Lecture 14
Visual recognition

Announcements:

- Mid-term is released today at 12:15pm
- Due on Thursday at 11am

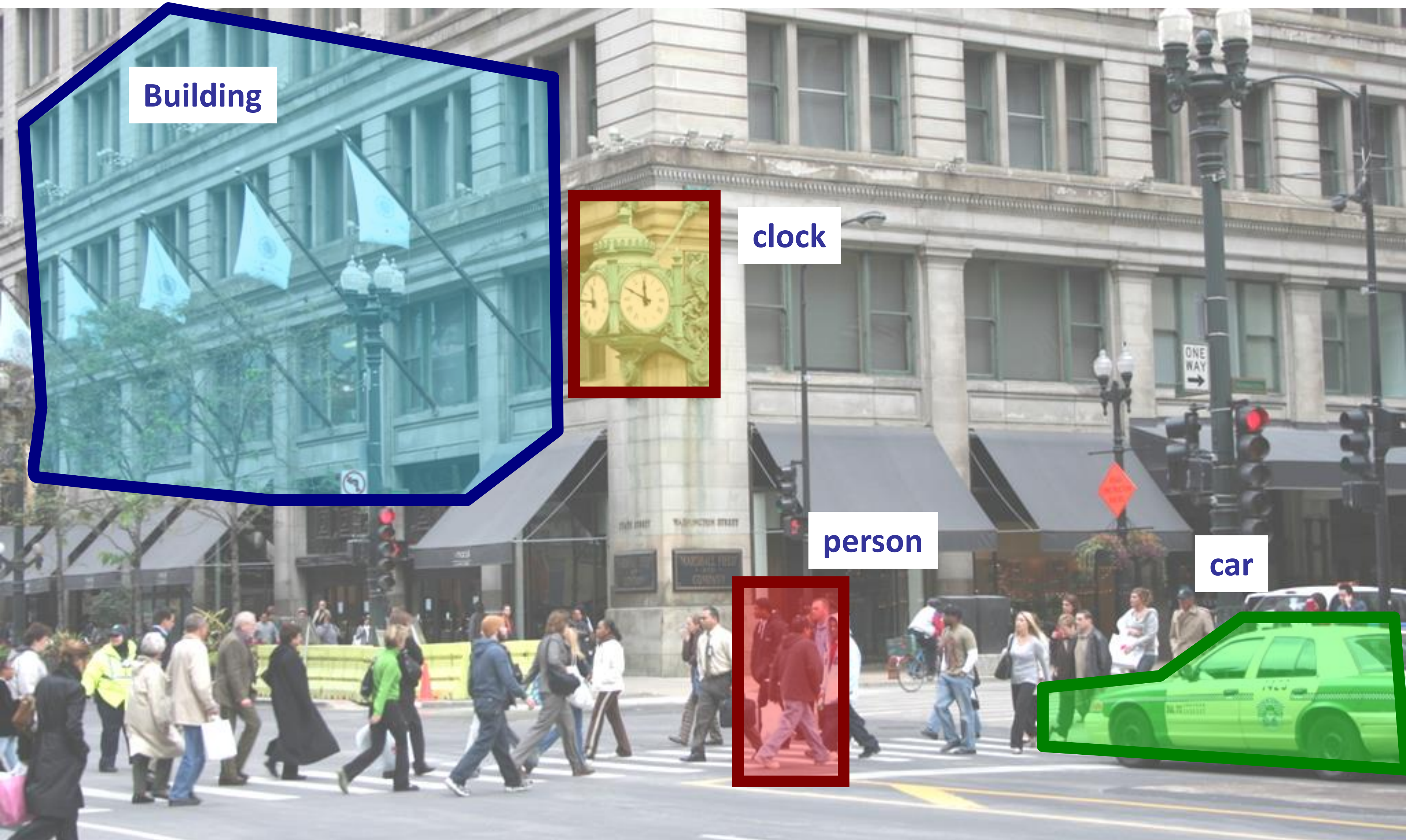
Lecture 14

Visual recognition

- 2D object detection
    - Template based approaches
    - Part-based approaches

Detection

Which object does this image contain? [where?]



Detection

– Recognition task

– Search strategy: Sliding Windows Viola, Jones 2001,

- Simple
- Computational complexity (x,y, S, θ, N of classes)
  - BSW by Lampert et al 08
  - Also, Alexe, et al 10

Detection

– Recognition task

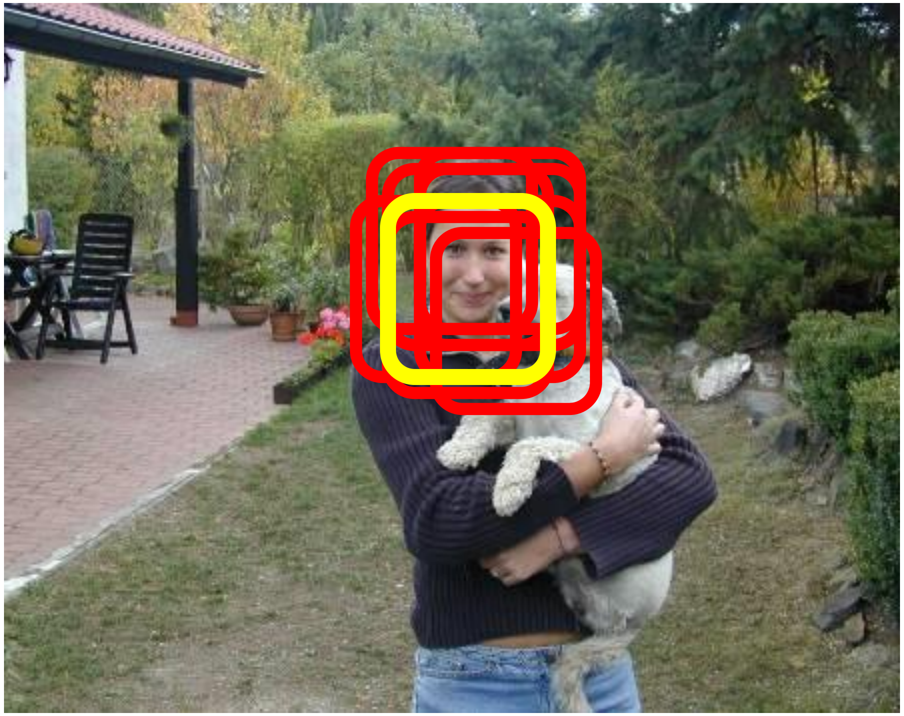
– Search strategy: Sliding Windows Viola, Jones 2001,

- Simple
- Computational complexity (x,y, S, θ, N of classes)

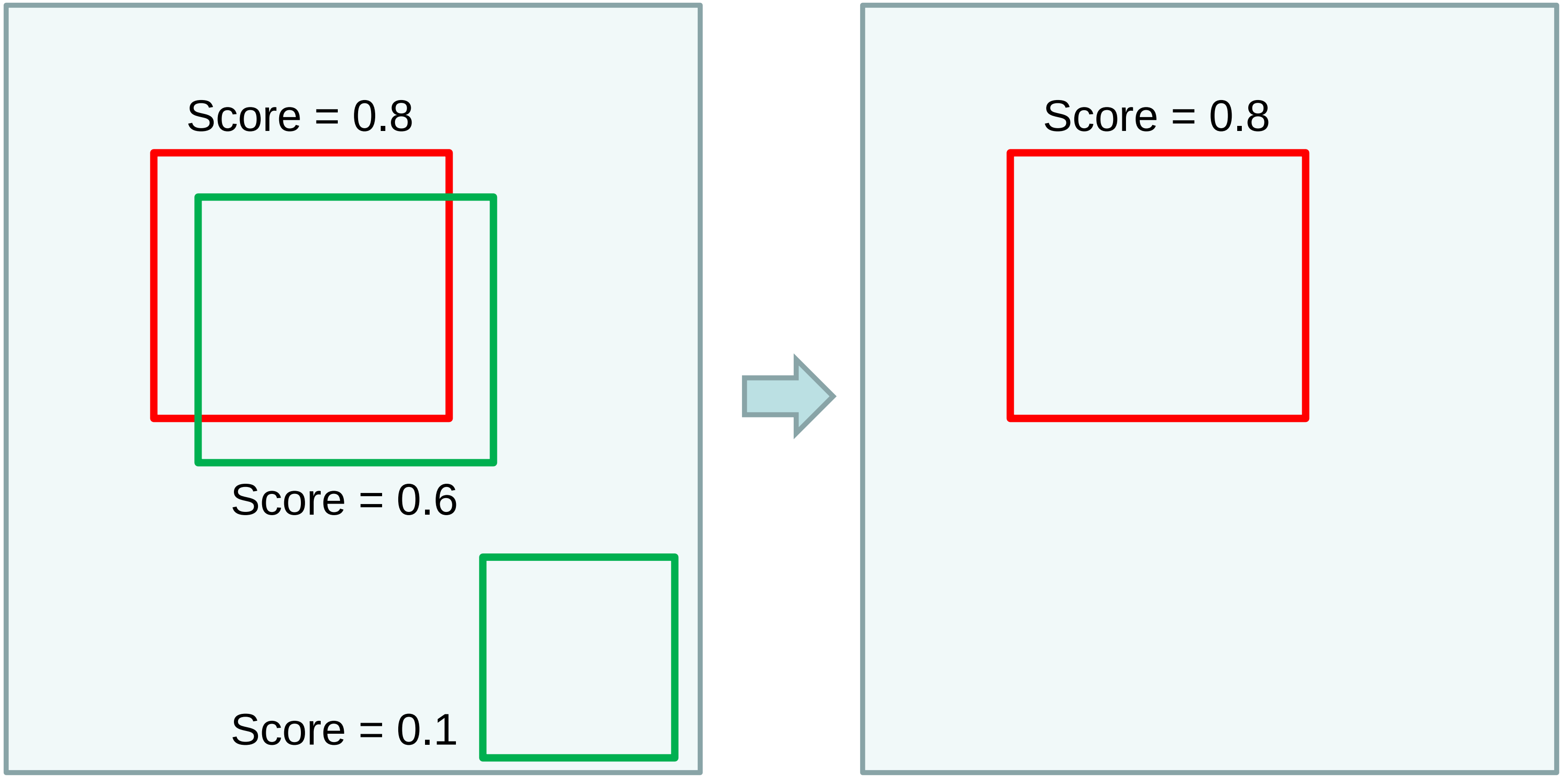
  - BSW by Lampert et al 08
  - Also, Alexe, et al 10

- Localization
  - Prone to false positive

    Non max suppression:
    Canny ’86
    ….
    Desai et al , 2009



Non-max suppression

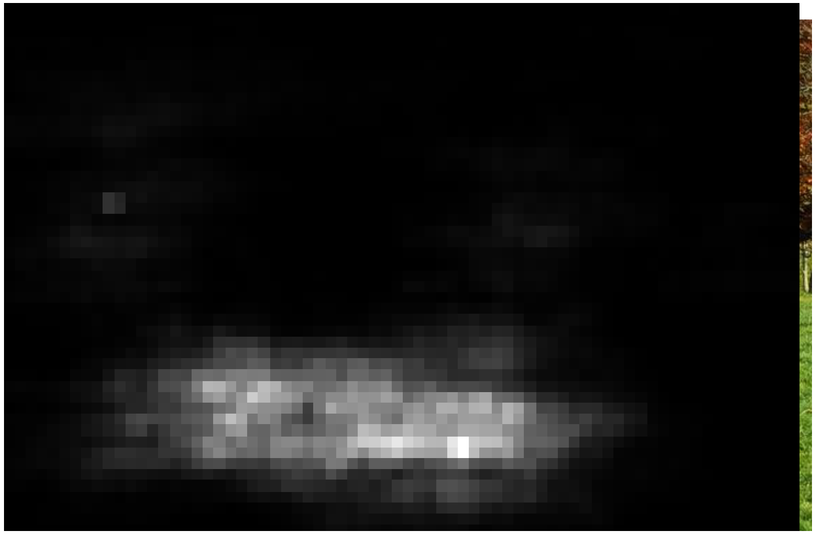


Detection

– Recognition task

– Search strategy : Probabilistic “heat maps”

- Fergus et al 03
- Leibe et al 04



Lecture 14

Visual recognition

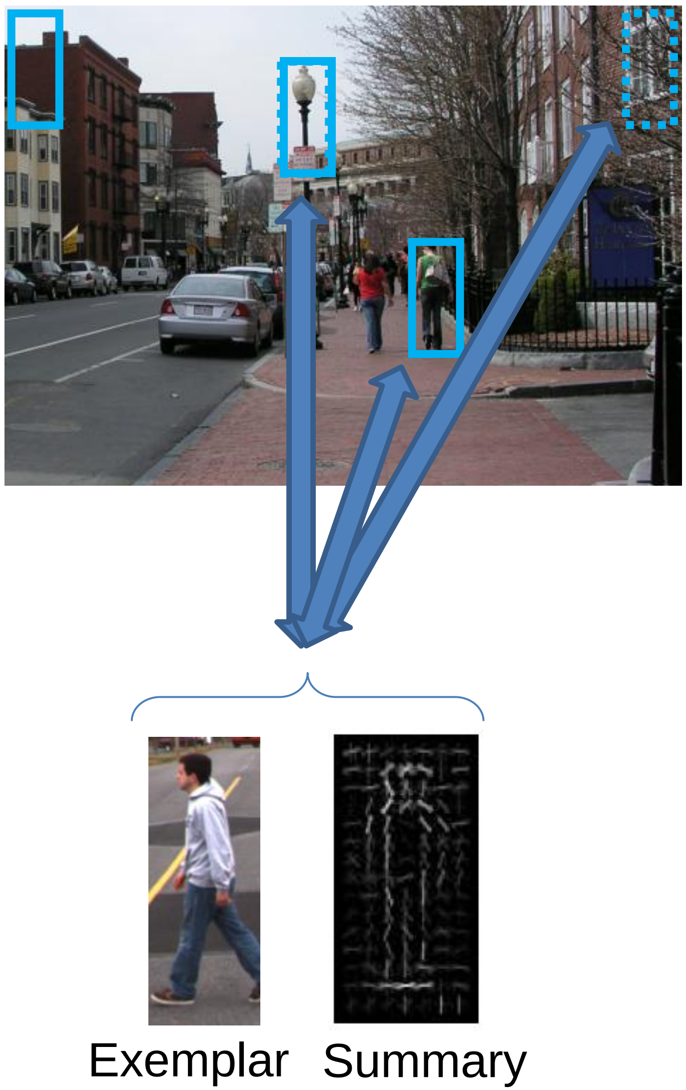
- 2D object detection
  - Template based approaches
  - Part-based approaches

Template-based detection

1. Slide a window in image
   – E.g., choose position, scale orientation

2. Compare it with a template
   – Compute similarity to an example object or to a summary representation

3. Compute a score for each comparison and compute non-max suppression to remove weak scores



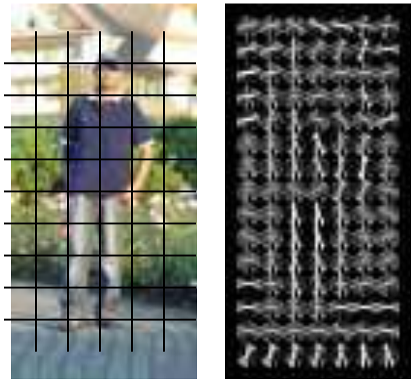
Dalal-Triggs pedestrian detector

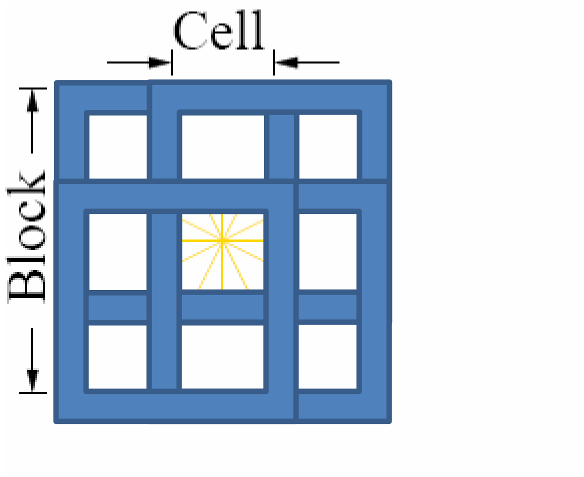
Represent an object as a collection of HoG templates

Navneet Dalal and Bill Triggs, Histograms of Oriented Gradients for Human Detection, CVPR05

