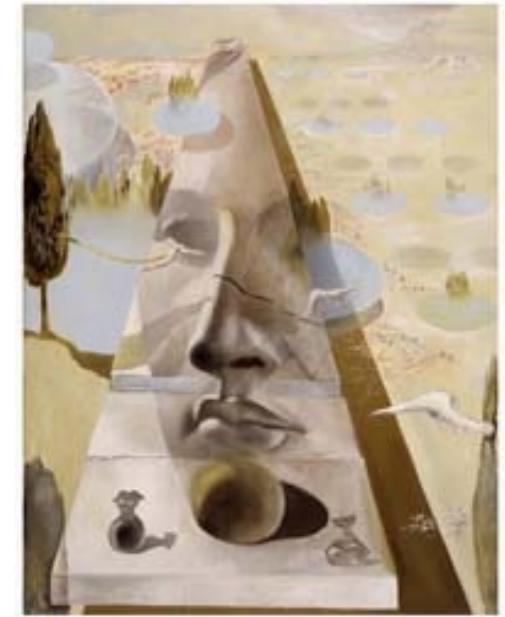


Lecture 12

Visual recognition



- Object Classification – BoW models (part 2)
- 2D Object Detection
 - Template based approaches
 - Part-based approaches

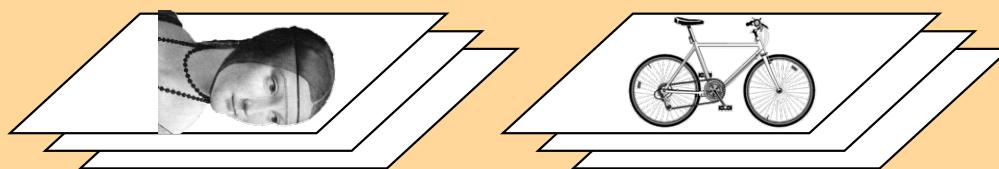
Object

Bag of ‘words’



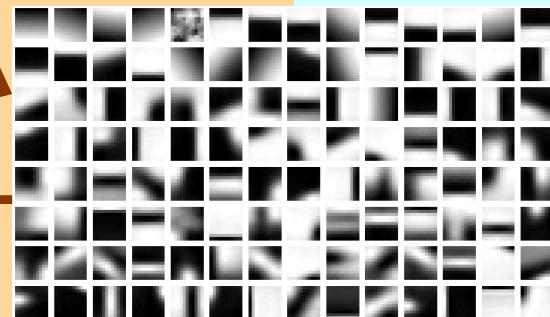
Representation

recognition



1. feature detection
& representation

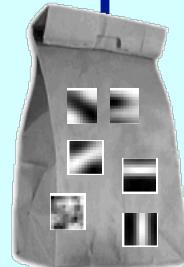
2. codewords dictionary



3. image representation



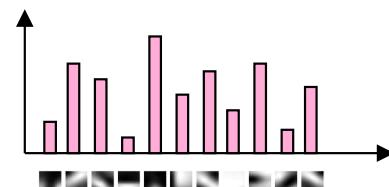
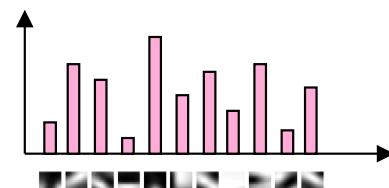
**category models
(and/or) classifiers**



**category
decision**

learning

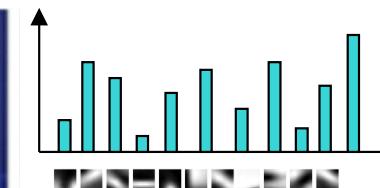
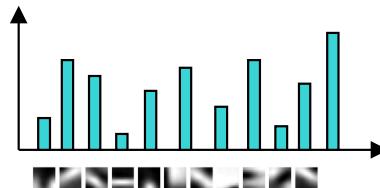
Category models



...



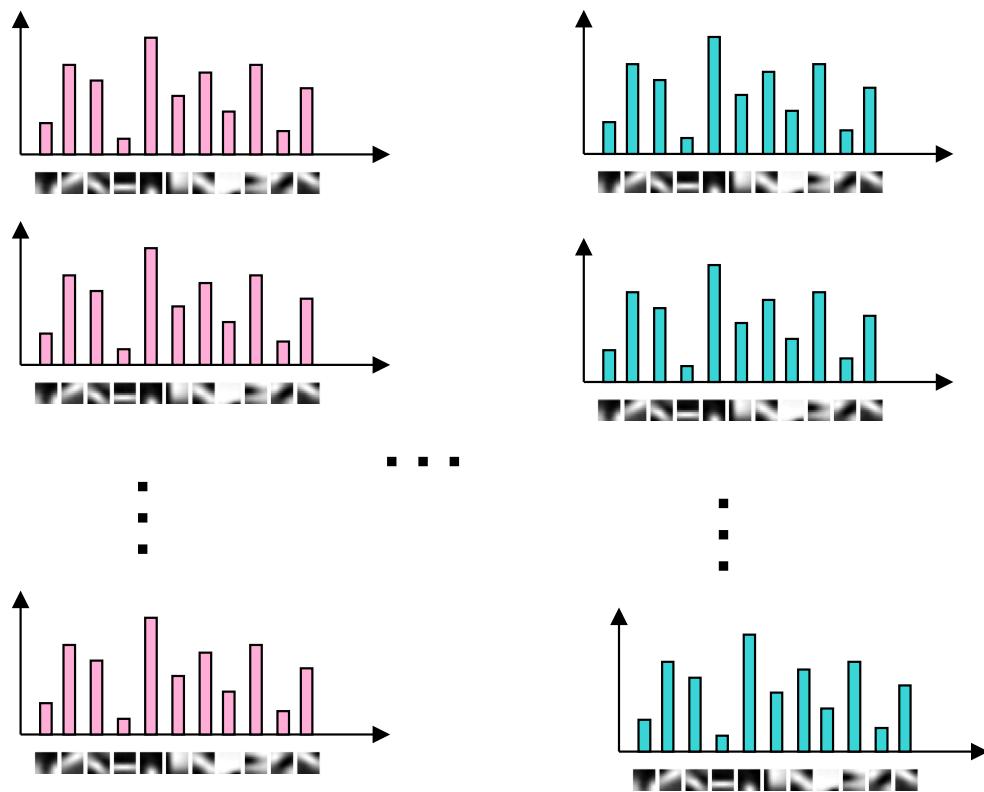
Class 1



Class N

Discriminative classifiers

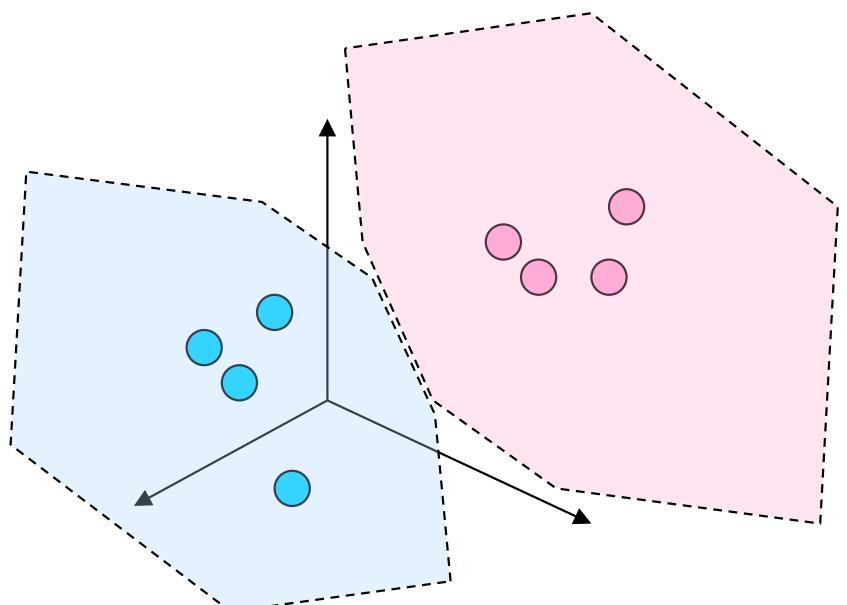
category models



Class 1

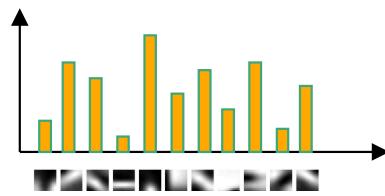
Class N

model space



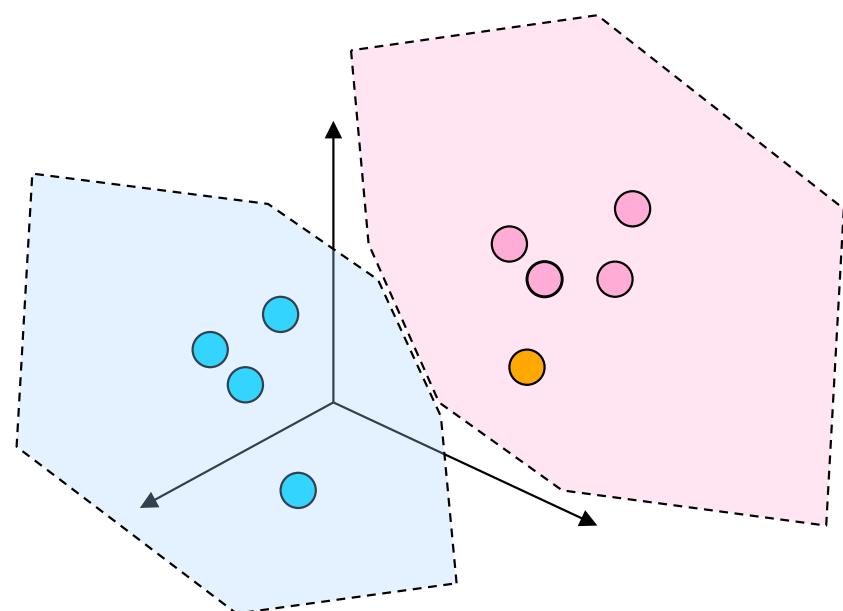
Discriminative classifiers

Query image



Winning class: pink

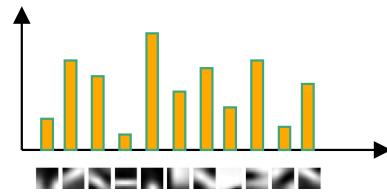
model space



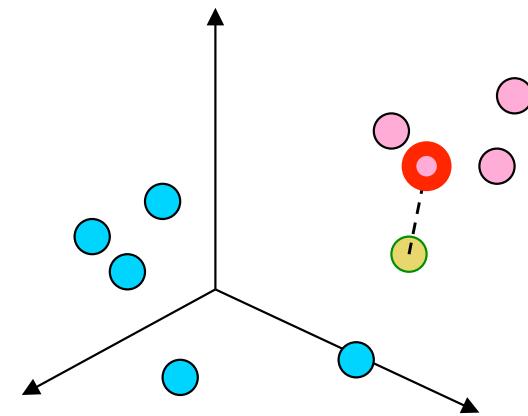
- Nearest neighbors
- Linear classifier
- SVM

Nearest neighbors classifiers

Query image



model space

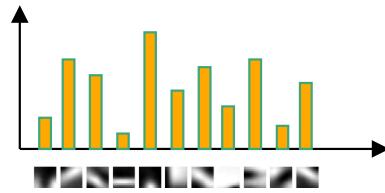


Winning class: pink

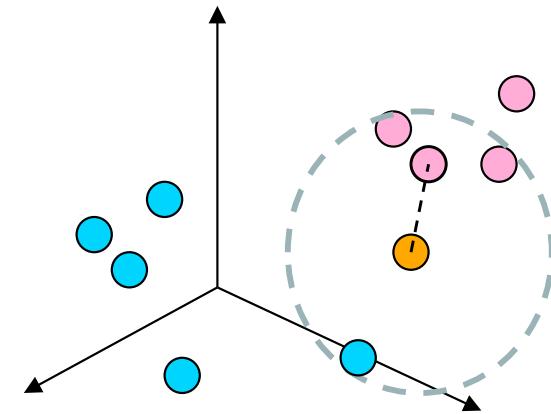
- Assign label of nearest training data point to each test data point

K-nearest neighbors classifiers

Query image



model space

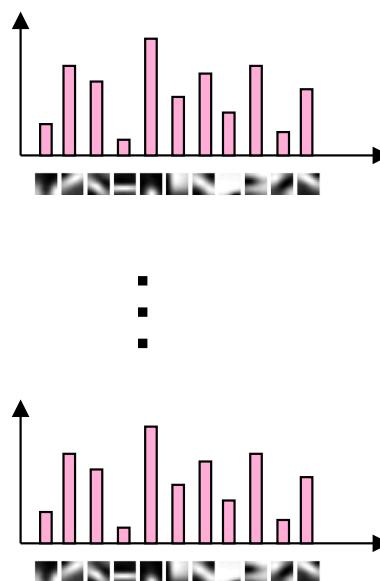


Winning class: pink

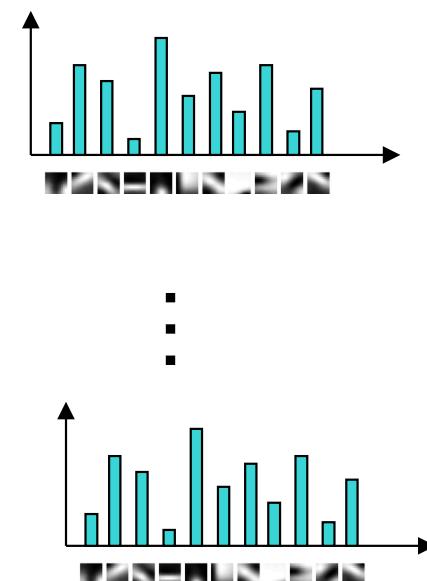
- For a new query histogram, find the k closest points from training data
- Query histogram is classified by a majority vote of its K nearest neighbors
- Works well provided there is lots of data

Linear classifiers

category models

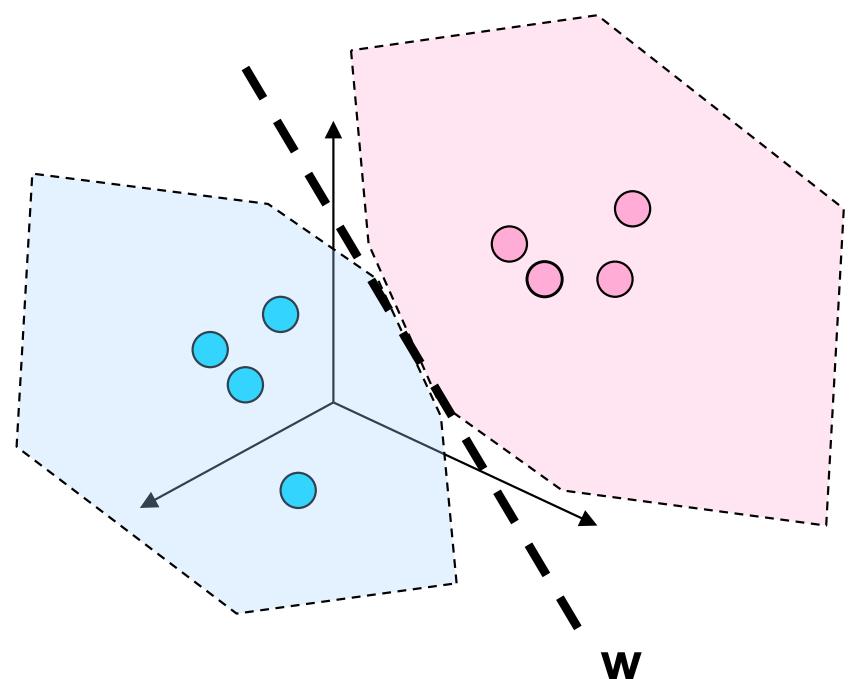


Class 1



Class N

Model space

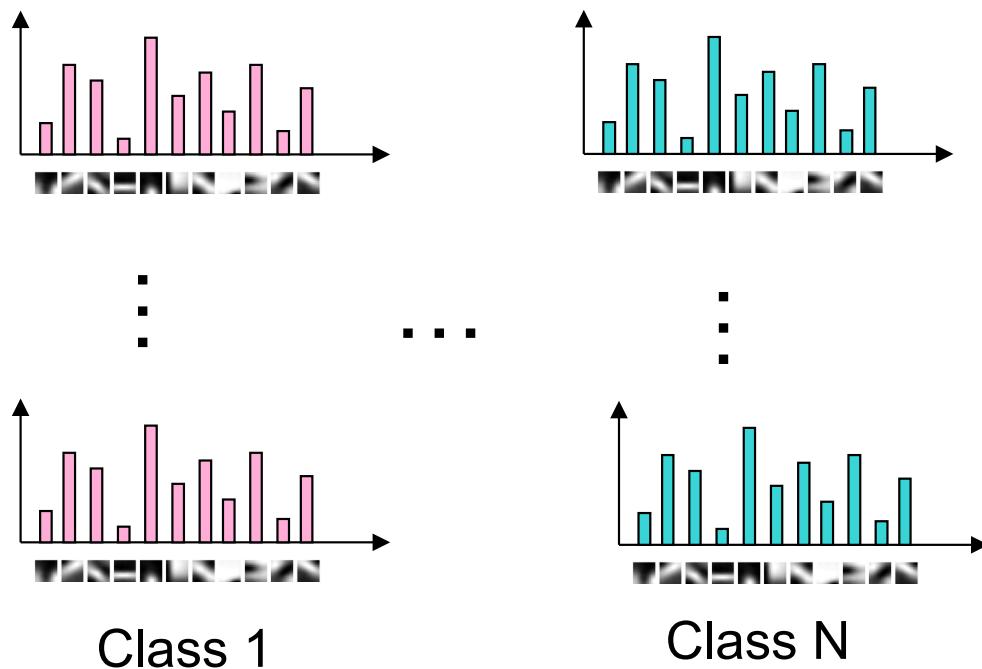


w

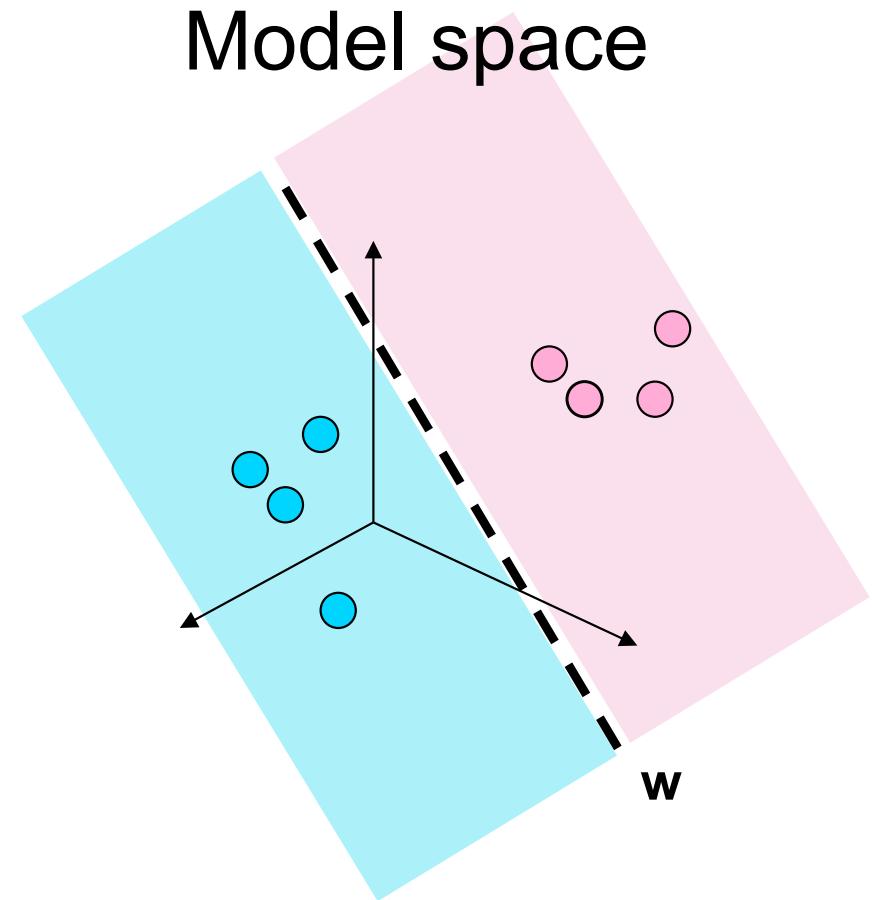
- Instead of modeling decision regions, it estimates a hyperplane w that separates training points that belong to different classes

Linear classifiers

category models



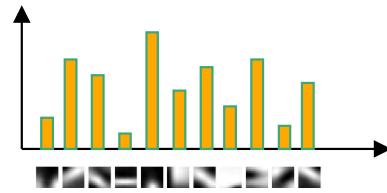
Model space



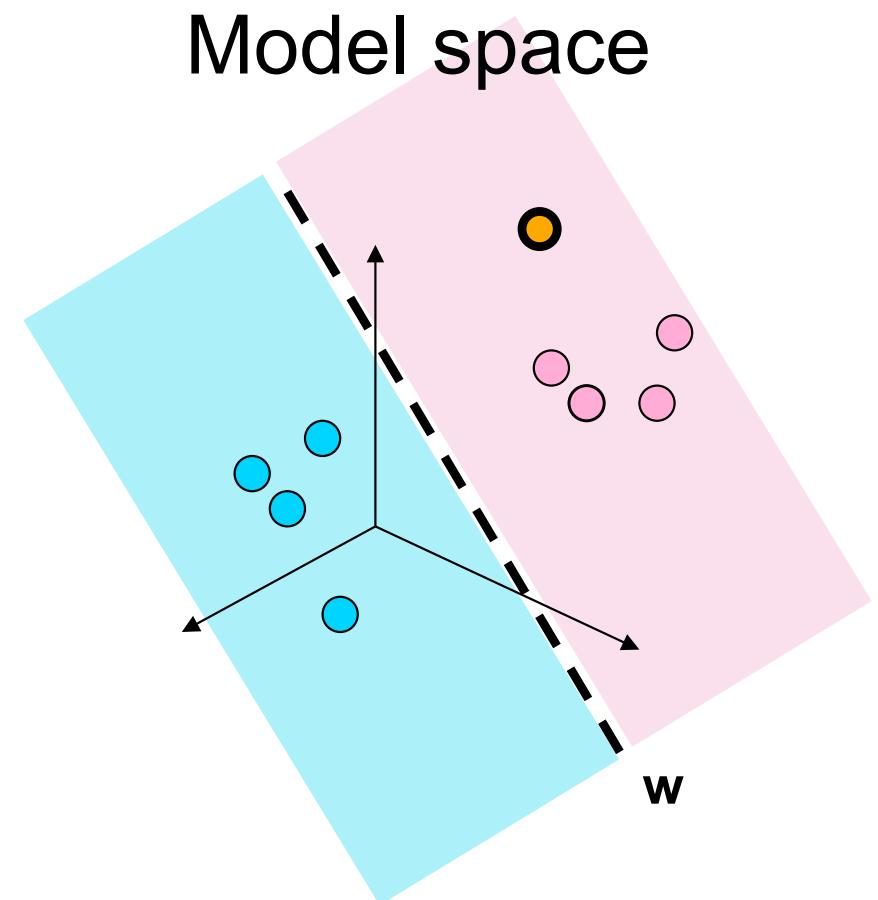
- Instead of modeling decision regions, it estimates a hyperplane w that separates training points that belong to different classes
- The training data is used to learn parameters of w and then discarded

