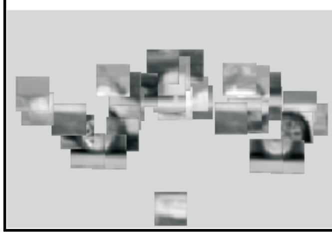
Part-based models

February 12, 2009

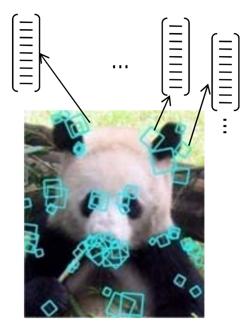
Kristen Grauman

UT-Austin



Last time: local features and bags of words representations

- Pros:
  - Good local descriptors give rich representation
  - Orderless nature means much flexibility to viewpoint
  - Able to forgo segmentation, yet still focus on particular regions
  - Quantization to words gives us discrete tokens
  - Strong empirical results



Last time: local features and bags of words representations

- Cons:
  - Lack of structure can be limiting
  - For quantized words, unclear how to best impose vocabulary
  - For a bag of words rep. left with region-of-interest / sliding window issue

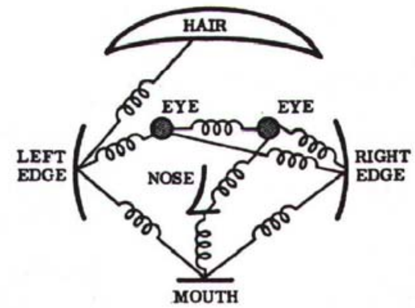
Today: part-based models

- Encode appearance of a sparse set of parts, plus their structure or relative layout

Figure credit: Rob Fergus

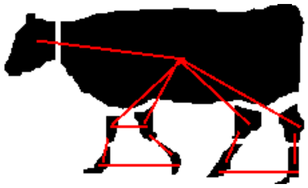
Part-based models

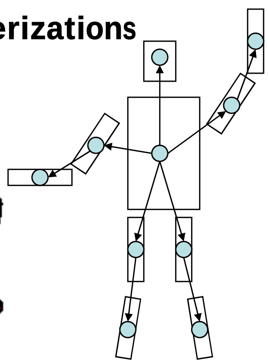
- **Fischler & Elschlager 1973**
- **Model has two components**
  - **parts (2D image fragments)**
  - **structure (configuration of parts)**

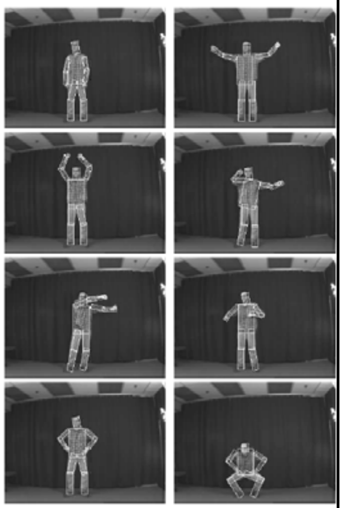


Examples of class-specific graphs

- **Articulated motion**
  - **People**
  - **Animals**
- **Special parameterizations**
  - **Limb angles**

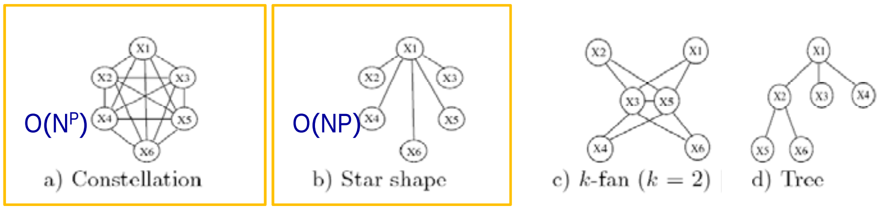






Slide credit: Rob Fergus B. Leibe 6 Images from [Kumar05, Felzenszwalb05]