HoG = Histogram of Oriented Gradients

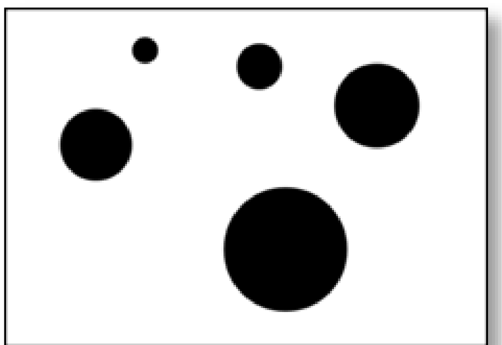
- Like SIFT, but…
  - Sampled on a dense, regular grid around the object
  - Gradients are contrast normalized in overlapping blocks

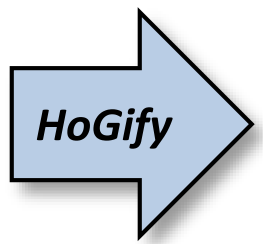


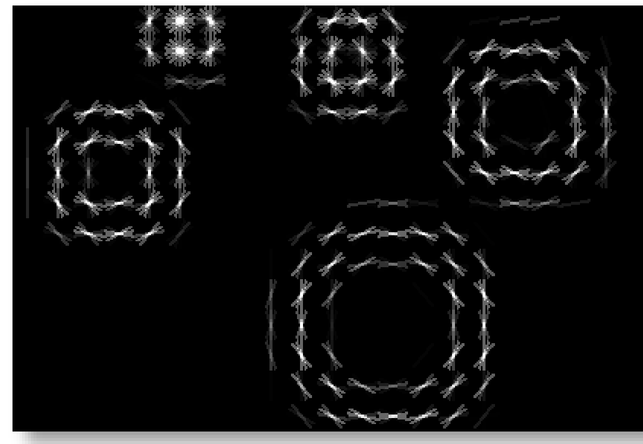


Courtesy of J Hayes

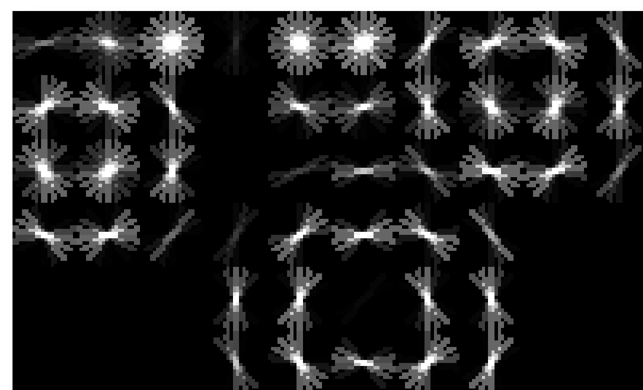
Histogram of Oriented Gradients (HoG)







10x10 cells



20x20 cells

[Dalal and Triggs, CVPR 2005]

Courtesy of N Snavely

Dalal-Triggs pedestrian detector

1. Extract fixed-sized window at each position and scale
2. Compute HOG (histogram of gradient) features within each window
3. Score the window with a linear SVM classifier
4. Perform non-maxima suppression to remove overlapping detections with lower scores

Courtesy of J Hayes

Dalal-Triggs pedestrian detector

Results

Tricks of the trade

- Details in feature computation really matter
  - E.g., normalization in Dalal-Triggs significantly improves detection rate at fixed false positive rate
- Template size
  - Typical choice is size of smallest detectable object
- “Jittering” to create synthetic positive examples
  - Create slightly rotated, translated, scaled, mirrored versions as extra positive examples
- Bootstrapping to get hard negative examples
  1. Randomly sample negative examples
  2. Train detector
  3. Keep negative examples that score > T
  4. Repeat until all high-scoring negative examples fit in memory

Courtesy of J Hayes

Limitation of template based approaches

They work

- *very well* for faces
- *fairly well* for cars and pedestrians
- *badly* for cats and dogs

- Why are some classes easier than others?

Courtesy of J Hayes

Limitation of template based approaches

Strengths

- Works very well for non-deformable objects with canonical orientations: faces, cars, pedestrians
- Fast detection

Weaknesses

- Not so well for highly deformable objects or “stuff”
- Not robust to occlusion
- Requires lots of training data if view points need to be encoded

Courtesy of J Hayes

Classic template-based Detectors

- Sung-Poggio (1994, 1998) : ~2000 citations
  - Basic idea of statistical template detection, bootstrapping to get “face-like” negative examples, multiple whole-face prototypes (in 1994)
- Rowley-Baluja-Kanade (1996-1998) : ~3600
  - “Parts” at fixed position, non-maxima suppression, simple cascade, rotation, pretty good accuracy, fast
- Schneiderman-Kanade (1998-2000,2004) : ~1700
  - Careful feature engineering, excellent results, cascade
- Viola-Jones (2001, 2004) : ~11,000
  - Haar-like features, Adaboost as feature selection, hyper-cascade, very fast, easy to implement
- Dalal-Triggs (2005) : ~6500
  - Careful feature engineering, excellent results, HOG feature, online code

Courtesy of J Hayes

Lecture 14

Visual recognition

- 2D object detection
  - Template based approaches
  - Part-based approaches

Part Based Representation

- Object as set of parts

- Model:
  – Relative locations between parts
  – Appearance of part

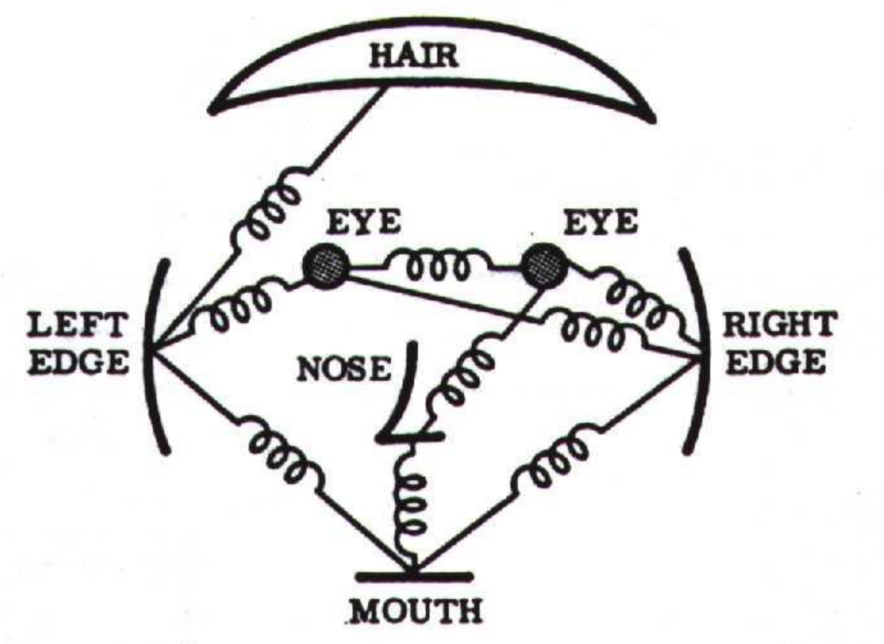
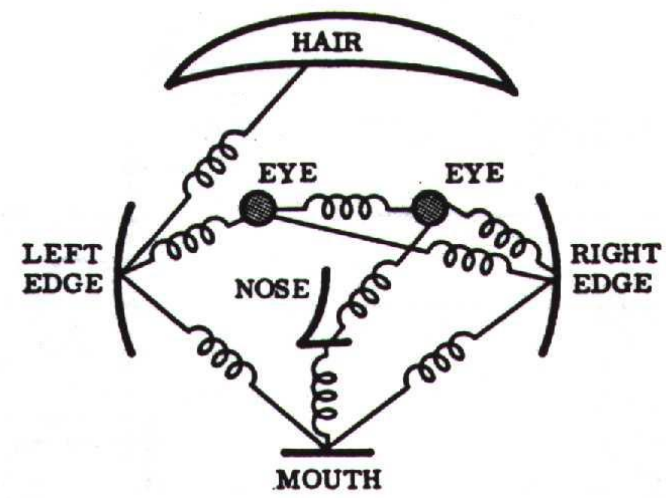


Figure from [Fischler & Elschlager 73]

History of Parts and Structure approaches

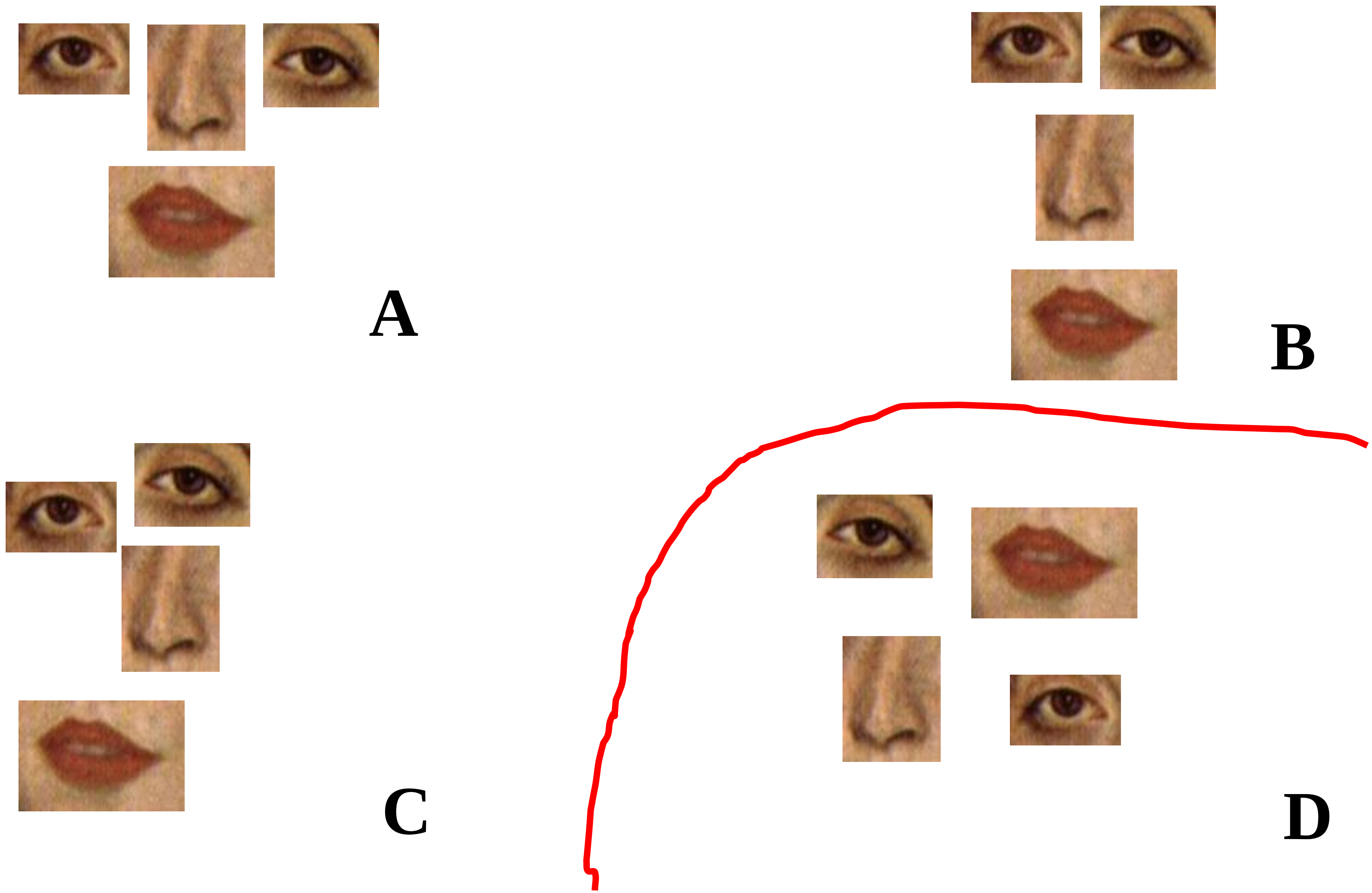


- Fischler & Elschlager 1973

- Yuille ‘91
- Brunelli & Poggio ‘93
- Lades, v.d. Malsburg et al. ‘93
- Cootes, Lanitis, Taylor et al. ‘95
- Amit & Geman ‘95, ‘99
- Perona et al. ‘95, ‘96, ’98, ’00, ’03, ‘04, ’05
- Ullman et al. 02
- Felzenszwalb & Huttenlocher ’00, ’04
- Crandall & Huttenlocher ’05, ’06
- Leibe & Schiele ’03, ’04

- Many papers since 2000

Deformations



Presence / Absence of Features

Background clutter

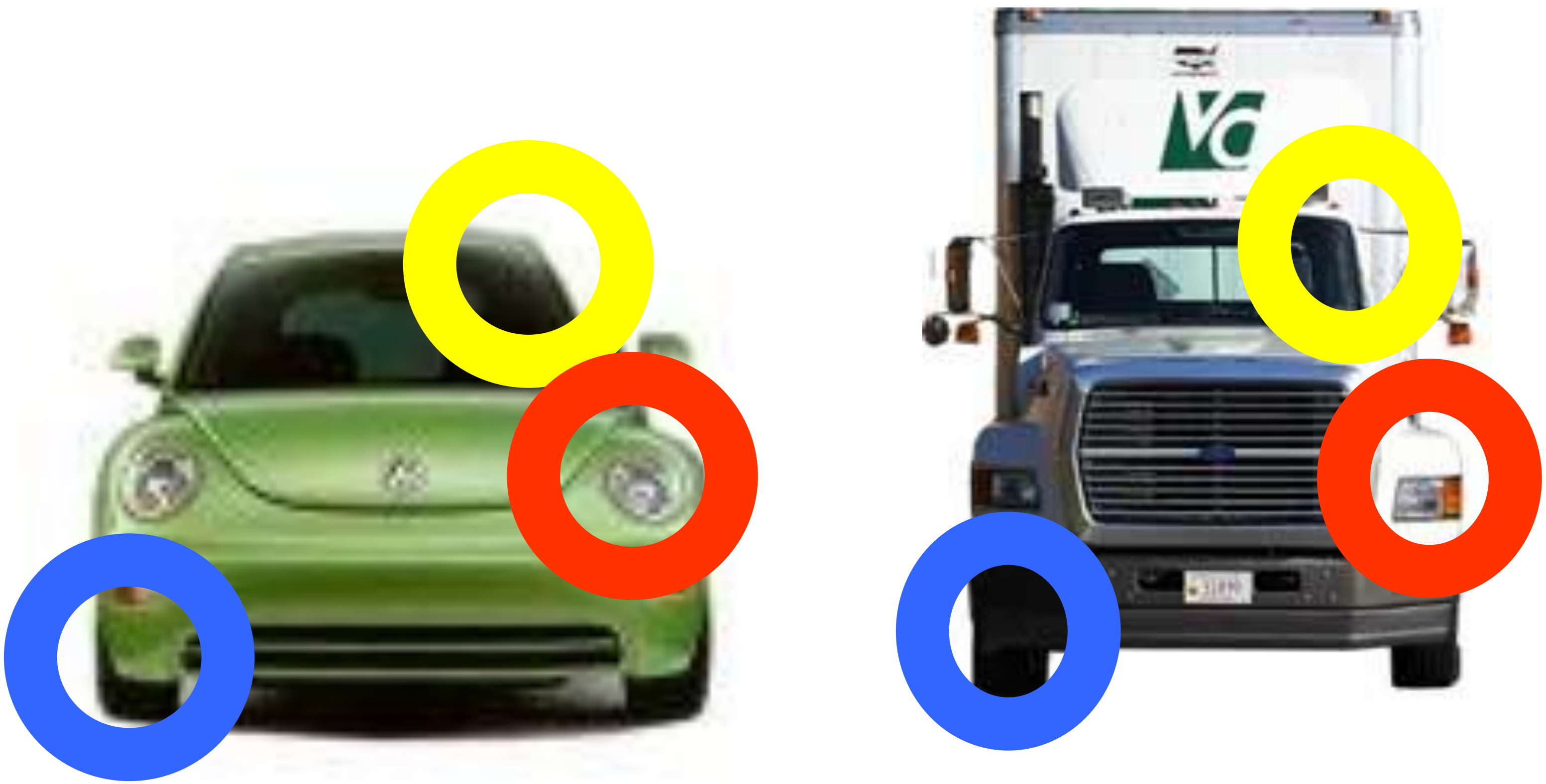
Sparse representation

Computationally tractable (10^5 pixels → 10^1 -- 10^2 parts)

