计算机视觉
Computer Vision

周琬婷

wanting.zhou@bupt.edu.cn

计算机视觉-2024春...

扫一扫二维码，加入群聊

QQ

Fitting

(Hough transform)

RANSAC

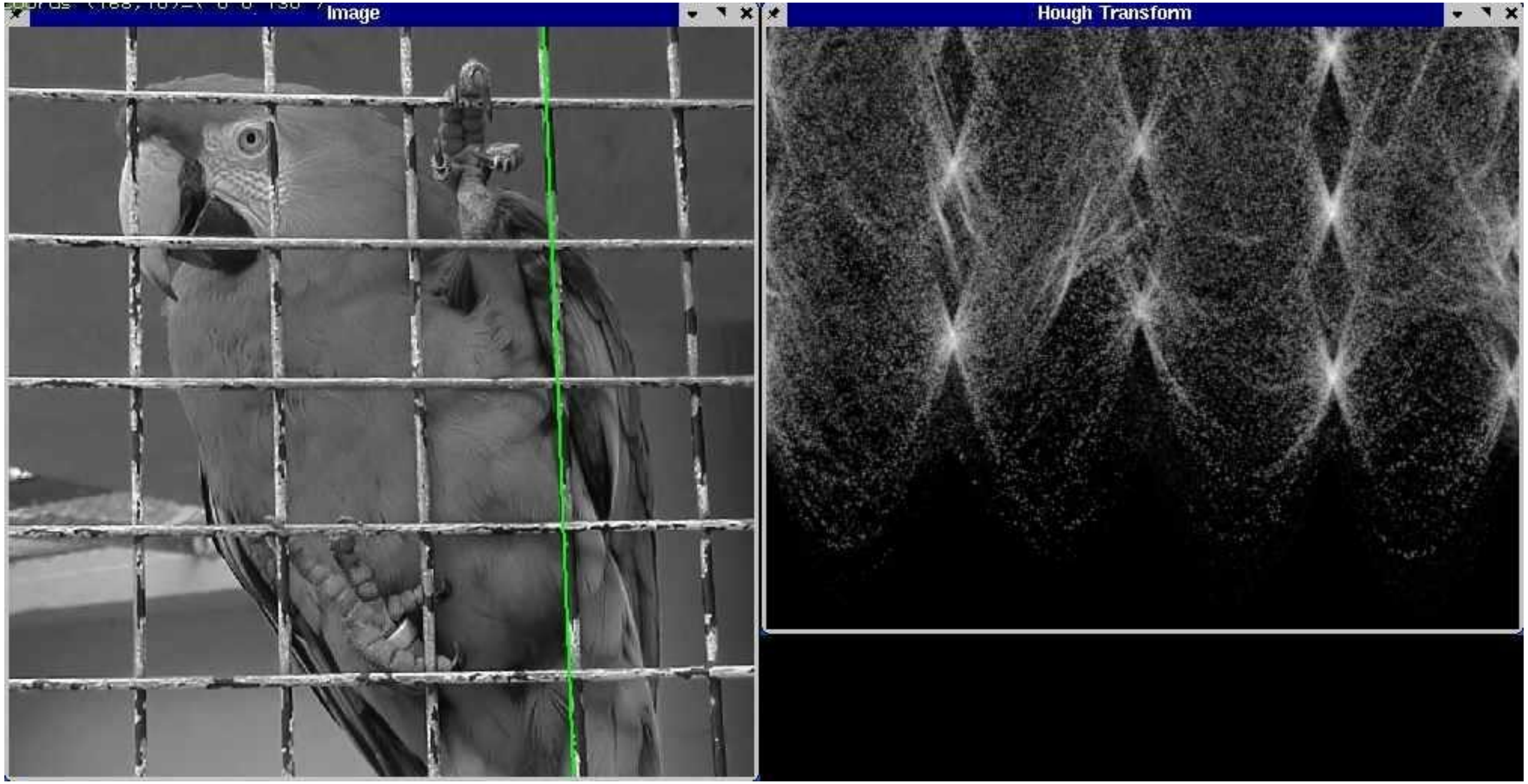
- Robust fitting can deal with a few outliers – what if we have very many?
- Random sample consensus (RANSAC):
  Very general framework for model fitting in the presence of outliers
- Outline
  - Choose a small subset of points uniformly at random
  - Fit a model to that subset
  - Find all remaining points that are "close" to the model and reject the rest as outliers
  - Do this many times and choose the best model

M. A. Fischler, R. C. Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. Comm. of the ACM, Vol 24, pp 381-395, 1981.

Source： S. Lazebnik

Machine Vision Technology							
Semantic information				Metric 3D information			
Pixels	Segments	Images	Videos	Camera		Multi-view Geometry	
Convolutions Edges & Fitting Local features Texture	Segmentation Clustering	Recognition Detection	Motion Tracking	Camera Model	Camera Calibration	Epipolar Geometry	SFM
10	4	4	2	2	2	2	2

Fitting: The Hough transform



Source： S. Lazebnik

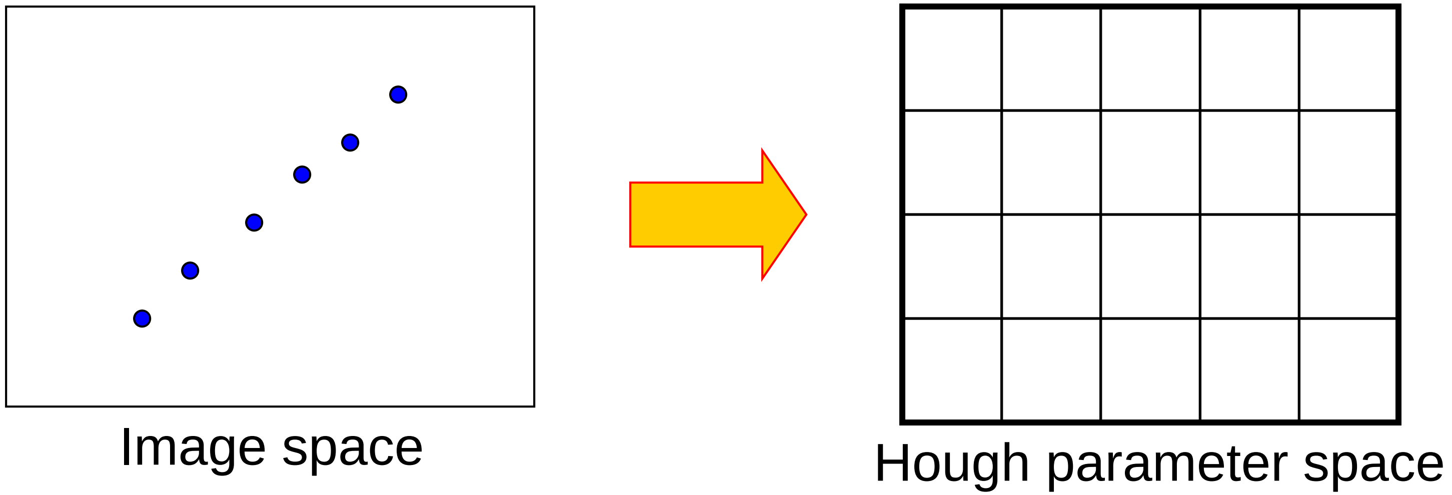
Voting schemes

- 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

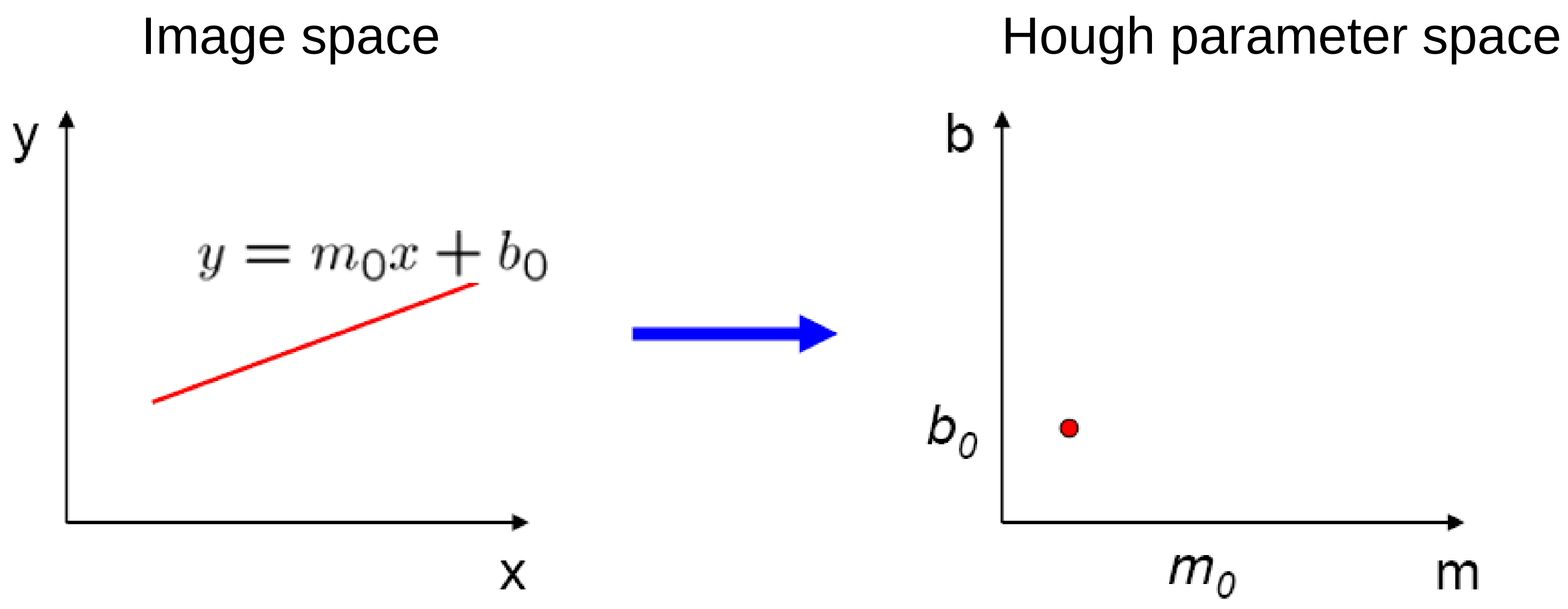


P.V.C. Hough, *Machine Analysis of Bubble Chamber Pictures,* Proc. Int. Conf. High Energy Accelerators and Instrumentation, 1959

