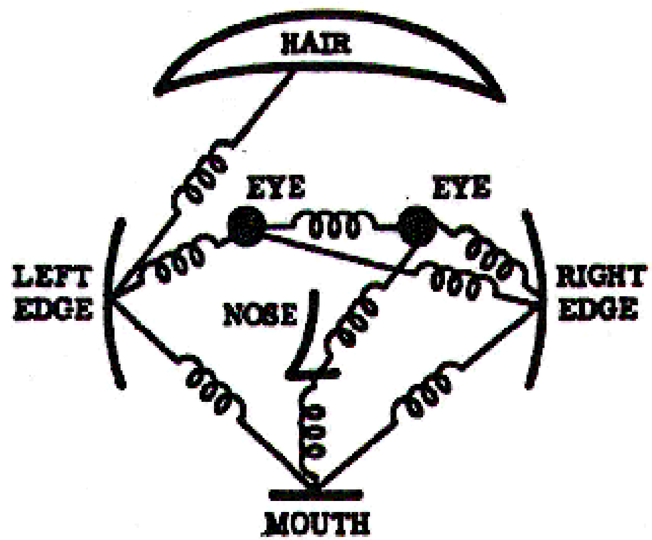


Part-based models

Lecture 10

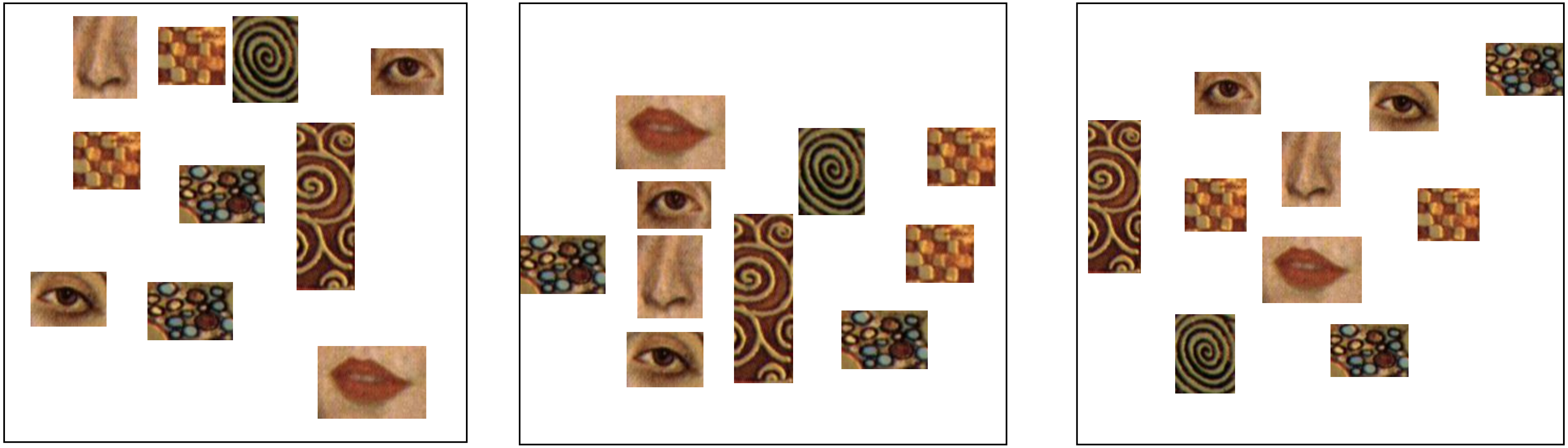
Overview

- Representation
  - Location
  - Appearance
  - Generative interpretation
- Learning
- Distance transforms
- Other approaches using parts
- Felzenszwalb, Girshick, McAllester, Ramanan CVPR 2008

Overview

- Representation
  - Location
  - Appearance
  - Generative interpretation
- Learning
- Distance transforms
- Other approaches using parts
- Felzenszwalb, Girshick, McAllester, Ramanan CVPR 2008

Problem with bag-of-words



- All have equal probability for bag-of-words methods
- Location information is important

Model: Parts and Structure

Representation

- Object as set of parts
  - Generative representation

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

- Issues:
  - How to model location
  - How to represent appearance
  - Sparse or dense (pixels or regions)
  - How to handle occlusion/clutter

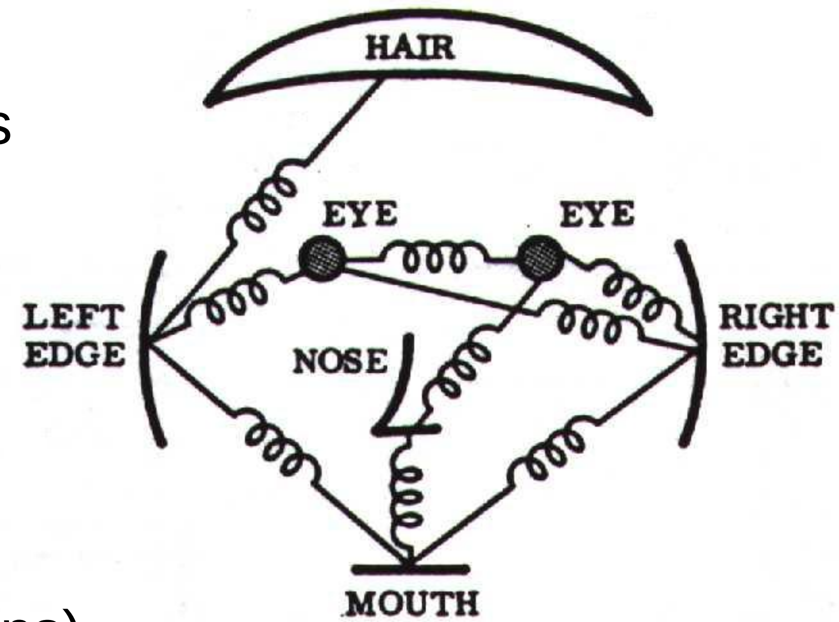
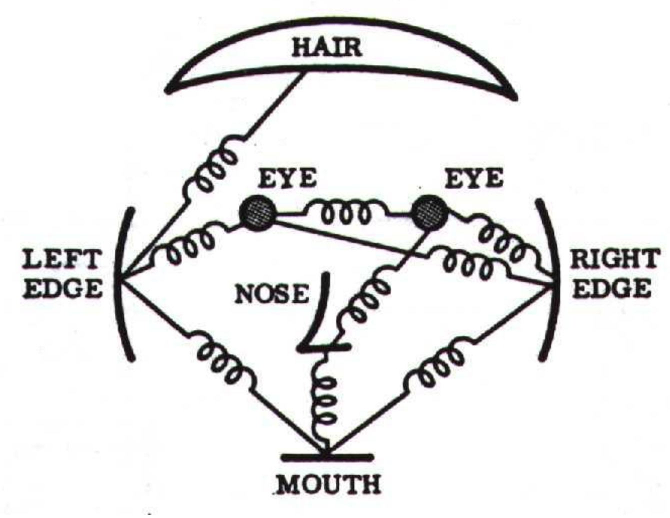


Figure from [Fischler & Elschlager 73]

History of Parts and Structure approaches

- Fischler & Elschlager 1973

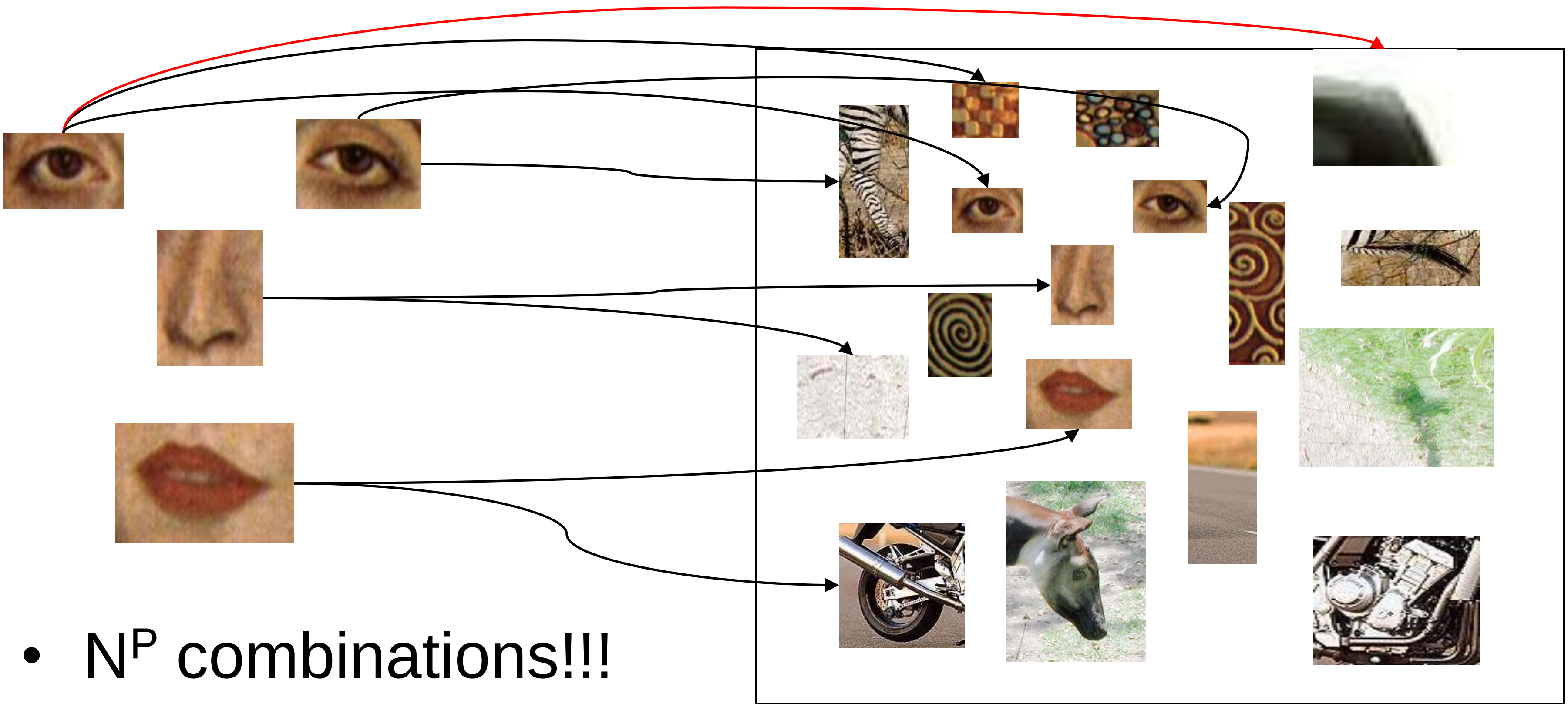


- Yuille ‘91
- Brunelli & Poggio ‘93
- Lades, v.d. Malsburg et al. ‘93
- Cootes, Lanitis, Taylor et al. ‘95
- Amit & Geman ‘95, ‘99
- Perona et al. ‘95, ‘96, ’98, ’00, ’03, ‘04, ‘05
- Felzenszwalb & Huttenlocher ’00, ’04
- Crandall & Huttenlocher ’05, ’06
- Leibe & Schiele ’03, ’04

- Many papers since 2000

The correspondence problem

- Model with P parts
- Image with N possible locations for each part



- N^P combinations!!!

Sparse representation

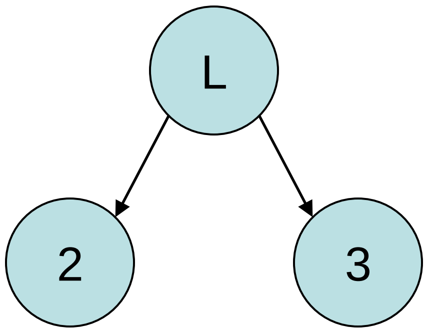
+ Computationally tractable (10^5 pixels → 10^1 -- 10^2 parts)
+ Generative representation of class
+ Avoid modeling global variability
+ Success in specific object recognition

- Throw away most image information
- Parts need to be distinctive to separate from other classes

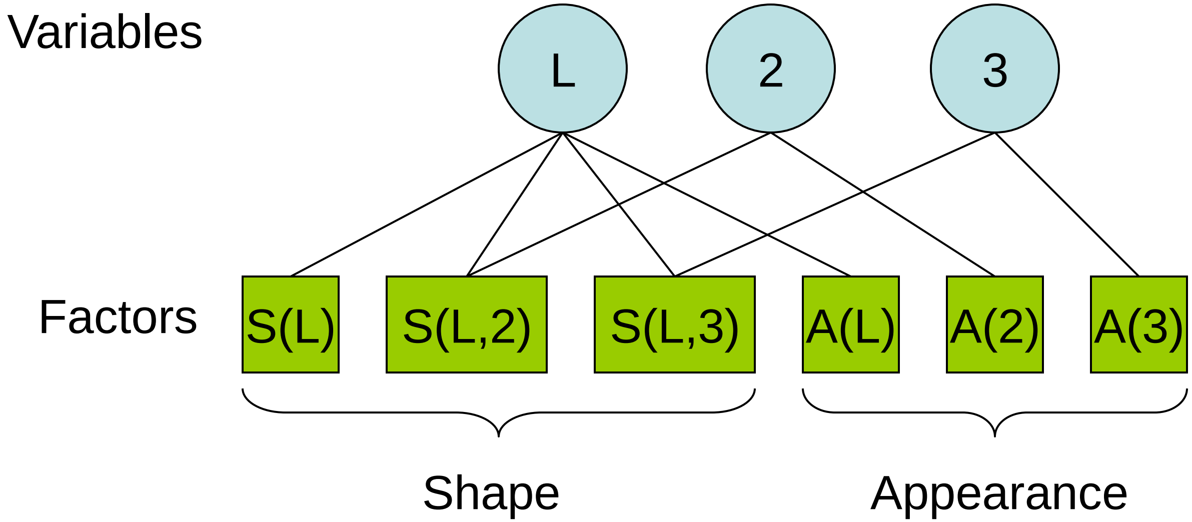
Connectivity of parts

- Complexity is given by size of maximal clique in graph
- Consider a 3 part model
  - Each part has set of N possible locations in image
  - Location of parts 2 & 3 is independent, given location of L
  - Each part has an appearance term, independent between parts.

Shape Model



Factor graph



Different connectivity structures

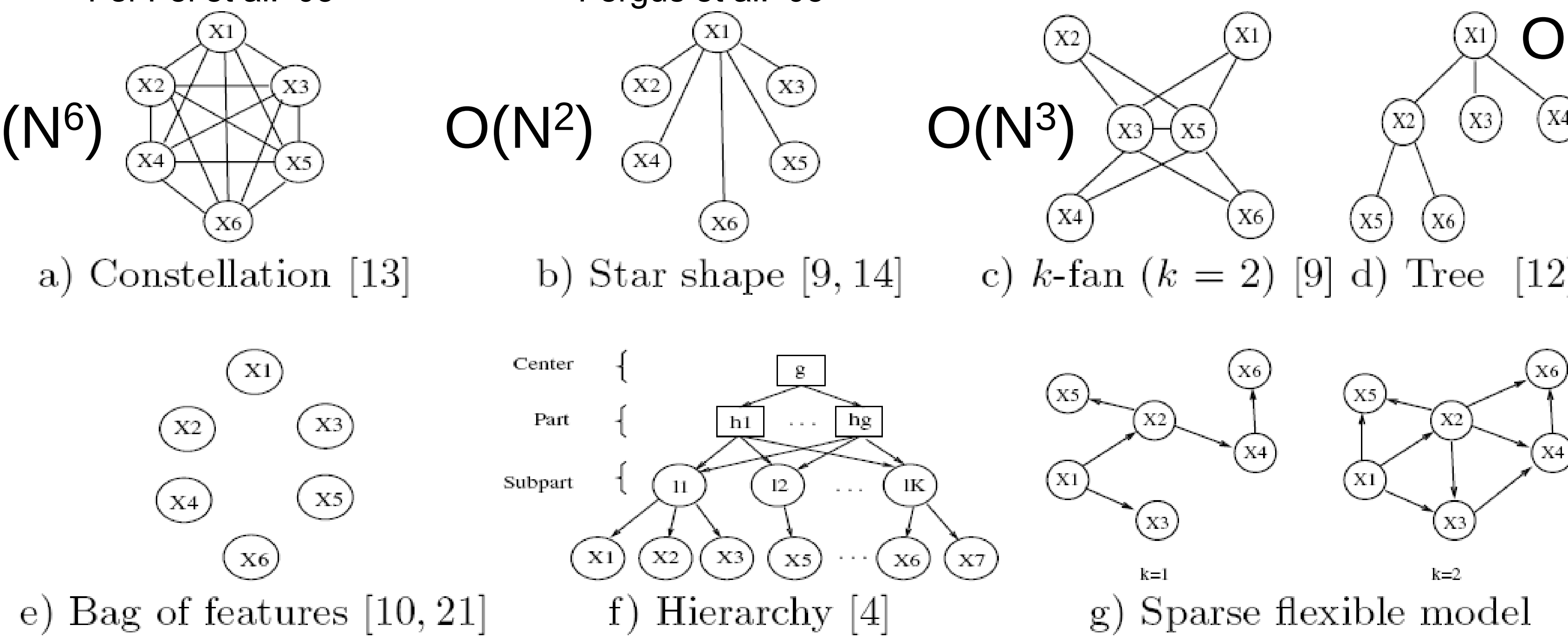
Fergus et al. ’03
Fei-Fei et al. ‘03

Crandall et al. ‘05
Fergus et al. ’05

Crandall et al. ‘05

Felzenszwalb &
Huttenlocher ‘00

$O(N^6)$ $O(N^2)$ $O(N^3)$ $O(N^2)$



Csurka ’04
Vasconcelos ‘00

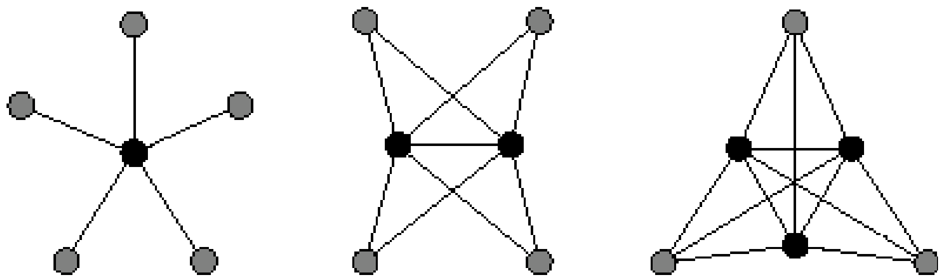
Bouchard & Triggs ‘05

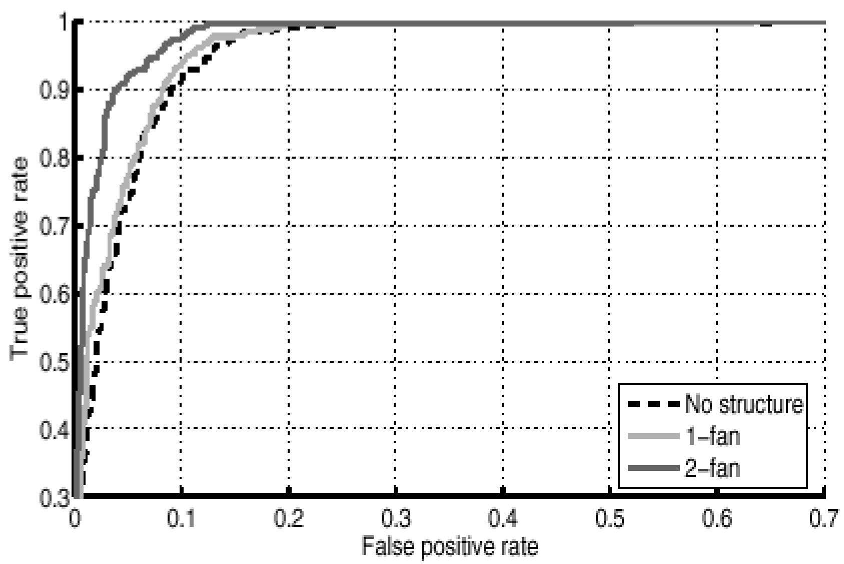
Carneiro & Lowe ‘06

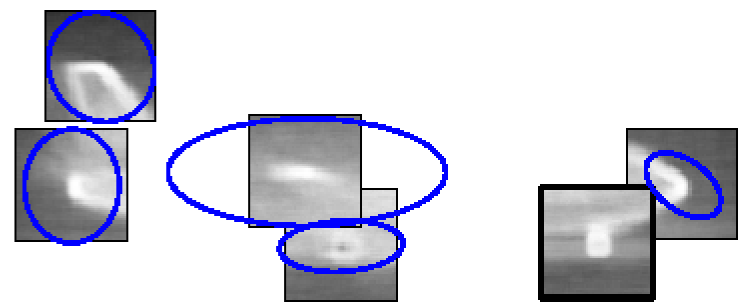
from Sparse Flexible Models of Local Features
Gustavo Carneiro and David Lowe, ECCV 2006

How much does shape help?

