Human Body Analysis Towards Human Interactions

Petia Ivanova Radeva

BCN Perceptual Computing Lab

Introduction

- Applications
  - Pedestrian detection for smart cars
  - Visual surveillance, behavior analysis
  - Images, films and multi-media analysis

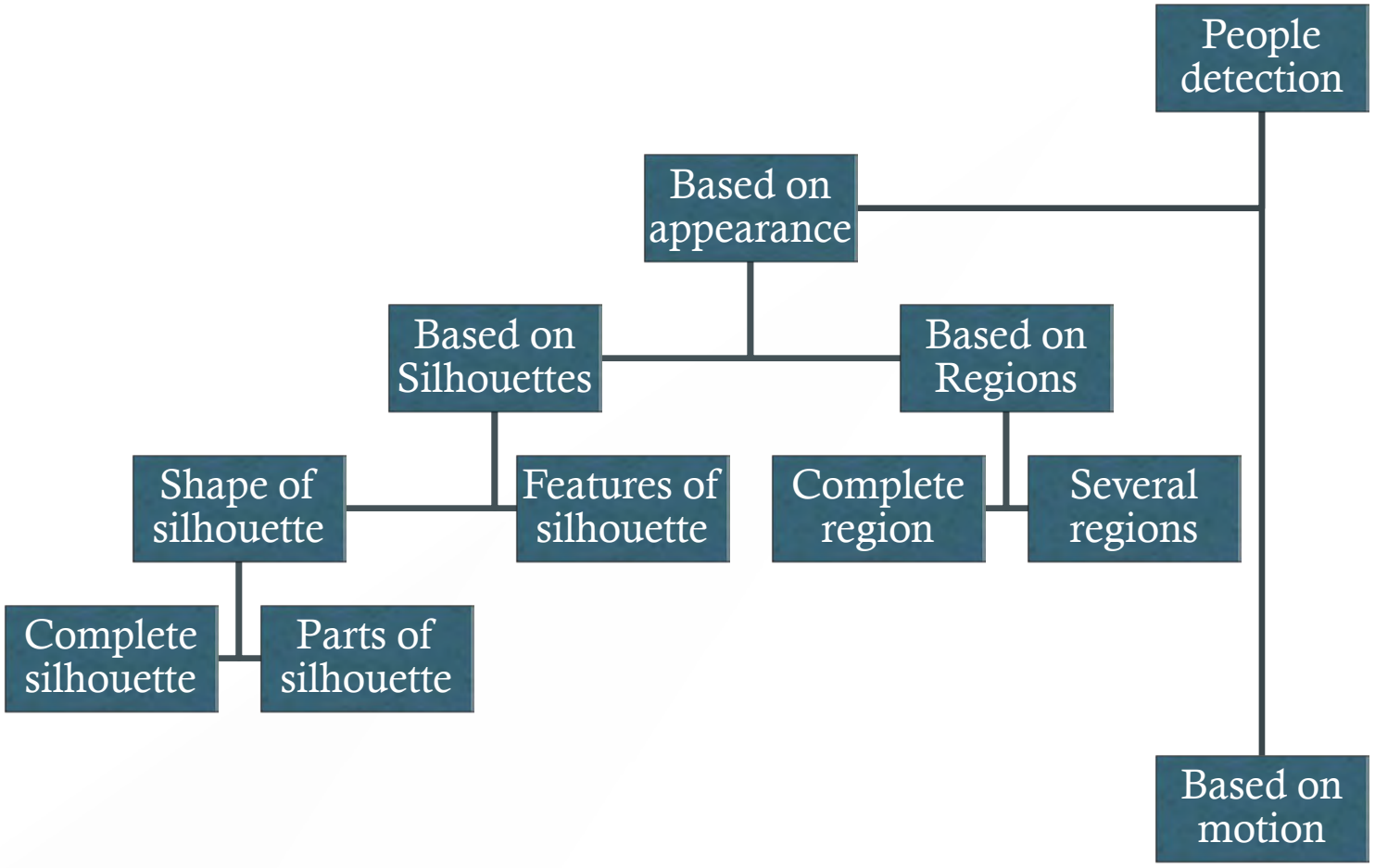
Difficulties

- Wide variety of points of view & scales
- Complex background and occlusions
- Unconstrained illumination
- Videos with moving subject, camera, background

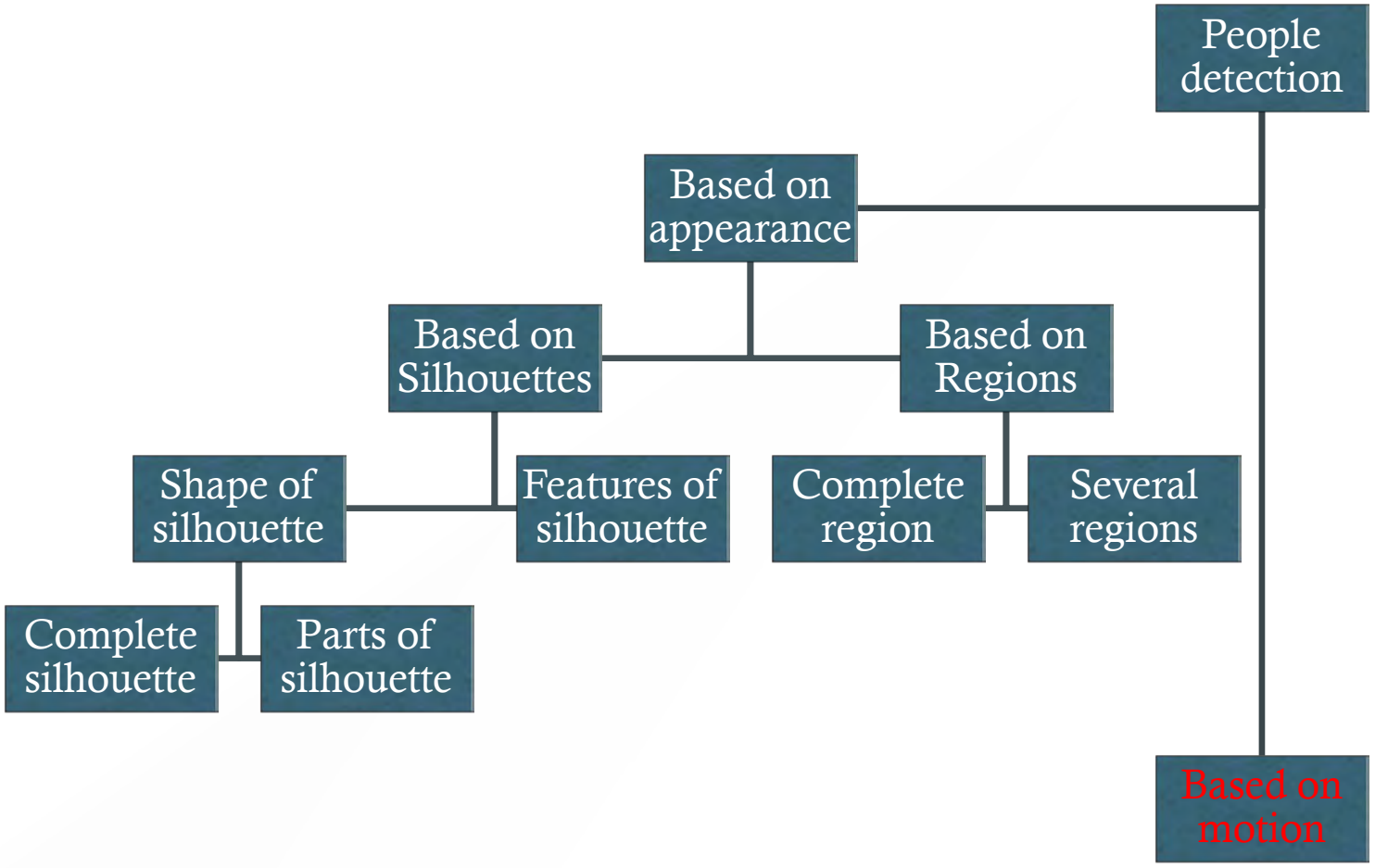
Why is it hard?

- Objects in rich categories exhibit significant variability
- Intra-class variability
  - Cars come in a variety of shapes (sedan, minivan, etc)
  - People wear different clothes and take different poses
- Pose Intra-class variability
- Non-rigid deformations
- We need rich object models
  - But this leads to difficult matching and training problems

People detection



People detection

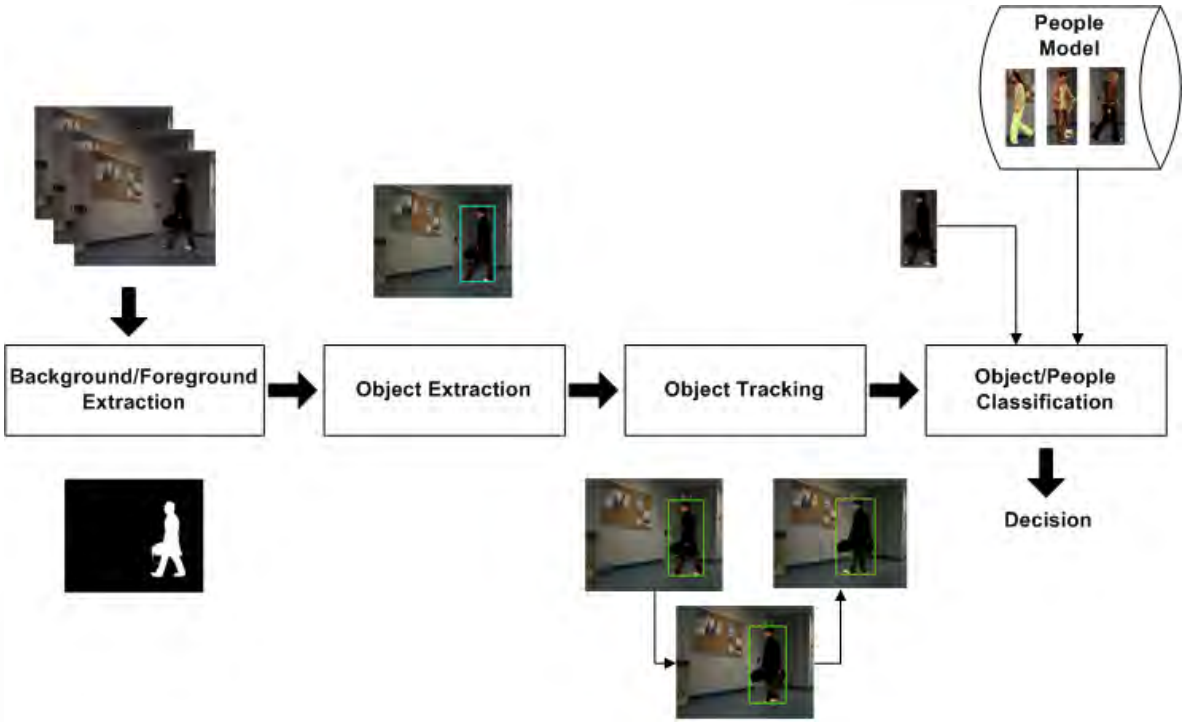


Motion-based Methods

- Methods based on motion
  - Requires sequences of images
  - Try to avoid appearance variability
    - Environmental factors
      - Light conditions, clothing, contrast, etc.
    - Intrinsic people variability
      - Different heights, widths, poses, etc.
  - Detection based only on motion information

Motion-based Methods

- Canonical sequence analysis system
  - Required for motion-based approaches
  - Useful for appearance-based approaches



Motion-based Methods

- Background/foreground extraction
  - Moving objects are segmented from static background
  - The popular idea is to model temporal samples in multi-modal distribution, in either parametric or nonparametric way
    - GMM (parametric) is the most popular technique. Each pixel is modeled independently using a mixture of Gaussians and updated by an online approximation.

Motion-based Methods

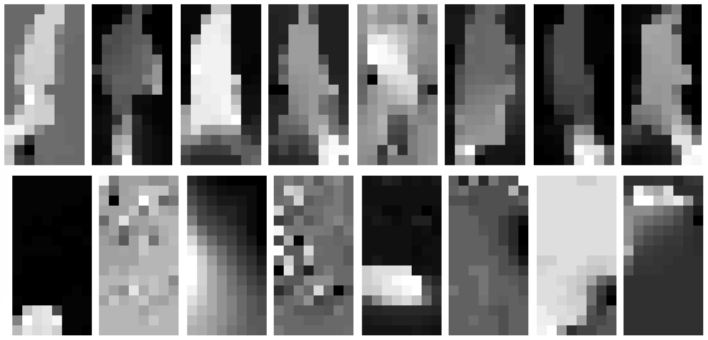
- GMM Background modeling
  - Initial background model
    - The first N frames of the input sequence
    - K-means clustering
  - The history of each pixel, $\{X_1,...,X_N\}$, is modeled by three independent Gaussian distributions, and X is the color value, i.e. $X = \{X^R, X^G, X^B\}$
  - For computational reasons, we assume the red, green, and blue pixel values are independent.

Motion-based Methods

- GMM Background subtraction

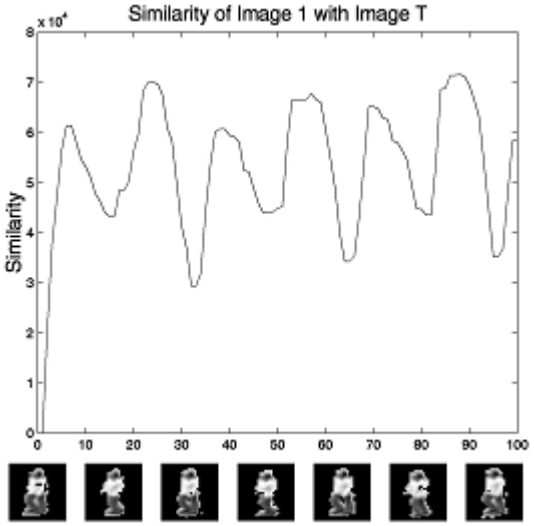
Motion-based Methods

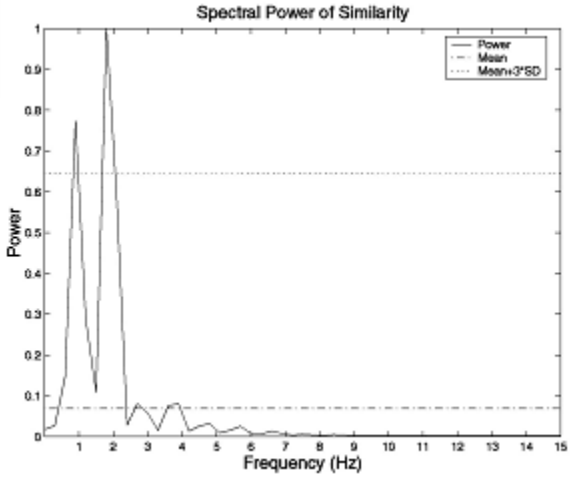
- Motion-based classification
  - Motion patterns [H. Sidenbladh 2004]
    - For each object present in two consecutive images
      - Size normalization is performed
      - Flow pattern is calculated using dense optical horizontal and vertical flows
    - The resulting pattern is then classified using a Support Vector Machine (SVM)