Connectivity and structure

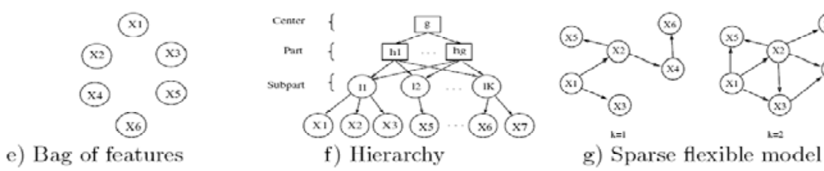


Fergus et al. '03
Fei-Fei et al. '03

Leibe et al. '04, '08
Crandall et al. '05
Fergus et al. '05

Crandall et al. '05

Felzenszwalb & Huttenlocher '05



Csurka '04
Vasconcelos '00

Bouchard & Triggs '05

Carneiro & Lowe '06

from [Carneiro & Lowe, ECCV'06]

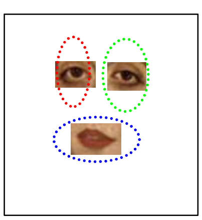
Constellation model [Fergus et al. 2003]

- Joint model for appearance and shape

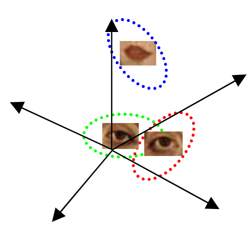
$$p(\mathbf{X},\mathbf{S},\mathbf{A}|\,\theta) = \sum_{\mathbf{h}\in H} p(\mathbf{X},\mathbf{S},\mathbf{A},\mathbf{h}|\,\theta)$$

$$= \sum_{\mathbf{h}\in H} \underbrace{p(\mathbf{A}|\mathbf{X},\mathbf{S},\mathbf{h},\theta)}_{Appearance}\underbrace{p(\mathbf{X}|\mathbf{S},\mathbf{h},\theta)}_{Shape}\underbrace{p(\mathbf{S}|\mathbf{h},\theta)}_{Rel.\ Scale}\underbrace{p(\mathbf{h}|\theta)}_{Other}$$

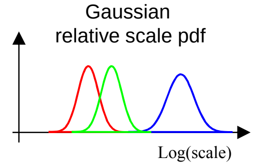
Gaussian shape pdf



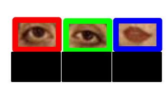
Gaussian part appearance pdf



Gaussian relative scale pdf

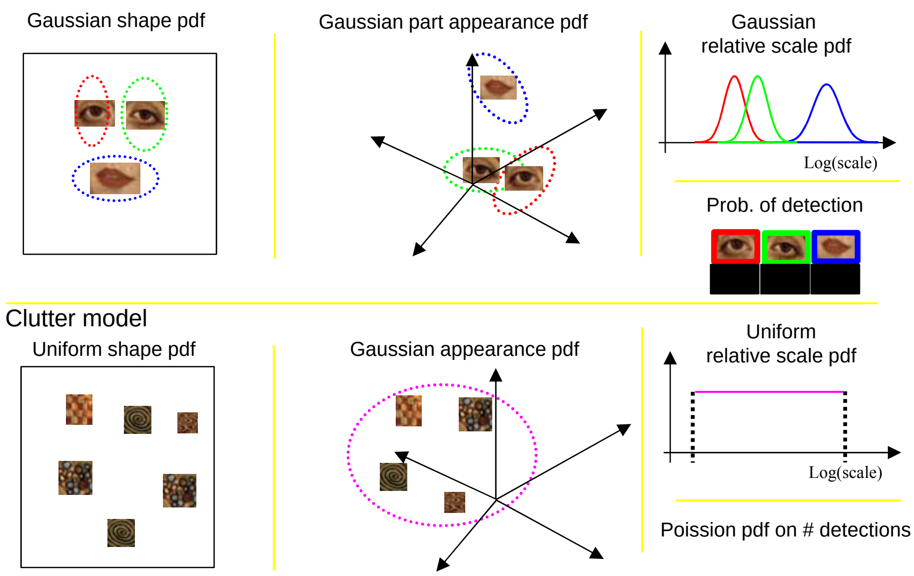


Prob. of detection



Burl et al. 1998, Weber et al. 2000, Fergus et al. 2003

Constellation model



“Weak” supervision

versus

Fig. 1. Which objects appear consistently in the left images, but not on the right side? Can a machine learn to recognize instances of the two object classes (*faces* and *cars*) without any further information provided?