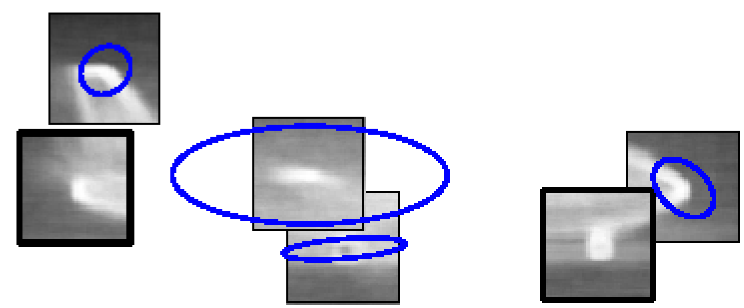
- Crandall, Felzenszwalb, Huttenlocher CVPR'05
- Shape variance increases with increasing model complexity
- Do get some benefit from shape







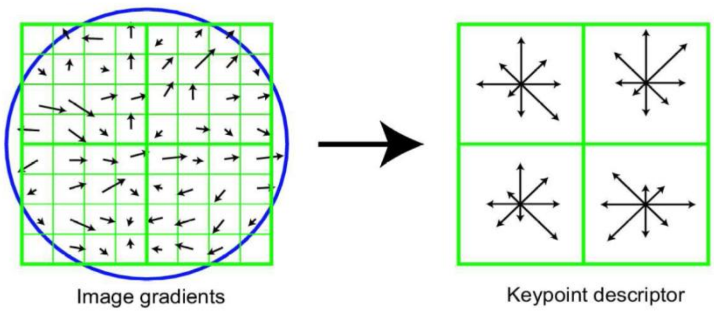
(a) Airplane, 1-fan



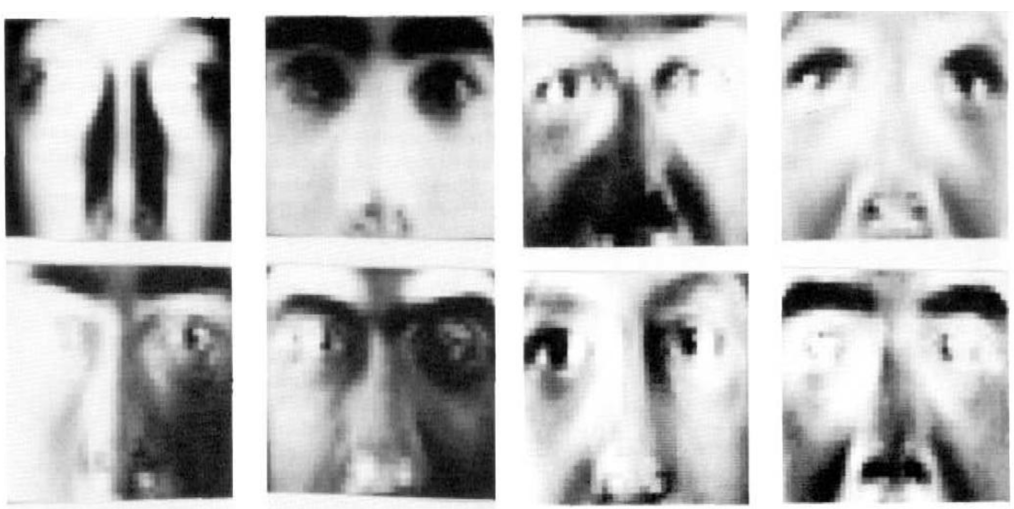
(b) Airplane, 2-fan

Appearance representation

- SIFT



- PCA



- Decision trees

[Lepetit and Fua CVPR 2005]

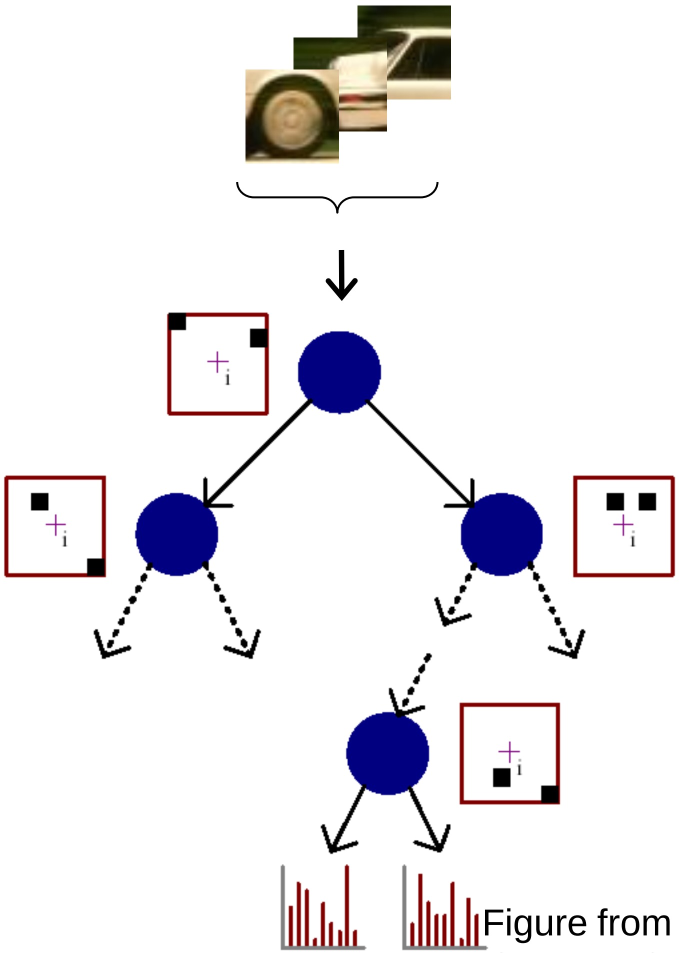


Figure from Winn & Shotton, CVPR '06

Overview

- Representation
  - Location
  - Appearance
  - Generative interpretation
- Learning
- Distance transforms
- Other approaches using parts
- Felzenszwalb, Girshick, McAllester, Ramanan CVPR 2008

Generative part-based models

R. Fergus, P. Perona and A. Zisserman, **Object Class Recognition by Unsupervised Scale-Invariant Learning**, CVPR 2003

Probabilistic model

$$P(image \mid object) = P(appearance, shape \mid object)$$

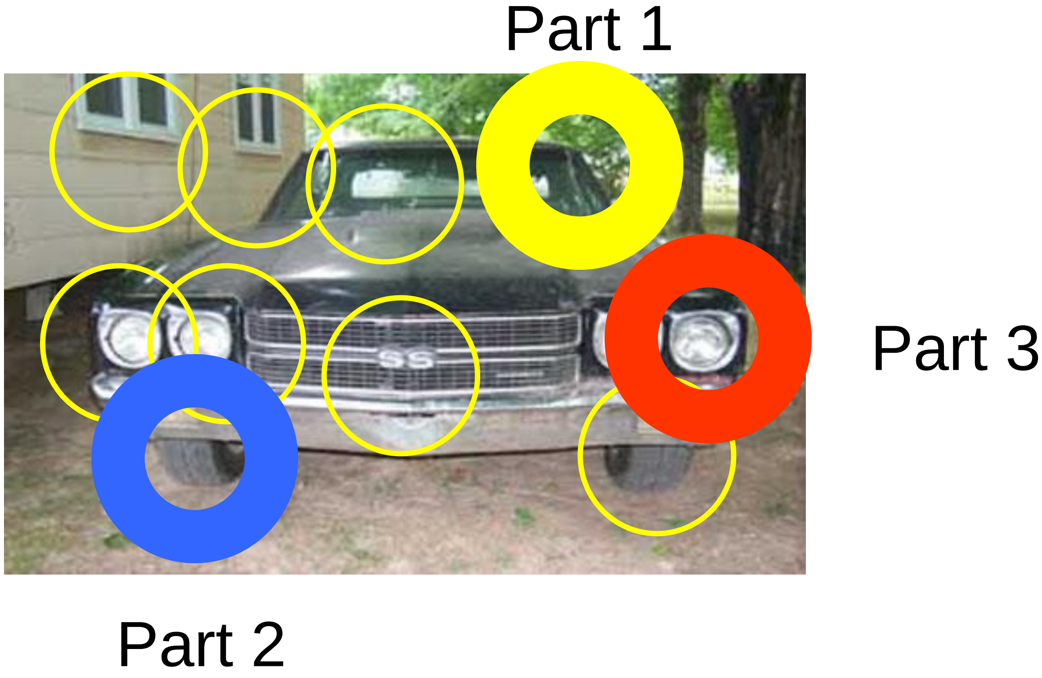
Part descriptors

Part locations

Candidate parts

Probabilistic model

$$P(image \mid object) = P(appearance, shape \mid object)$$



Probabilistic model

$$P(image \mid object) = P(appearance, shape \mid object)$$

$$= \max_h P(appearance \mid h, object)\, p(shape \mid h, object)\, p(h \mid object)$$

h: assignment of features to parts

Part 1

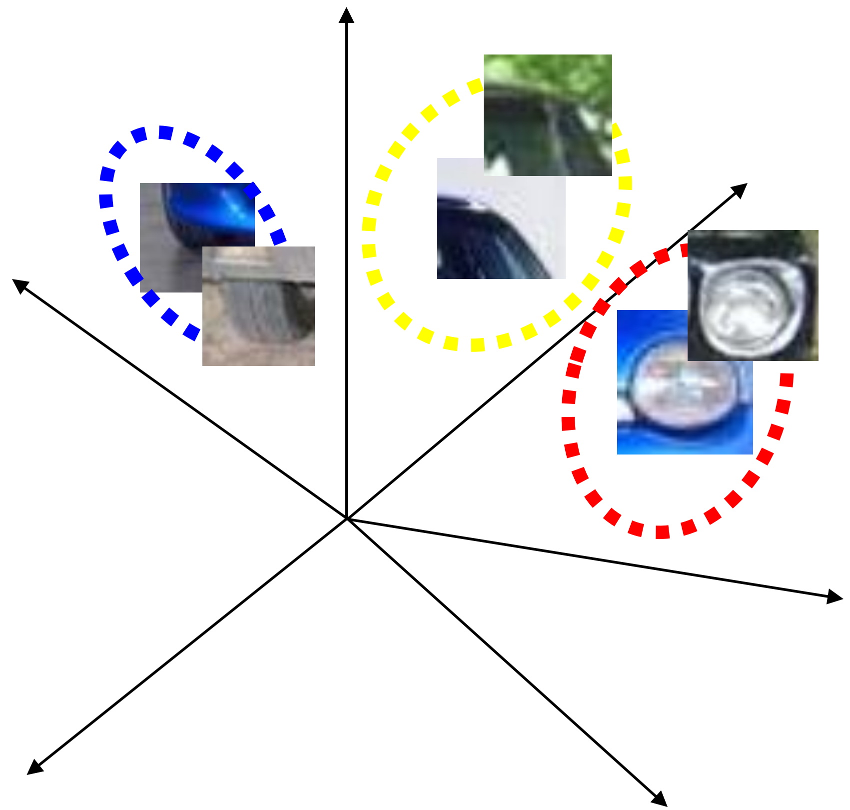
Part 3

Part 2

Probabilistic model

$$P(image \mid object) = P(appearance, shape \mid object)$$

$$= \max_h \boxed{P(appearance \mid h, object)}\, p(shape \mid h, object)\, p(h \mid object)$$



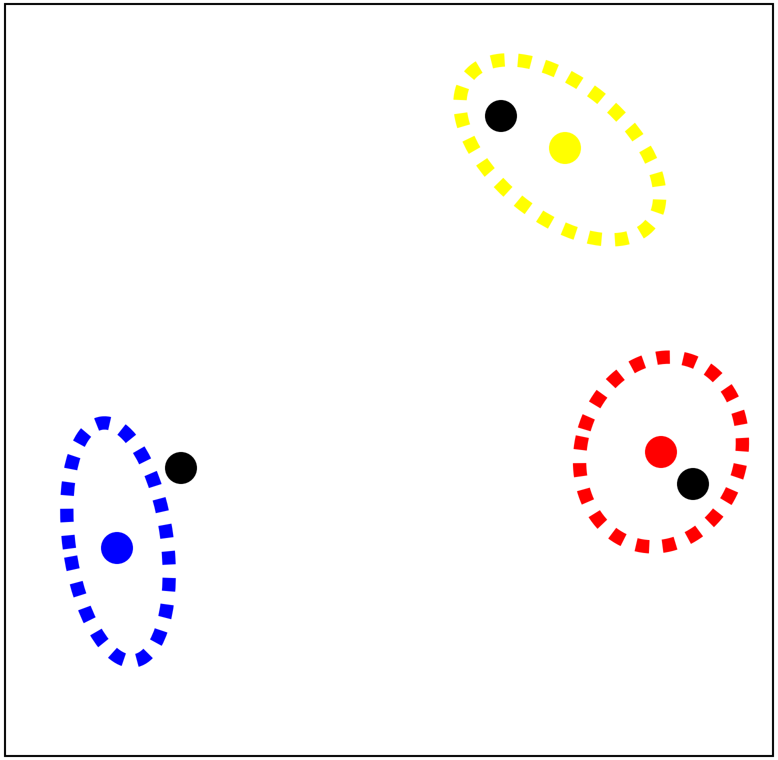
Distribution over patch descriptors

High-dimensional appearance space

Probabilistic model

$$P(image \mid object) = P(appearance, shape \mid object)$$

$$= \max_h P(appearance \mid h, object)\, \boxed{p(shape \mid h, object)}\, p(h \mid object)$$



2D image space

Distribution over joint part positions

Overview

Representation

- Location
- Appearance
- Generative interpretation

Learning

Distance transforms

Other approaches using parts

Felzenszwalb, Girshick, McAllester, Ramanan
CVPR 2008

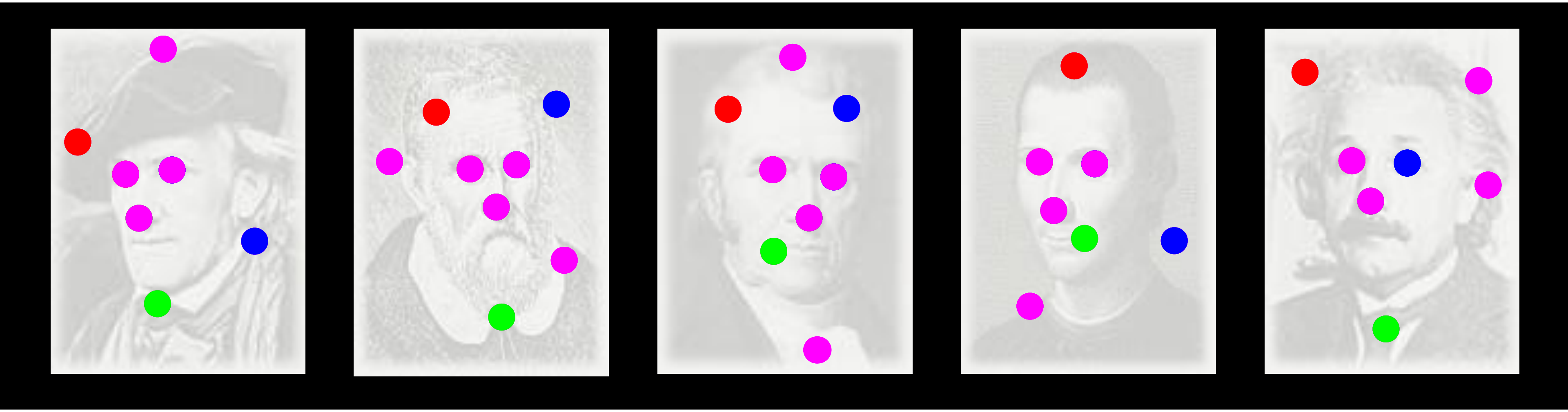
Learning procedure

- Find regions & their location & appearance
- Initialize model parameters
- Use EM and iterate to convergence:

  E-step: Compute assignments for which regions belong to which part

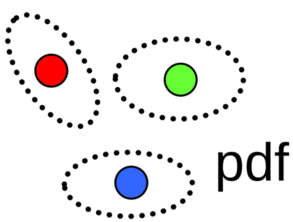
  M-step: Update model parameters

- Trying to maximize likelihood – consistency in shape & appearance

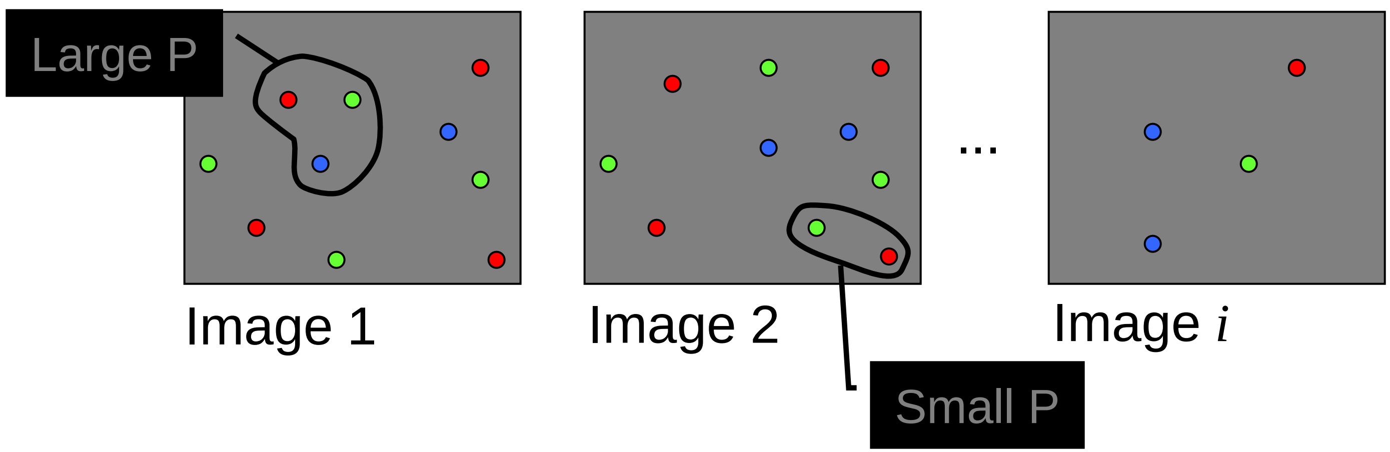


Example scheme, using EM for maximum likelihood learning

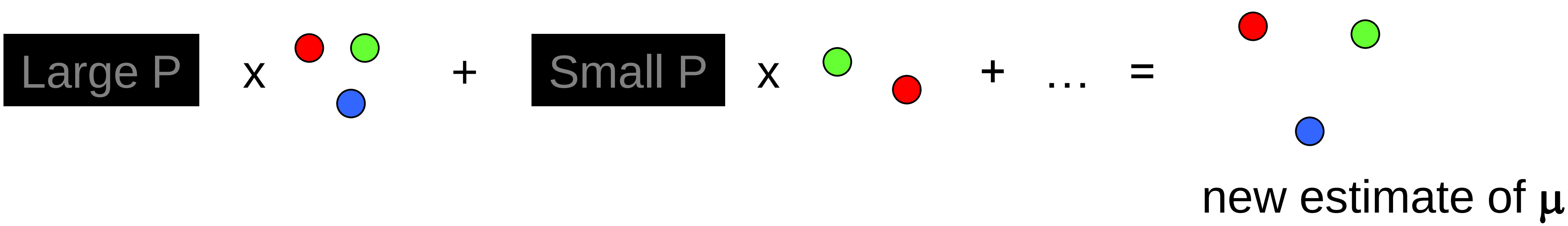
1. Current estimate of θ



2. Assign probabilities to constellations

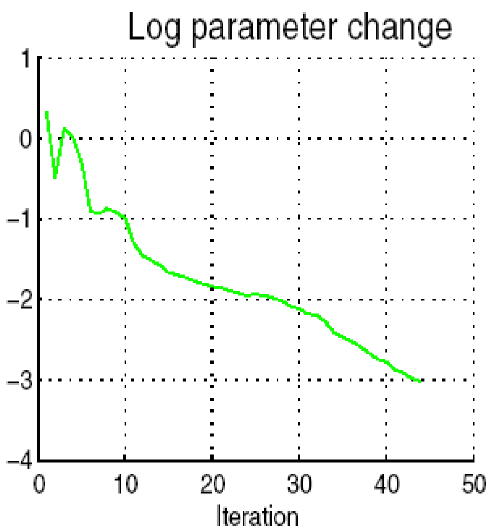


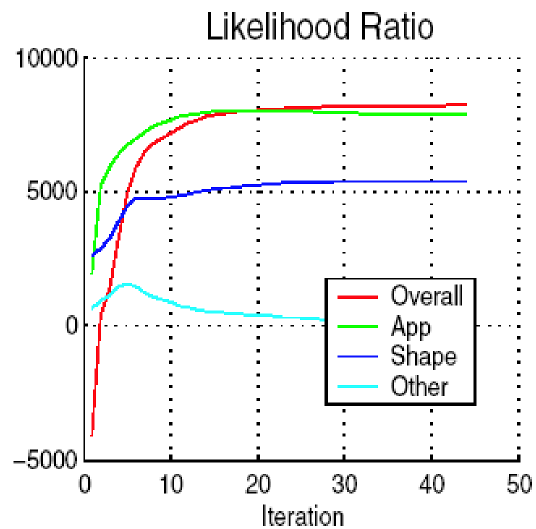
3. Use probabilities as weights to re-estimate parameters. Example: μ

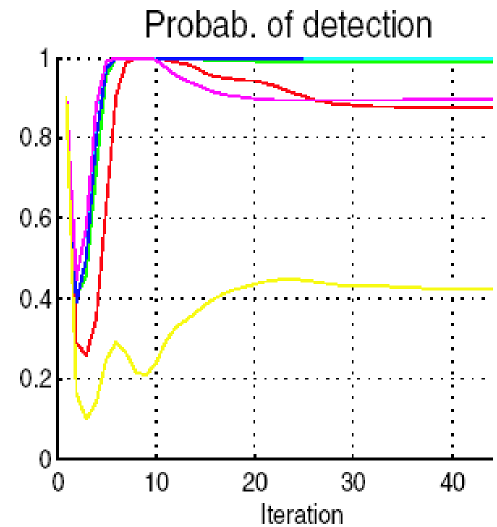


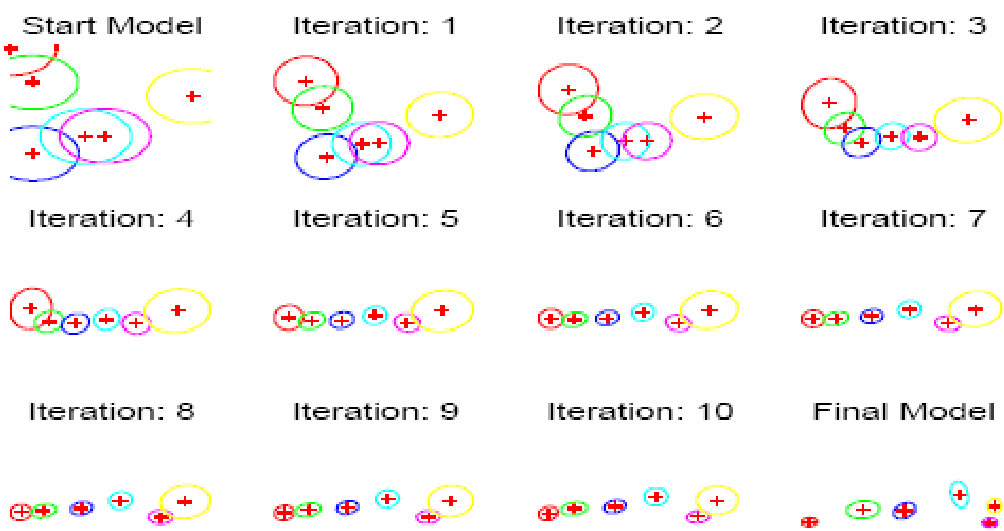
Learning Shape & Appearance simultaneously