Source：S. Lazebnik

Parameter space representation

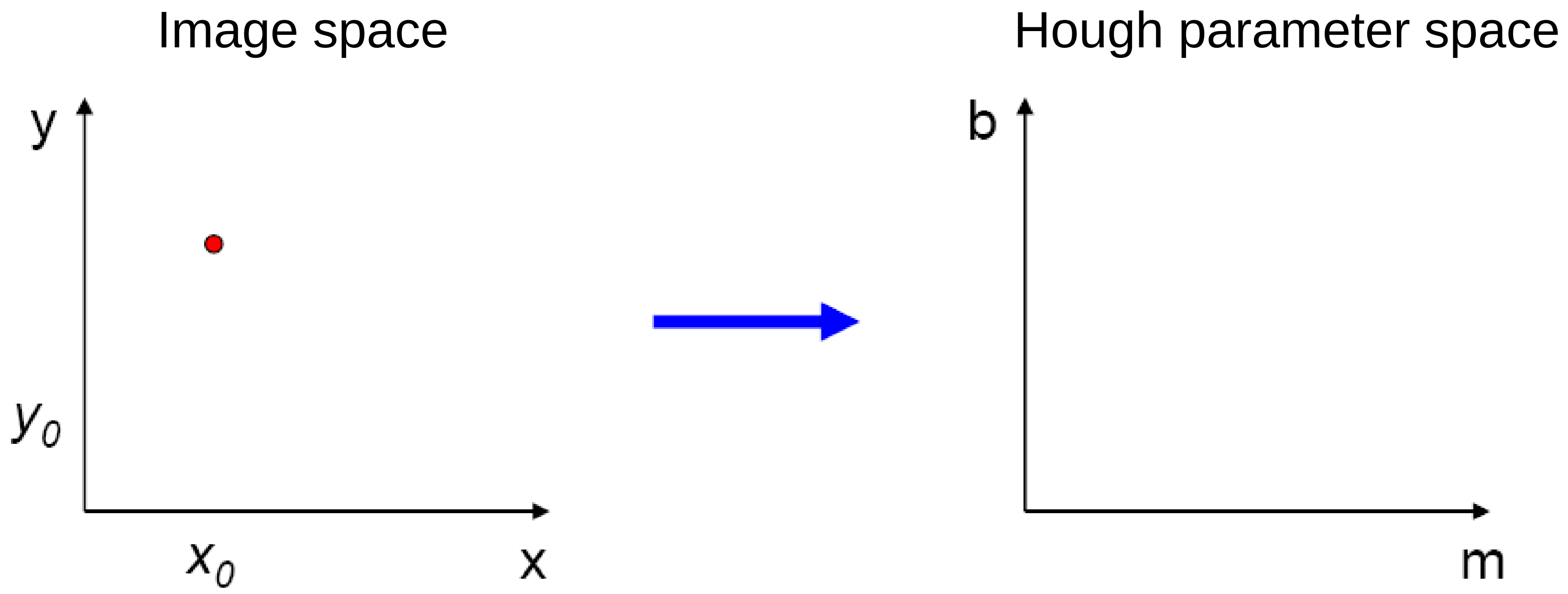
- A line in the image corresponds to a point in Hough space



Source: S. Seitz

Parameter space representation

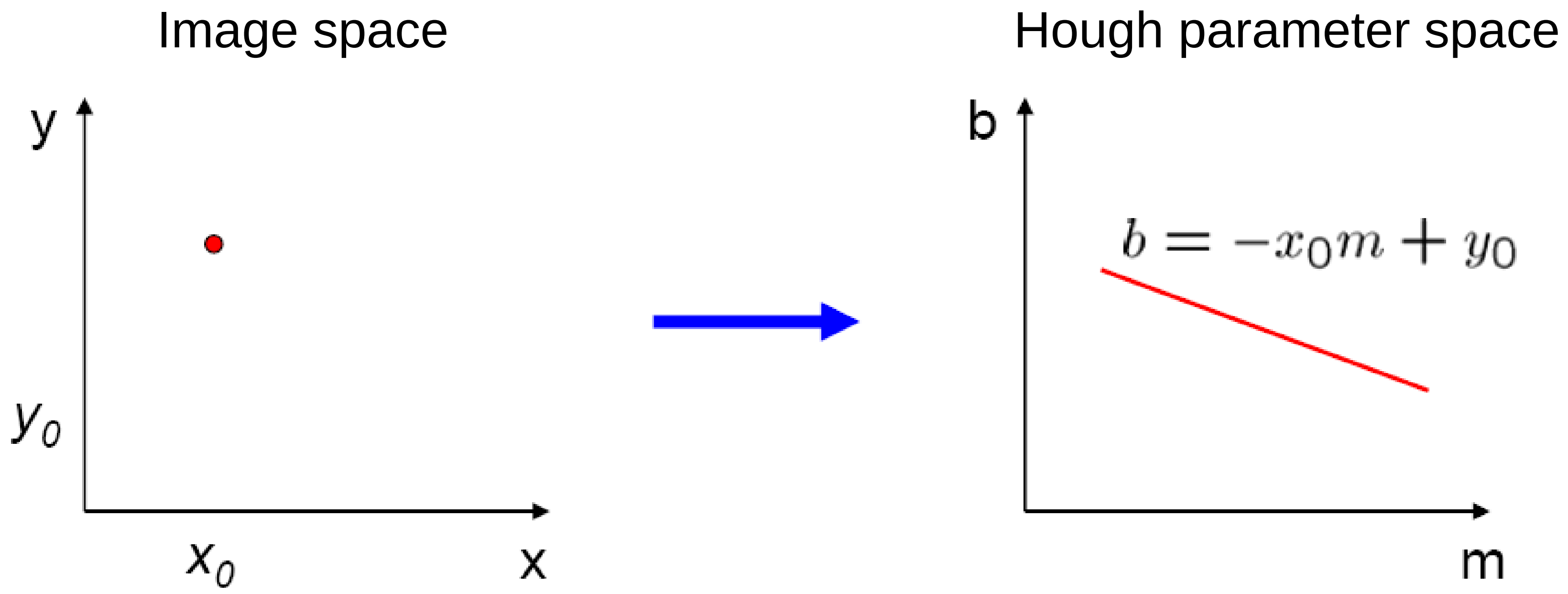
- What does a point (x_0, y_0) in the image space map to in the Hough space?



Source： S. Lazebnik

Parameter space representation

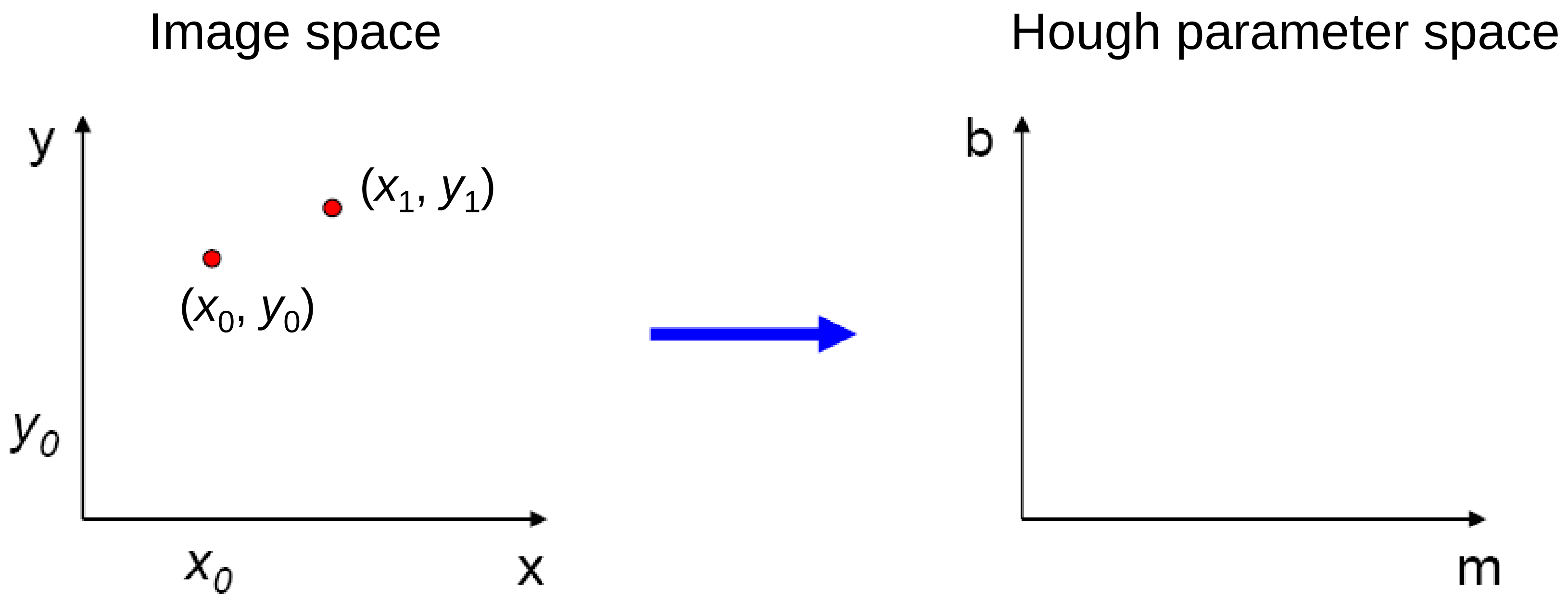
- What does a point (x_0, y_0) in the image space map to in the Hough space?
  - Answer: the solutions of $b = -x_0 m + y_0$
  - This is a line in Hough space