But throw away potentially useful image information

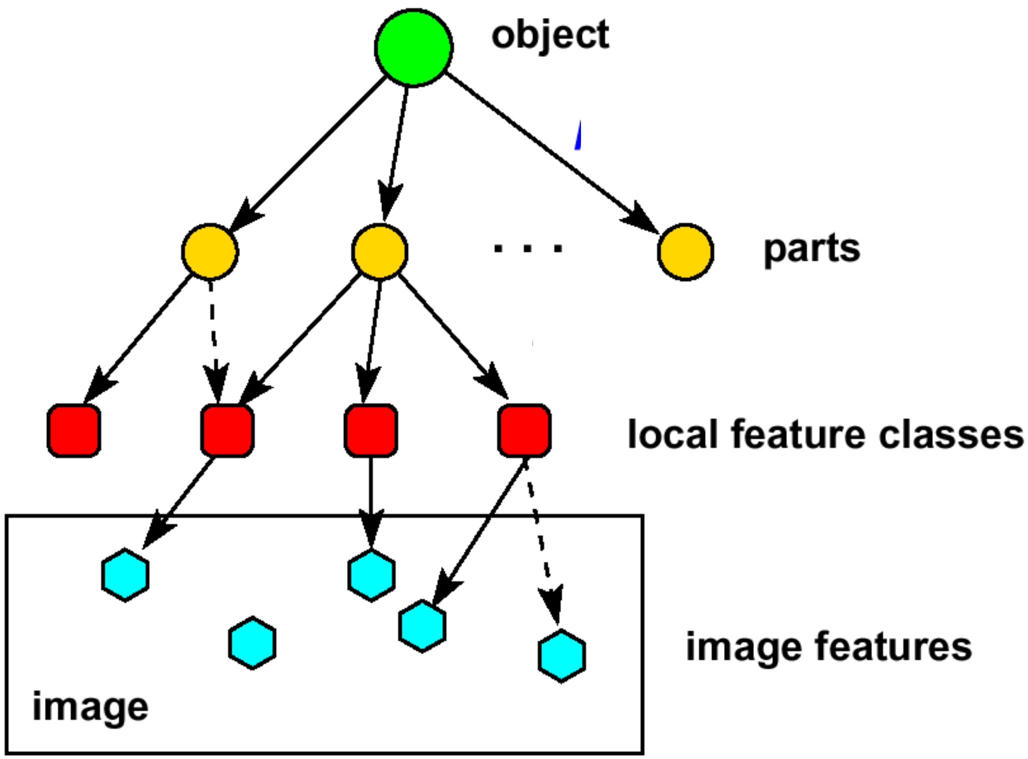
Discriminative

Parts need to be distinctive to separate from other classes



Hierarchical representations

- Pixels → Pixel groupings → Parts → Object

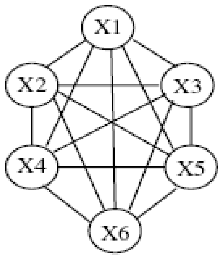


Images from [Amit98,Bouchard05]

Different connectivity structures

Fergus et al. ’03
Fei-Fei et al. ‘03

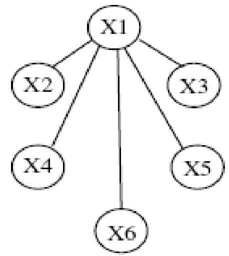
$O(N^6)$



a) Constellation [13]

Crandall et al. ‘05
Leibe 05; Felzenszwalb 09

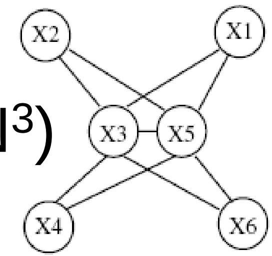
$O(N^2)$



b) Star shape [9, 14]

Crandall et al. ‘05

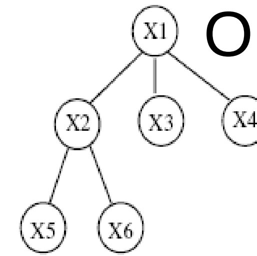
$O(N^3)$



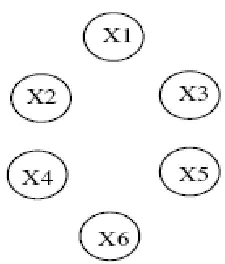
c) k-fan ($k = 2$) [9]

Felzenszwalb &
Huttenlocher ‘00

$O(N^2)$

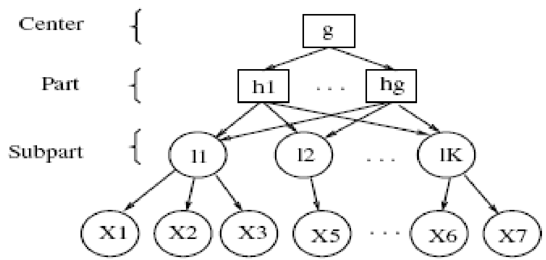


d) Tree [12]



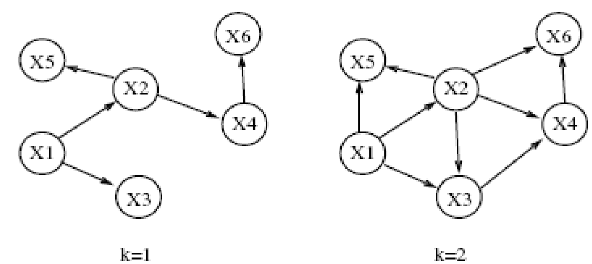
e) Bag of features [10, 21]

Csurka ’04
Vasconcelos ‘00



f) Hierarchy [4]

Bouchard & Triggs ‘05

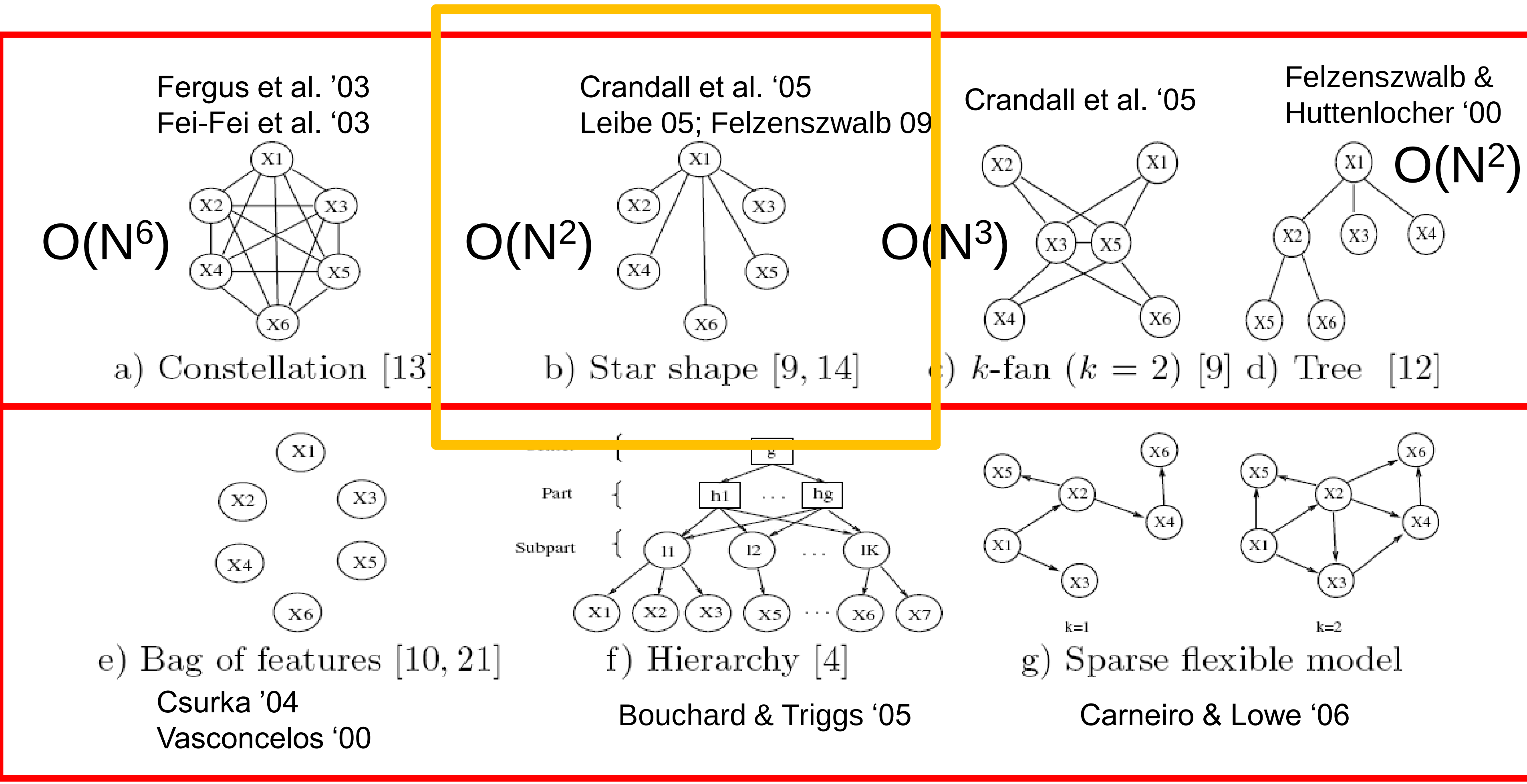


g) Sparse flexible model

Carneiro & Lowe ‘06

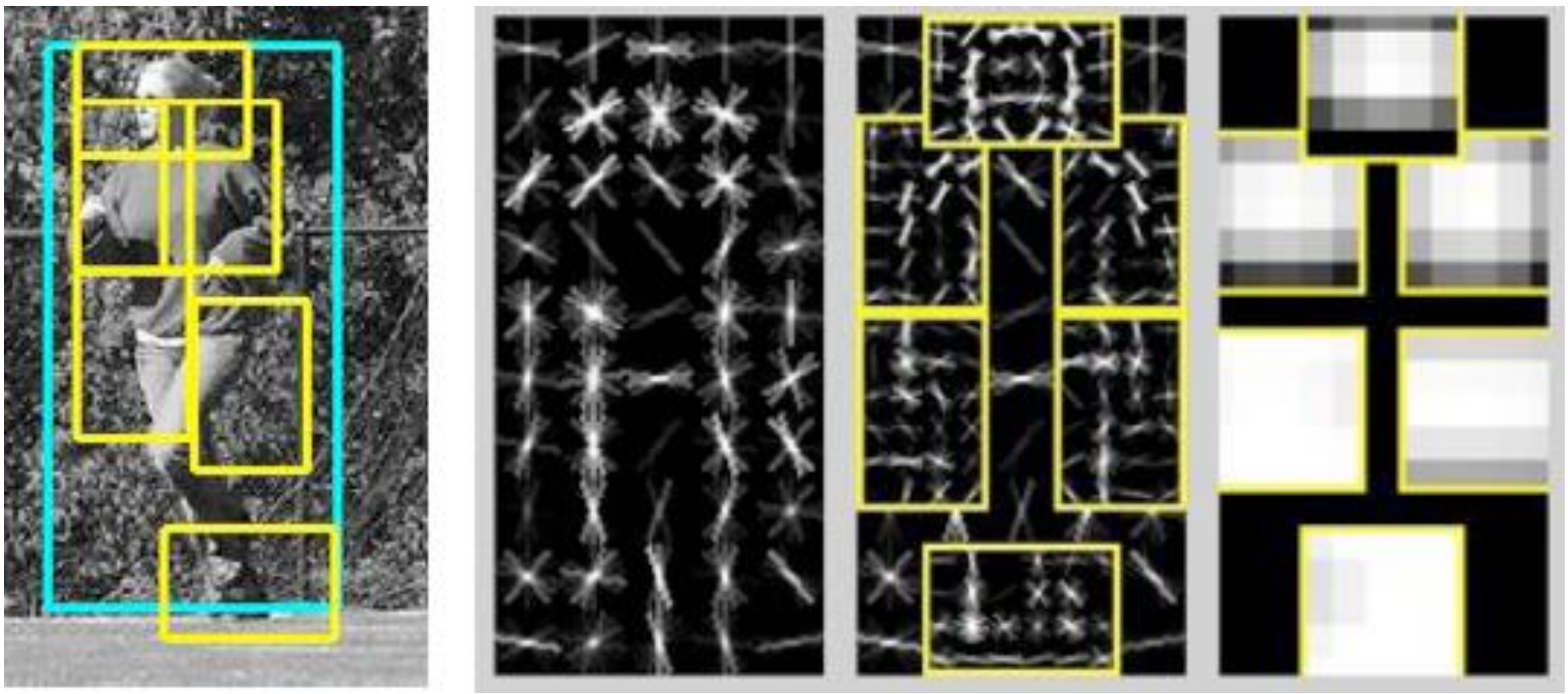
from Sparse Flexible Models of Local Features
Gustavo Carneiro and David Lowe, ECCV 2006

Different connectivity structures



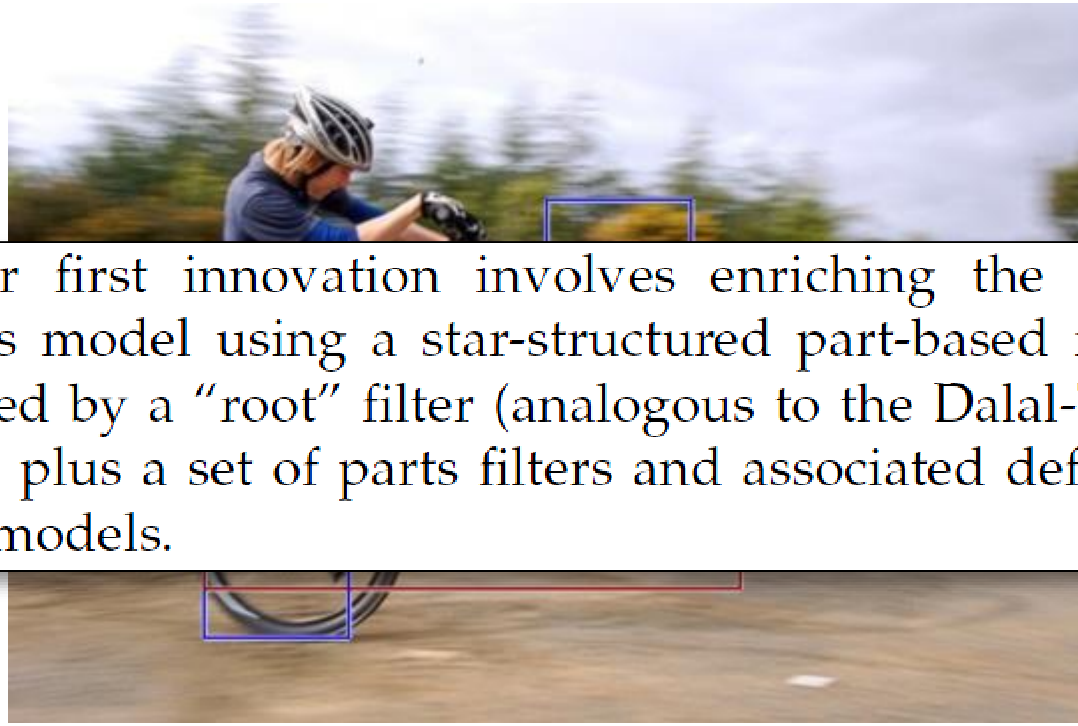
from Sparse Flexible Models of Local Features
Gustavo Carneiro and David Lowe, ECCV 2006

Star models by Latent SVM

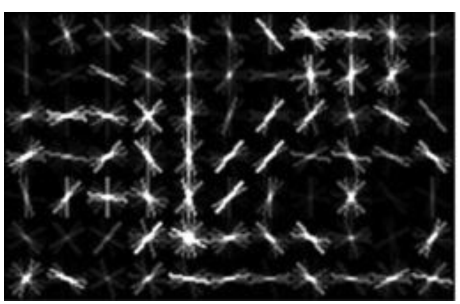


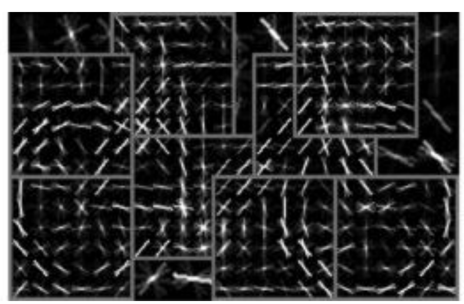
Felzenszwalb, McAllester, Ramanan, 08
• Source code:

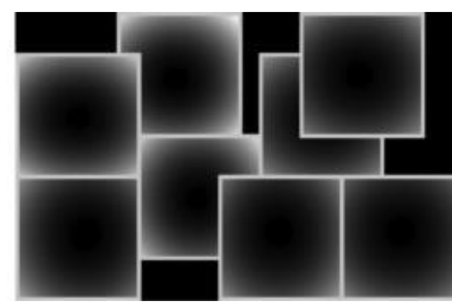
Deformable Part Models (DPM)



Our first innovation involves enriching the Dalal-Triggs model using a star-structured part-based model defined by a “root” filter (analogous to the Dalal-Triggs filter) plus a set of parts filters and associated deformation models.