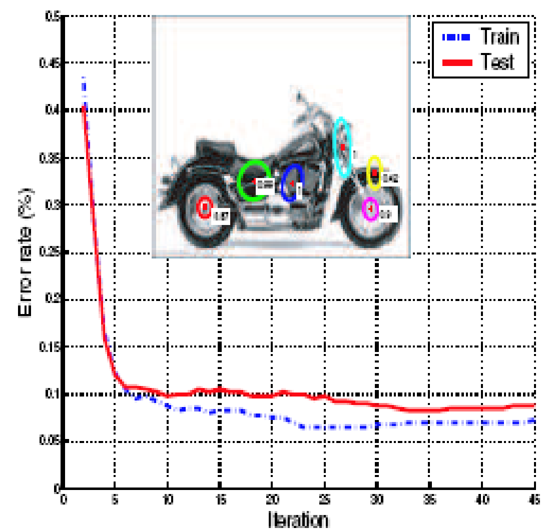
Fergus et al. '03





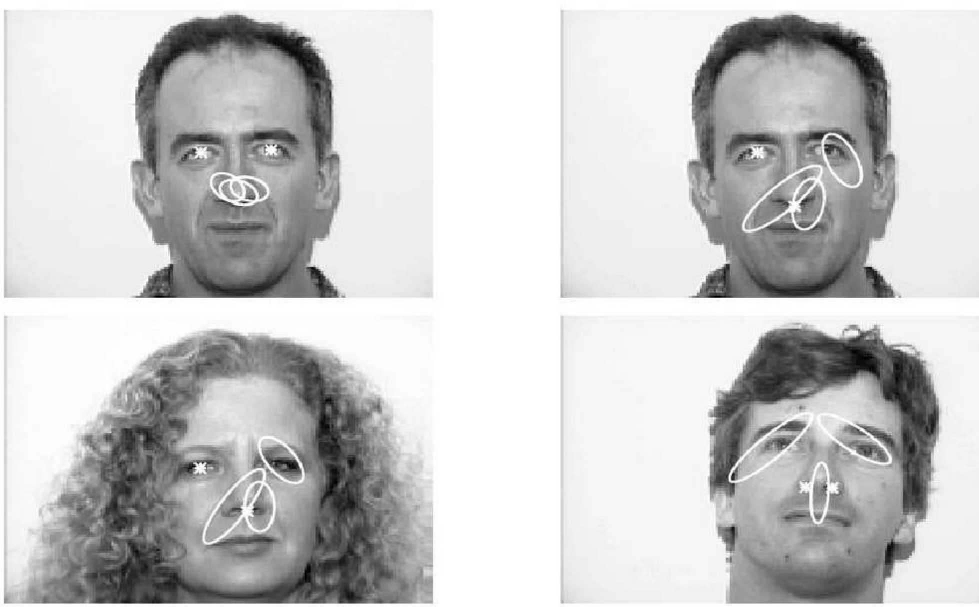


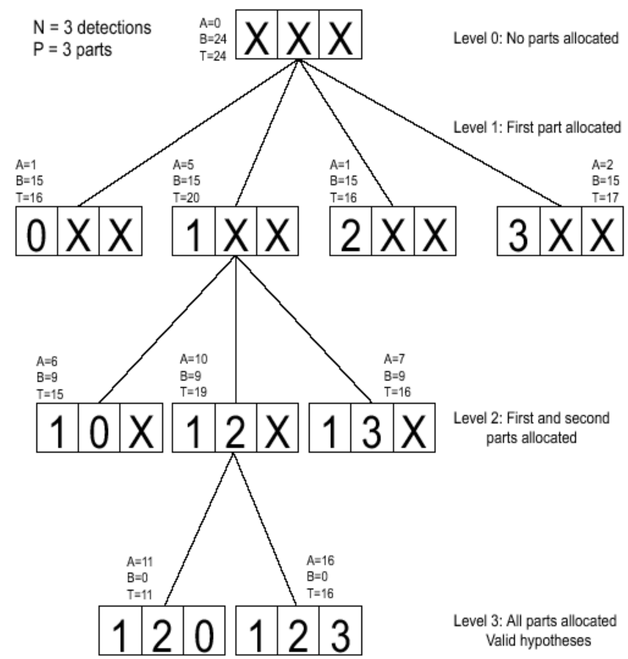




Efficient search methods

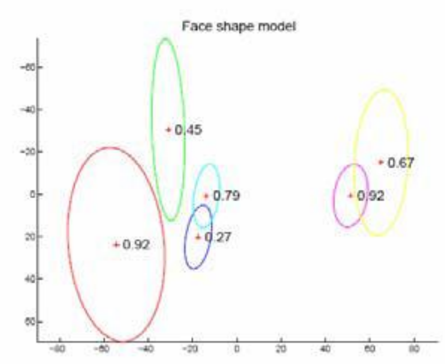
- Interpretation tree (Grimson ’87)
  – Condition on assigned parts to give search regions for remaining ones
  – Branch & bound, A*





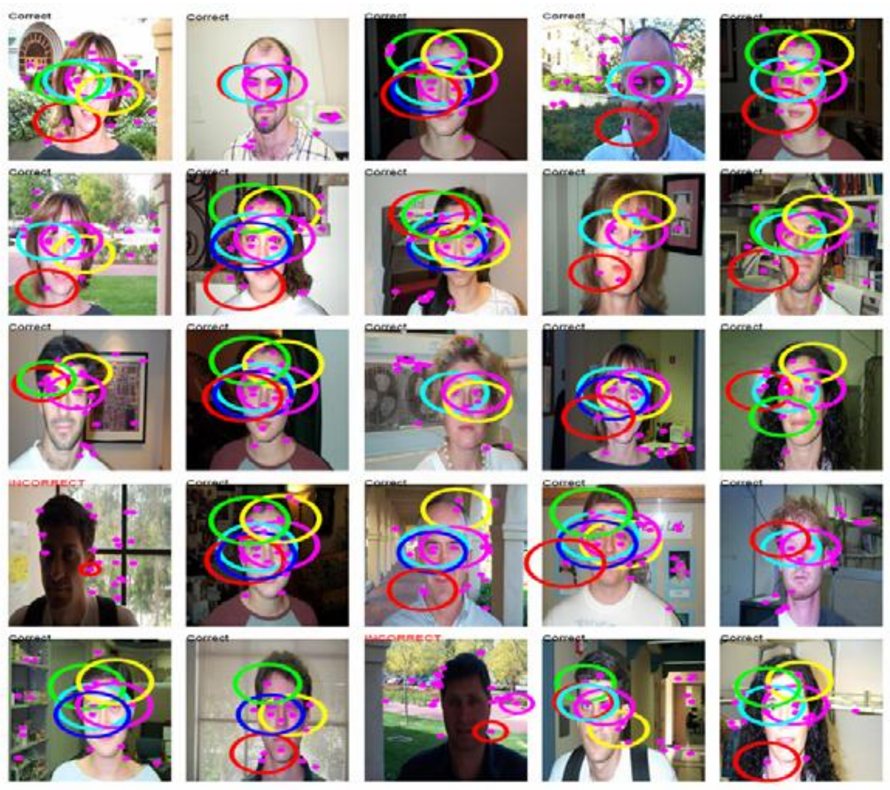
Results: Faces

Face shape model

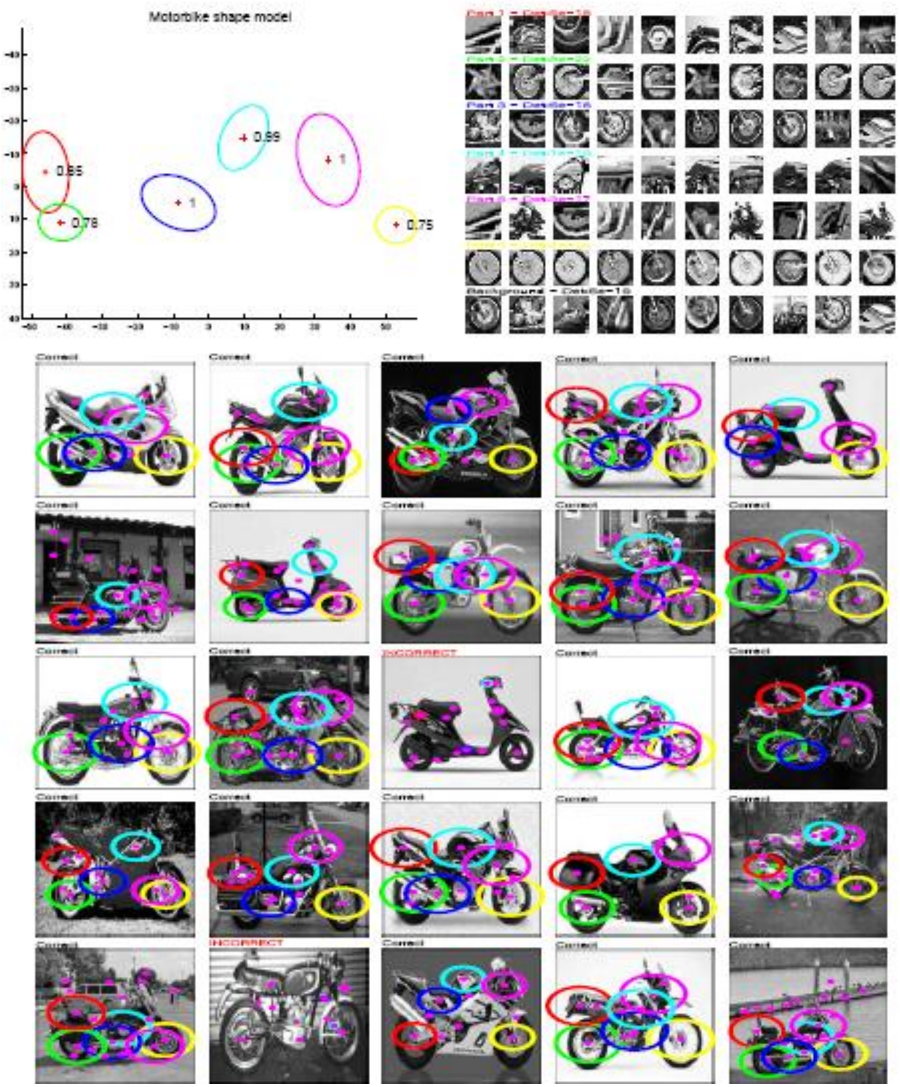


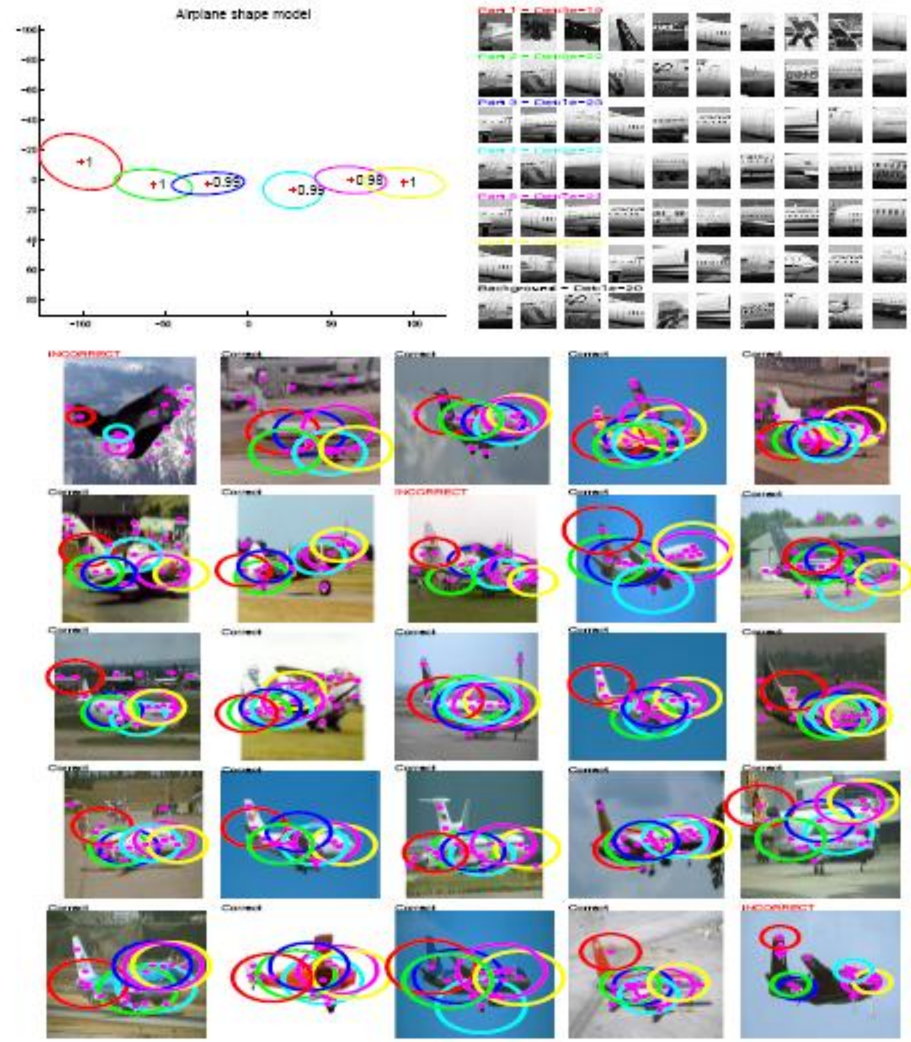
Patch appearance model

Recognition results



Results: Motorbikes and airplanes



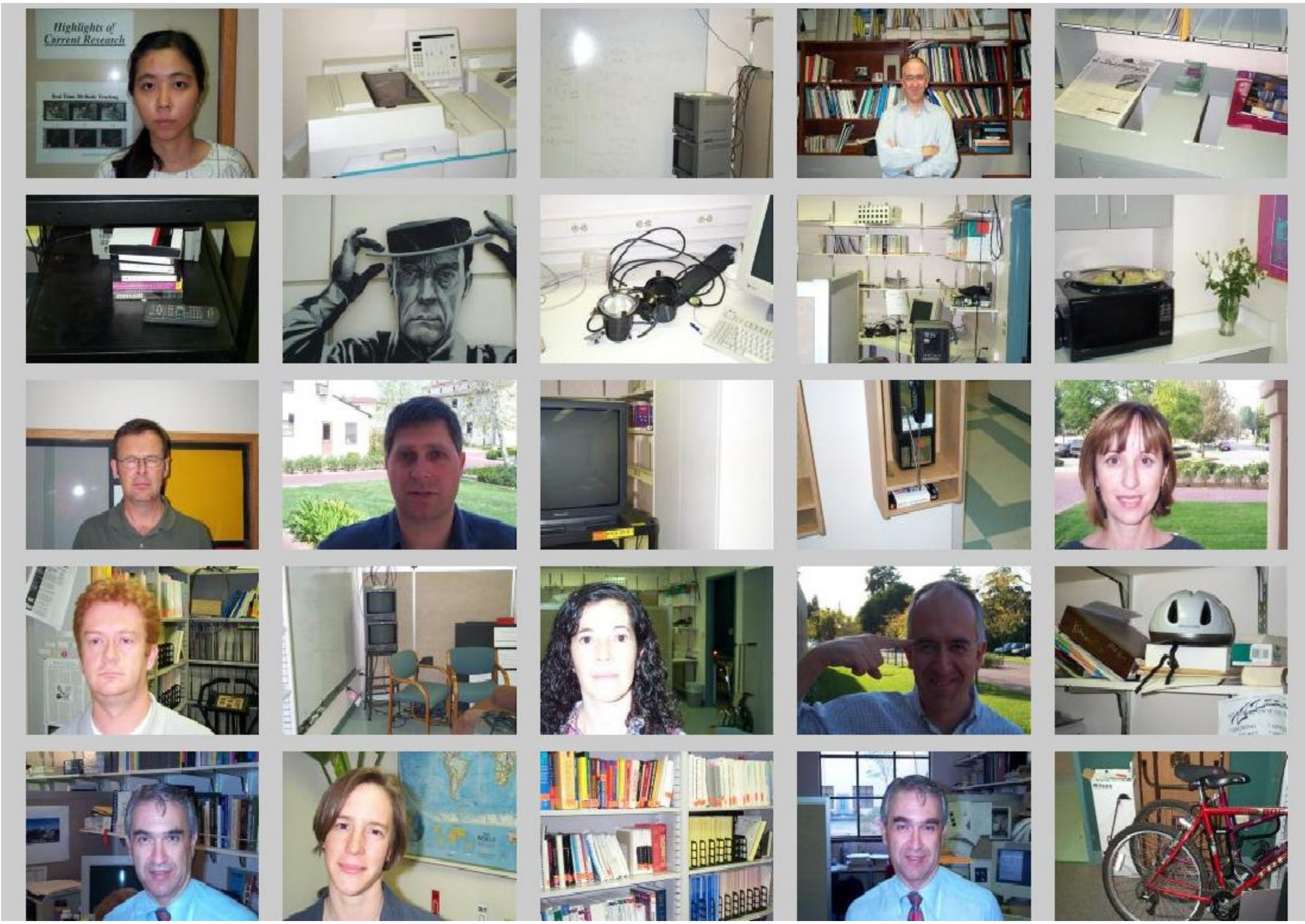


Parts and Structure demo

- Gaussian location model – star configuration
- Translation invariant only
  - Use 1st part as landmark
- Appearance model is template matching
- Manual training
  - User identifies correspondence on training images
- Recognition
  - Run template for each part over image
  - Get local maxima → set of possible locations for each part
  - Impose shape model - $O(N^2P)$ cost
  - Score of each match is combination of shape model and template responses.

Demo images

- Sub-set of Caltech face dataset
- Caltech background images



Demo Web Page

Demo (2)

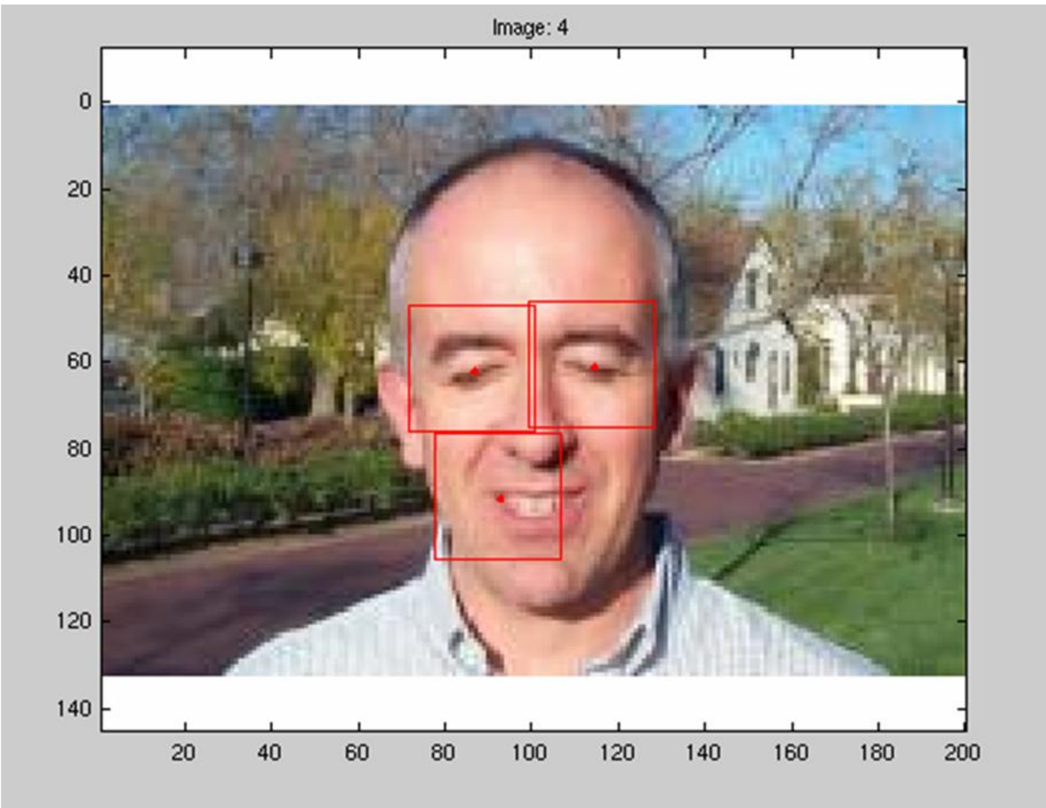
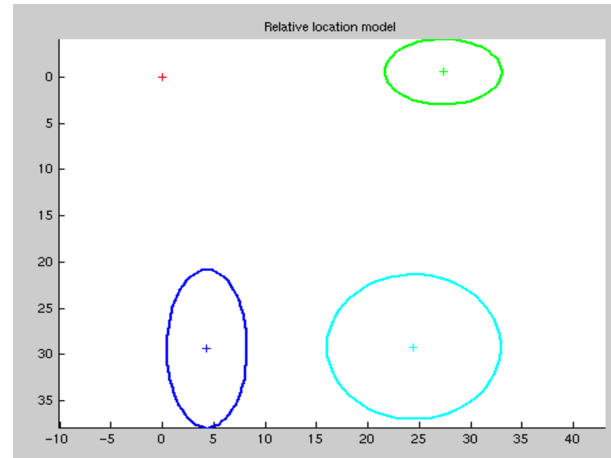
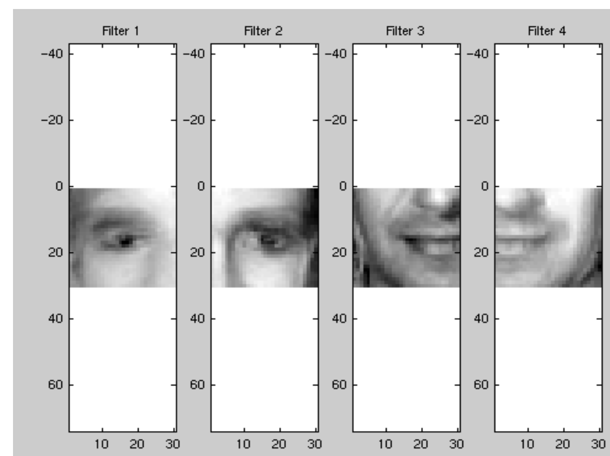


