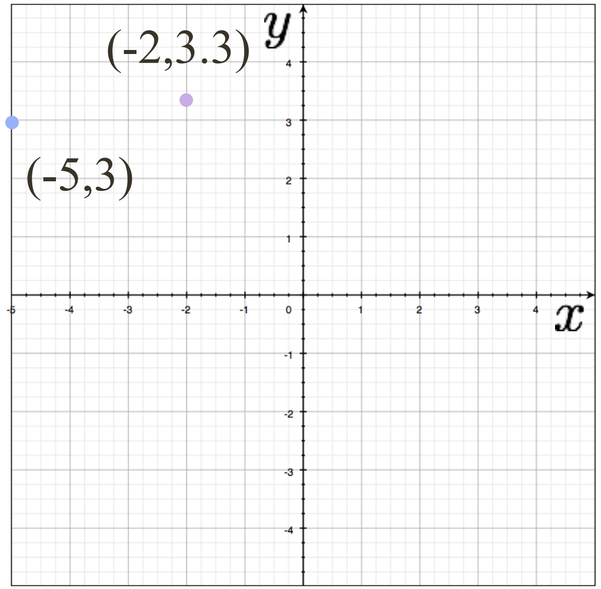
— Different choices of θ, r give different lines

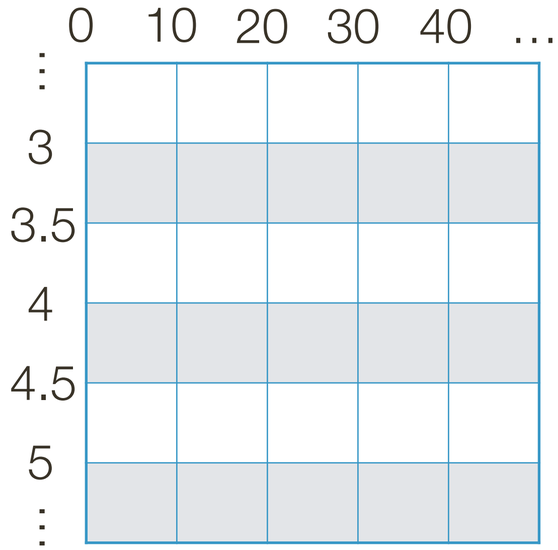
— For any (x, y) there is a one parameter family of lines through this point. Just let (x, y) be constants and for each value of θ the value of r will be determined

— Each point enters votes for each line in the family

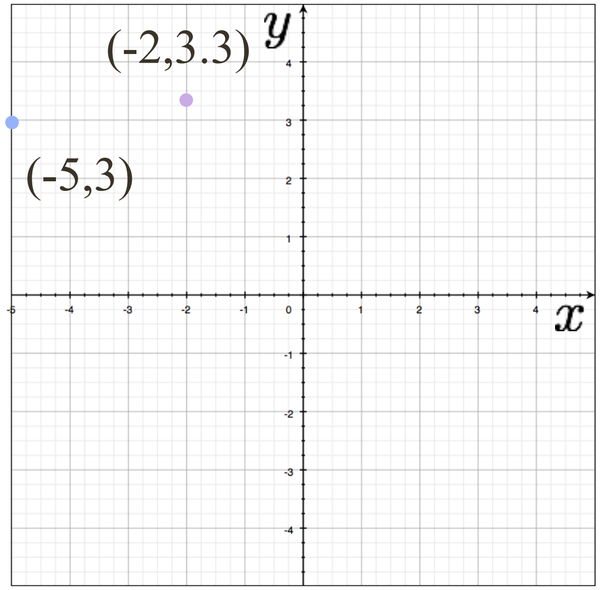
— If there is a line that has lots of votes, that will be the line passing near the points that voted for it