Fig: Weber, Welling, Perona., 2000.

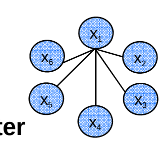
Constellation model: pros and cons

- **Advantages**
  - – Works well for many different object categories
  - – Can adapt well to categories where
    - Shape is more important
    - Appearance is more important
  - – Everything is learned from training data
  - – Weakly-supervised training possible
- **Disadvantages**
  - – Model contains many parameters that need to be estimated
  - – Cost increases exponentially with increasing number of parameters
  - $\Rightarrow$ Fully connected model restricted to small number of parts.

Slide credit: B. Leibe

Implicit Shape Model [Leibe et al. 2004]

- **Basic ideas**
  - **Learn an appearance codebook**
  - **Learn a star-topology structural model**
    - **– Features are considered independent given obj. center**



- **Algorithm: probabilistic Gen. Hough Transform**
  - **Exact correspondences** → **Prob. match to object part**
  - **NN matching** → **Soft matching**
  - **Feature location on obj.** → **Part location distribution**
  - **Uniform votes** → **Probabilistic vote weighting**
  - **Quantized Hough array** → **Continuous Hough space**

B. Leibe

Voting

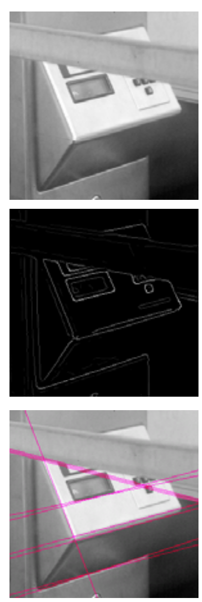
- It's not feasible to check all combinations of features by fitting a model to each possible subset.
- **Voting** is a general technique where we let the features vote for all models that are compatible with it.
  - Cycle through features, cast votes for model parameters.
  - Look for model parameters that receive a lot of votes.
- Noise & clutter features will cast votes too, *but* typically their votes should be inconsistent with the majority of "good" features.
- Ok if some features not observed, as model can span multiple fragments.

Example of voting: Fitting lines

- Given points that belong to a line, what is the line?
- How many lines are there?
- Which points belong to which lines?

- **Hough Transform** is a voting technique that can be used to answer all of these

  Main idea:
  1. Record all possible lines on which each edge point lies.
  2. Look for lines that get many votes.



Generalized Hough transform

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

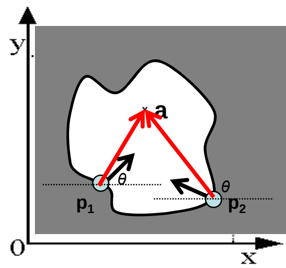


Image space

At each boundary point, compute displacement vector: $\mathbf{r} = \mathbf{a} - \mathbf{p}_i$.

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]

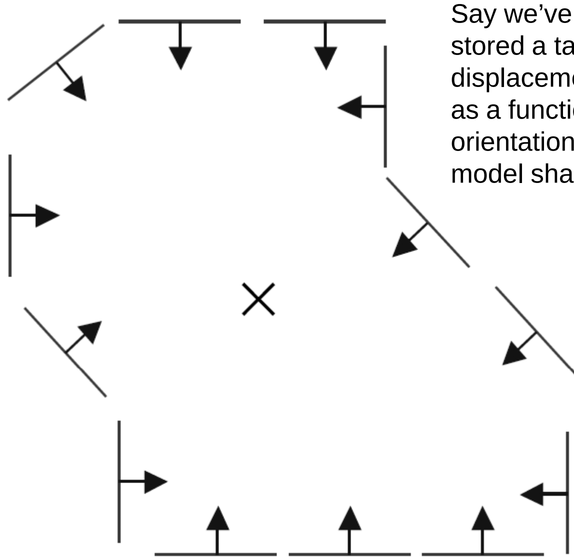
Generalized Hough transform

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.