Source: S. Lazebnik

Parameter space representation

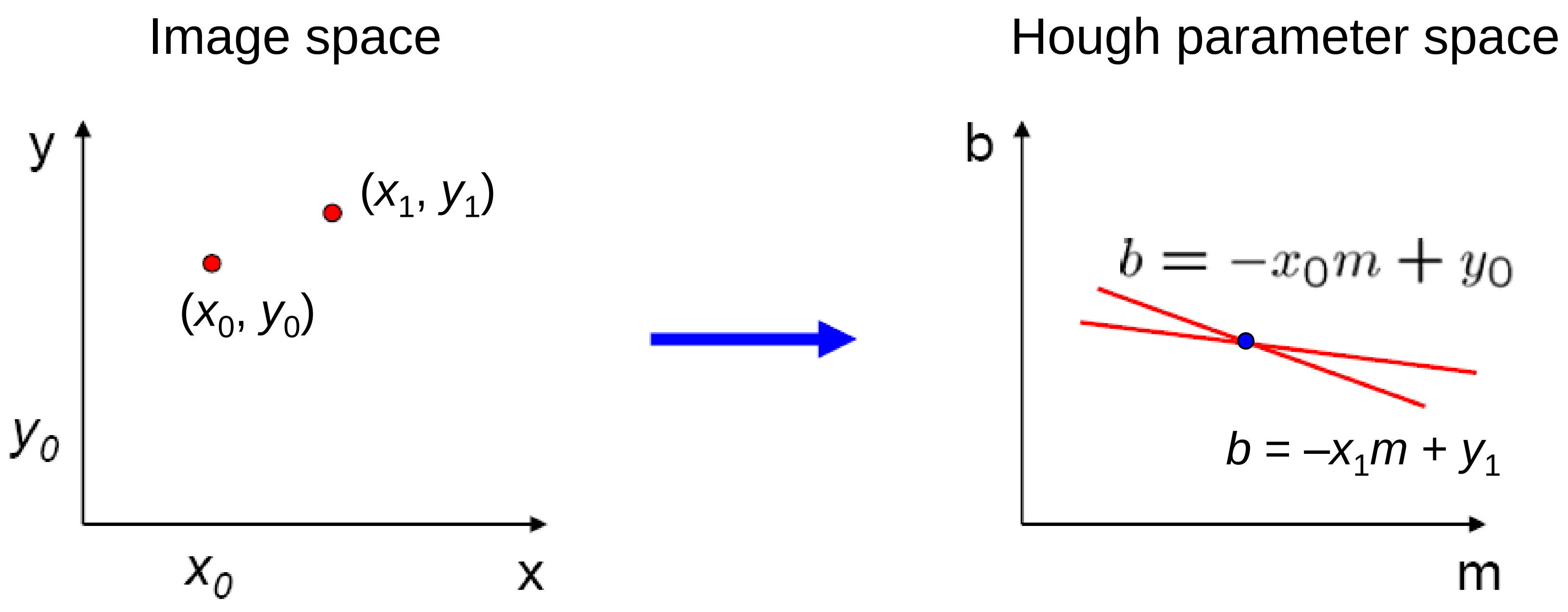
- Where is the line that contains both (x_0, y_0) and (x_1, y_1)?



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

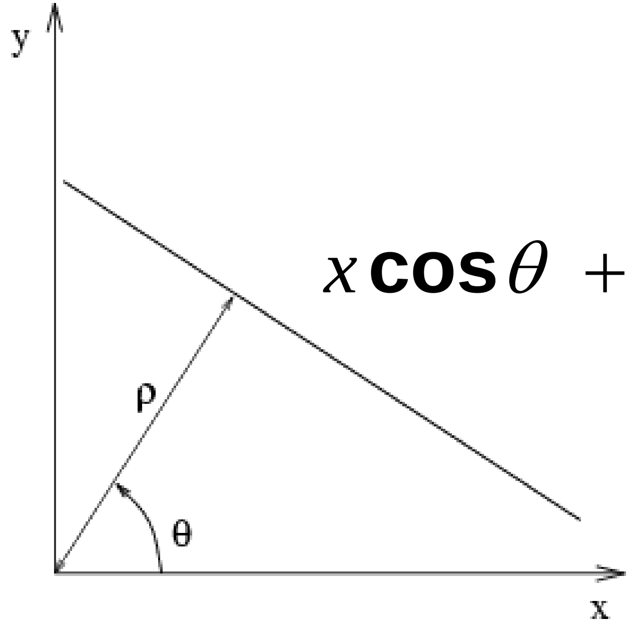
Parameter space representation

- 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

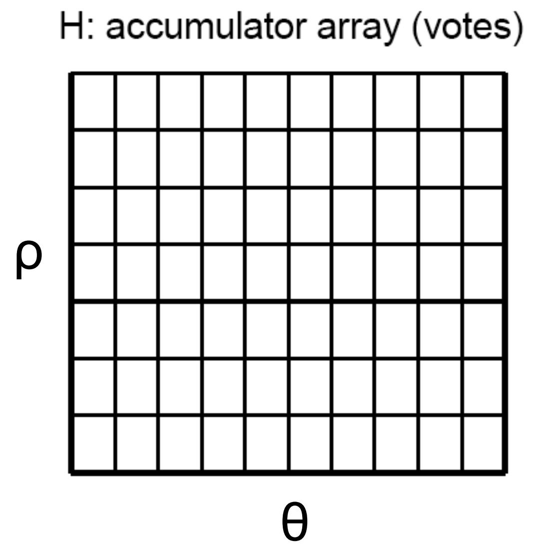
  For θ = 0 to 180

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

  $H(\theta, \rho) = H(\theta, \rho) + 1$

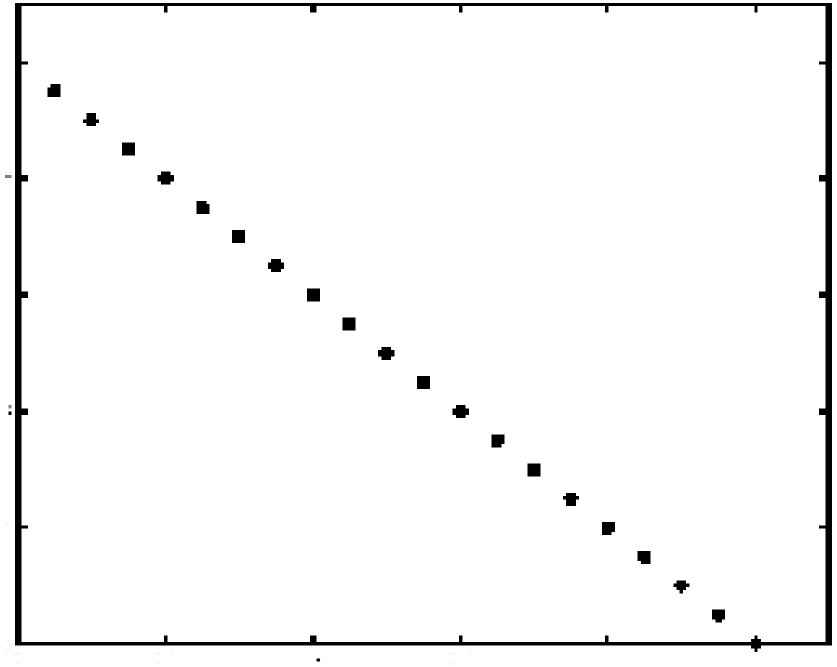
  end

  end
- Find the value(s) of (θ, ρ) where $H(\theta, \rho)$ is a local maximum
  - The detected line in the image is given by $\rho = x \cos \theta + y \sin \theta$