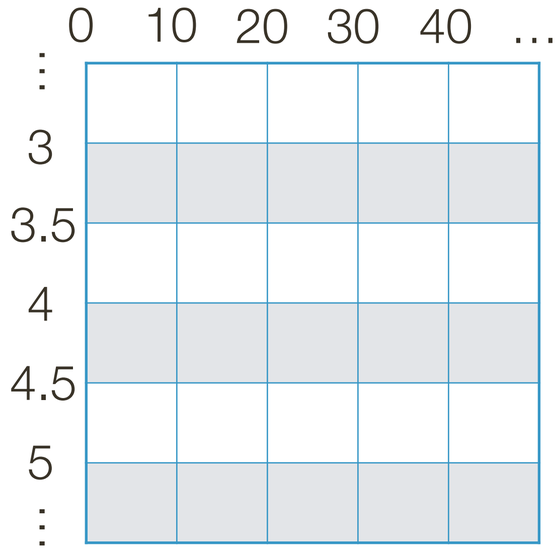
Example: Hough Transform for Lines





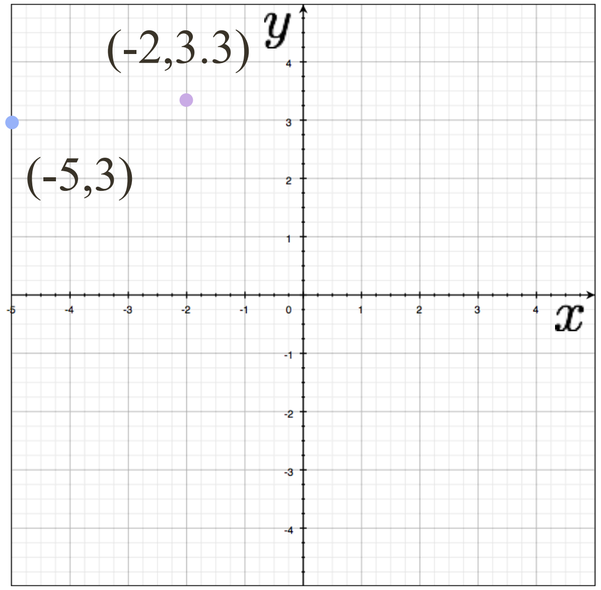
Example: Hough Transform for Lines

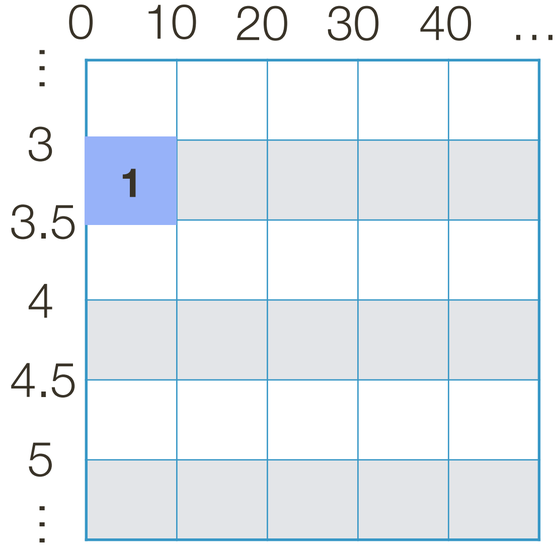




$$-5\sin(5°) - 3\cos(5°) + r = 0 => r = 3.42$$

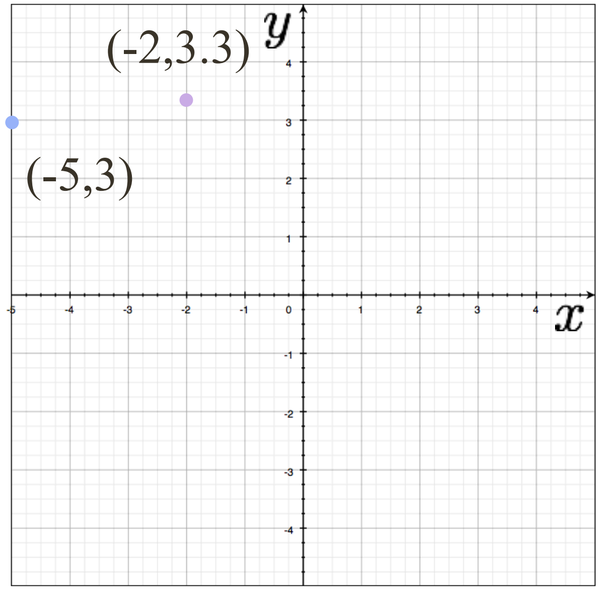
Example: Hough Transform for Lines

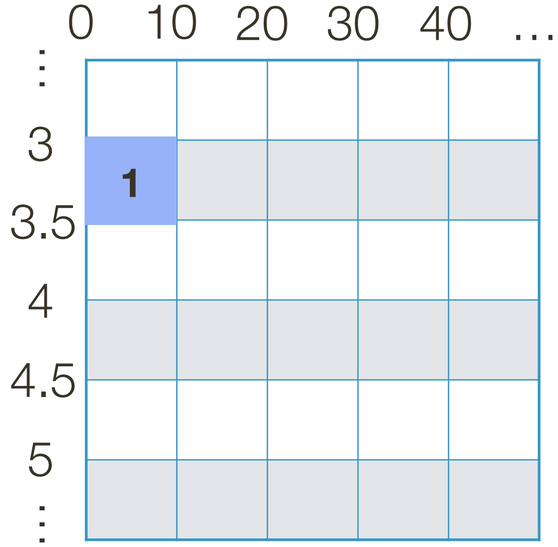




$$-5\sin(5°) - 3\cos(5°) + r = 0 => r = 3.42$$

Example: Hough Transform for Lines

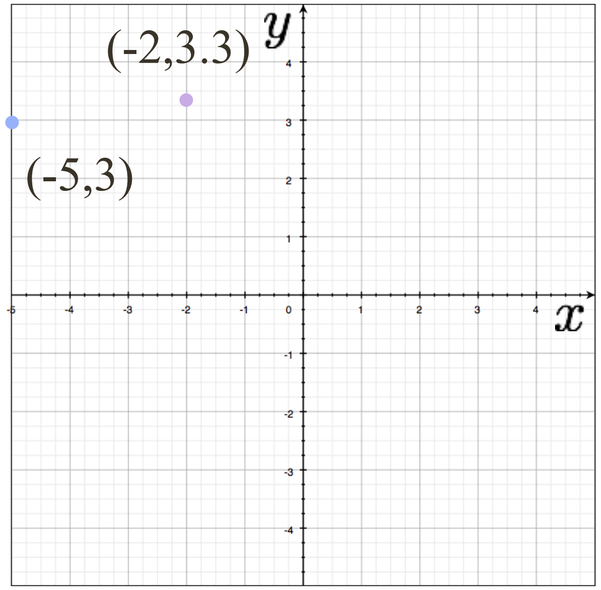


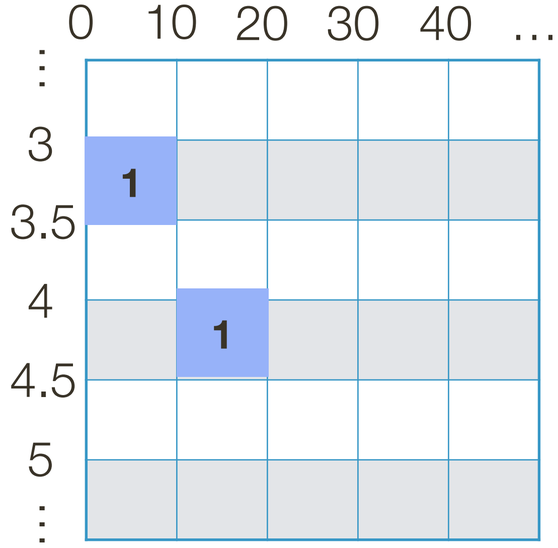


$$-5\sin(5°) - 3\cos(5°) + r = 0 => r = 3.42$$

$$-5\sin(15°) - 3\cos(15°) + r = 0 => r = 4.18$$

Example: Hough Transform for Lines

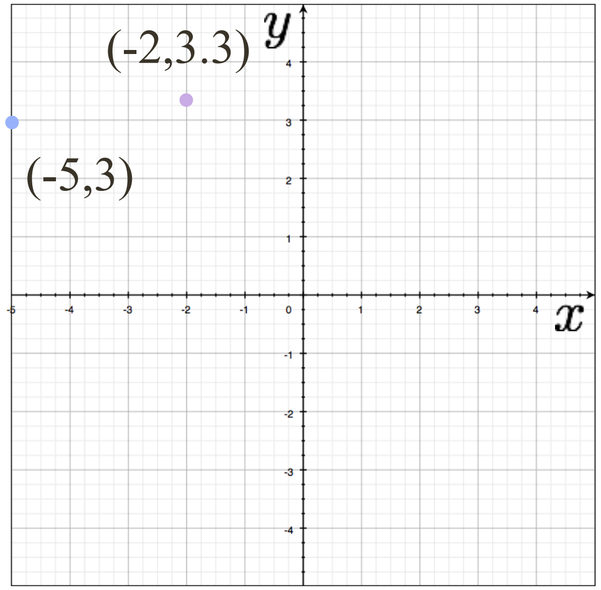


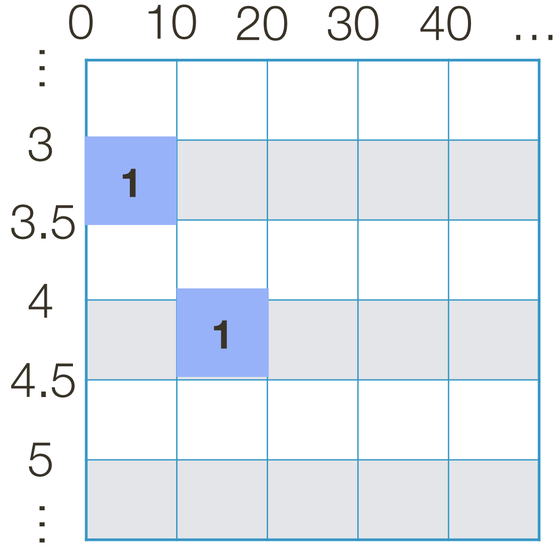


$$-5\sin(5°) - 3\cos(5°) + r = 0 => r = 3.42$$

$$-5\sin(15°) - 3\cos(15°) + r = 0 => r = 4.18$$

Example: Hough Transform for Lines



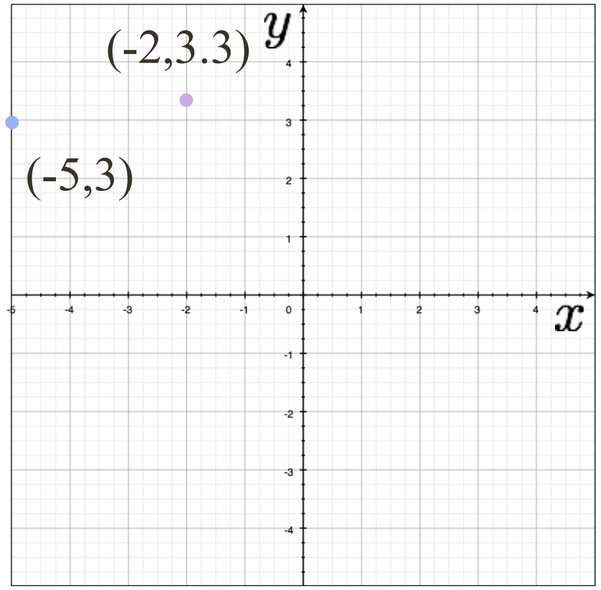


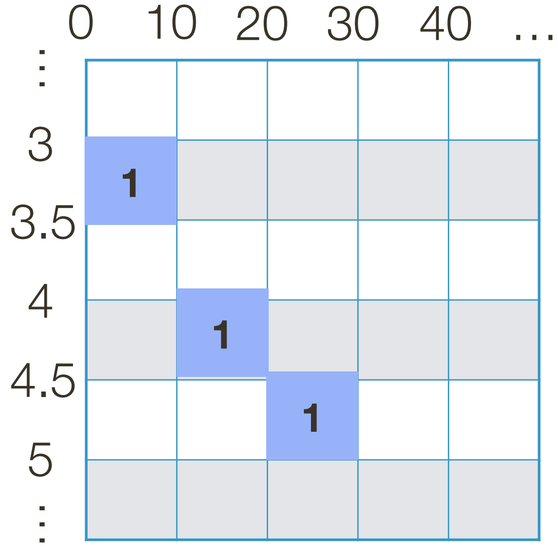
$$-5\sin(5°) - 3\cos(5°) + r = 0 => r = 3.42$$

$$-5\sin(15°) - 3\cos(15°) + r = 0 => r = 4.18$$

$$-5\sin(25°) - 3\cos(25°) + r = 0 => r = 4.83$$

Example: Hough Transform for Lines



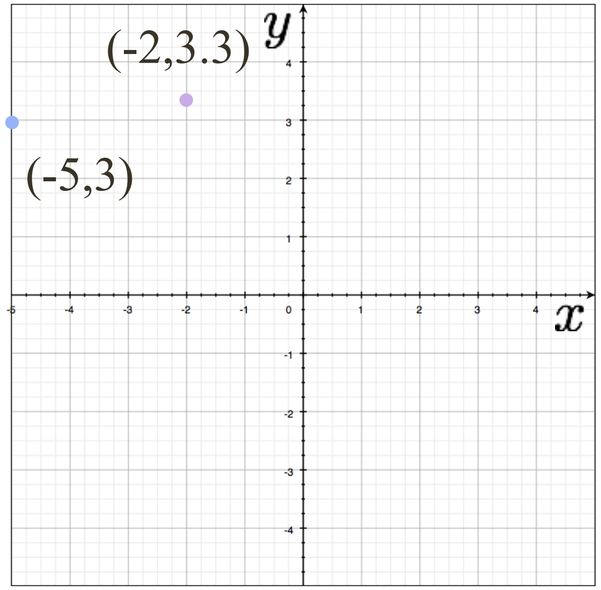


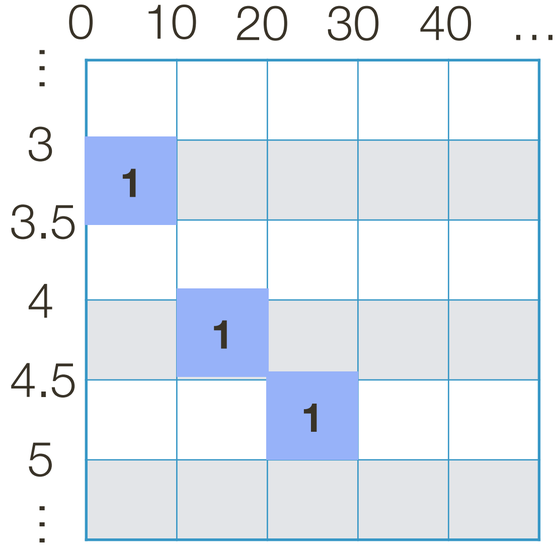
$$-5\sin(5°) - 3\cos(5°) + r = 0 => r = 3.42$$