Linear classifiers

Query image



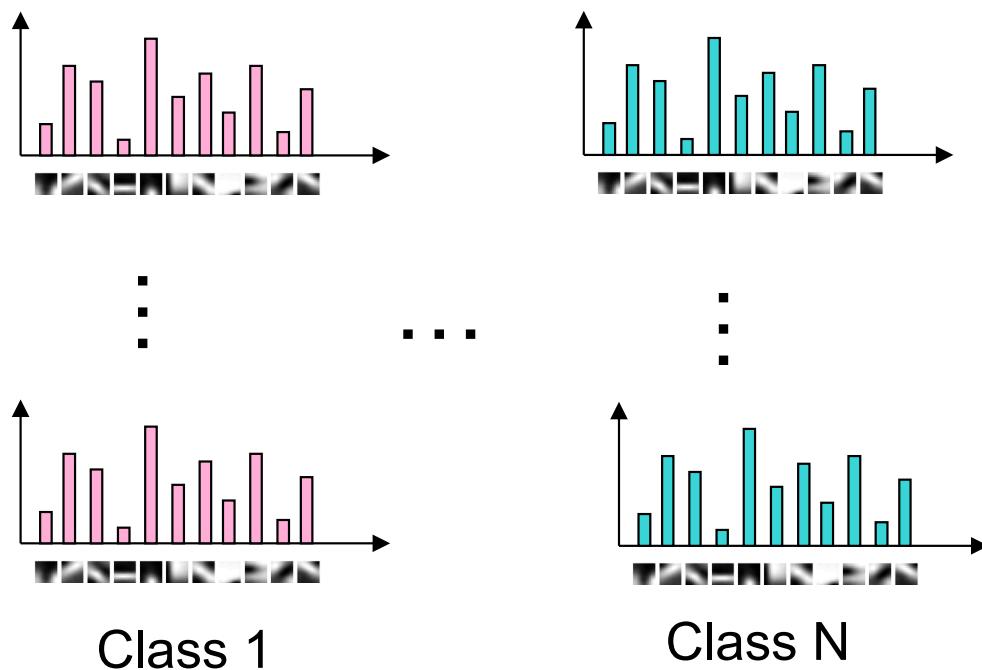
Winning class: pink



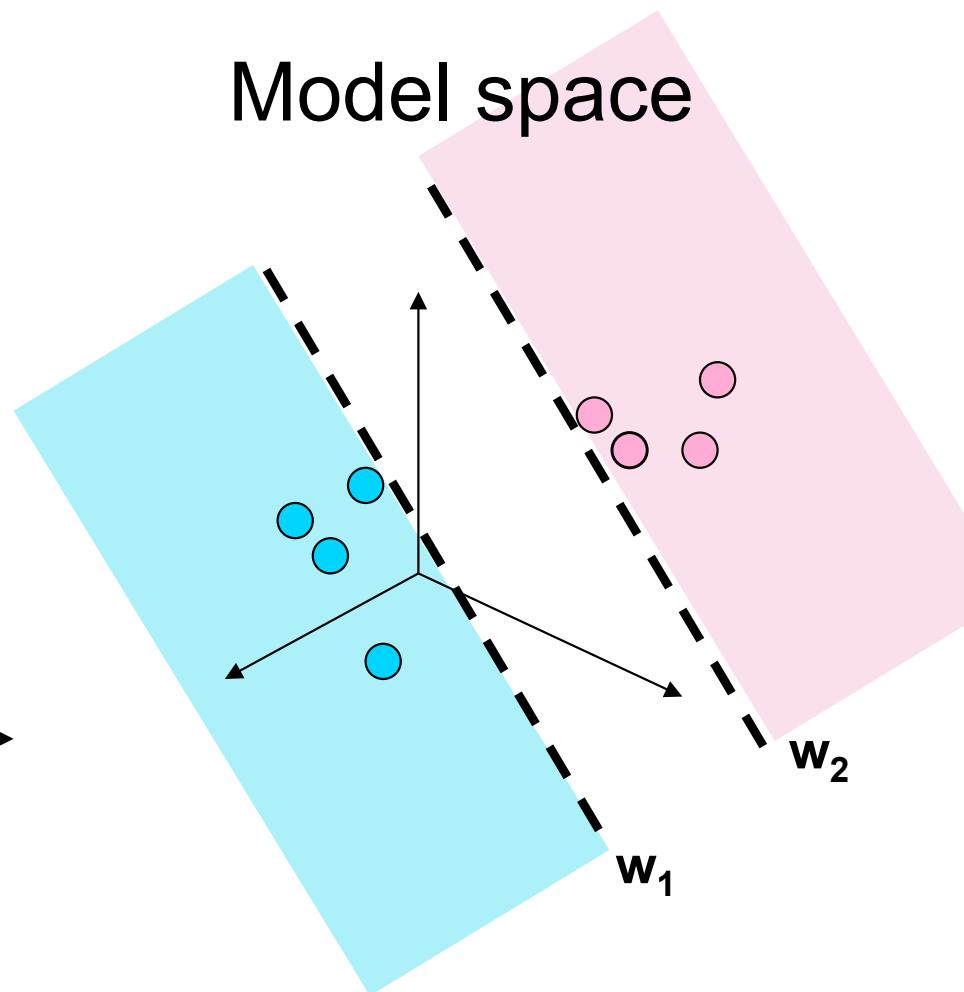
- In training, we only need w for classifying new data!

SVM classifiers

category models



Model space

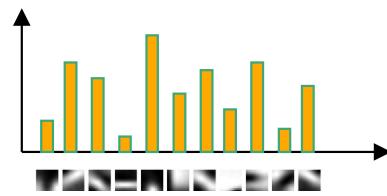


Select two hyperplanes such:

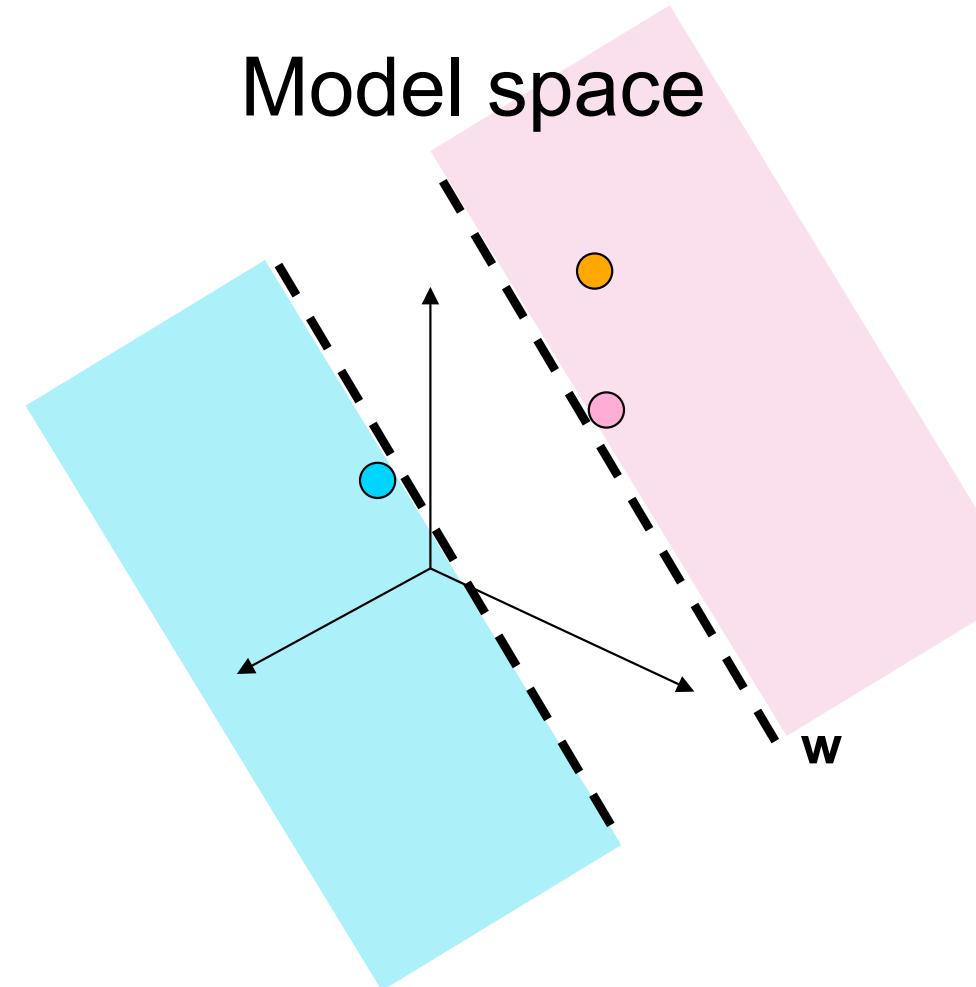
1. They separate the training points
2. There are no points between them
3. Their distance is maximized

SVM classifiers

Query image



Winning class: pink



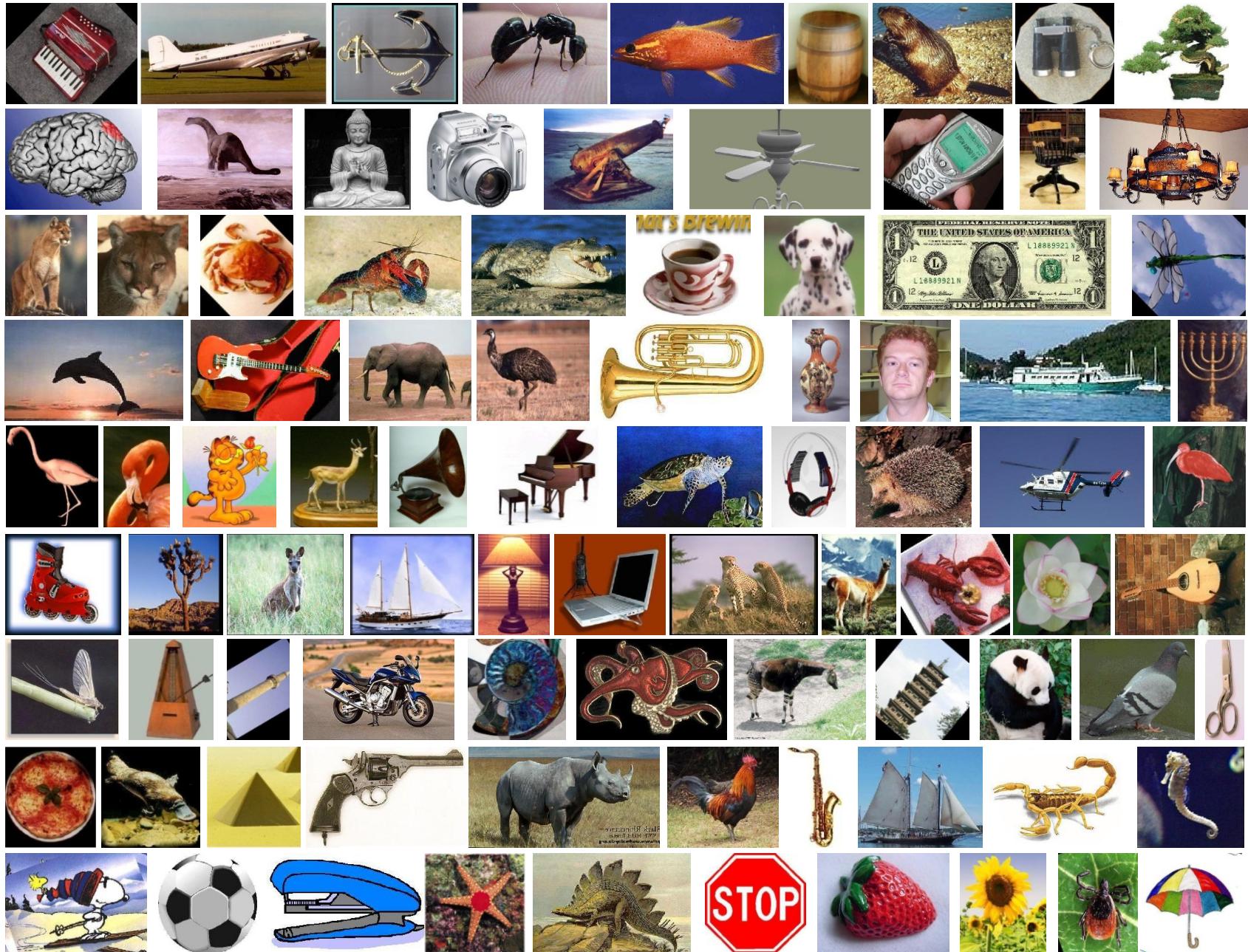
SVMs: Pros and cons

- Pros
 - Many publicly available SVM packages:
<http://www.kernel-machines.org/software>
 - Kernel-based framework is very powerful, flexible
 - SVMs work very well in practice, even with very small training sample sizes (unlike neural networks or CNNs)
- Cons
 - Computation, memory
 - Learning can take a very long time for large-scale problems
 - No “direct” multi-class SVM, must combine two-class SVMs

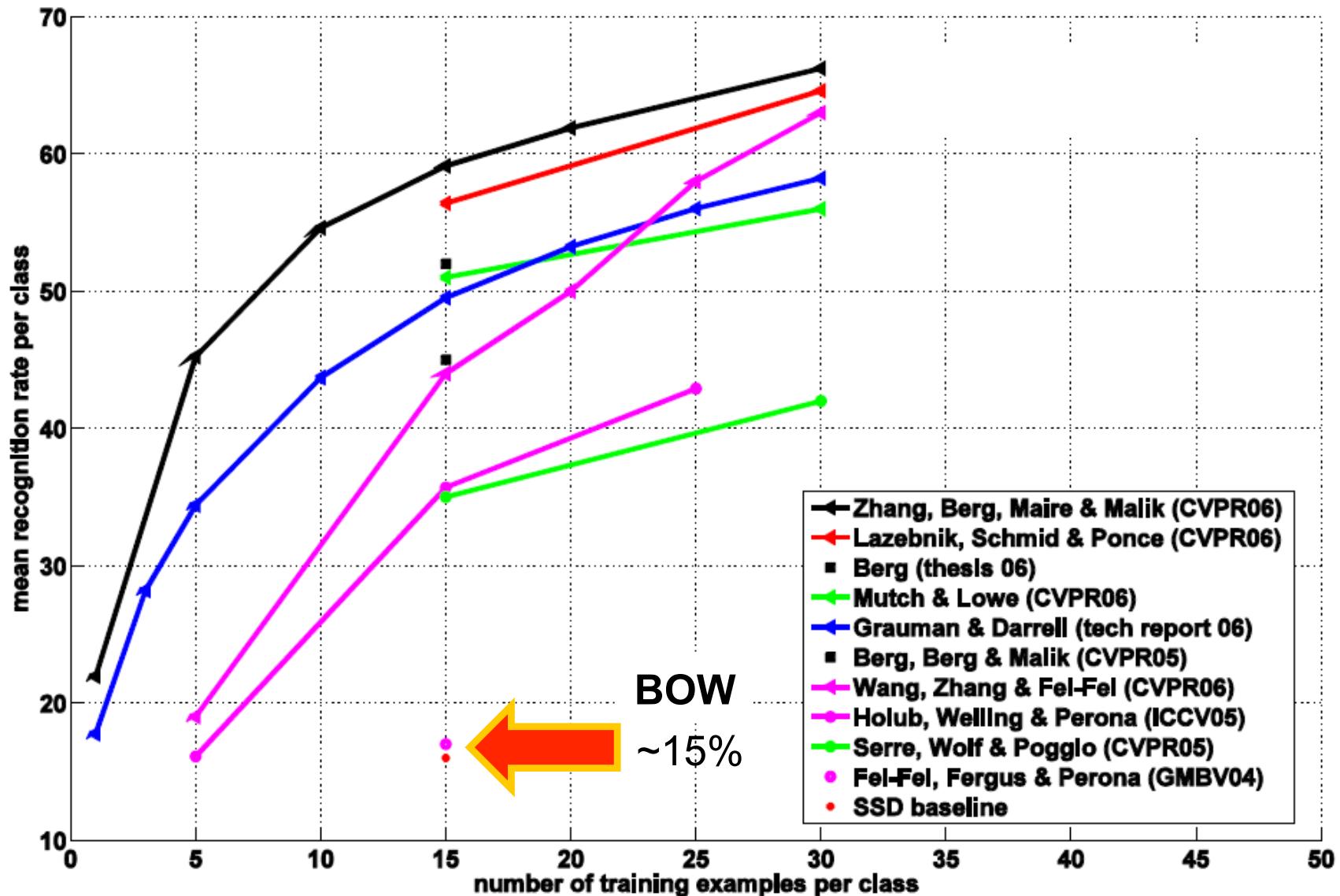
How to design multi-class SVMs?

- A multi-class SVM can be obtained by combining multiple two-class SVMs
- One vs. others
 - Training: learn an SVM for each class vs. the others
 - Testing: apply each SVM to test example and assign to it the class of the SVM that returns the highest decision value
- One vs. one
 - Training: learn an SVM for each pair of classes
 - Testing: each learned SVM “votes” for a class to assign to the test example

Caltech 101 images



Caltech 101



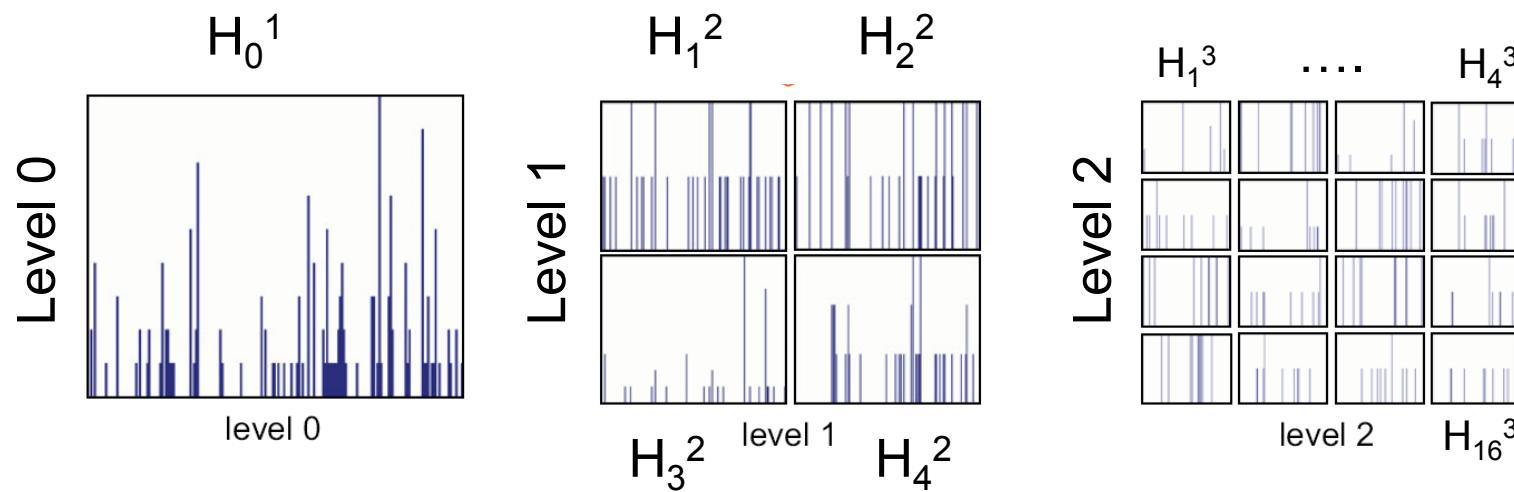
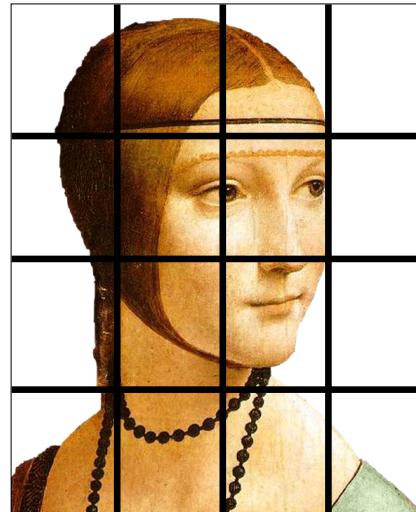
Random classification: 0.01%

Major drawback of BOW models

Don't capture spatial information!

Spatial Pyramid Matching

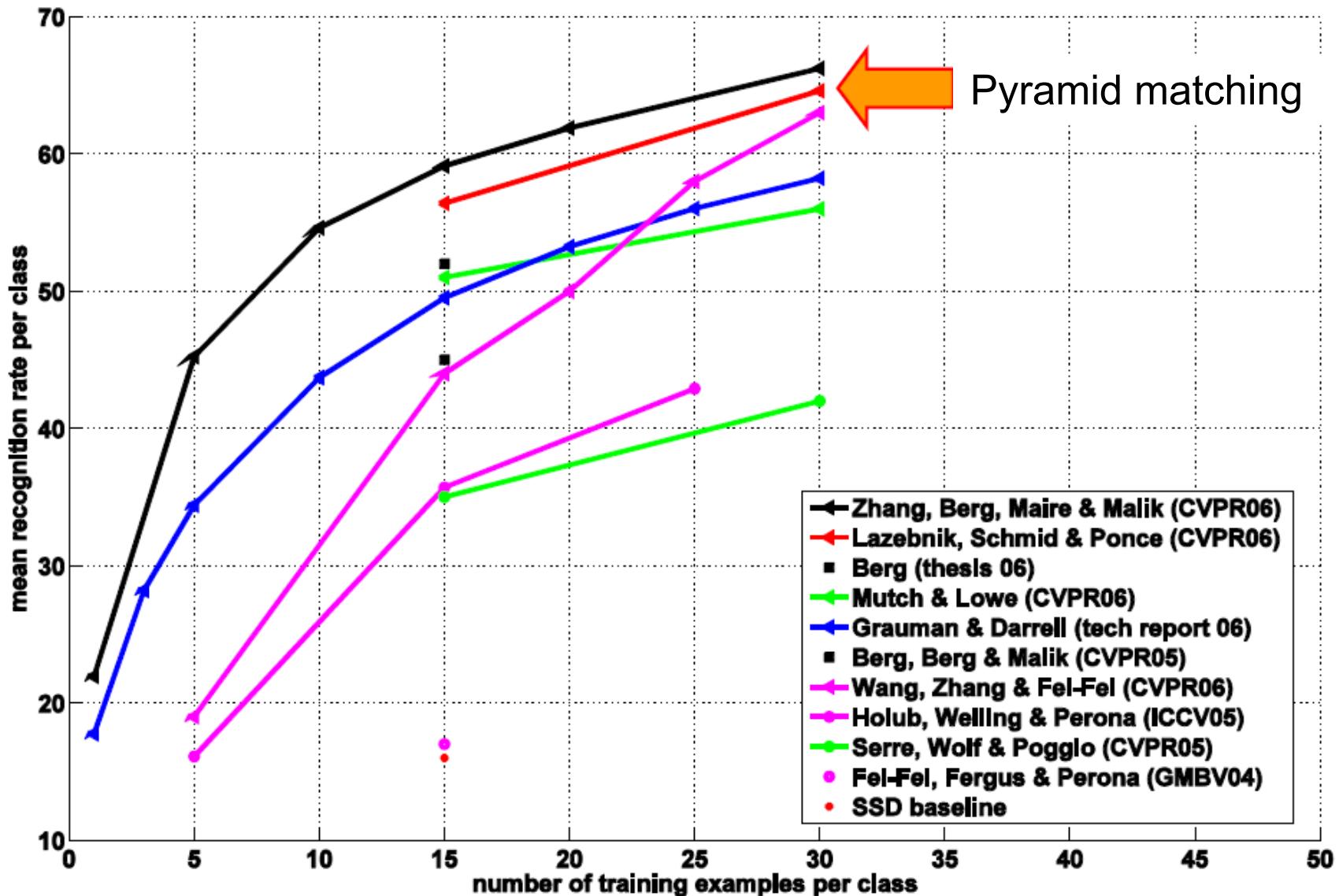
Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories. S. Lazebnik, C. Schmid, and J. Ponce.. 2006



$$H = [H_0^1 \ H_1^2 \ \dots \ H_4^2 \ H_1^3 \ \dots \ H_{16}^3]$$

or, H = combination of H_i^j with appropriate weights

Caltech 101



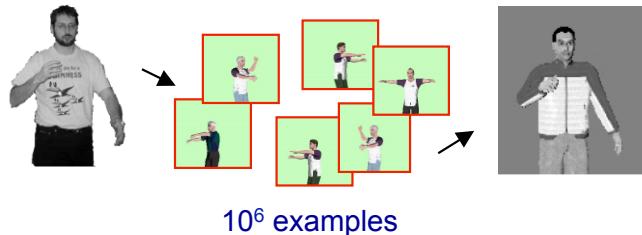
Conclusions

- Pros:
 - Very simple and effective
 - Still used today for image retrieval problems
- Cons:
 - Not ideal for solving detection problems
 - Outperformed by CNNs but require way less training data

Discriminative models

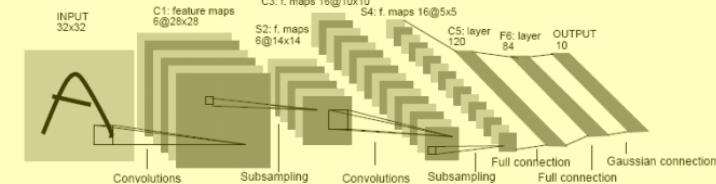
See lecture 15!

