Fitting: The Hough transform
Voting schemes

- Let each feature vote for all the models that are compatible with it
- Hopefully the noise features will not vote consistently for any single model
- Missing data doesn’t matter as long as there are enough features remaining to agree on a good model
Hough transform

- An early type of voting scheme
- General outline:
  - Discretize parameter space into bins
  - For each feature point in the image, put a vote in every bin in the parameter space that could have generated this point
  - Find bins that have the most votes

Parameter space representation

- A line in the image corresponds to a point in Hough space

\[ y = m_0 x + b_0 \]

Source: S. Seitz
Parameter space representation

- What does a point $(x_0, y_0)$ in the image space map to in the Hough space?
Parameter space representation

• What does a point \((x_0, y_0)\) in the image space map to in the Hough space?
  • Answer: the solutions of \(b = -x_0m + y_0\)
  • This is a line in Hough space
Parameter space representation

- Where is the line that contains both \((x_0, y_0)\) and \((x_1, y_1)\)?

\[ b = -x_1 m + y_1 \]
Parameter space representation

- Where is the line that contains both \((x_0, y_0)\) and \((x_1, y_1)\)?
  - It is the intersection of the lines \(b = -x_0m + y_0\) and \(b = -x_1m + y_1\)
Parameter space representation

- Problems with the (m,b) space:
  - Unbounded parameter domain
  - Vertical lines require infinite m
Parameter space representation

- Problems with the (m,b) space:
  - Unbounded parameter domain
  - Vertical lines require infinite m

- Alternative: polar representation

Each point will add a sinusoid in the (θ,ρ) parameter space

\[ x \cos \theta + y \sin \theta = \rho \]
Algorithm outline

- Initialize accumulator $H$ to all zeros
- For each edge point $(x,y)$ in the image
  - For $\theta = 0$ to $180$
    - $\rho = x \cos \theta + y \sin \theta$
    - $H(\theta, \rho) = H(\theta, \rho) + 1$
  - end
- end
- Find the value(s) of $(\theta, \rho)$ where $H(\theta, \rho)$ is a local maximum
  - The detected line in the image is given by $\rho = x \cos \theta + y \sin \theta$
Basic illustration

features

votes
Other shapes

Square

Circle
Several lines
A more complicated image
Effect of noise

features
Effect of noise

Peak gets fuzzy and hard to locate
Effect of noise

- Number of votes for a line of 20 points with increasing noise:
Random points

Uniform noise can lead to spurious peaks in the array
Random points

- As the level of uniform noise increases, the maximum number of votes increases too:

![Graph showing the relationship between the number of noise points and the maximum number of votes. The x-axis represents the number of noise points, ranging from 20 to 200, and the y-axis represents the maximum number of votes, ranging from 3 to 12.]
Dealing with noise

• 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

• Increment neighboring bins (smoothing in accumulator array)

• Try to get rid of irrelevant features
  • Take only edge points with significant gradient magnitude
Incorporating image gradients

• Recall: when we detect an edge point, we also know its gradient direction
• But this means that the line is uniquely determined!

• Modified Hough transform:

For each edge point \((x,y)\)

\[
\theta = \text{gradient orientation at (x,y)}
\]
\[
\rho = x \cos \theta + y \sin \theta
\]
\[
H(\theta, \rho) = H(\theta, \rho) + 1
\]
end

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

\[
\theta = \tan^{-1} \left( \frac{\partial f}{\partial y} / \frac{\partial f}{\partial x} \right)
\]
Hough transform for circles

- How many dimensions will the parameter space have?
- Given an oriented edge point, what are all possible bins that it can vote for?
Hough transform for circles

image space

Hough parameter space

\((x, y) + r \nabla I(x, y)\)

\((x, y) - r \nabla I(x, y)\)
Hough transform for circles

- Conceptually equivalent procedure: for each \((x, y, r)\), draw the corresponding circle in the image and compute its “support”

Is this more or less efficient than voting with features?
Application in recognition

Hough circles vs. Laplacian blobs

Generalized Hough transform

• We want to find a template defined by its reference point (center) and several distinct types of andmark points in stable spatial configuration
Generalized Hough transform

- Template representation: for each type of landmark point, store all possible displacement vectors towards the center.

Template

Model
Generalized Hough transform

• Detecting the template:
  • For each feature in a new image, look up that feature type in the model and vote for the possible center locations associated with that type in the model.
Application in recognition

- Index displacements by “visual codeword”

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
Application in recognition

• Index displacements by “visual codeword”

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
Implicit shape models: Training

1. Build codebook of patches around extracted interest points using clustering (more on this later in the course)
Implicit shape models: Training

1. Build codebook of patches around extracted interest points using clustering
2. Map the patch around each interest point to closest codebook entry
Implicit shape models: Training

1. Build codebook of patches around extracted interest points using clustering
2. Map the patch around each interest point to closest codebook entry
3. For each codebook entry, store all positions it was found, relative to object center
Implicit shape models: Testing

1. Given test image, extract patches, match to codebook entry
2. Cast votes for possible positions of object center
3. Search for maxima in voting space
4. Extract weighted segmentation mask based on stored masks for the codebook occurrences
Additional examples

Implicit shape models: Details

• Supervised training
  • Need reference location and segmentation mask for each training car

• Voting space is continuous, not discrete
  • Clustering algorithm needed to find maxima

• How about dealing with scale changes?
  • Option 1: search a range of scales, as in Hough transform for circles
  • Option 2: use scale-covariant interest points

• Verification stage is very important
  • Once we have a location hypothesis, we can overlay a more detailed template over the image and compare pixel-by-pixel, transfer segmentation masks, etc.
Hough transform: Discussion

- **Pros**
  - Can deal with non-locality and occlusion
  - Can detect multiple instances of a model
  - Some robustness to noise: noise points unlikely to contribute consistently to any single bin

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

- Hough transform vs. RANSAC