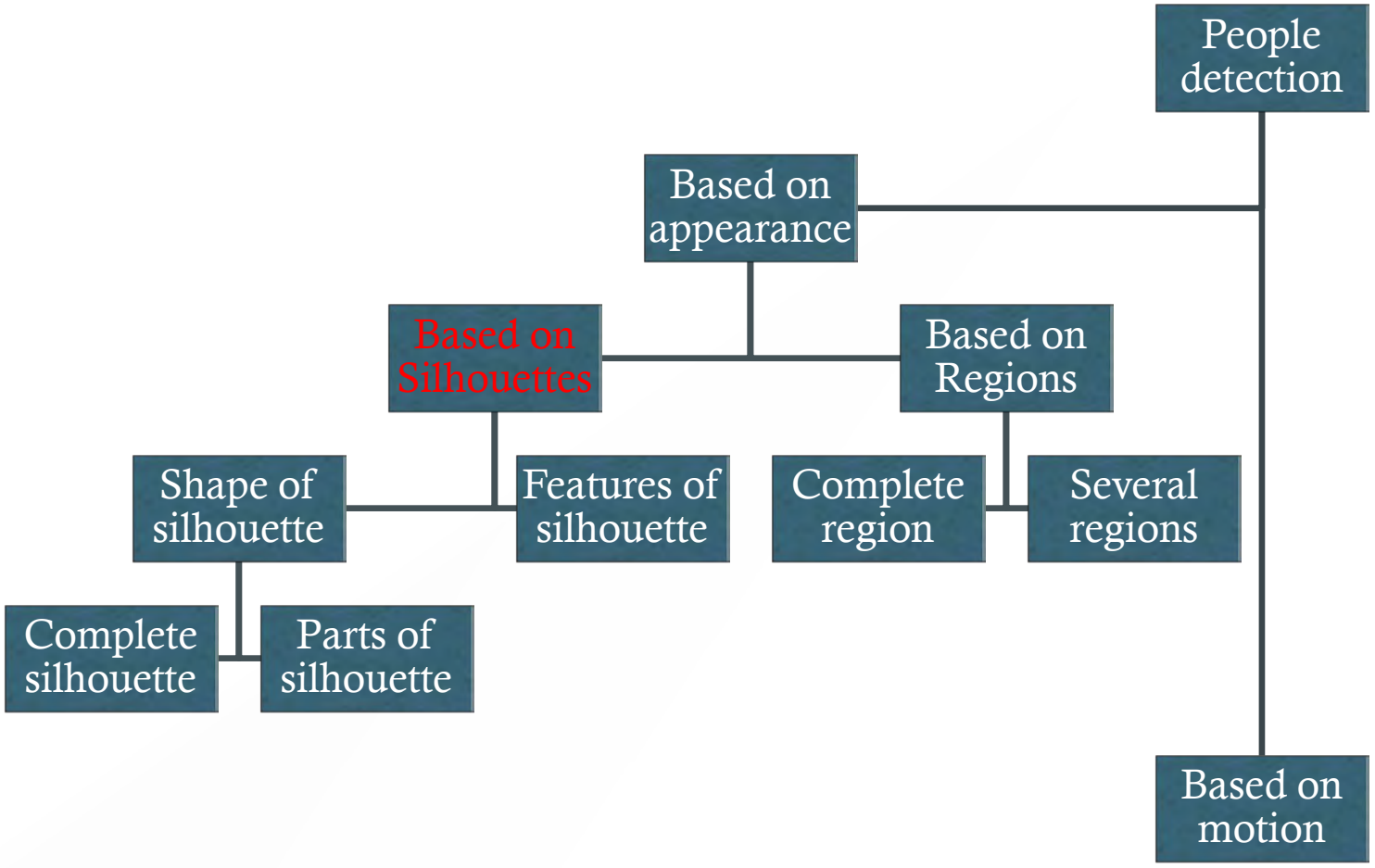
Motion-based Methods

- Motion-based classification
  - Periodic motion analysis [Cutler & Davis 2000].
    - Segment and track moving objects
    - Align each object along time
    - Compute the object's self-similarity and how it evolves in time.
    - Analyze the periodicity of this measure using Time-Frequency analysis
    - Classify objects using periodicity





People detection

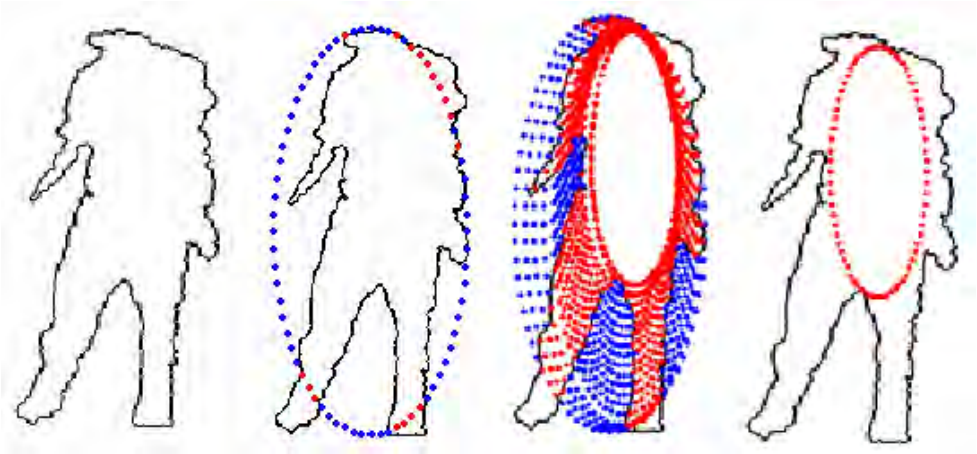


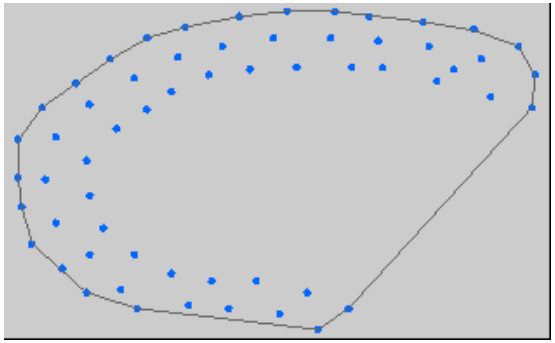
Appearance-based Methods

- Most of the existing approaches use people appearance information.
  - Based on silhouettes
    - Focused on extracting objects contours and over this processed image creating or adjusting their classification model.
  - Based on regions
    - Do not need to extract contours.
    - Look for regions over the image which could be objects and put into practice their classification model.

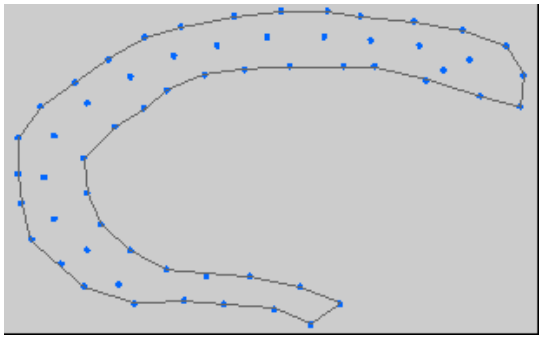
Appearance-based Methods

- Based on silhouettes
  - Silhouette features
    - Aspect ratio
    - Ellipse fitting
    - Convex and concave hull





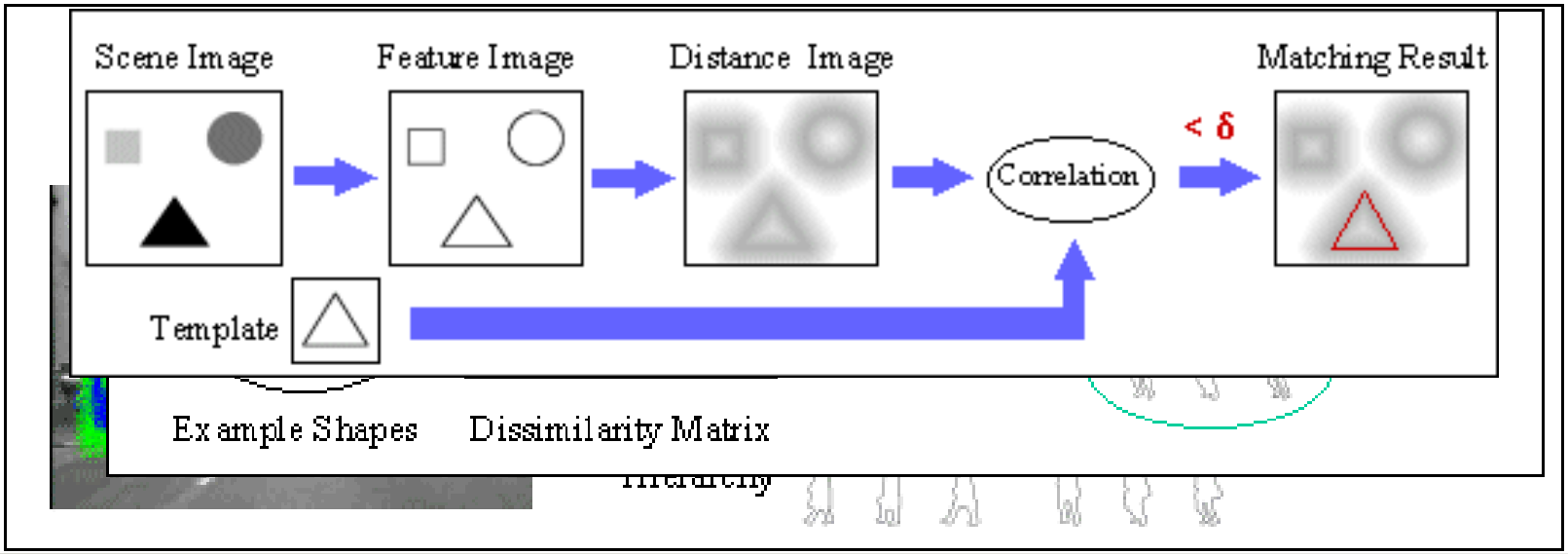
Convex hull



Concave hull

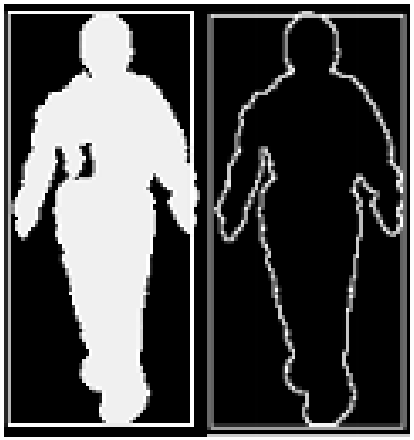
Appearance-based Methods

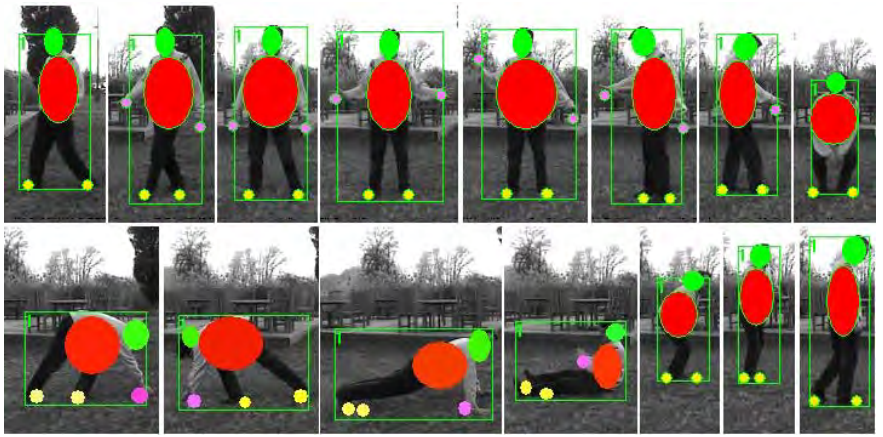
- Based on silhouettes
  - Silhouette shape
    - Model of person as a complete silhouette
      - Use trained codebook of people shapes in order to classify between humans and other objects
      - Distance metric



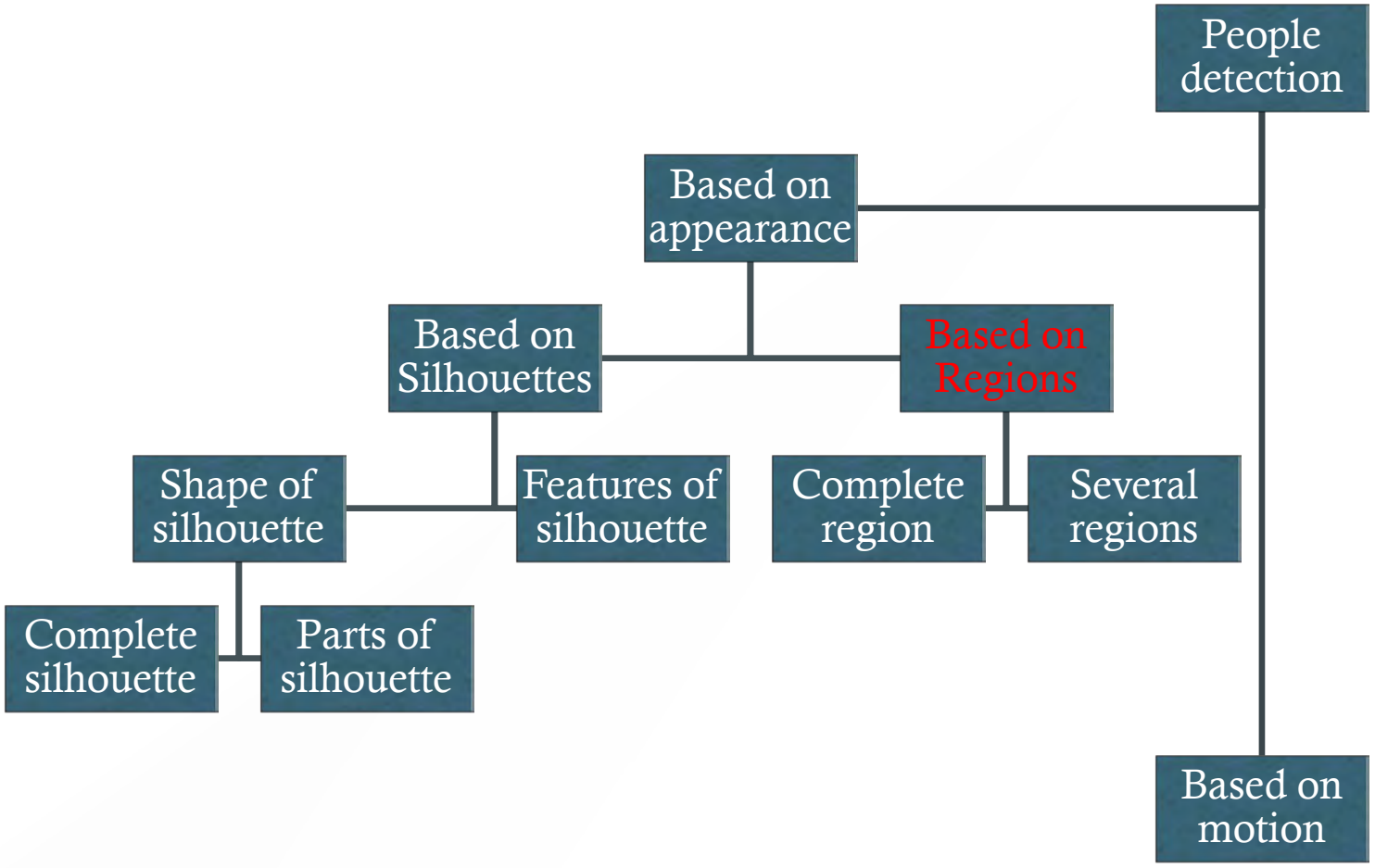
Appearance-based Methods

- Based on silhouettes
  - Silhouette shape
    - Model of person as the union of parts of the same silhouette
      - Head, arms, legs, etc.
      - Estimates the human body posture (standing, sitting, …) using normalized horizontal and vertical projection histograms and posture models previously trained