Nearest neighbor



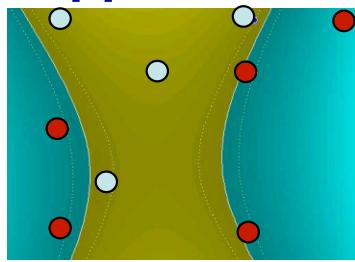
Shakhnarovich, Viola, Darrell 2003
Berg, Berg, Malik 2005...

Neural networks



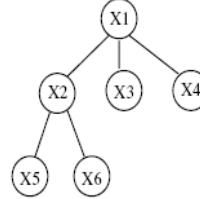
LeCun, Bottou, Bengio, Haffner 1998
Krizhevsky, Sutskever, Hinton, 2012

Support Vector Machines



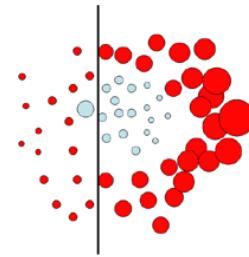
Guyon, Vapnik, Heisele,
Serre, Poggio...

Latent SVM Structural SVM



Felzenszwalb 2000
Ramanan 2003,2008...

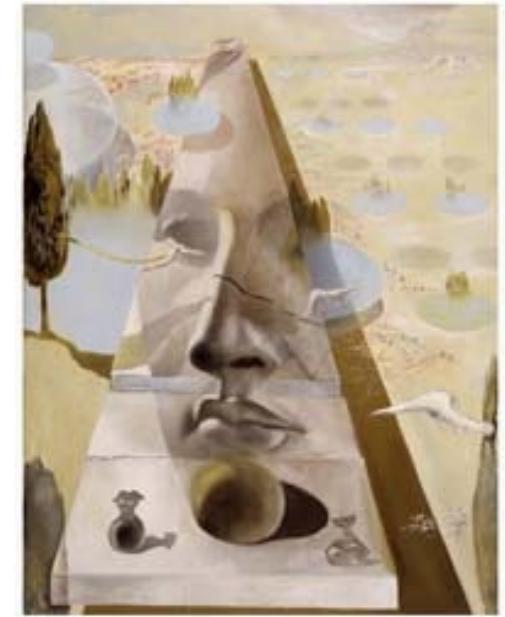
Boosting



Viola, Jones 2001,
Torralba et al. 2004,
Opelt et al. 2006,...

Lecture 14

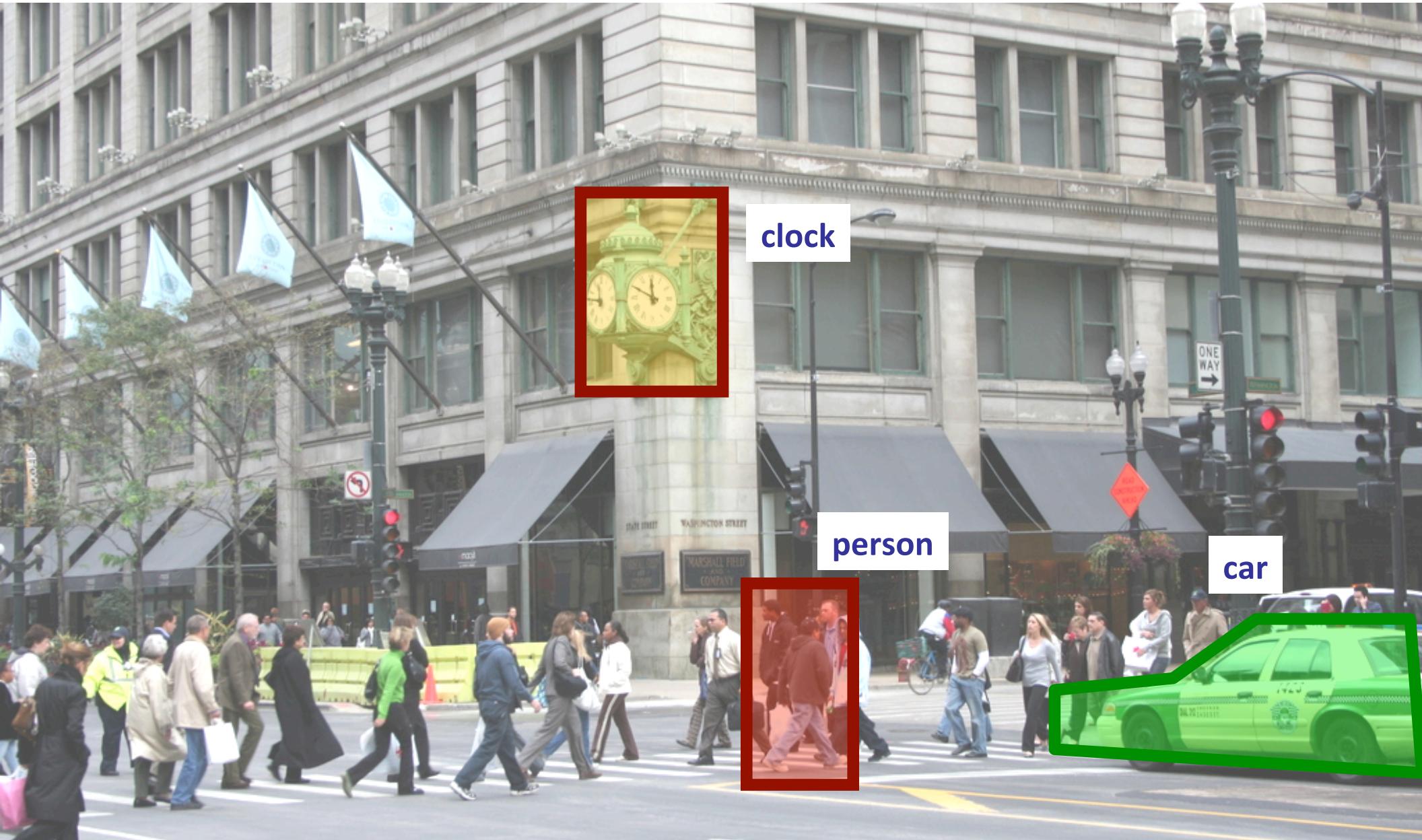
Visual recognition



- Object Classification – BoW models (part 2)
- 2D Object Detection
 - Template based approaches
 - Part-based approaches

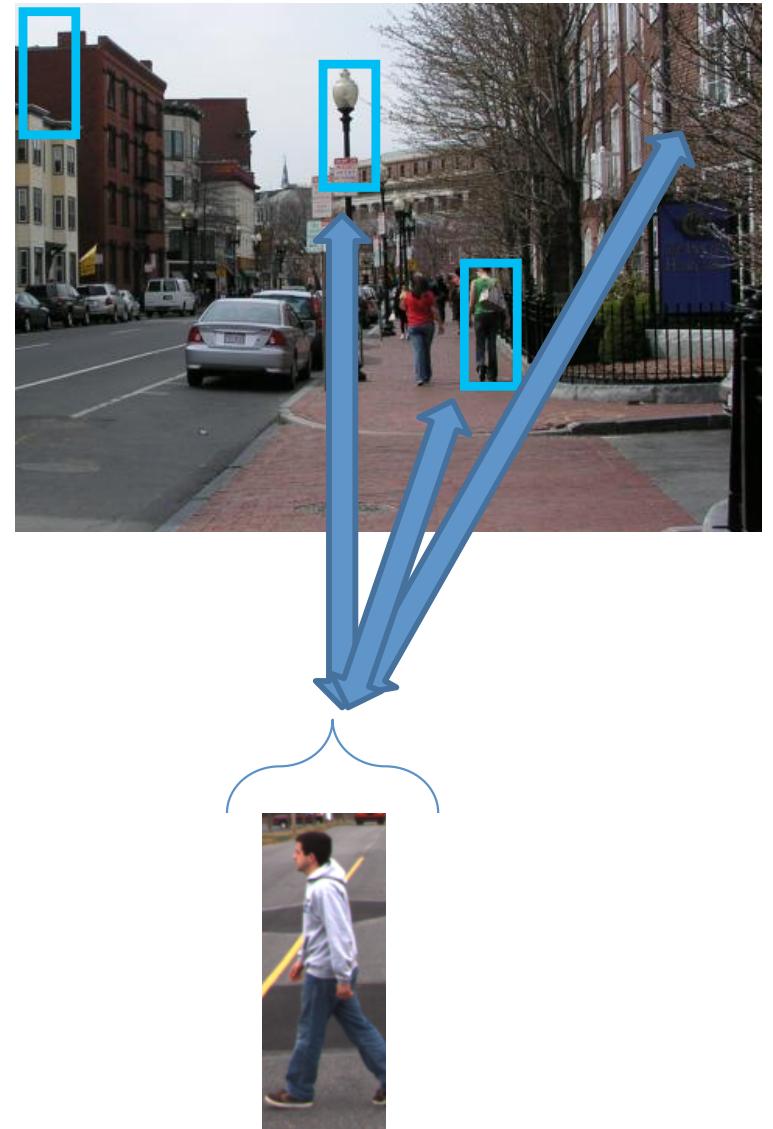
Detection

Which object does this image contain? [where?]



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



Exemplar

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

Limitations 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?

Limitations of template based approaches

Strengths

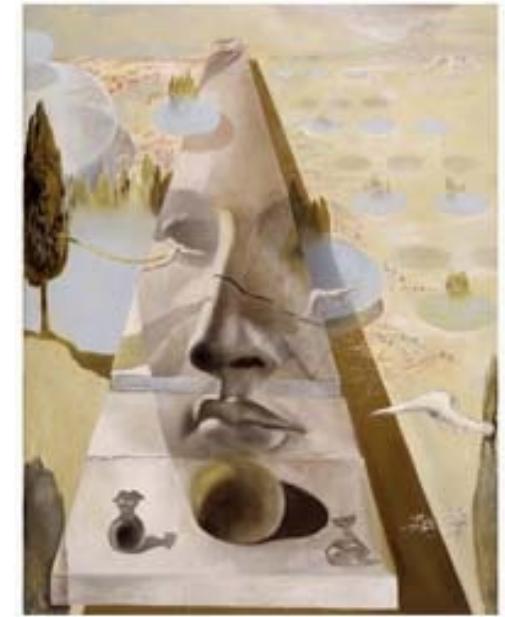
- Works very well for non-deformable objects with canonical orientations: faces, cars, pedestrians
- Fast detection

Weaknesses

- Not so well for highly deformable objects or “stuff”
- Not robust to occlusion
- Requires lots of training data if one wants to model deformations, large degree of intra-class variation or view point changes

Lecture 14

Visual recognition

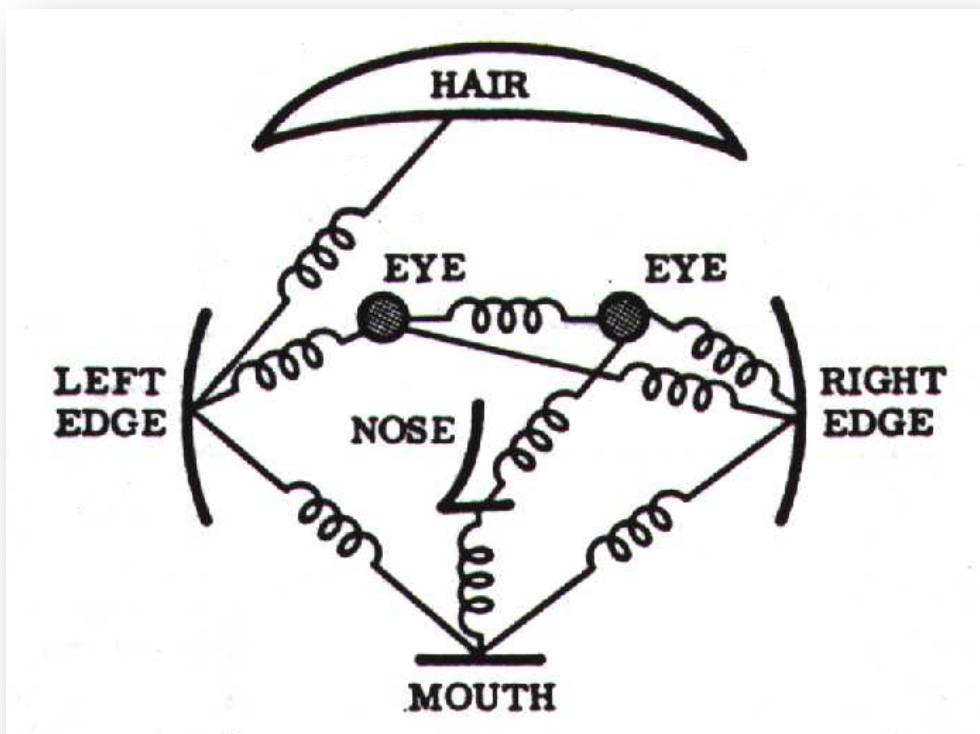


- Object Classification – BoW models (part 2)
- 2D Object Detection
 - Template based approaches
 - Part-based approaches

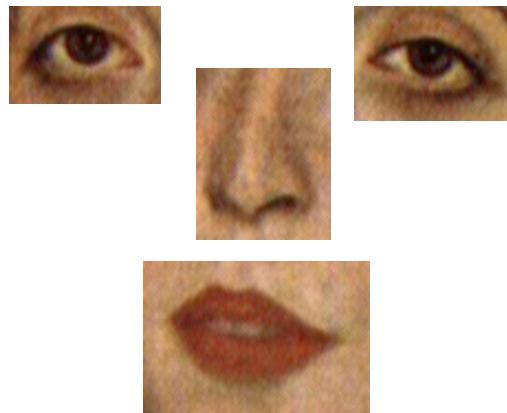
Part Based Representation

- Object as set of parts
- Model:
 - Relative locations between parts
 - Appearance of each part

Figure from [Fischler & Elschlager 73]



Deformations



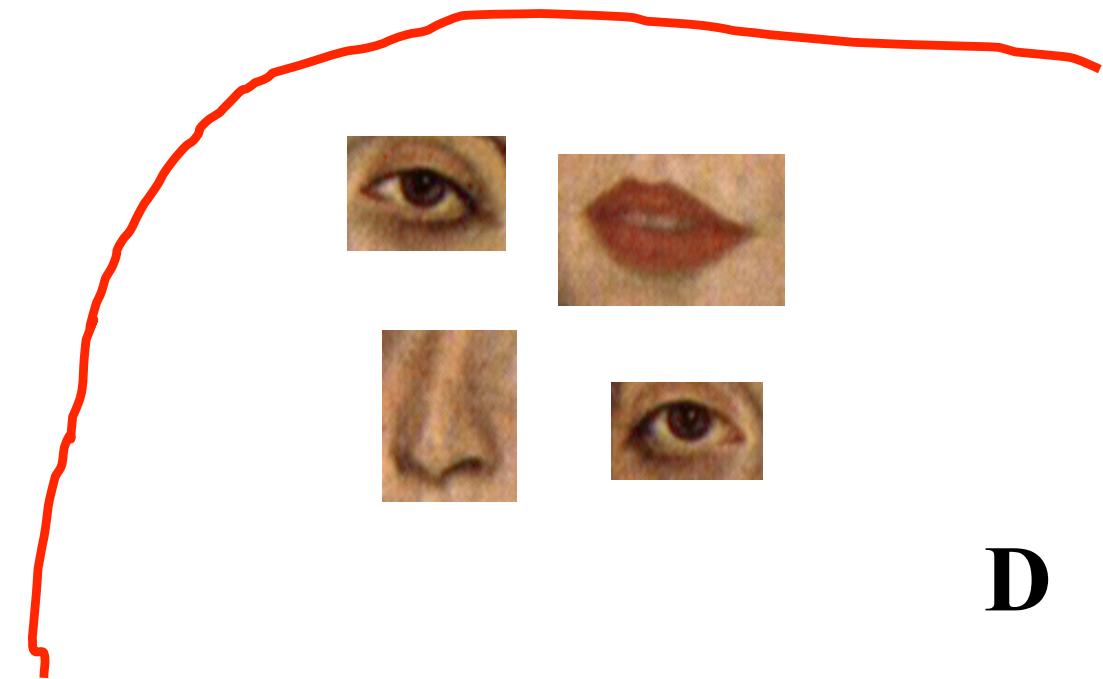
A



B



C



D

Presence / Absence of Features



www.corbis.com



Sparse representation

Computationally tractable (10^5 pixels \rightarrow 10^1 -- 10^2 parts)
... but throw away potentially useful image information



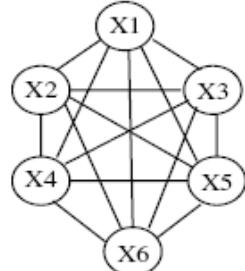
Larger discrimination power

Parts can help fine-grained discrimination



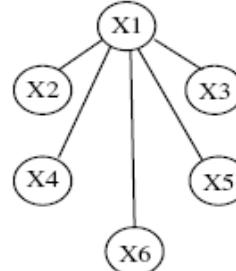
Different connectivity structures

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



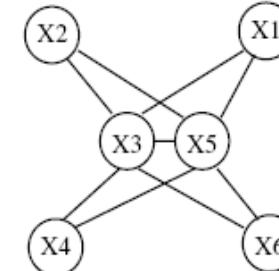
a) Constellation

Crandall et al. '05
Leibe 05; Felzenszwalb 09



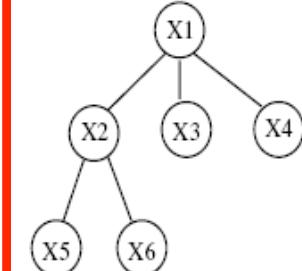
b) Star shape

Crandall et al. '05

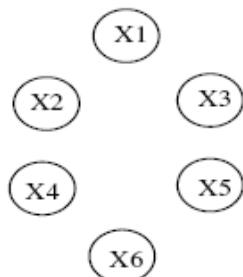


c) k -fan ($k = 2$)

Felzenszwalb & Huttenlocher '00

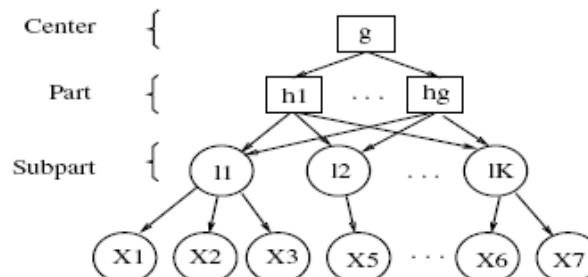


d) Tree



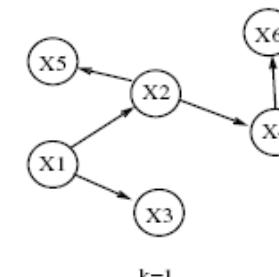
e) Bag of features

Csurka '04
Vasconcelos '00



f) Hierarchy

Bouchard & Triggs '05

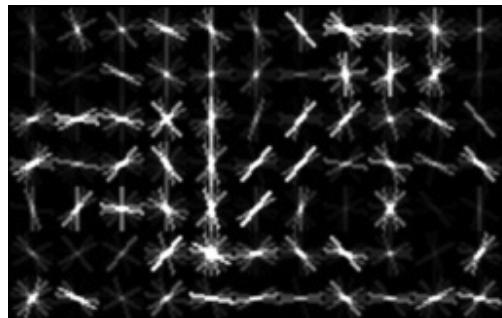


g) Sparse flexible model

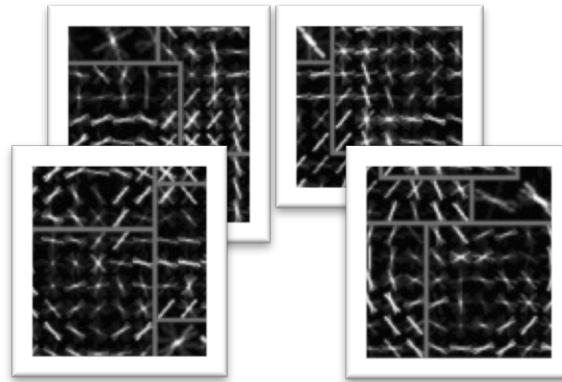
Carneiro & Lowe '06

Deformable Part Models (DPM)

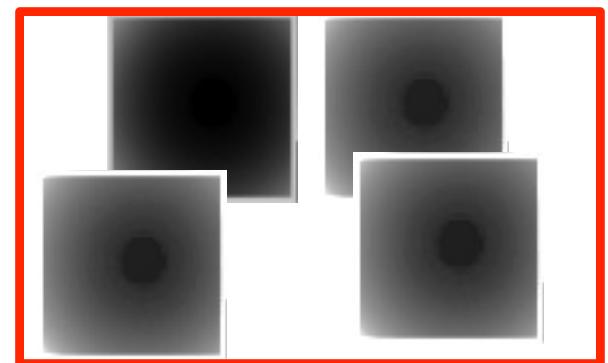
Felzenszwalb, et al., **Discriminatively Trained Deformable Part Models**, <http://people.cs.uchicago.edu/~pff/latent/>



root filters (coarse)



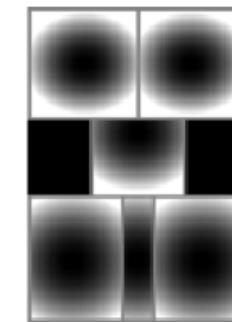
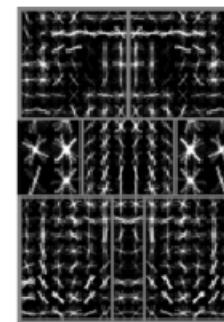
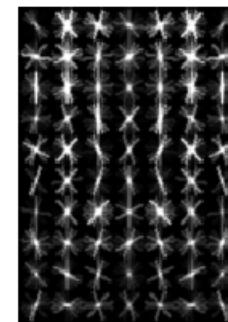
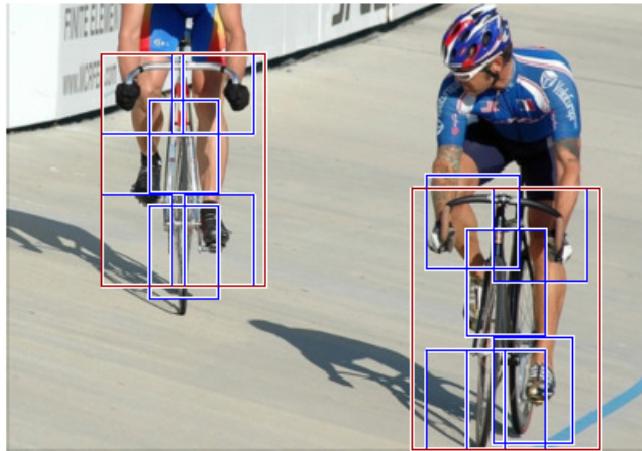
part filters (fine)



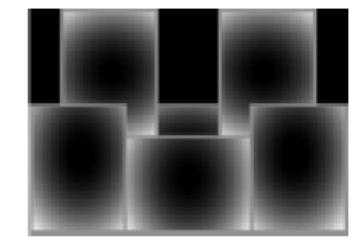
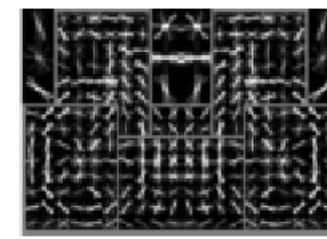
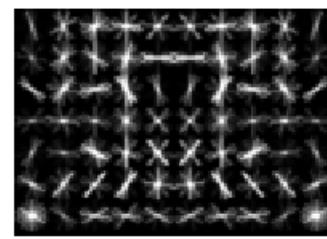
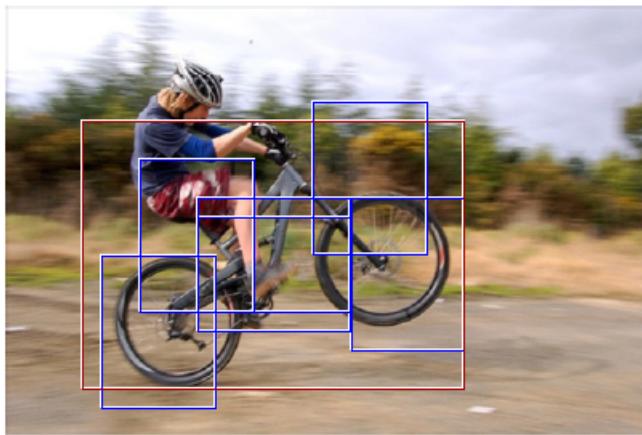
deformation models