Image: 4

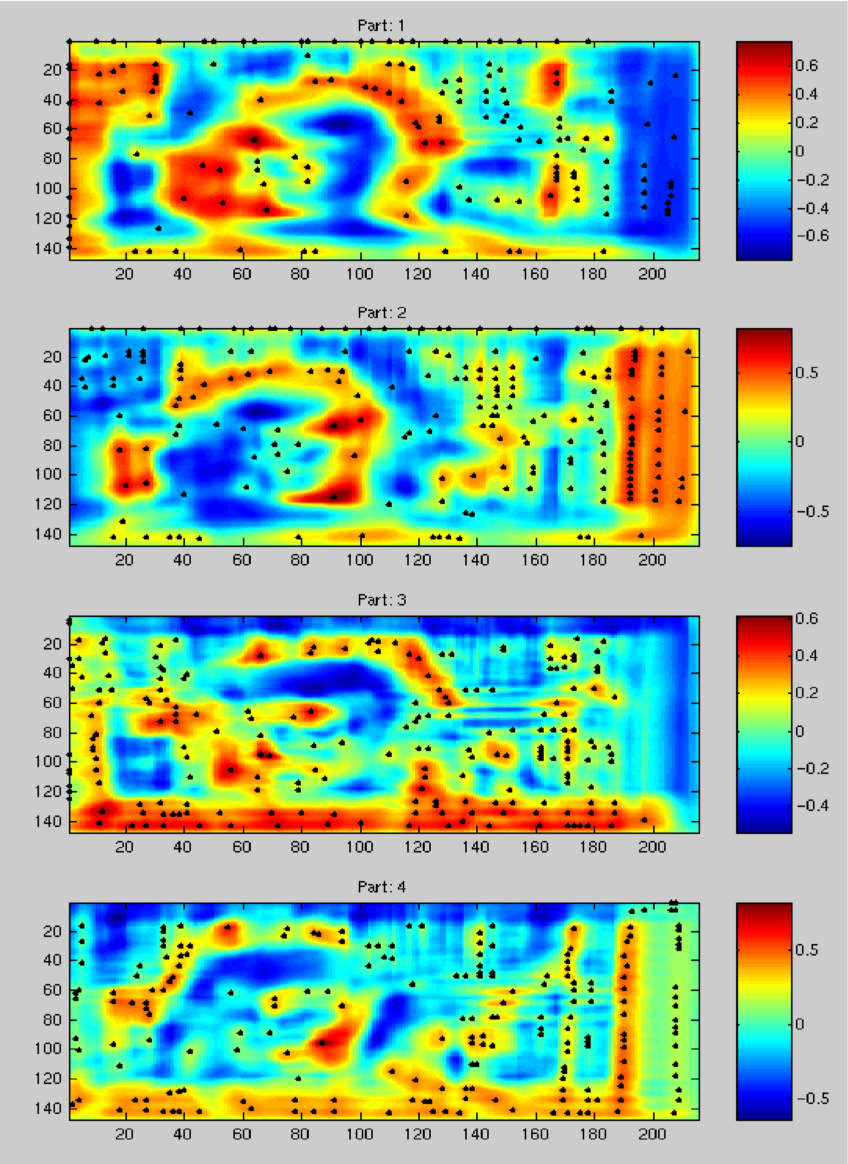


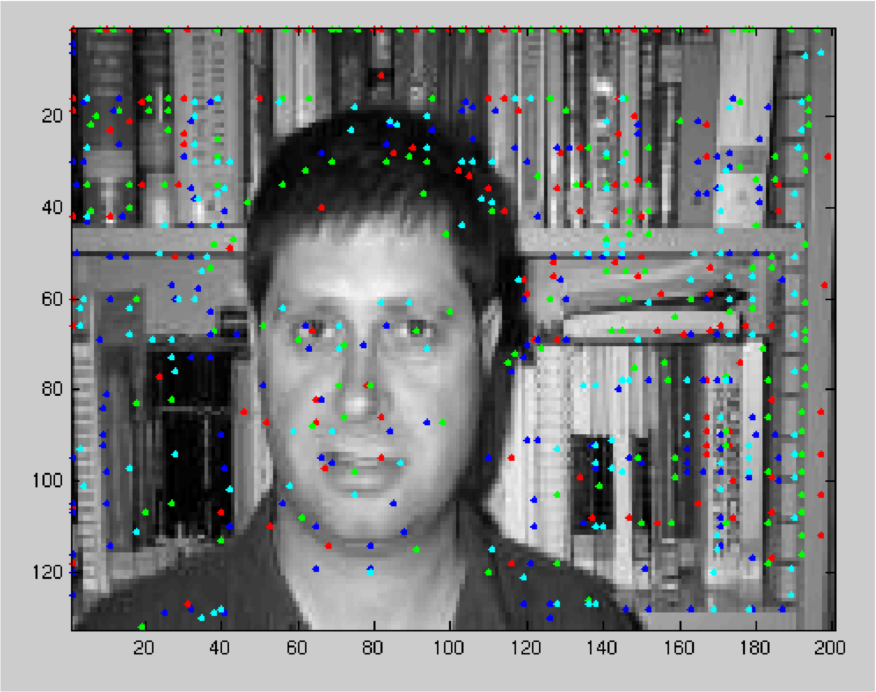
Relative location model



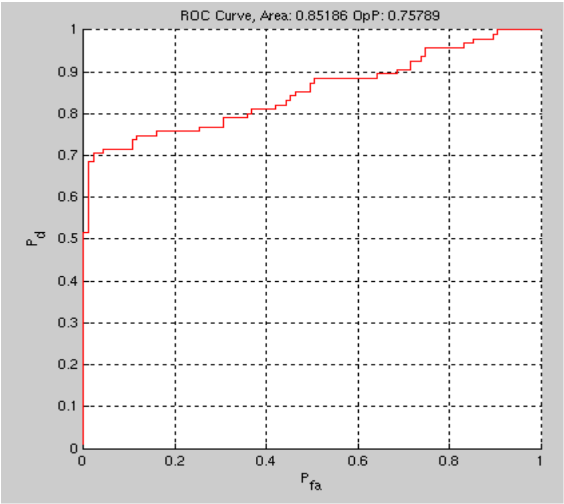
Filter 1
Filter 2
Filter 3
Filter 4

Demo (3)

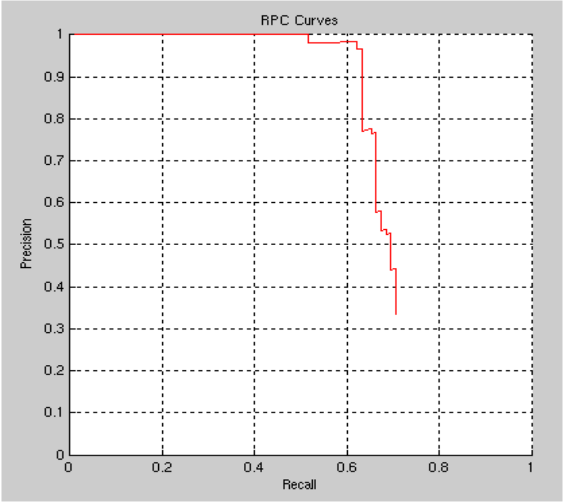




Demo (4)



ROC Curve, Area: 0.85186 OpP: 0.75789
P_d
P_fa



RPC Curves
Precision
Recall

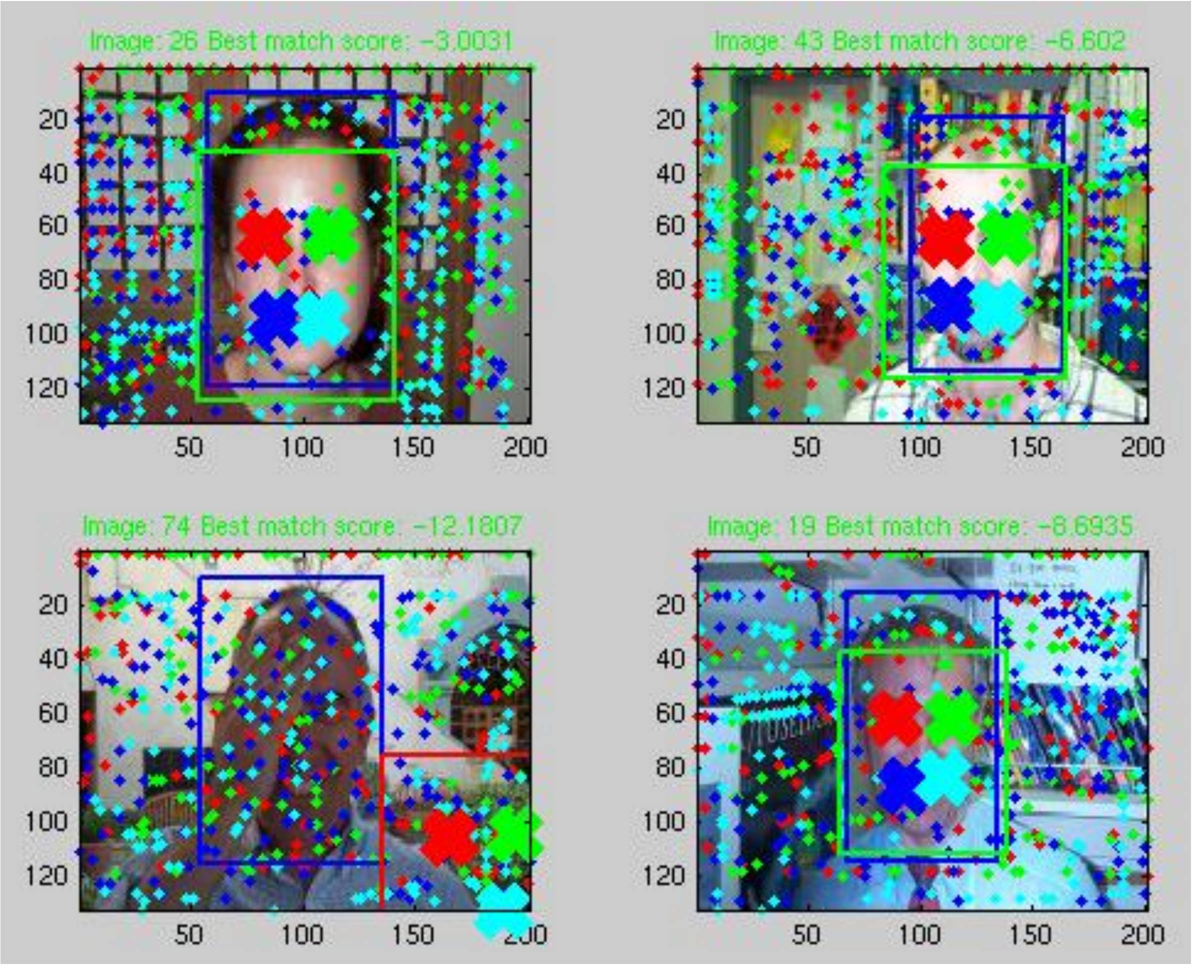


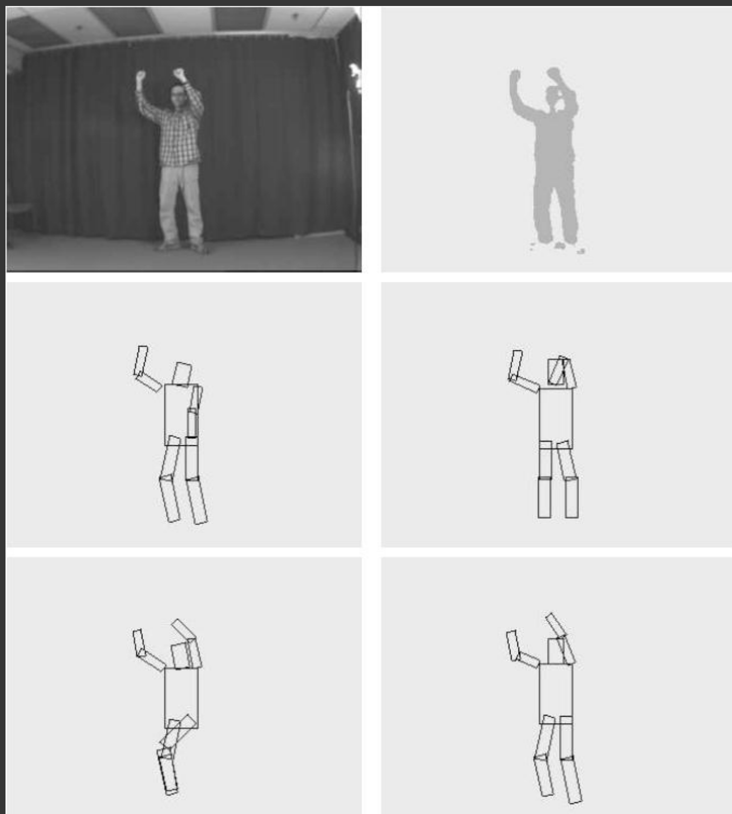
Image: 26 Best match score: -3.0031
Image: 43 Best match score: -6.602
Image: 74 Best match score: -12.1807
Image: 19 Best match score: -8.6935

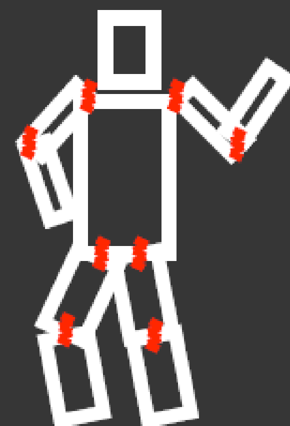
Overview

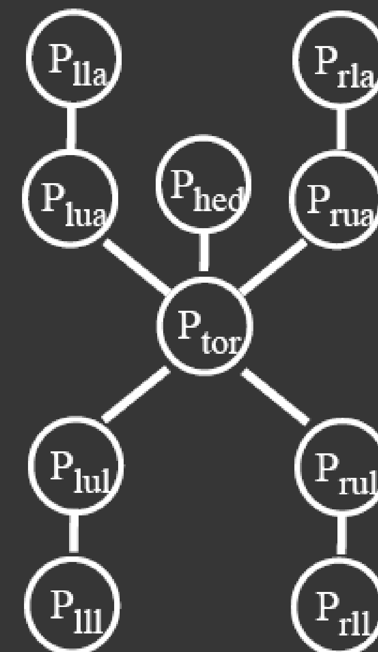
- Representation
  - Location
  - Appearance
  - Generative interpretation
- Learning
- Distance transforms
- Other approaches using parts
- Felzenszwalb, Girshick, McAllester, Ramanan CVPR 2008

Pictorial structure model

Fischler and Elschlager(73), Felzenszwalb and Huttenlocher(00)







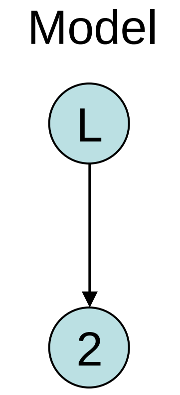
$$\Pr(P_{tor}, P_{arm}, \ldots | \text{Im}) \propto \prod_{i,j} \Pr(P_i \mid P_j) \prod_i \Pr(\text{Im}(P_i))$$

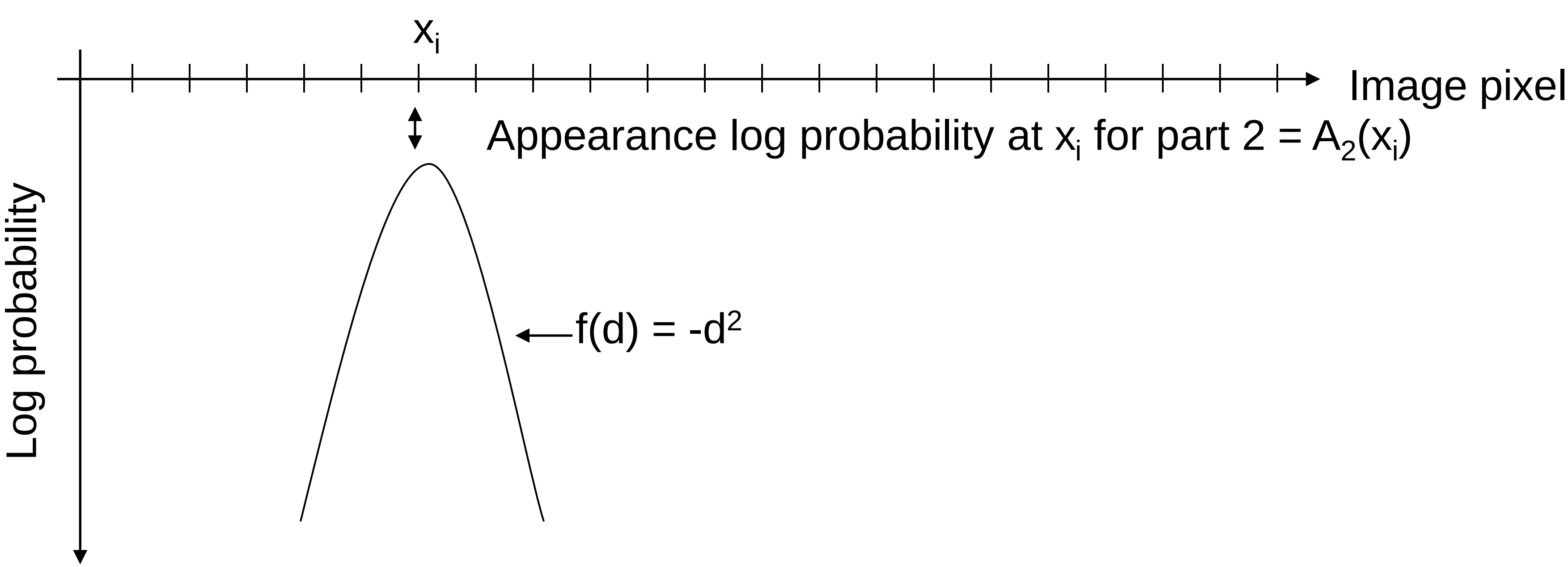
part geometry

part appearance

Distance transforms

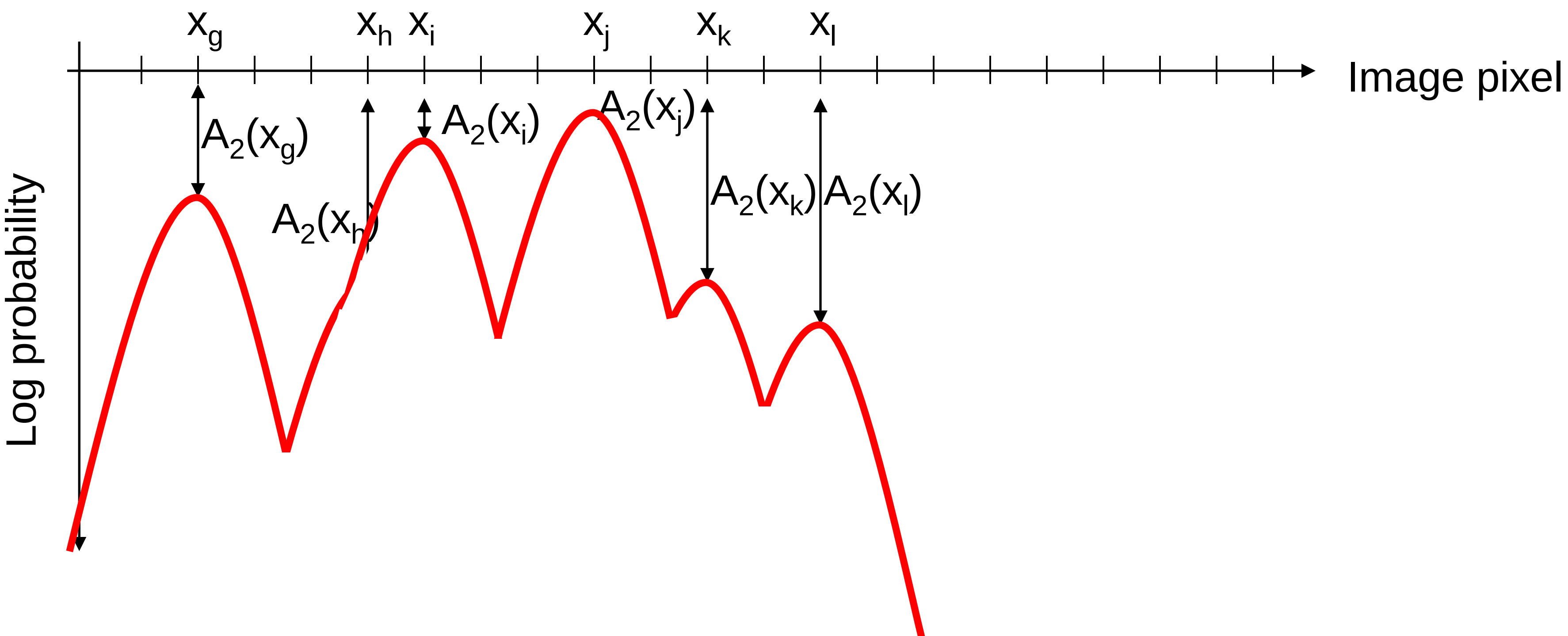
- Felzenszwalb and Huttenlocher '00 & '05
- Distance transforms
  - $O(N^2P) \rightarrow O(NP)$ for tree structured models
- How it works
  - Assume location model is Gaussian (i.e. e^{-d^2})
  - Consider a two part model with μ=0, σ=1 on a 1-D image





Distance transforms 2

- For each position of landmark part, find best position for part 2
  - Finding most probable x_i is equivalent finding maximum over set of offset parabolas
  - Upper envelope computed in O(N) rather than obvious O(N^2) via distance transform (see Felzenszwalb and Huttenlocher '05).
- Add $A_L(x)$ to upper envelope (offset by μ) to get overall probability map



Admin

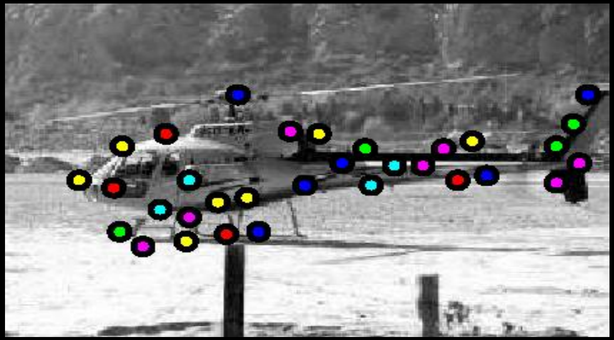
- Need to move next week’s class to Tuesday 7pm.