$$-5\sin(15°) - 3\cos(15°) + r = 0 => r = 4.18$$

$$-5\sin(25°) - 3\cos(25°) + r = 0 => r = 4.83$$

Example: Hough Transform for Lines





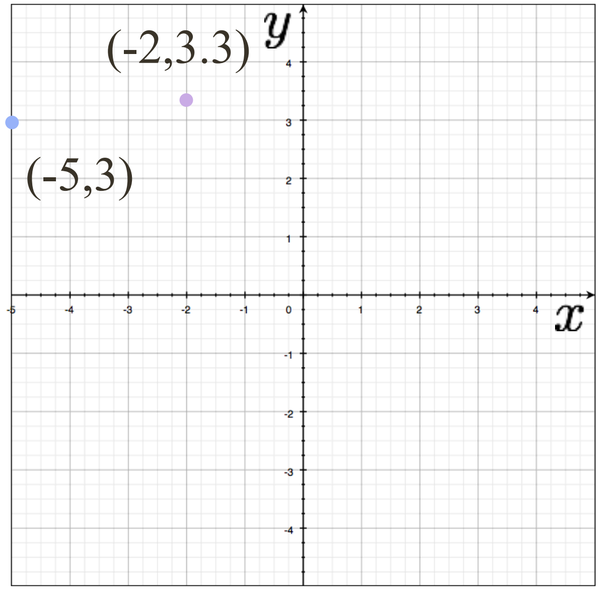
$$-5\sin(5^\circ) - 3\cos(5^\circ) + r = 0 => r = 3.42$$

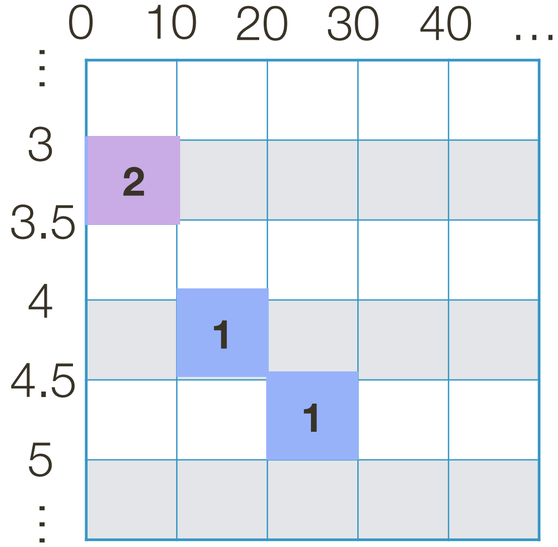
$$-5\sin(15^\circ) - 3\cos(15^\circ) + r = 0 => r = 4.18$$

$$-5\sin(25^\circ) - 3\cos(25^\circ) + r = 0 => r = 4.83$$

$$-2\sin(5^\circ) - 3.3\cos(5^\circ) + r = 0 => r = 3.46$$

Example: Hough Transform for Lines





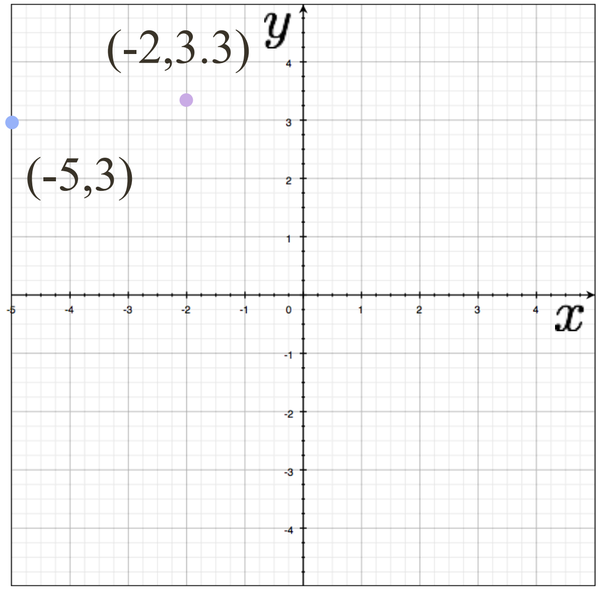
$$-5\sin(5°) - 3\cos(5°) + r = 0 => r = 3.42$$

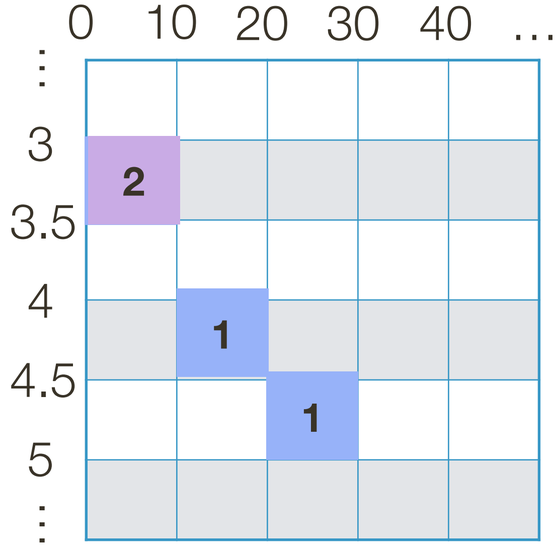
$$-5\sin(15°) - 3\cos(15°) + r = 0 => r = 4.18$$

$$-5\sin(25°) - 3\cos(25°) + r = 0 => r = 4.83$$

$$-2\sin(5°) - 3.3\cos(5°) + r = 0 => r = 3.46$$

Example: Hough Transform for Lines





$$-5\sin(5°) - 3\cos(5°) + r = 0 => r = 3.42$$

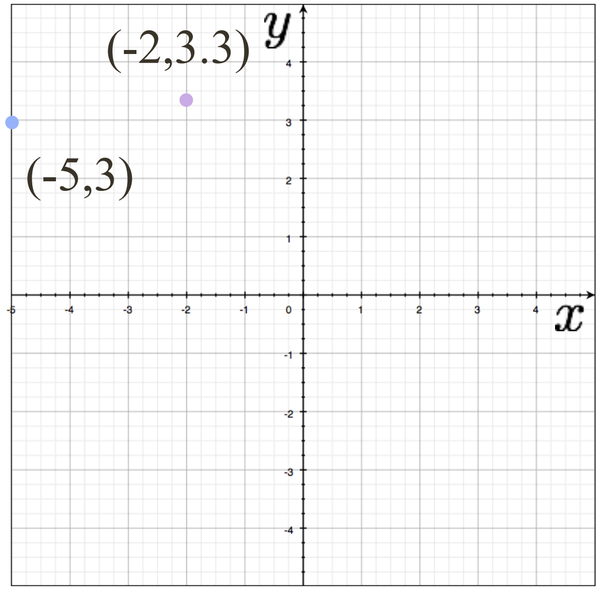
$$-5\sin(15°) - 3\cos(15°) + r = 0 => r = 4.18$$

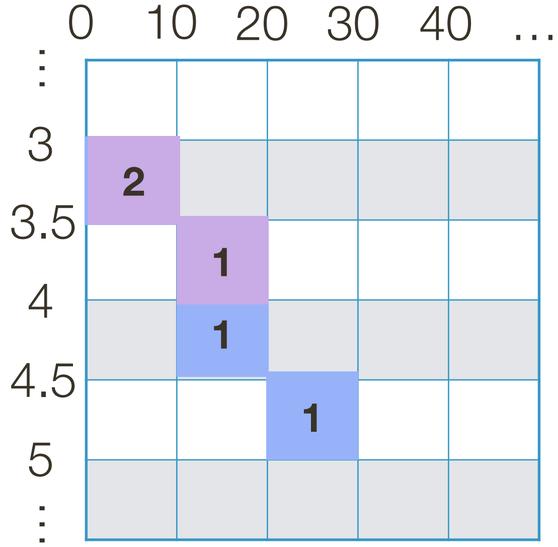
$$-5\sin(25°) - 3\cos(25°) + r = 0 => r = 4.83$$

$$-2\sin(5°) - 3.3\cos(5°) + r = 0 => r = 3.46$$

$$-2\sin(15°) - 3.3\cos(15°) + r = 0 => r = 3.71$$

Example: Hough Transform for Lines





$$-5\sin(5°) - 3\cos(5°) + r = 0 => r = 3.42$$

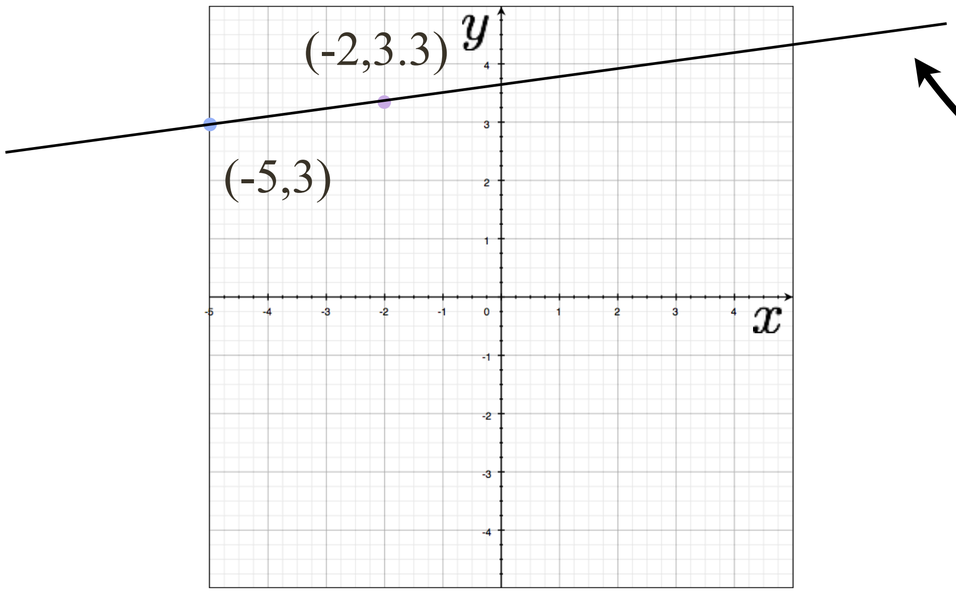
$$-5\sin(15°) - 3\cos(15°) + r = 0 => r = 4.18$$