Mixture of components

“frontal” component



“side” component

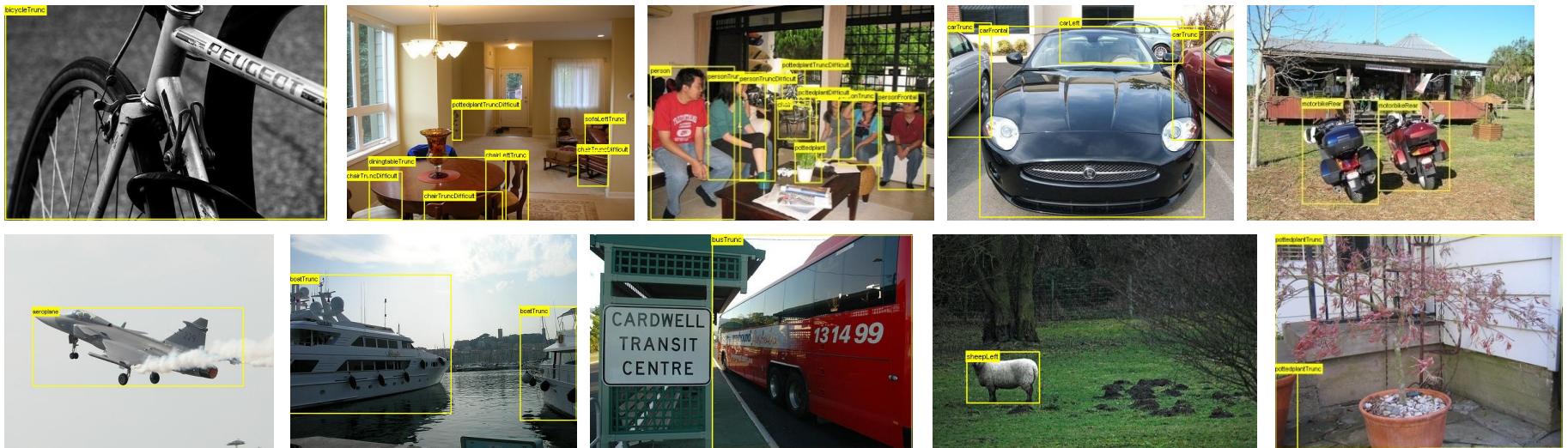


SVM with mixtures of components

Pascal Dataset

<http://host.robots.ox.ac.uk/pascal/VOC/voc2012/index.html>

- 20 classes: aeroplane, bicycle, boat, bottle, bus, car, cat, chair, cow, dining table, dog, horse, motorbike, person, potted plant, sheep, train, TV
- Real images downloaded from flickr, not filtered for “quality”

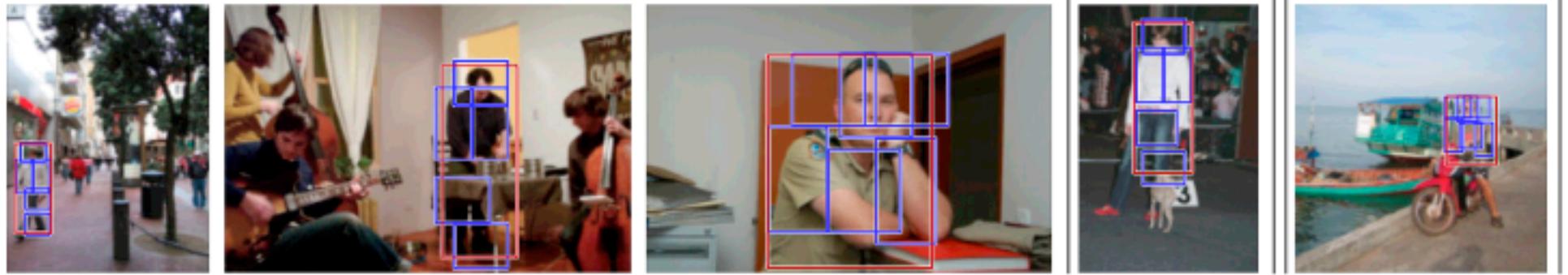


- Complex scenes, scale, pose, lighting, occlusion, ...

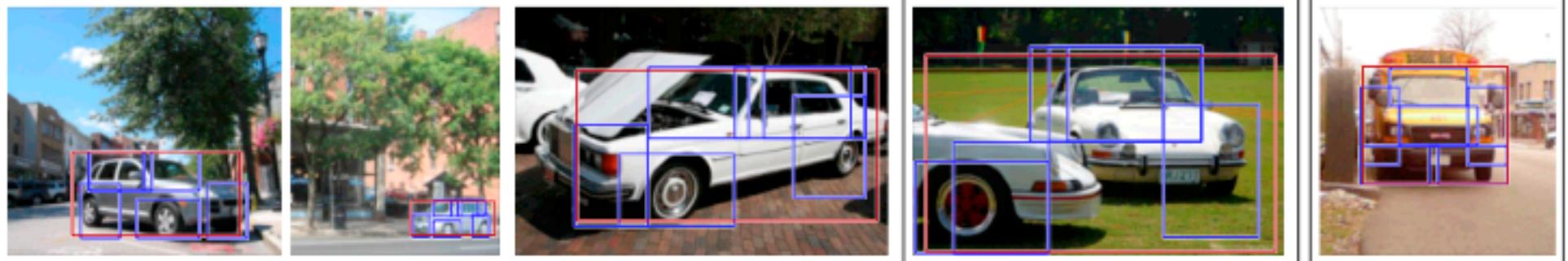
Results on the PASCAL dataset

One of the most advanced version of DPM achieves an average precision of 41%

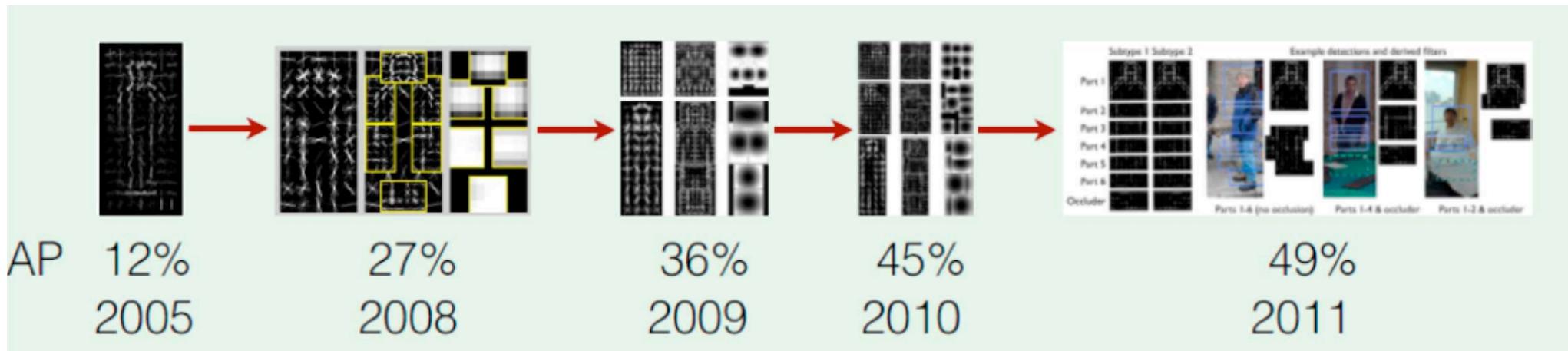
person



car



Results on detecting deformable objects (humans)



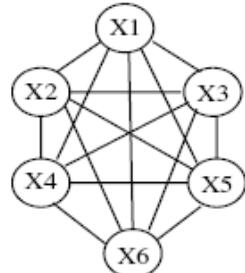
HOG templates
Dalal & Triggs 2005

DPM
Felzenszwalb et al 2010

DPM by flexible
Mixtures-of-Parts
Yang & Ramanan, 2012

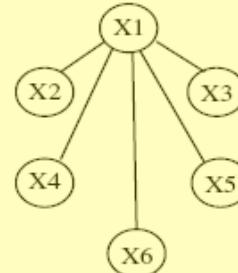
Different connectivity structures

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



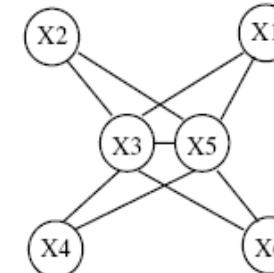
a) Constellation [13]

Crandall et al. '05
Leibe 05; Felzenszwalb 09



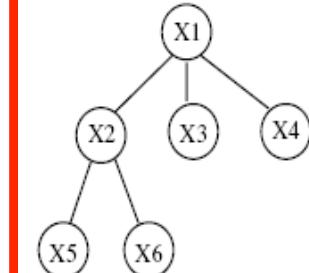
b) Star shape [9, 14]

Crandall et al. '05

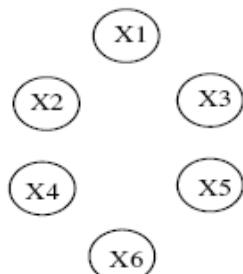


c) k -fan ($k = 2$) [9]

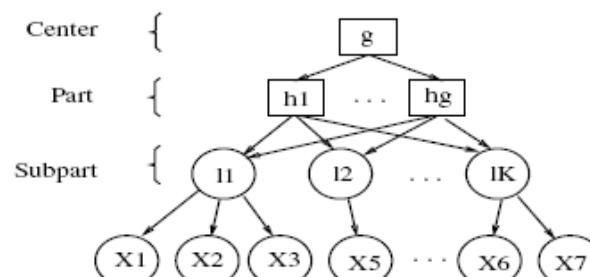
Felzenszwalb & Huttenlocher '00



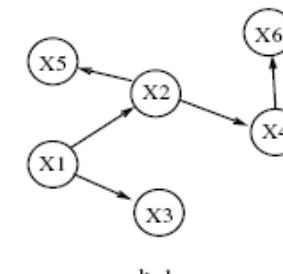
d) Tree [12]



e) Bag of features [10, 21]
Csurka '04
Vasconcelos '00



f) Hierarchy [4]
Bouchard & Triggs '05

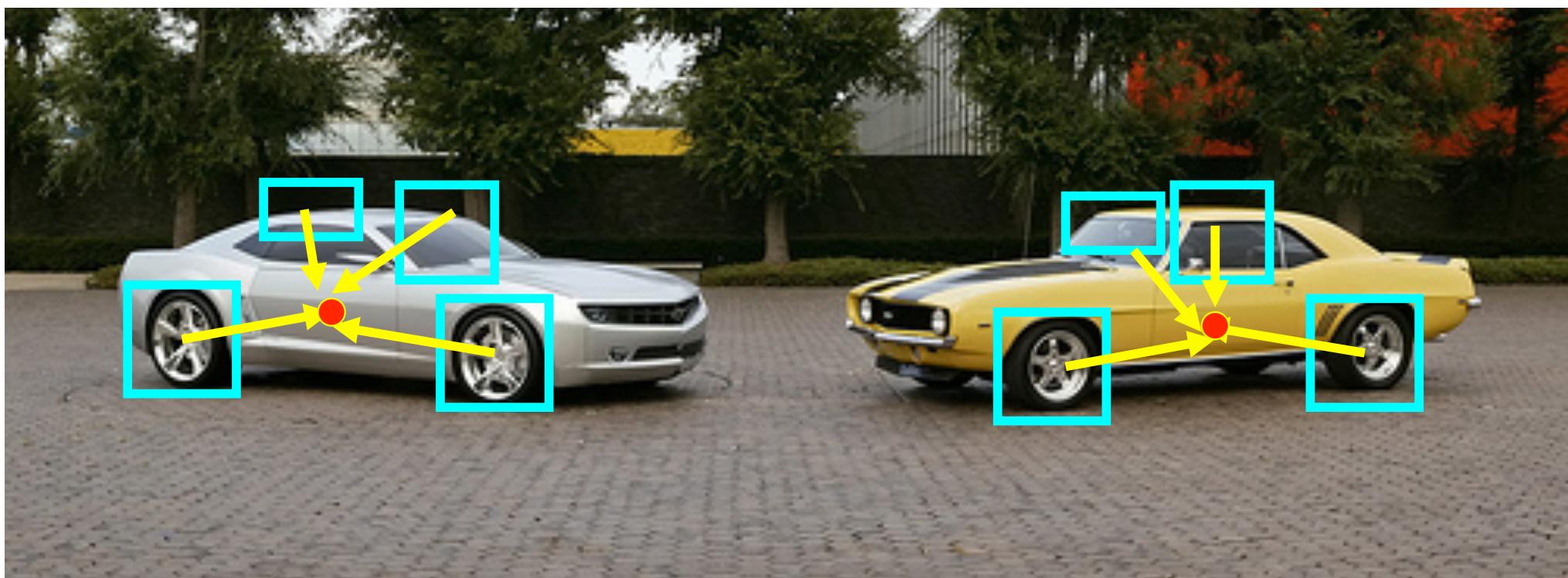


g) Sparse flexible model
Carneiro & Lowe '06

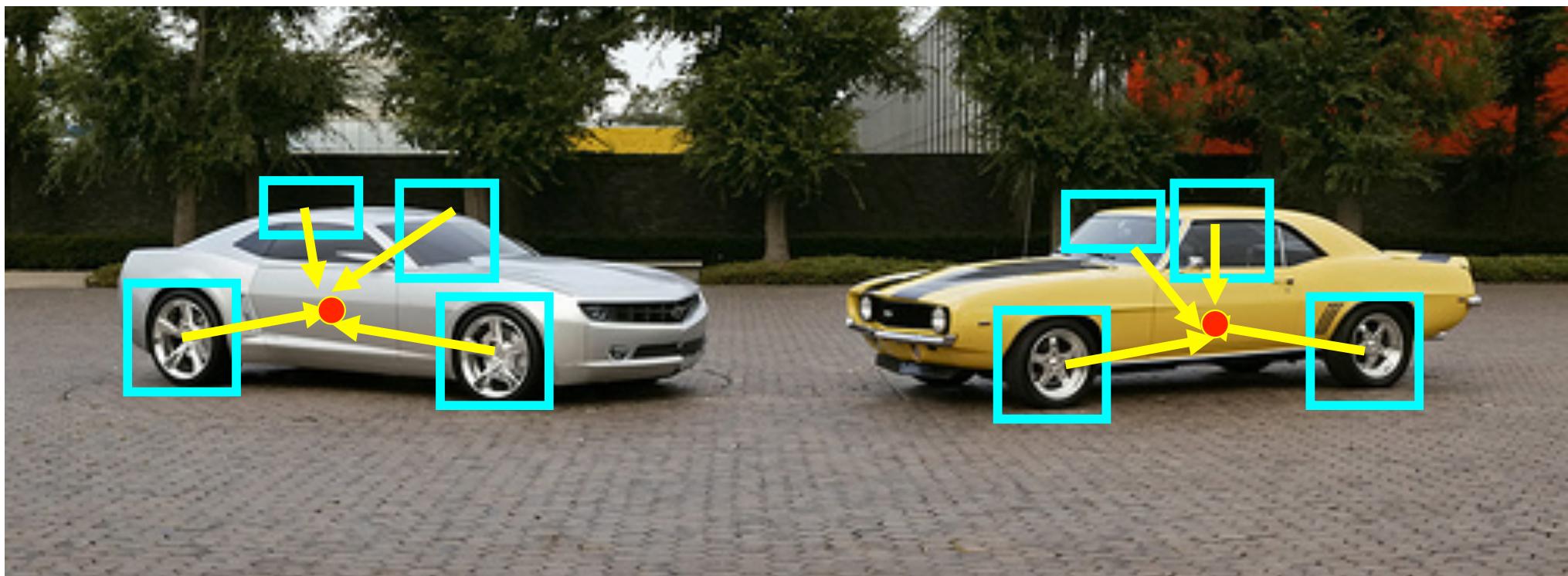
Implicit shape models by generalized Hough voting



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



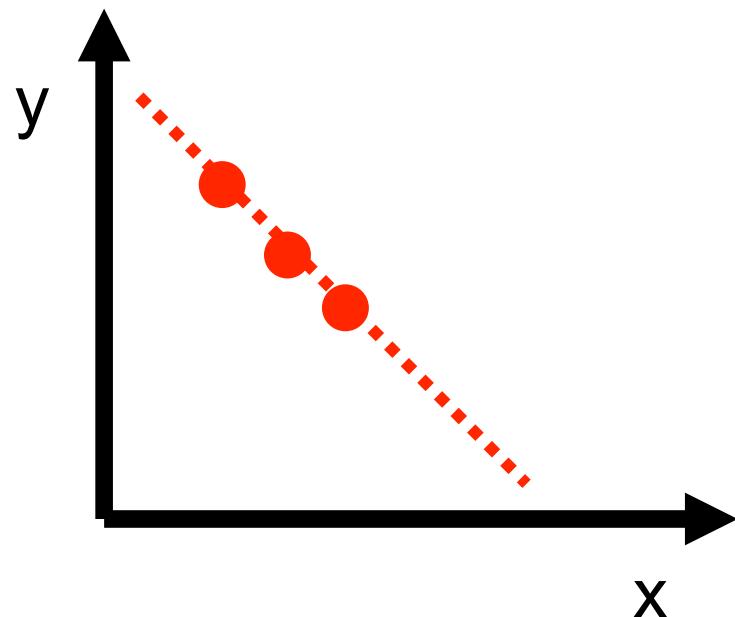
Object representation: How to capture constellation of parts? Using Hough Voting



Hough transform

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

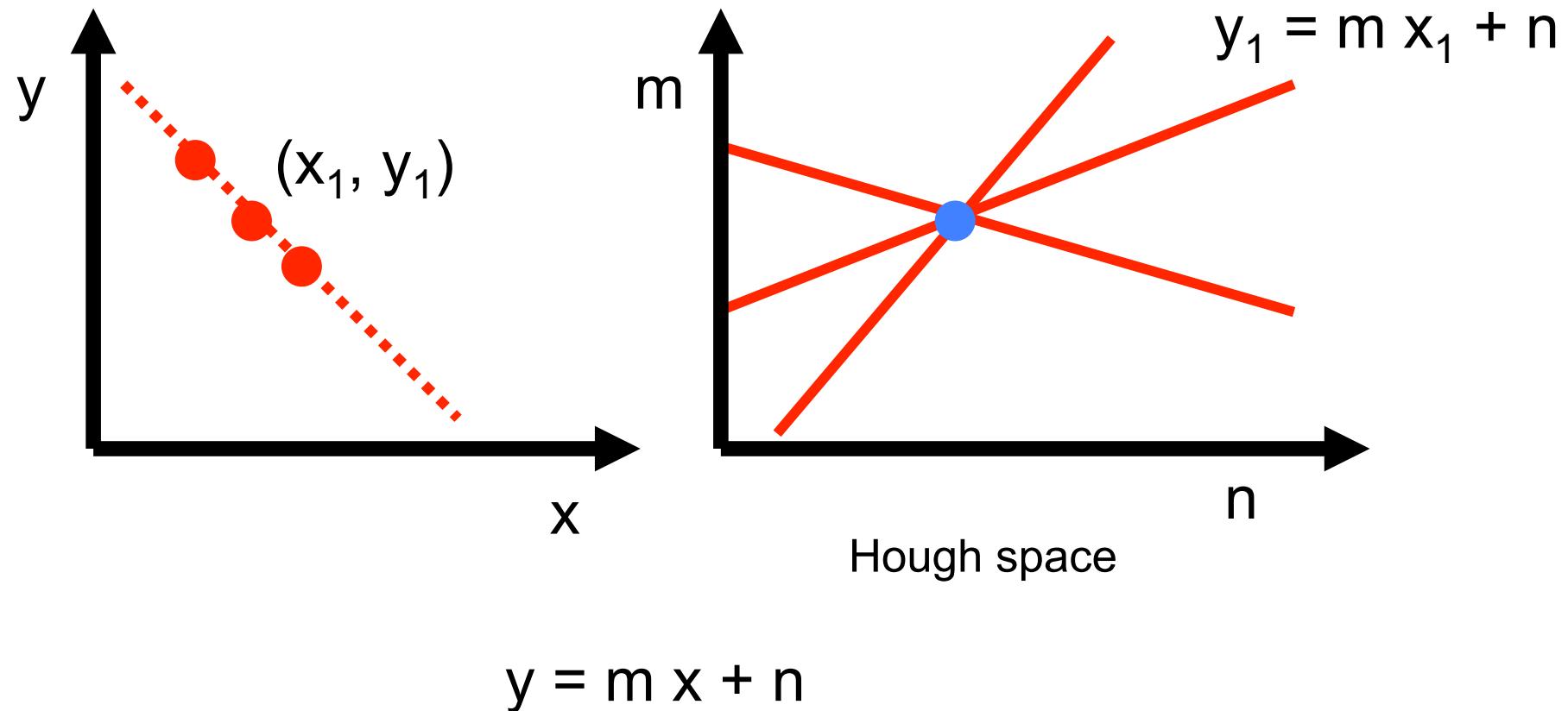
Given a set of points, find the curve or line that explains the data points best