Overview

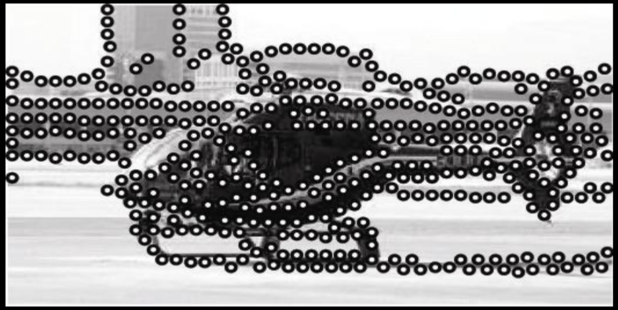
- Representation
  - Location
  - Appearance
  - Generative interpretation
- Learning
- Distance transforms
- Other approaches using parts
- Felzenszwalb, Girshick, McAllester, Ramanan CVPR 2008

Deformable Template Matching

Berg et al. CVPR 2005



Template



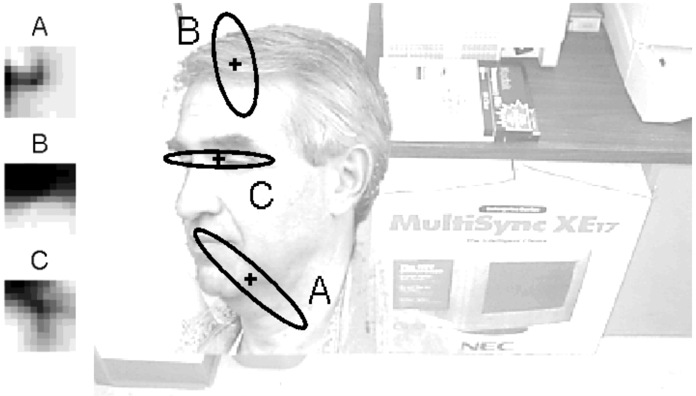
Query

- Formulate problem as Integer Quadratic Programming
- $O(N^P)$ in general
- Use approximations that allow P=50 and N=2550 in <2 secs

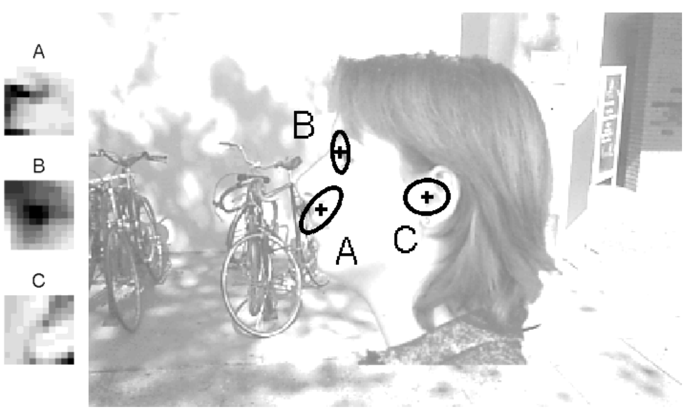
Multiple views

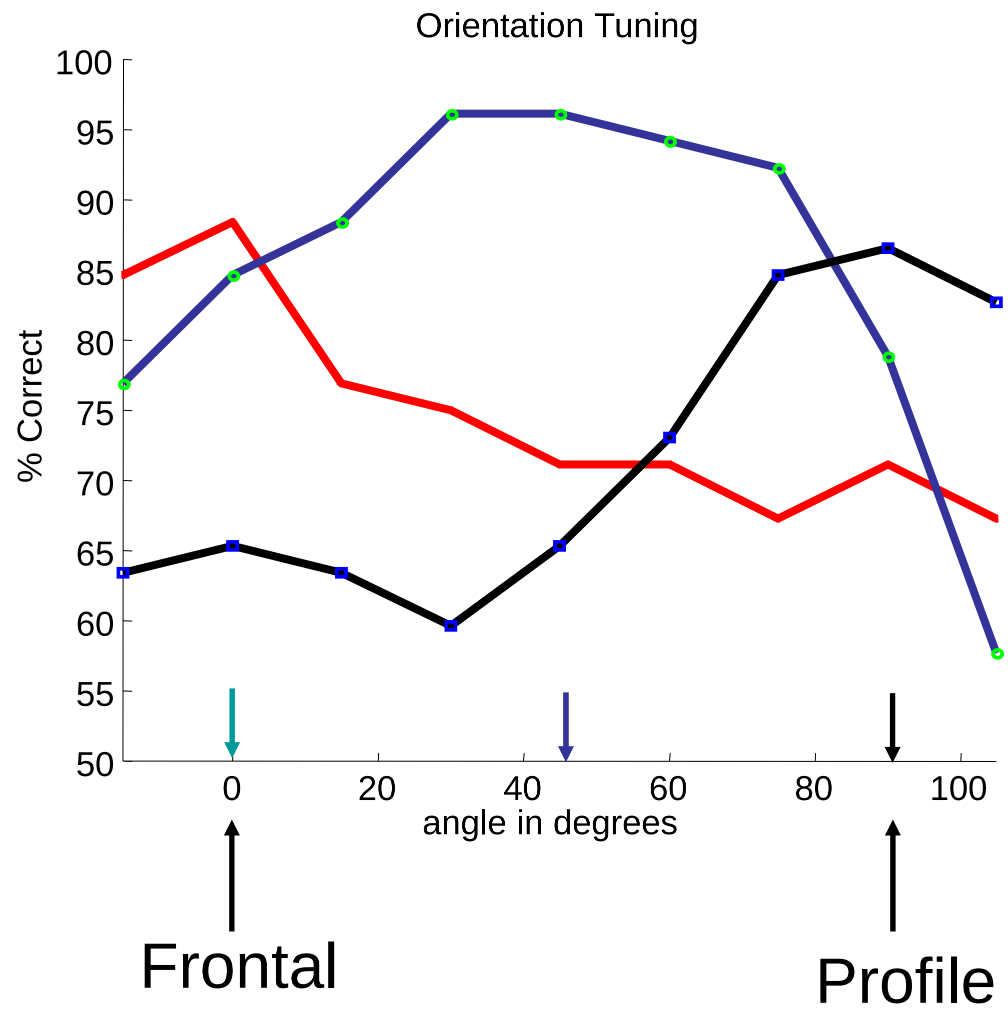
- Full 3-D location model
- Mixture of 2-D models
  - Weber CVPR ‘00

Component 1

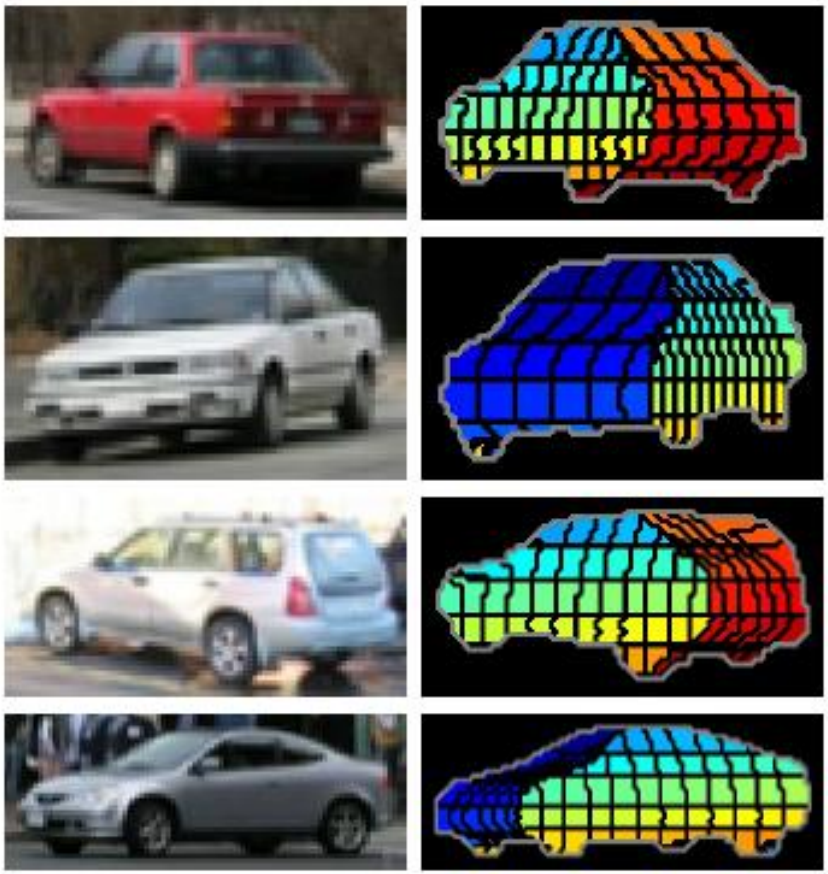


Component 2

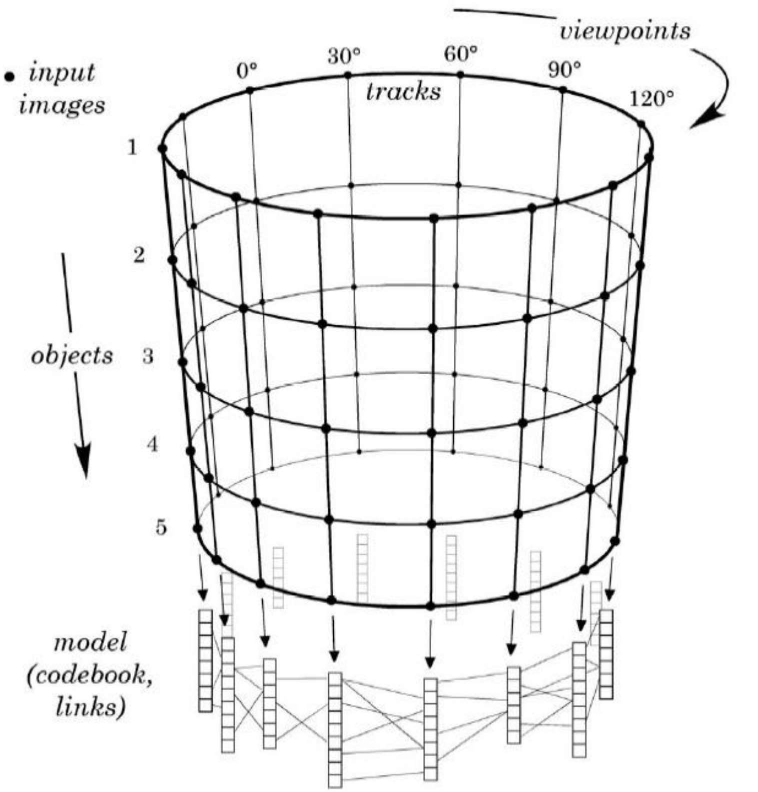




Multiple view points



Hoiem, Rother, Winn, 3D LayoutCRF for Multi-View Object Class Recognition and Segmentation, CVPR ‘07

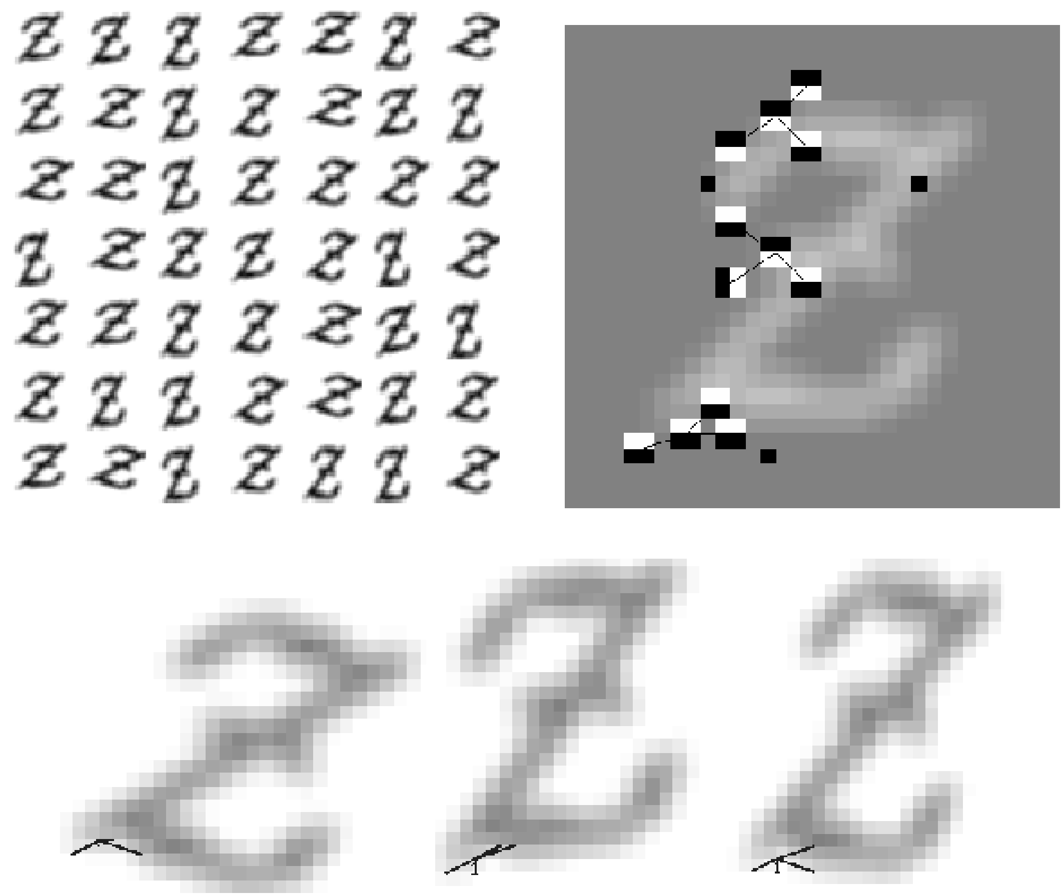


Thomas, Ferrari, Leibe, Tuytelaars, Schiele, and L. Van Gool. Towards Multi-View Object Class Detection, CVPR 06

Hierarchical Representations

- Pixels → Pixel groupings → Parts → Object
- Multi-scale approach increases number of low-level features

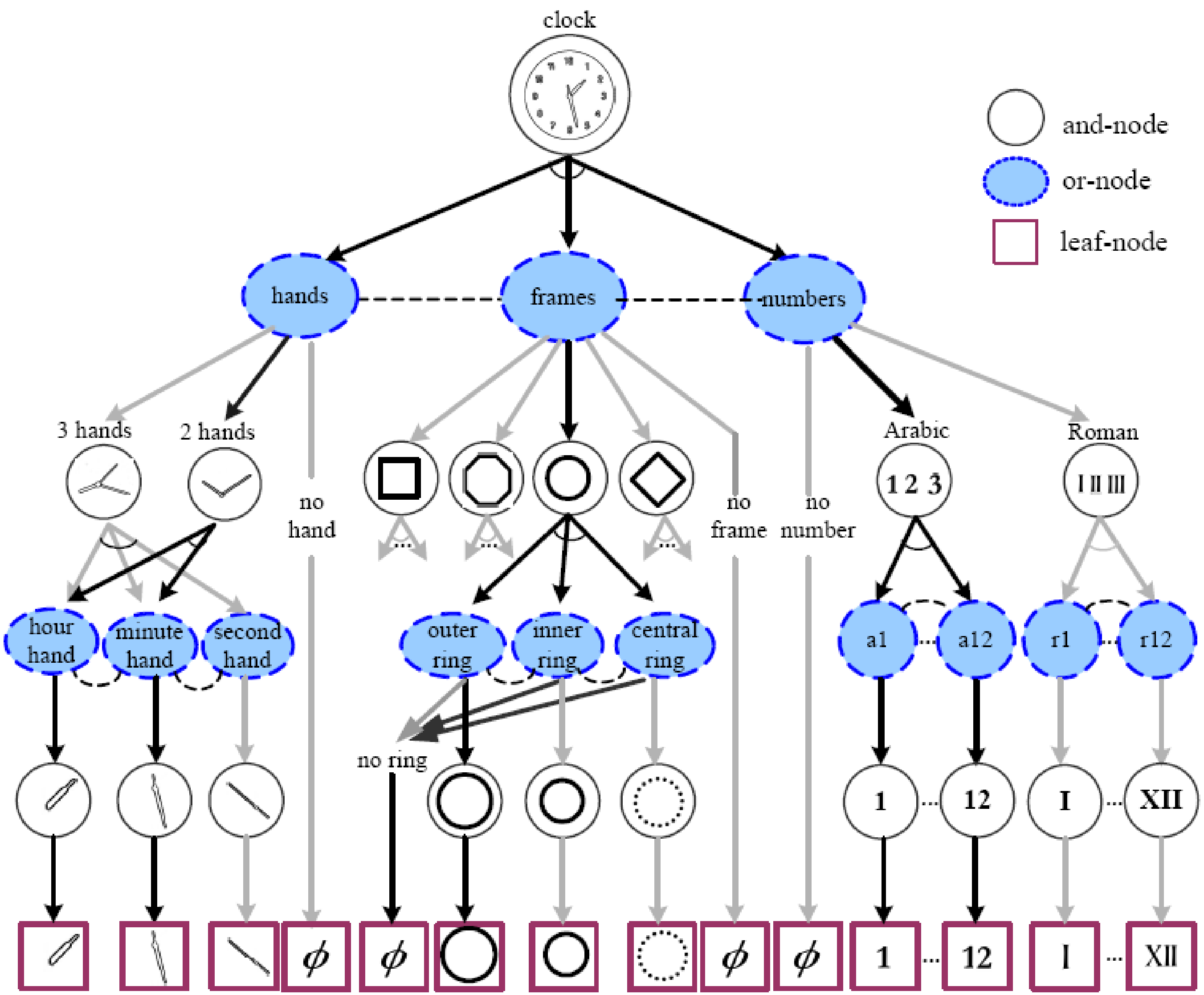
- Amit and Geman ’98
- Ullman et al.
- Bouchard & Triggs ’05
- Zhu and Mumford
- Jin & Geman ‘06
- Zhu & Yuille ’07
- Fidler & Leonardis ‘07



Images from [Amit98]

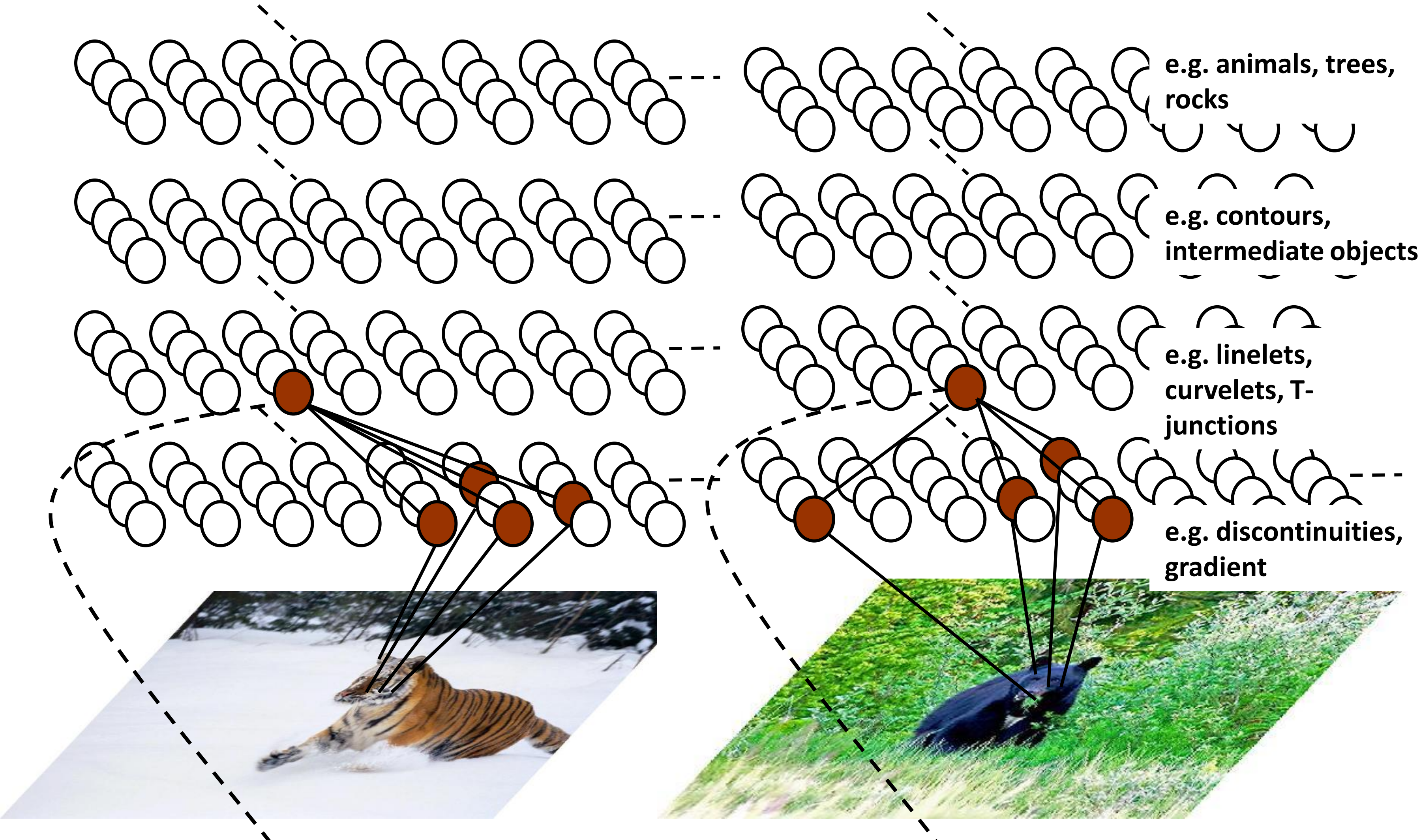
Stochastic Grammar of Images

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



Context and Hierarchy in a Probabilistic Image Model

Jin & Geman (2006)

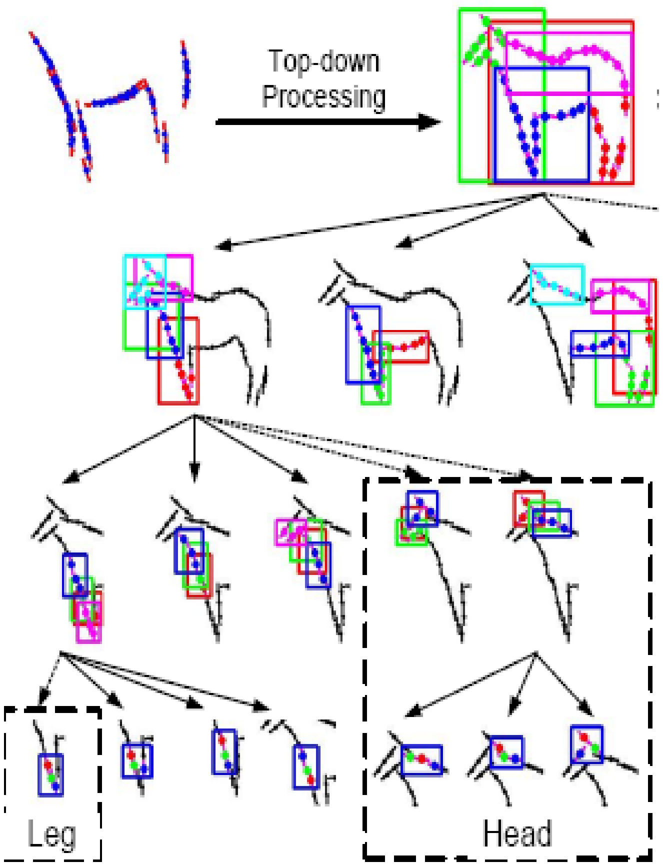


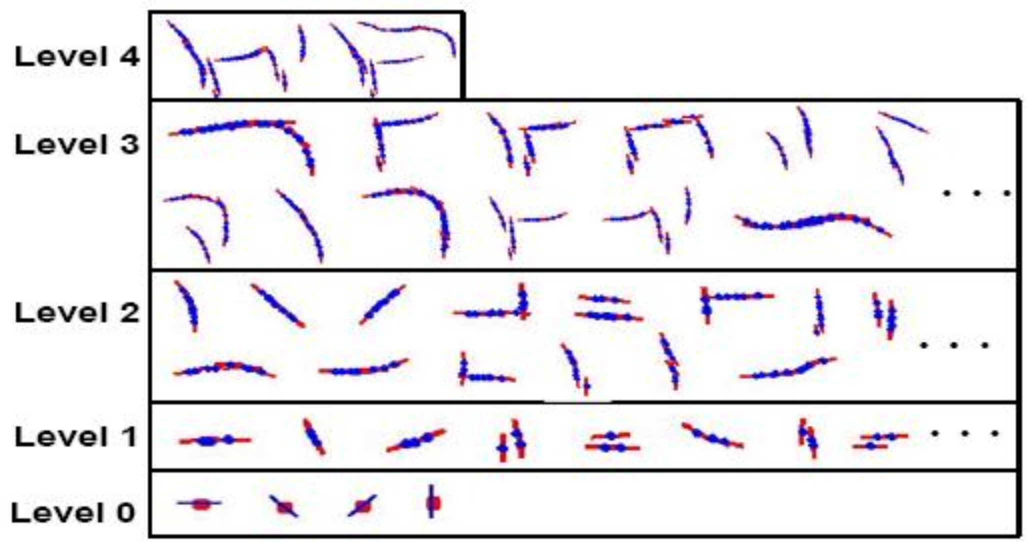
animal head instantiated by tiger head

animal head instantiated by bear head

A Hierarchical Compositional System for Rapid Object Detection

Long Zhu, Alan L. Yuille, 2007.