Example

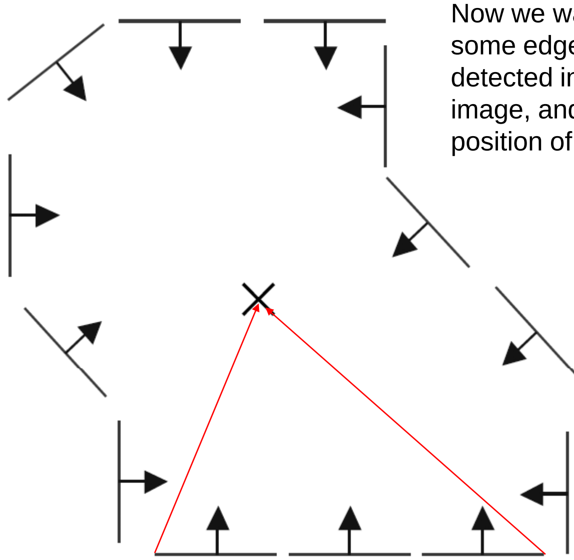


Say we've already stored a table of displacement vectors as a function of edge orientation for this model shape.

model shape

Source: L. Lazebnik

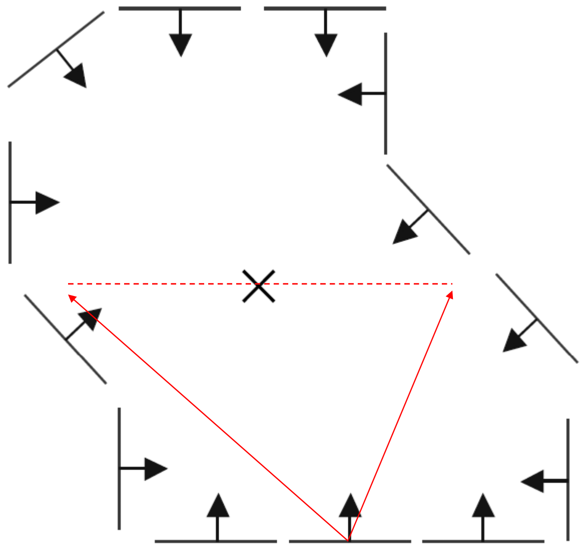
Example



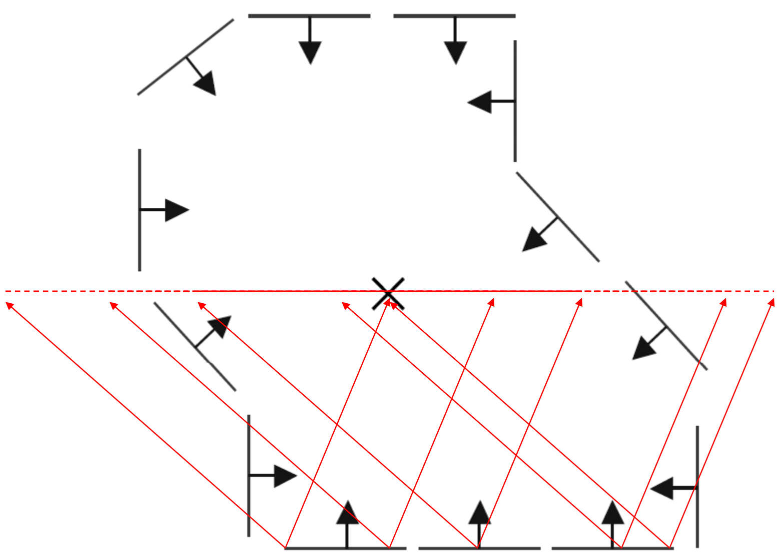
Now we want to look at some edge points detected in a *new* image, and vote on the position of that shape.