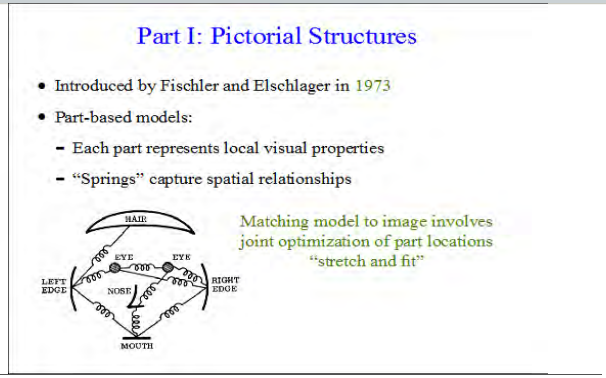


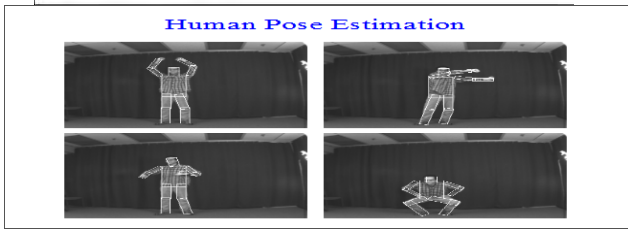
People detection

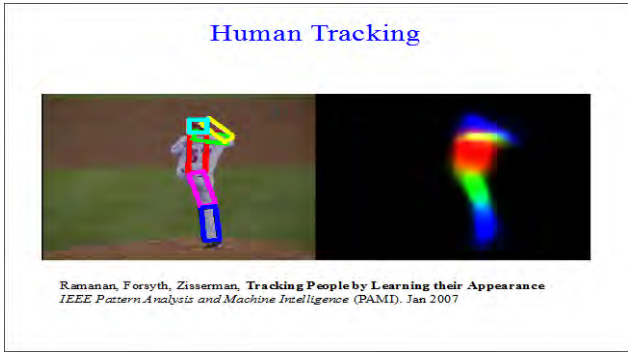


Pictorial structures

- Pictorial structures represent objects by a collection of parts arranged in a deformable configuration.
  - Introduced by Fischler and Elschlager in 1973
- Part-based (deformable) models (DM):
  - Each part captures local appearance properties of an object.
  - The deformable configuration is characterized by spring-like connections between certain parts.
- Matching model to image involves joint optimization of part locations “stretch and fit”.



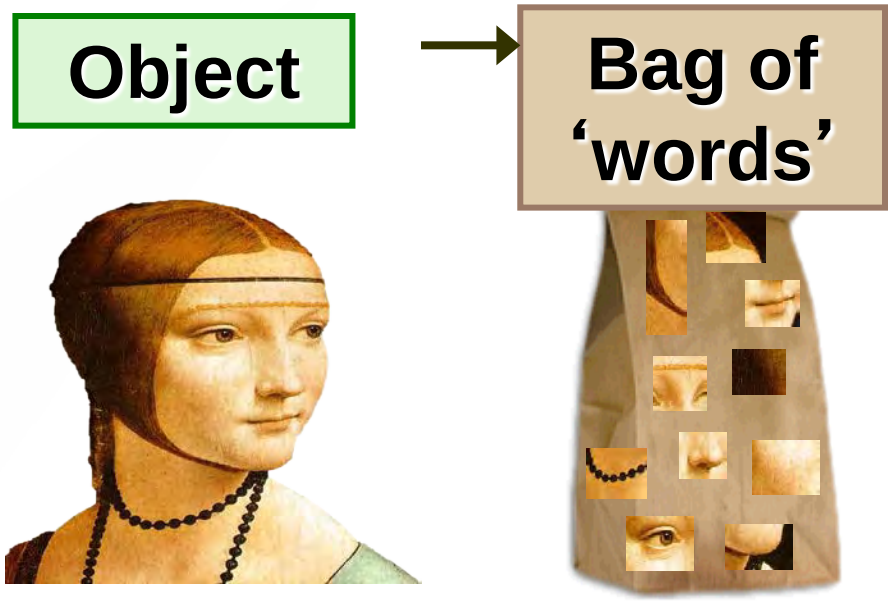




Pictorial structures vs. Bag-of-words

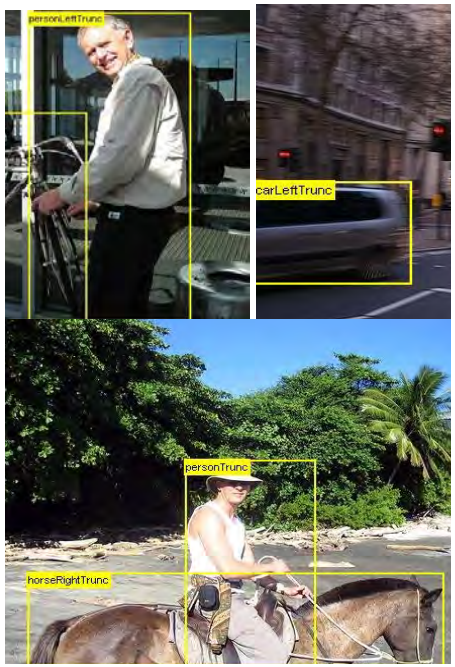
Simple models have historically outperformed sophisticated models since:

- Rich models suffer from difficulties in training, often using latent information
- A single DM is not expressive enough to represent rich object categories (e.g. Bicycles)
- On difficult datasets DM were outperformed by rigid templates or bag-of-features.



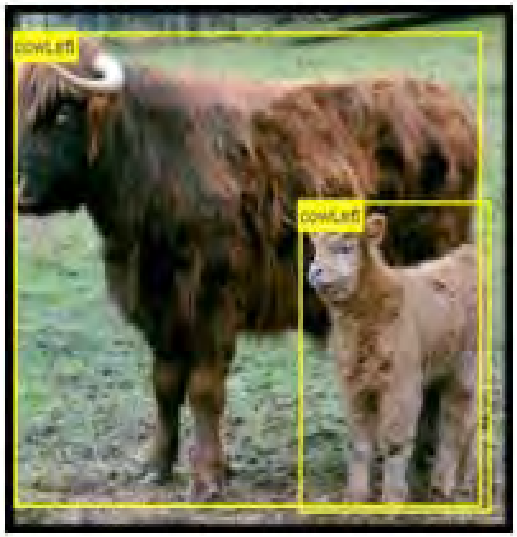
Use of latent information

- If the whole information is the bounding box
  - the algorithm should deduce the parts of the human body.
- More complete labeling might support better training
  - but it can use suboptimal parts
  - time consuming and expensive.
- Challenge: How to define a robust human body detector by a discriminatively trained part-based model?!

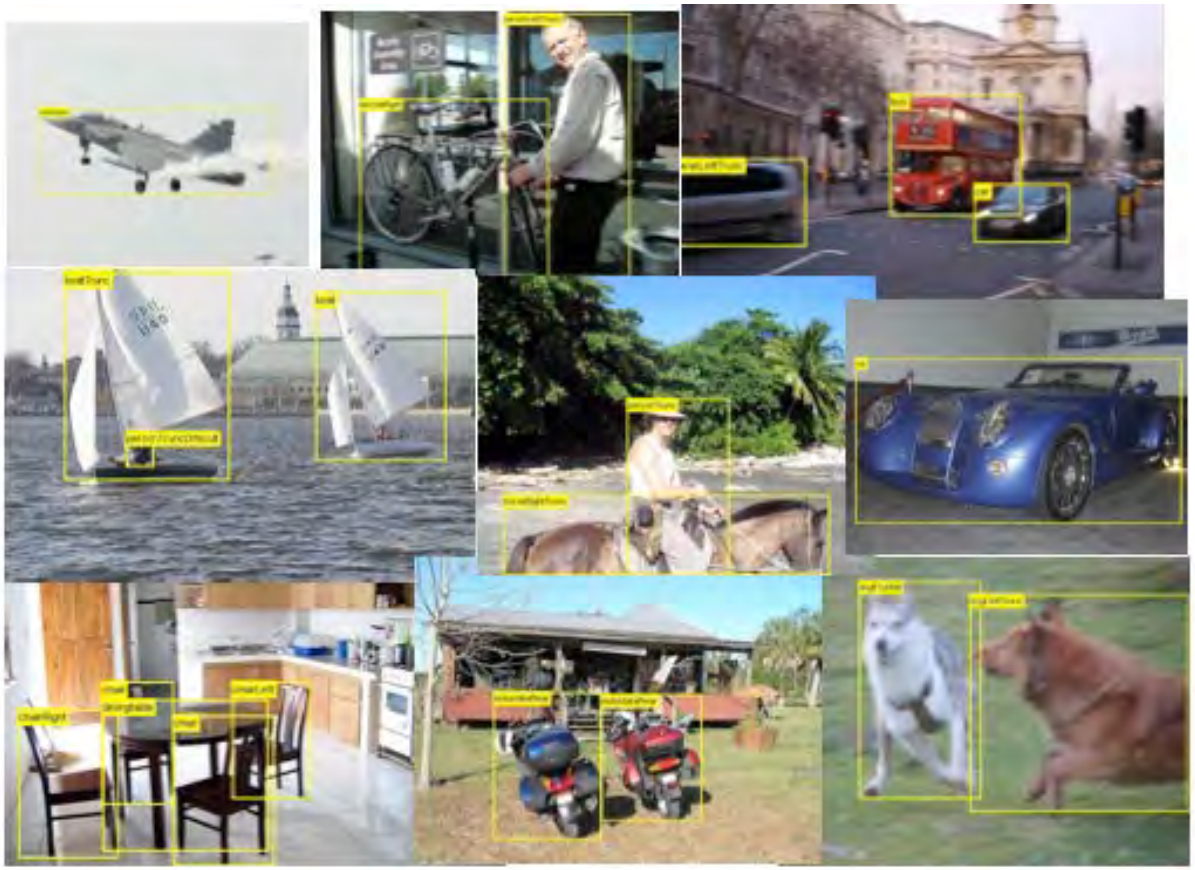


Pascal challenge

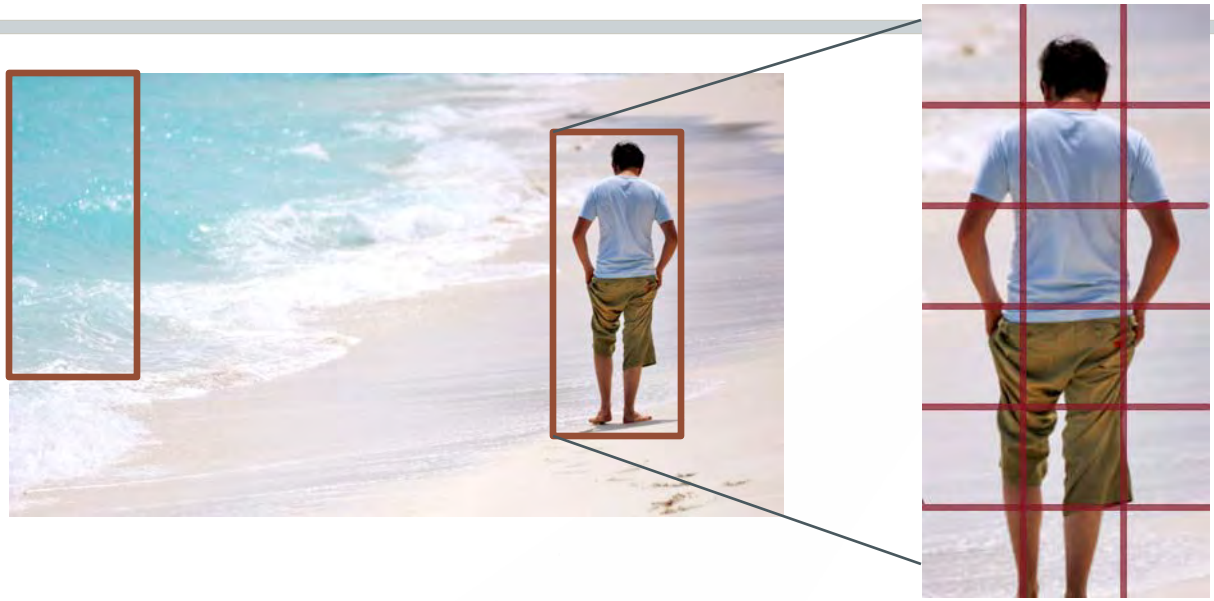
- ~10,000 images, with ~25,000 target objects
  - Objects from 20 categories (person, car, bicycle, cow, table...)
  - Objects are annotated with labeled bounding boxes



Pascal challenge



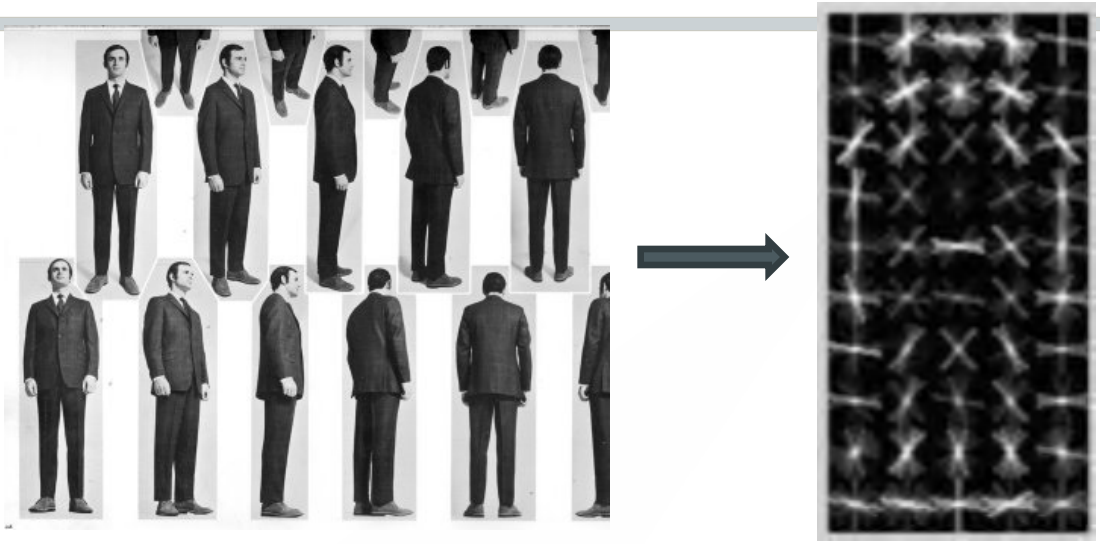
Starting point: sliding window classifiers



Feature Vector
X = […, …, …, …]

- Detect objects by testing each subwindow
  - Reduce object detection to binary classification
  - Detection models: Linear Filters & Feature Map
  - Feature map: Array of Feature Vectors (local image patch)

Histogram of Gradient (HOG) feature



- Image is partitioned into 8x8 pixels blocks
- Compute in each block a histogram of gradient orientation
  - Invariant to changes in lighting, small deformation, etc