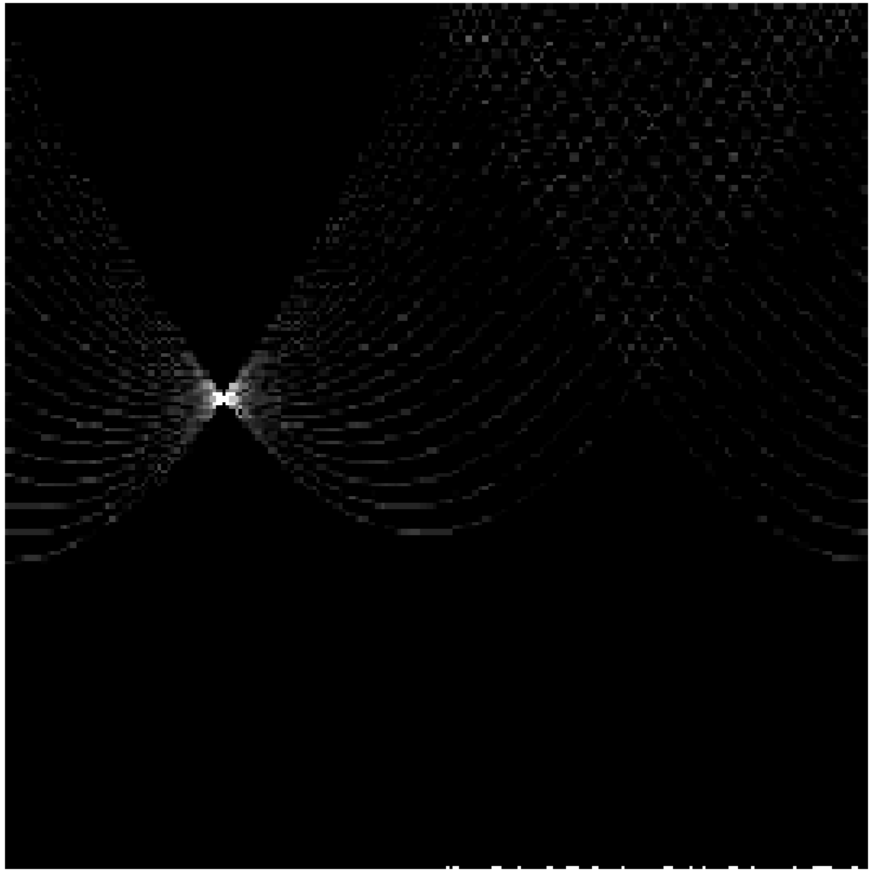


Source： S. Lazebnik

Basic illustration



features



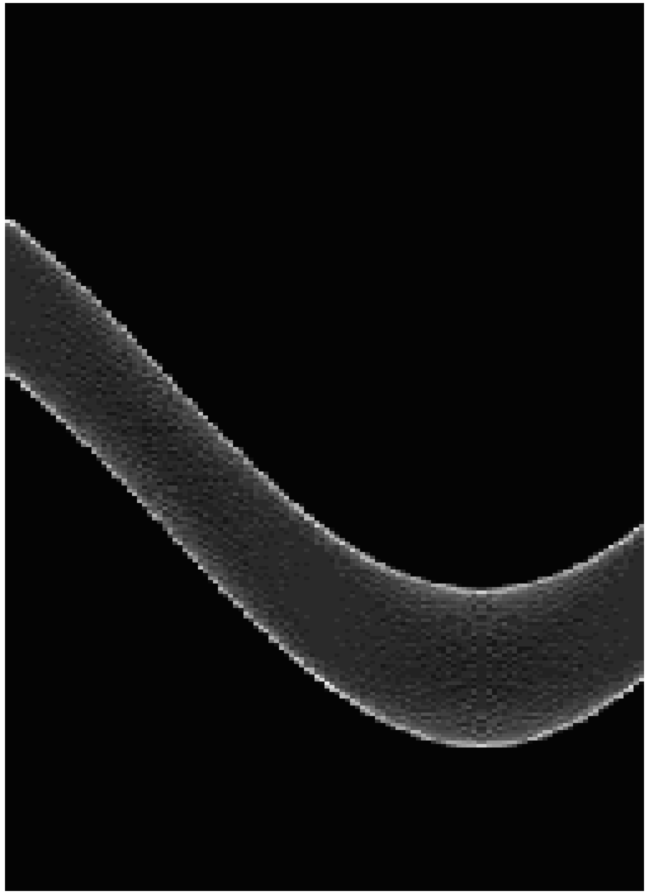
votes

Source： S. Lazebnik

Other shapes

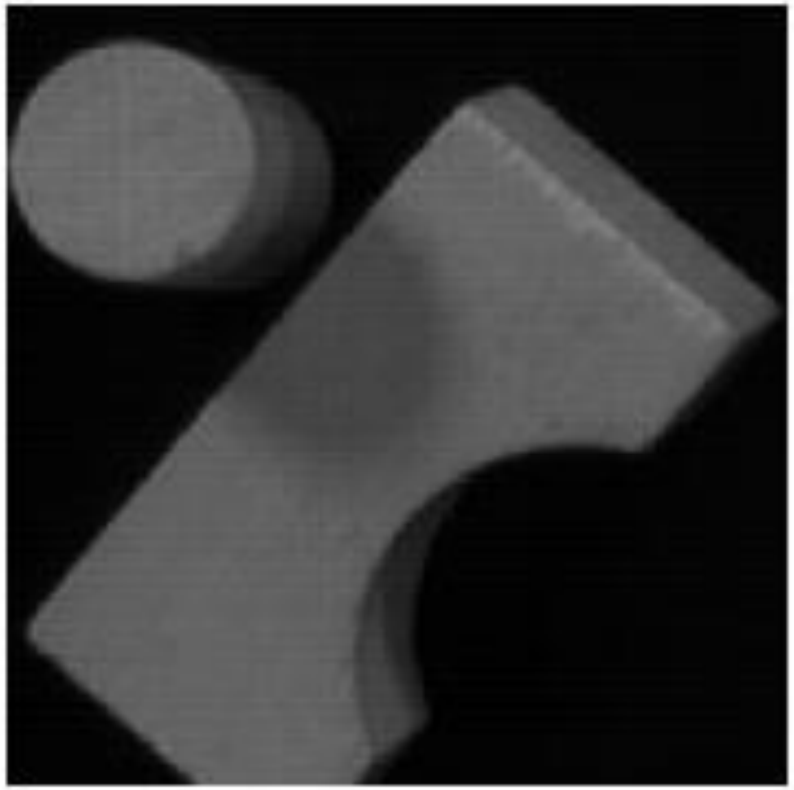
Square

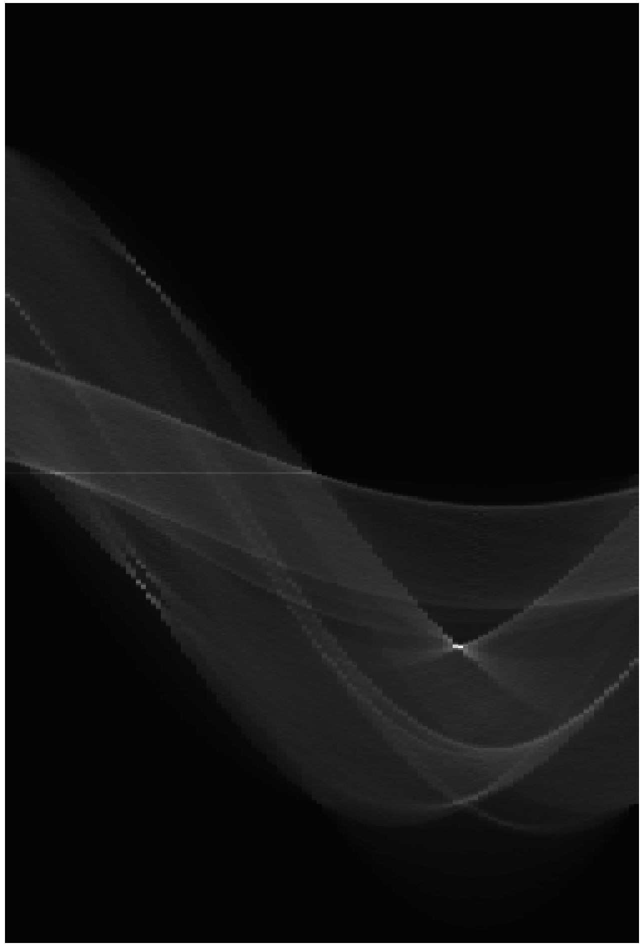
Circle



Source： S. Lazebnik

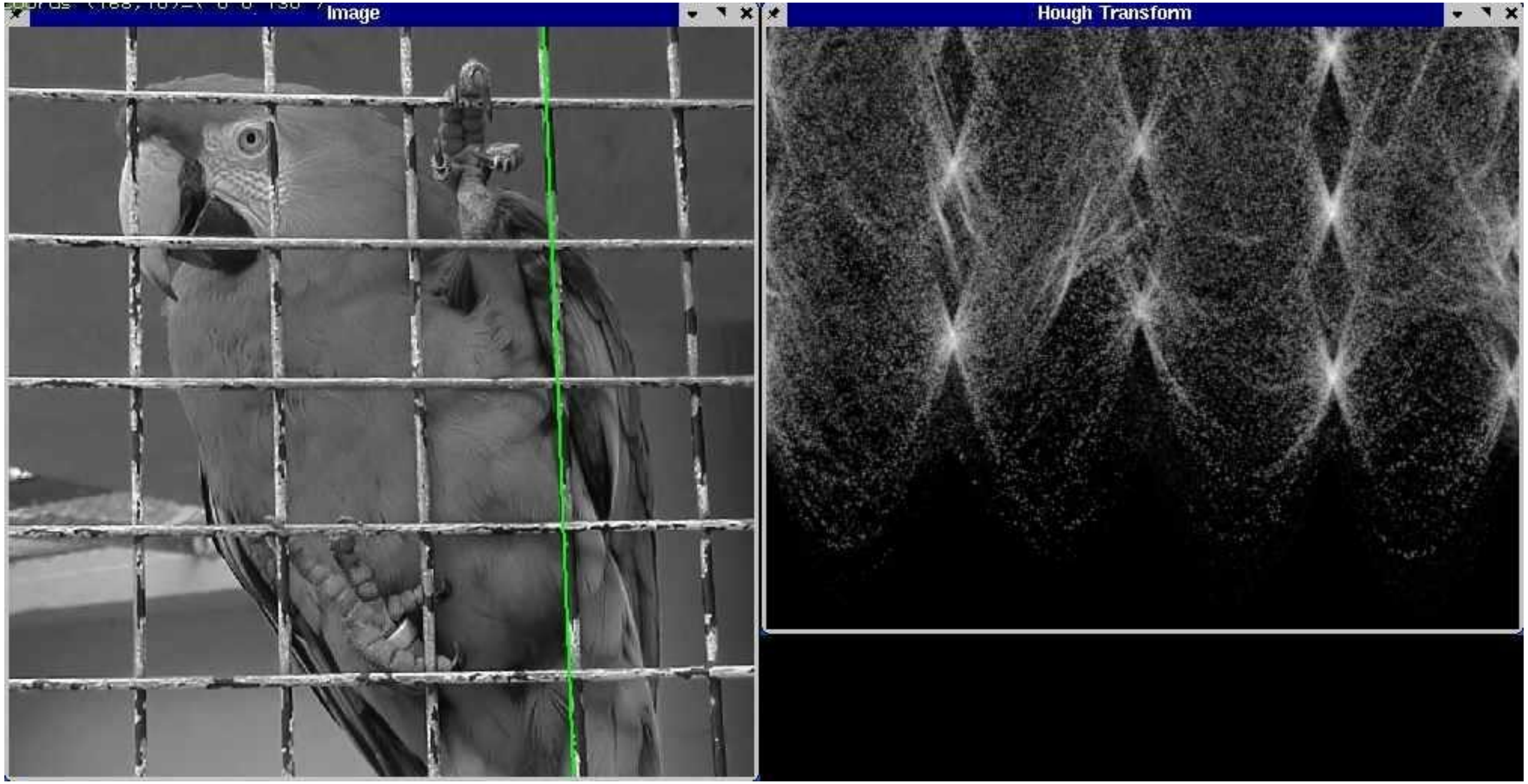
Several lines





Source： S. Lazebnik

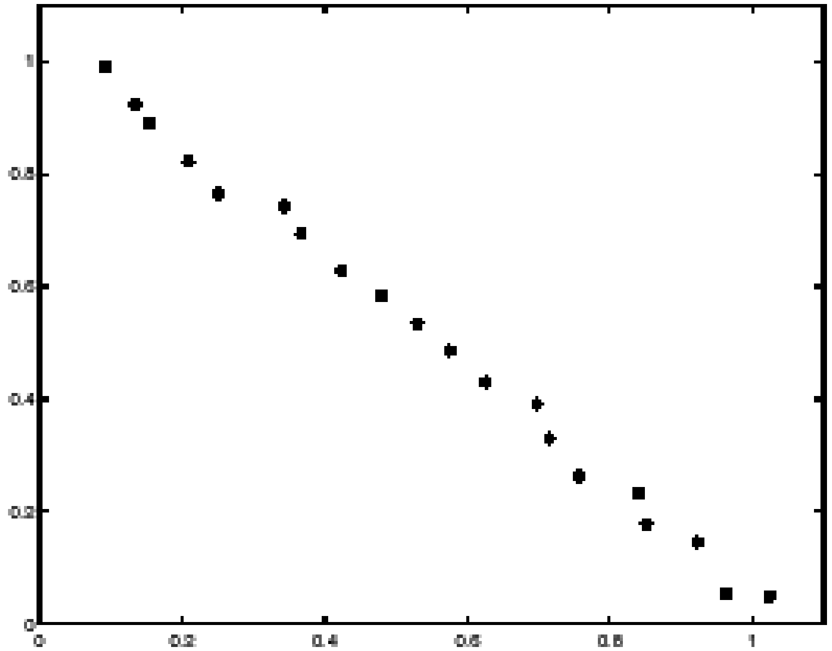
A more complicated image



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