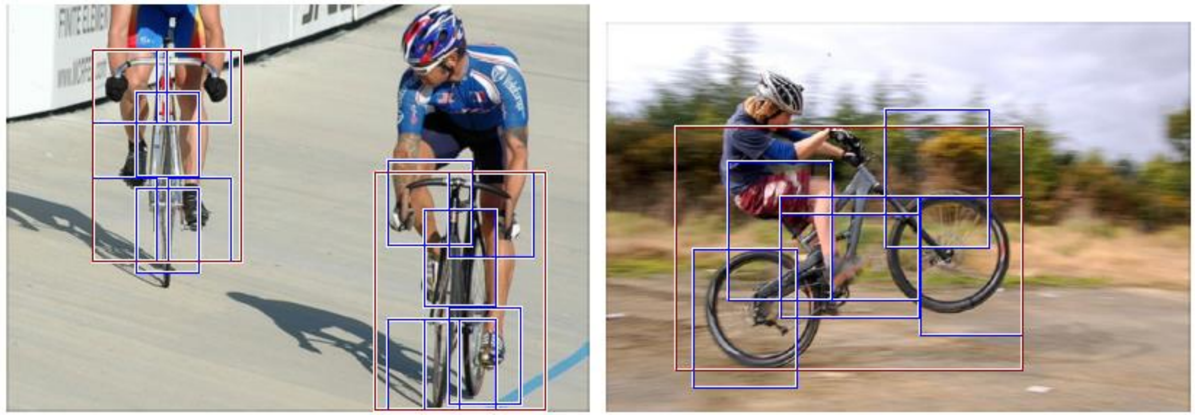


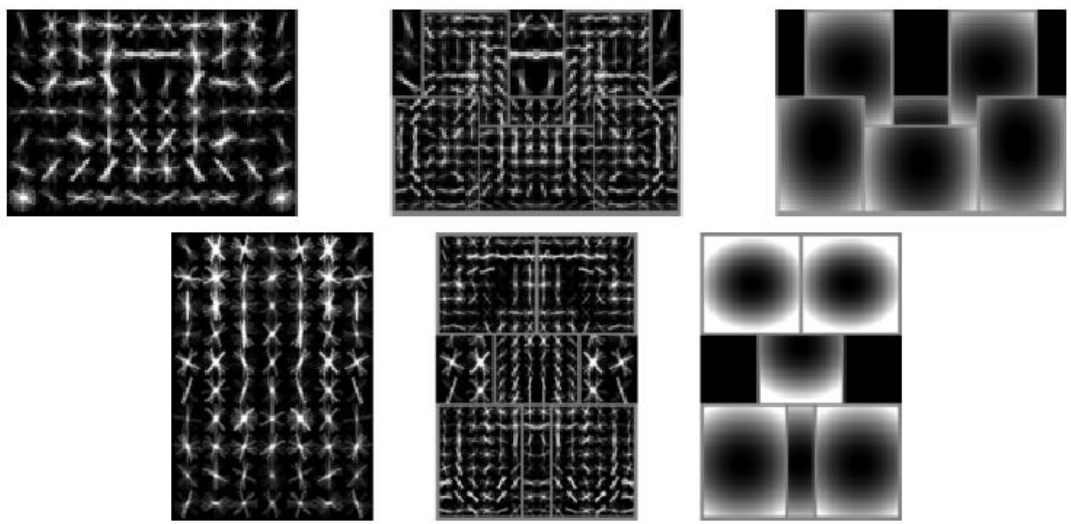
Felzenszwalb, et al., **Discriminatively Trained Deformable Part Models,**
http://people.cs.uchicago.edu/~pff/latent/

Latent SVMs

- Rather than training a single linear SVM separating positive examples…
- … cluster positive examples into “components” and train a classifier for each (using all negative examples)

Two-component bicycle model

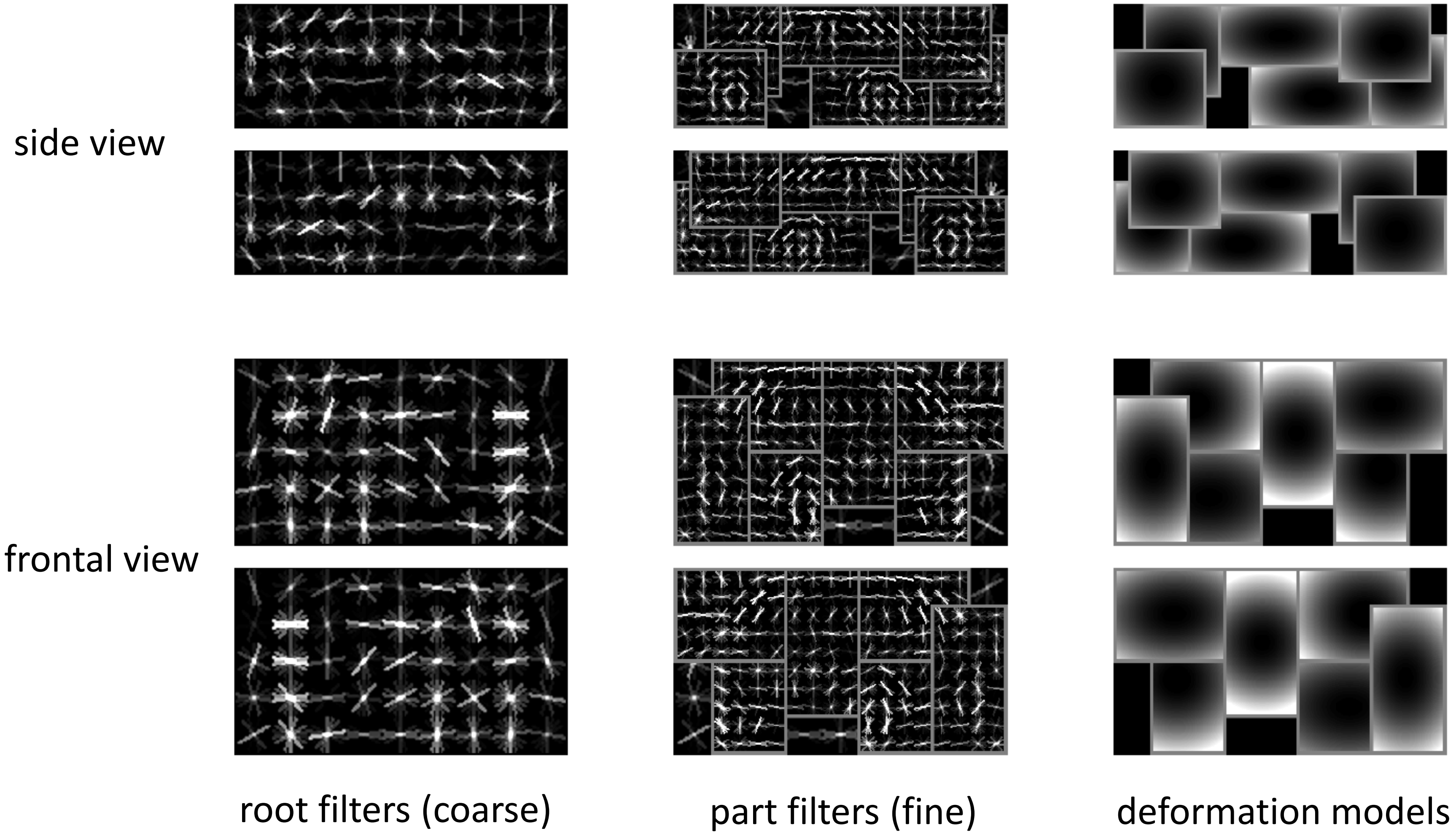




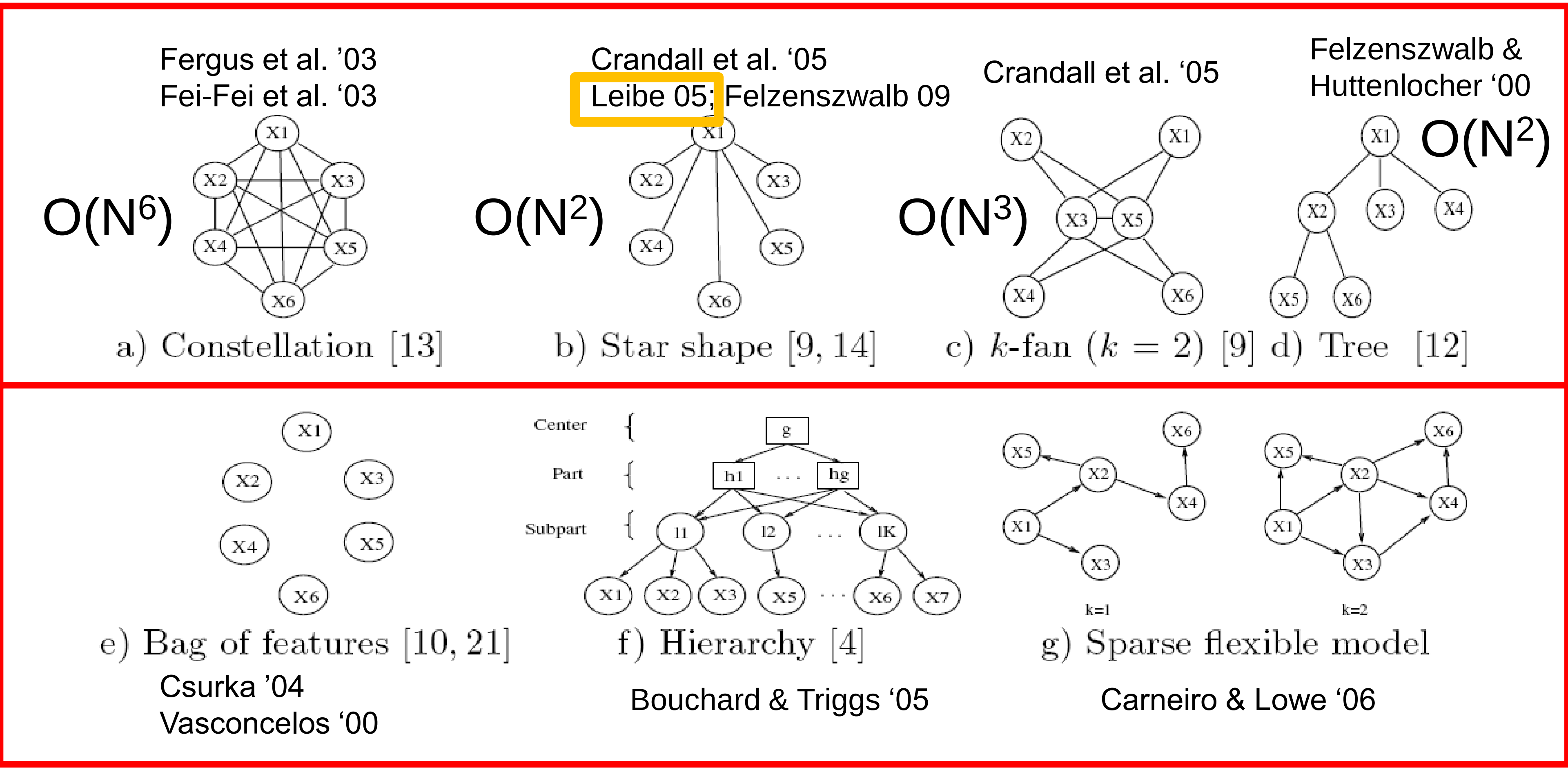
“side” component

“frontal” component

Six-component car model



Different connectivity structures

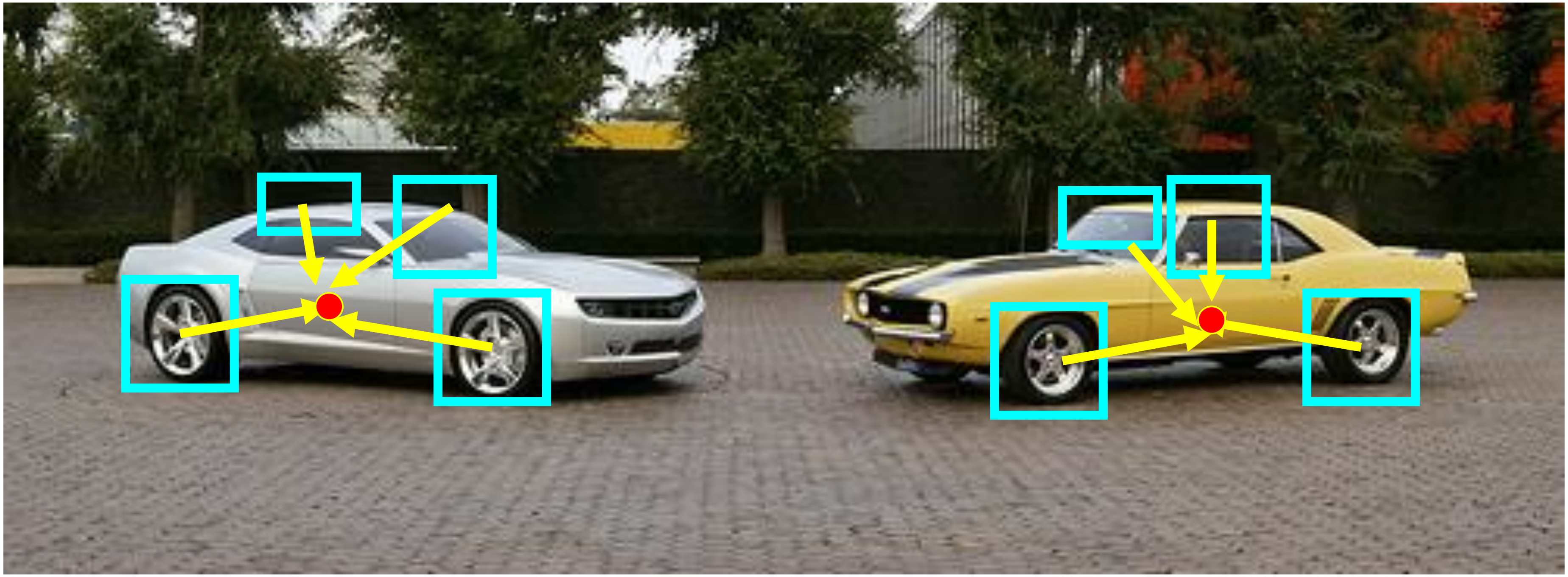


from Sparse Flexible Models of Local Features
Gustavo Carneiro and David Lowe, ECCV 2006

Implicit shape models by generalized Hough voting

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

Object representation:
Constellation of parts w.r.t object centroid

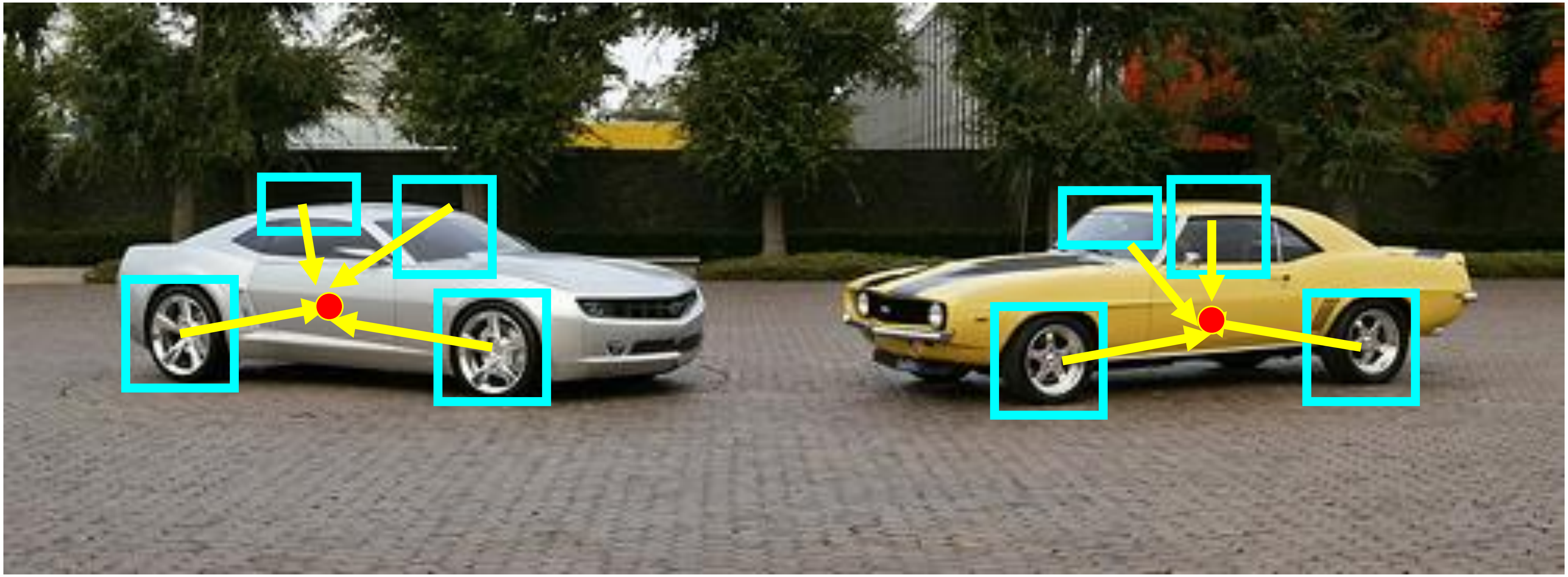


B. Leibe, A. Leonardis, and B. Schiele, Combined Object Categorization and Segmentation with an Implicit Shape Model, ECCV Workshop on Statistical Learning in Computer Vision 2004

Object representation:

How to capture constellation of parts?