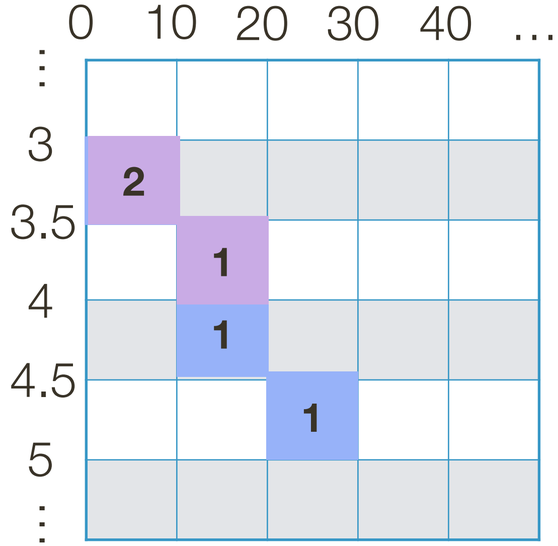
$$-5\sin(25°) - 3\cos(25°) + r = 0 => r = 4.83$$

$$-2\sin(5°) - 3.3\cos(5°) + r = 0 => r = 3.46$$

$$-2\sin(15°) - 3.3\cos(15°) + r = 0 => r = 3.71$$

Example: Hough Transform for Lines





$$-5\sin(5°) - 3\cos(5°) + r = 0 => r = 3.42$$

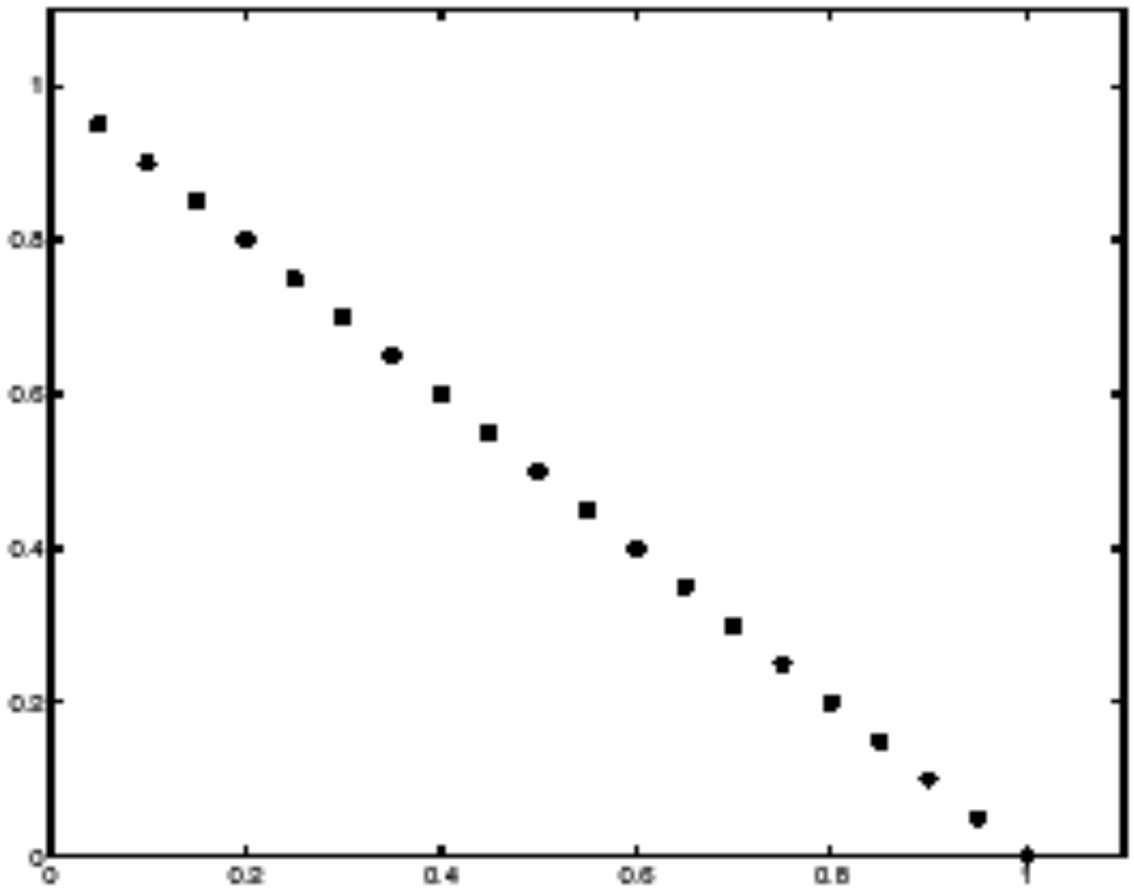
$$-5\sin(15°) - 3\cos(15°) + r = 0 => r = 4.18$$

$$-5\sin(25°) - 3\cos(25°) + r = 0 => r = 4.83$$

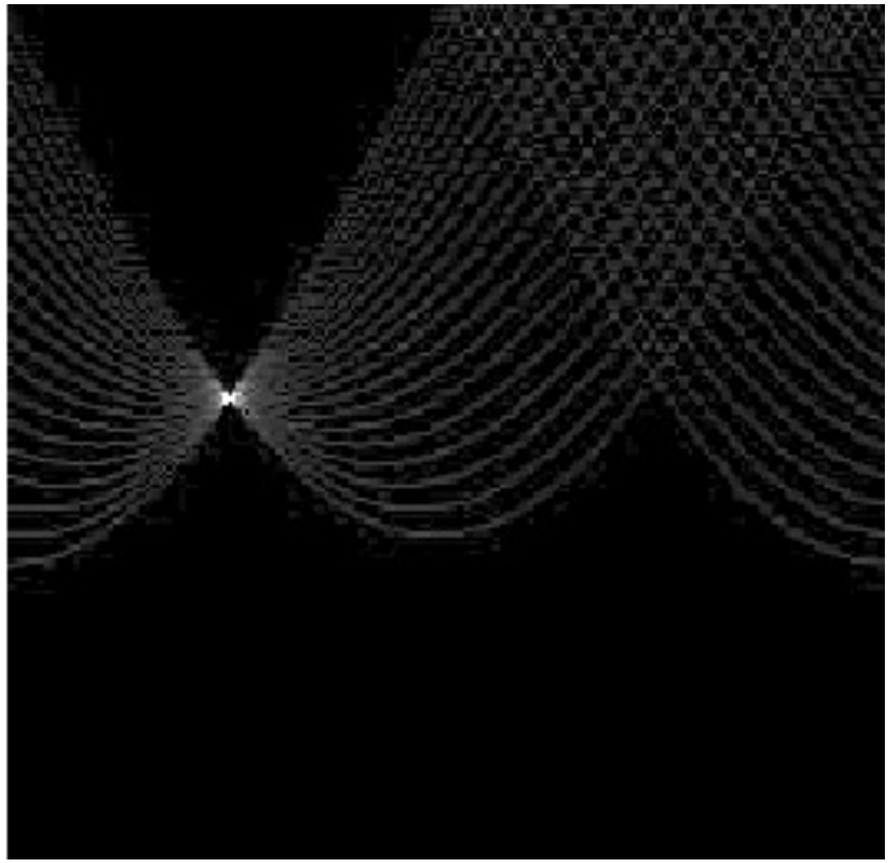
$$-2\sin(5°) - 3.3\cos(5°) + r = 0 => r = 3.46$$

