Fitting

(Hough transform)

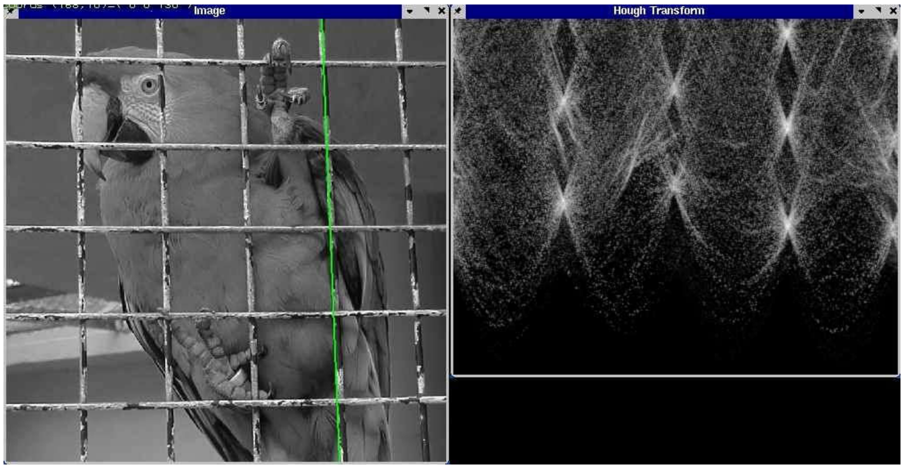
Lu Peng

School of Computer Science,

Beijing University of Posts and Telecommunications

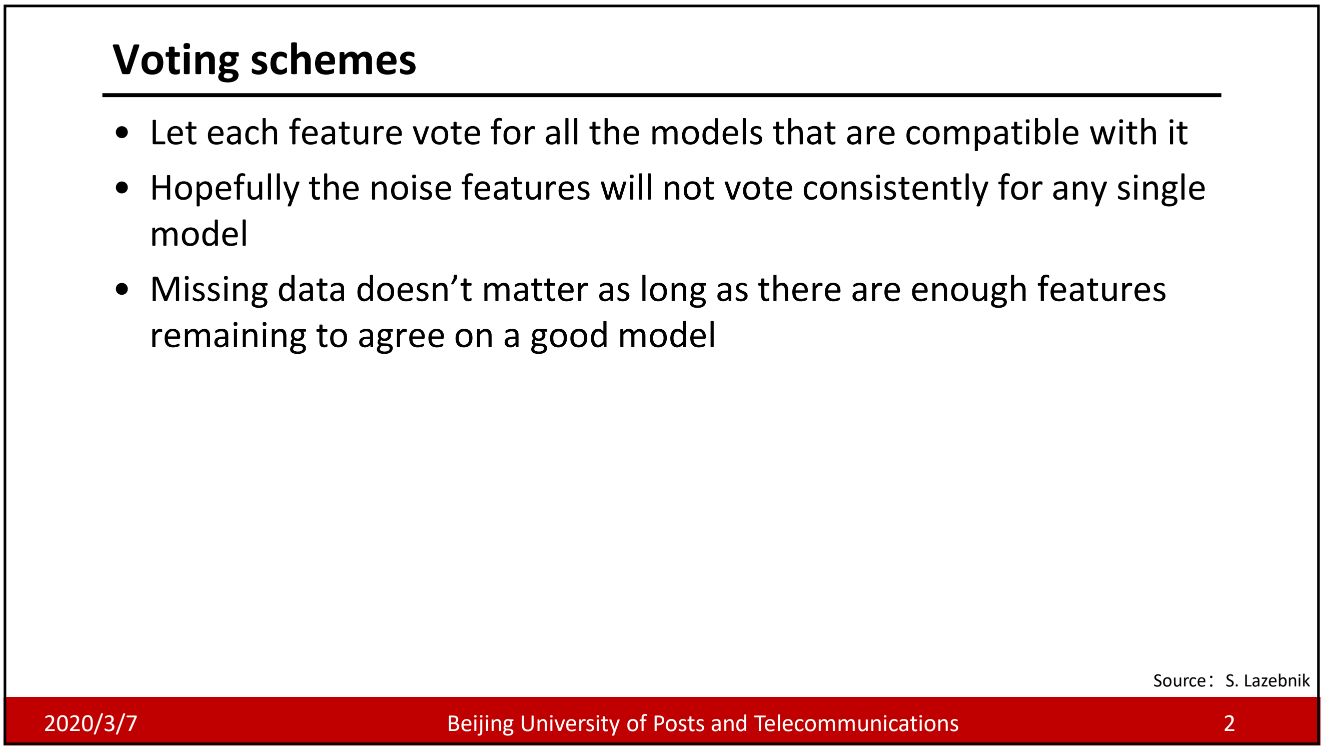
0

Fitting: The Hough transform



Source： S. Lazebnik

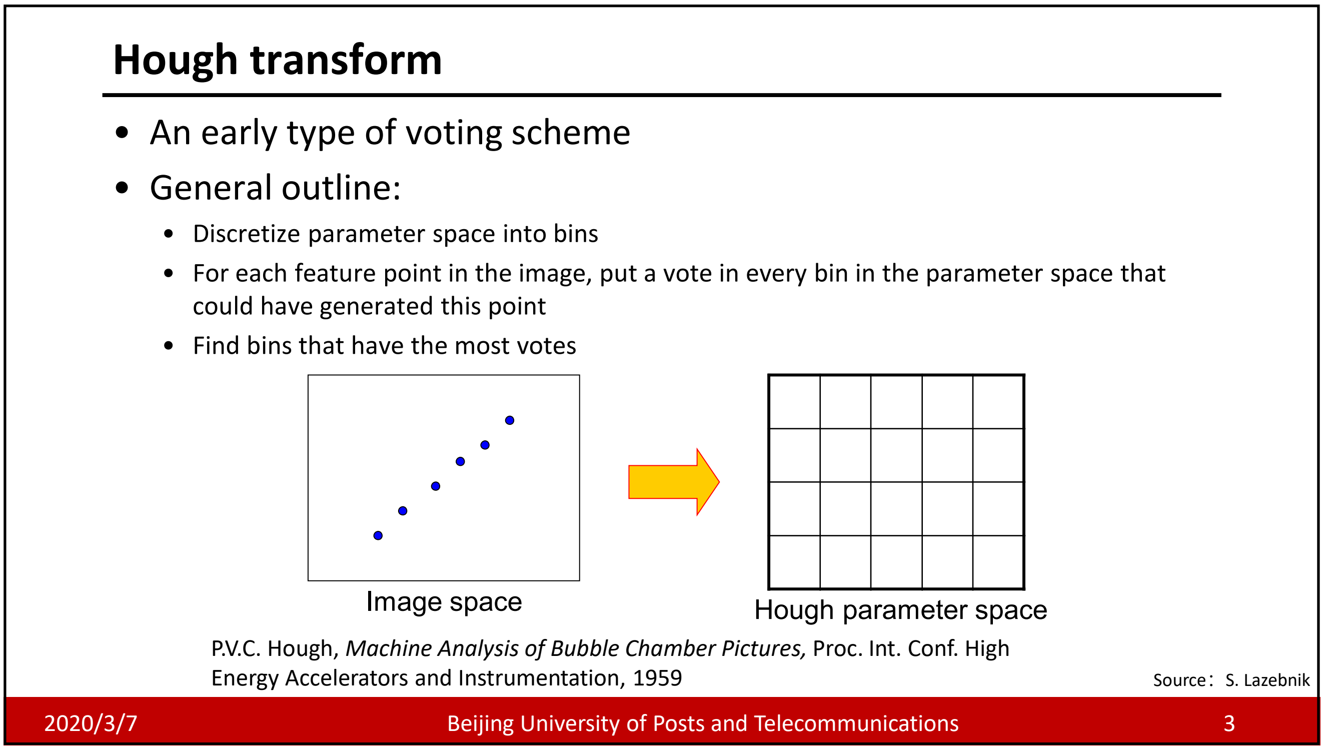
1



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

Source： S. Lazebnik



Hough transform

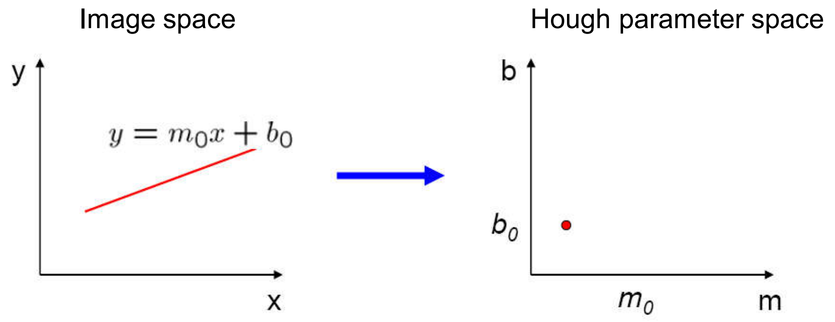
- An early type of voting scheme
- General outline:
  - Discretize parameter space into bins
  - For each feature point in the image, put a vote in every bin in the parameter space that could have generated this point
  - Find bins that have the most votes

P.V.C. Hough, *Machine Analysis of Bubble Chamber Pictures,* Proc. Int. Conf. High Energy Accelerators and Instrumentation, 1959