Source： S. Lazebnik

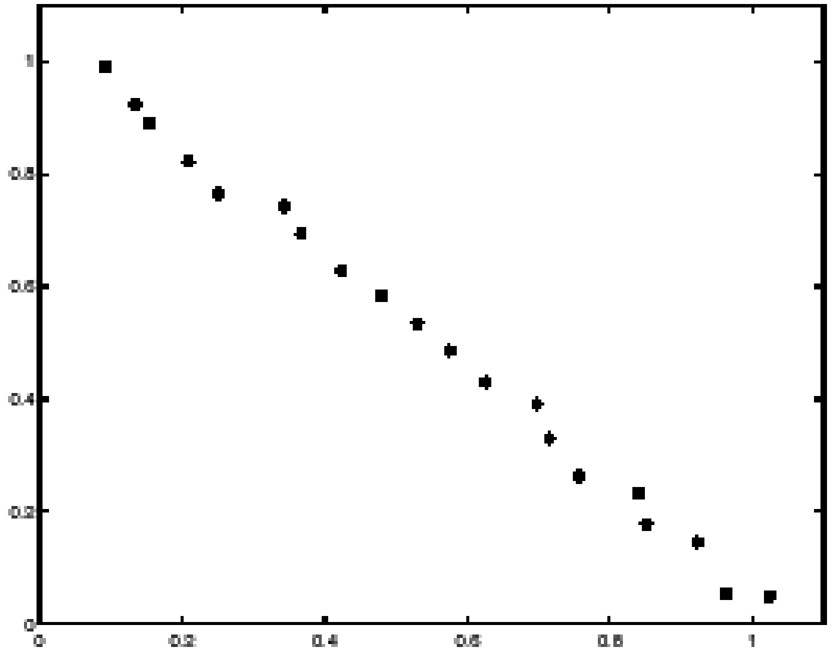
Effect of noise



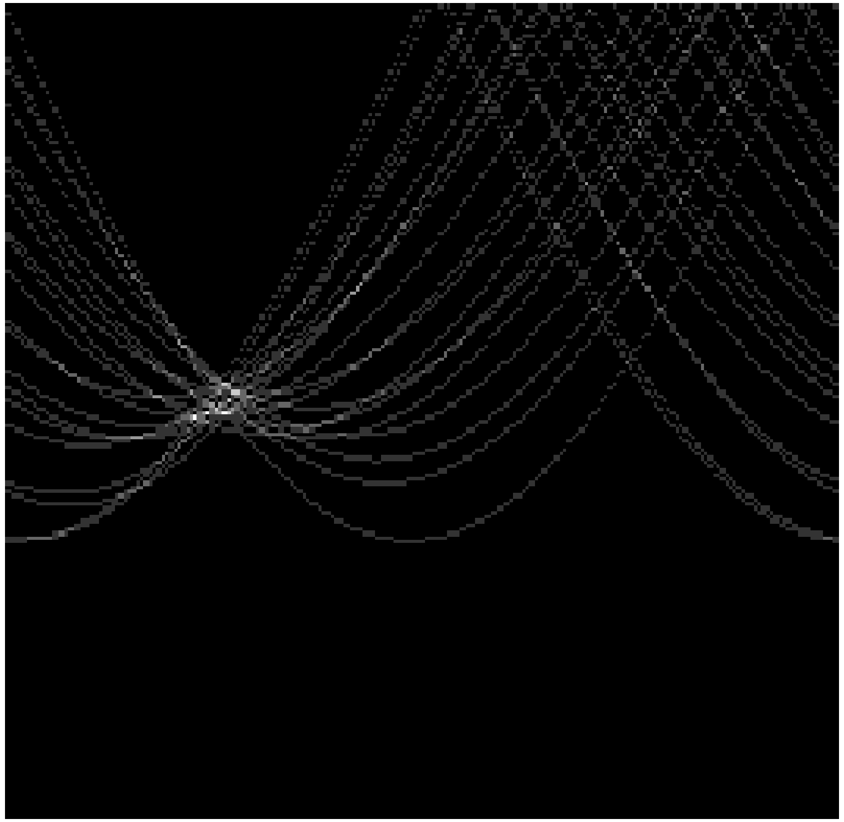
features

Source： S. Lazebnik

Effect of noise



features



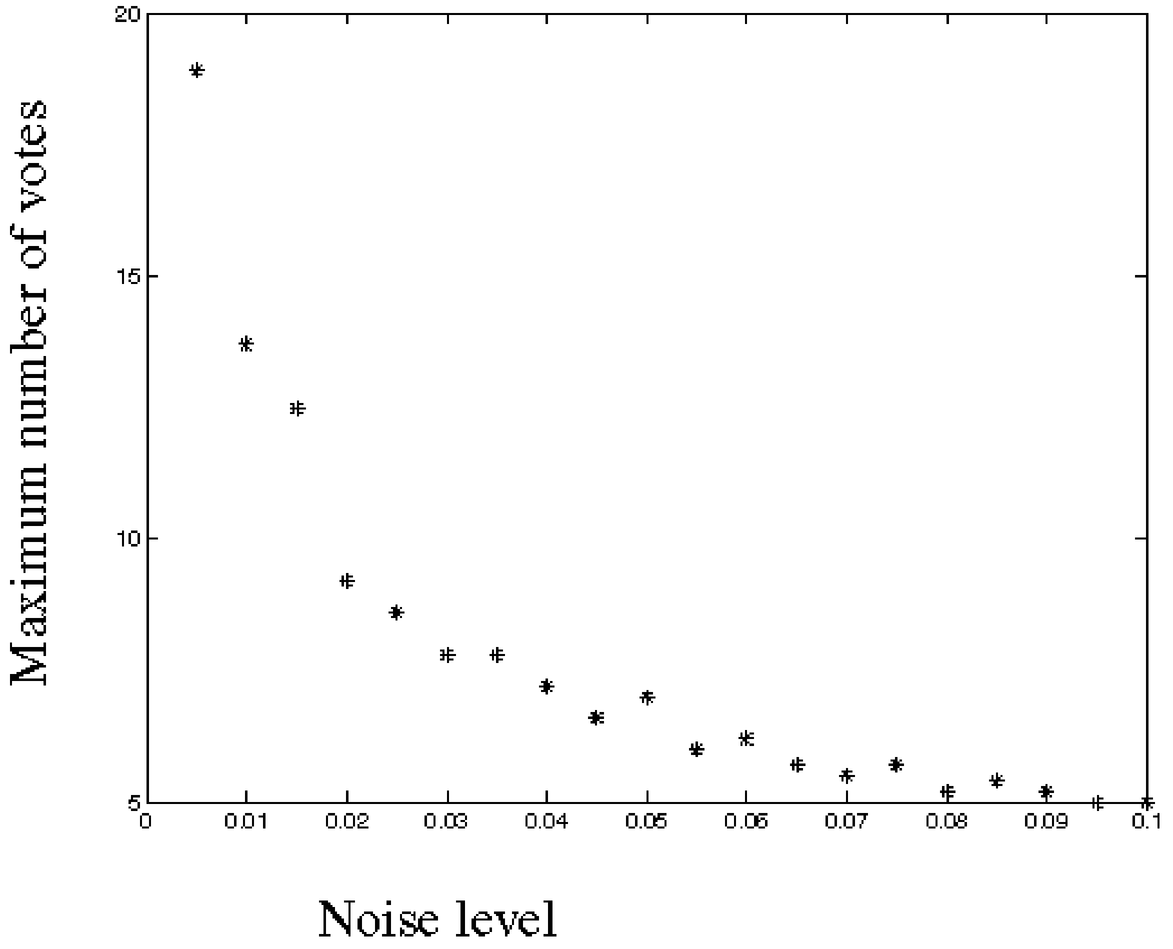
votes

Peak gets fuzzy and hard to locate

Source： S. Lazebnik

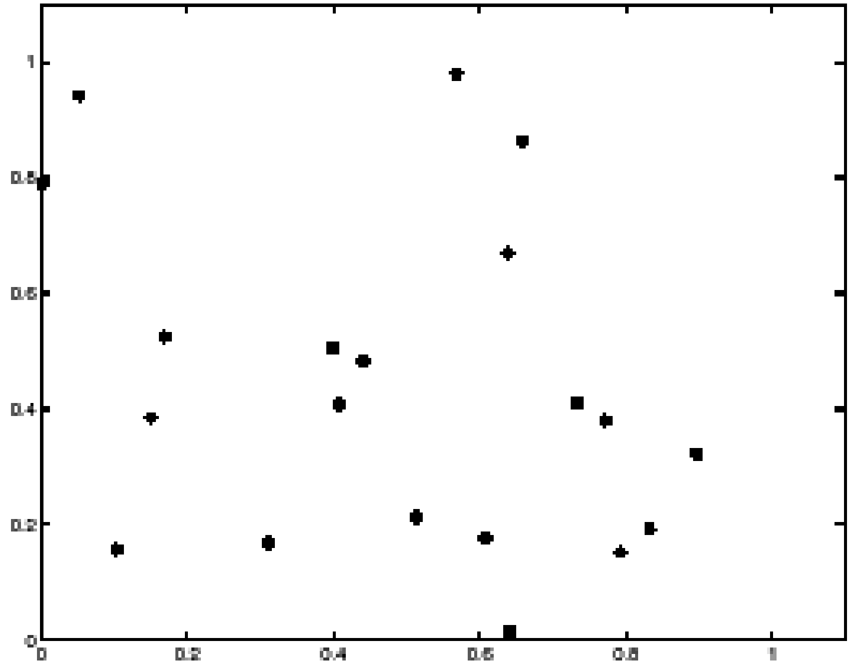
Effect of noise

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

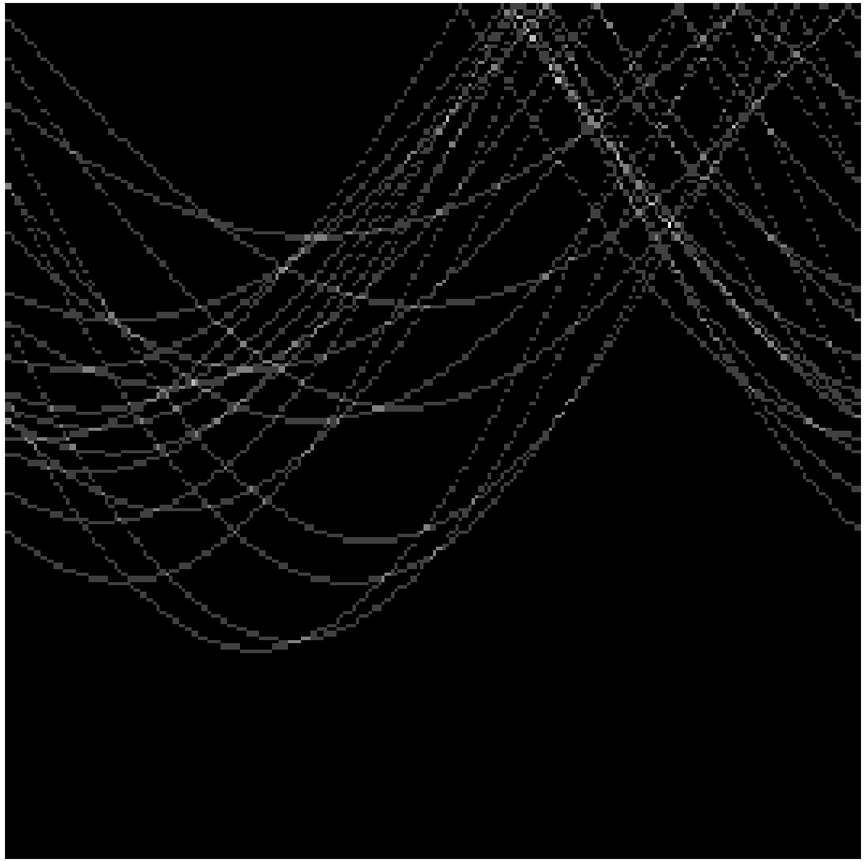


Source： S. Lazebnik

Random points



features