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

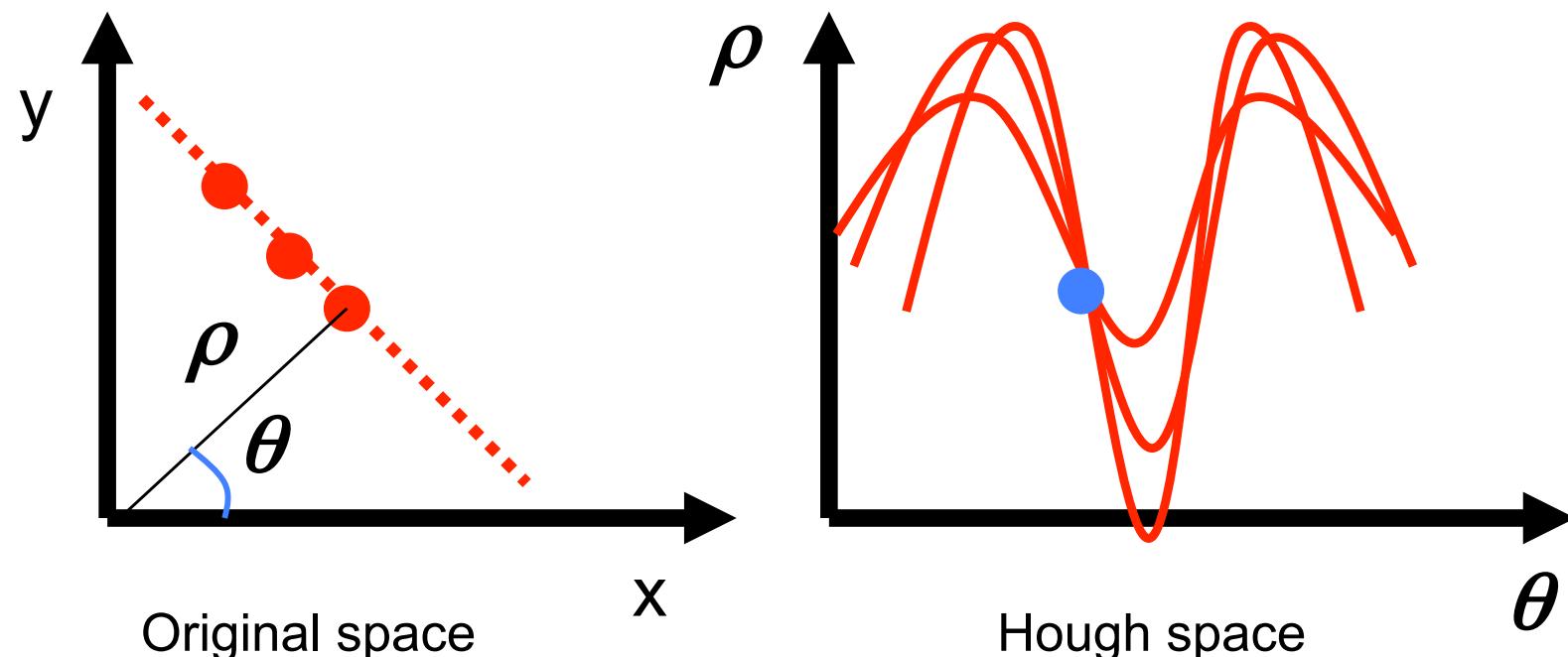


Hough transform

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

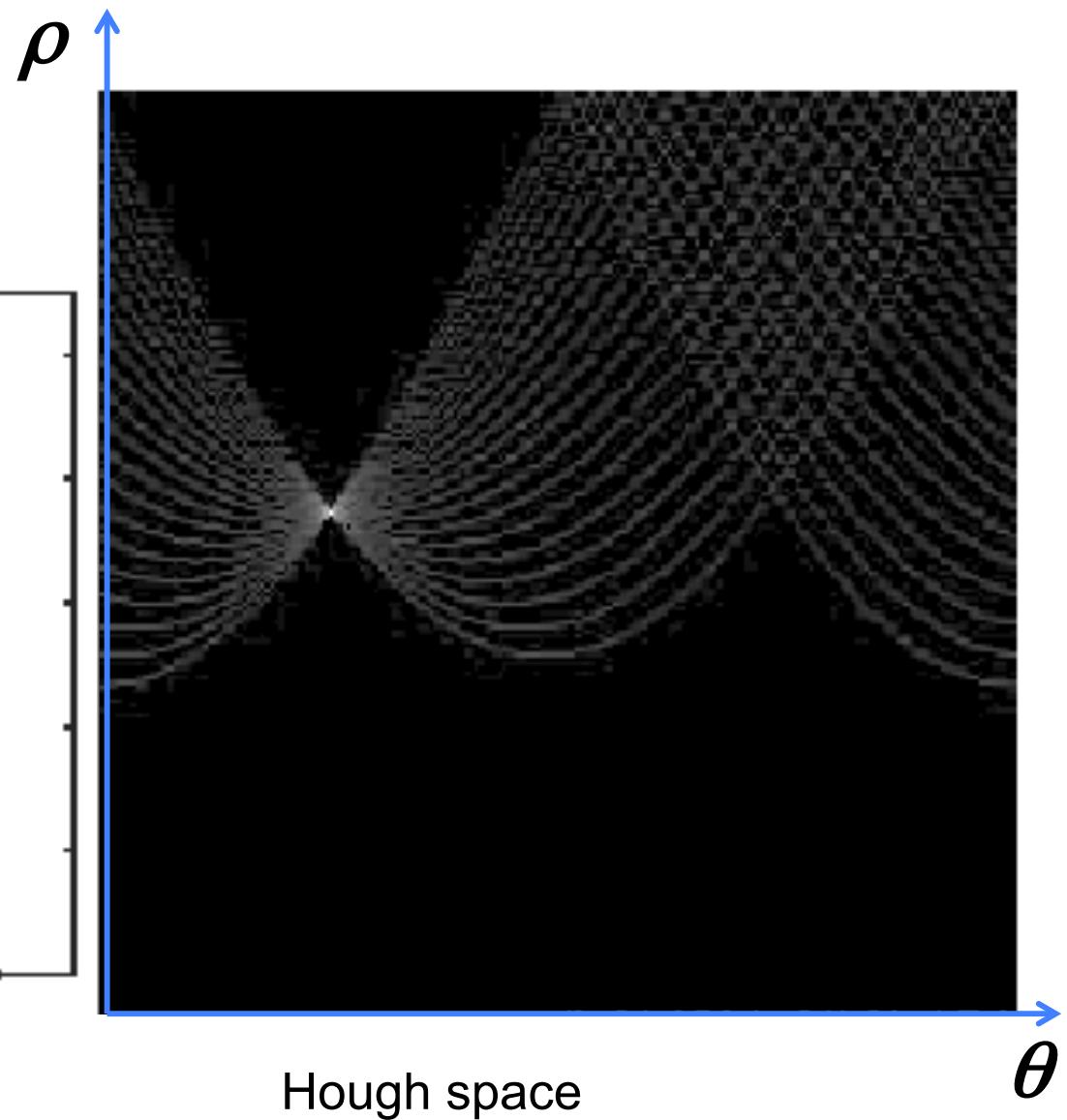
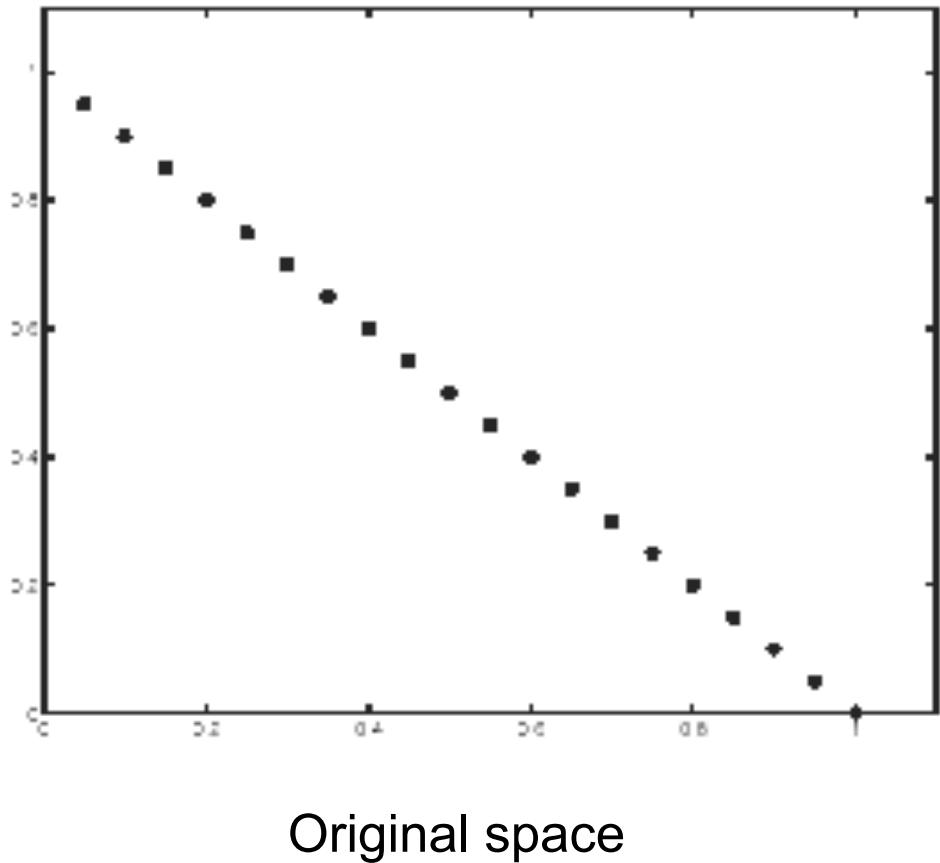
Any Issue? The parameter space $[m,n]$ is unbounded...

- Use a polar representation for the parameter space

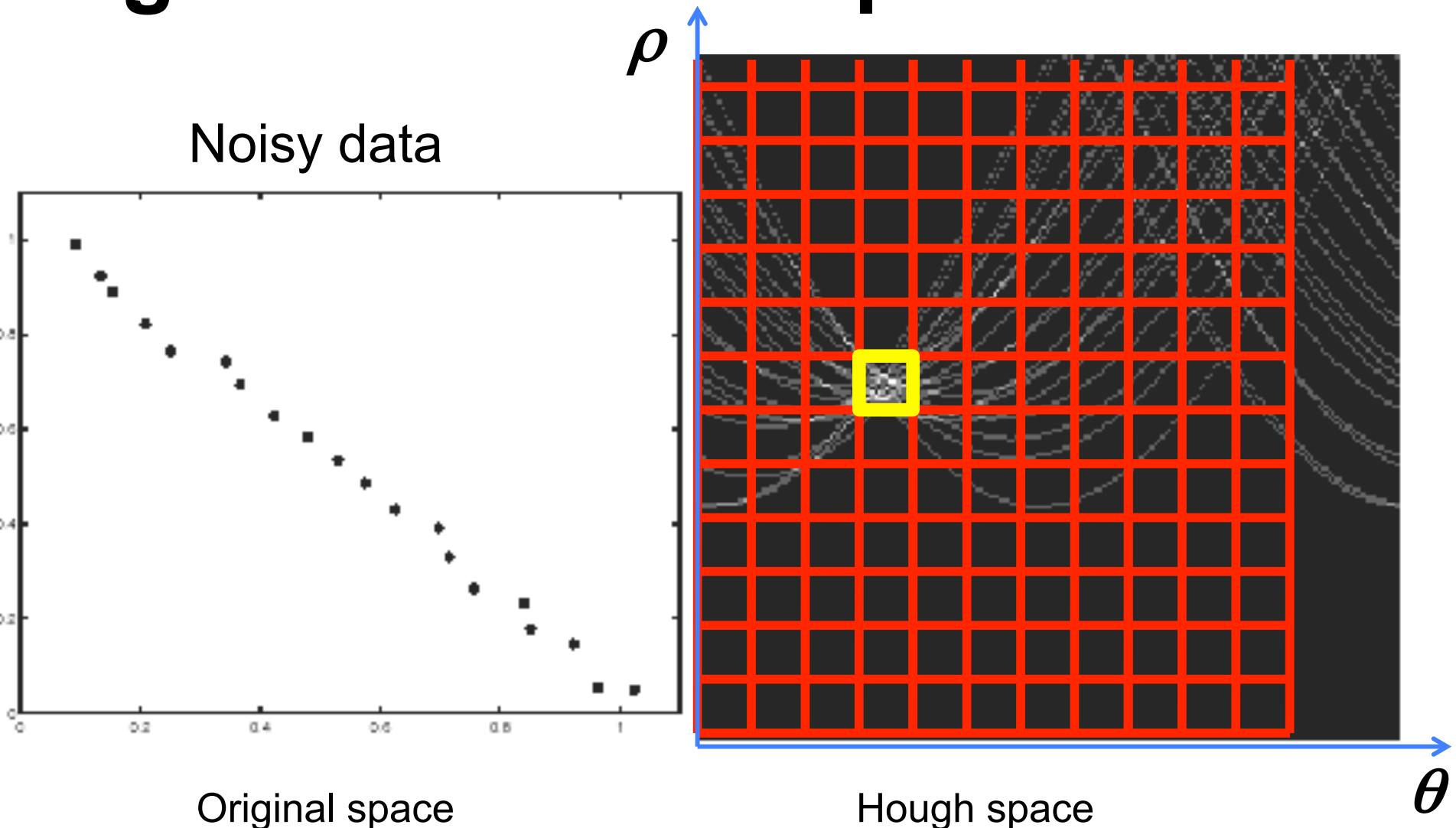


$$x \cos \theta + y \sin \theta = \rho$$

Hough transform - experiments



Hough transform - experiments



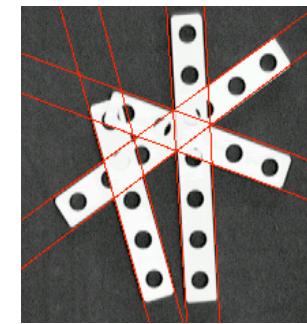
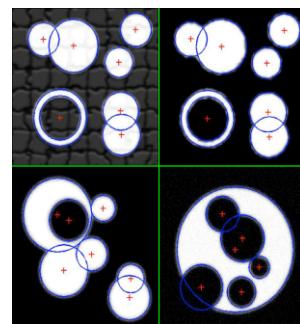
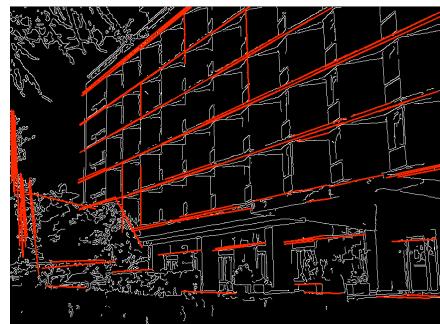
How to compute the intersection point?

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
- Popular for detecting parameterized shapes
 - Hough'59, Duda&Hart'72, Ballard'81,...



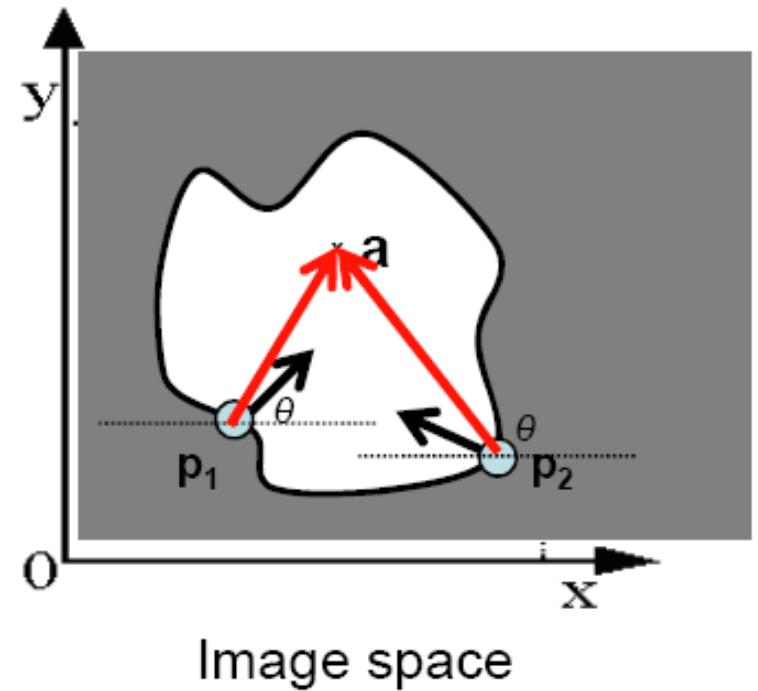
Generalized Hough Transform

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

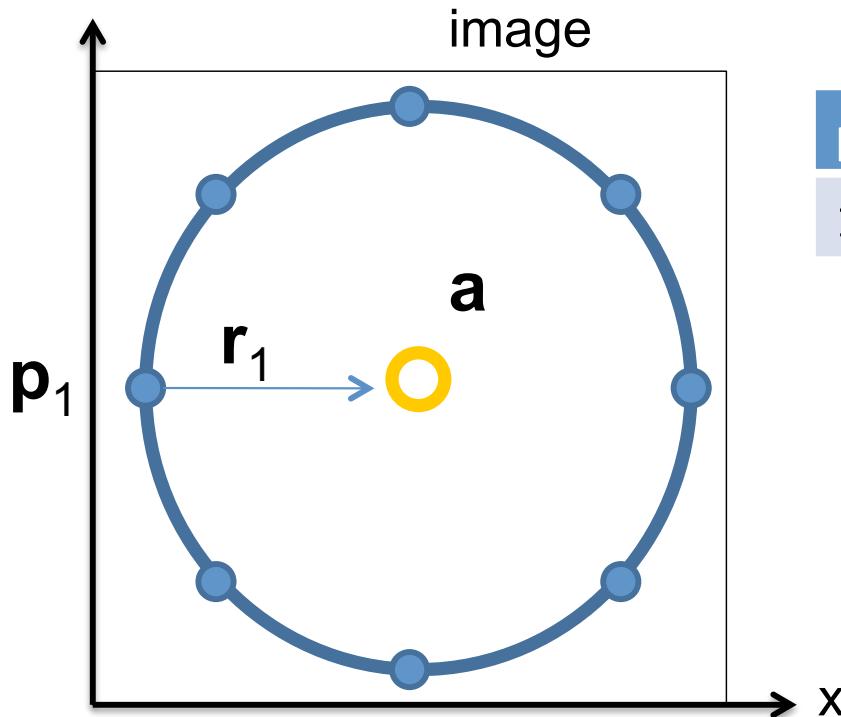
Learning a model of shape S:

For a given shape S , at each boundary point p_i of S , compute displacement vector $r_i = a - p_i$ [Eq. 2]

Store these vector r_i in a table indexed by the gradient orientation θ_i computed at p_i



Example: a circle



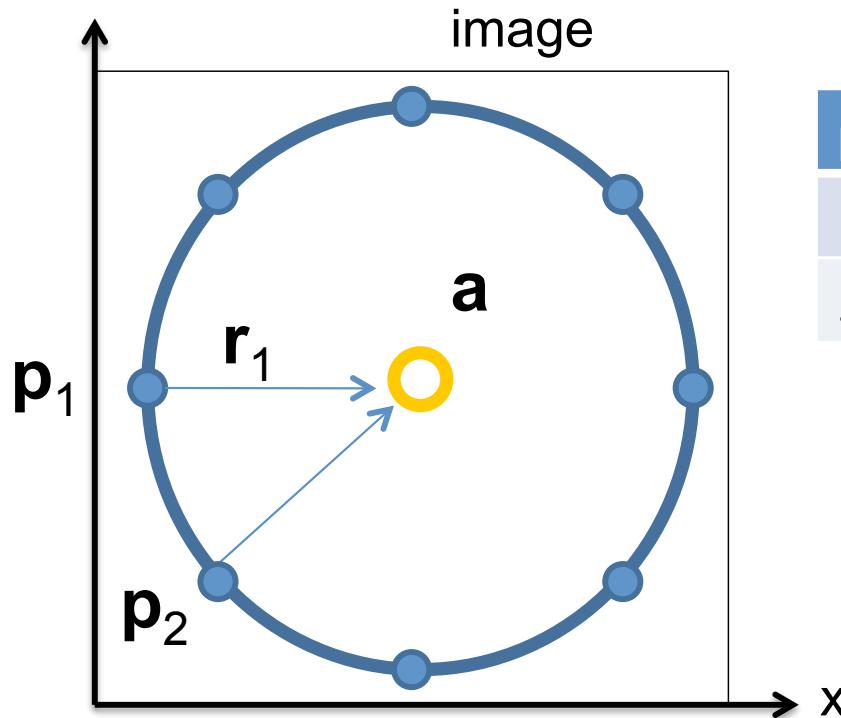
p	θ	r_x	r_y
1	0	1	0

Learning a model of shape S :

For a given shape S , at each boundary point p_i of S , compute displacement vector $r_i = a - p_i$ [Eq. 2]

Store these vector r_i in a table indexed by the gradient orientation θ_i computed at p_i

Example: a circle



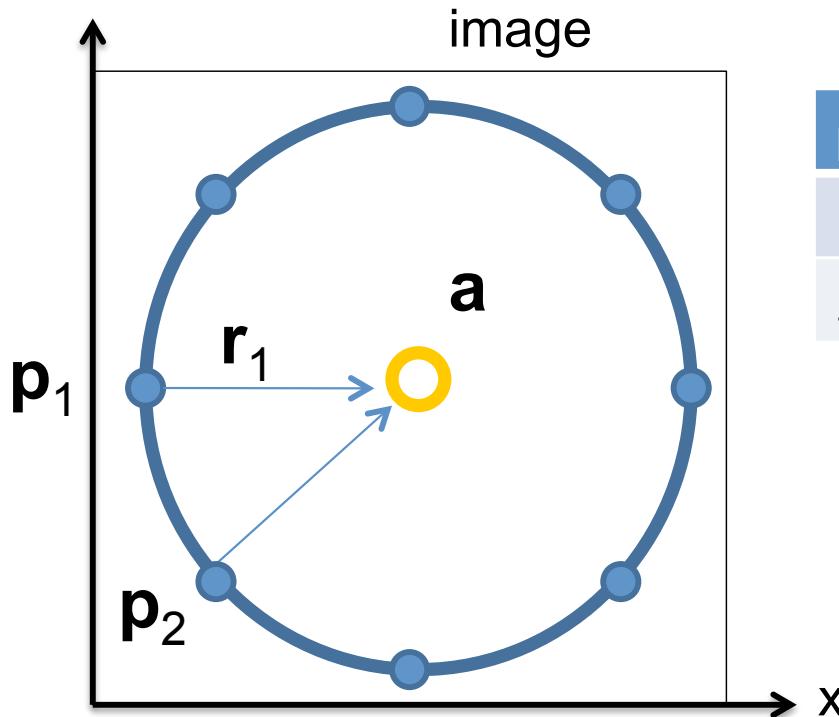
p	θ	r_x	r_y
1	0	1	0
2	?	?	?

Learning a model of shape S :

For a given shape S , at each boundary point p_i of S , compute displacement vector $r_i = a - p_i$ [Eq. 2]

Store these vector r_i in a table indexed by the gradient orientation θ_i computed at p_i

Example: a circle



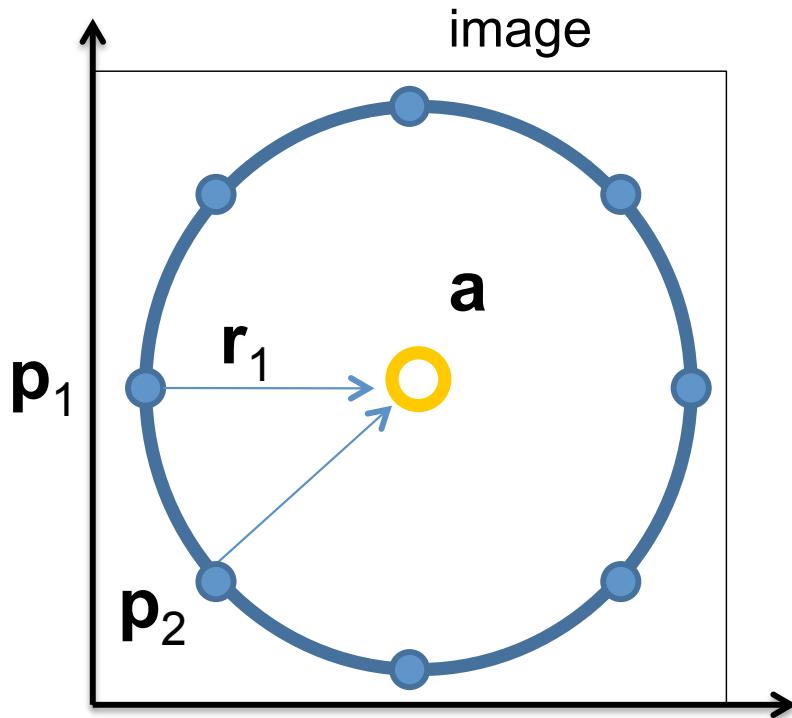
p	θ	r_x	r_y
1	0	1	0
2	45	0.7	0.7

Learning a model of shape S :

For a given shape S , at each boundary point p_i of S , compute displacement vector $r_i = a - p_i$ [Eq. 2]

Store these vector r_i in a table indexed by the gradient orientation θ_i computed at p_i

Example: a circle



p	θ	r_x	r_y
1	0	1	0
2	45	0.7	0.7
3	90	0	1
4	135	-0.7	0.7
	...		
8	270	0.7	-0.7

Learning a model of shape S :

For a given shape S , at each boundary point p_i of S , compute displacement vector $r_i = a - p_i$ [Eq. 2]

Store these vector r_i in a table indexed by the gradient orientation θ_i computed at p_i

Detecting query model shape

Assume a model of shape S is learned.

The following steps allow to assess if a unknown query shape is S or not:

- For each edge point
 - Index into table with its gradient orientation θ
 - Use retrieved vectors r to vote for position of reference point (object centroid)
- Reference point that receives the largest aggregation of votes (= peak in the Hough space) is retained.
- If # of votes is above a threshold, the object shape is classified as shape S .
- The location with the largest aggregations of votes corresponds to the actual position of the object in image coordinates.