Gradients computation

-1	0	1

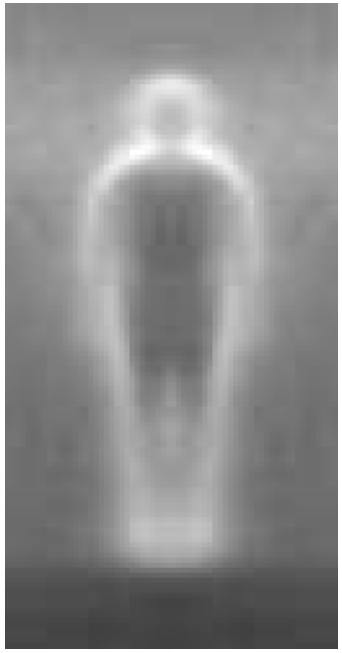
centered

-1	1

uncentered

1	-8	0	8	-1

cubic-corrected



0	1
-1	0

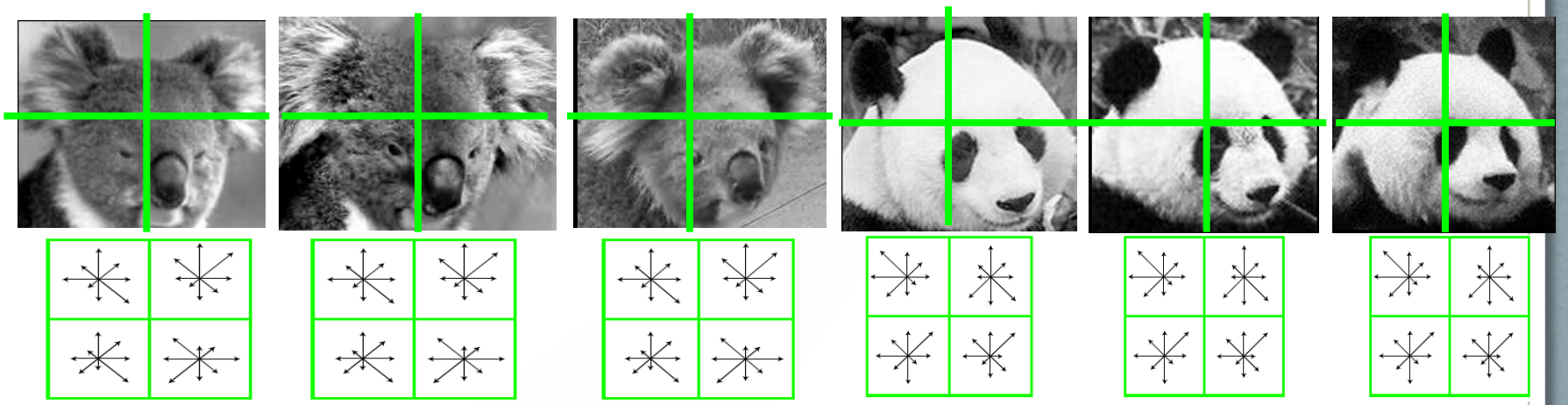
diagonal

-1	0	1
-2	0	2
-1	0	1

Sobel

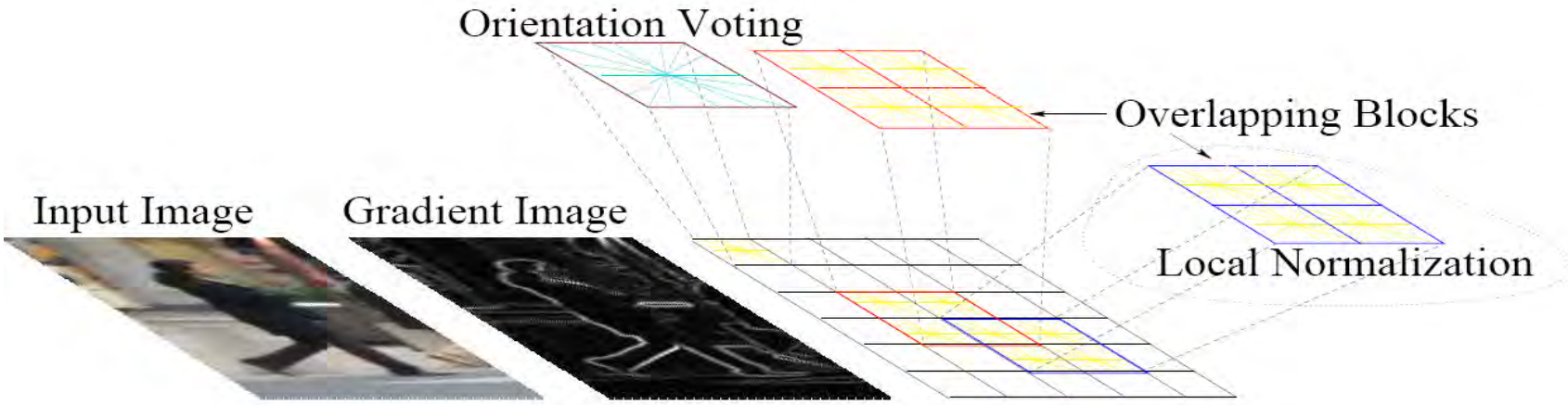
Gradient-based representations

- Consider edges, contours, and (oriented) intensity gradients



- Summarize local distribution of gradients with histogram
  - Locally orderless: offers invariance to small shifts and rotations
  - Contrast-normalization: try to correct for variable illumination

Gradient-based representations: histograms of oriented gradients (HoG)



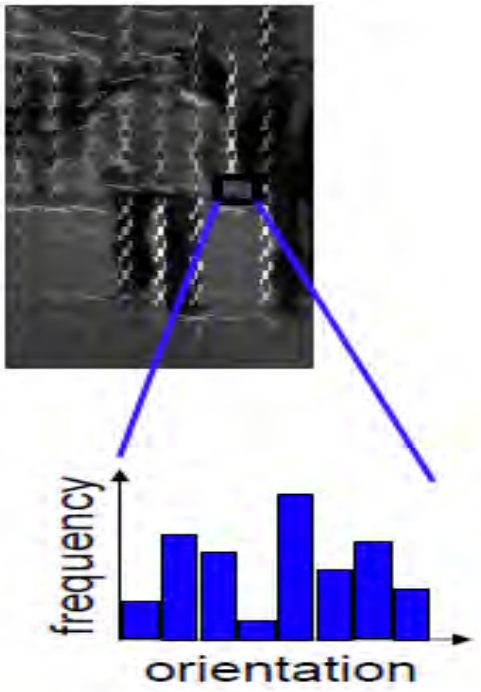
Map each grid cell in the input window to a histogram counting the gradients per orientation.

Code available: http://pascal.inrialpes.fr/soft/olt/

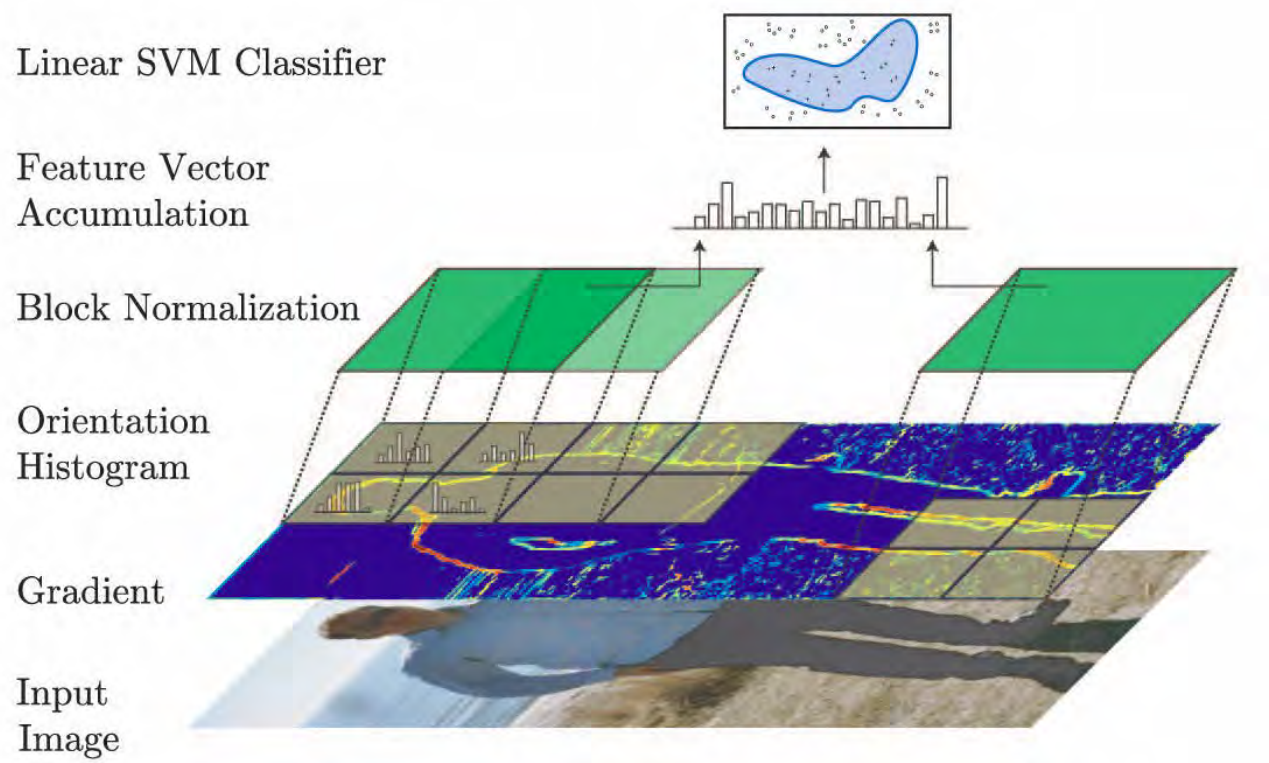
Dalal & Triggs, CVPR 2005

Dalal & Triggs: HOG + linear SVMs

- Collect HOG's over detection window
  - Build a feature vector from HOGs
    - Dimension = 16 x 8 (for tiling) x 8 (orientations) = 1024