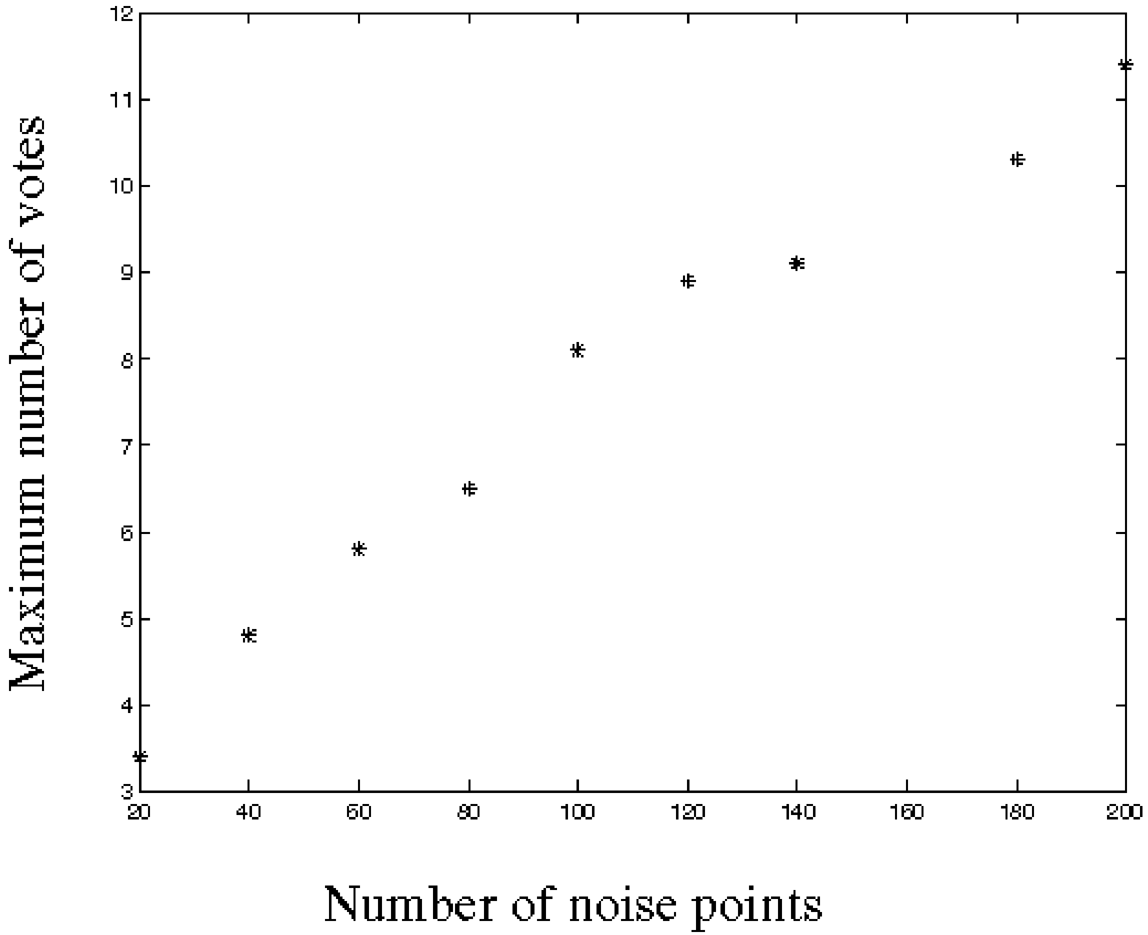
votes

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

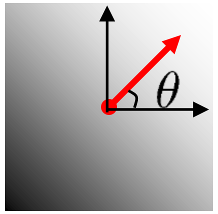
- Choose a good grid / discretization
  - Too coarse: large votes obtained when too many different lines correspond to a single bucket
  - Too fine: miss lines because some points that are not exactly collinear cast votes for different buckets
- Increment neighboring bins (smoothing in accumulator array)
- 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!