Example: a circle

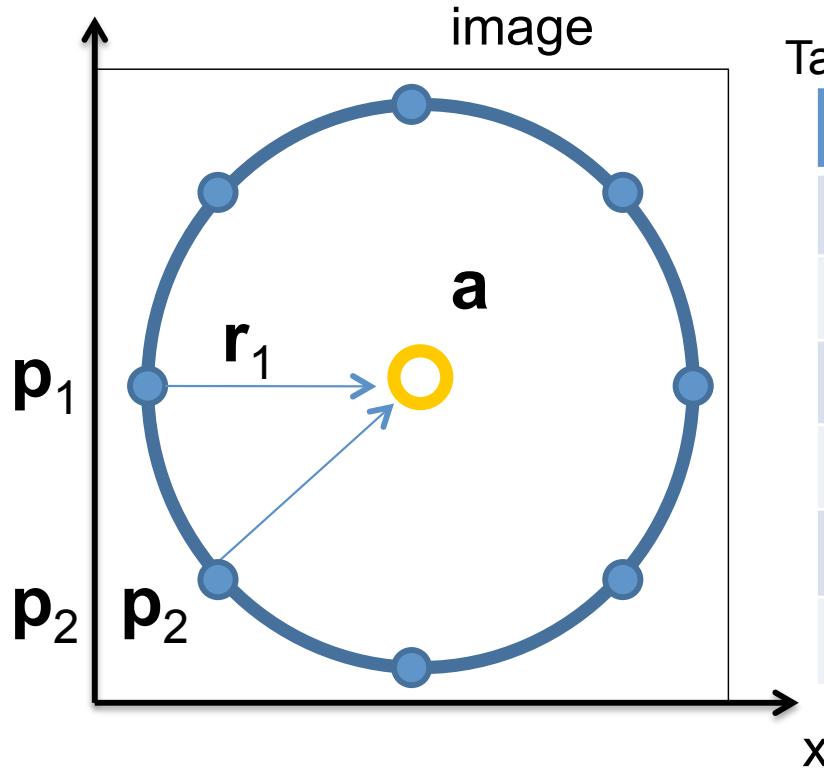
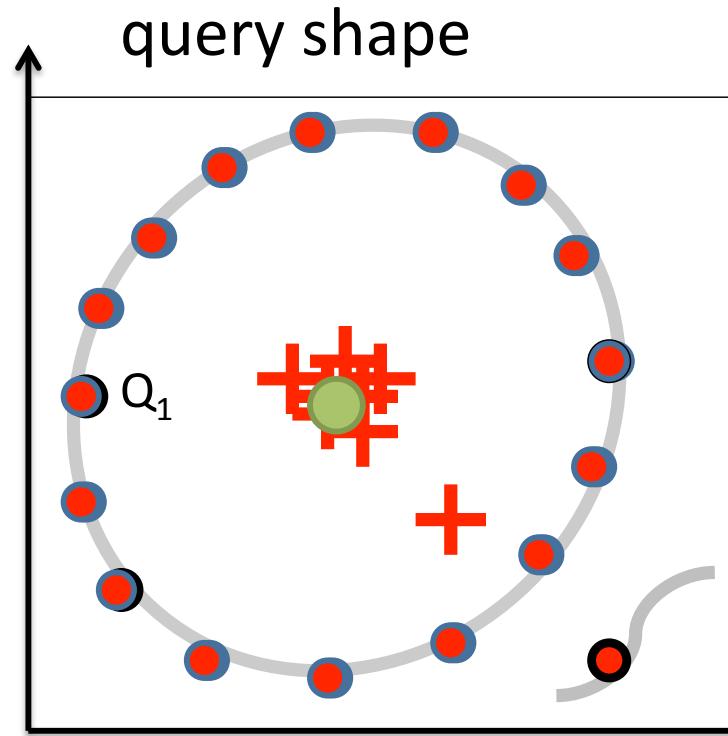


Table T = learnt model of the circle

p	θ	r_x	r_y
1	0	1	0
2	45	0.7	0.7
3	90	0	1
4	135	-0.7	0.7
	...		
8	270	0.7	-0.7



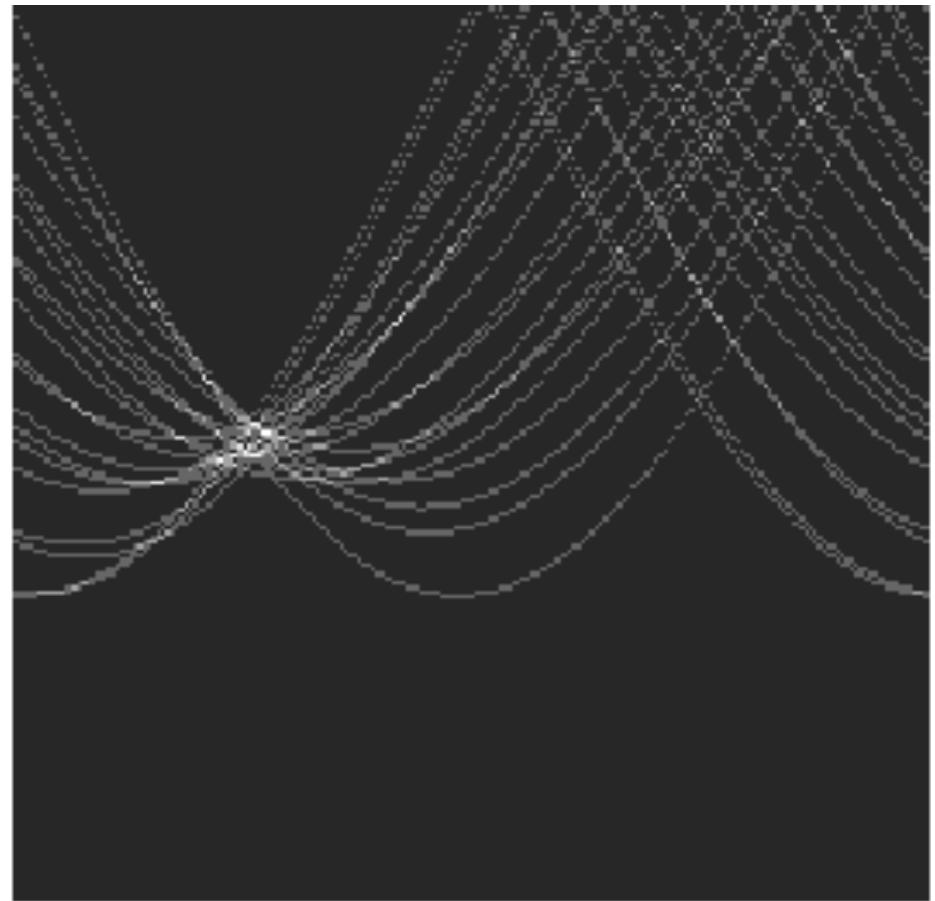
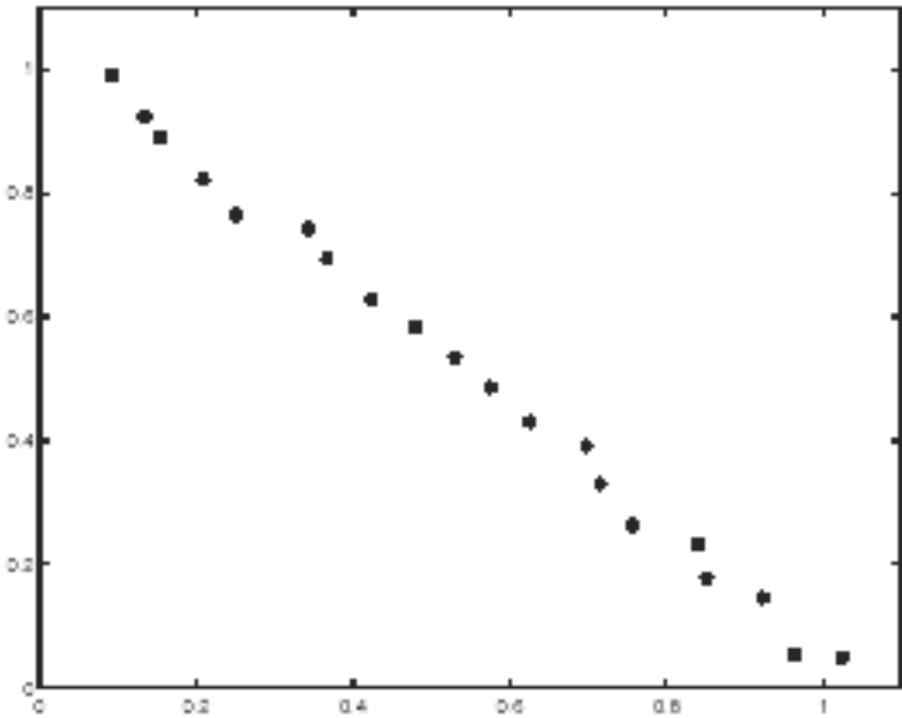
$$Q_1 \rightarrow \theta_1 = 0 \rightarrow R_1 = [r_x, r_y] = [1, 0] \rightarrow C_1 = Q_1 + R_1$$

$$Q_2 \rightarrow \theta_2 = 45 \rightarrow R_2 = [r_x, r_y] = [.7, .7] \rightarrow C_2 = Q_2 + R_2$$

$$Q_k \rightarrow \theta_k = -180 \rightarrow R_k = [r_x, r_y] = [-1, 0] \rightarrow C_k = Q_k + R_k$$

:

Conceptually similar to



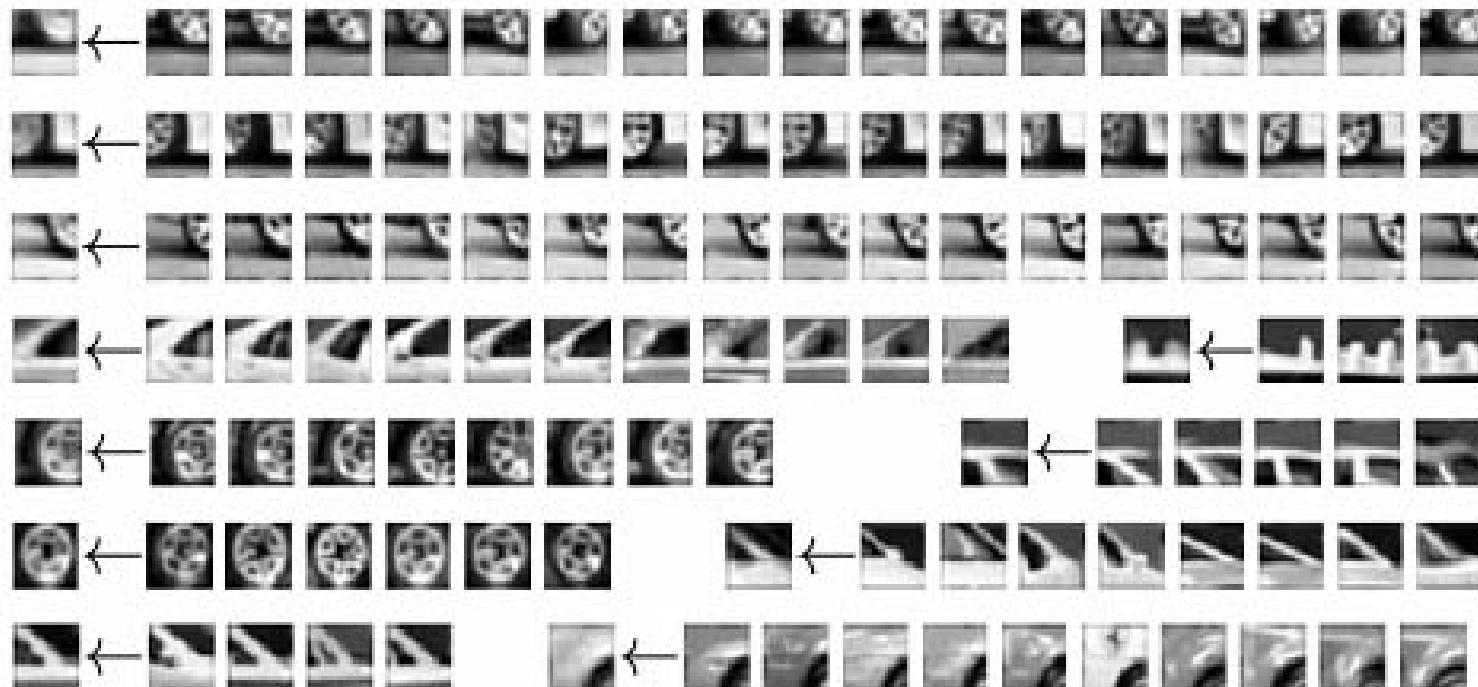
Implicit shape models

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

- Instead of indexing displacements by gradient orientations, index by “visual codeword”
- Enable detection without sliding windows!

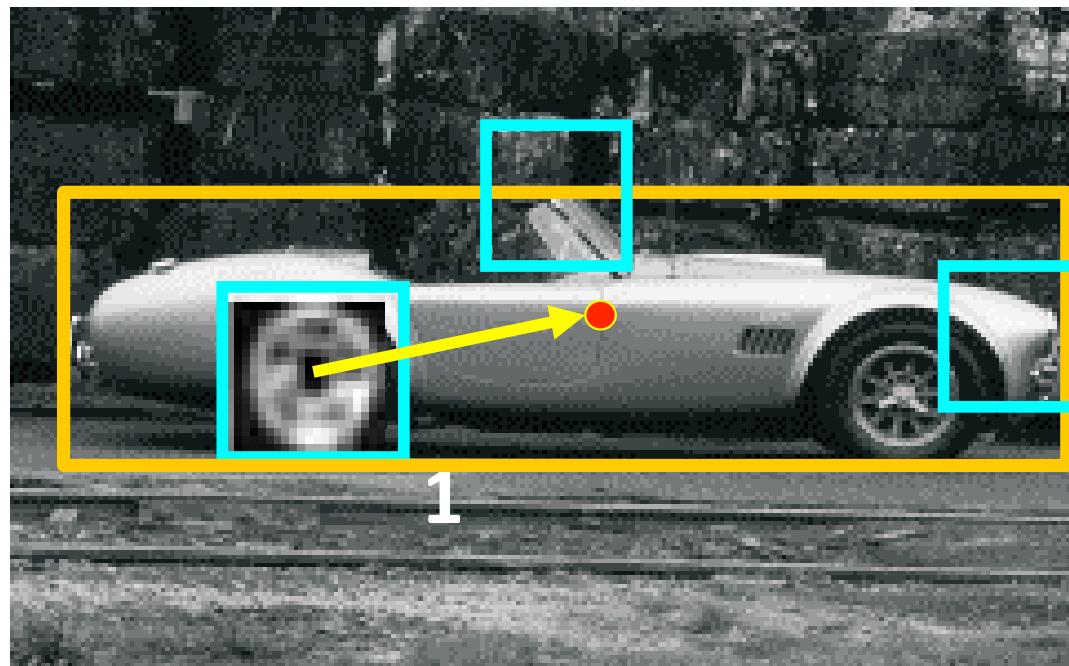
Implicit shape models: Training

1. Build codebook of descriptors around extracted interest points using clustering (same as for BOW models)



Implicit shape models: Training

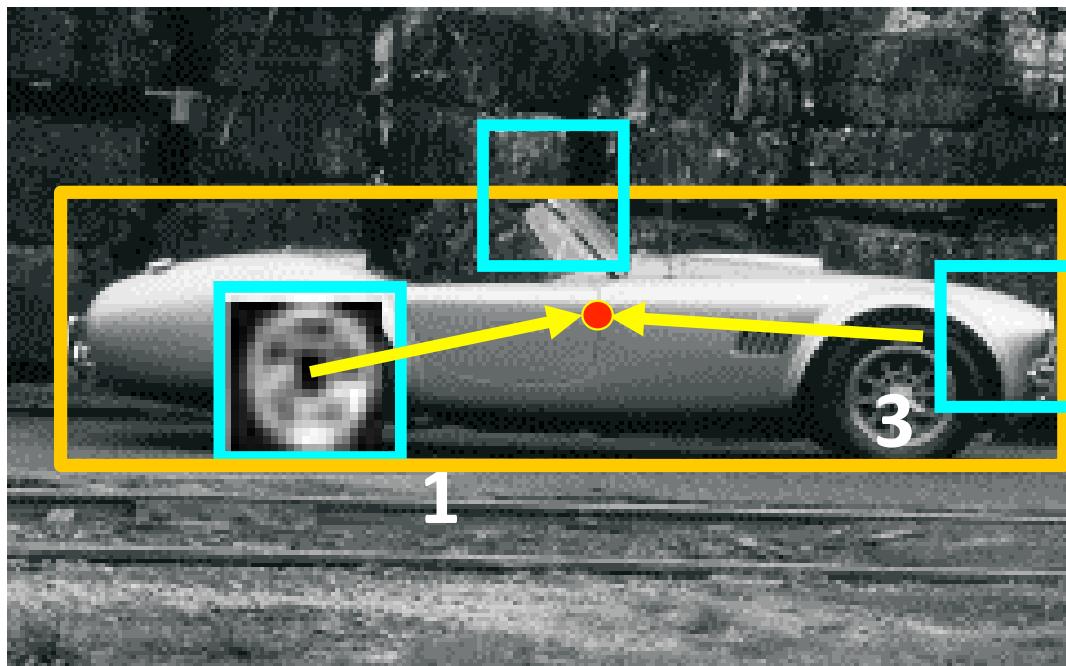
1. Build codebook of descriptors around extracted interest (key) points using clustering
 2. Map the descriptor at each interest point to closest codebook entry
 3. For each codebook entry, store all positions relative to object center [center is given] and scale [bounding box is given]



CW	rx	ry	Scale
1	0.9	0.1	1

Implicit shape models: Training

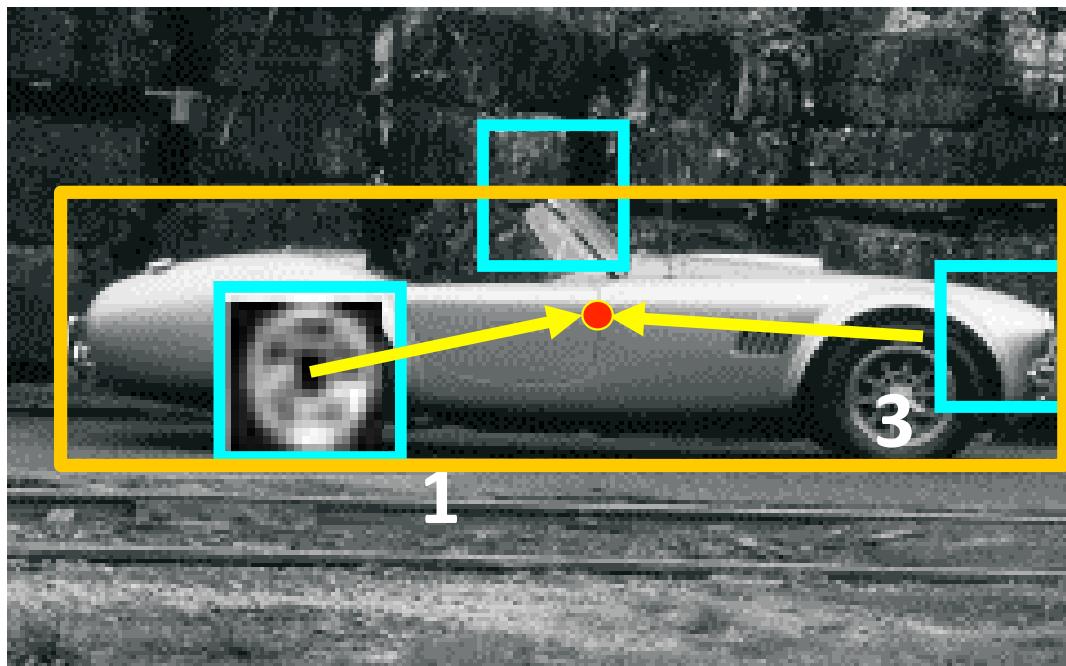
1. Build codebook of descriptors around extracted interest points using clustering
2. Map the descriptor at interest point to closest codebook entry
3. For each codebook entry, store all positions relative to object center [center is given] and scale [bounding box is given]



CW	rx	ry	Scale
1	0.9	0.1	1
...			
3	?	?	?

Implicit shape models: Training

1. Build codebook of descriptors around extracted interest points using clustering
2. Map the descriptor at 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]

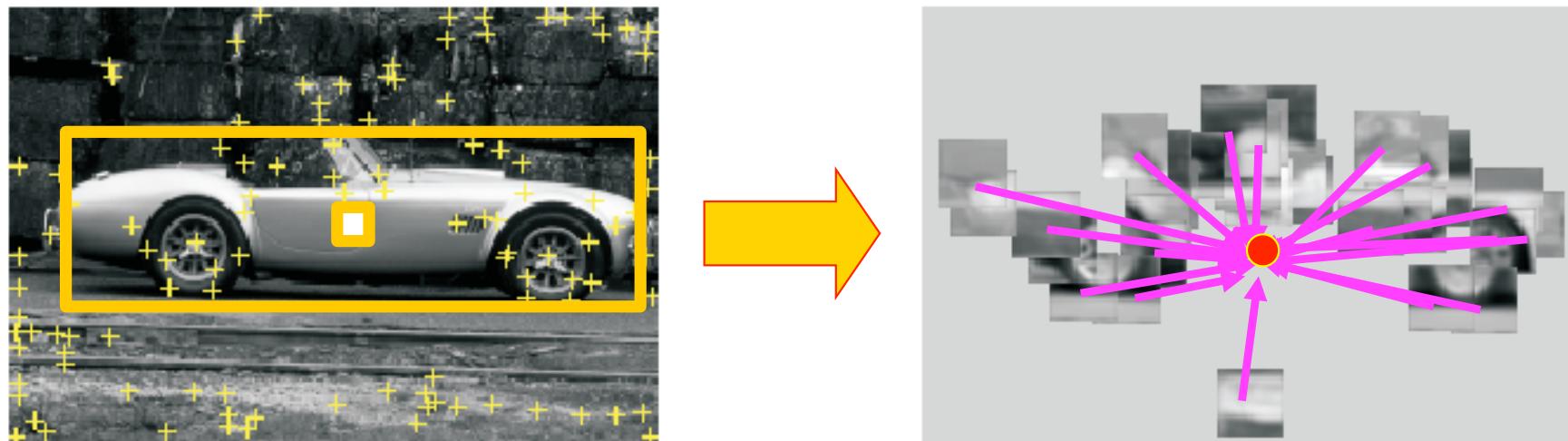


CW	rx	ry	Scale
1	0.9	0.1	1
...			
3	-1	0.1	1
...			
...			
N	0.2	-1	1

table T = car model

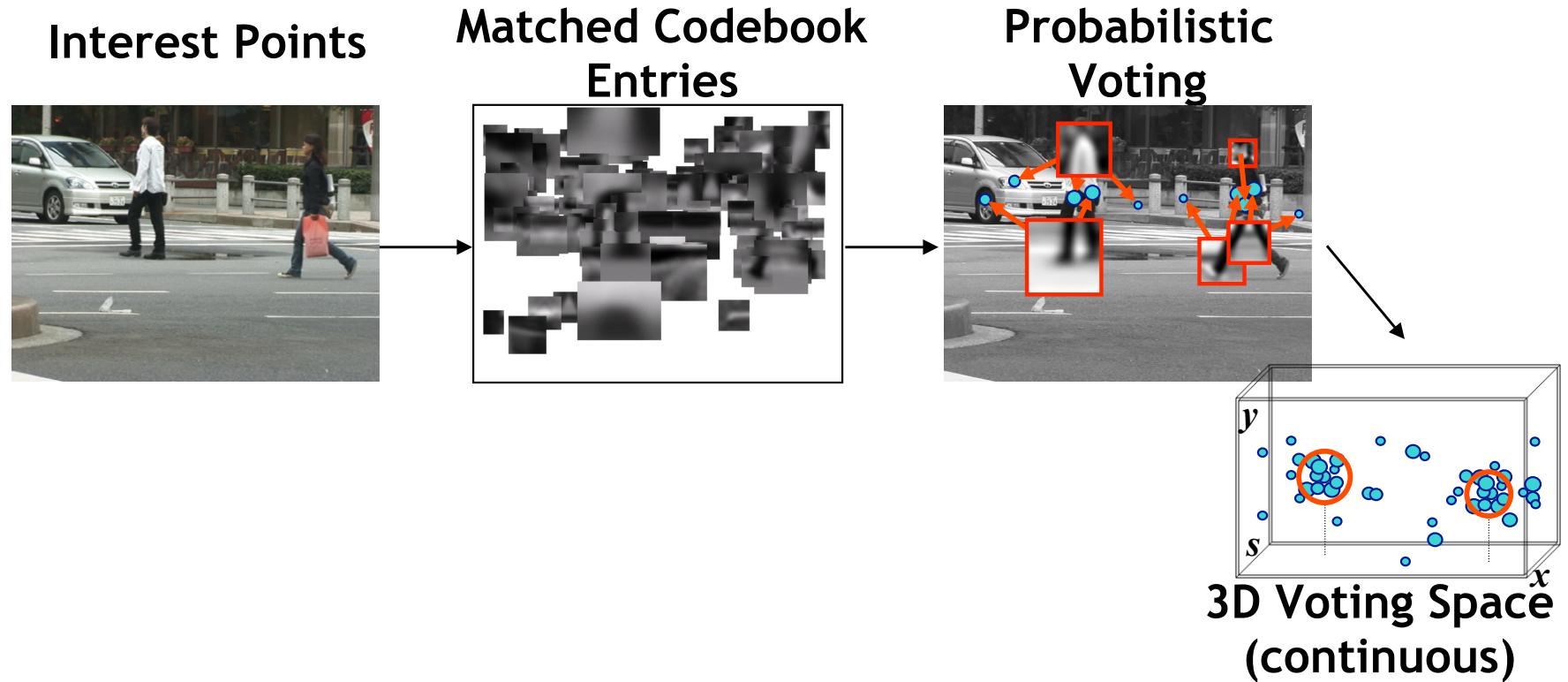
Implicit shape models: Training

1. Build codebook of descriptors around extracted interest points using clustering
2. Map the descriptor at 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]