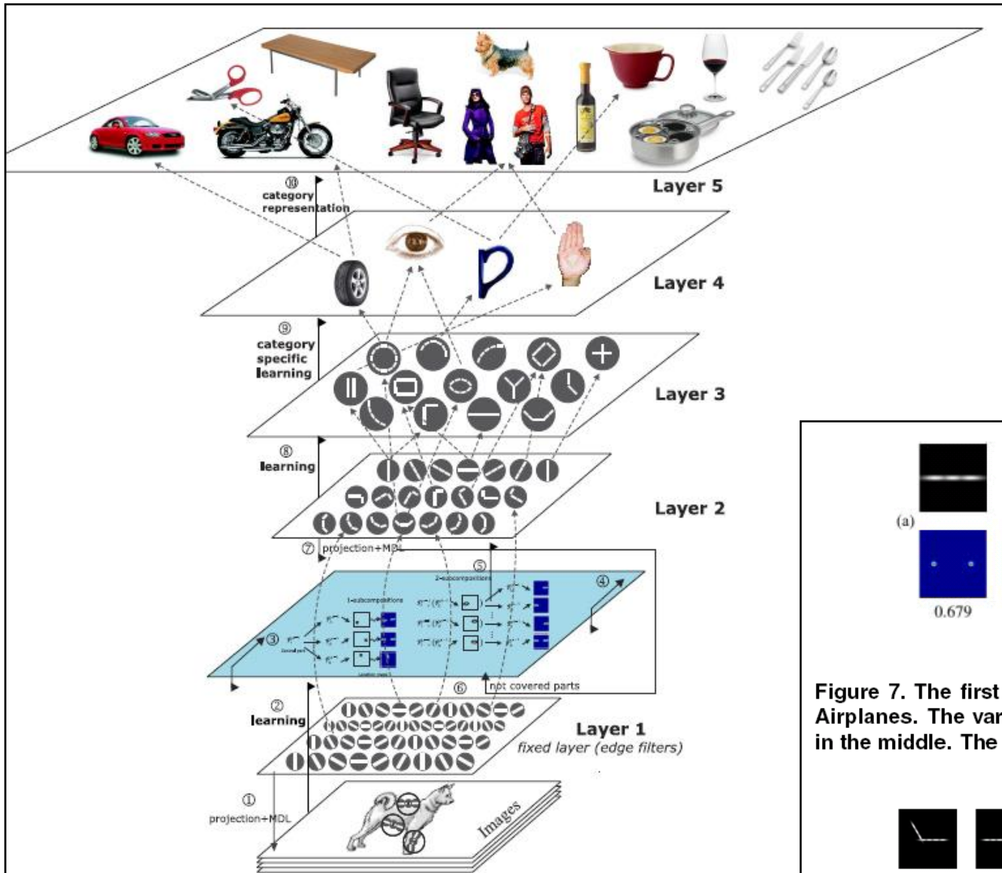




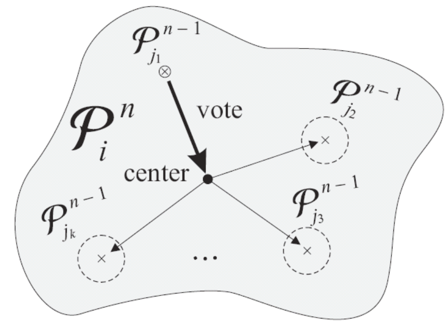
Able to learn #parts at each level

Learning a Compositional Hierarchy of Object Structure

Fidler & Leonardis, CVPR'07; Fidler, Boben & Leonardis, CVPR 2008



The architecture



Parts model

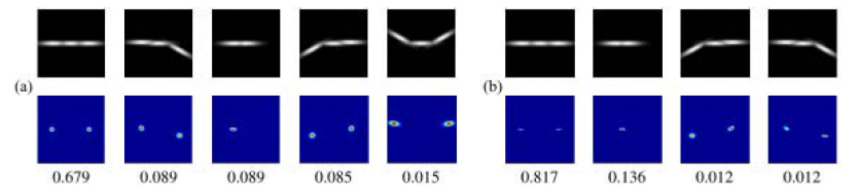


Figure 7. The first row depicts the final parts comprising Layer II obtained for (a) Cliparts and (b) Airplanes. The variances of position distributions of parts, relative to the central part, are depicted in the middle. The feature probabilities are listed in the last row.

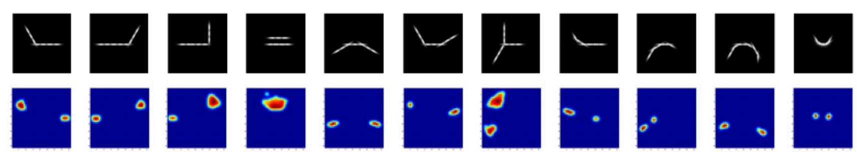
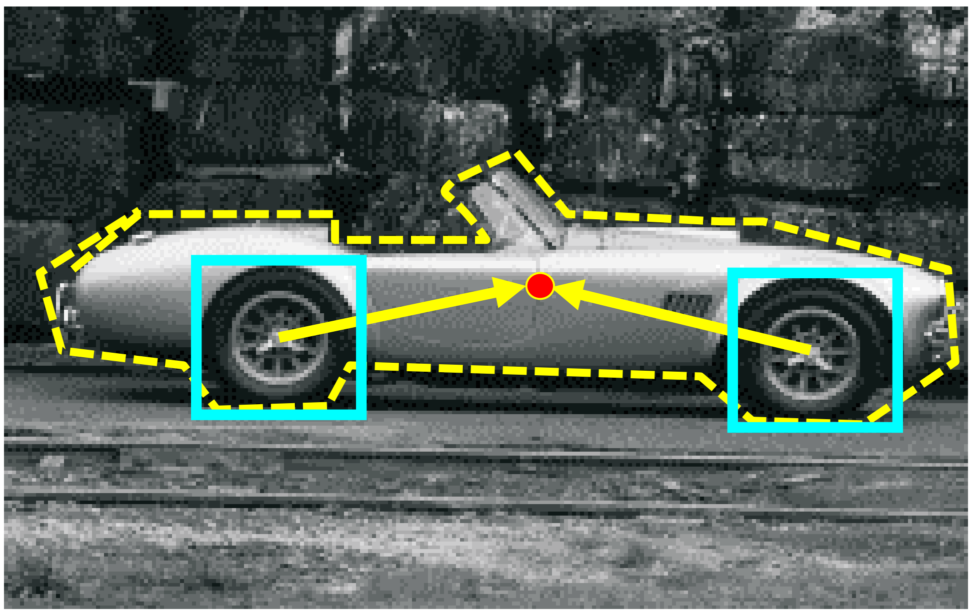


Figure 8. (a) Examples of Layer 3 parts, (b) variances of positions of the surrounding subparts

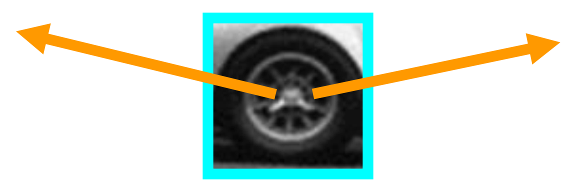
Learned parts

Implicit shape models

- Visual codebook is used to index votes for object position



training image annotated with object localization info



visual codeword with displacement vectors

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

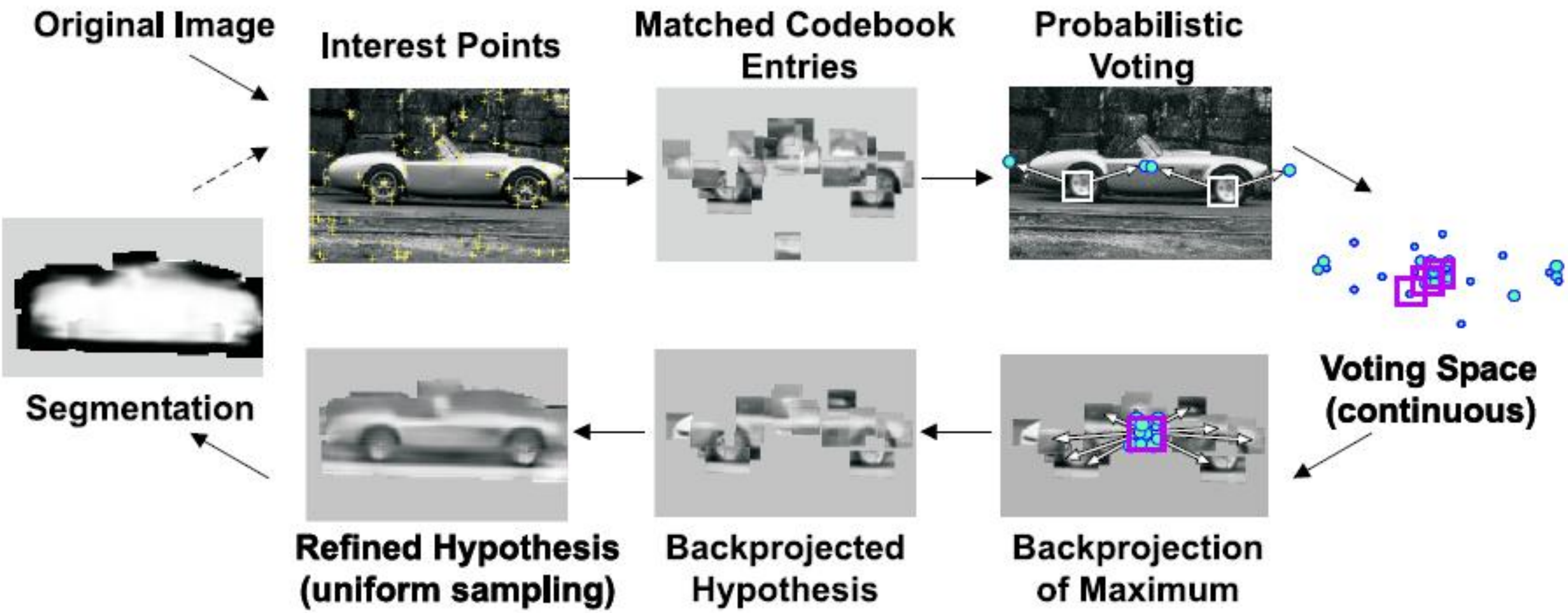
Implicit shape models

- Visual codebook is used to index votes for object position

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

Implicit shape models: Details



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

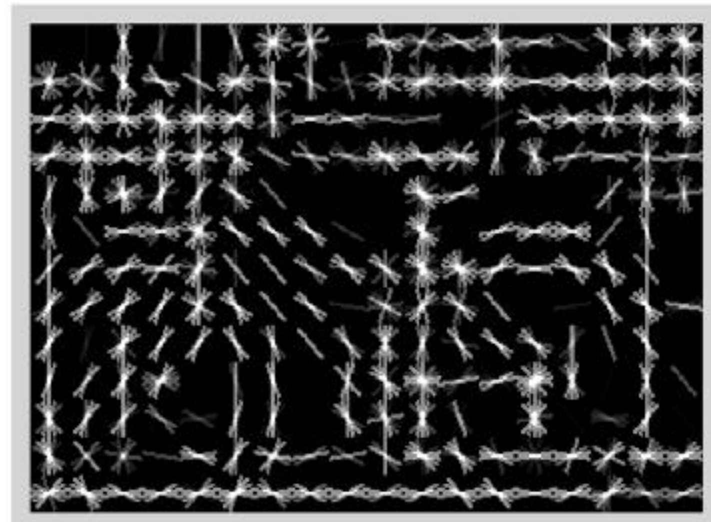
Overview

- Representation
  - Location
  - Appearance
  - Generative interpretation
- Learning
- Distance transforms
- Other approaches using parts
- Felzenszwalb, Girshick, McAllester, Ramanan CVPR 2008

Object Detection with Discriminatively Trained Part Based Models

Pedro F. Felzenszwalb, Ross B. Girshick, David McAllester and Deva Ramanan

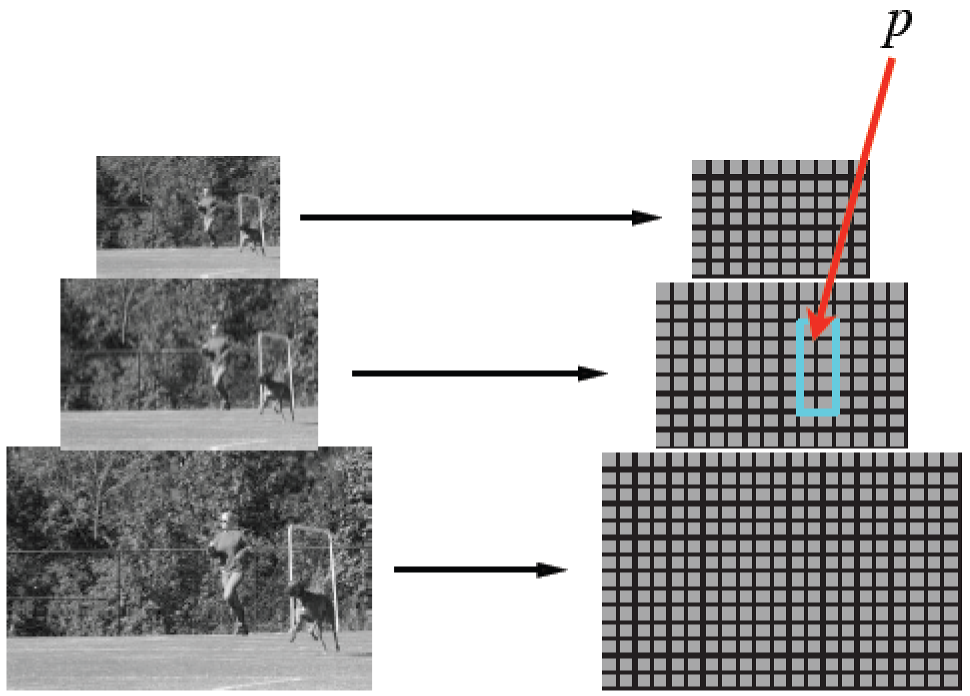
Histogram of Gradient (HOG) features



- Image is partitioned into 8x8 pixel blocks
- In each block we compute a histogram of gradient orientations
  - Invariant to changes in lighting, small deformations, etc.
- Compute features at different resolutions (pyramid)

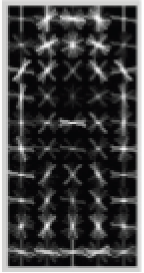
HOG Filters

- Array of weights for features in subwindow of HOG pyramid
- Score is dot product of filter and feature vector



HOG pyramid H

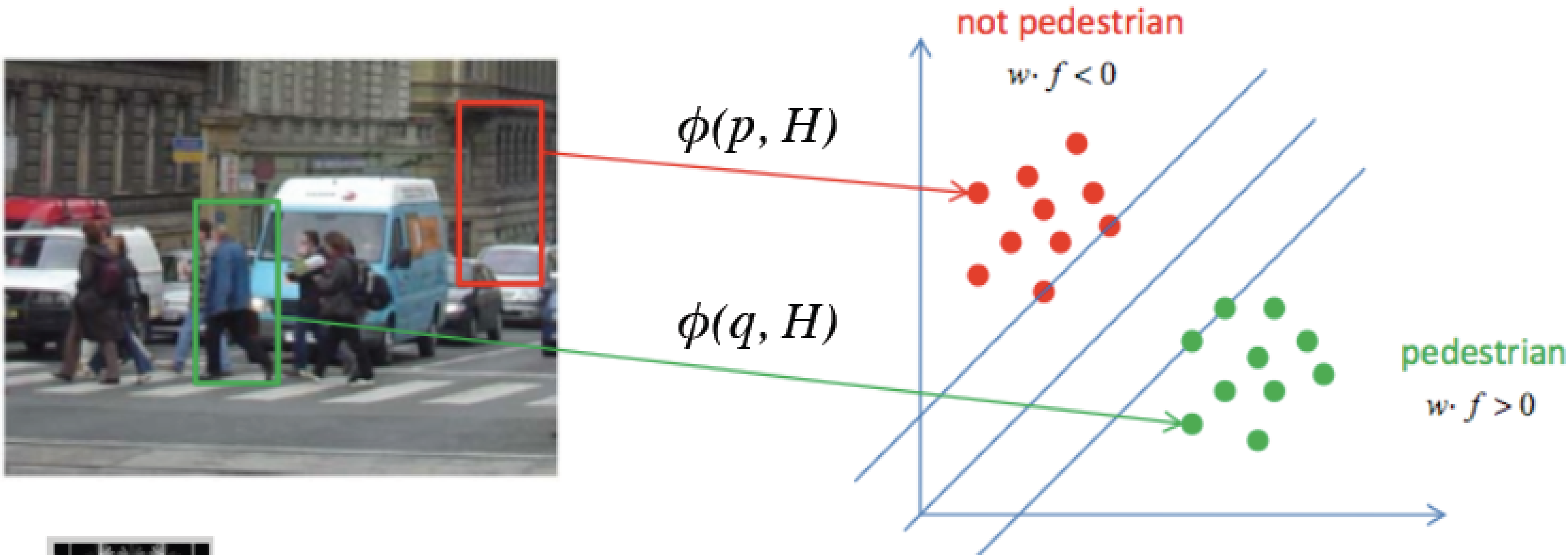
Filter F

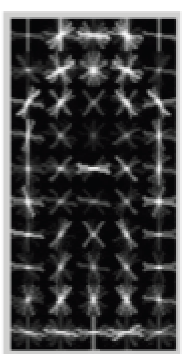


Score of F at position p is
$F \cdot \phi(p, H)$

$\phi(p, H)$ = concatenation of HOG features from subwindow specified by p

Dalal & Triggs: HOG + linear SVMs





Typical form of a model

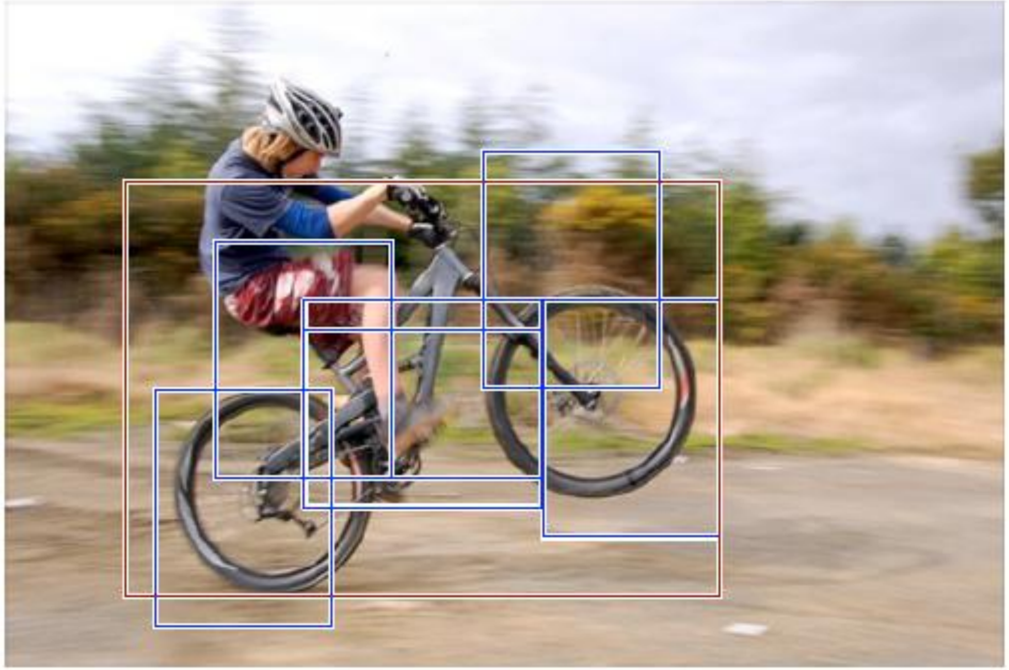
There is much more background than objects