$$-2\sin(15°) - 3.3\cos(15°) + r = 0 => r = 3.71$$

Example: Clean Data



Tokens



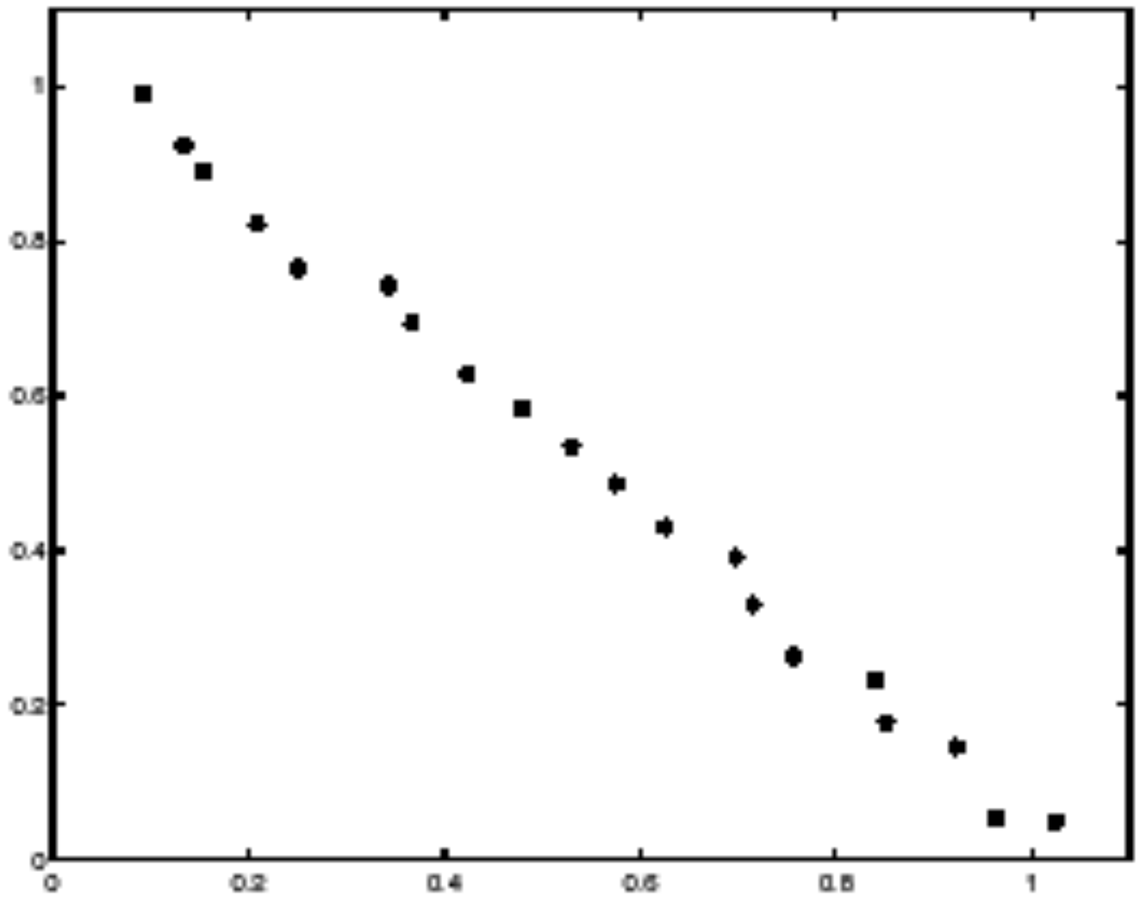
Votes

Horizontal axis is θ

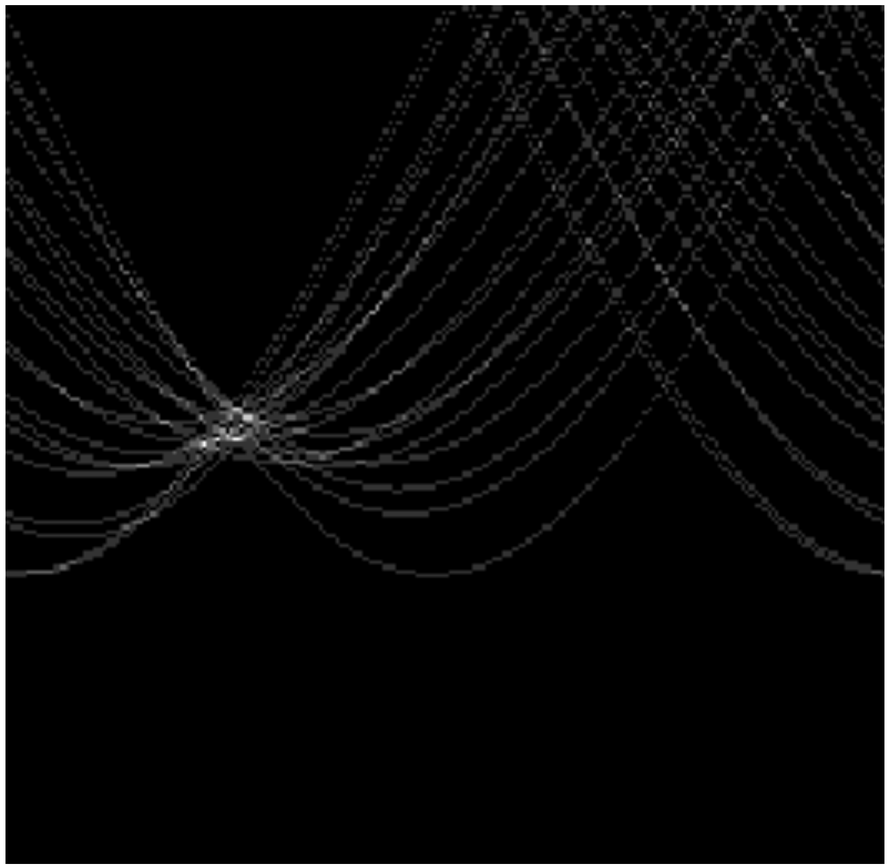
Vertical Axis is r

Forsyth & Ponce (2nd ed.) Figure 10.1 (Top)