displacement vectors for model points

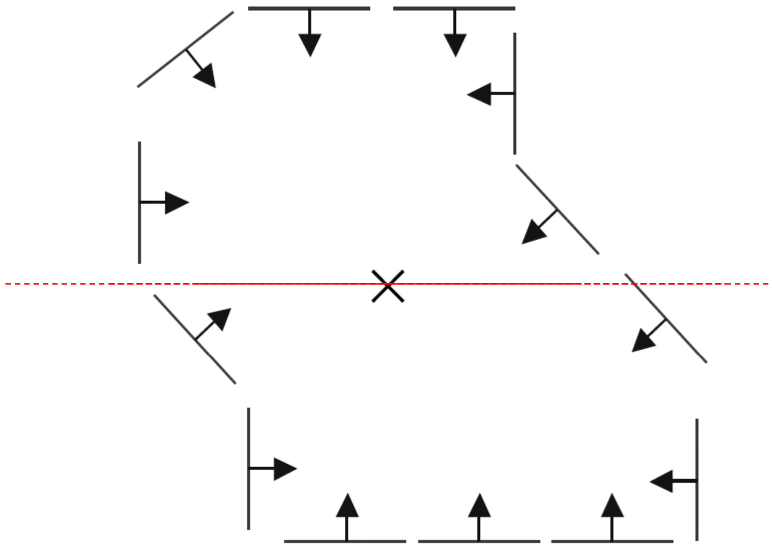
Example



Example

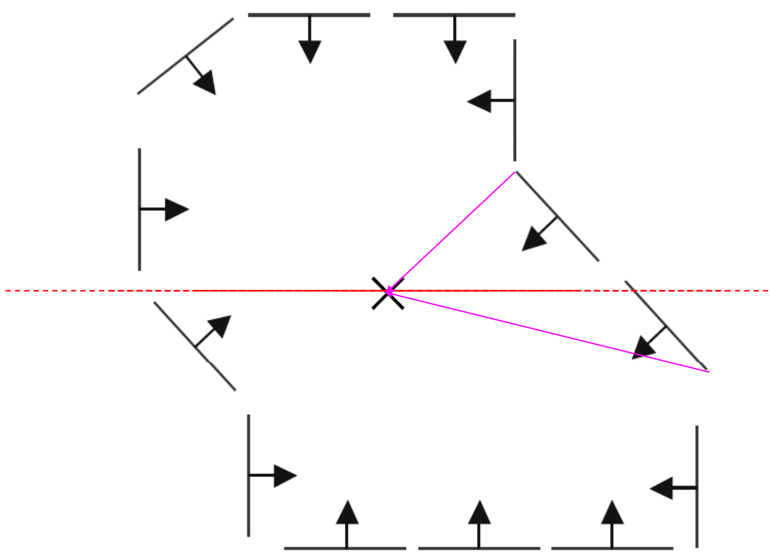


Example



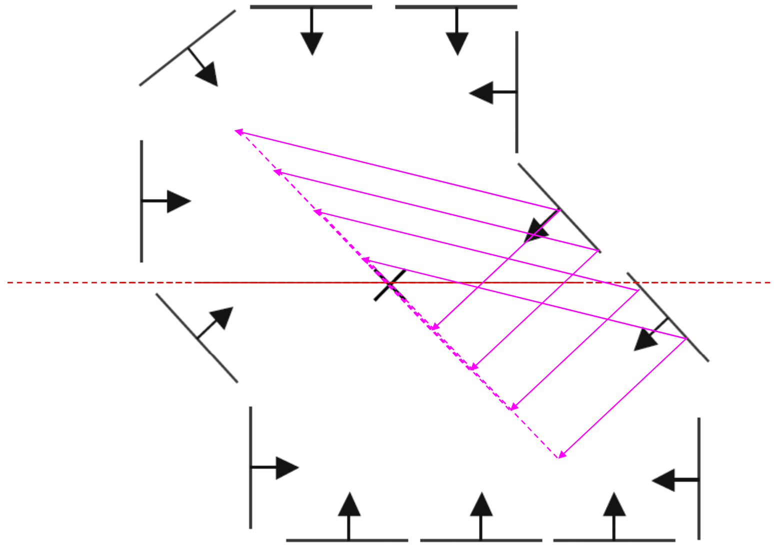
votes for points with $\theta = \uparrow$