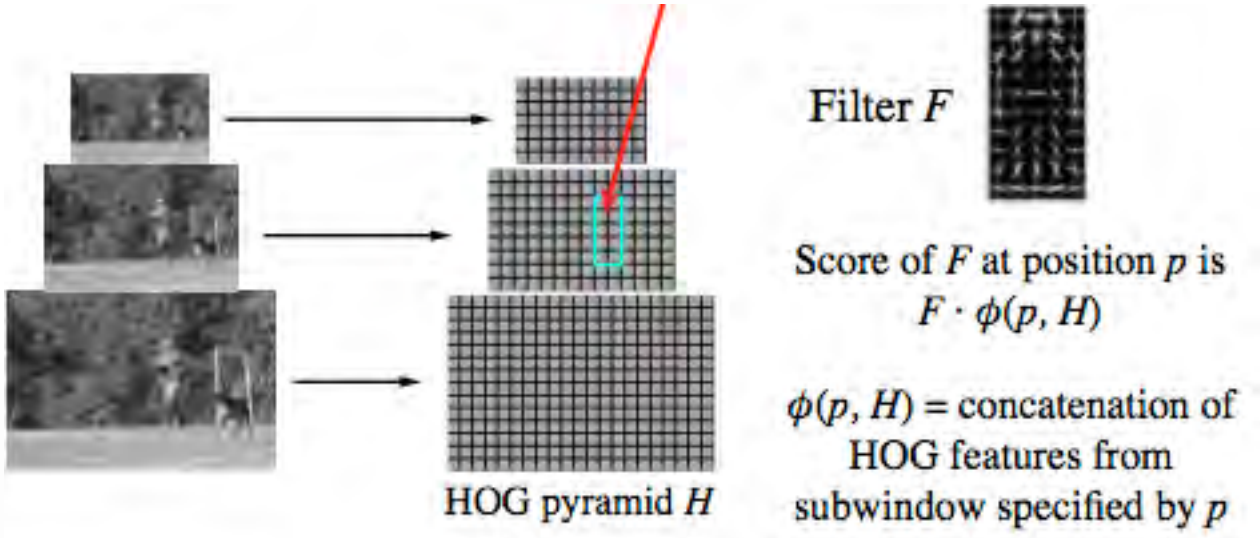


Dalal & Triggs: HOG + linear SVMs

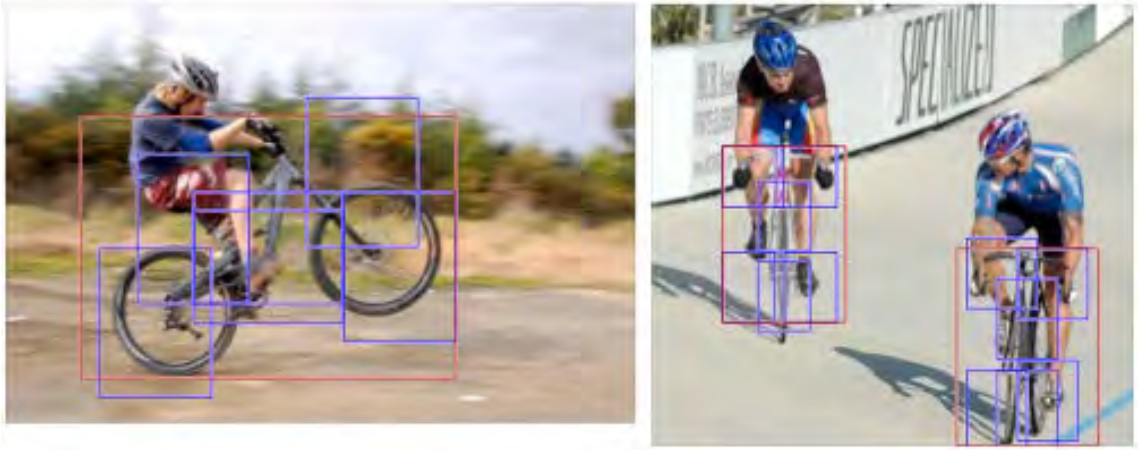


HOG Filters

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

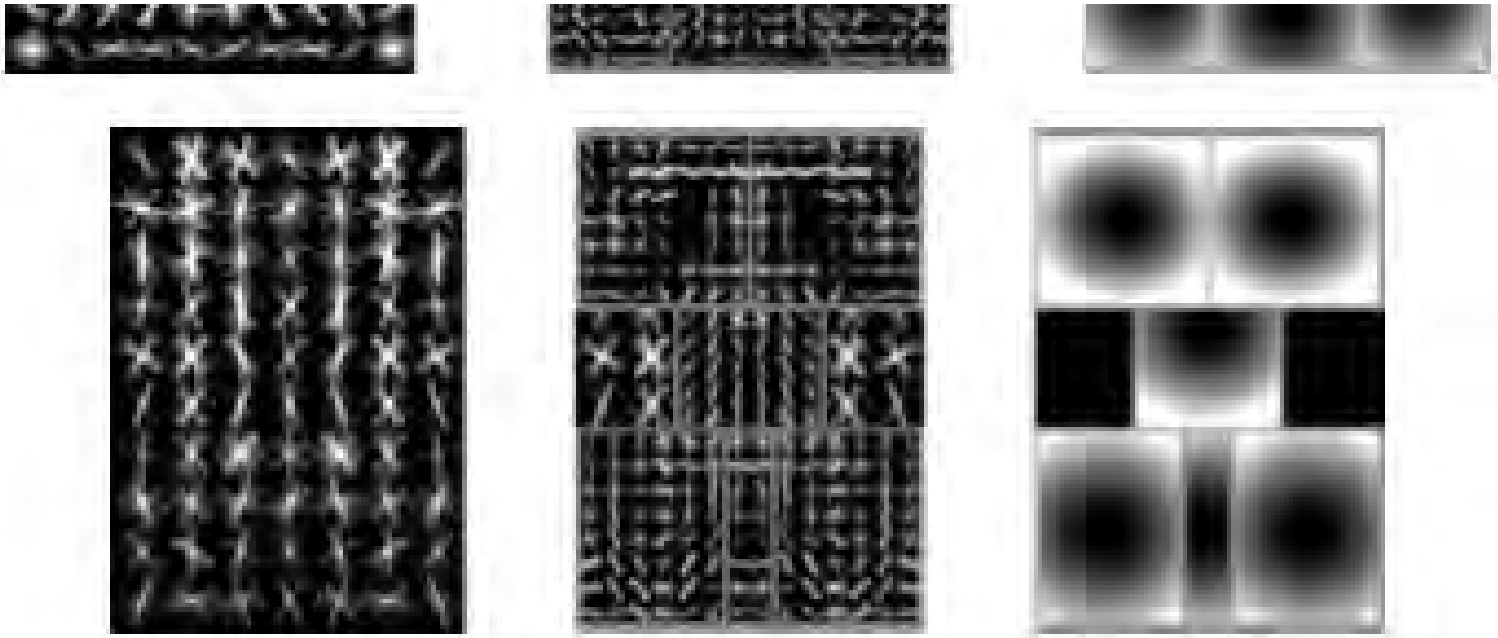


Overiew of the models



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

2 component bicycle model

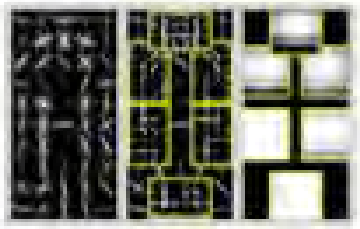


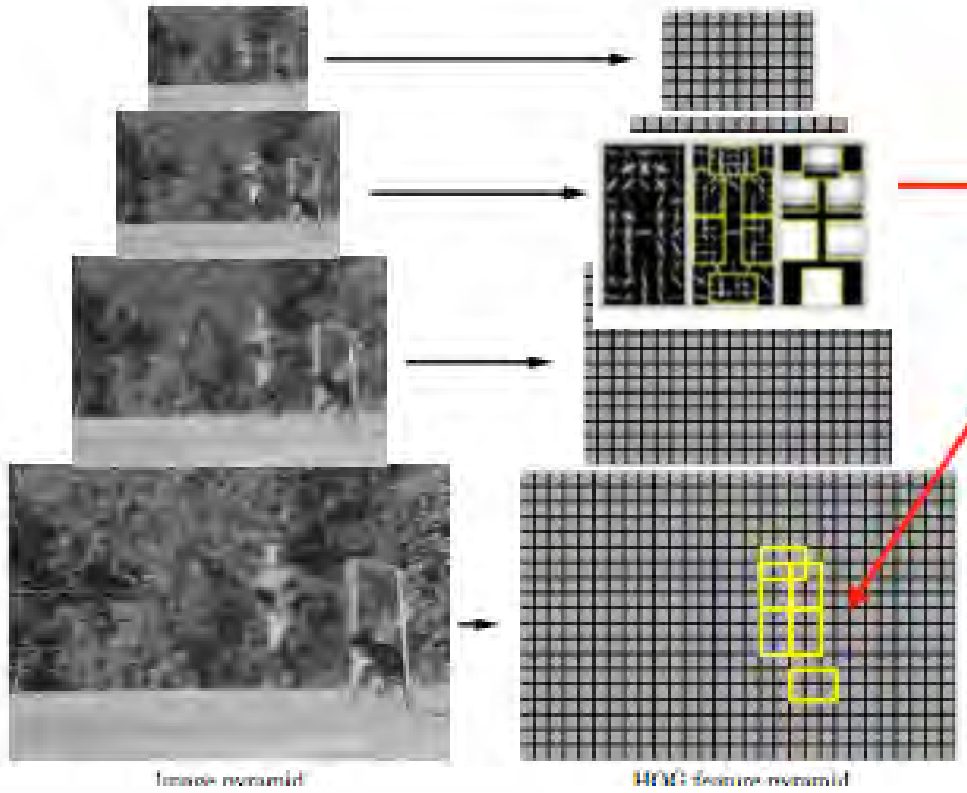
root filters
coarse resolution

part filters
finer resolution

deformation
models

Object hypothesis





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

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

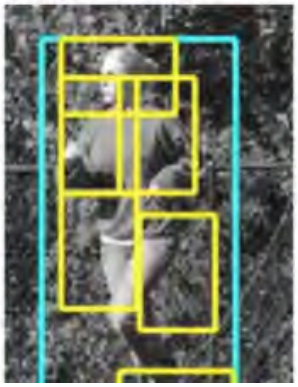
Score of a hypothesis

$$\text{score}(p_0, \ldots, p_n) = \underbrace{\sum_{i=0}^{n} F_i \cdot \phi(H, p_i)}_{\text{“data term”}} - \underbrace{\sum_{i=1}^{n} d_i \cdot (dx_i^2, dy_i^2)}_{\text{“spatial prior”}}$$

F_i: filters

d_i: displacements

(dx_i^2, dy_i^2): deformation parameters



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

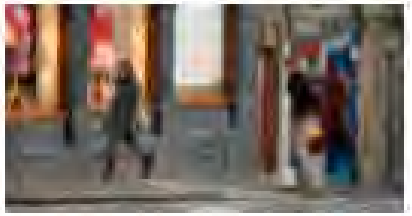
β: concatenation filters and deformation parameters

$\Psi(H, z)$: concatenation of HOG features and part

Connection with linear classifiers => Support Vector Machines.

head filter

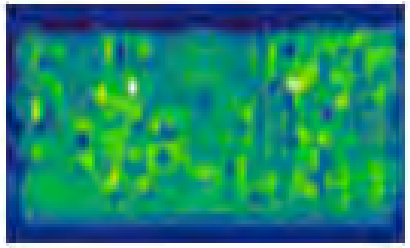
input image



Response of filter in l-th pyramid level

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

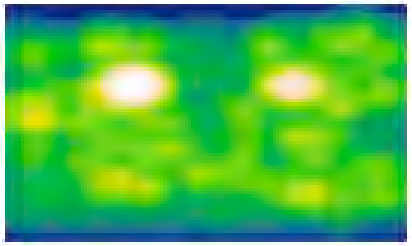
cross-correlation



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)



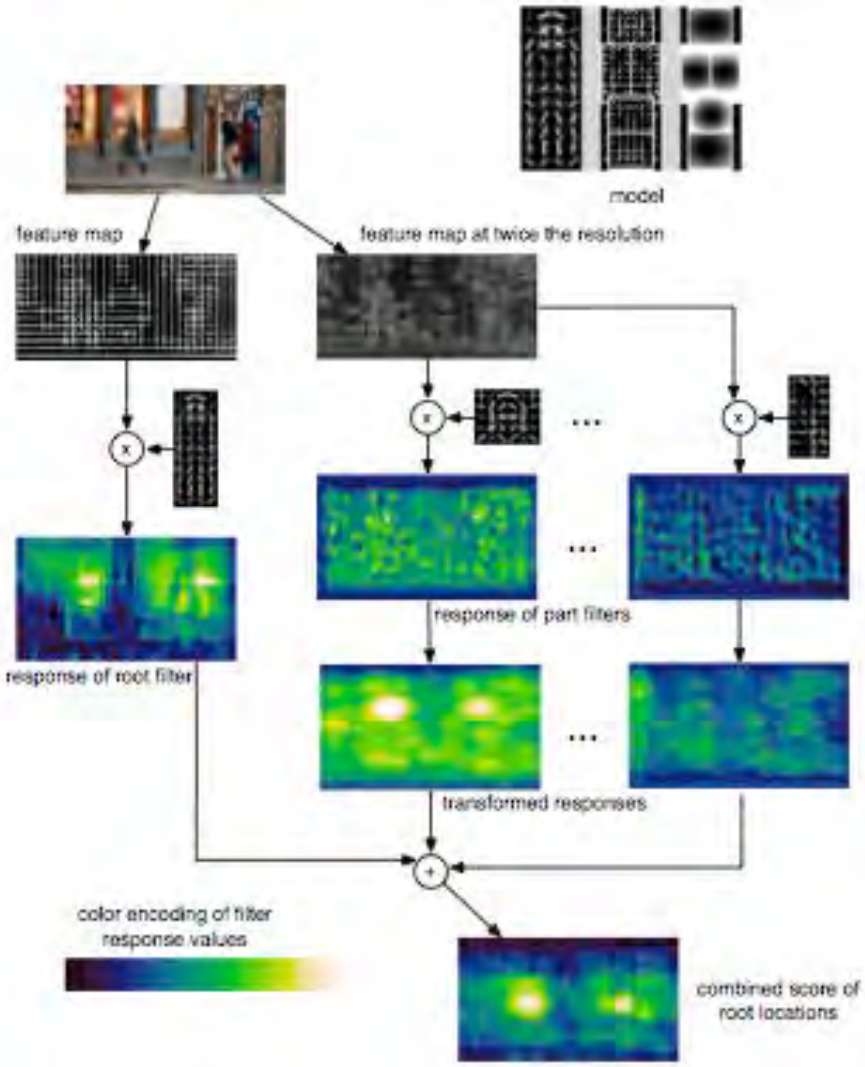
•The matching process at one scale:

•Responses from the root and part filters are computed at different resolutions in the feature pyramid.

•The transformed responses are combined to yield a final score for each root location.

- The responses and transformed responses for the "head" and "right shoulder" parts are shown.
- Note how the "head" filter is more discriminative.

•The combined scores clearly show two good hypothesis for the object at this scale.



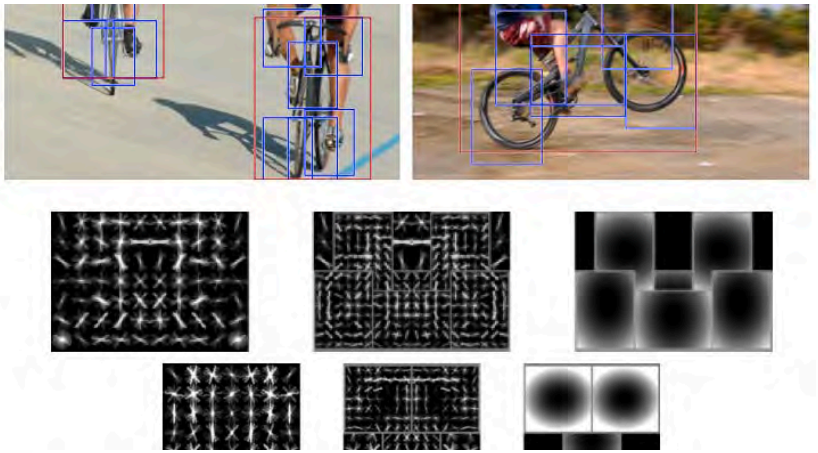
Mixture models

- A mixture model with m components is defined by a m-tuple, $M = (M_1,\ldots,M_m)$, where M_c is the model for the c-th component.

  $\beta \cdot \psi(H, z) = \beta c \cdot \psi(H, z')$, where

  $\beta = (\beta 1,\ldots,\beta m)$.
  $\psi(H,z) = (0,\ldots,0,\psi(H,z'),0,\ldots,0)$.

- The matching algorithm is used to find root locations that yield high scoring hypotheses independently for each component.



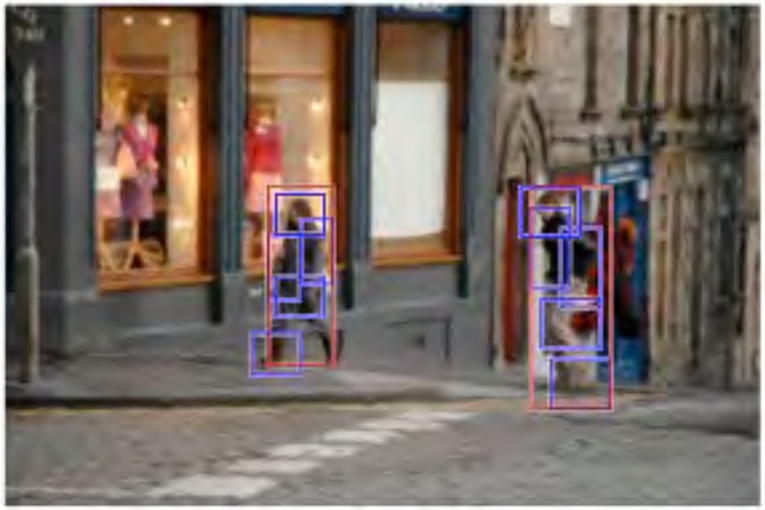
Detections obtained with a 2 component bicycle model.

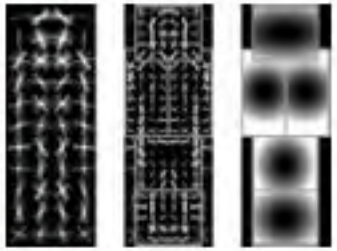
These examples illustrate the importance of deformations mixture models.

In this model the first component captures sideways views of bicycles while the second component captures frontal and near frontal views.

The sideways component can deform to match a "wheelie".

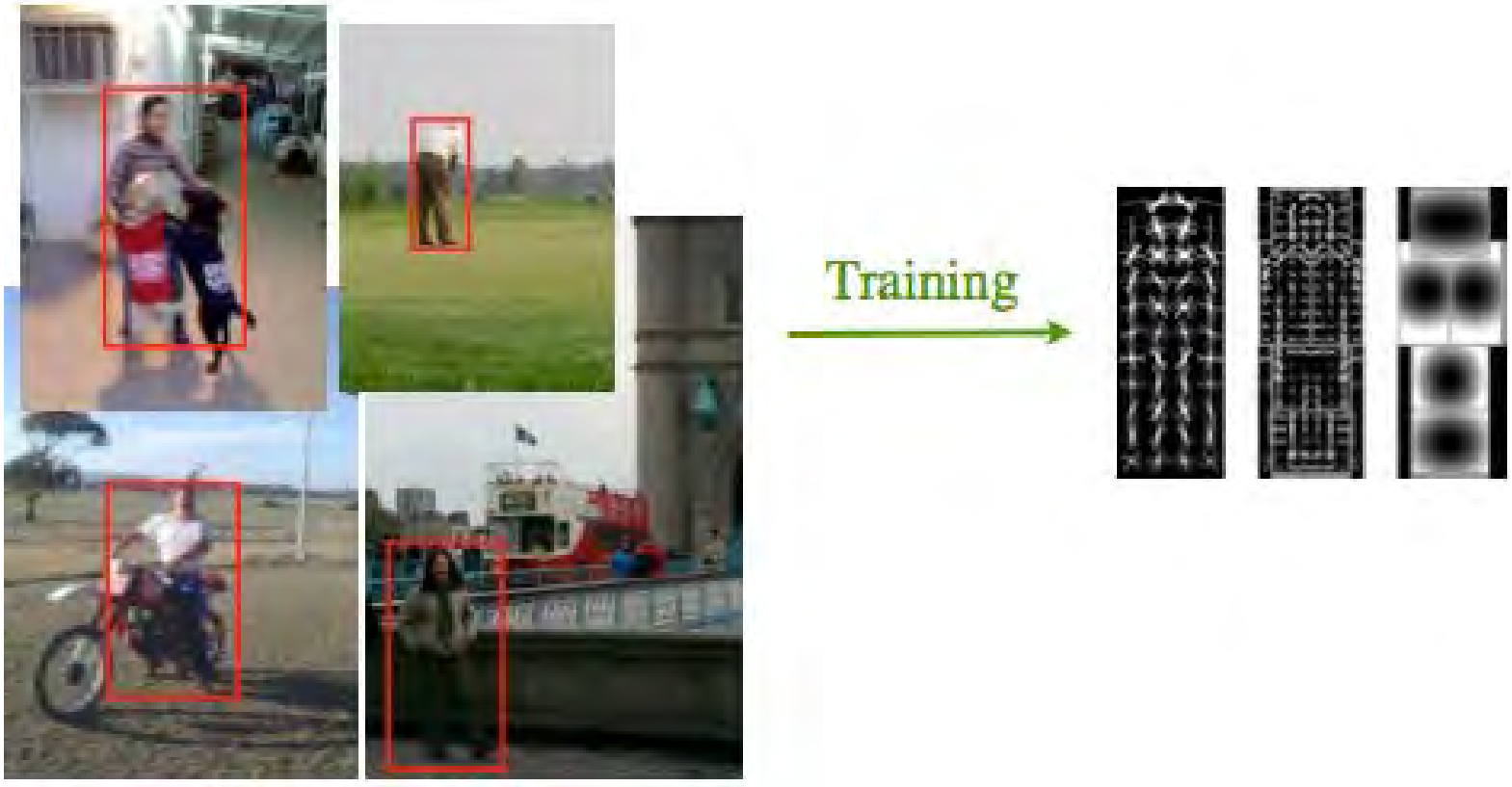
Matching results





- (after non-maximum suppression) ~1 second to search all scales

Training



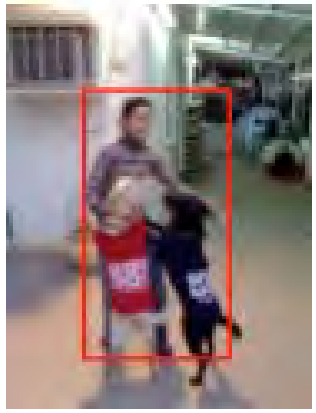
- Training data consists of images with labeled bounding boxes.
- Need to learn the model structure, filters and deformation costs.