Example: Some Noise



Tokens



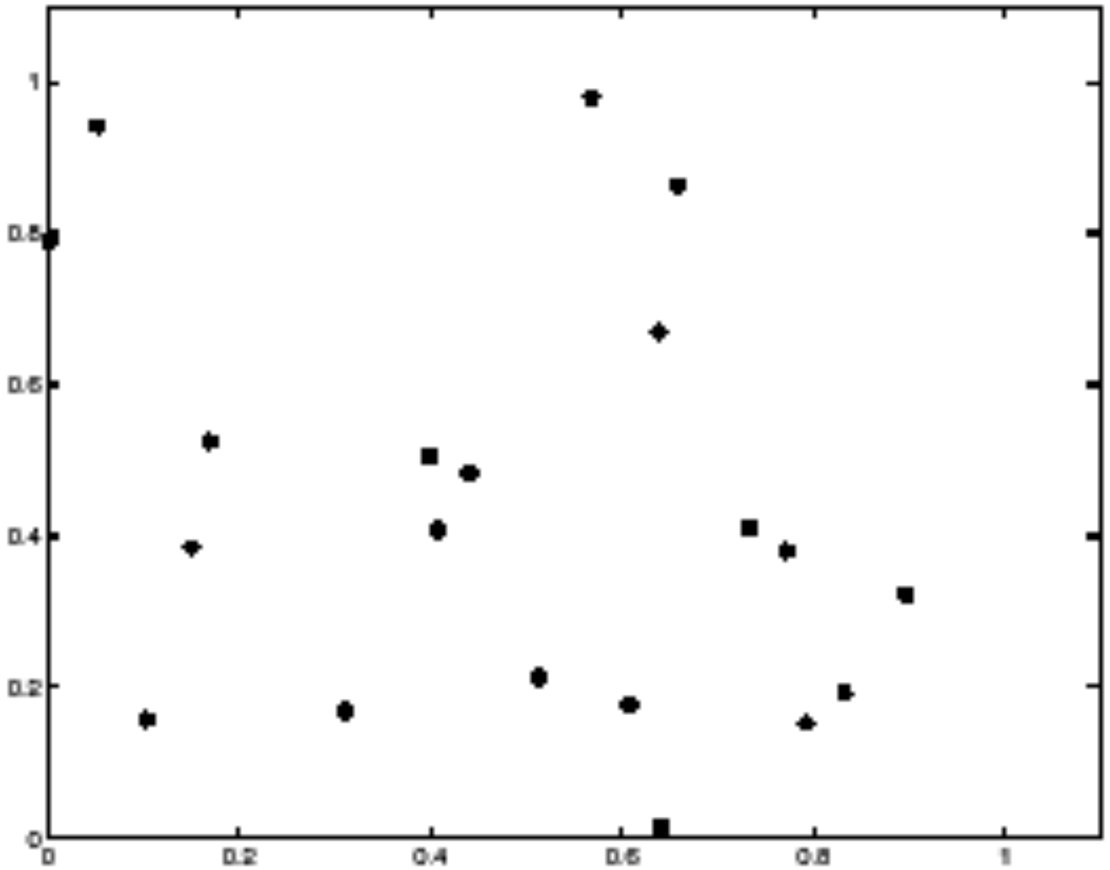
Votes

Horizontal axis is θ

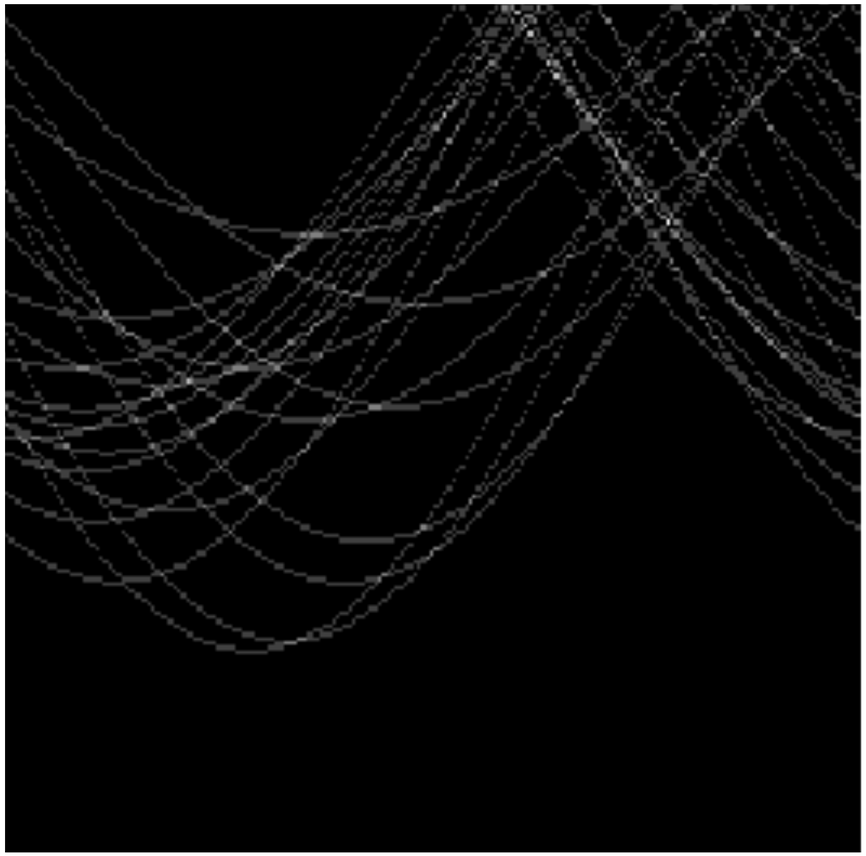
Vertical Axis is r

Forsyth & Ponce (2nd ed.) Figure 10.1 (Bottom)

Example: Too Much Noise



Tokens



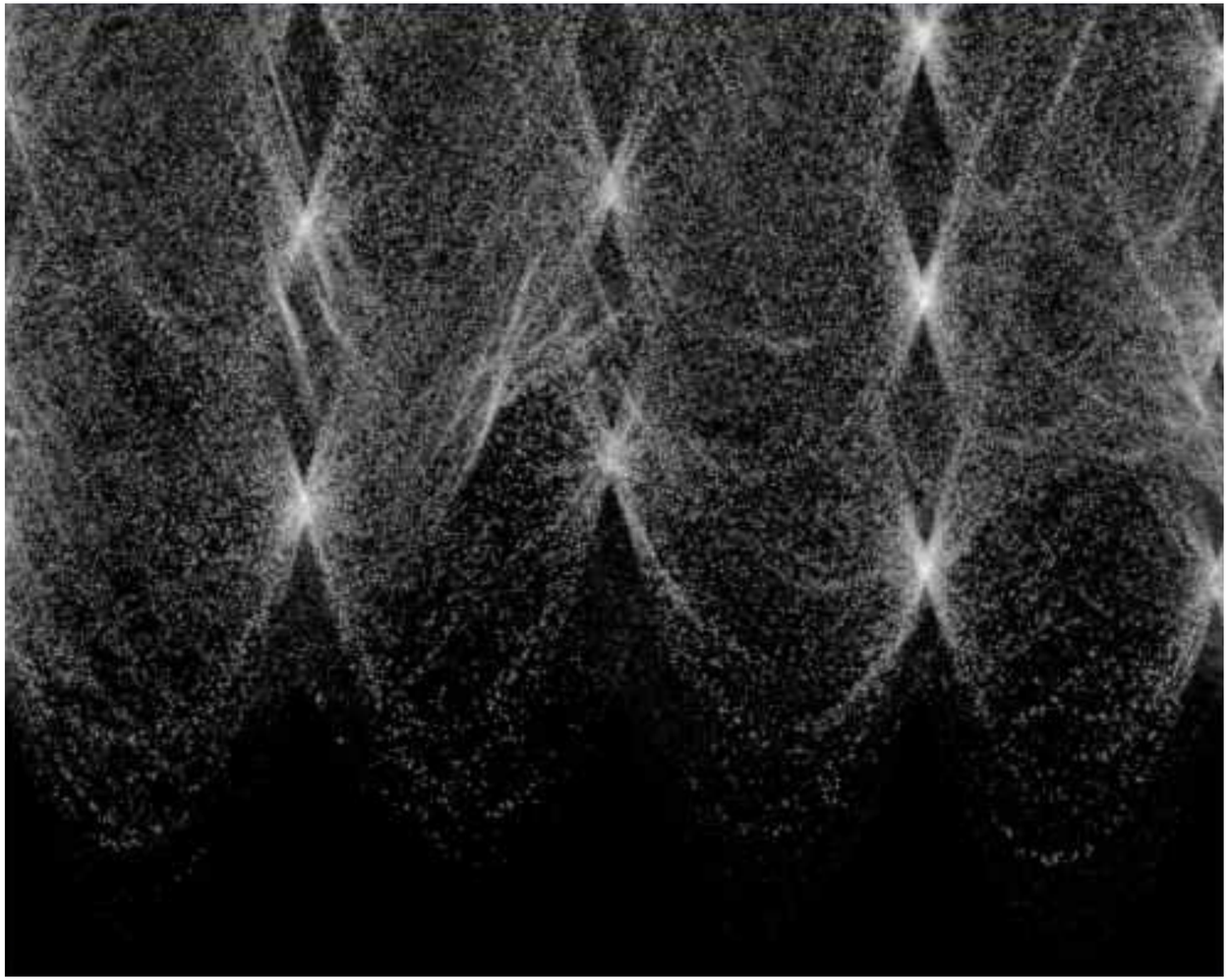
Votes

Horizontal axis is θ

Vertical Axis is r

Forsyth & Ponce (2nd ed.) Figure 10.2

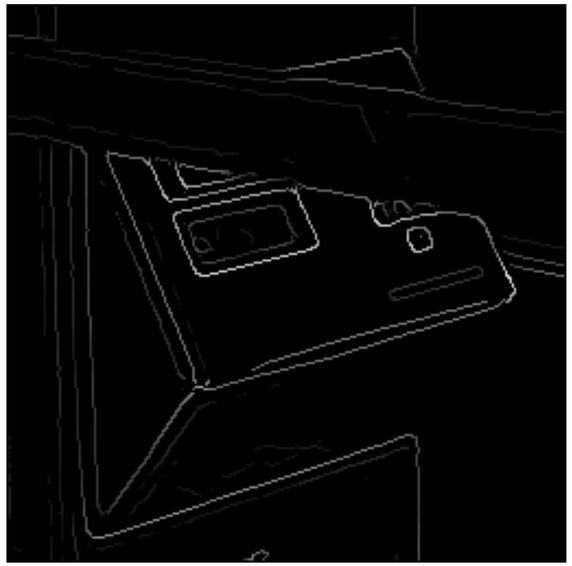
Real World **Example**



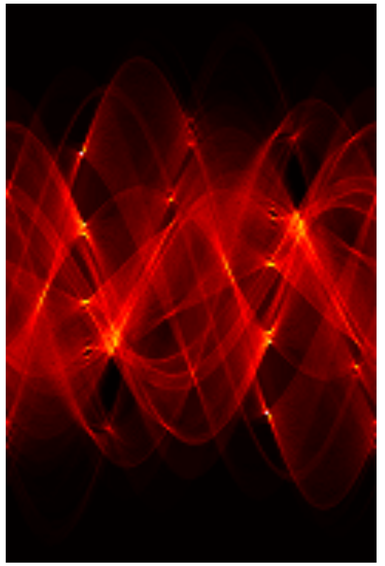
Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Real World **Example**

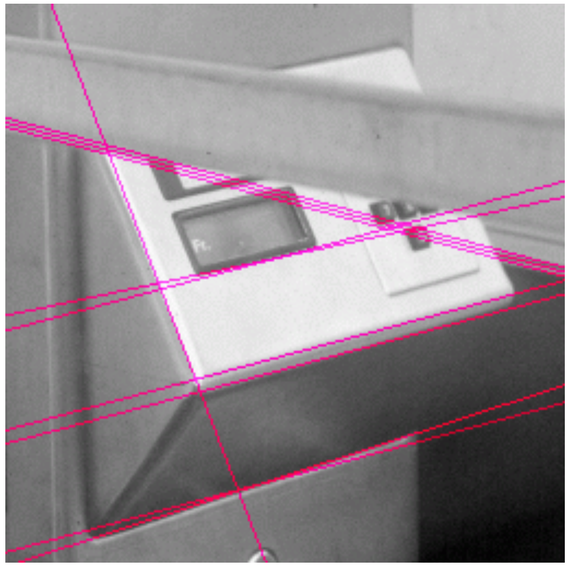
Original



Edges



Parameter space



Hough Lines

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Mechanics of Hough Transform

1. Construct a quantized array to represent θ and r

2. For each point, render curve (θ, r) into this array adding one vote at each cell

Difficulties:
— How big should the cells be? (too big, and we merge quite different lines; too small, and noise causes lines to be missed)

How many lines?