Source： S. Lazebnik

Parameter space representation

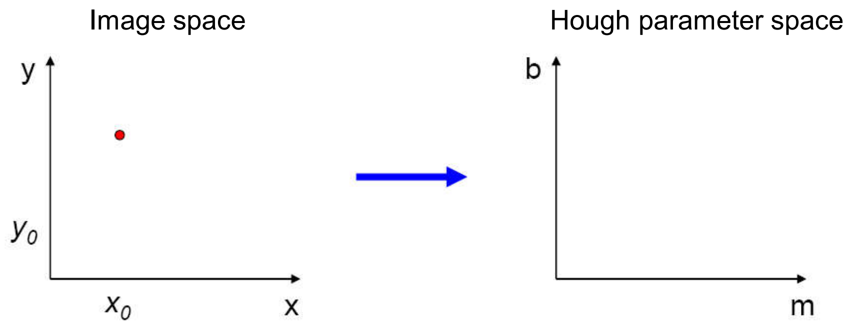
- A line in the image corresponds to a point in Hough space



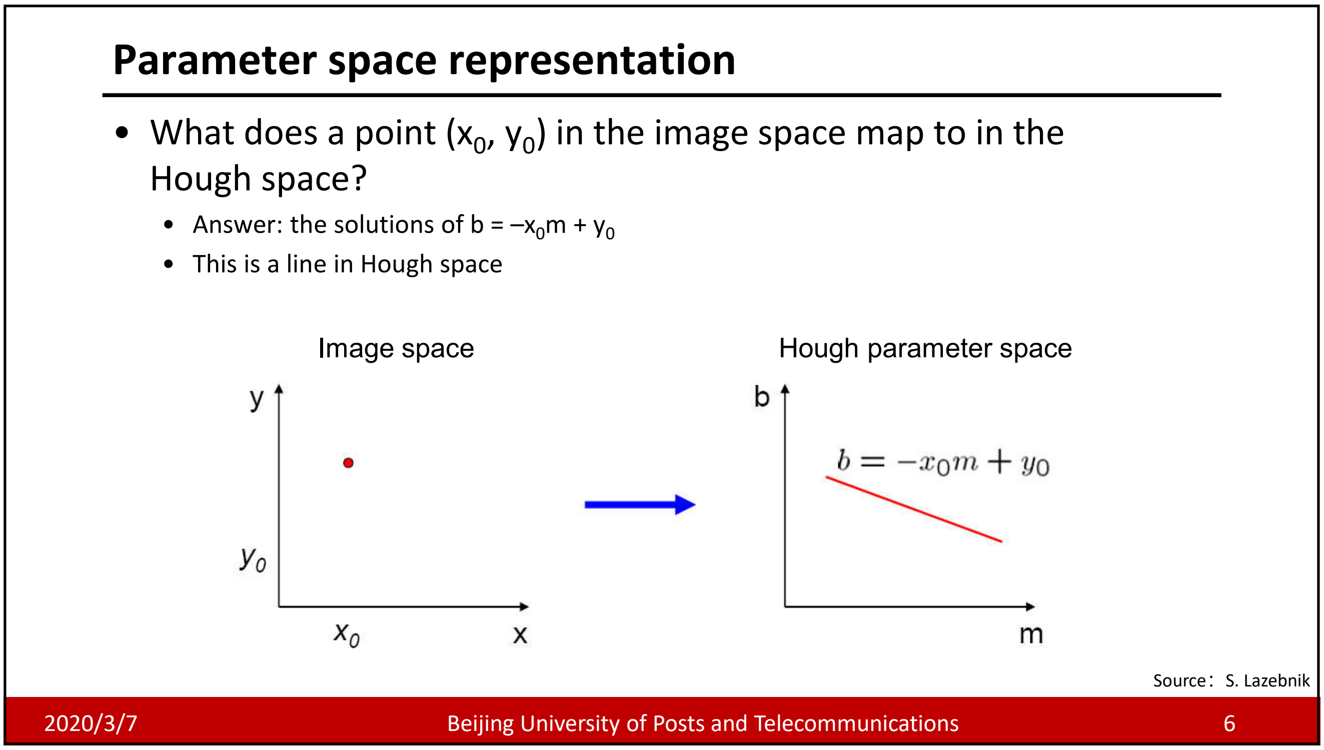
Source: S. Seitz

Parameter space representation

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



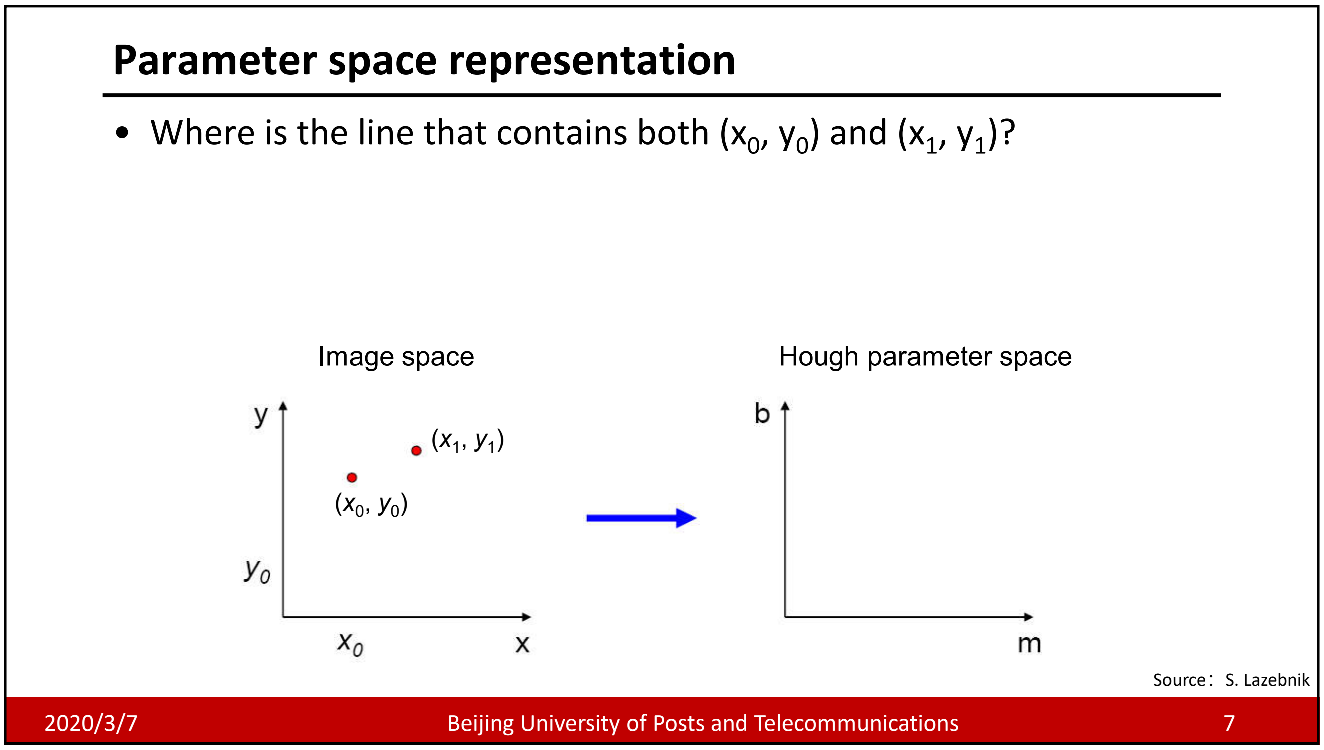
Source： S. Lazebnik



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

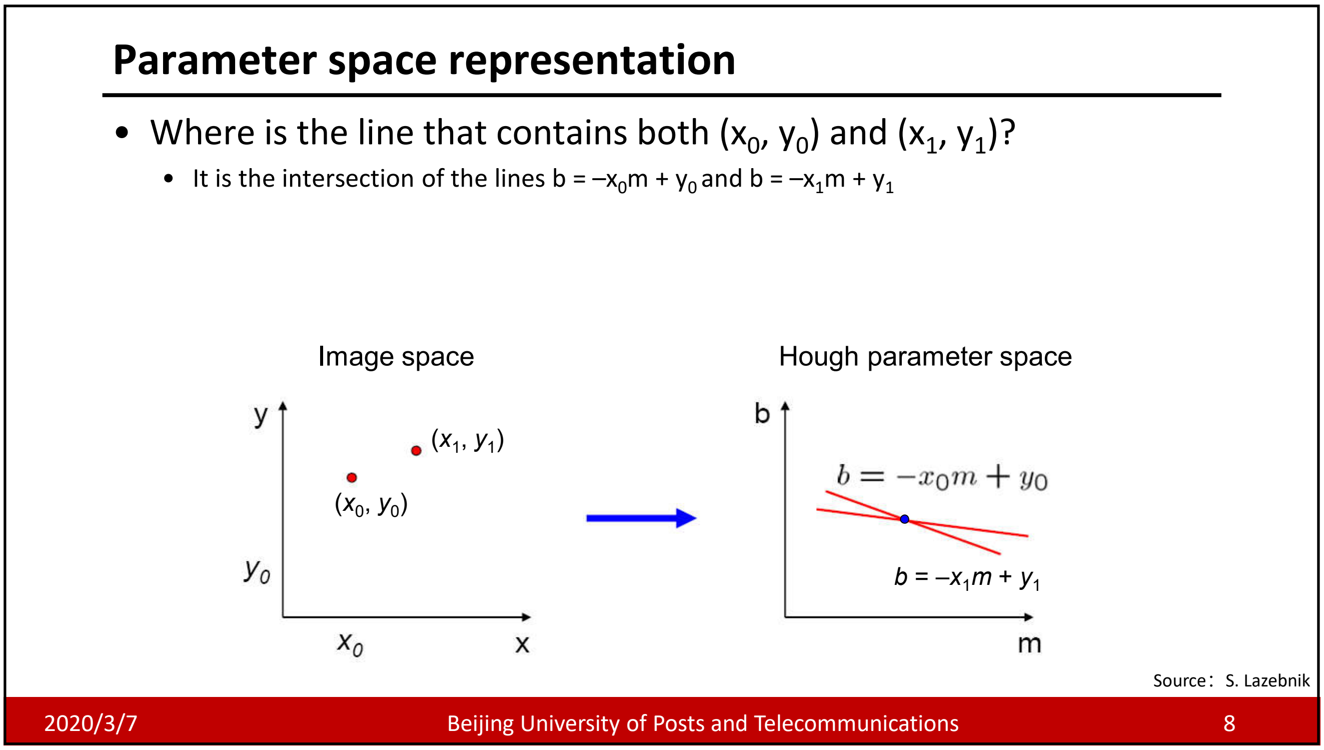
Source： S. Lazebnik



Parameter space representation

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

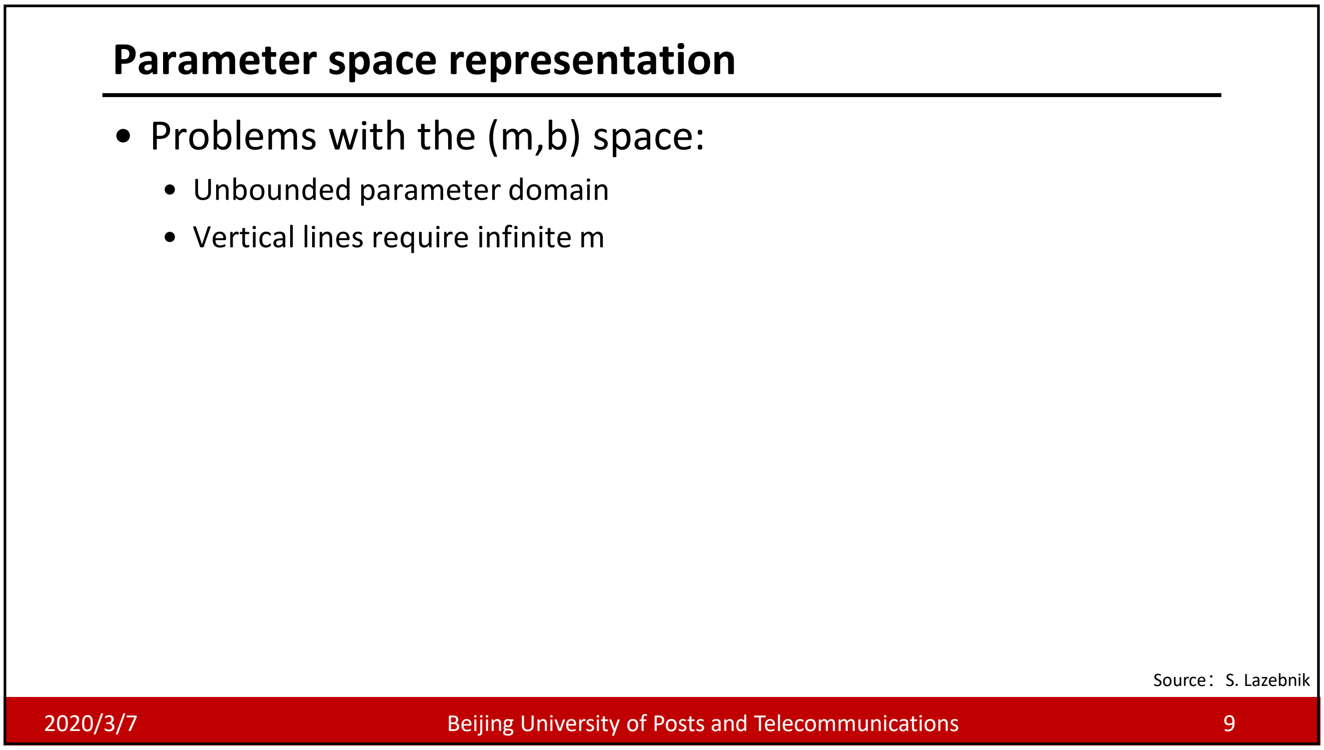
Source： S. Lazebnik



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 $b = -x_1m + y_1$

Source： S. Lazebnik



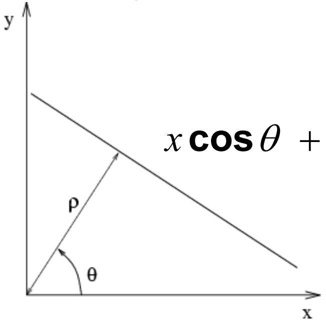
Parameter space representation

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

Source： S. Lazebnik

Parameter space representation

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



$$x\cos\theta + y\sin\theta = \rho$$

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

Source： S. Lazebnik

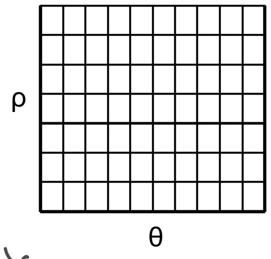
Algorithm outline

- Initialize accumulator H to all zeros
- For each edge point (x,y) in the image

```
For θ = 0 to 180
    ρ = x cos θ + y sin θ
    H(θ, ρ) = H(θ, ρ) + 1
end
end
```

- Find the value(s) of (θ, ρ) where H(θ, ρ) is a local maximum 被投票最多
  - The detected line in the image is given by ρ = x cos θ + y sin θ

H: accumulator array (votes)