Using Hough Voting

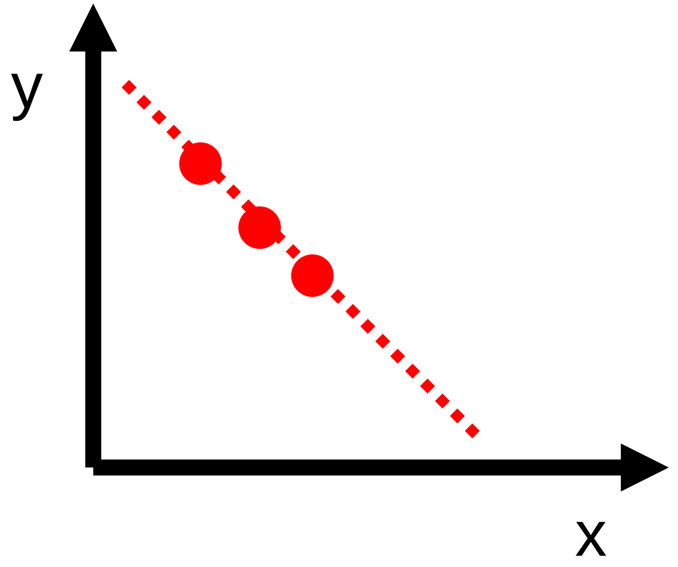


B. Leibe, A. Leonardis, and B. Schiele, [Combined Object Categorization and Segmentation with an Implicit Shape Model](), ECCV Workshop on Statistical Learning in Computer Vision 2004

Hough transform

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

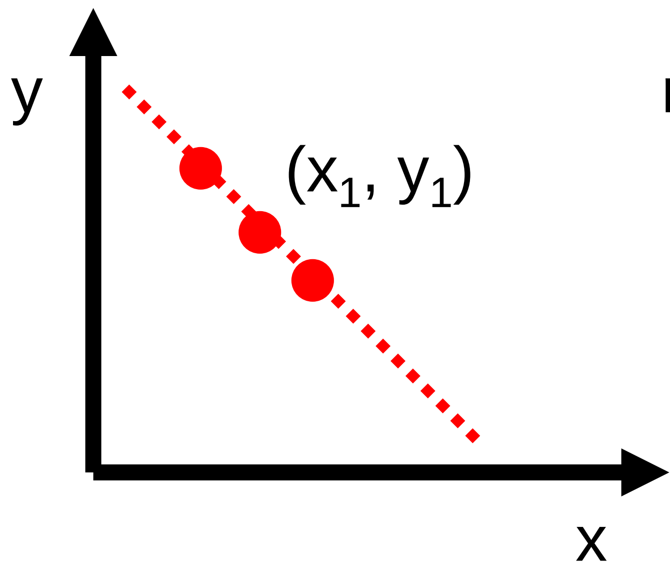
Given a set of points, find the curve or line that explains the data points best