- Modified Hough transform:



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

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

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

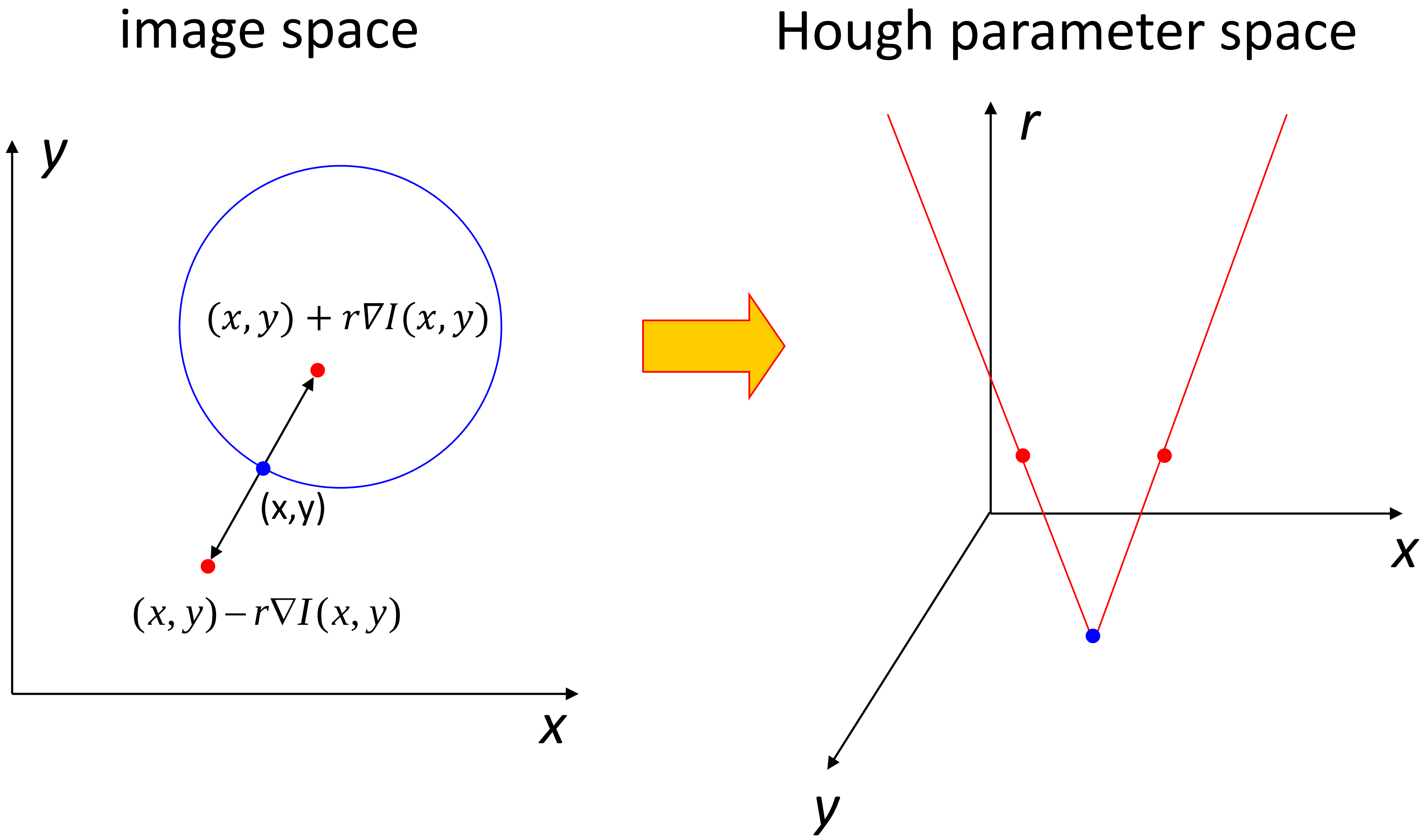
Source： S. Lazebnik

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

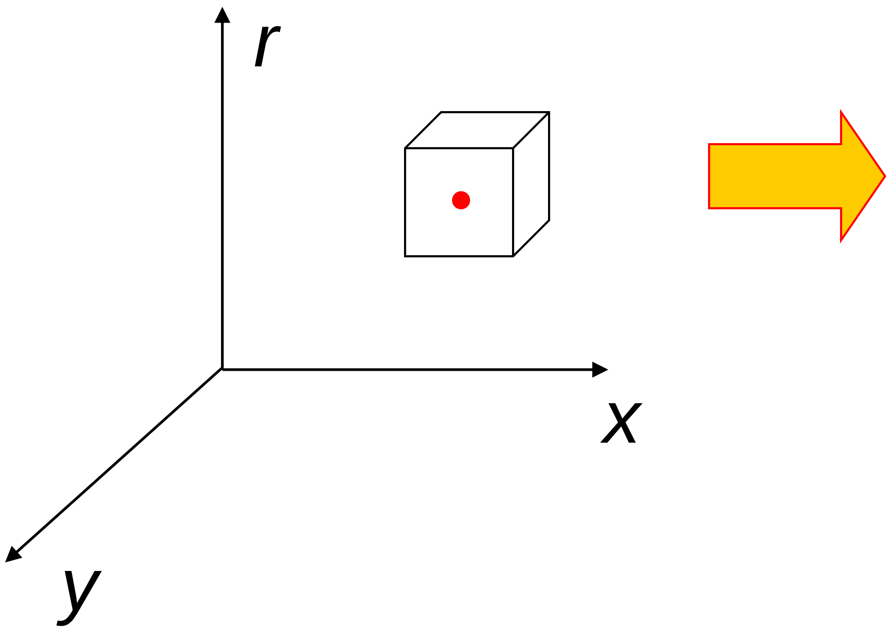
Hough transform for circles

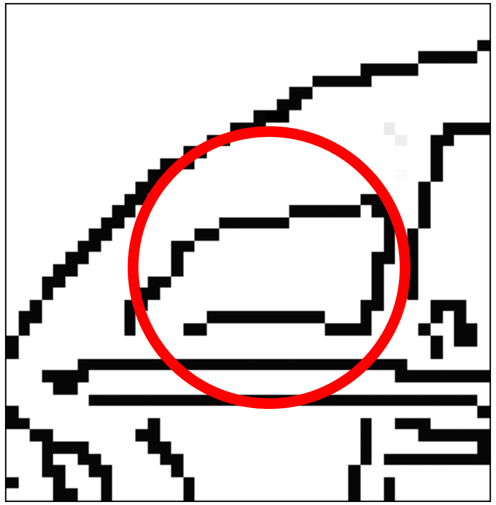


Source： S. Lazebnik

Hough transform for circles

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





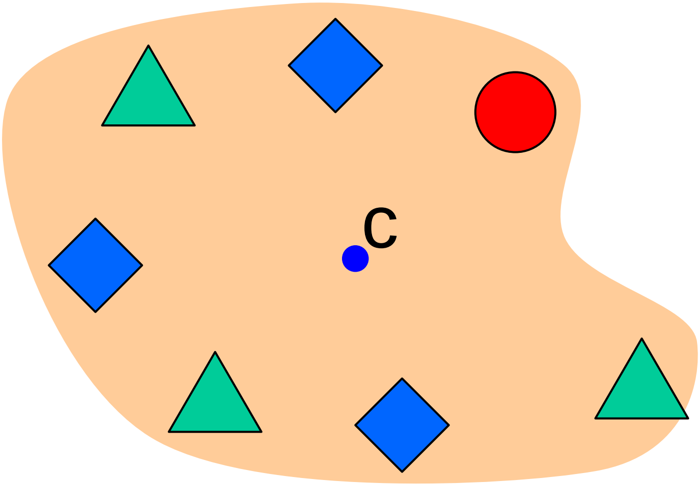
Is this more or less efficient than voting with features?