— Count the peaks in the Hough array

— Treat adjacent peaks as a single peak

Some **Practical Details** of Hough Transform

It is best to **vote** for the two closest bins in each dimension, as the locations of the bin boundaries are arbitrary

— This means that peaks are “blurred” and noise will not cause similar votes to fall into separate bins

Can use a **hash table** rather than an array to store the votes

— This means that no effort is wasted on initializing and checking empty bins

— It avoids the need to predict the maximum size of the array, which can be non-rectangular

Some **Practical Details** of Hough Transform

A key is to have each feature (token) determine as many parameters as possible
— Lines are detected more effectively from edge elements with

both position and orientation

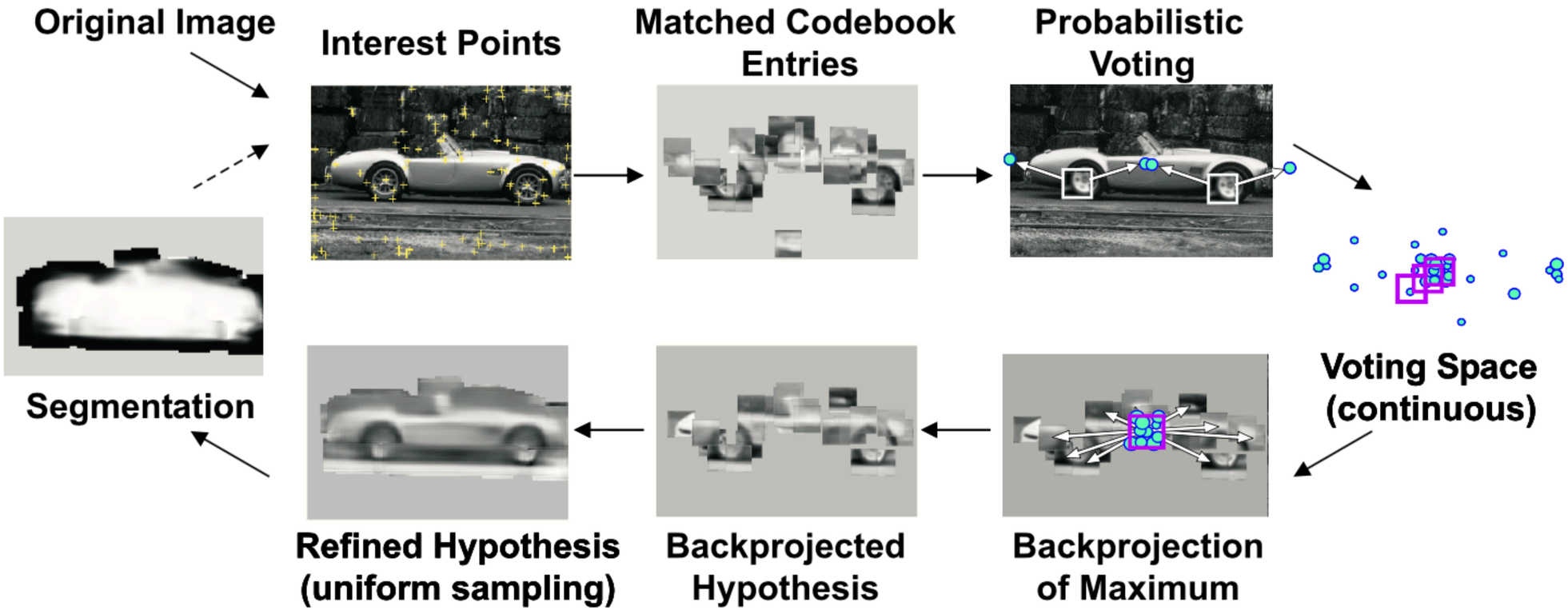
— For object recognition, each token should predict **position,**

orientation, and scale

The Hough transform can extract feature groupings from clutter in linear time

Example 1: Object Recognition — Implicit Shape Model

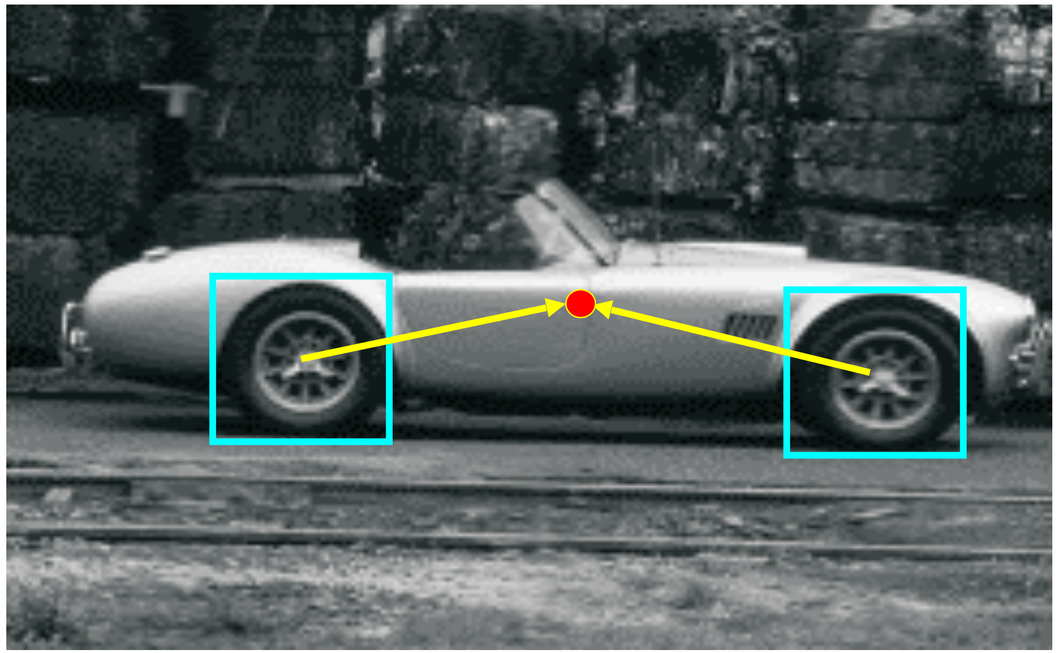
Combined object detection and segmentation using an implicit shape model. Image patches cast weighted votes for the object centroid.



B. Leibe, A. Leonardis, and B. Schiele, Combined Object Categorization and Segmentation with an Implicit Shape Model, ECCV Workshop on Statistical Learning in Computer Vision 2004

Example 1: Object Recognition — Implicit Shape Model

Index displacements by “visual codeword”



training image

visual codeword with
displacement vectors

B. Leibe, A. Leonardis, and B. Schiele, Combined Object Categorization and Segmentation with an Implicit Shape Model, ECCV Workshop on Statistical Learning in Computer Vision 2004

Example 1: Object Recognition — Implicit Shape Model

B. Leibe, A. Leonardis, and B. Schiele, Combined Object Categorization and Segmentation with an Implicit Shape Model, ECCV Workshop on Statistical Learning in Computer Vision 2004

Example 2: Object Recognition — Boundary Fragments

Boundary fragments cast weighted votes for the object centroid. Also obtains an estimate of the object's contour.

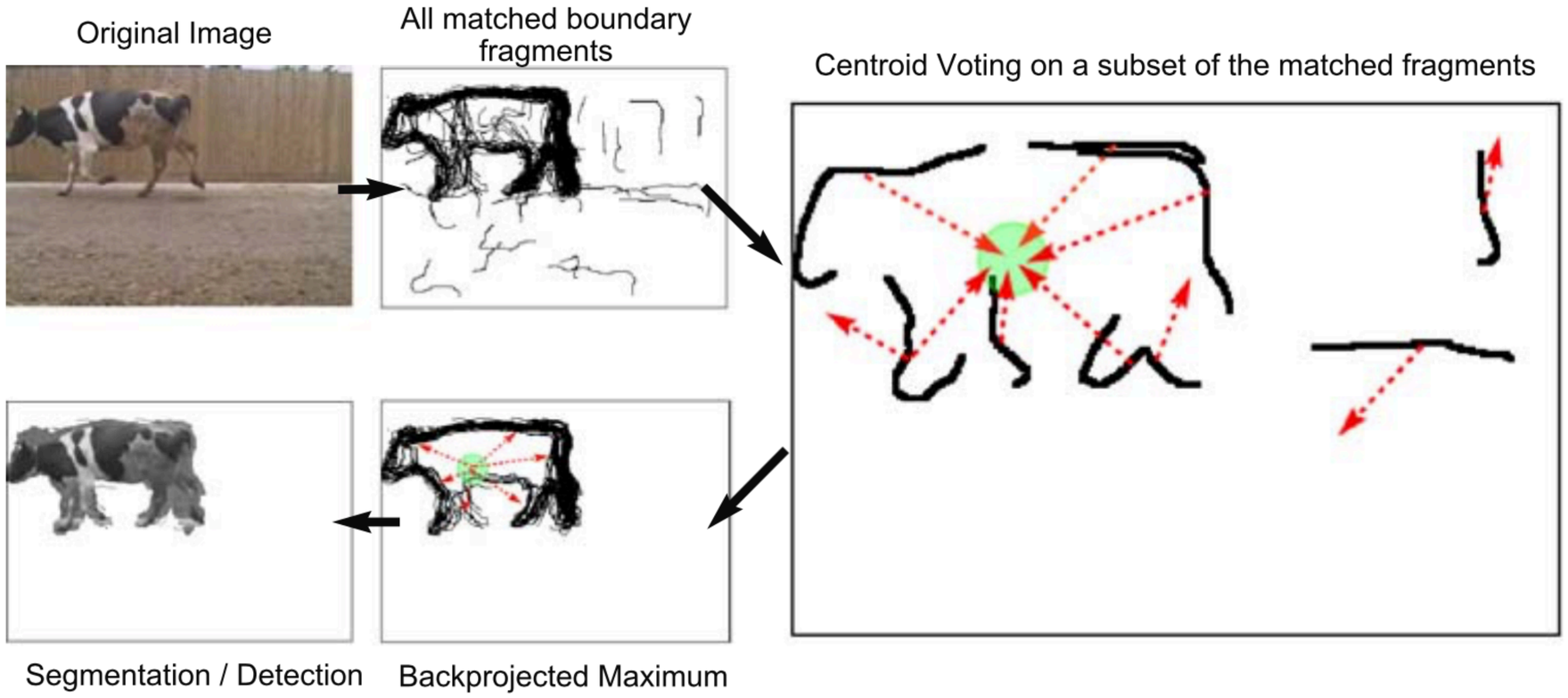


Image credit: Opelt et al., 2006

Example 2: Object Recognition — Boundary Fragments

Boundary fragments cast weighted votes for the object centroid. Also obtains an estimate of the object's contour.