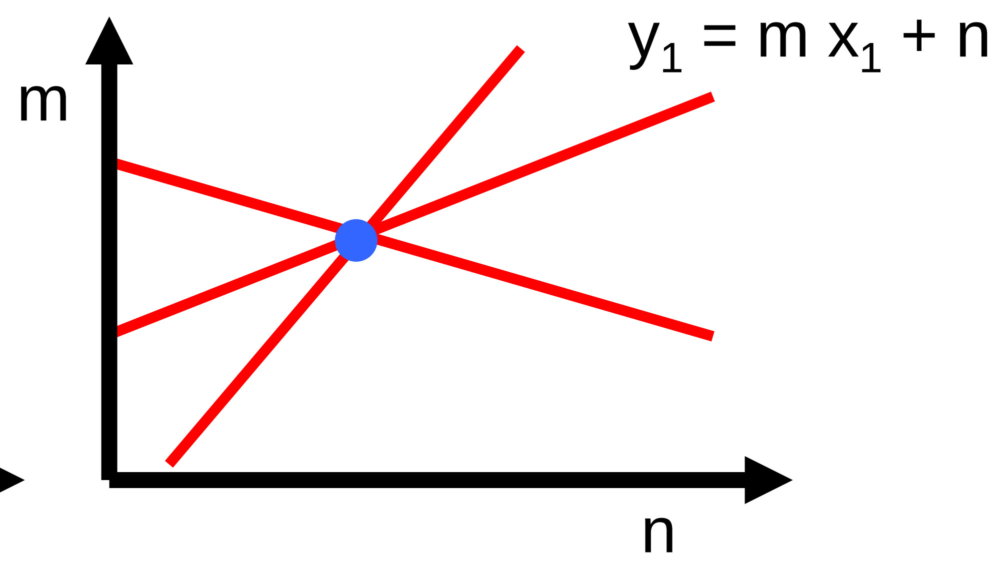


Hough transform

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

Given a set of points, find the curve or line that explains the data points best





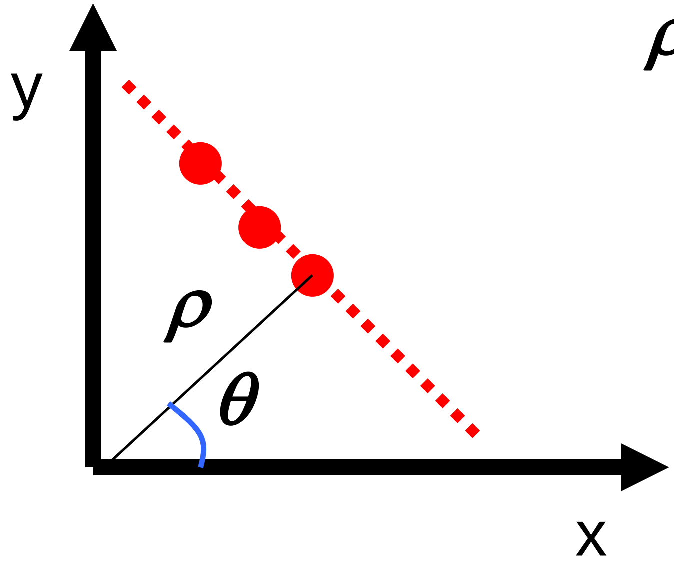
Hough space

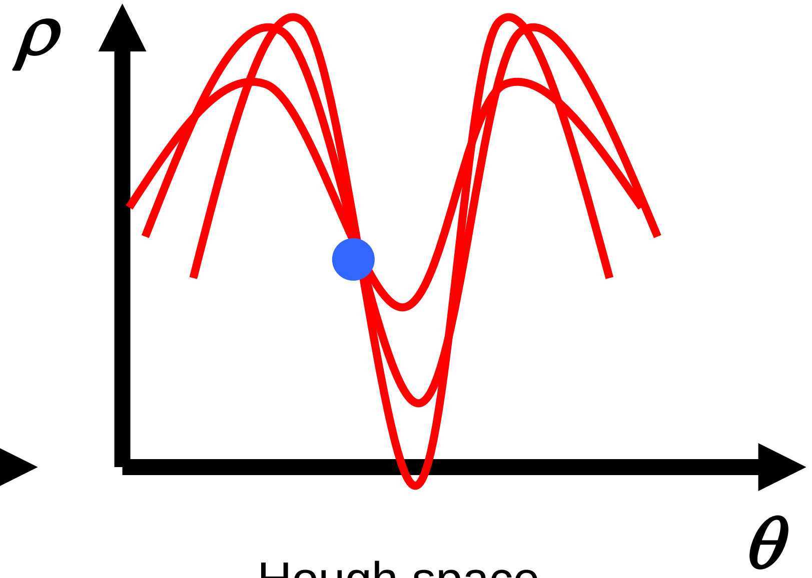
$$y = m\ x + n$$

Hough transform

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

- Use a polar representation for the parameter space

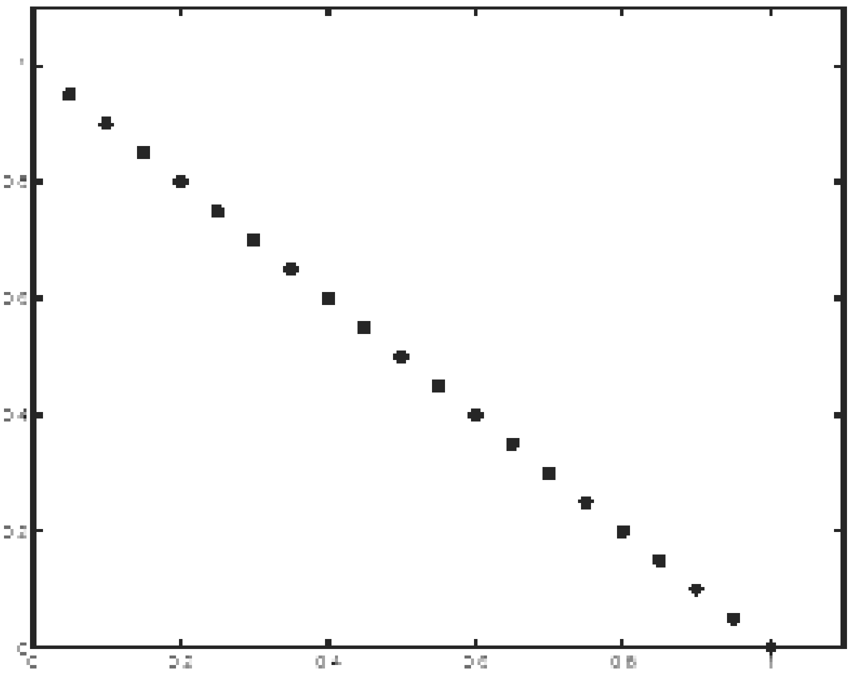




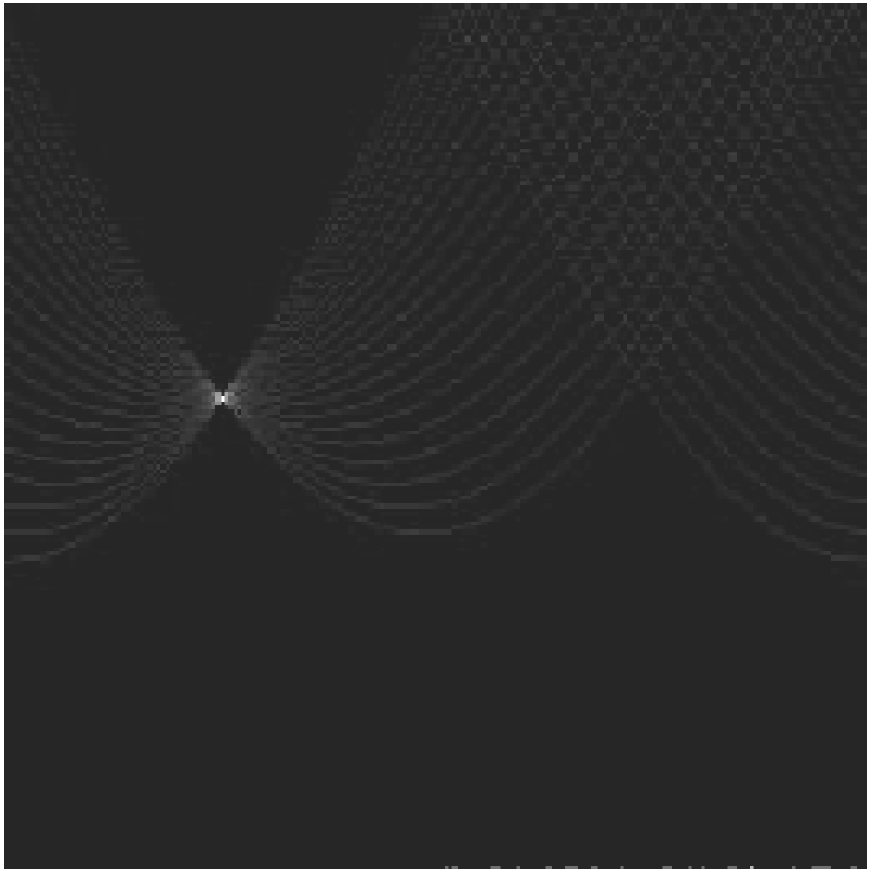
Hough space

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

Hough transform - experiments



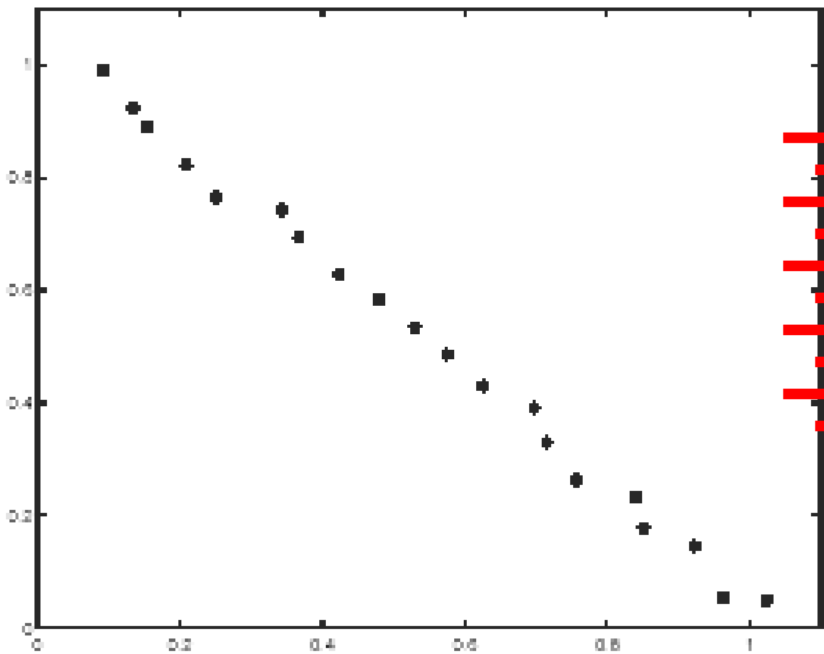
features



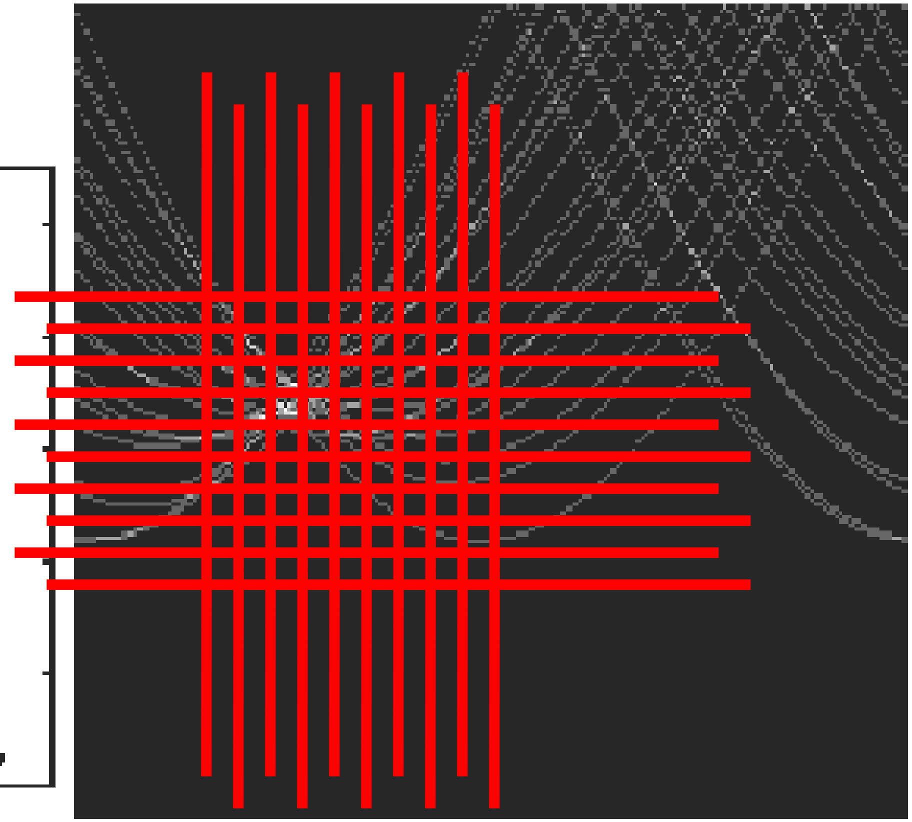
votes

Hough transform - experiments

Noisy data



features



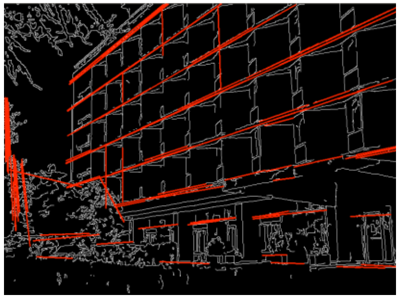
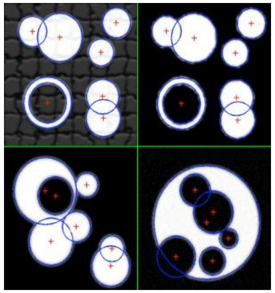
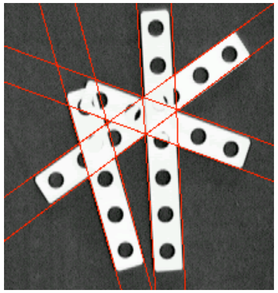
votes

IDEA: introduce a grid a count intersection points in each cell

Issue: Grid size needs to be adjusted…

Generalized Hough Transform

- Parts in query image vote for a learnt model
- Significant aggregations of votes correspond to models
- Complexity : # parts * # votes
  - Significantly lower than brute force search (e.g., sliding window detectors)
- Popular for detecting parameterized shapes
  - Hough'59, Duda&Hart'72, Ballard'81,…

Slide Modified from *S. Maji*

Generalized Hough Transform

- GOAL: detect arbitrary shapes defined by boundary points and a reference point

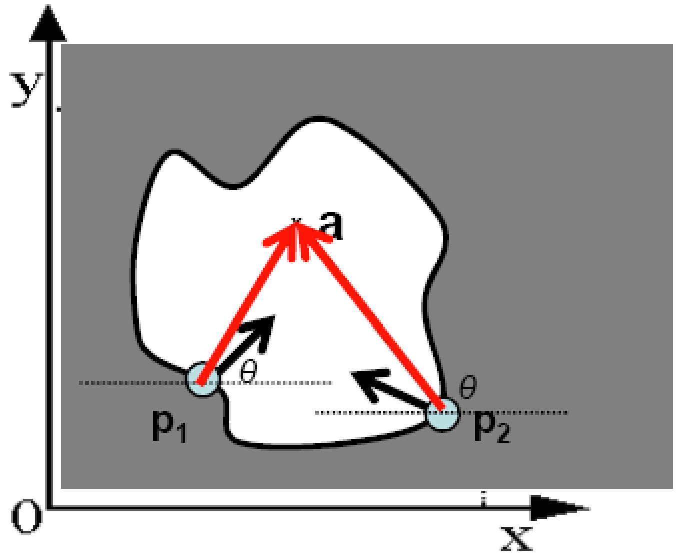


Image space

Learning a model:

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

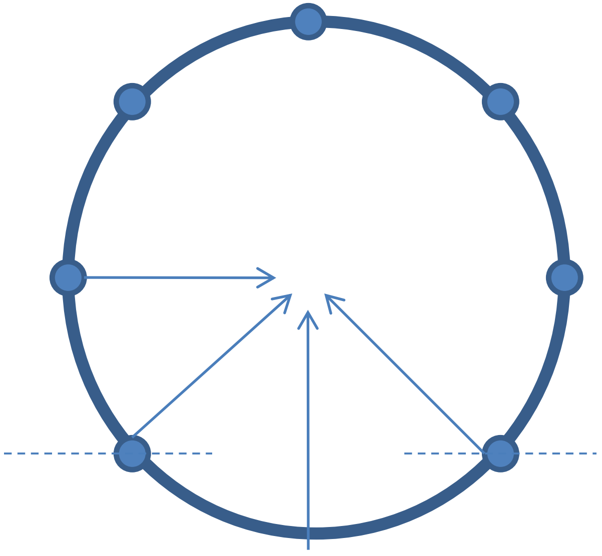
For a given model shape: store these vectors in a table indexed by gradient orientation θ.

[Dana H. Ballard, Generalizing the Hough Transform to Detect Arbitrary Shapes, 1980]

Credit slide: C. *Grauman*

Example

Circle model



θ	rx	ry
0	1	0
45	0.7	0.7
90	0	1
135	-0.7	0.7
…		
270	0.7	-0.7

Generalized Hough Transform

<u>Detecting the *model shape in a new image:*</u>

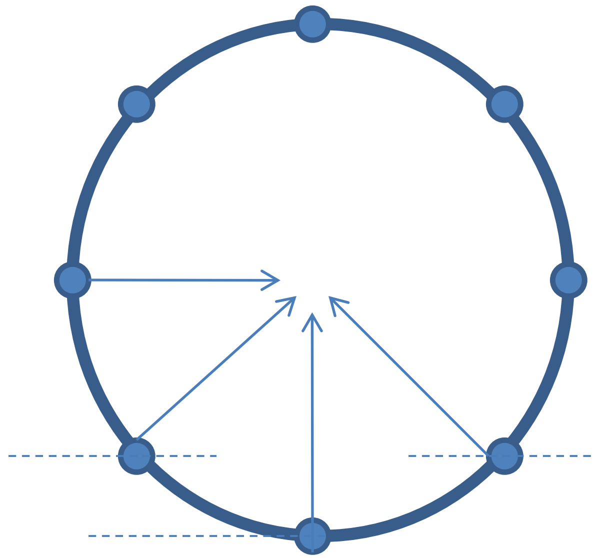
- For each edge point
  - Index into table with its gradient orientation ϑ
  - Use retrieved r vectors to vote for position of reference point
- Peak in this Hough space is reference point with most supporting edges

Assuming translation is the only transformation here, i.e., orientation and scale are fixed.

Credit slide: C. *Grauman*

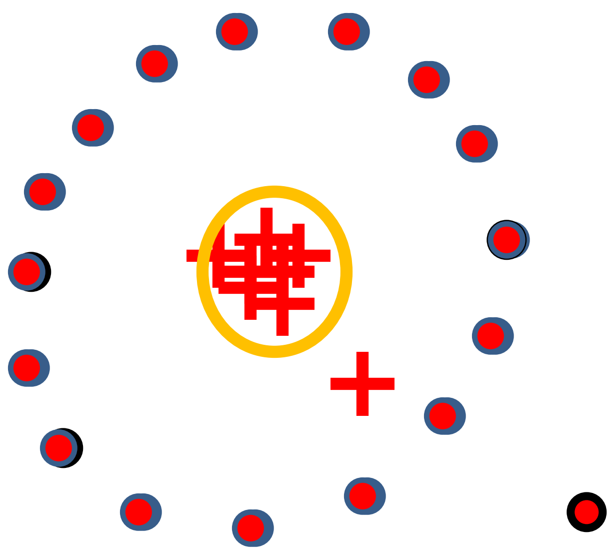
Example

Circle model



θ	rx	ry
0	1	0
45	0.7	0.7
90	0	1
135	-0.7	0.7
…		
270	0.7	-0.7

Query



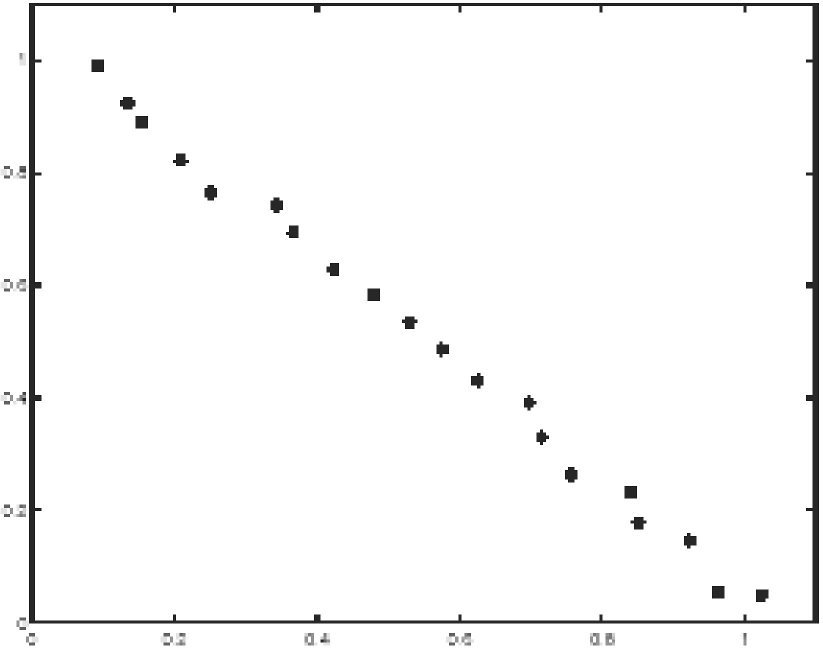
P1 → $\theta = 0$ → R = [rx,ry] = [1,0] → C1 = P1 + R

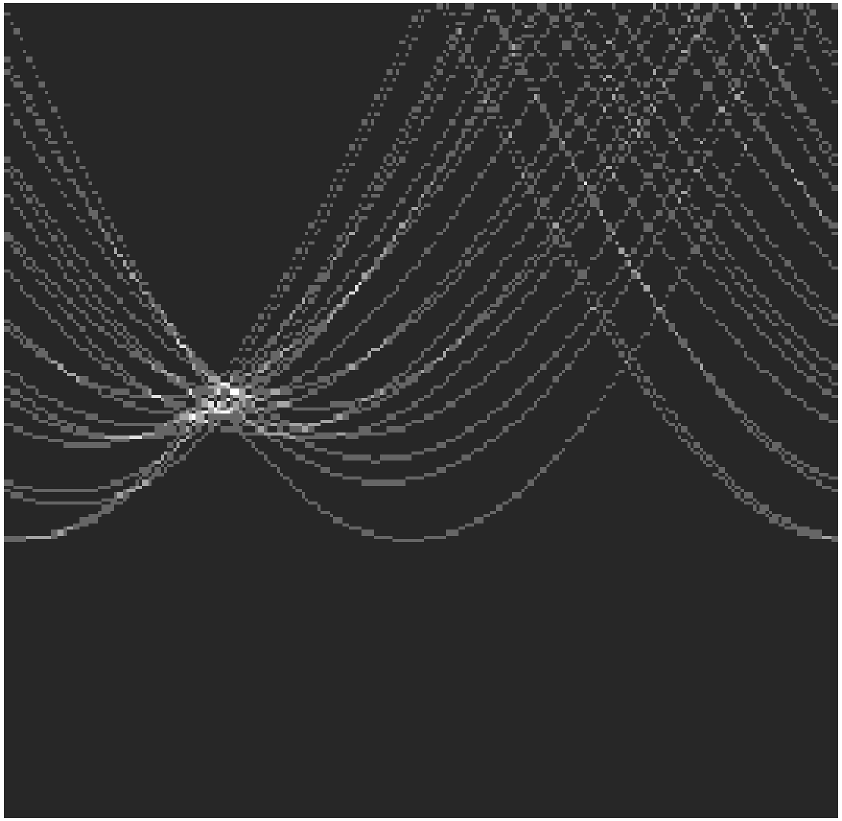
P2 → $\theta = 45$ → R = [rx,ry] = [.7,.7] → C2 = P2 + R