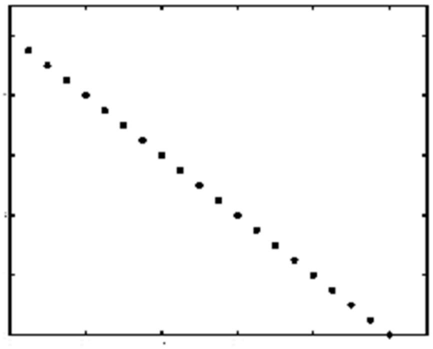
优化：

θ可以不用从[0,180]遍历

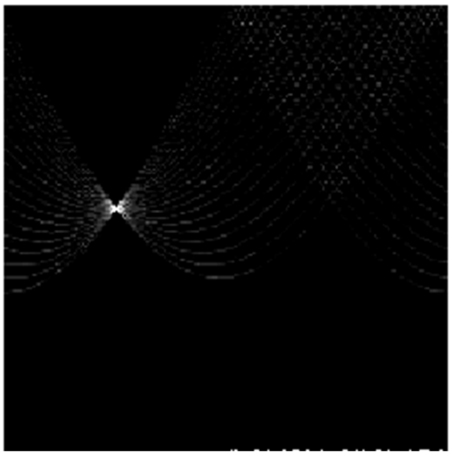
因为点有梯度方向，可以在这个方向周围遍历

Source： S. Lazebnik

Basic illustration



features

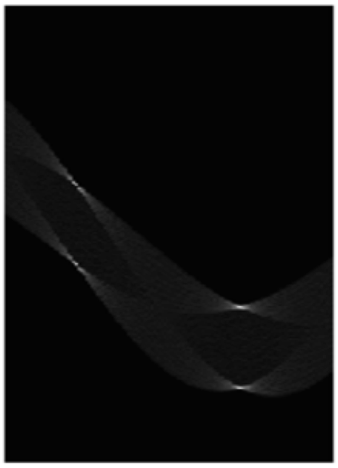


votes

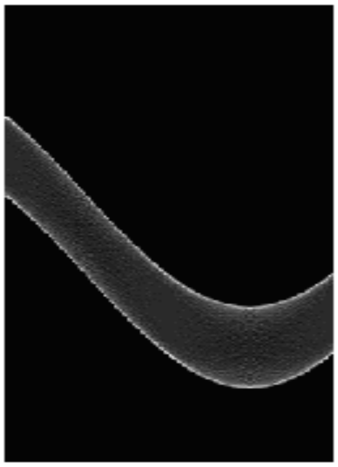
Source： S. Lazebnik

Other shapes

Square

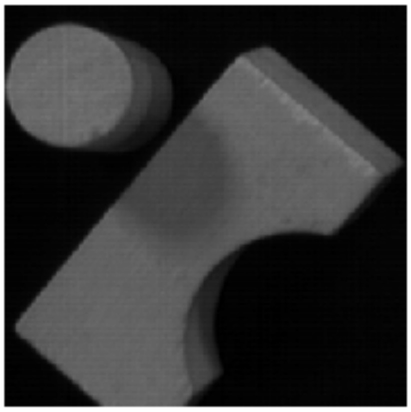


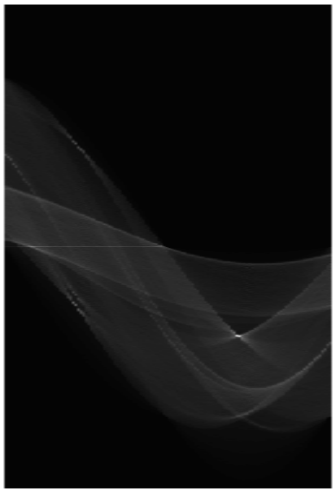
Circle



Source： S. Lazebnik

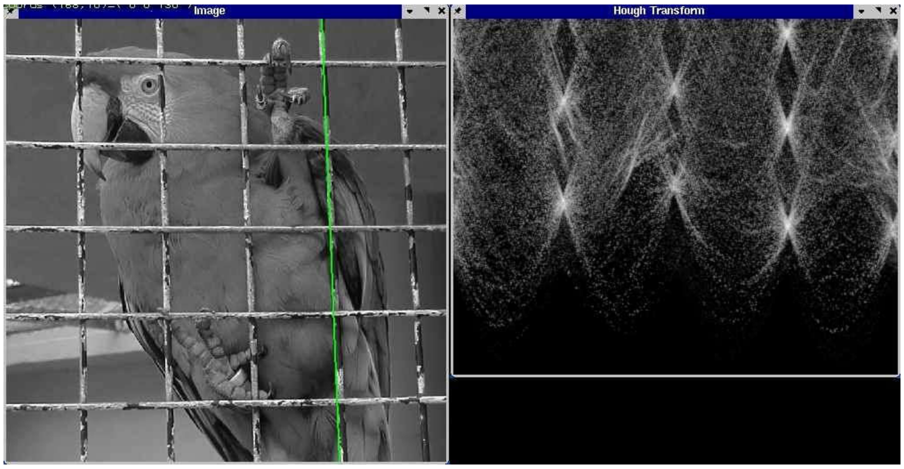
Several lines





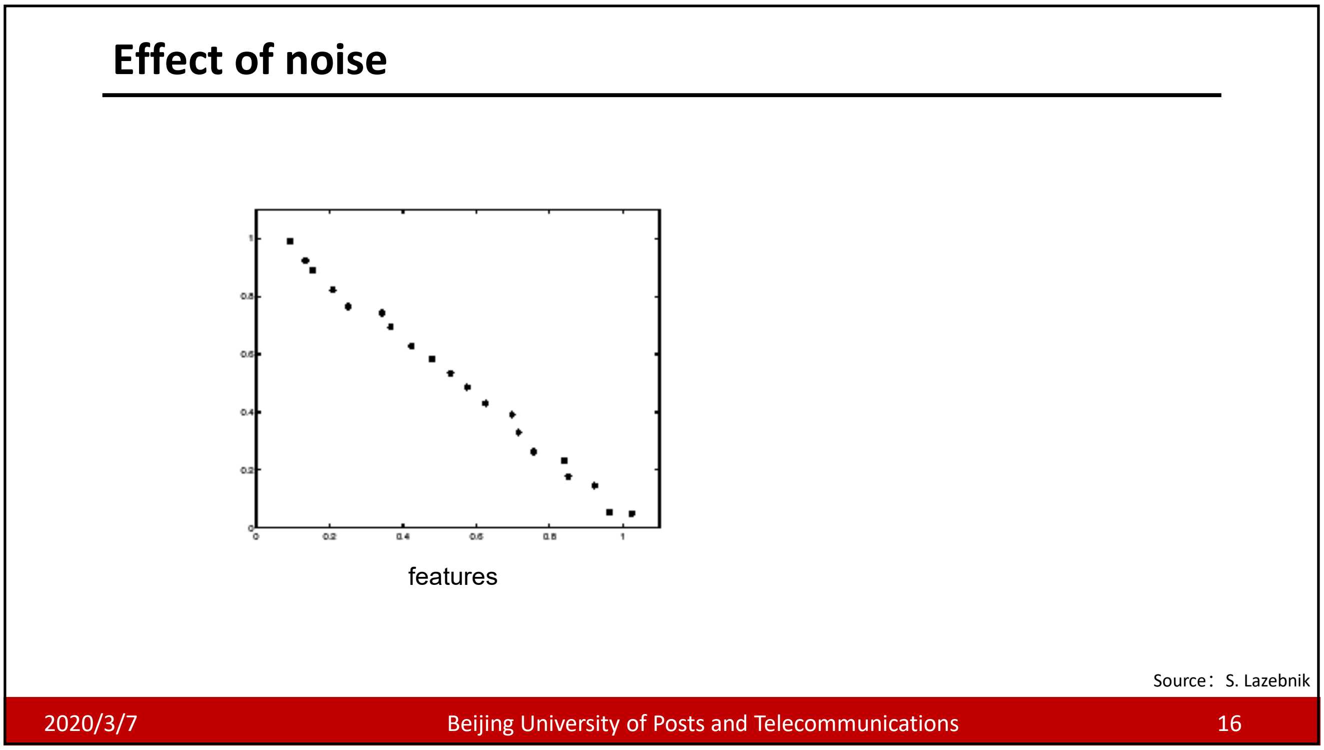
Source： S. Lazebnik

A more complicated image



http://ostatic.com/files/images/ss_hough.jpg

Source： S. Lazebnik

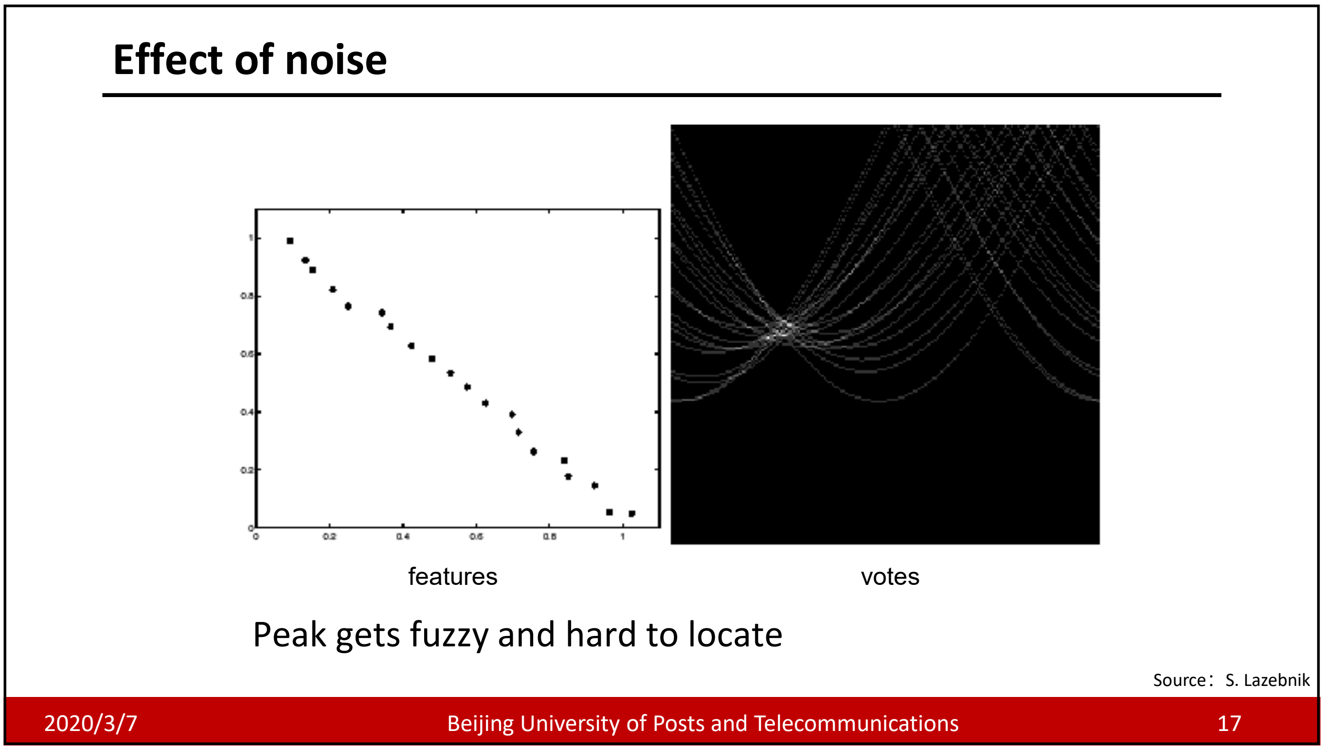


Effect of noise

features

Source： S. Lazebnik

16



Effect of noise

features

votes

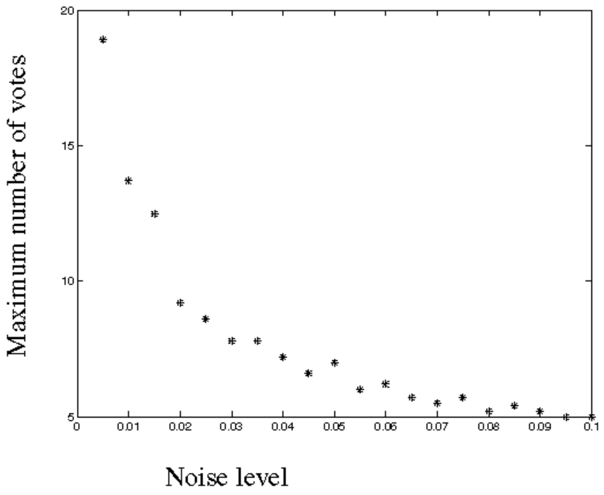
Peak gets fuzzy and hard to locate

Source： S. Lazebnik

17

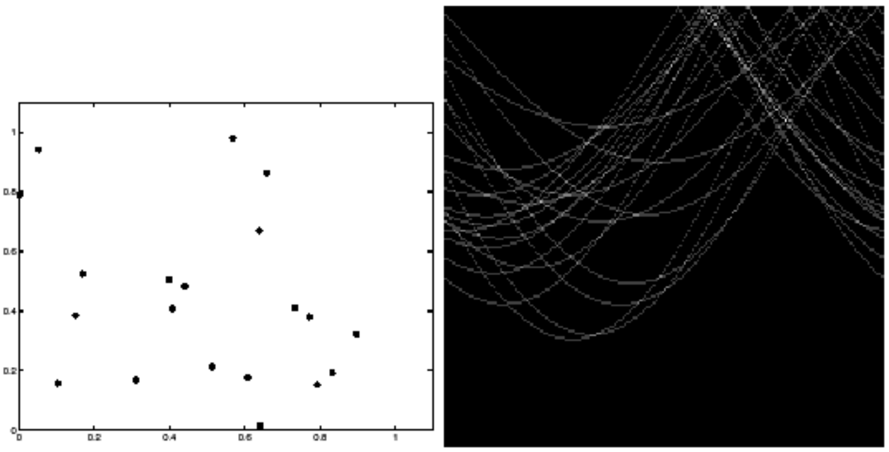
Effect of noise

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



Source： S. Lazebnik

Random points



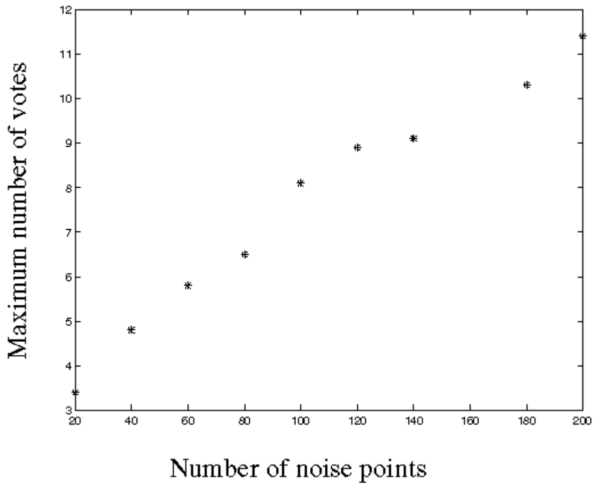
features votes

Uniform noise can lead to spurious peaks in the array

Source： S. Lazebnik

Random points

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



Source： S. Lazebnik