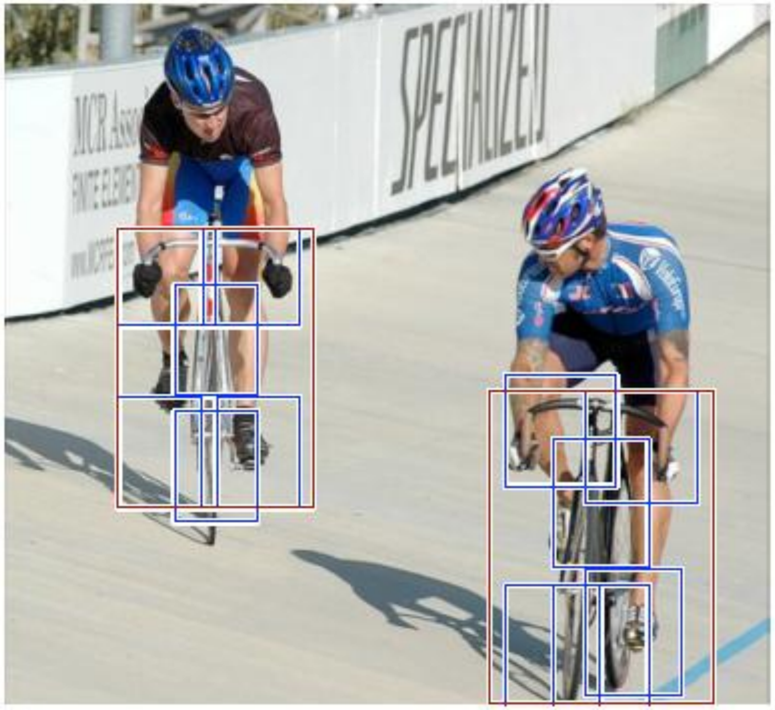
Start with random negatives and repeat:

1) Train a model

2) Harvest false positives to define "hard negatives"

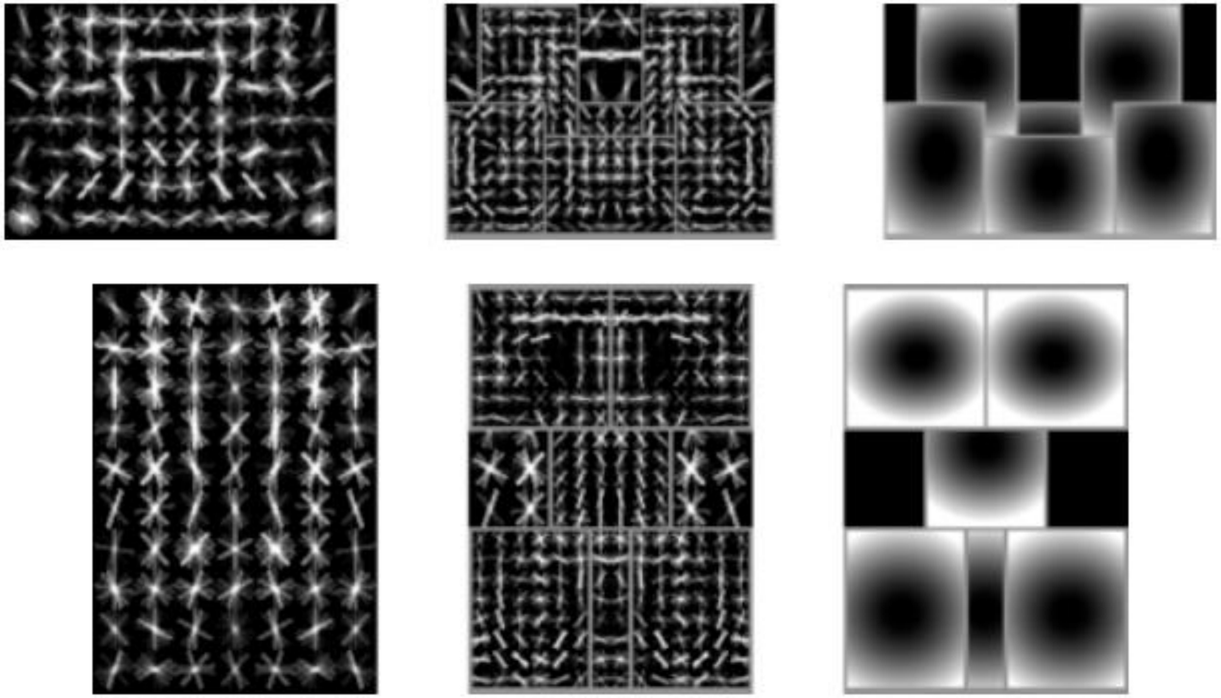
Overview of our models





- Mixture of deformable part models
- Each component has global template + deformable parts
- Fully trained from bounding boxes alone

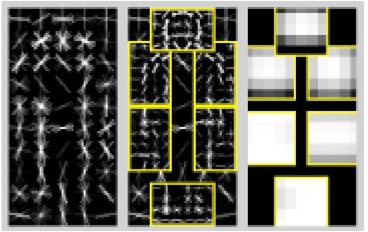
2 component bicycle model



root filters coarse resolution | part filters finer resolution | deformation models

Each component has a root filter F_0
and n part models (F_i, v_i, d_i)

Object hypothesis



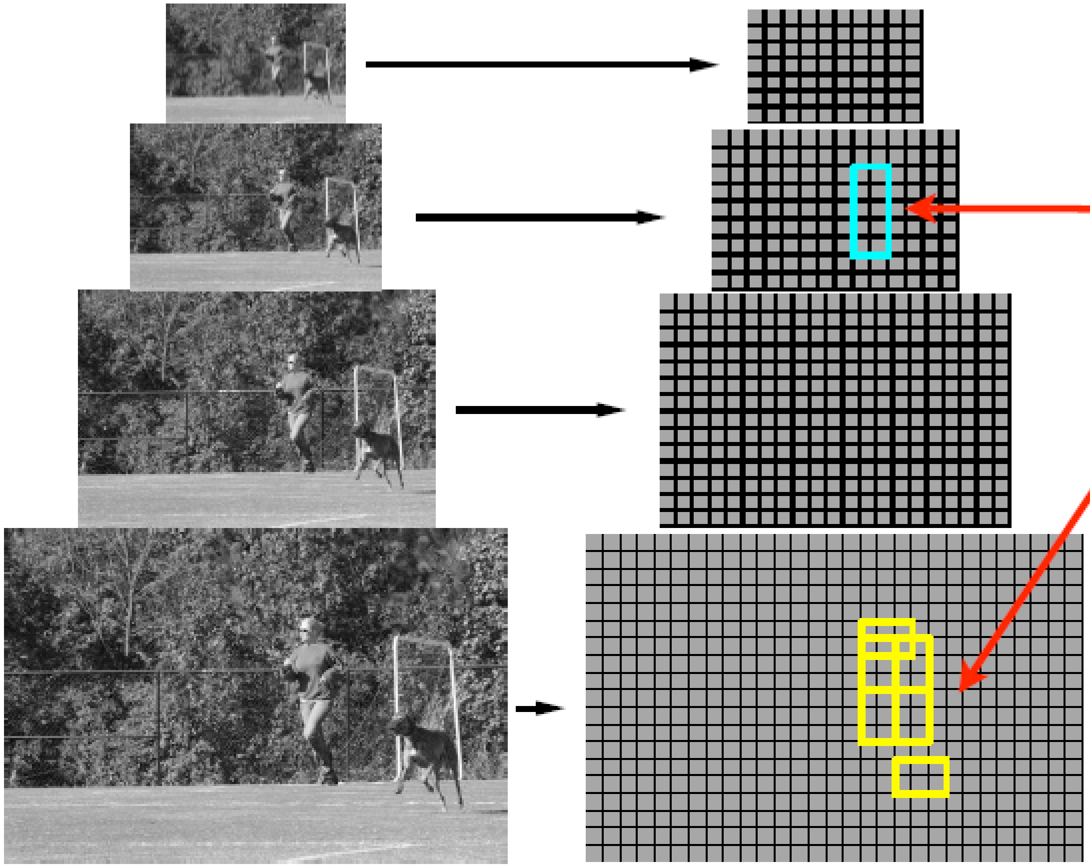


Image pyramid HOG feature pyramid

$z = (p_0, \ldots, p_n)$

p_0 : location of root

$p_1, \ldots, p_n$: location of parts

Score is sum of filter scores minus deformation costs

Multiscale model captures features at two-resolutions

Score of a hypothesis

"data term" "spatial prior"

$$\text{score}(p_0, \ldots, p_n) = \sum_{i=0}^{n} F_i \cdot \phi(H, p_i) - \sum_{i=1}^{n} d_i \cdot (dx_i^2, dy_i^2)$$

filters

deformation parameters

displacements

$$\text{score}(z) = \beta \cdot \Psi(H, z)$$

concatenation filters and deformation parameters

concatenation of HOG features and part displacement features

Matching

- Define an overall score for each root location
  - Based on best placement of parts

$$\text{score}(p_0) = \max_{p_1,\ldots,p_n} \text{score}(p_0,\ldots,p_n).$$

- High scoring root locations define detections
  - "sliding window approach"
- Efficient computation: dynamic programming + generalized distance transforms (max-convolution)

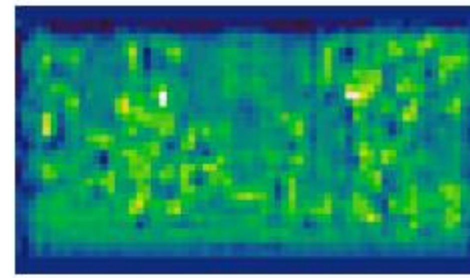
head filter

input image

Response of filter in l-th pyramid level

$$R_l(x, y) = F \cdot \phi(H, (x, y, l))$$

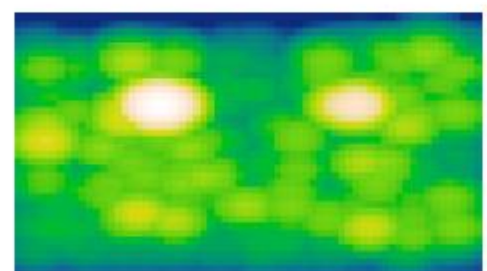
cross-correlation

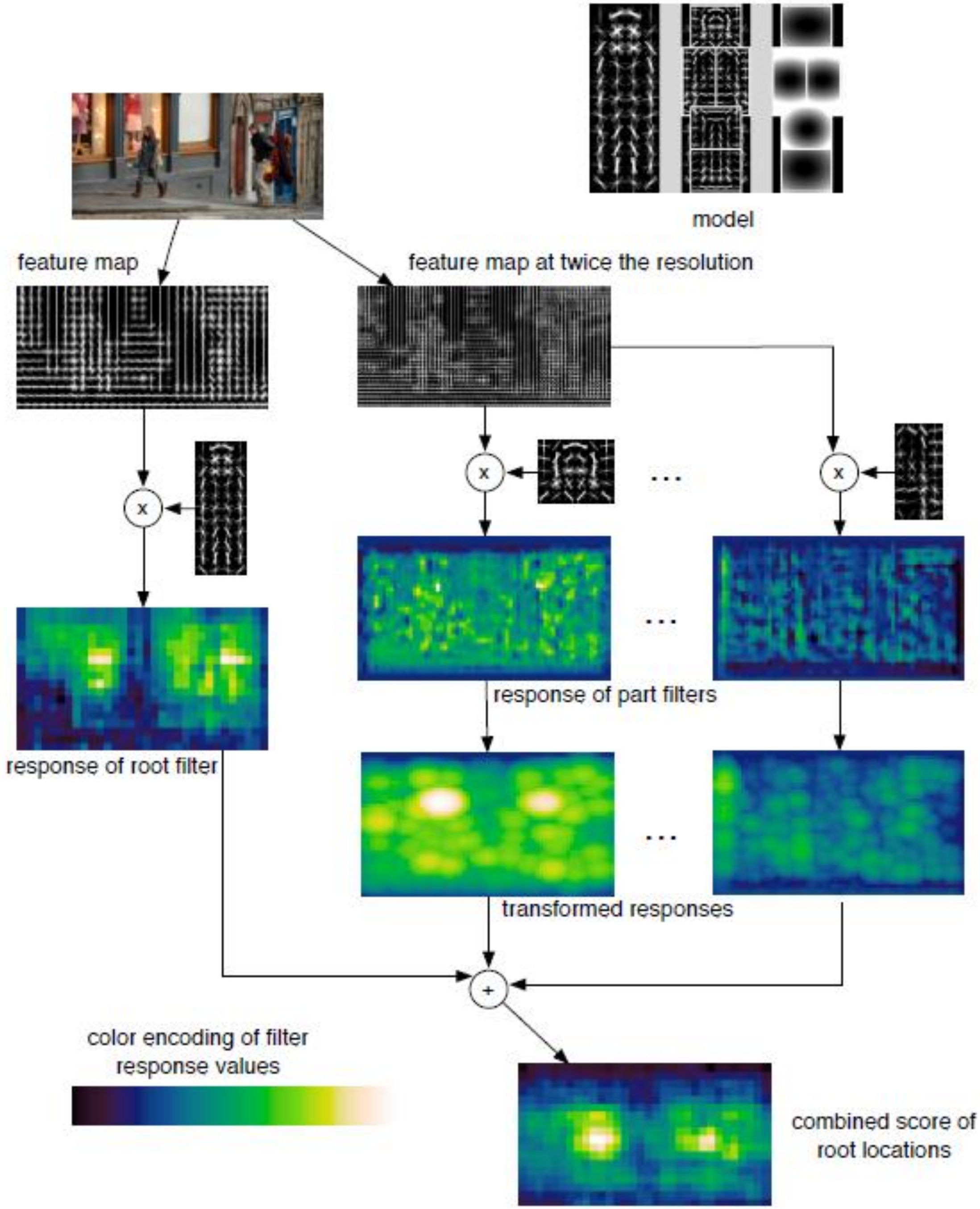


Transformed response

$$D_l(x, y) = \max_{dx, dy} \left(R_l(x + dx, y + dy) - d_i \cdot (dx^2, dy^2) \right)$$

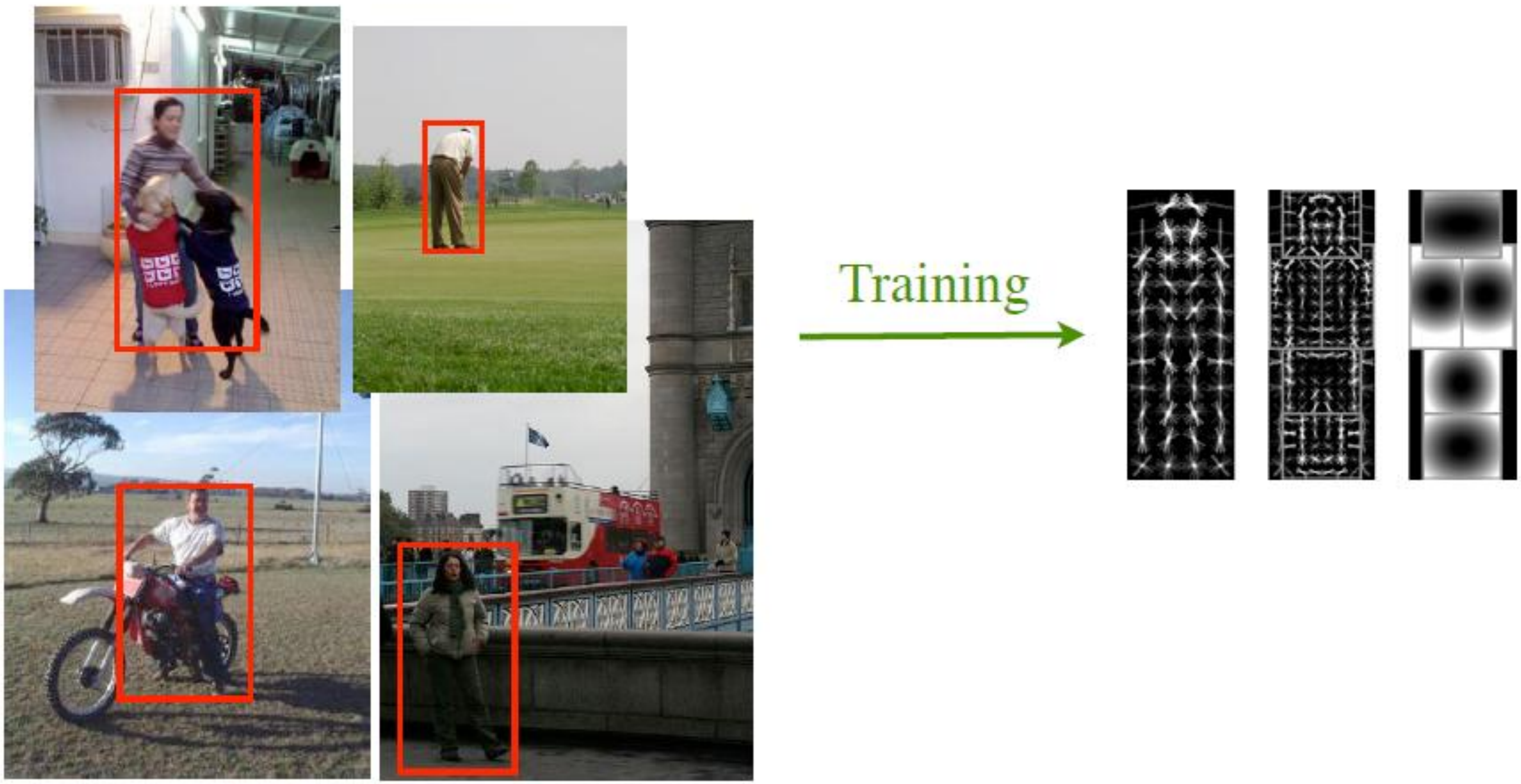
max-convolution, computed in linear time
(spreading, local max, etc)




model
feature map
feature map at twice the resolution
x
x
x
...
...
...
response of part filters
response of root filter
transformed responses
+
color encoding of filter
response values
combined score of
root locations

Training

- Training data consists of images with labeled bounding boxes.
- Need to learn the model structure, filters and deformation costs.



Latent SVM (MI-SVM)

Classifiers that score an example x using

$$f_\beta(x) = \max_{z \in Z(x)} \beta \cdot \Phi(x, z)$$

β are model parameters
z are latent values

Training data $D = (\langle x_1, y_1 \rangle, \ldots, \langle x_n, y_n \rangle)$ $\quad y_i \in \{-1, 1\}$

We would like to find β such that: $y_i f_\beta(x_i) > 0$

Minimize

$$L_D(\beta) = \frac{1}{2}||\beta||^2 + C \sum_{i=1}^{n} \max(0, 1 - y_i f_\beta(x_i))$$

Latent SVM training

$$L_D(\beta) = \frac{1}{2}||\beta||^2 + C\sum_{i=1}^{n} \max(0, 1 - y_i f_\beta(x_i))$$

- Convex if we fix z for positive examples
- Optimization:
  - Initialize β and iterate:
    - Pick best z for each positive example
    - Optimize β via gradient descent with data-mining

Training algorithm, nested iterations

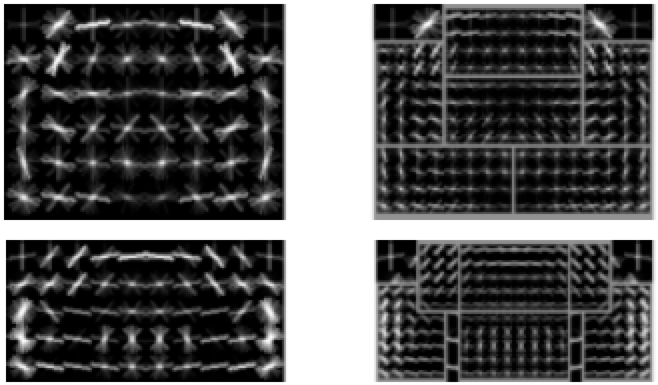
Fix "best" positive latent values for positives

Harvest high scoring (x,z) pairs from background images

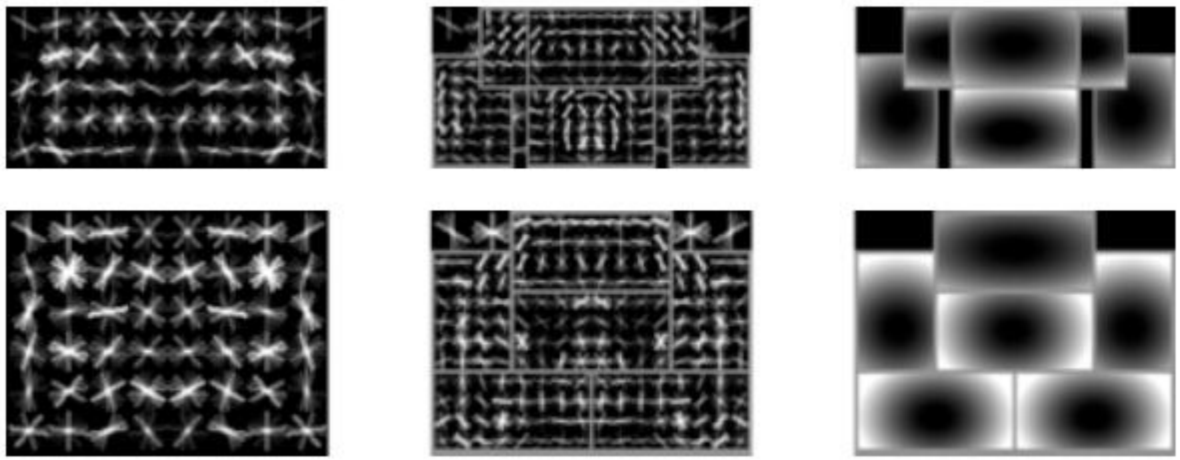
Update model using gradient descent

Trow away (x,z) pairs with low score

- Sequence of training rounds
  - Train root filters
  - Initialize parts from root
  - Train final model