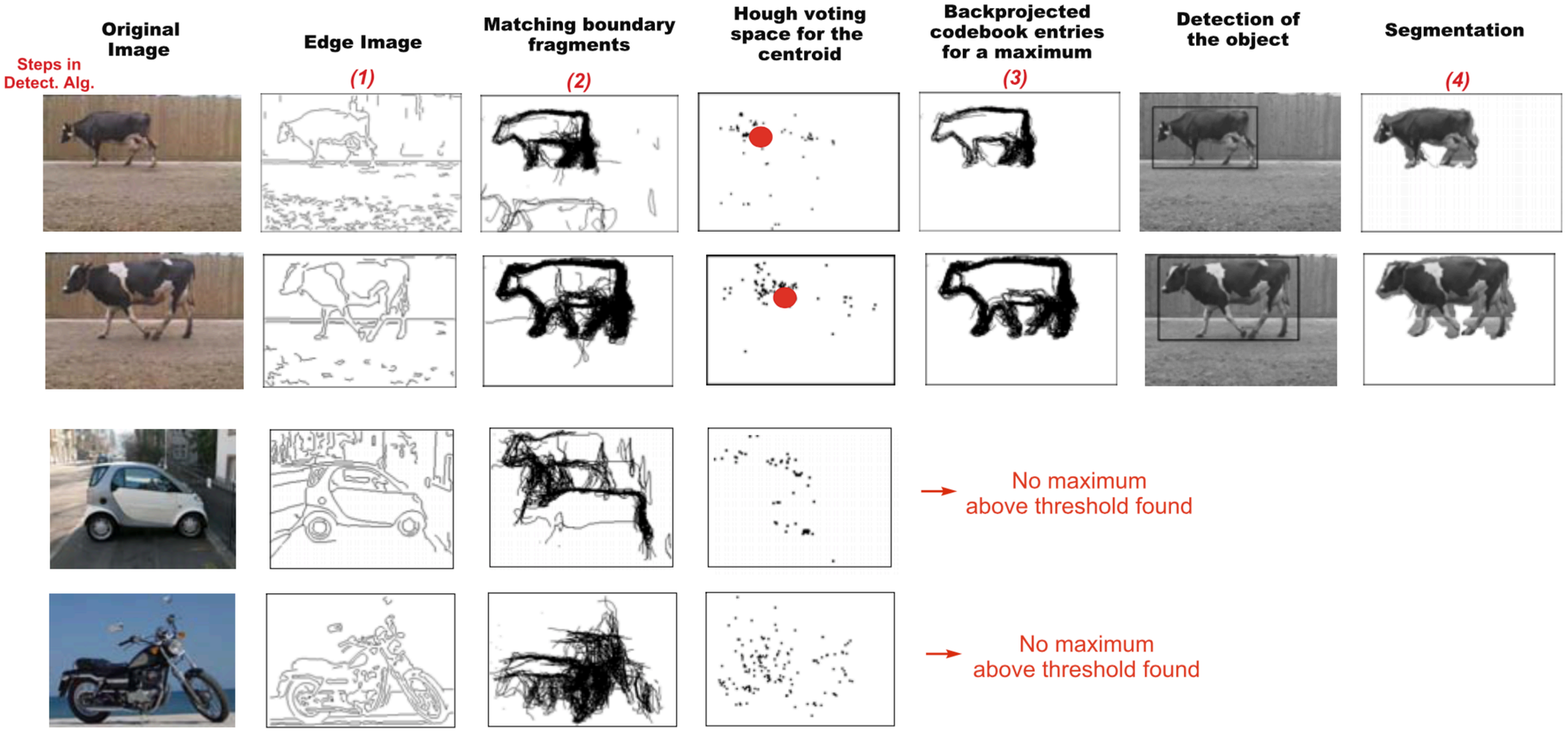


Image credit: Opelt et al., 2006

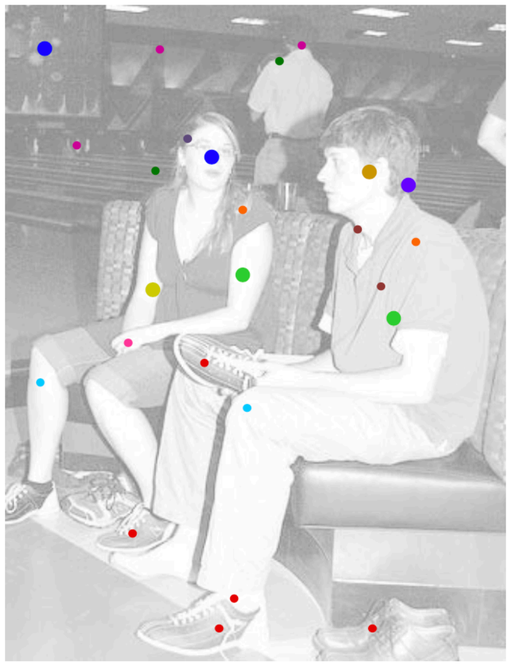
Example 3: Object Recognition — Poselets

Poselets are image patches that have distinctive appearance and can be used to infer some of the configuration of a parts-based object. Detected poselets vote for the object configuration.

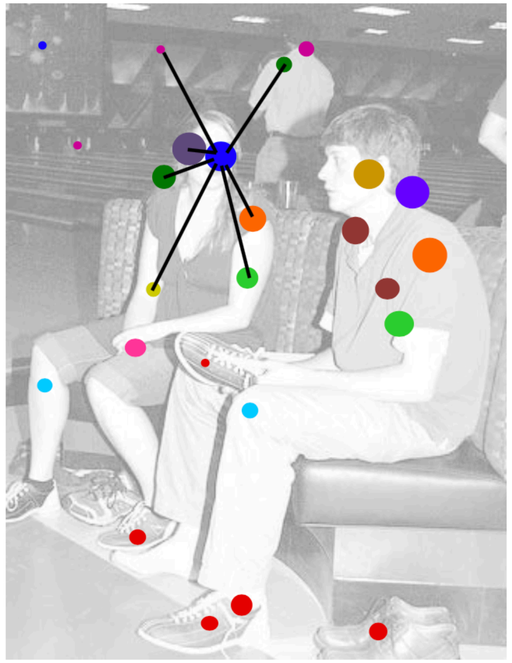
Image credit: Bourdev and Malik, 2009

Example 3: Object Recognition — Poselets

Poselets are image patches that have distinctive appearance and can be used to infer some of the configuration of a parts-based object. Detected poselets vote for the object configuration.

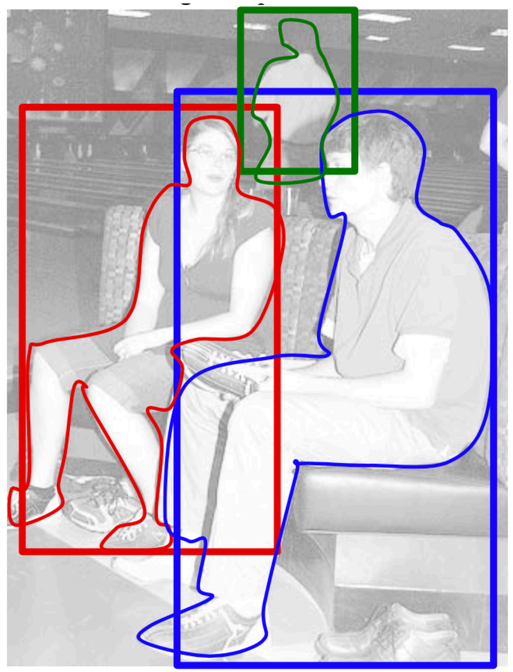


1. q-scores. Different colors illustrate different poselet detectors firing in the image. The blob size illustrates the score of the independent poselet classifier.



2. Q-scores (Section 4). Evidence from consistent poselet activations leads to a reranking based on mutual activation (Q-scores). Weaker activations consistent with others gain importance, whereas inconsistent ones get damped.

3. Clustering (Section 5). Activations are merged in a greedy manner starting with the strongest activation. Merging is based on pairwise consistency.



4. Bounding boxes (Section 6) and segmentations (Section 7). We predict the visible bounds and the contour of the person using the poselets within the cluster.

Image credit: Bourdev and Malik, 2009

Discussion of Hough Transform

Advantages:

— Can handle high percentage of outliers: each point votes separately

— Can detect multiple instances of a model in a single pass

Disadvantages:

— Complexity of search time increases exponentially with the number of model parameters

— Can be tricky to pick a good bin size

Summary of Hough Transform

The **Hough transform** is another technique for fitting data to a model

— a voting procedure

— possible model parameters define a quantized accumulator array

— data points “vote" for compatible entries in the accumulator array

A key is to have each data point (token) constrain model parameters as tightly as possible