Example



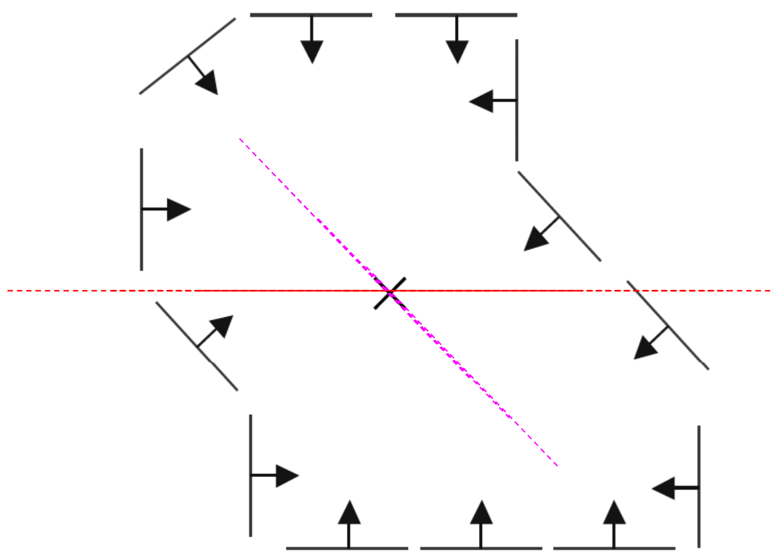
displacement vectors for model points

Example



range of voting locations for test point

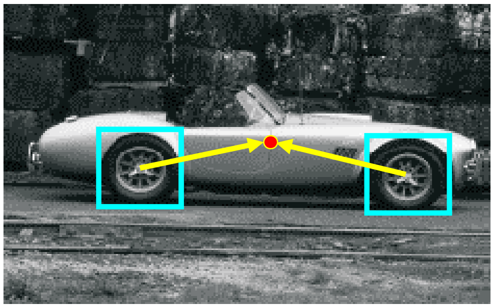
Example

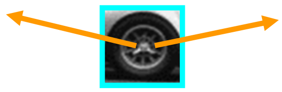


votes for points with $\theta =$ ↙

Application: Generalized Hough

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





visual codeword with displacement vectors

training 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: L. Lazebnik

Application: Generalized Hough

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

test image

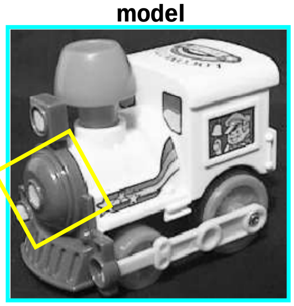
B. Leibe, A. Leonardis, and B. Schiele, Combined Object Categorization and Segmentation with an Implicit Shape Model, ECCV Workshop on Statistical Learning in Computer Vision 2004

Source: L. Lazebnik

Application: Generalized Hough

- Suppose our features are scale- and rotation-invariant
  - Then a single feature match provides an alignment hypothesis (translation, scale, orientation).

model



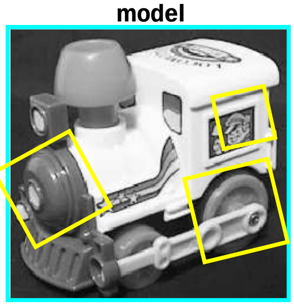
Slide credit: Svetlana Lazebnik

Fig: David Lowe

Application: Generalized Hough

- Suppose our features are scale- and rotation-invariant
  - Then a single feature match provides an alignment hypothesis (translation, scale, orientation).
  - Of course, a hypothesis from a single match is unreliable.
  - Solution: let each match vote for its hypothesis in a Hough space with very coarse bins.