Car model

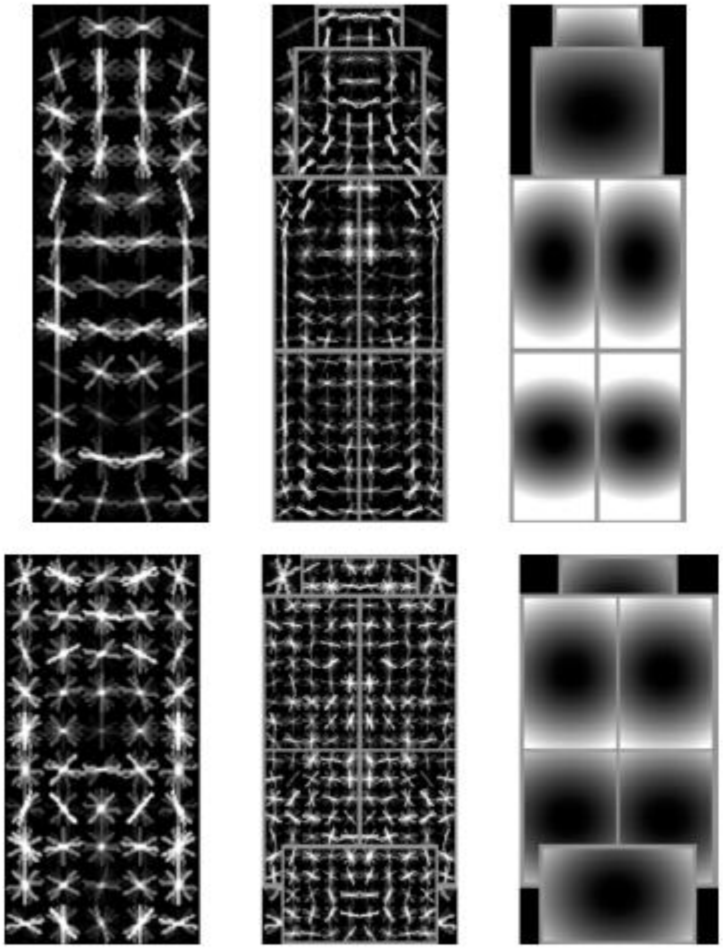


root filters
coarse resolution

part filters
finer resolution

deformation
models

Bottle model



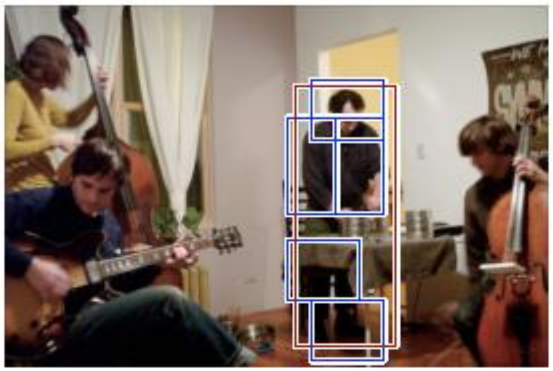
root filters
coarse resolution

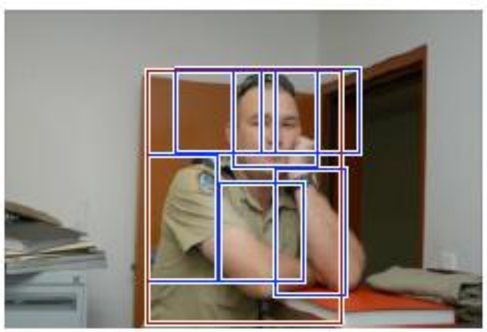
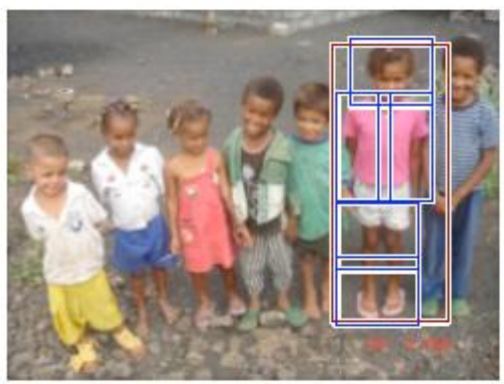
part filters
finer resolution

deformation
models

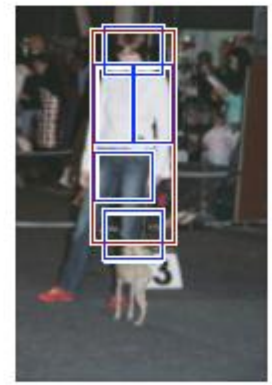
Person detections

high scoring true positives



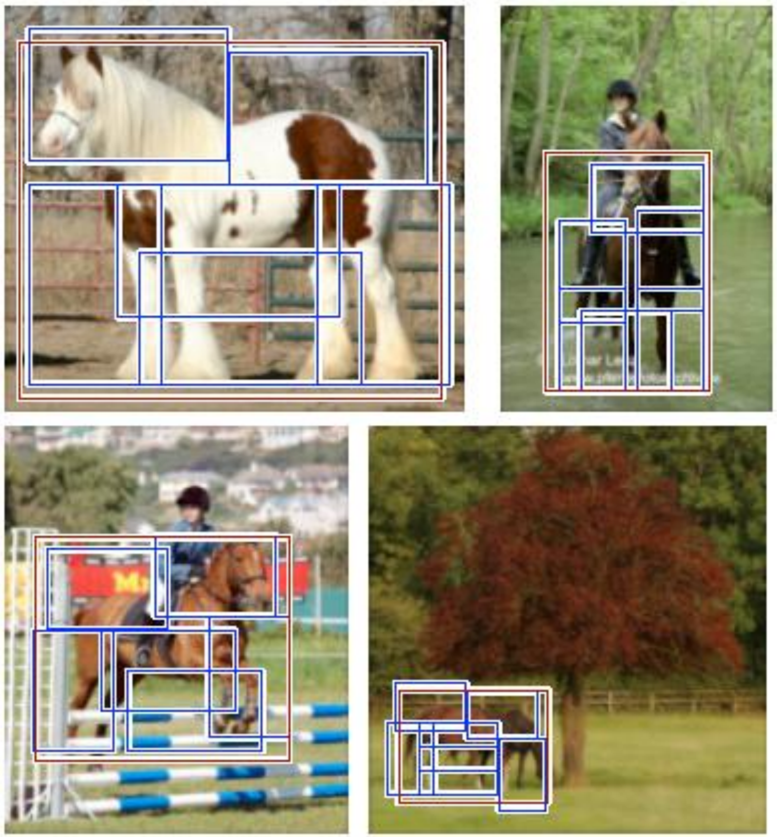



high scoring false positives
(not enough overlap)

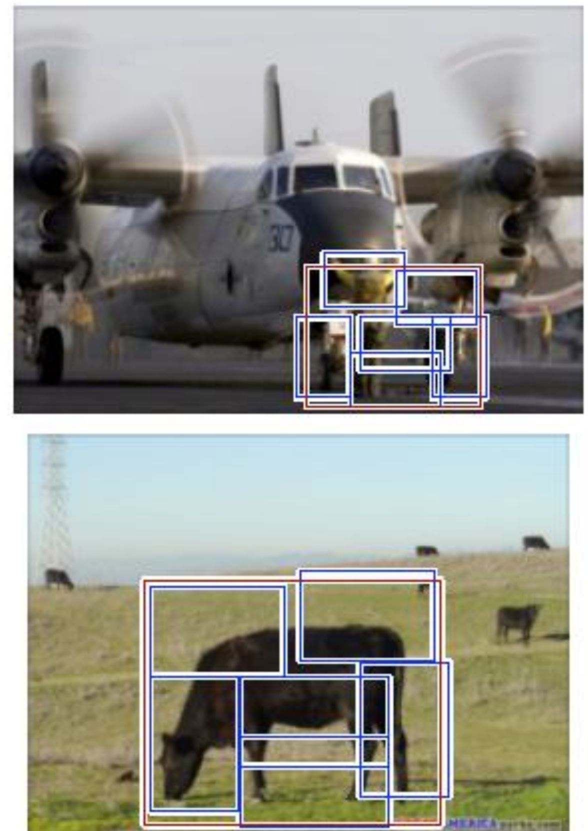



Horse detections

high scoring true positives

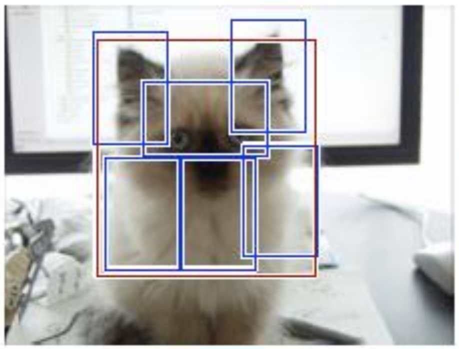


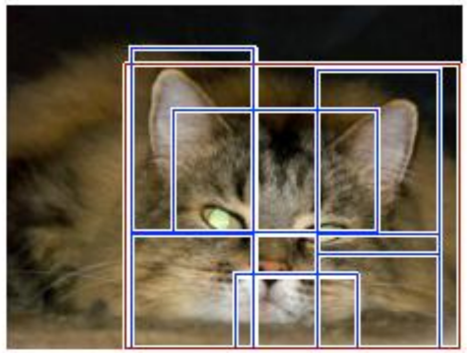
high scoring false positives

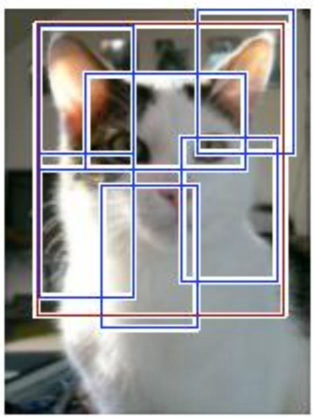


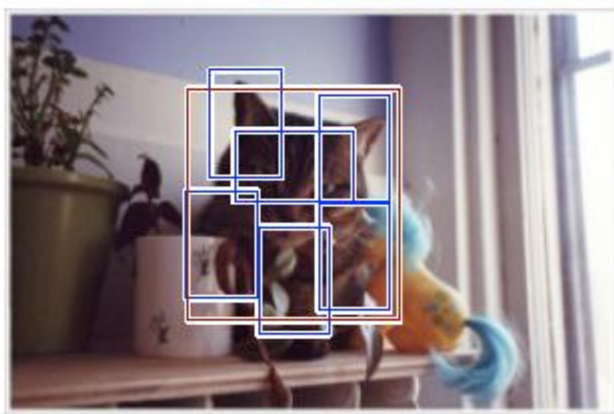
Cat detections

high scoring true positives

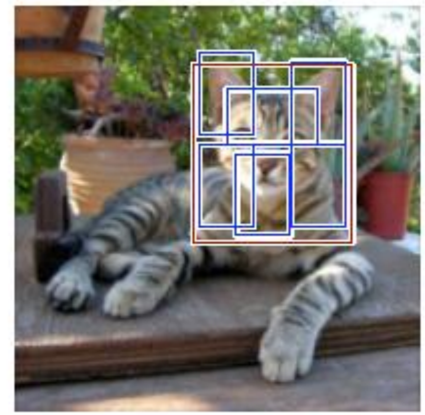


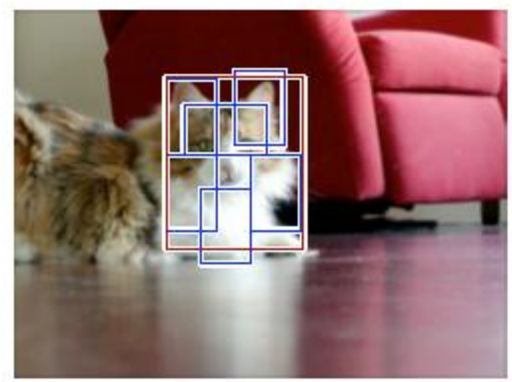






high scoring false positives
(not enough overlap)

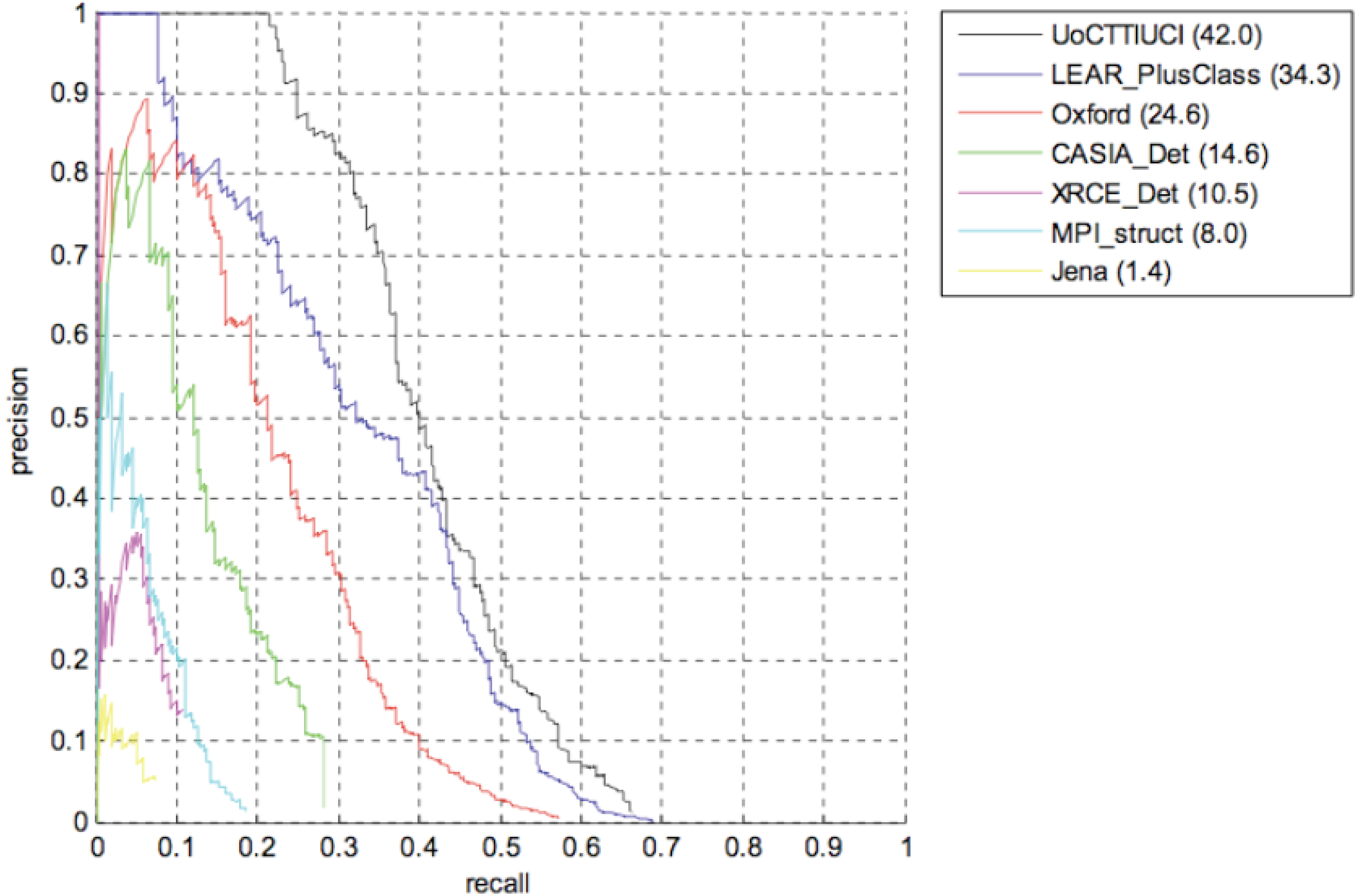




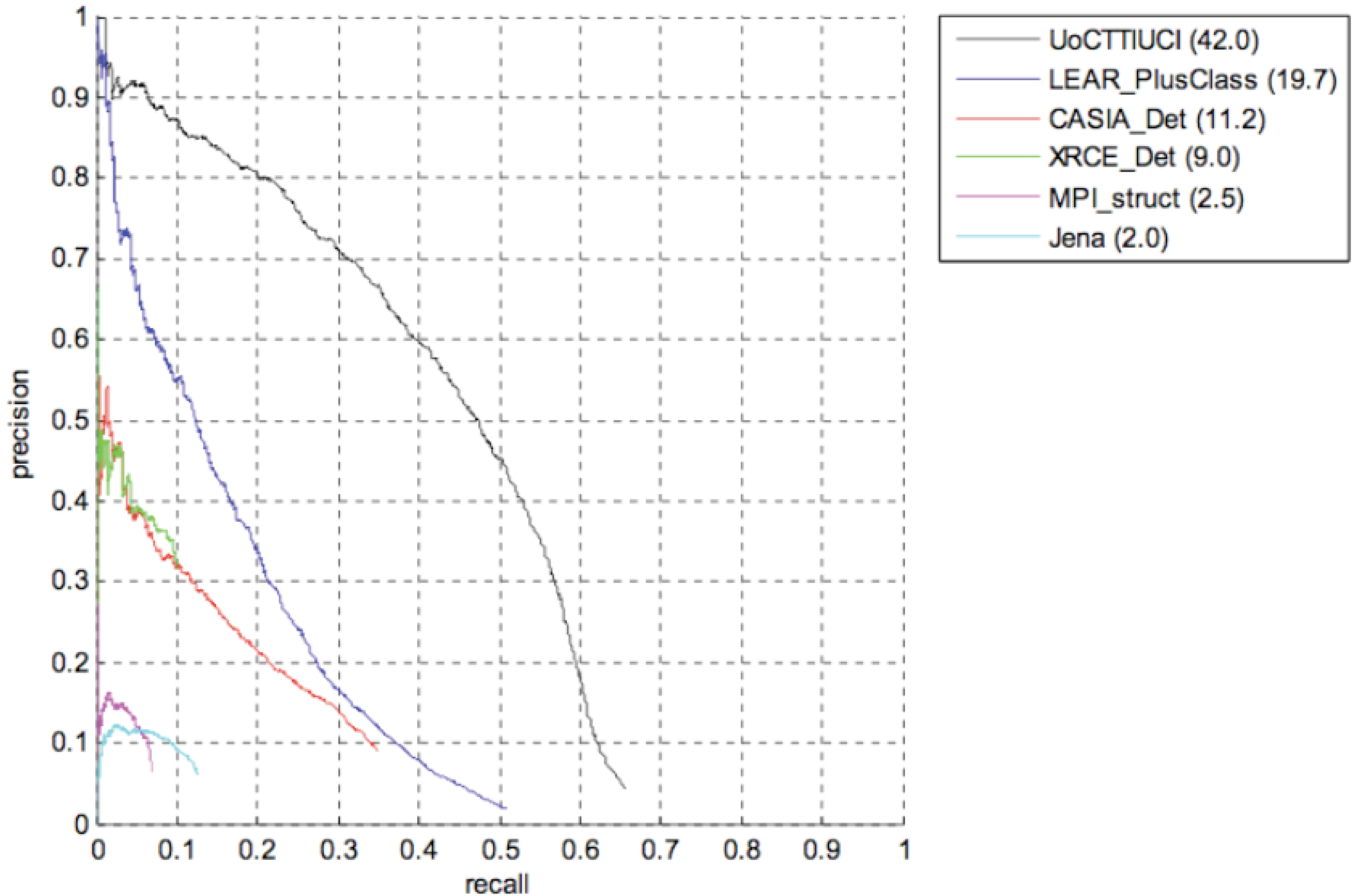
Quantitative results

- 7 systems competed in the 2008 challenge
- Out of 20 classes we got:
  - First place in 7 classes
  - Second place in 8 classes
- Some statistics:
  - It takes ~2 seconds to evaluate a model in one image
  - It takes ~4 hours to train a model
  - MUCH faster than most systems.

Precision/Recall results on Bicycles 2008

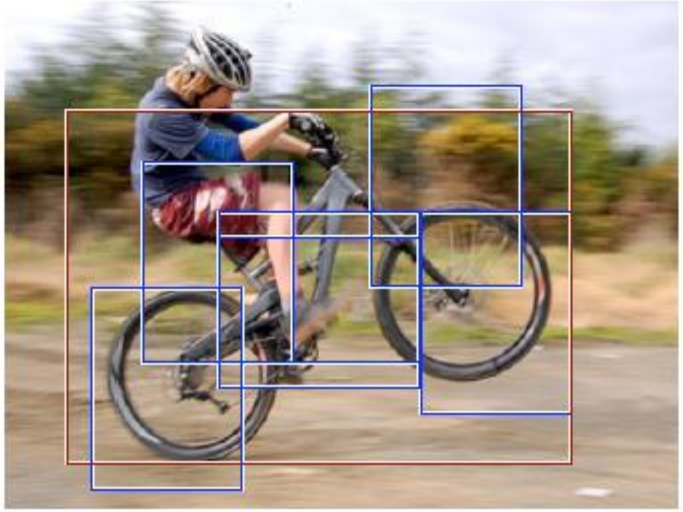


Precision/Recall results on Person 2008



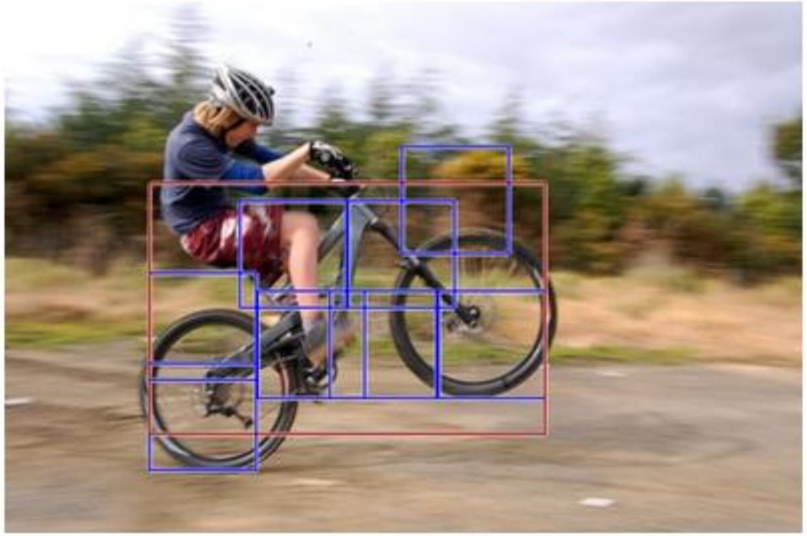
Summary

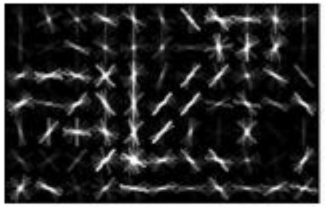
- Deformable models for object detection
  - Fast matching algorithms
  - Learning from weakly-labeled data
  - Leads to state-of-the-art results in PASCAL challenge
- Future work:
  - Hierarchical models
  - Visual grammars
  - AO* search (coarse-to-fine)

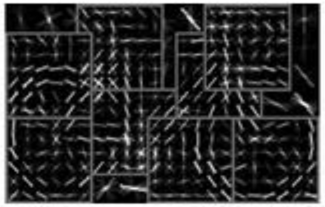


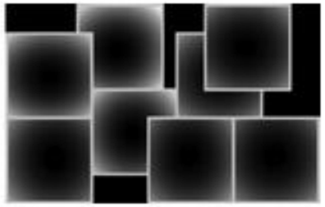
Discriminatively Trained Deformable Part Models

Version 4. Updated on April 21, 2010.









er the past few years we have developed a complete learning-based system for detecting and localizing objects in images. Our system represents objects using tures of deformable part models. These models are trained using a discriminative method that only requires bounding boxes for the objects in an image. The proach leads to efficient object detectors that achieve state of the art results on the PASCAL and INRIA person datasets.

a high level our system can be characterized by the combination of
Strong low-level features based on histograms of oriented gradients (HOG).
Efficient matching algorithms for deformable part-based models (pictorial structures).
Discriminative learning with latent variables (latent SVM).

SCAL VOC "Lifetime Achievement" Prize