Pk → $\theta = -180$ → R = [rx,ry] = [-1, 0] → Ck = Pk + R

⋮

Conceptually similar to

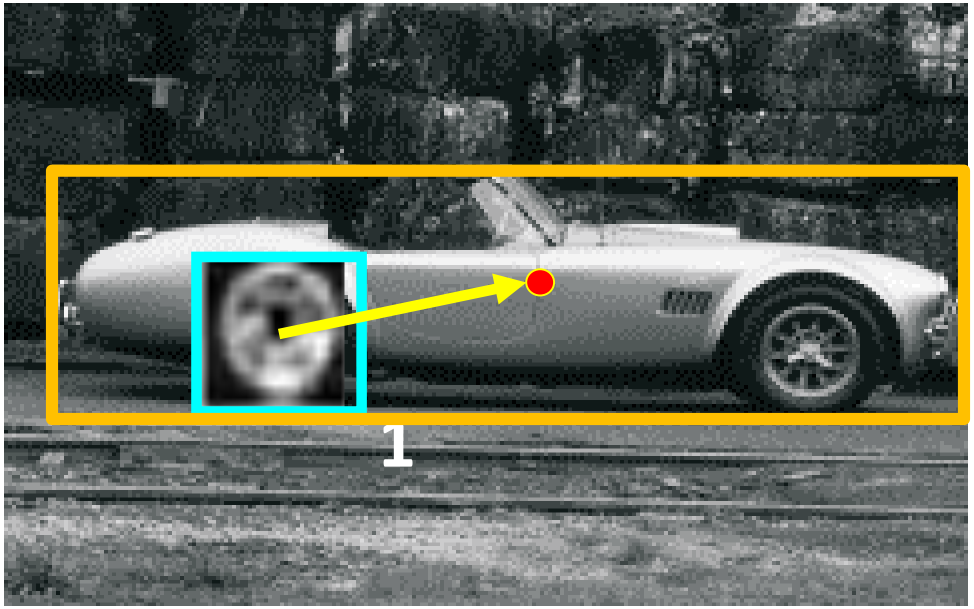




Implicit shape models

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

→ Visual codebook is used to index votes for object position [center] and scale



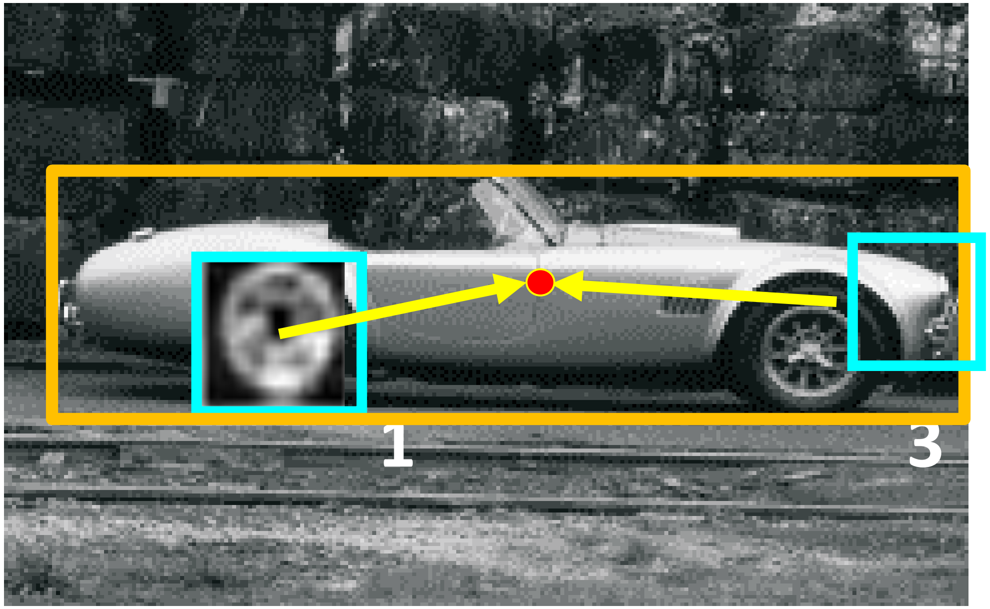
CW	rx	ry
1	0.9	0.1

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

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

→ Visual codebook is used to index votes for object position [center] and scale



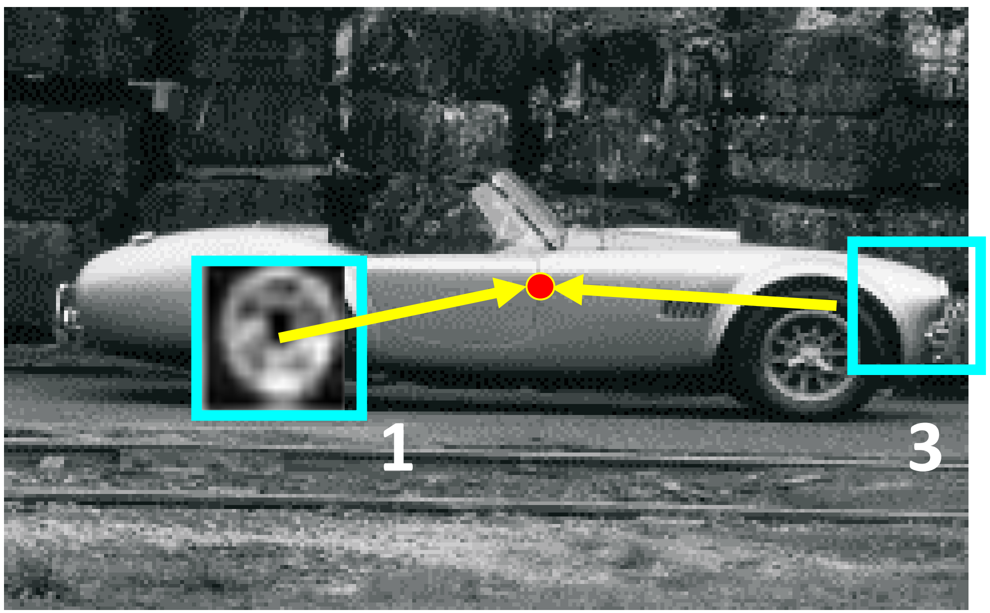
CW	rx	ry
1	0.9	.1
3	?	?

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

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

→ Visual codebook is used to index votes for object position [center] and scale

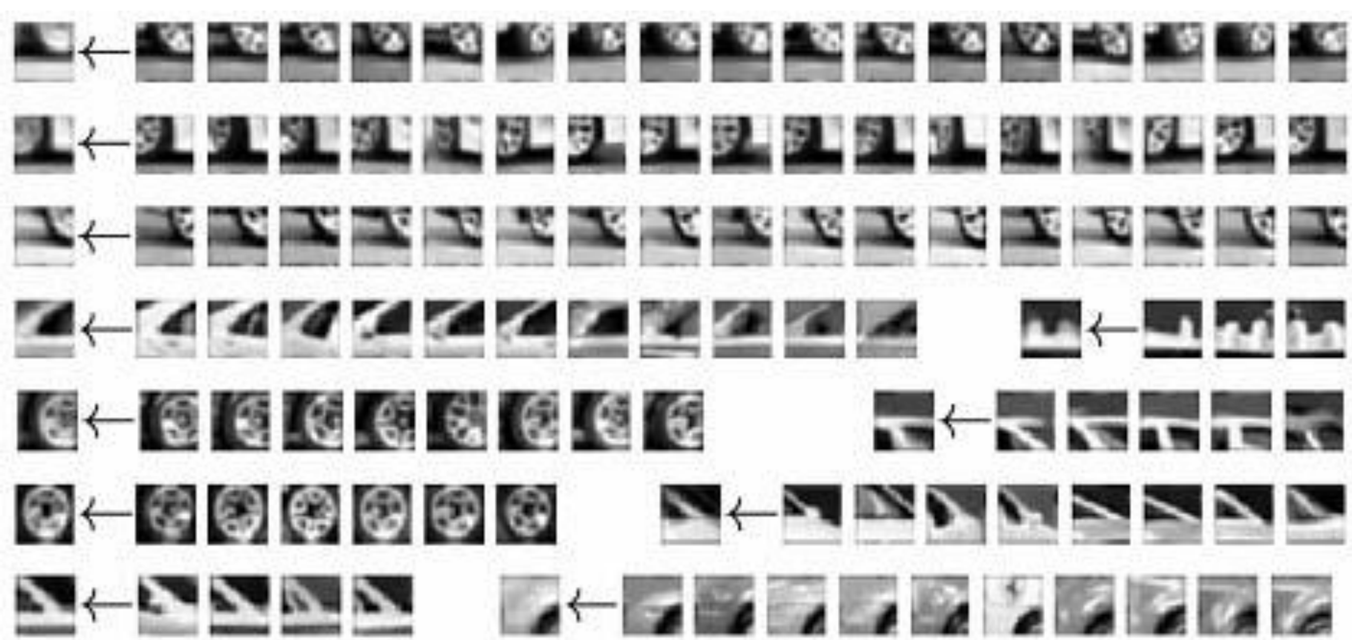


CW	rx	ry
1	0.9	.1
3	-1	0
…	…	…
N	0.7	-0.7

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

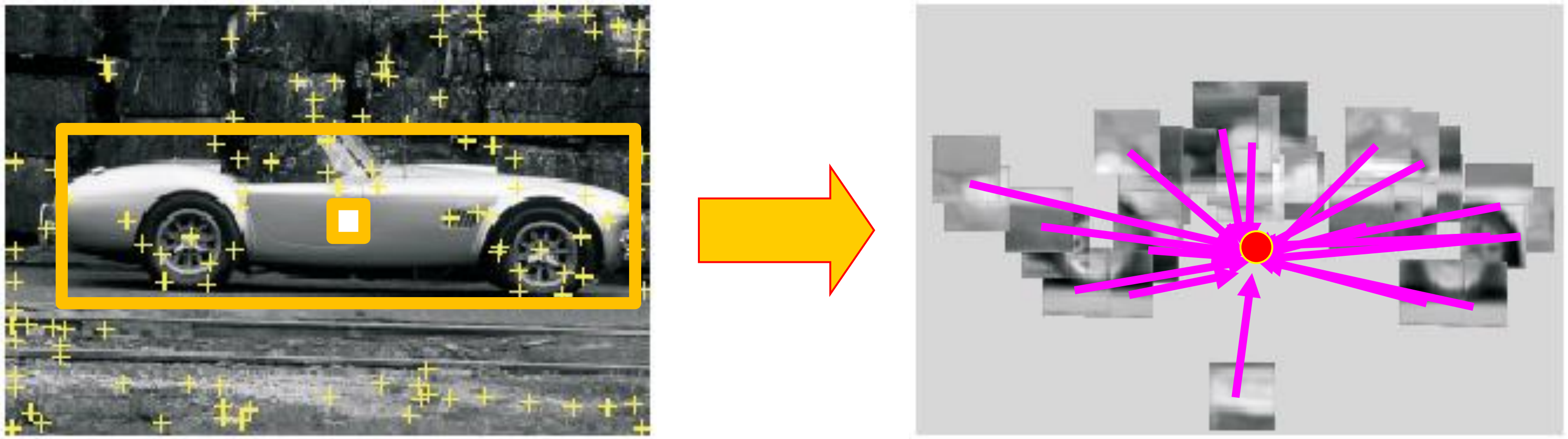
1. Build codebook of patches around extracted interest points using clustering



Credit slide: S. Lazebnik

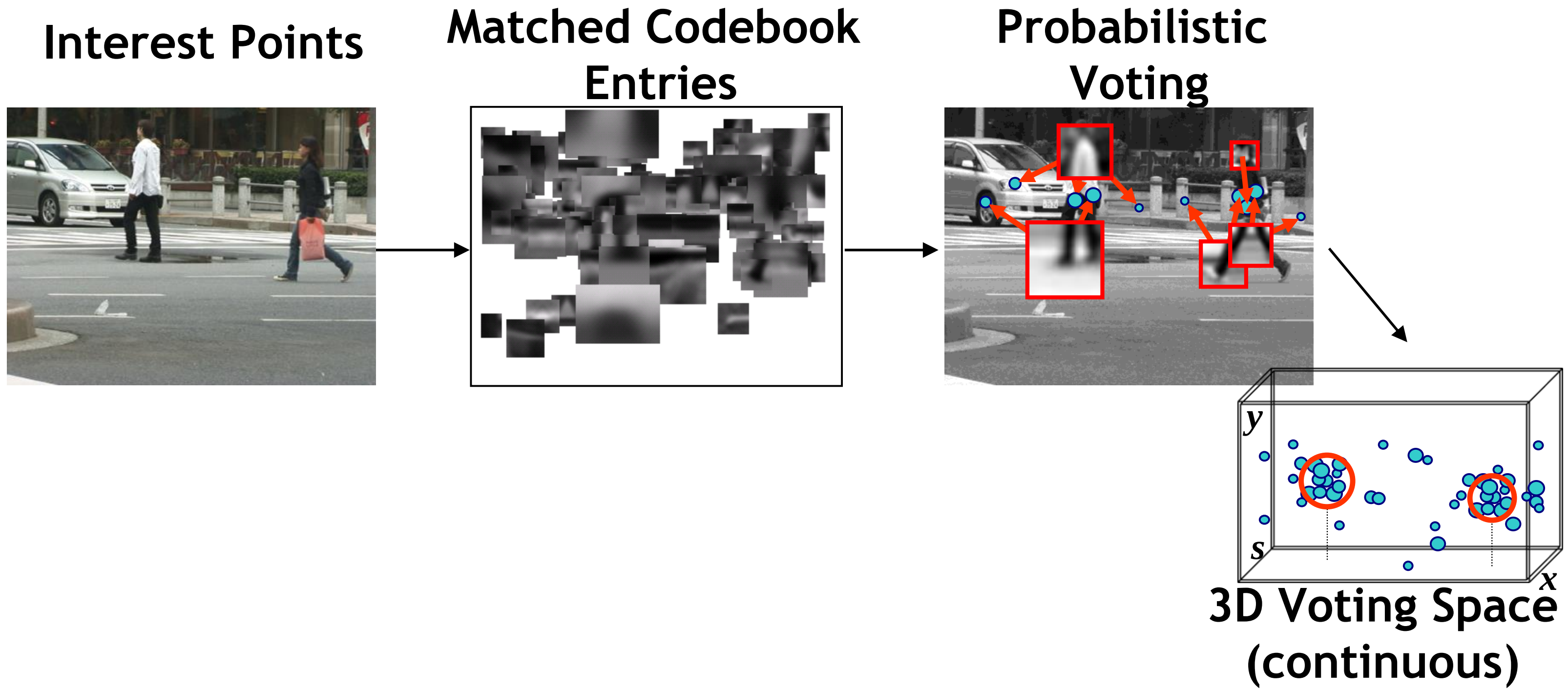
Implicit shape models: Training

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 relative to object center [center is given] and scale [bounding box is given]



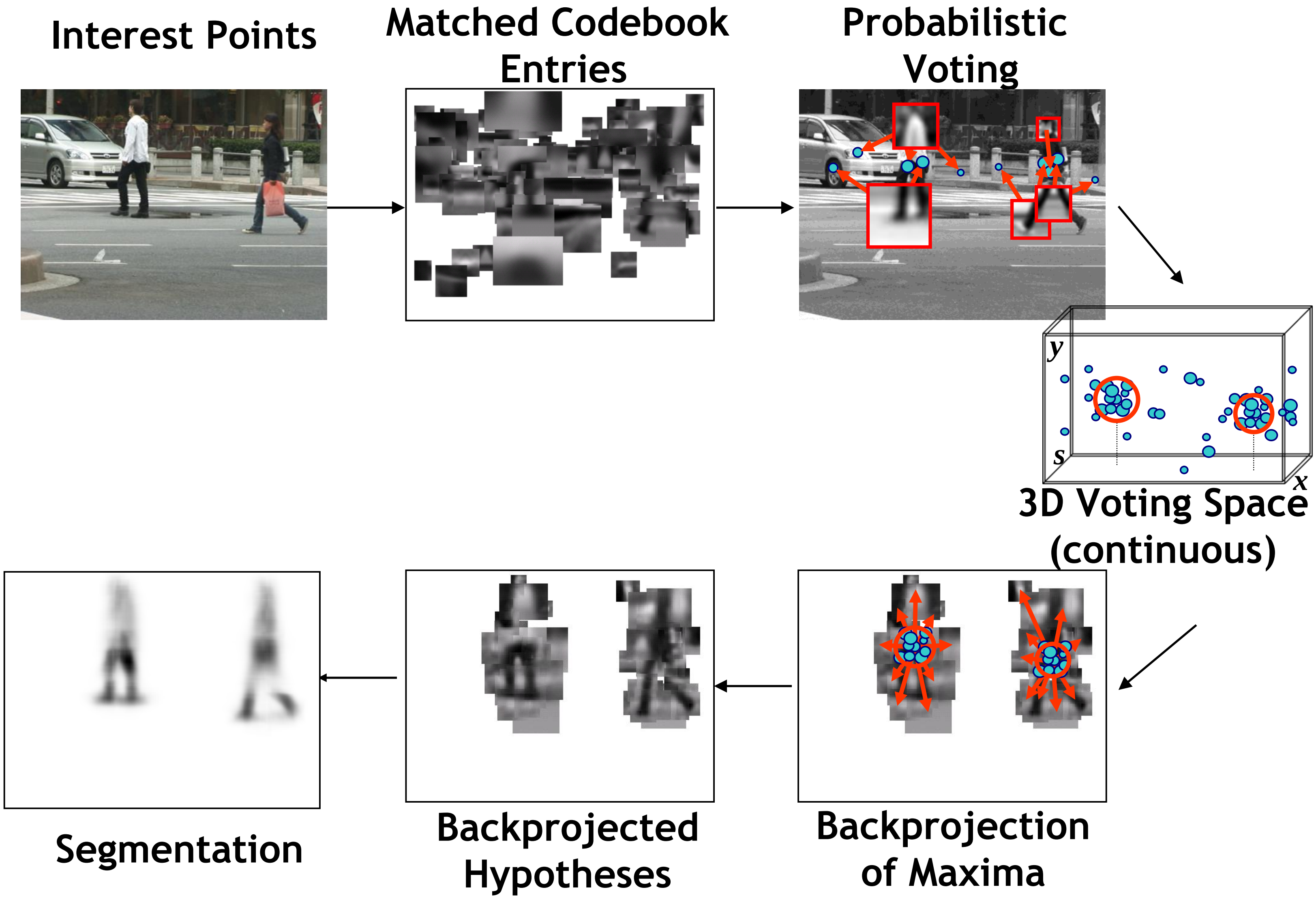
Credit slide: S. Lazebnik

Implicit Shape Model - Recognition



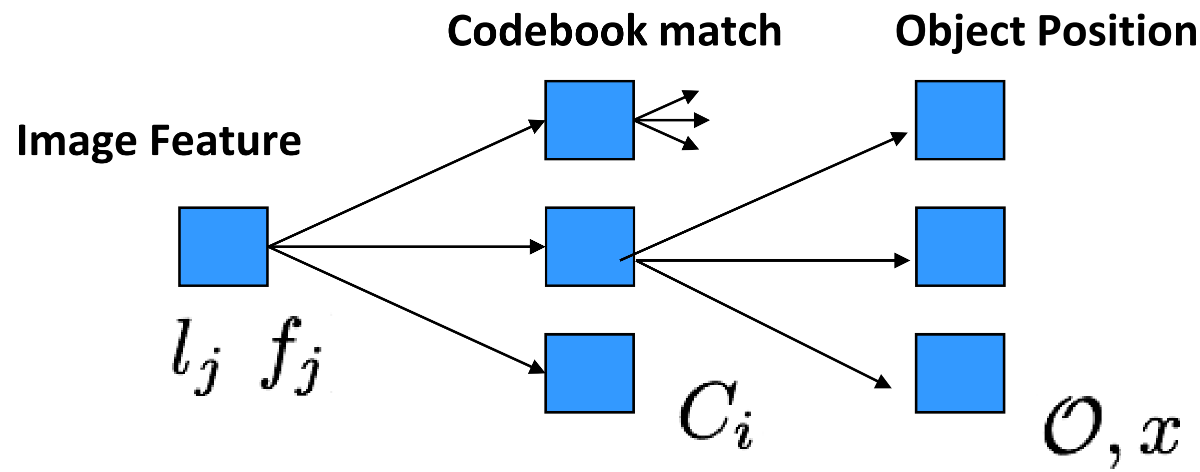
[Leibe, Leonardis, Schiele, SLCV'04; IJCV'08]

Implicit Shape Model - Recognition



[Leibe, Leonardis, Schiele, SLCV'04; IJCV'08]

Probabilistic Hough Transform



f = features
l = location of the features.
C = codebook entry
O = object class
x = object center

Learnt using a max margin formulation

Maji et al , CVPR 2009

$$S(\mathcal{O}, x) \propto \sum_{i,j} p(x, \mathcal{O}, C_i, l_j, f_j)$$

$$\propto \sum_{i,j} p(x|\mathcal{O}, C_i, l_j) p(C_i|f_j) p(\mathcal{O}|C_i, l_j)$$

Detection Score

Position Posterior
distribution of the centroid given the Codeword Ci observed at location lj.