model



Slide credit: Svetlana Lazebnik

Fig: David Lowe

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

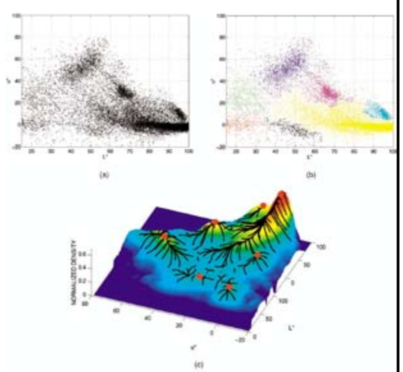
- Complexity of search time increases exponentially with the number of model parameters
- Non-target shapes can produce spurious peaks in parameter space
- Quantization: hard to pick a good grid size

Discretization in the vote space

- Choosing 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
  - …In Leibe paper, this is handled instead with continuous vote space, and mode finding via Mean Shift

Mean shift

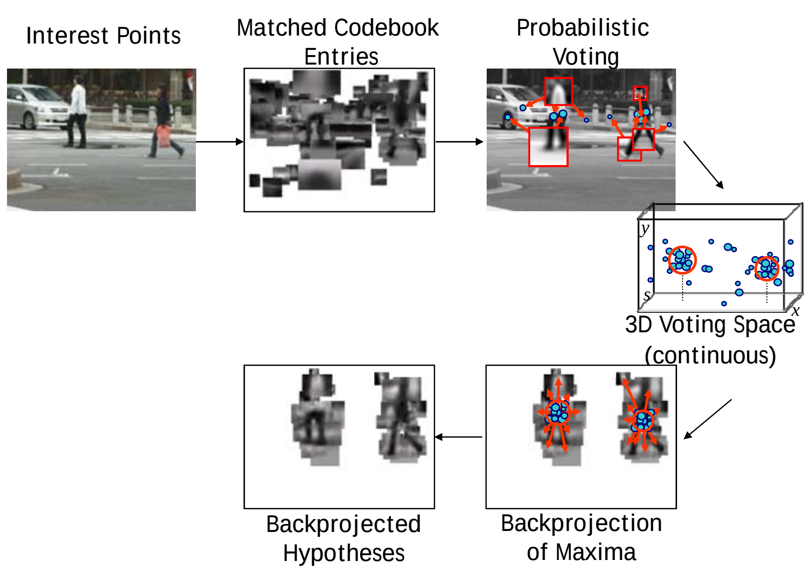
- Seeks the mode among sampled data, or point of highest density
  - Choose search window size
  - Choose initial location of search window
  - Compute mean location (centroid) in window
  - Re-center search window at mean location
  - Repeat until convergence



Fukunaga & Hostetler 1975

Comaniciu & Meer, PAMI 2002

Implicit Shape Model



Slide credit: Bastian Leibe

[Leibe04, Leibe08]

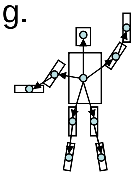
Implicit Shape Model: pros and cons

- Pros:
  - Works well for many different object categories
    - Both rigid and articulated objects
  - Flexible geometric model
    - Can recombine parts seen on different training examples
  - Optimized for detection, good localization properties
- Cons:
  - Needs bounding boxes, and seg if doing segm.
  - Only weak geometric constraints
    - Result segmentations may contain superfluous body parts.
  - Purely representative model
    - No discriminative learning

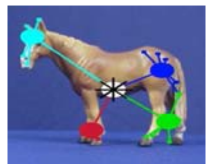
Slide credit: Bastian Leibe

Other examples of part-based models

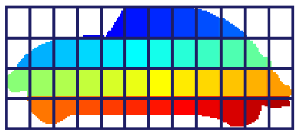
- Several other part-based models in active use, e.g.
  - Tree-structured models
    e.g. [Felzenszwalb & Huttenlocher '05]



  - Hierarchical representations
    e.g. [Bouchard & Triggs '04]



  - Dense part layouts
    e.g. [Winn & Shotton '06]



Slide credit: B. Leibe

Part-based models: issues and choices

- Invariance of the structure representation
- Part (appearance) representation
- Learning cost
- Cost of fitting to new examples
- Generative vs. discriminative
- Supervision required for training examples
- Data-driven vs. knowledge-driven model construction