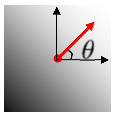
Dealing with noise

- 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

Source： S. Lazebnik

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!



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

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

- Modified Hough transform:

```
For each edge point (x,y)
        θ = gradient orientation at (x,y)
        ρ = x cos θ + y sin θ
        H(θ, ρ) = H(θ, ρ) + 1
end
```

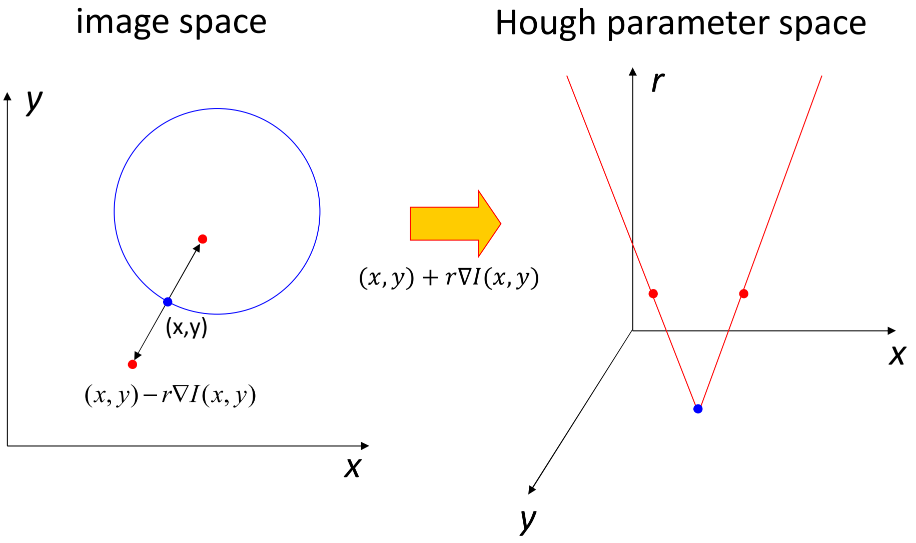
Source： S. Lazebnik

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?

Source： S. Lazebnik

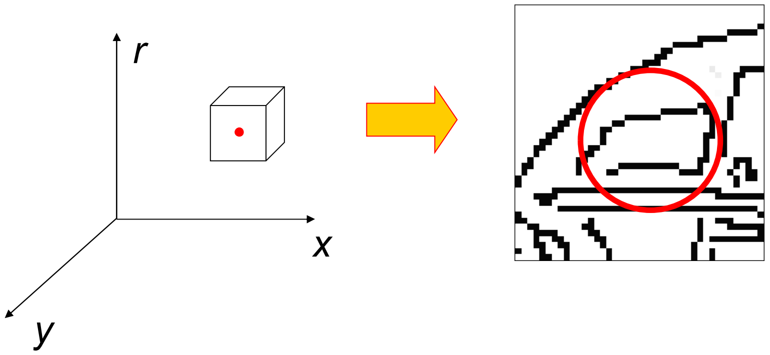
Hough transform for circles



Source： S. Lazebnik

Hough transform for circles

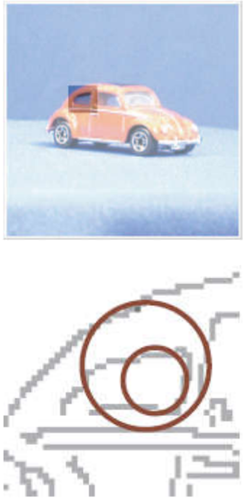
- Conceptually equivalent procedure: for each (x,y,r), draw the corresponding circle in the image and compute its “support”

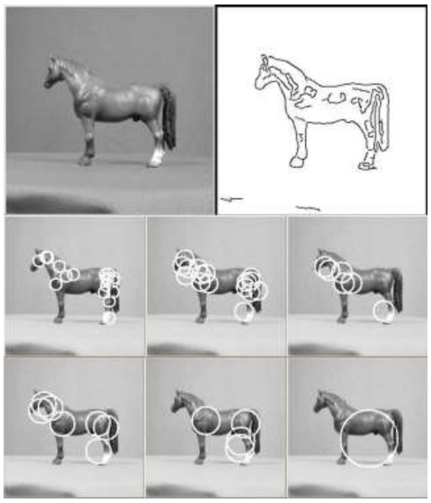


Is this more or less efficient than voting with features?

Source： S. Lazebnik

Application in recognition

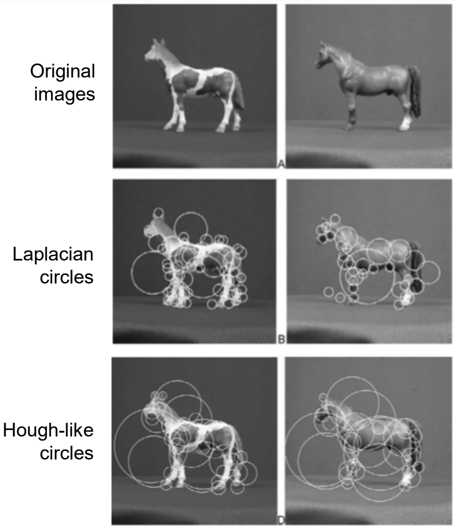




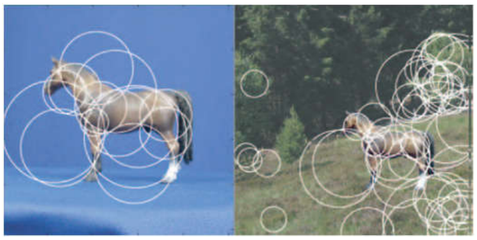
F. Jurie and C. Schmid, *Scale-invariant shape features for recognition of object categories*, CVPR 2004