Source: S. Lazebnik

Generalized Hough transform

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

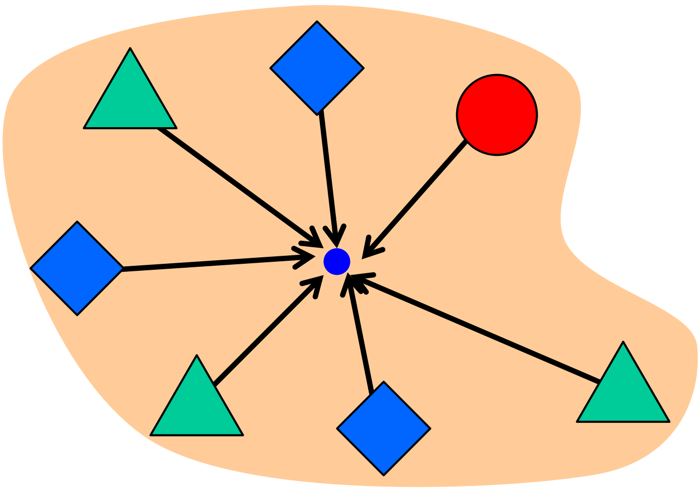
Template

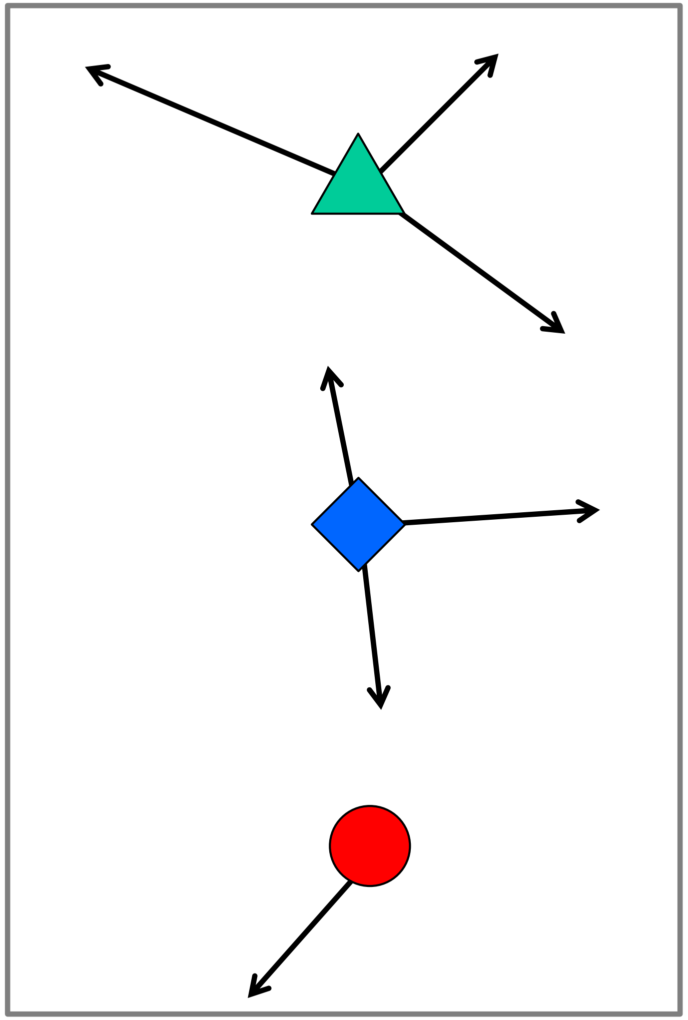


Source： S. Lazebnik

Generalized Hough transform

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



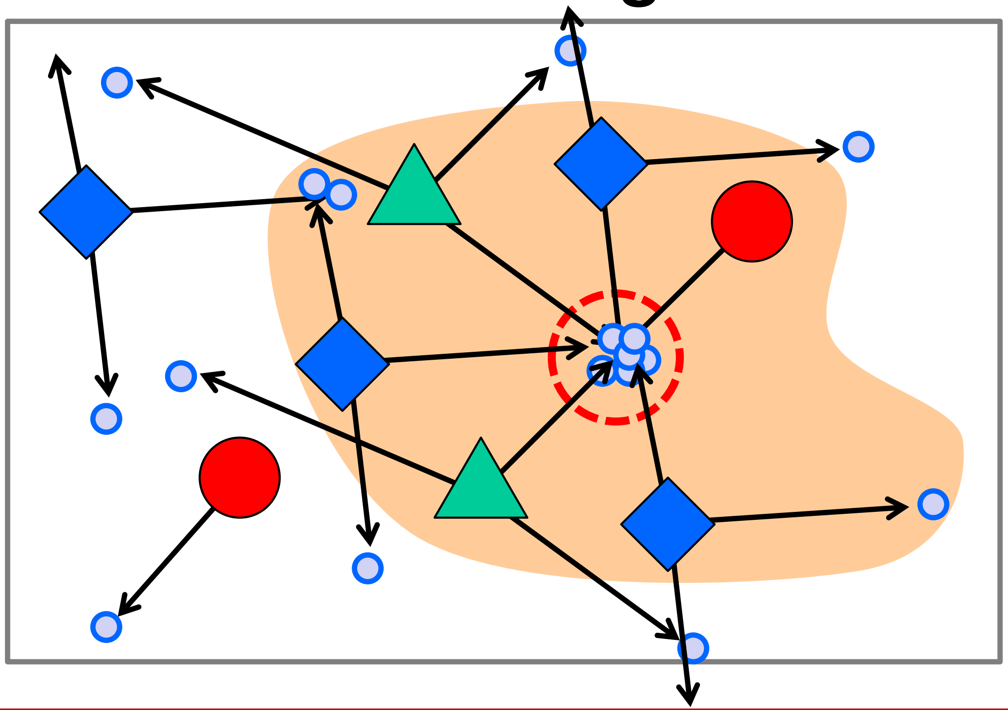


Source： S. Lazebnik

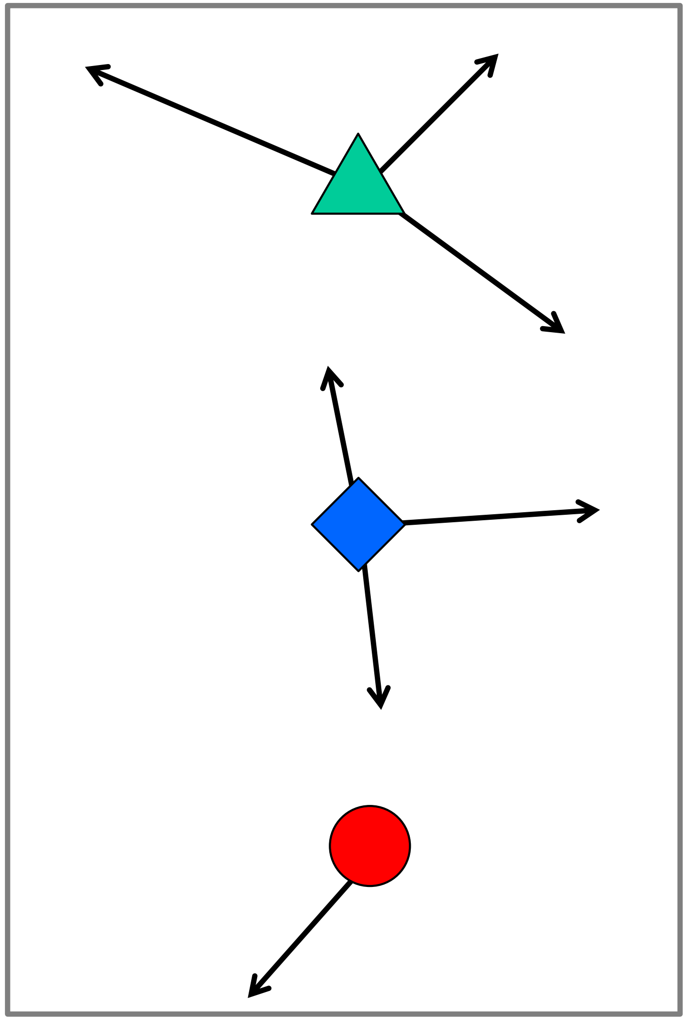
Generalized Hough transform

- 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



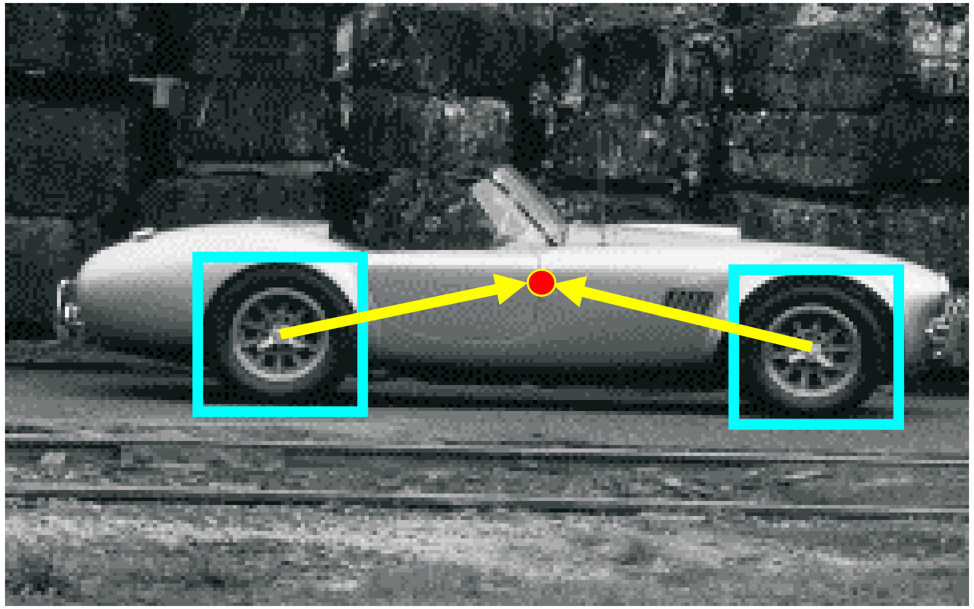
Model



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

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

Fitting: Review

- If we know which points belong to the line, how do we find the "optimal" line parameters?
  - Least squares
- What if there are outliers?
  - Robust fitting, RANSAC
- What if there are many lines?
  - Voting methods: RANSAC, Hough transform
- What if we're not even sure it's a line?
  - Model selection

Source：S. Lazebnik