Latent 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

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

Training Models

- Reduce to Latent SVM training problem
- Positive example specifies some *z* should have high score
- Bounding box defines range of root locations
  - Parts can be anywhere
  - This defines *Z(x)*

Background

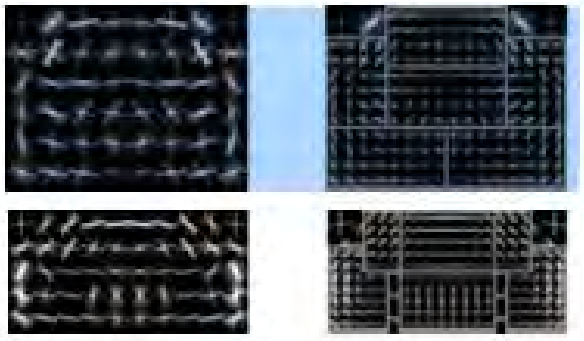
- Negative example specifies no z should have high score

- One negative example per root location in a background image
  - Huge number of negative examples
  - Consistent with requiring low false-positive rate

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
  - Throw away (x,z) pairs with low score

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



Data-mining with hard examples

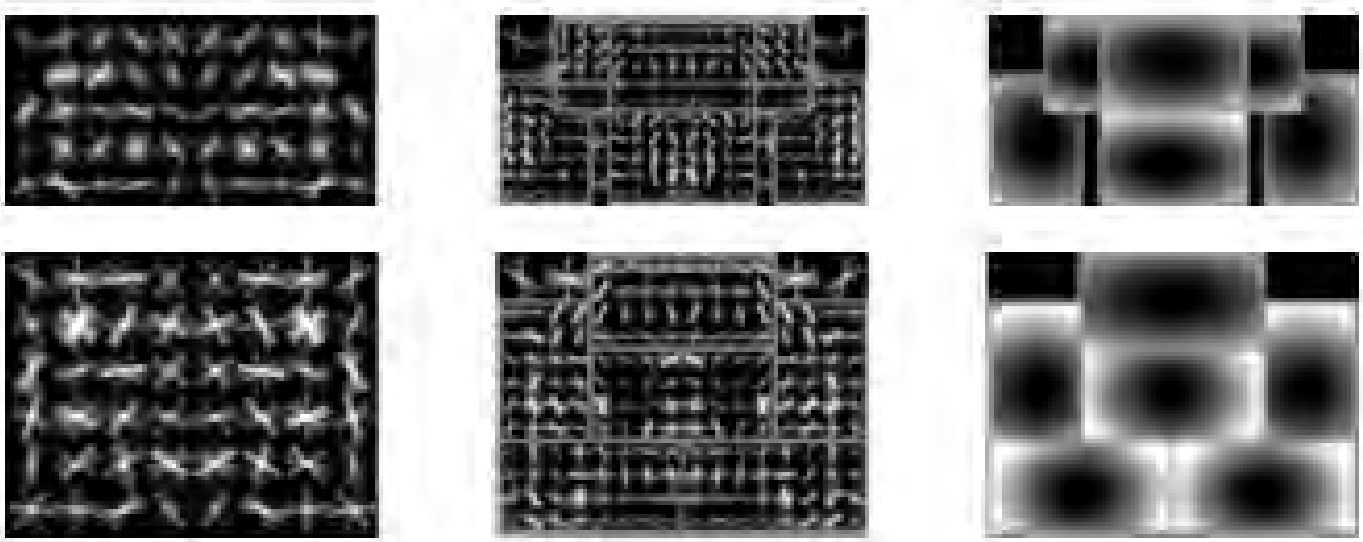
Let us consider:

- Hard cases: $H(\beta,D) = \{\langle x,y\rangle \in D \mid yf_\beta(x) < 1\}$.
- Easy cases: $E(\beta,D) = \{\langle x,y\rangle \in D \mid yf_\beta(x) > 1\}$.

Let $C_1 \subseteq D$ be an initial cache of examples. The algorithm repeatedly trains a model and updates the cache as follows:

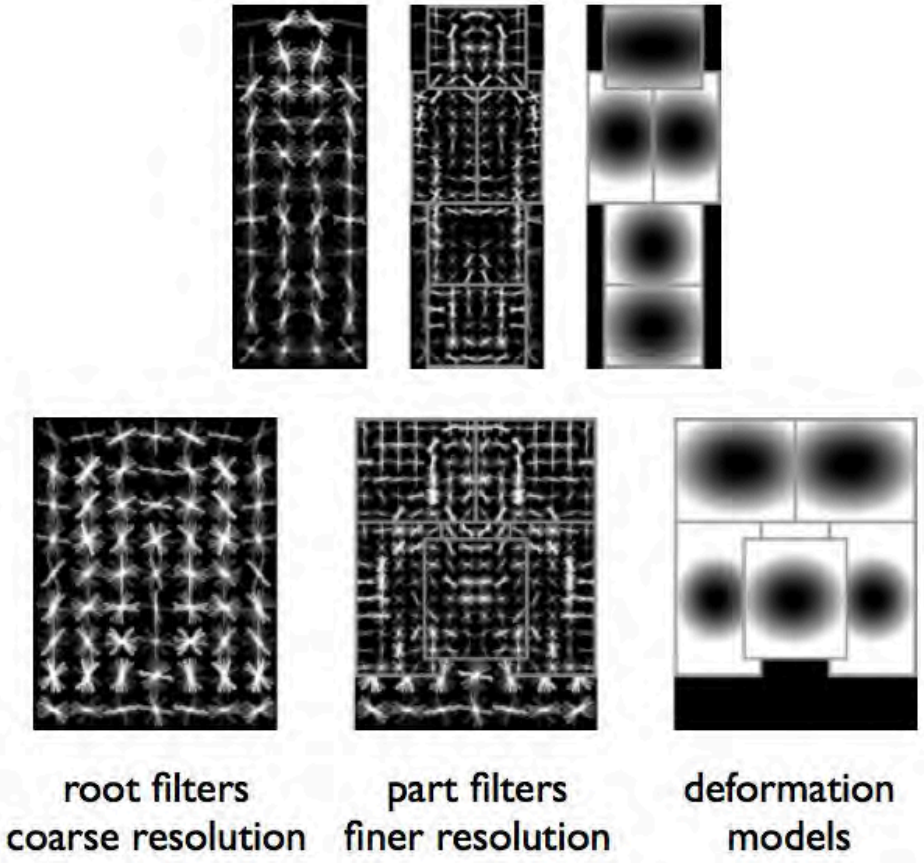
- 1) Let $\beta_t := \beta*(C_t)$ (train a model using C_t).
- 2) If $H(\beta_t, D) \subseteq C_t$ stop and return β_t.
- 3) Let $C_t' := C_t \backslash X$ for any X such that $X \subseteq E(\beta_t, C_t)$ (shrink the cache).
- 4) Let $C_{t+1} := C_t' \cup X$ for any X such that $X \subseteq D$ and $X \cap H(\beta_t,D) \backslash Ct \neq$ empty set (grow the cache).

Car model

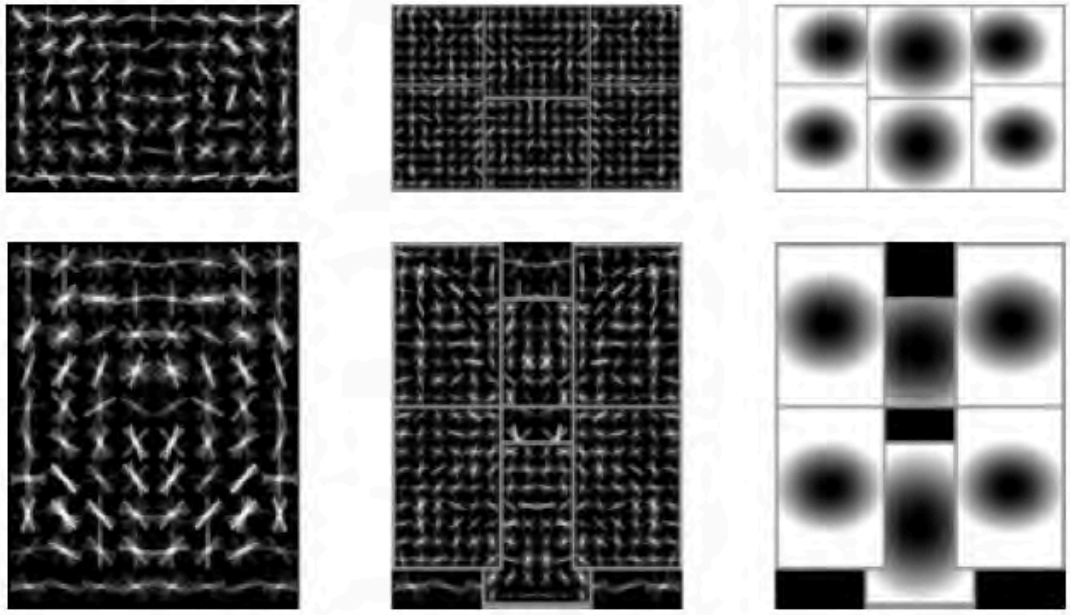


Two components trained per object.