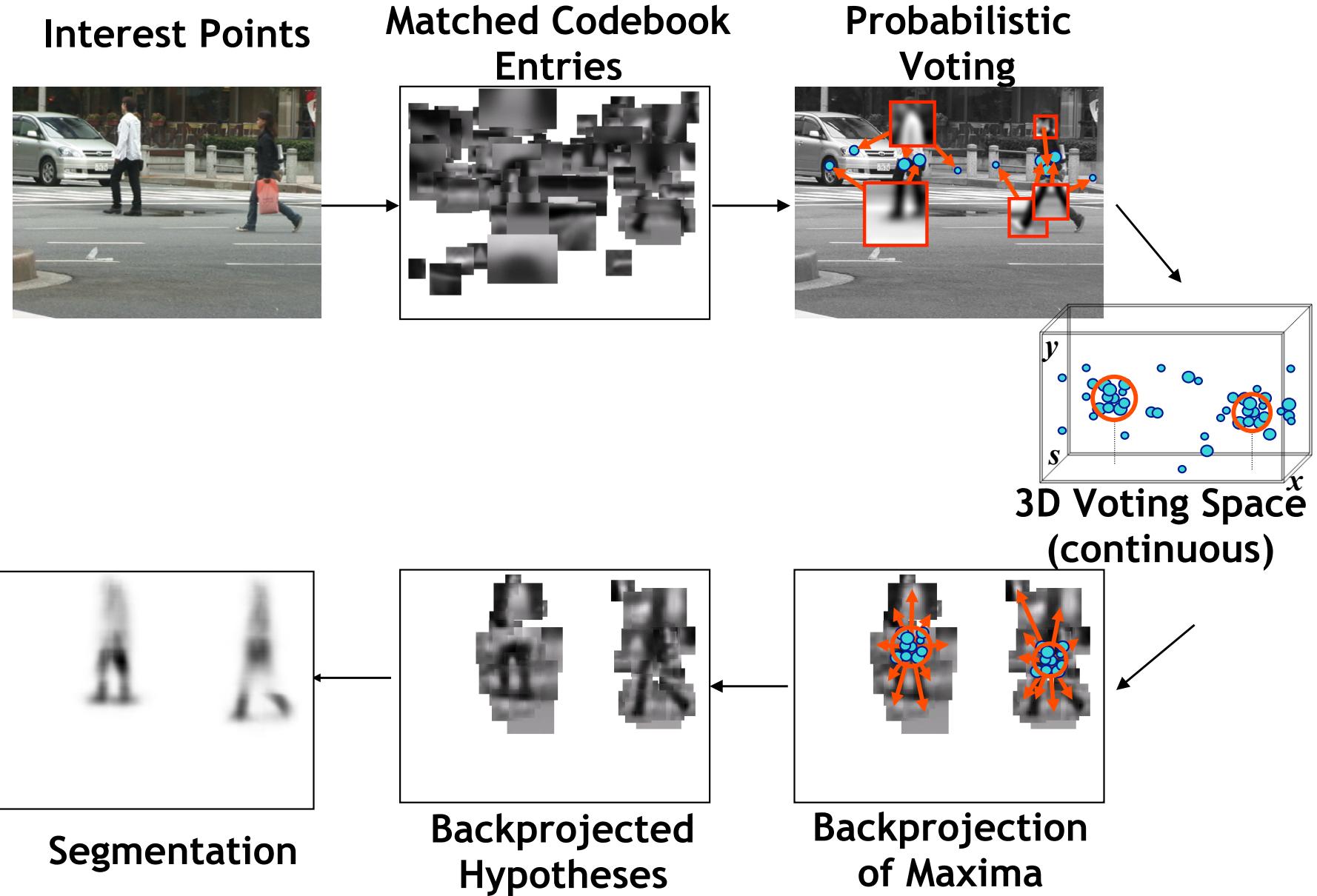


Credit slide: S. Lazebnik

Implicit Shape Model - Recognition



Implicit Shape Model - Recognition



Example: Results on Cows



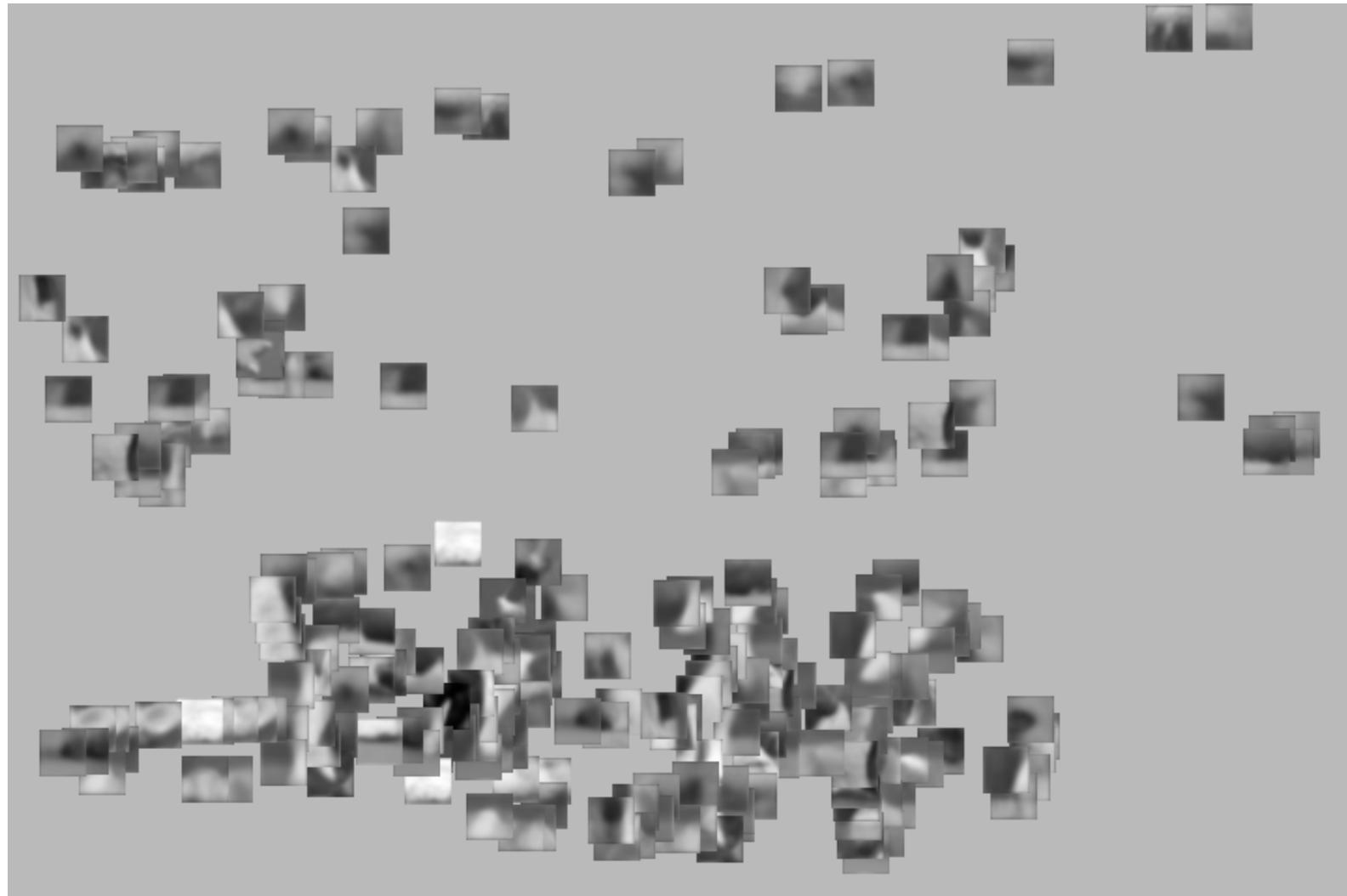
Original image

Example: Results on Cows



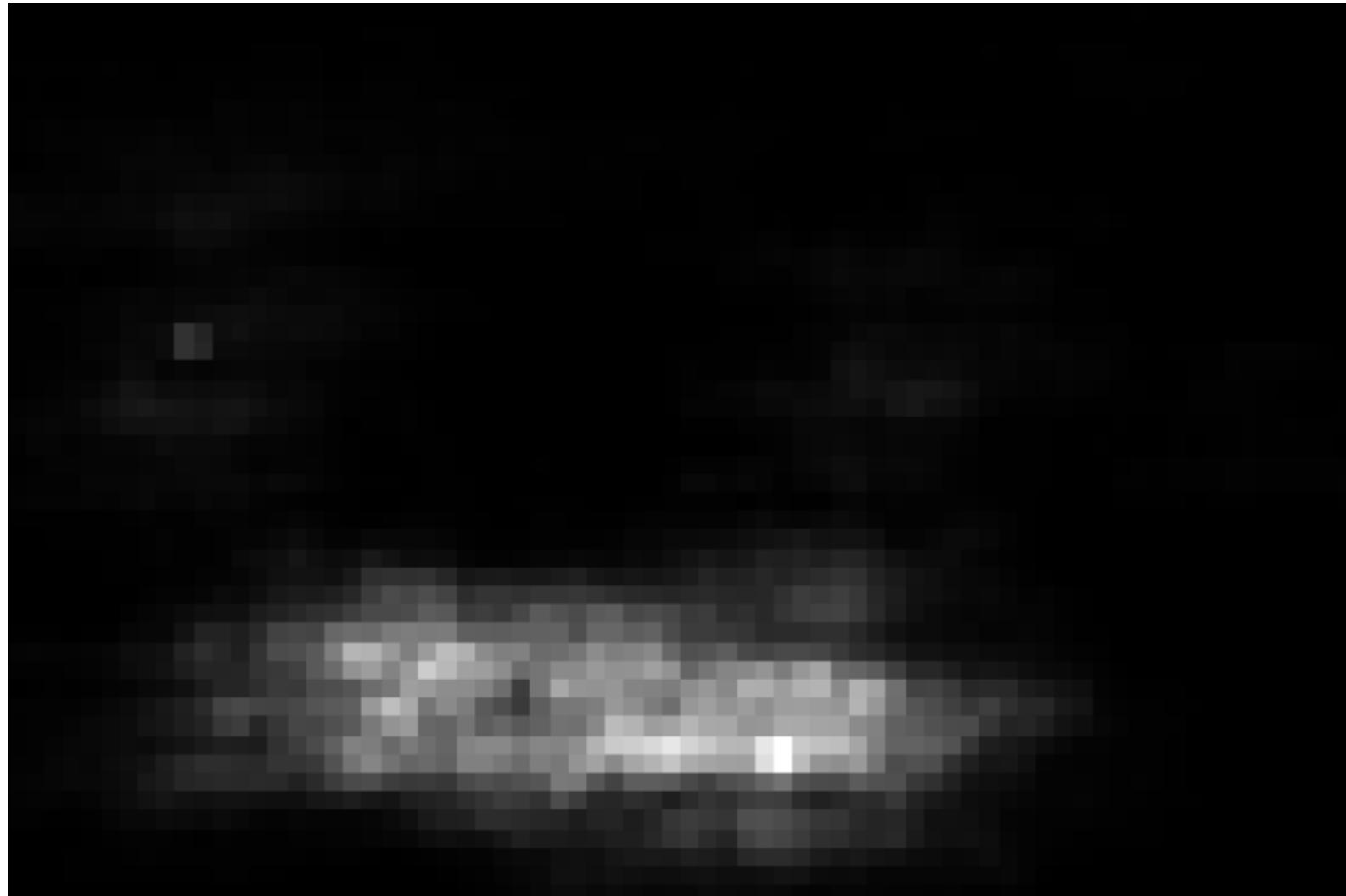
Interest points

Example: Results on Cows



Matched patches

Example: Results on Cows



Prob. Votes

Example: Results on Cows



1st hypothesis

Example: Results on Cows



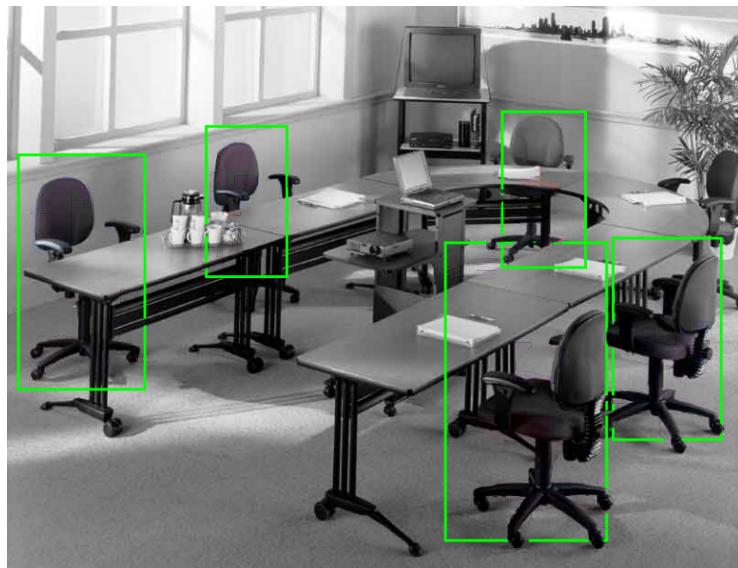
2nd hypothesis

Example: Results on Cows

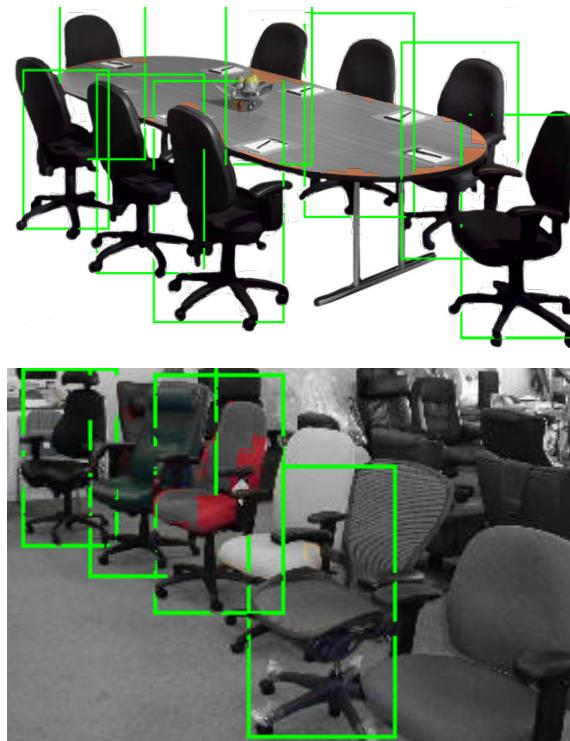


3rd hypothesis

Example Results: Chairs



Office chairs



Dining room chairs



You Can Try It At Home...

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

Many follow up projects:

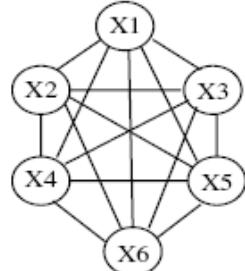
- Subhransu Maji and Jitendra Malik, Object Detection Using a Max-Margin Hough Transform, CVPR 2009
- ...
- Hough-based tracking of non-rigid objects, M Godec, PM Roth, H Bischof, 2013

Conclusions

- Pros:
 - Works well for many different object categories
 - Both rigid and articulated objects
 - Doesn't need a sliding window strategy
 - Learns from relatively few (50-100) training examples
 - Enables joint detection and segmentation
 - Still used today for certain 3D object detection problems
- Cons:
 - Requires supervision
 - Outperformed by CNN-based methods, but require way less training data!

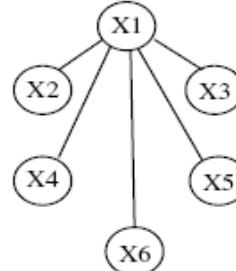
Different connectivity structures

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



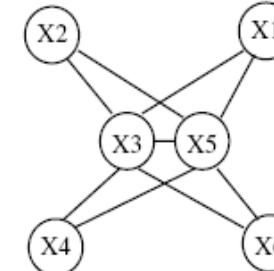
a) Constellation [13]

Crandall et al. '05
Leibe 05; Felzenszwalb 09



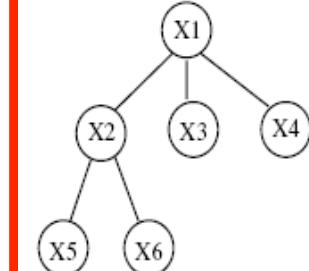
b) Star shape [9, 14]

Crandall et al. '05

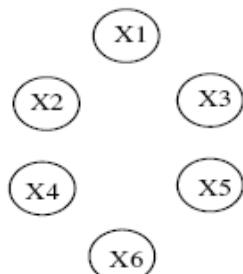


c) k -fan ($k = 2$) [9]

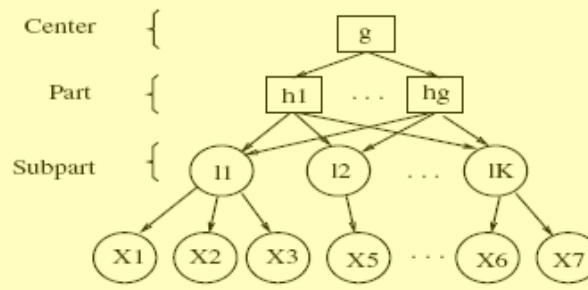
Felzenszwalb & Huttenlocher '00



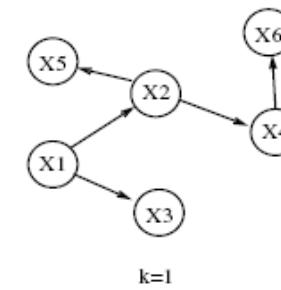
d) Tree [12]