Source： S. Lazebnik

Hough circles vs. Laplacian blobs

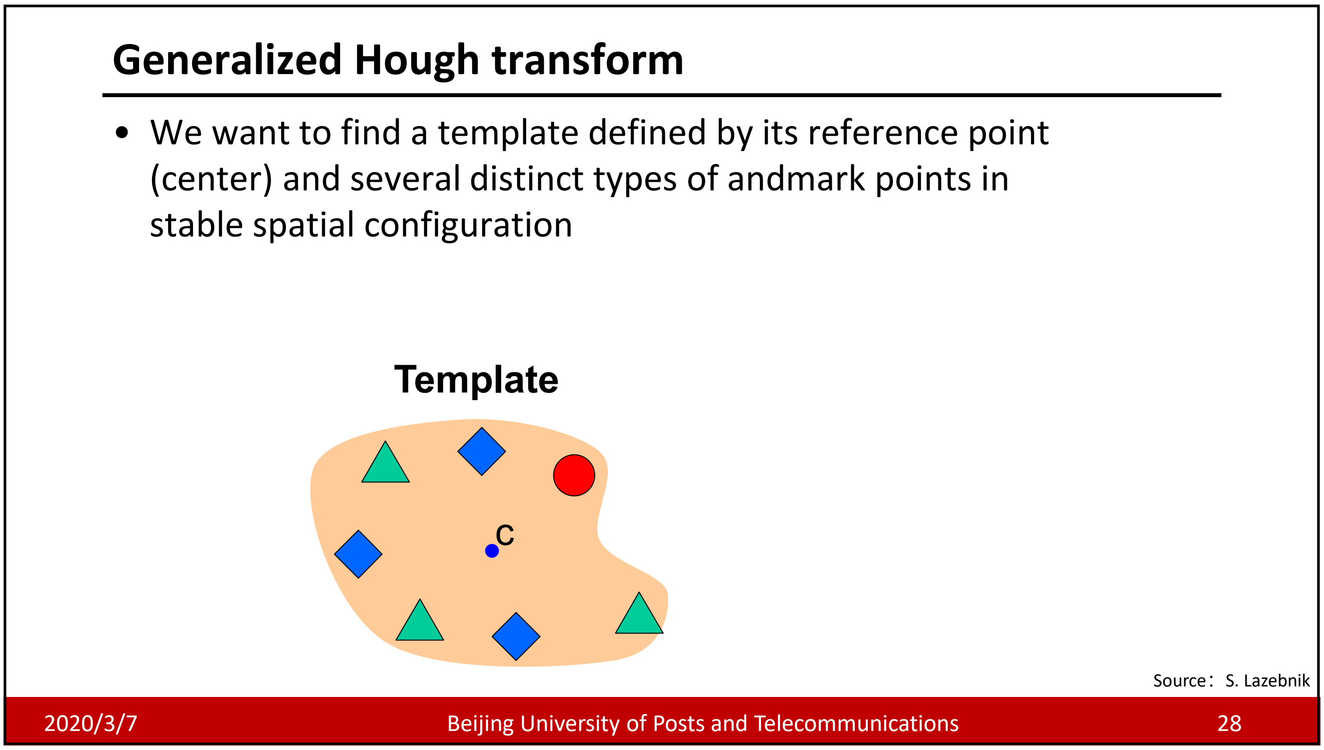


Robustness to scale and clutter



F. Jurie and C. Schmid, *Scale-invariant shape features for recognition of object categories*, CVPR 2004

Source： S. Lazebnik

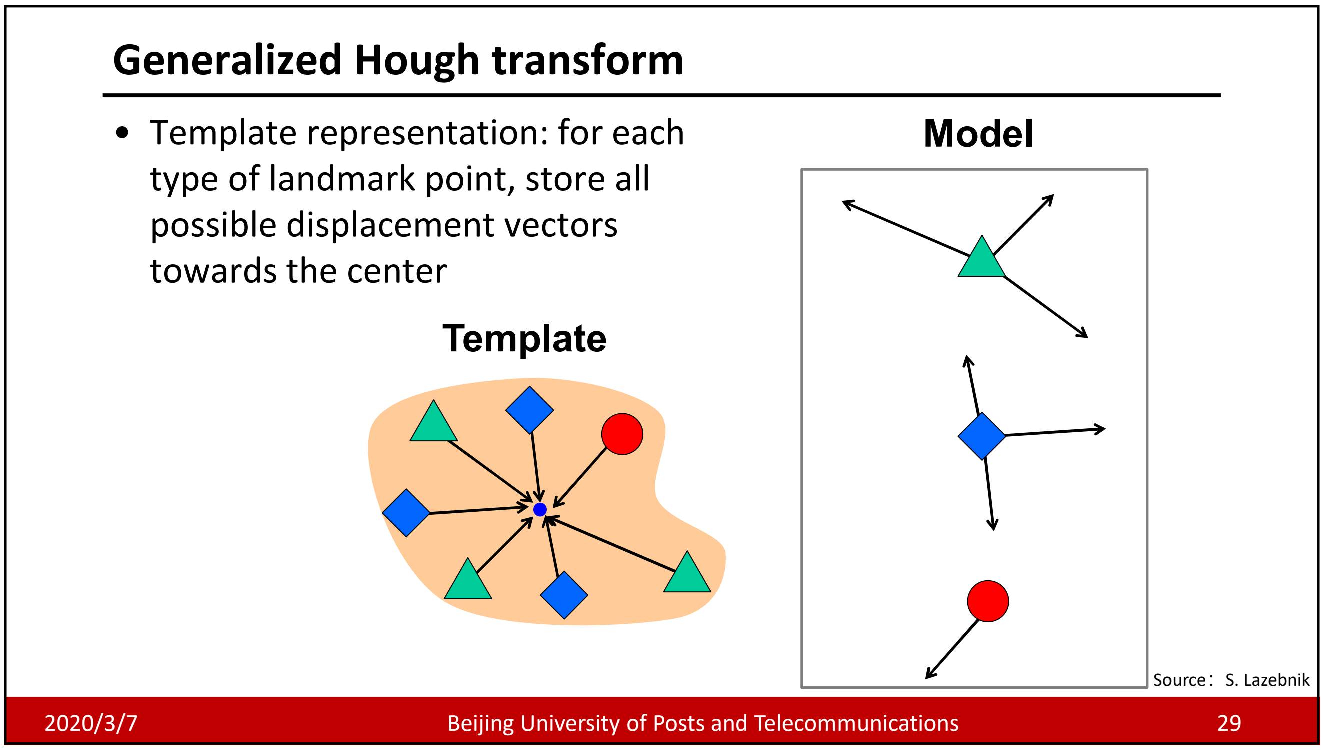


Generalized Hough transform

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

Template

Source： S. Lazebnik



Generalized Hough transform

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

Model

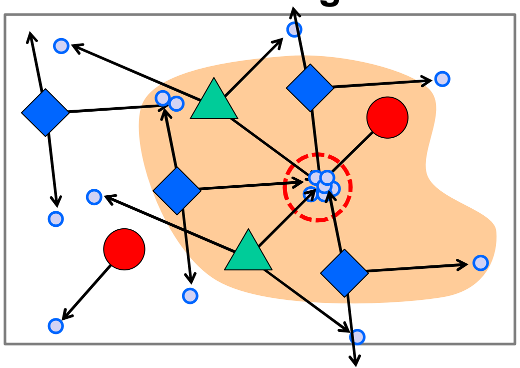
Template

Source： S. Lazebnik

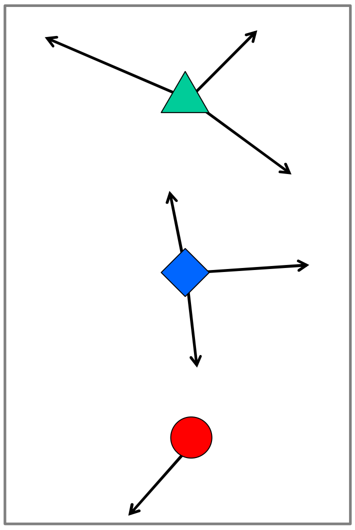
Generalized Hough transform

- 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

Test image



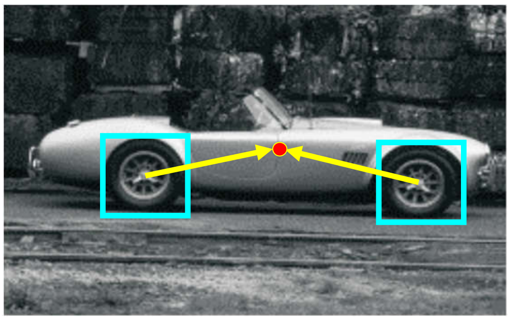
Model



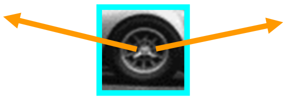
Source：S. Lazebnik

Application in recognition

- Index displacements by “visual codeword”



training image



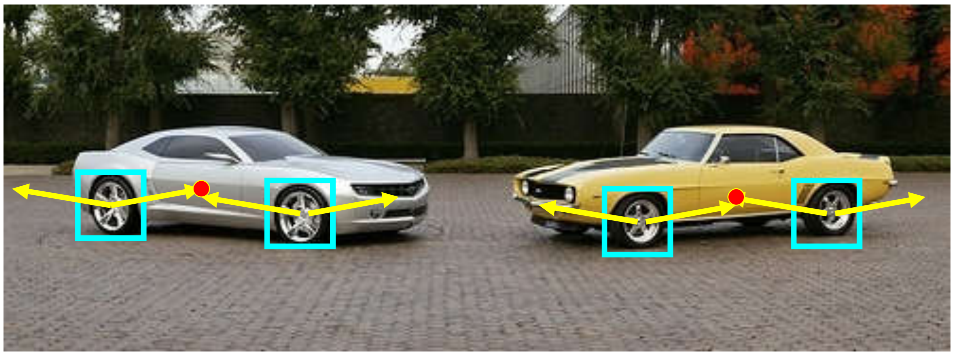
visual codeword with displacement vectors

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

Source：S. Lazebnik

Application in recognition

- Index displacements by “visual codeword”