Person model



Cat model

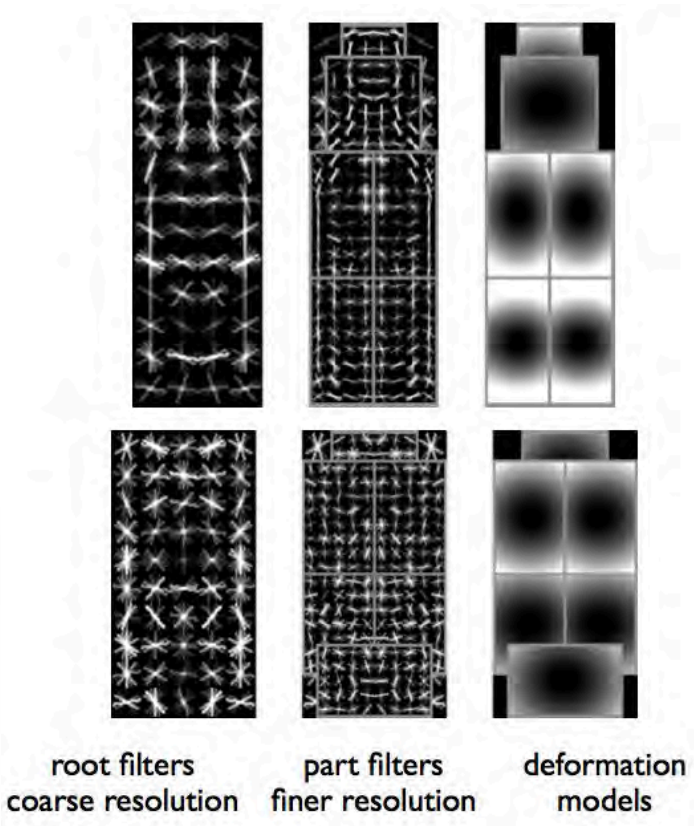


root filters
coarse resolution

part filters
finer resolution

deformation
models

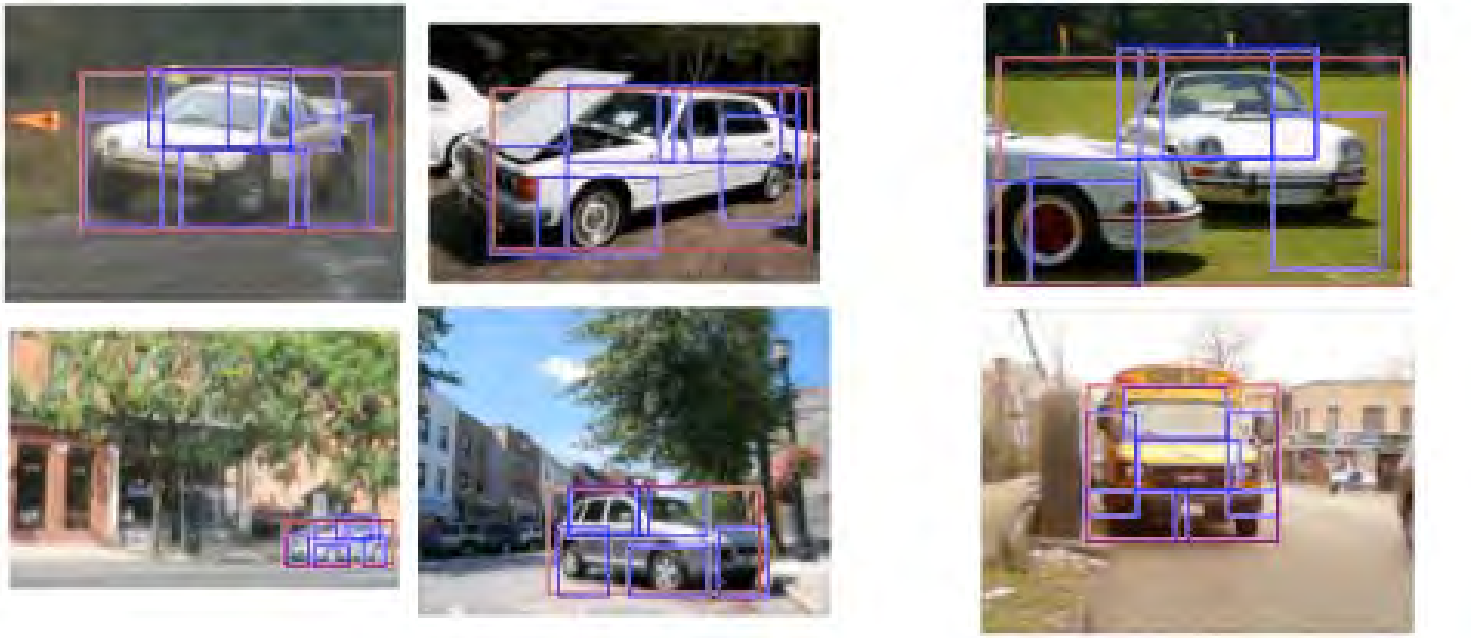
Bottle model



Car detections

high scoring true positives

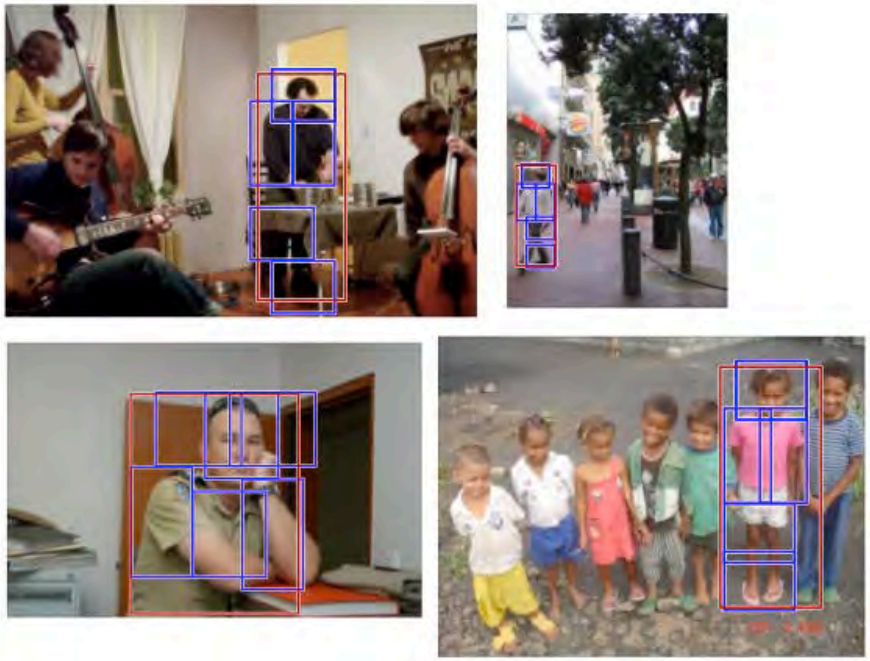
high scoring false positives



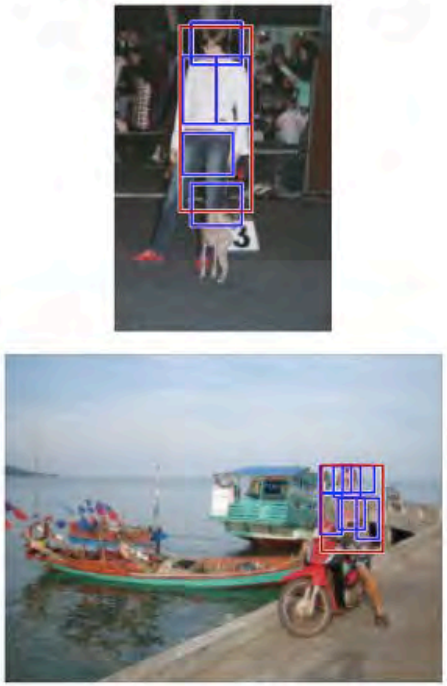
Correct if bounding boxes overlap more than 50%.
Confusions used to be btw cars and buses.

Person detections

high scoring true positives

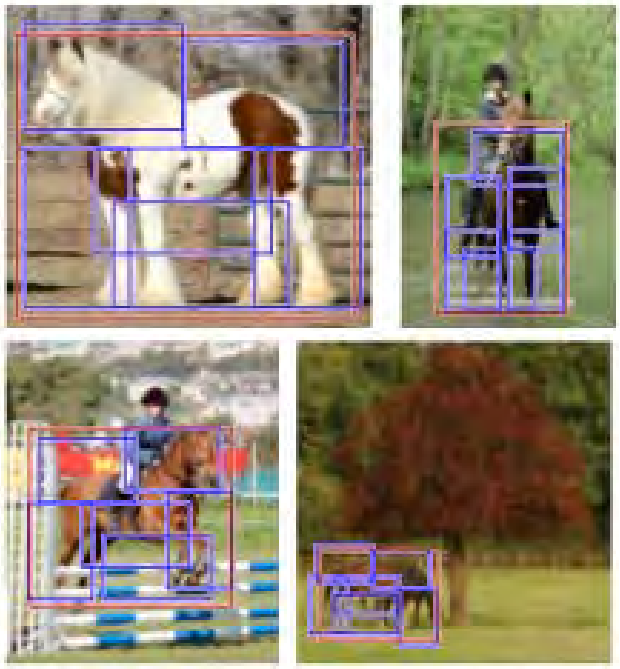


high scoring false positives
(not enough overlap)

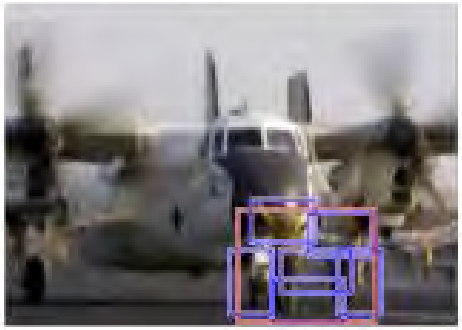


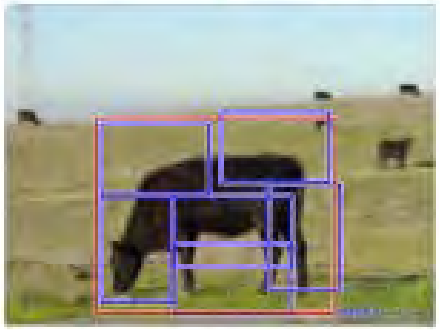
Horse detections

high scoring true positives



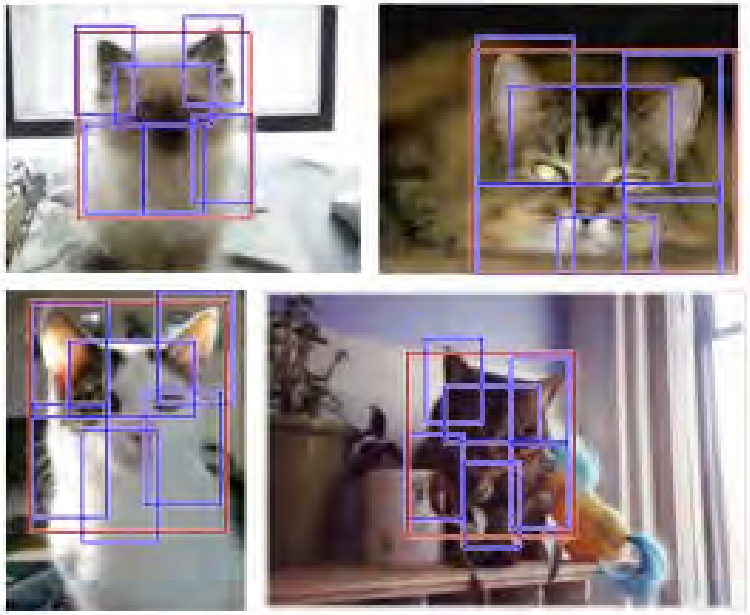
high scoring false positives



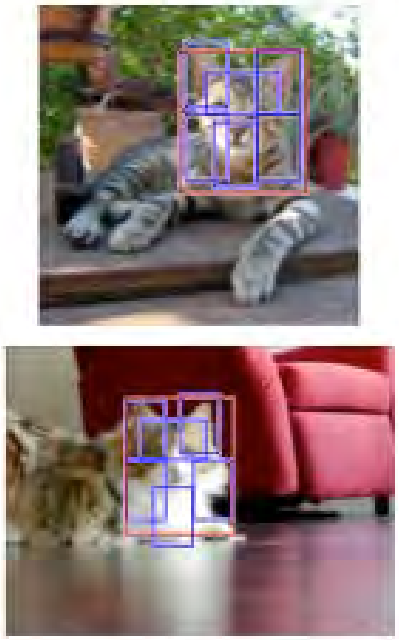


Cat detections

high scoring true positives



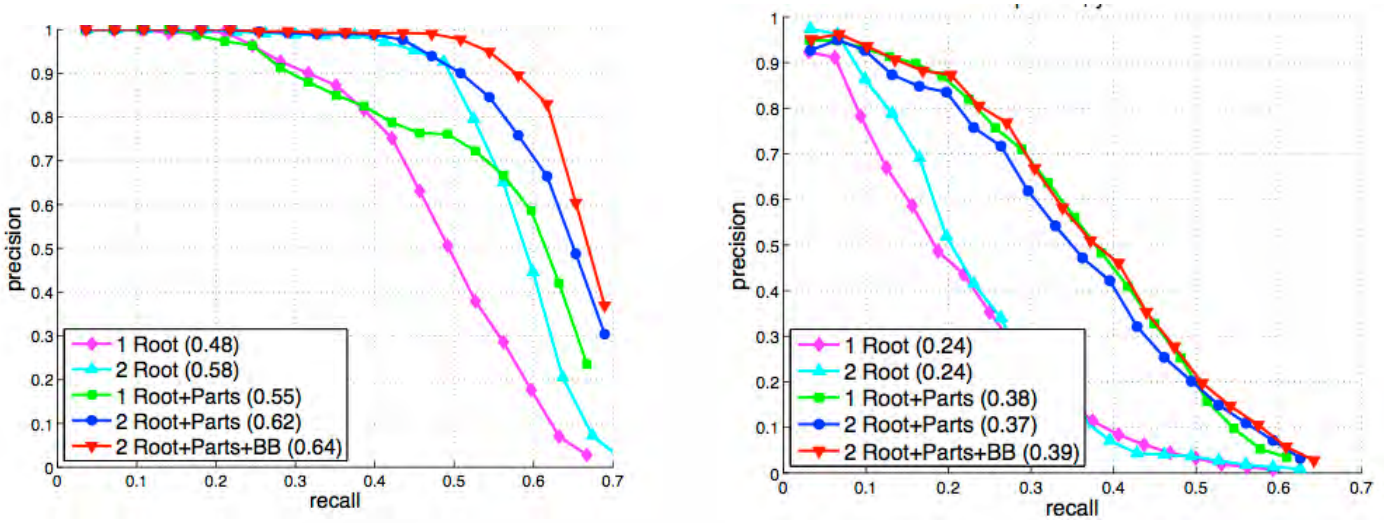
high scoring false positives
(not enough overlap)



Quantitative results

- 7 systems competed on the Pascal challenge
- In out of 20 classes they 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 Cars and Persons



Precision/Recall curves for models trained on the person and car categories of the PASCAL 2006 dataset. We show results for 1 and 2 component models with and without parts, and a 2 component model with parts and bounding box prediction. In parenthesis we show the average precision score for each model.

Precision = TP/(TP+FP) Recall=TP/(TP+FN)

Mixture models are more important for cars than persons.

Comparison to the state-of-the-art

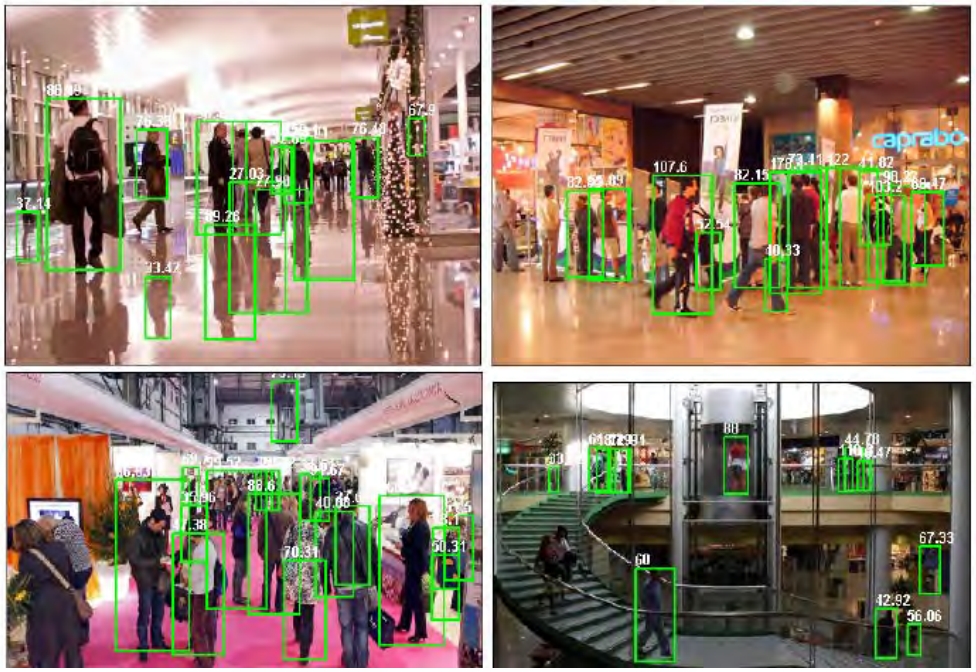
	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbik	pers	plant	sheep	sofa	train	tv
a) base	.336	.371	.066	.099	.267	.229	.319	.143	.149	.124	.119	.064	.321	.353	.407	.107	.157	.136	.228	.324
b) BB	.339	.381	.067	.099	.278	.229	.331	.146	.153	.119	.124	.066	.322	.366	.423	.108	.157	.139	.234	.328
c) context	.351	.402	.117	.114	.284	.251	.334	.188	.166	.114	.087	.078	.347	.395	.431	.117	.181	.166	.256	.347
d) rank	2	1	1	1	1	1	2	2	1	2	4	5	2	2	1	1	2	2	3	1
(UofCTTIUCI)	.326	.420	.113	.110	.282	.232	.320	.179	.146	.111	.066	.102	.327	.386	.420	.126	.161	.136	.244	.371
CASIA Det	.252	.146	.098	.105	.063	.232	.176	.090	.096	.100	.130	.055	.140	.241	.112	.030	.028	.030	.282	.146
Jena	.048	.014	.003	.002	.001	.010	.013		.001	.047	.004	.019	.003	.031	.020	.003	.004	.022	.064	.137
LEAR PC	.365	.343	.107	.114	.221	.238	.366	.166	.111	.177	.151	.090	.361	.403	.197	.115	.194	.173	.296	.340
MPI struct	.259	.080	.101	.056	.001	.113	.106	.213	.003	.045	.101	.149	.166	.200	.025	.002	.093	.123	.236	.015
Oxford	.333	.246					.291			.125			.325	.349						
XRCE Det	.264	.105	.014	.045	.000	.108	.040	.076	.020	.018	.045	.105	.118	.136	.090	.015	.061	.018	.073	.068

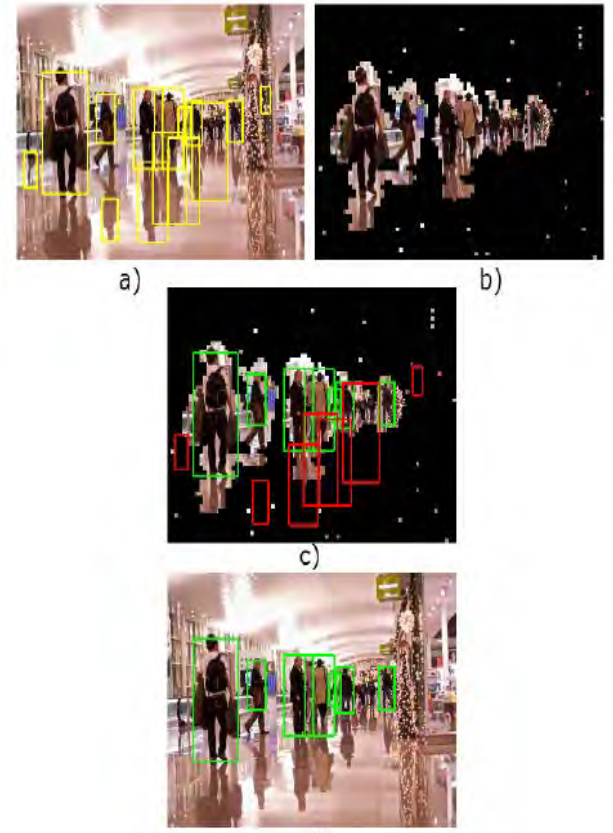
PASCAL VOC results. Top: (a) average precision scores of the base system, (b) scores using bounding box prediction, (c) scores using bounding box prediction and context rescoring, (d) ranking of final scores relative to systems in the competition. Bottom: the systems that participated in the competition.