e) Bag of features [10, 21]
Csurka '04
Vasconcelos '00



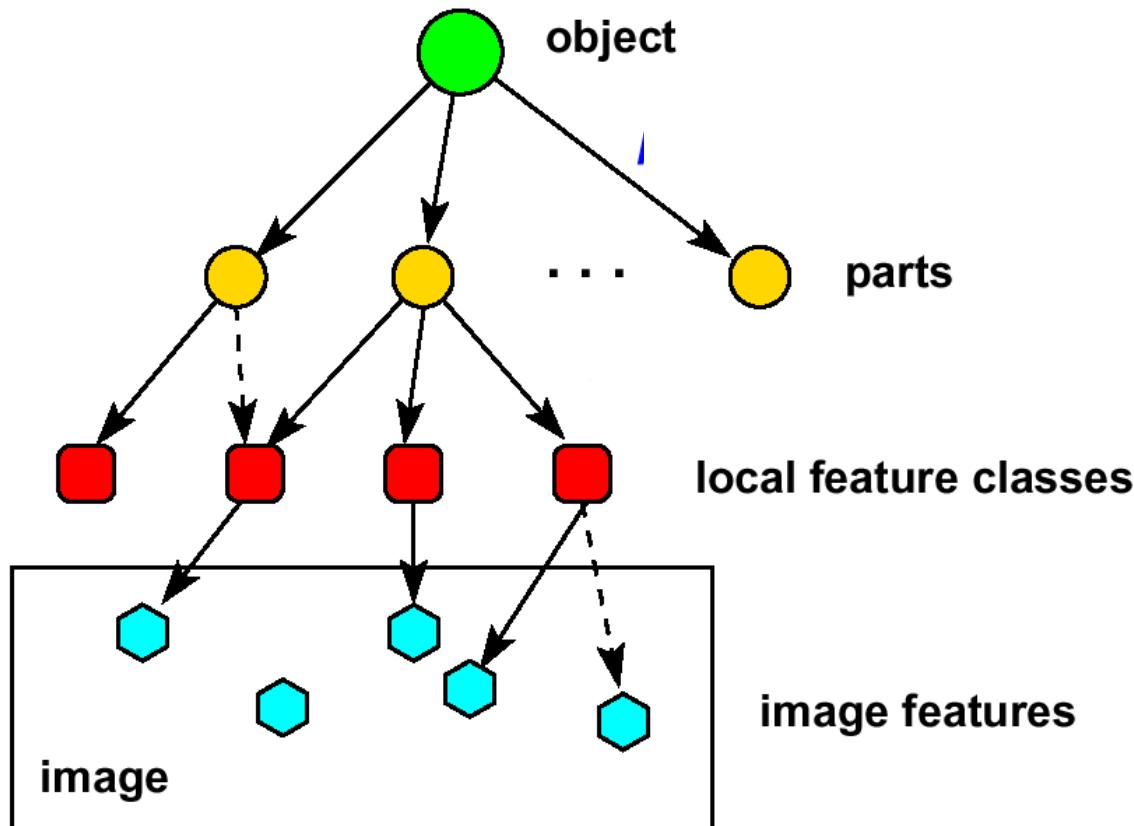
f) Hierarchy [4]
Bouchard & Triggs '05



g) Sparse flexible model
Carneiro & Lowe '06

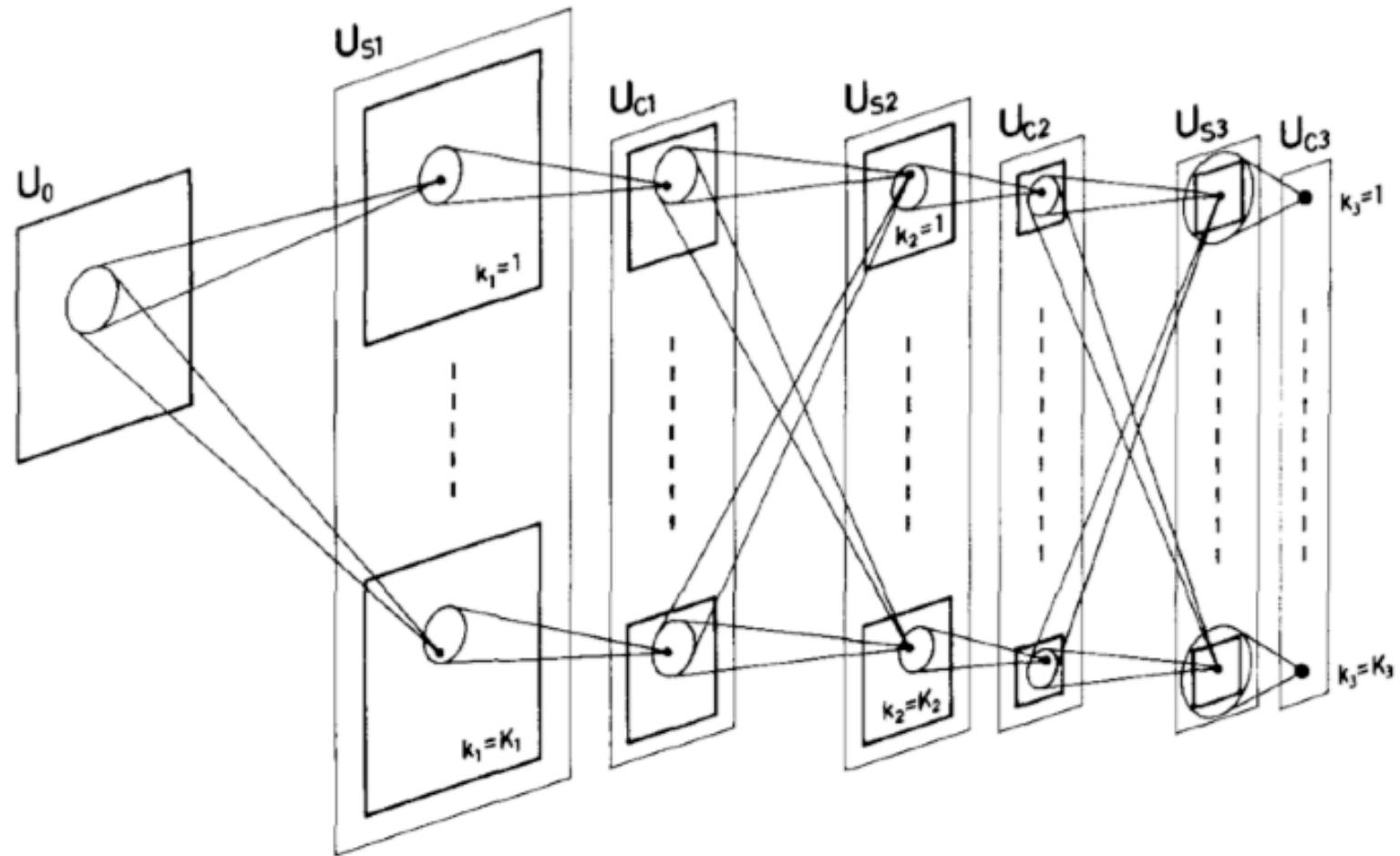
Hierarchical representations

- Pixels → Pixel groupings → Parts → Object



Hierarchical representations

- Deep learning architectures and ConvNets

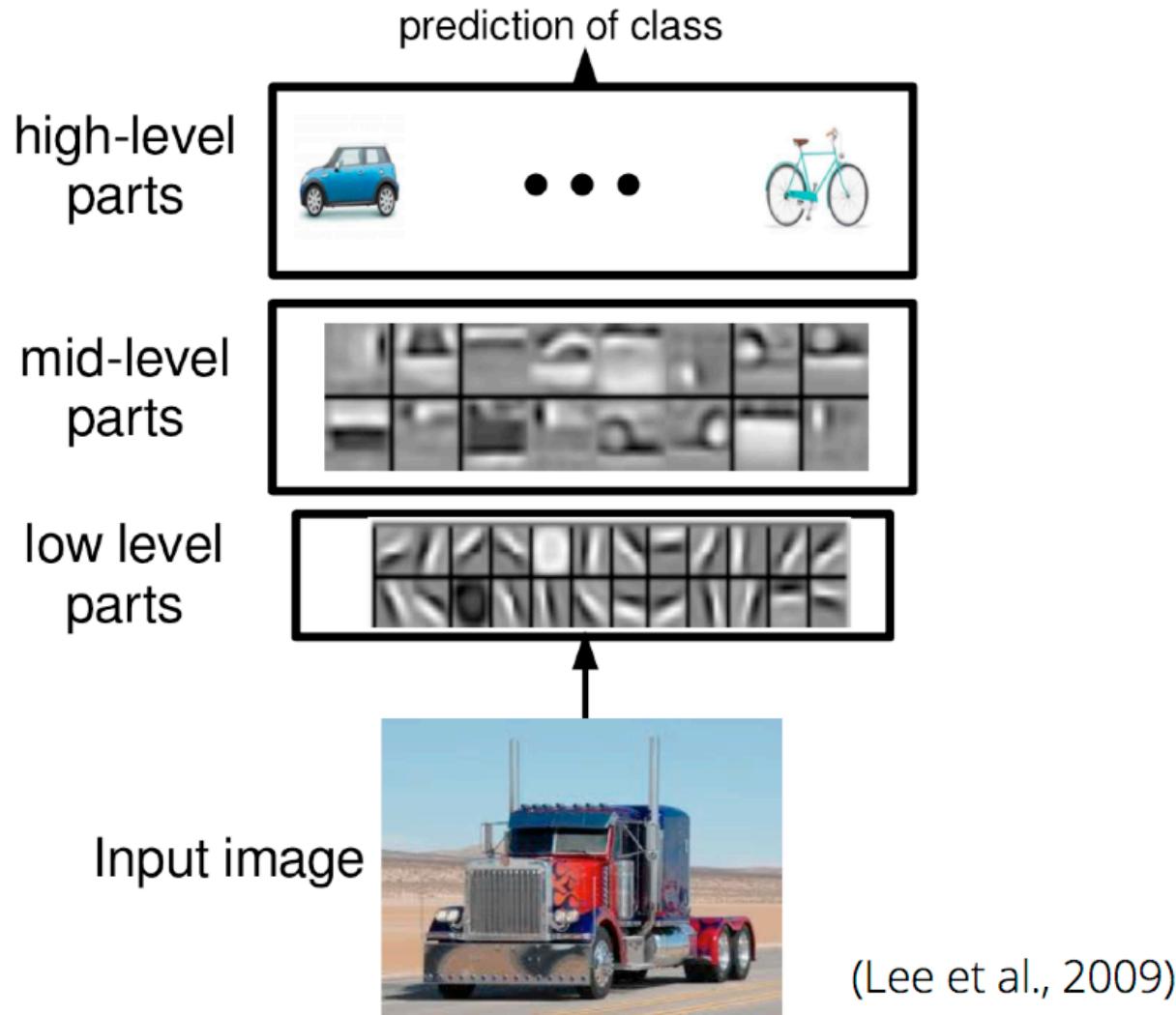


Fukushima, 1980
LeCun, 1987

Hierarchical representations

- Deep learning architectures and ConvNets

More on
lecture 15!



Next lecture (Wed 5/11)

- 2D scene understanding

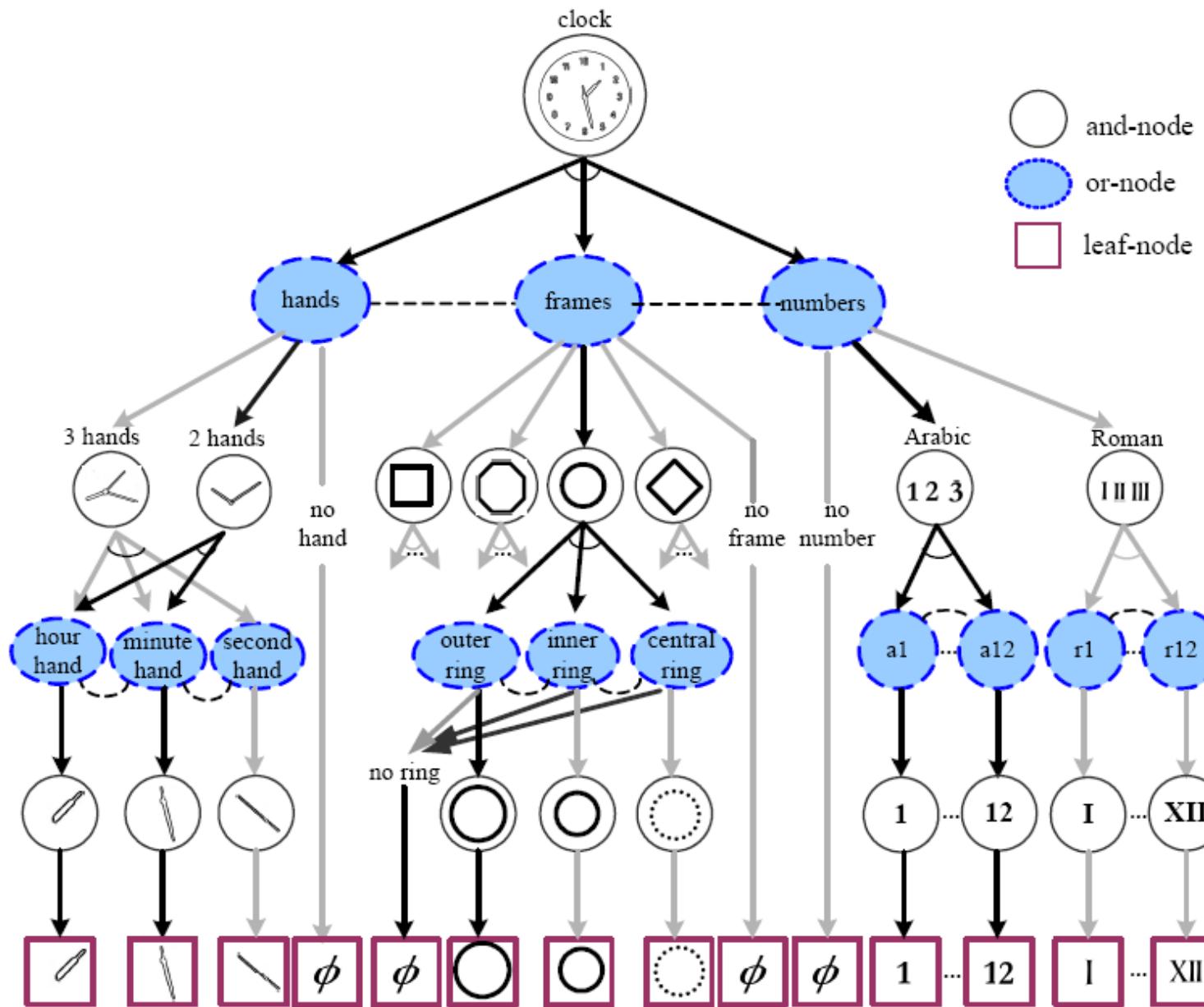
Appendix

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

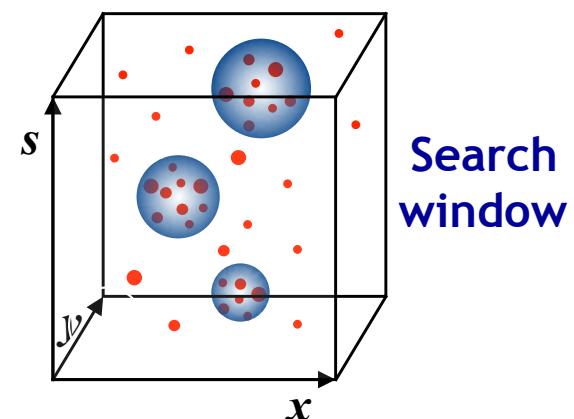
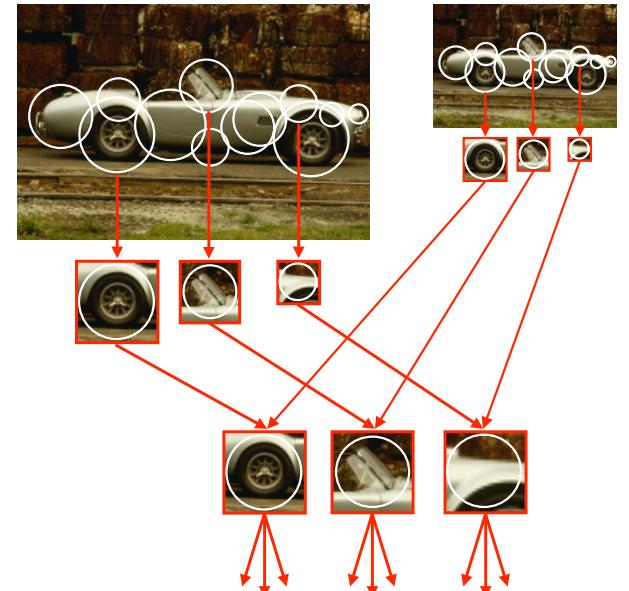
Hierarchical representations

S.C. Zhu et al. and D. Mumford



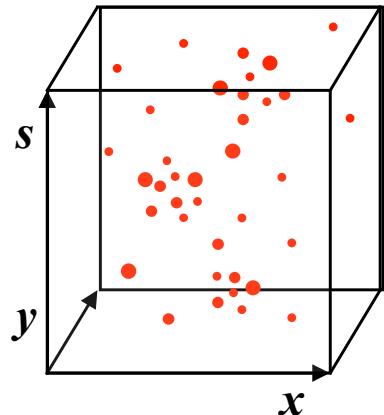
Scale Invariant Voting

- Scale-invariant feature selection
 - Scale-invariant interest points
 - Rescale extracted patches
 - Match to constant-size codebook
- Generate scale votes
 - Scale as 3rd dimension in voting space
 - Search for maxima in 3D voting space

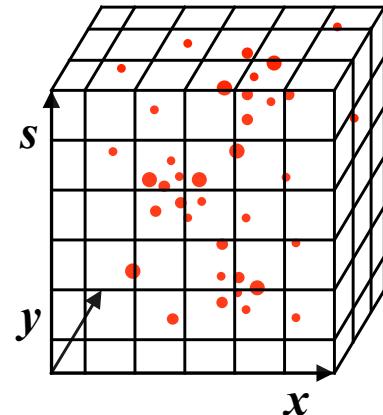


Source: Bastian Leibe

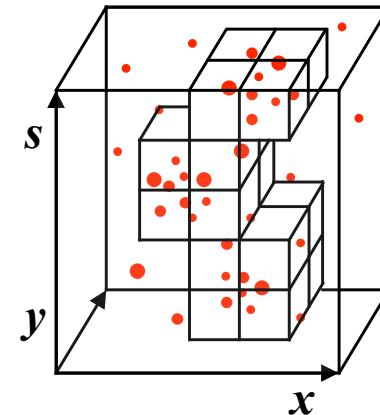
Scale Voting: Efficient Computation



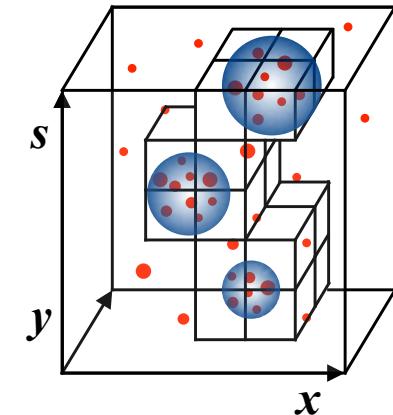
Scale votes



Binned accum. array



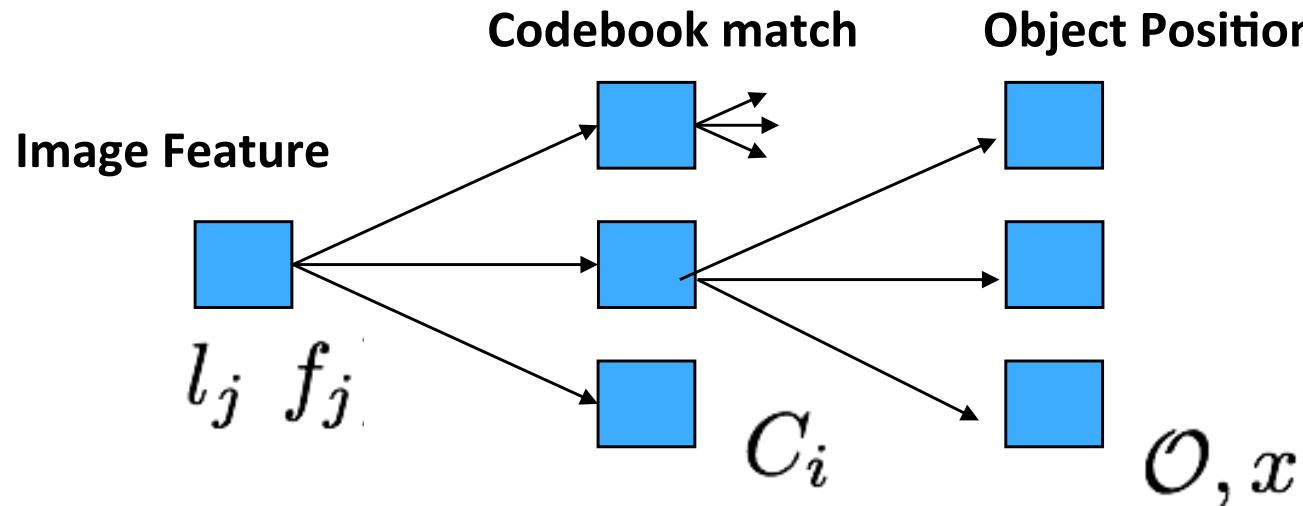
Candidate maxima



Refinement (Mean-Shift)

- Continuous Generalized Hough Transform
 - Binned accumulator array similar to standard Gen. Hough Transf.
 - Quickly identify candidate maxima locations
 - Refine locations by Mean-Shift search only around those points
⇒ Avoid quantization effects by keeping exact vote locations.
 - ⇒ Mean-shift interpretation as kernel prob. density estimation.

Probabilistic Hough Transform



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

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

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

Detection Score

Position Posterior
 distribution of the centroid given the Codeword C_i observed at location l_j .

Codeword Match

confidence (or weight) of the codeword C_i .

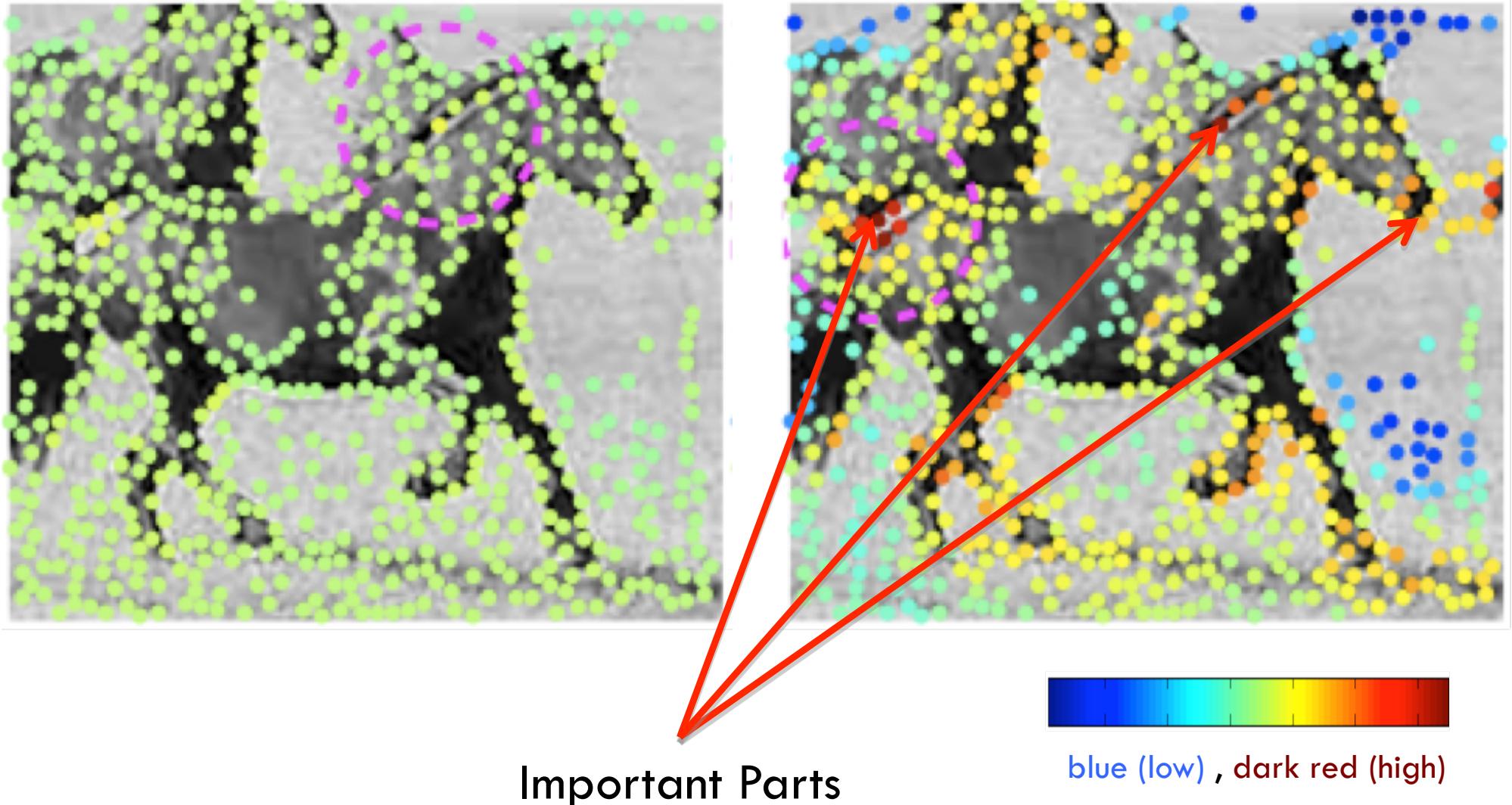
Learnt using a max margin formulation

Maji et al , CVPR 2009

Extension: Learning Feature Weights

Subhransu Maji and Jitendra Malik, Object Detection Using a Max-Margin Hough Transform, CVPR 2009

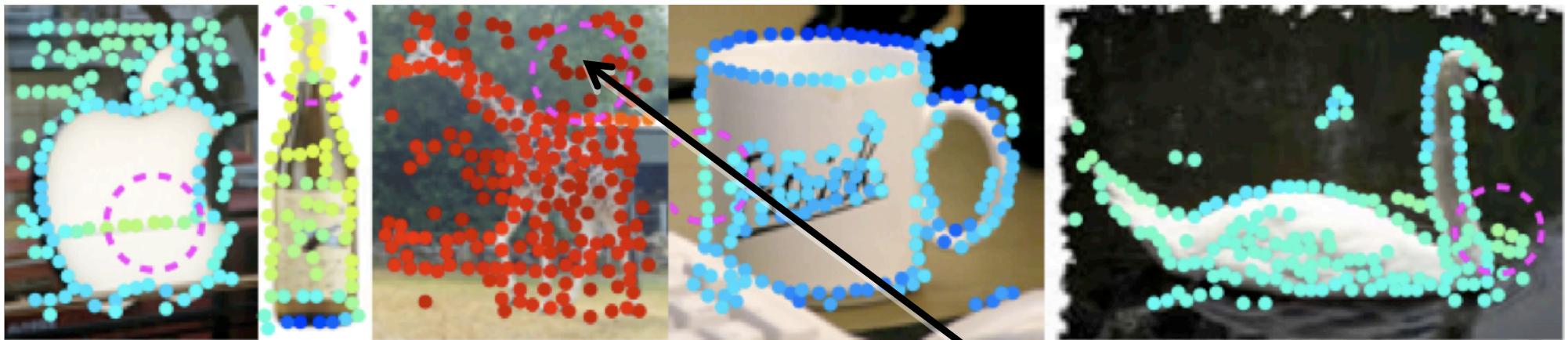
Weights can be learned optimally using a max-margin framework



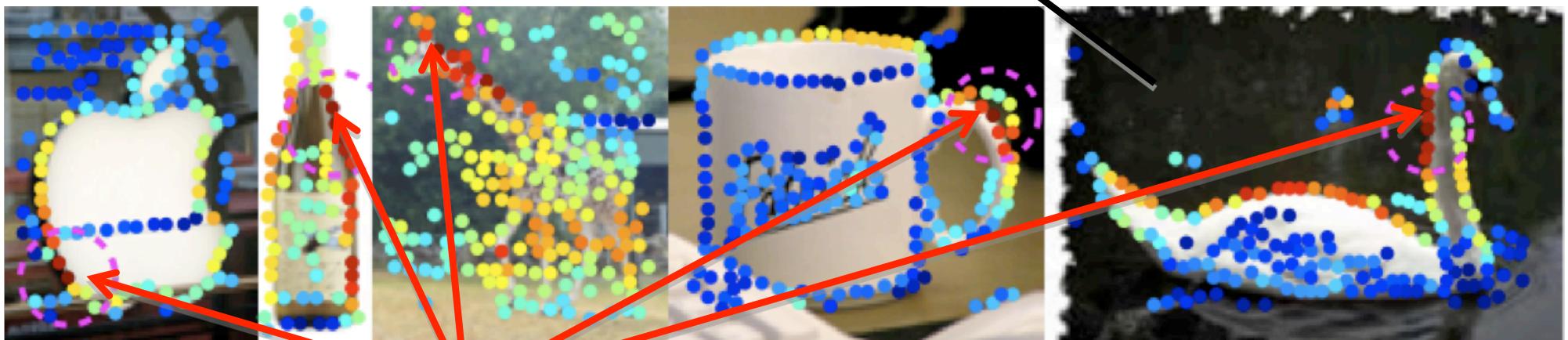
Extension: Learning Feature Weights

Subhransu Maji and Jitendra Malik, Object Detection Using a Max-Margin Hough Transform, CVPR 2009

Naïve Bayes



Max-Margin



Important Parts



blue (low) , dark red (high)

Detection Results (ETHZ dataset)

Recall @ 1.0 False Positives Per Window

Category	Uniform	Naive Bayes	Max-margin
Applelogos	70.0	70.0	85.0
Bottles	62.5	71.4	67.0
Giraffes	47.1	47.1	55.0
Mugs	35.5	35.5	55.0
Swans	47.1	47.1	42.5
Average	52.4	54.2	60.9