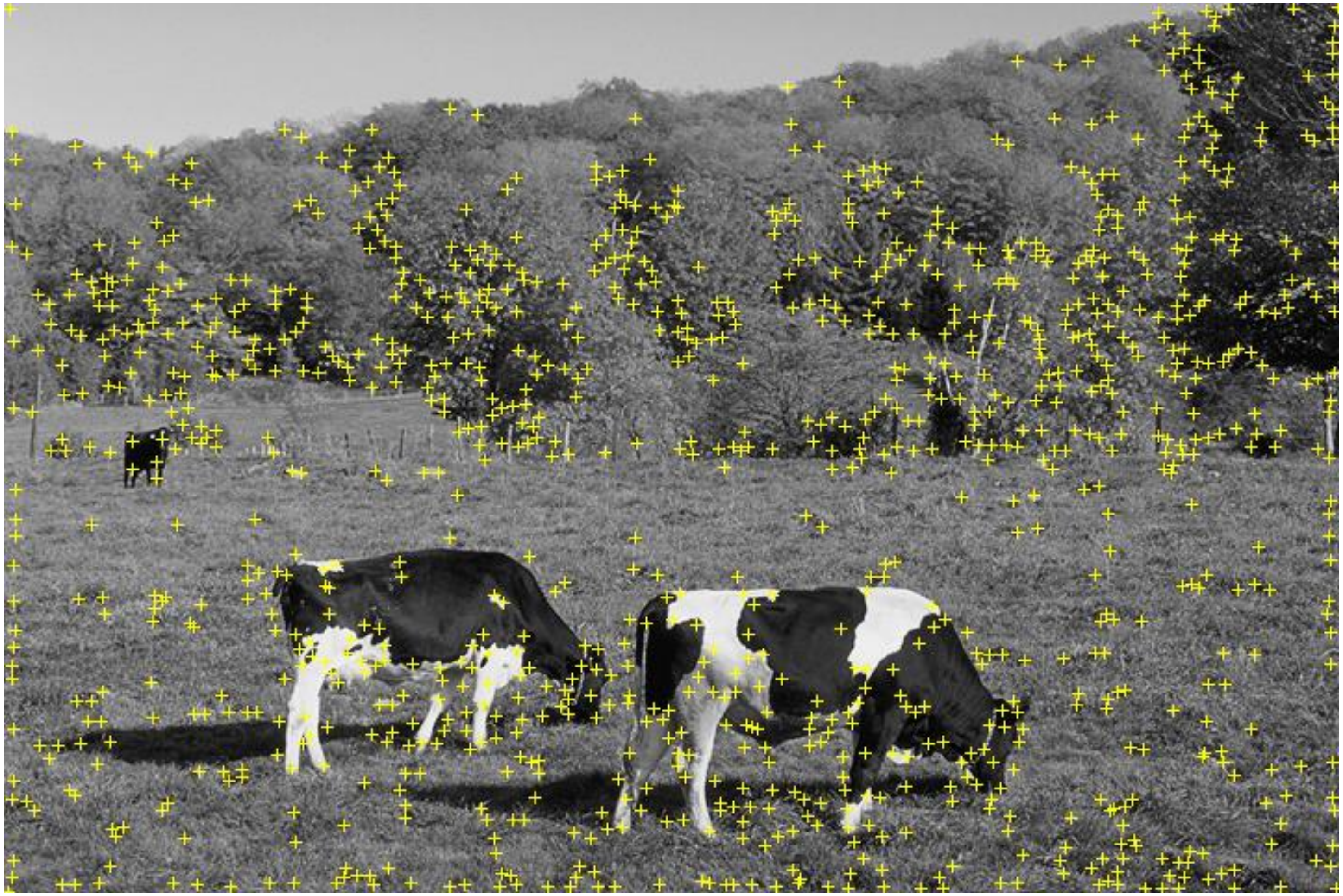
Codeword Match

confidence (or weight) of the codeword Ci.

Example: Results on Cows

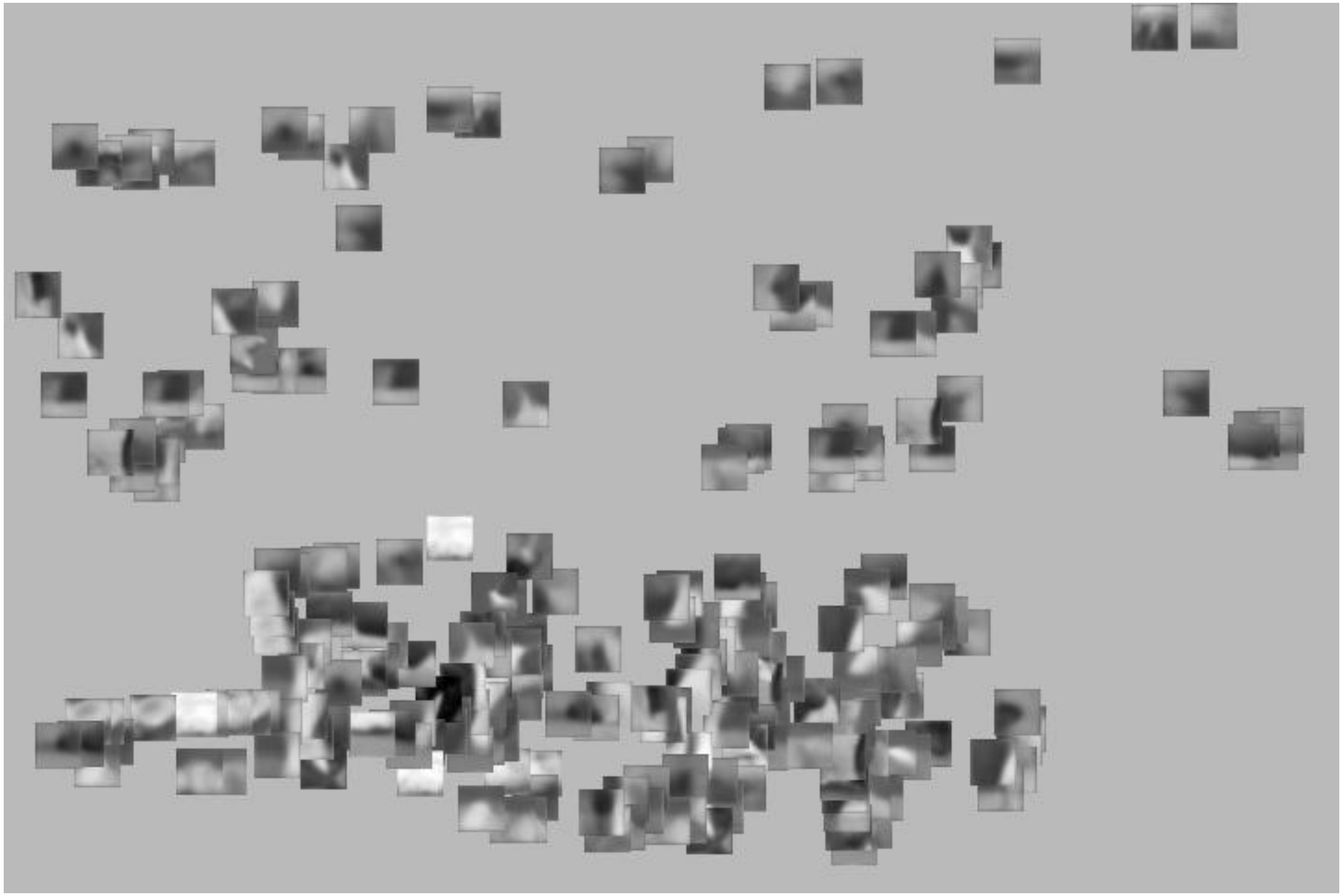
Original image

Example: Results on Cows



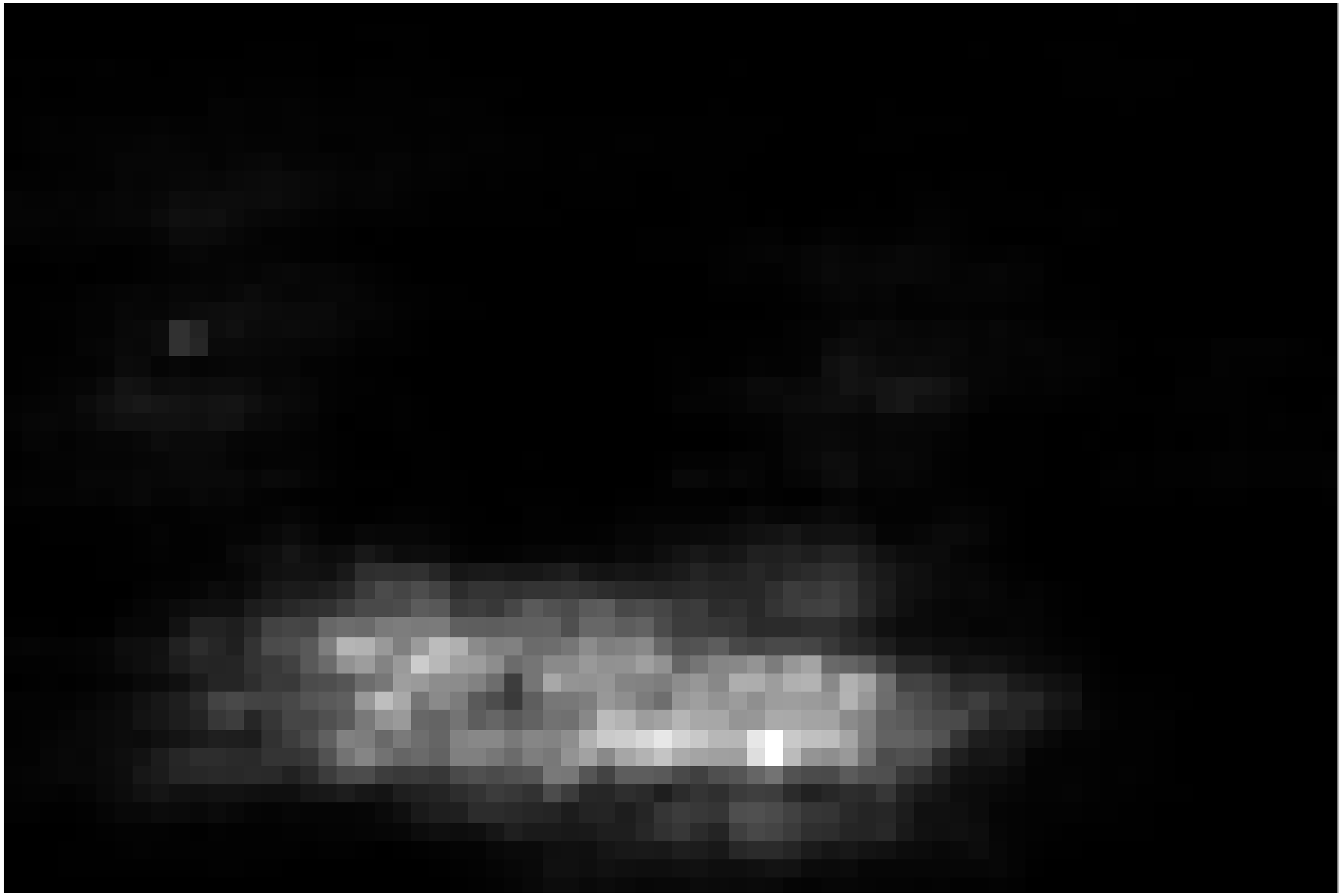
Interest points

Example: Results on Cows



Matched patches

Example: Results on Cows



Prob. Votes

Example: Results on Cows

1st hypothesis

Example: Results on Cows

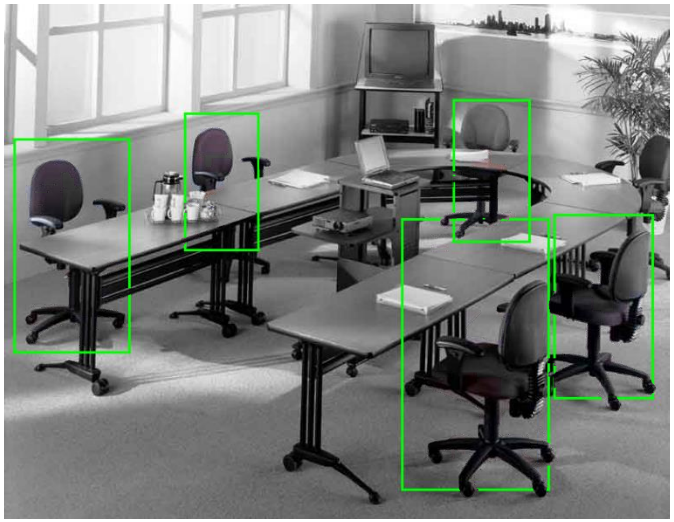
2nd hypothesis

Example: Results on Cows

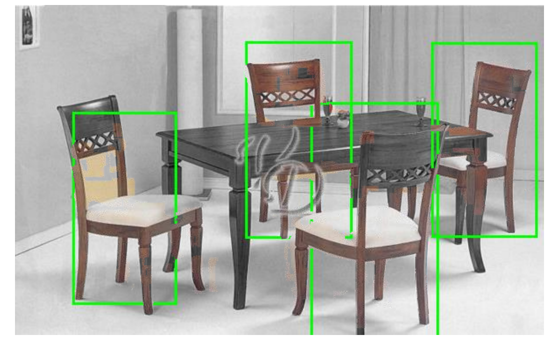
3rd hypothesis

Example Results: Chairs

Dining room chairs



Office chairs



You Can Try It At Home…

- **Linux binaries available**
  - **Including datasets & several pre-trained detectors**
  - **http://www.vision.ee.ethz.ch/bleibe/code**

Conclusions

- Pros:
  - Works well for many different object categories
    - Both rigid and articulated objects
  - Flexible geometric model
    - Can recombine parts seen on different training examples
  - Learning from relatively few (50-100) training examples
  - Optimized for detection, good localization properties

- Cons:
  - Needs supervised training data
    - Object bounding boxes for detection
    - Segmentations for top-down segmentation
  - No discriminative learning

Influential Works in Detection

- Sung-Poggio (1994, 1998) : ~2000 citations
  - Basic idea of statistical template detection, bootstrapping to get "face-like" negative examples, multiple whole-face prototypes (in 1994)
- Rowley-Baluja-Kanade (1996-1998) : ~3600
  - "Parts" at fixed position, non-maxima suppression, simple cascade, rotation, pretty good accuracy, fast
- Schneiderman-Kanade (1998-2000,2004) : ~1700
  - Careful feature engineering, excellent results, cascade
- Viola-Jones (2001, 2004) : ~11,000
  - Haar-like features, Adaboost as feature selection, hyper-cascade, very fast, easy to implement
- Dalal-Triggs (2005) : ~6500
  - Careful feature engineering, excellent results, HOG feature, online code
- Felzenszwalb-Huttenlocher (2000): ~2100
  - Efficient way to solve part-based detectors
- Weber et al. (2000)
  - Part-based model learnt in a unsupervised fashion; generative
- Felzenszwalb-McAllester-Ramanan (2008): ~1300
  - Excellent template/parts-based blend
- Leibe et al. (2005)
  - Generative approach to detection using hough voting

Courtesy of J Hayes

Next lecture

- 3D Object Detection