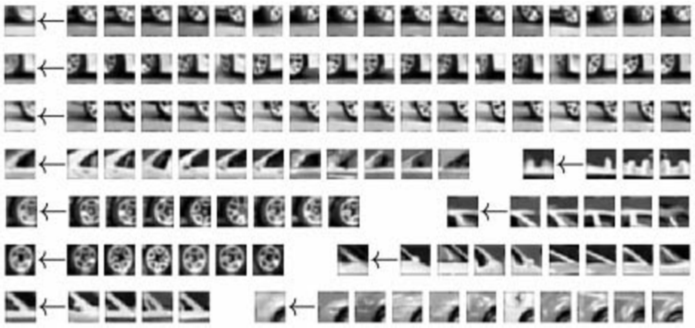
test image

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

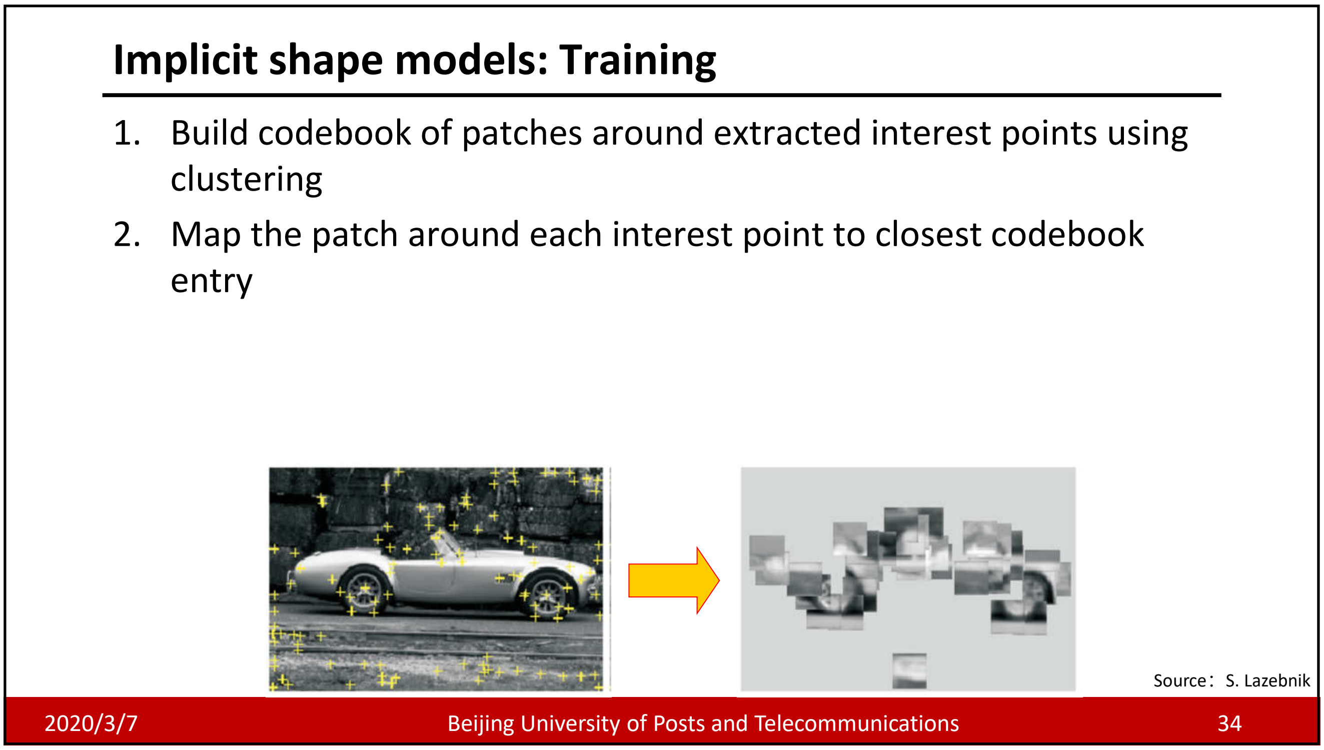
Source： S. Lazebnik

Implicit shape models: Training

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



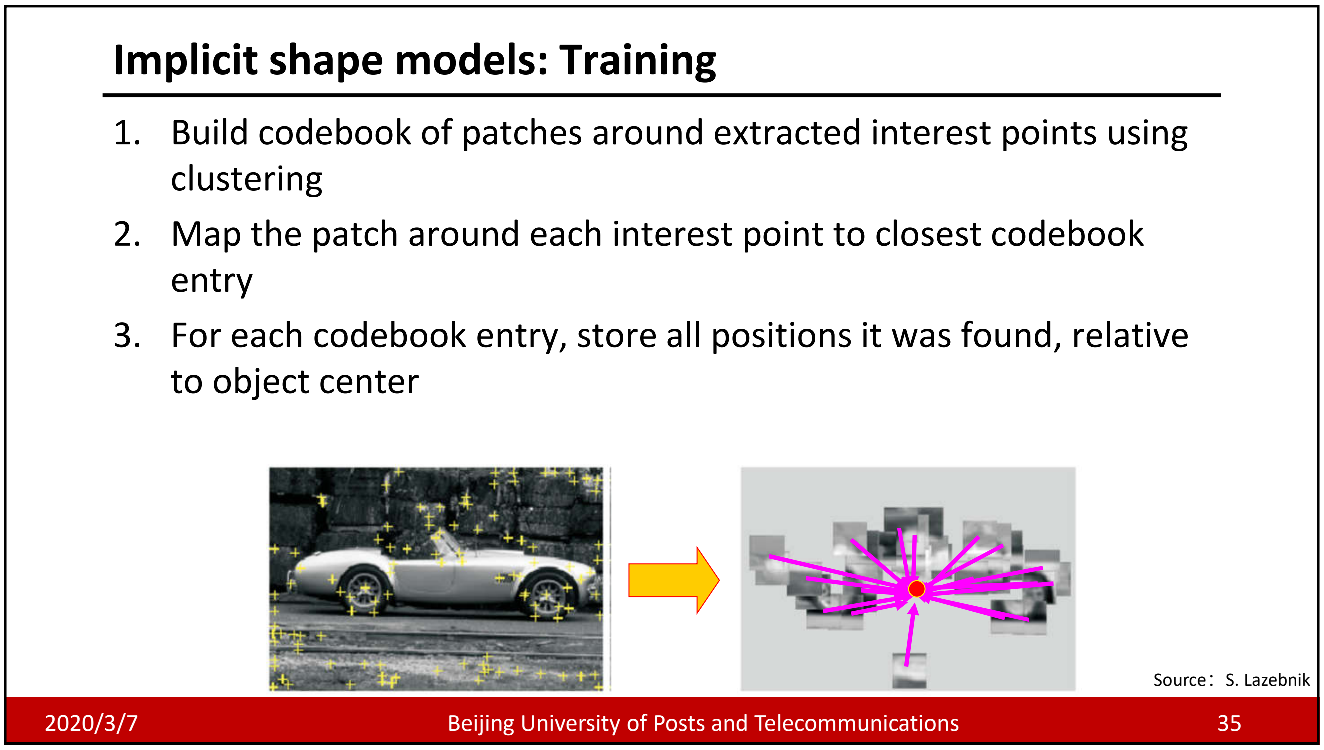
Source： S. Lazebnik



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

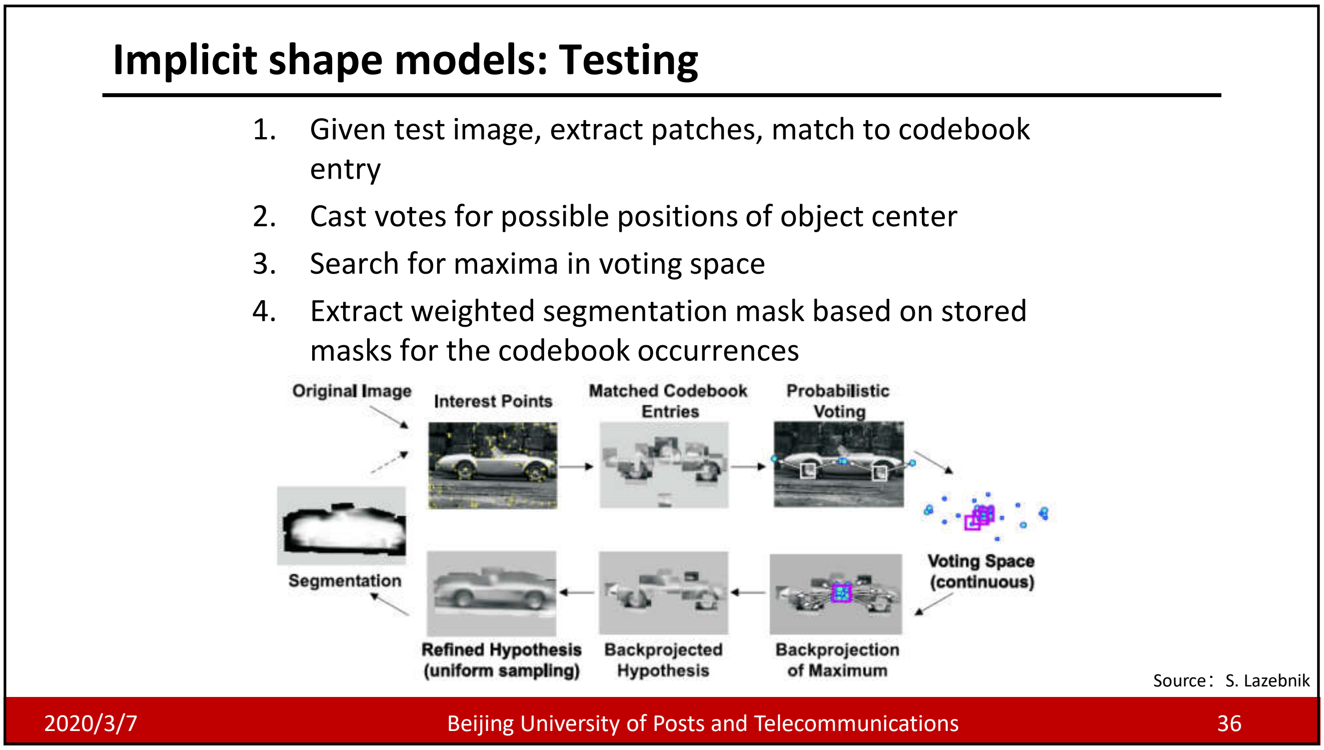
Source： S. Lazebnik



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

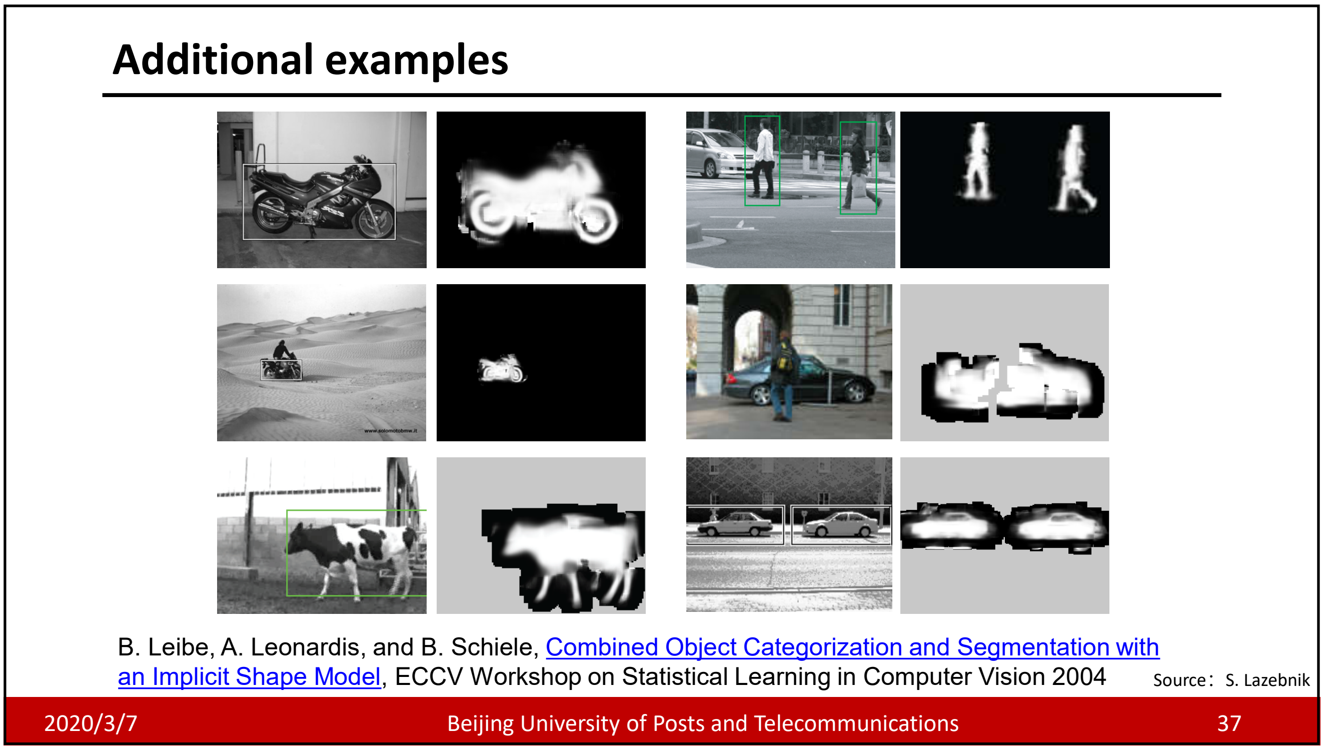
Source： S. Lazebnik



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

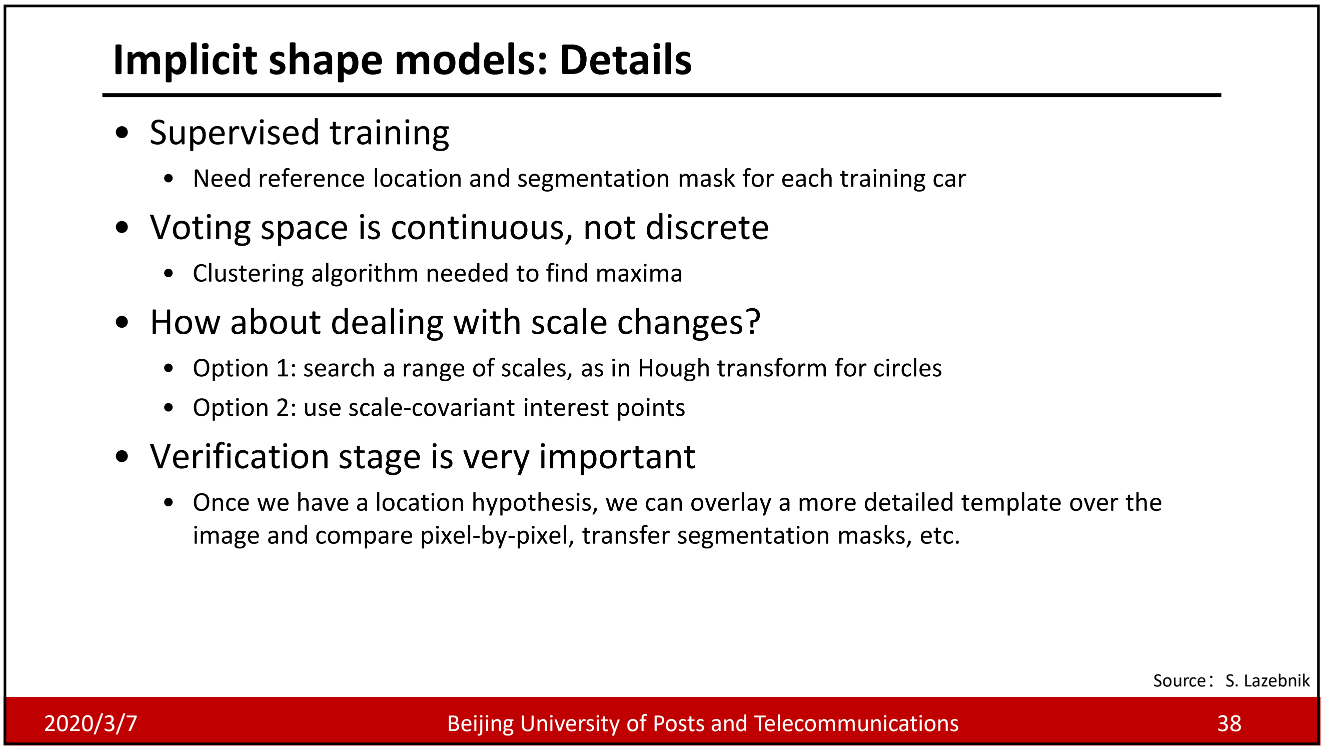
Source: S. Lazebnik



Additional examples

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

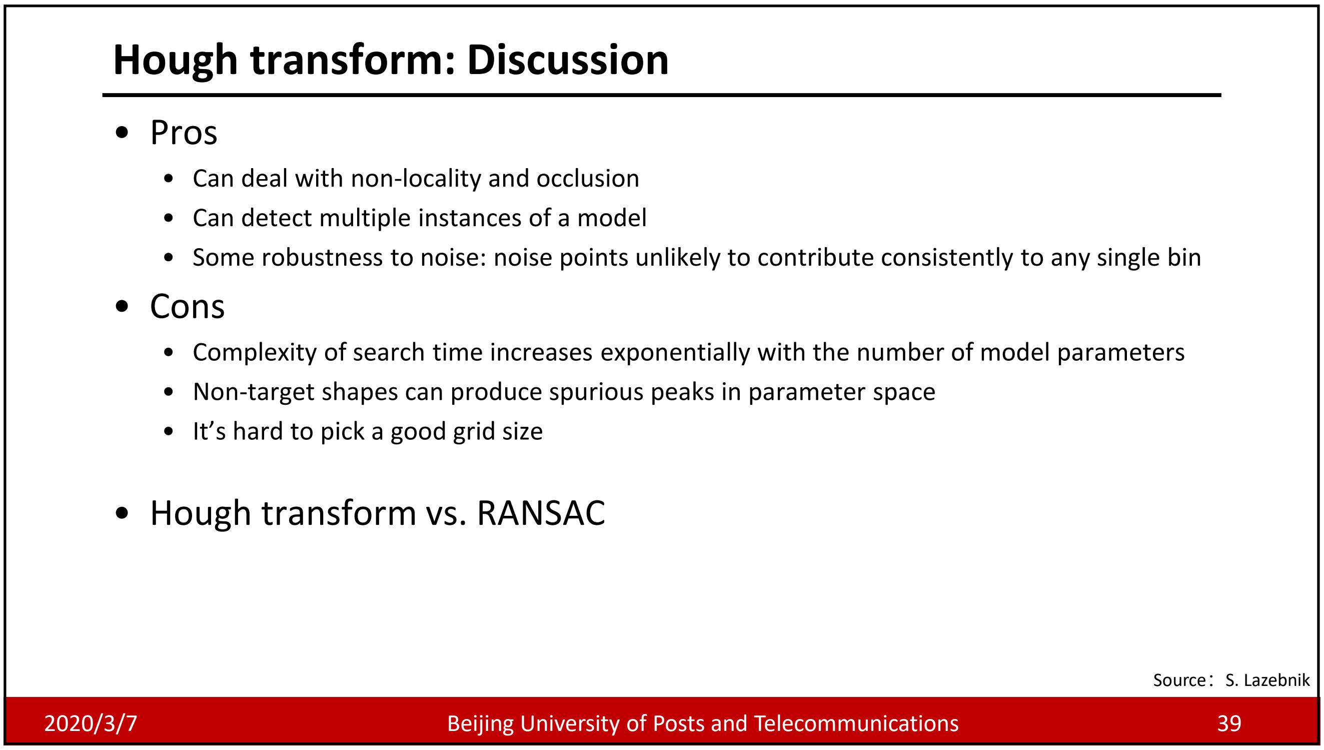
Source: S. Lazebnik



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.

Source： S. Lazebnik



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

Source： S. Lazebnik