Application

- To build home-made data-set of RGBD images, using Kinect
- Use of Depth information + RGB to obtain segmentation

Background modeling





Left: Person detection. Right: a) People Detection, b) Background Subtraction, c) Intersection between detections and foreground, d) Final result

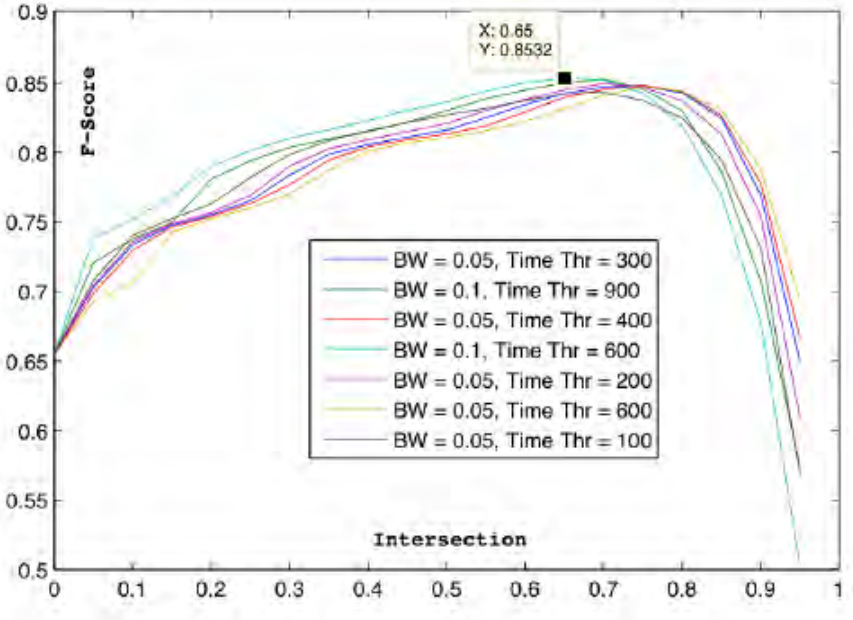
Background modeling

Algorithm 1 New Frame Computation

1: **for** Every new frame **do**
2: —Run FPDW to get *detections*
3: —Reject *detections* which have a bounding box height smaller than h specified minimum
4: —Compute HoG in Cells
5: —Add frame to *history* (set of frames determined by the time-window)
6: —Compute *distance* of current frame cells from *BG* clusters
7: —Threshold *distance* greater than a threshold from any *BG* cluster and define current frame background $cfBG$
8: —Compute *intersection* of $cfBG$ and *detections*
9: —Reject *detections* with *interesection* less than $minInt$ percentage
10: **end for**

Algorithm 2 Background Update

1: **for** all cells in *history* **do**
2: —Compute Mean-Shift *clusters* with *bandwidth* parameter
3: —Threshold *clusters* with number of members less than $timeThr$
4: **end for**
5: **Output** *BG* model from *clusters*



Results: precision of 75%, a recall of 99% and F-Score of 85%.


27
134
40
0.078502
0.1920
0.12305
0.13258


64
70
42
0.13454
0.14579
0.11984


57
52
50
0.15686
0.16522
0.10278


53
72
36
0.22447
7171
0.24069

Validation

- To ensure whether the proposed methodology is satisfying!
- To retrieve pose detection accuracy by calculating precision and recall of detection output.
  - Precision: fraction of retrieved instances that are relevant
  - Recall: fraction of relevant instances that are retrieved
    - Difficult to have high precision and high recall at the same time!
  - F-1 Score:
    - a single value obtained combining both the precision and recall measures
    - Indicates overall utility of the system

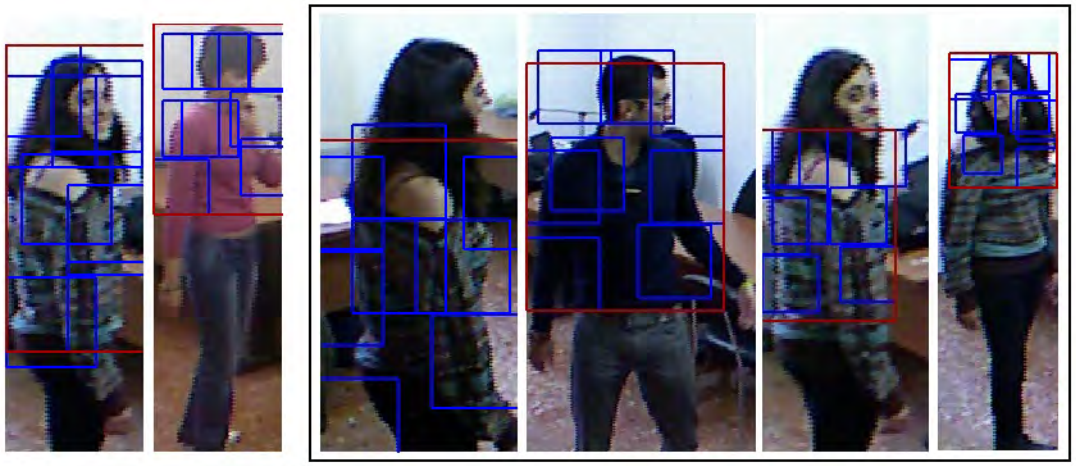
Data

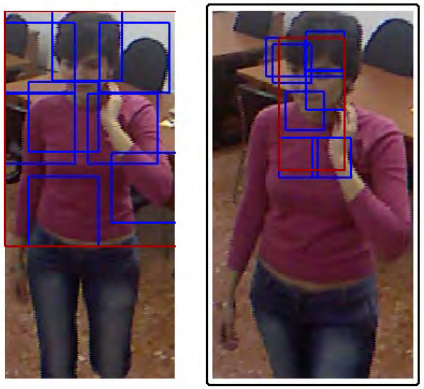
- Home-made test set using Kinect camera
- Four captured videos, each of which contains hundreds of frames
- Videos are captured inside an office
- Kinect camera located at height of ~2.5 meters above the floor and it is rotated toward the floor
- The acquired test set is a difficult set

Evaluation

- Goal: To correctly assign the people's body to a specific class (pose) separately in each frame (if any).
- We have trained a two component model for each class (pose).
- Output: A bounding box and numerical output of matching process (matched models with their corresponding score).
  - We are interested in finding highest matching score.

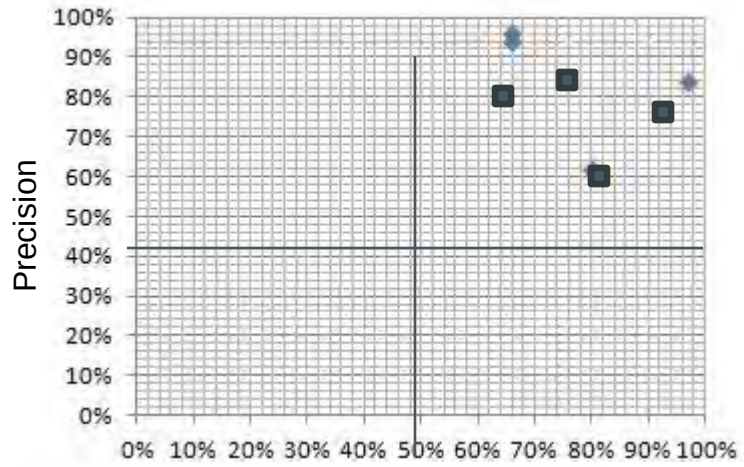
Result



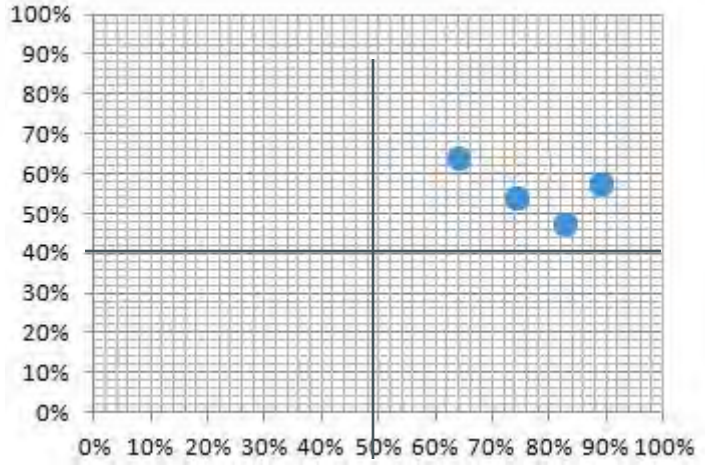


Result

	# Frames	# People	Precision (profile)	Recall (profile)	Accuracy (profile)	Precision (straight)	Recall (straight)	Accuracy (straight)
Video #1	625	2	**94%**	**66%**	**78%**	**90%**	**66%**	**81%**
Video #2	126	1	**84%**	**97%**	**90%**	**86%**	**92%**	**89%**
Video #3	1239	2	**62%**	**80%**	**70%**	**69%**	**83%**	**75%**
Video #4	557	3	**96%**	**66%**	**78%**	**97%**	**76%**	**87%**



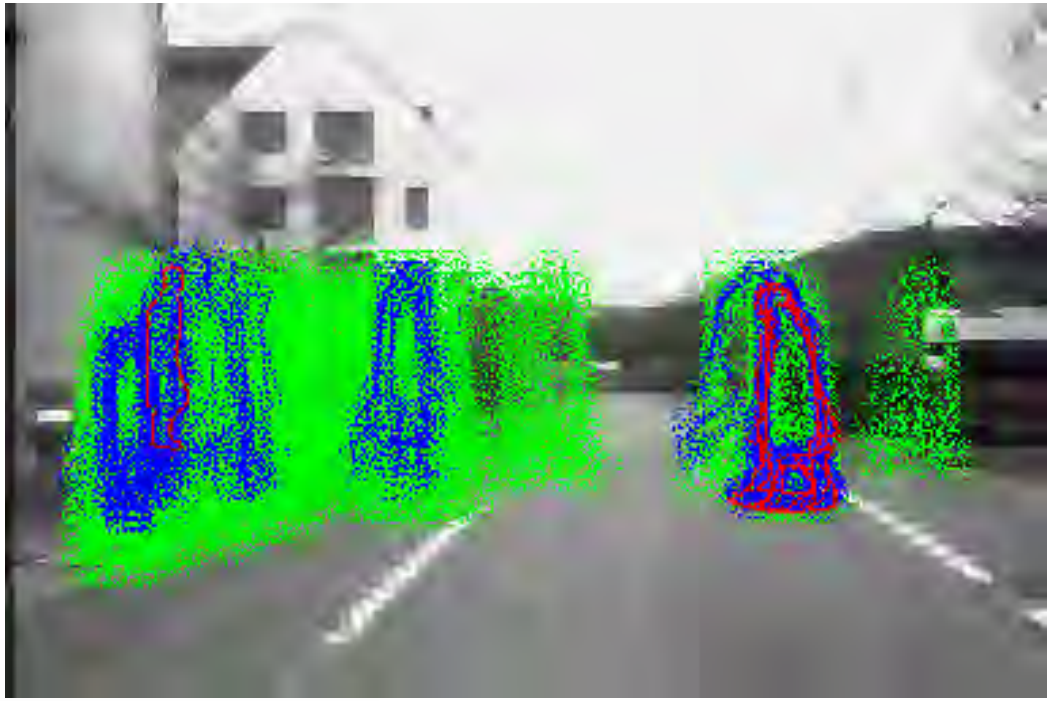
~ 1 second per frame



Without Segmentation

~ 1 minute per frame

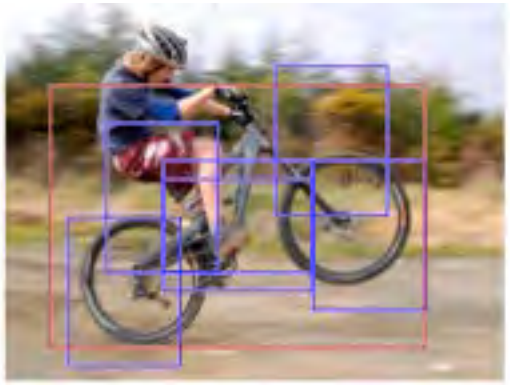
Applications



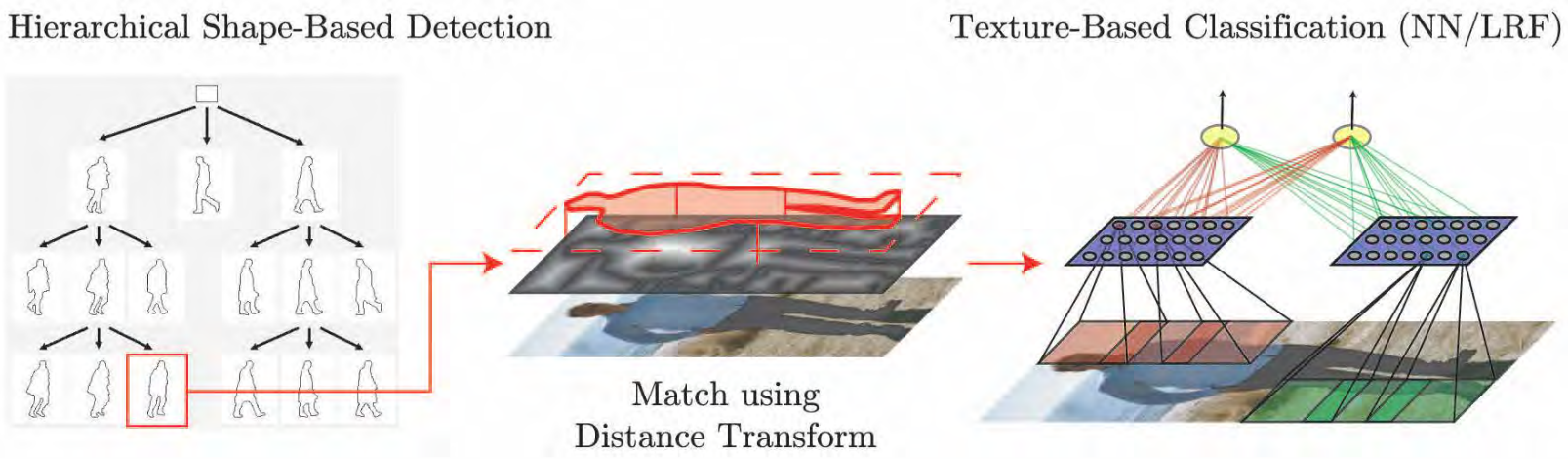
- Volvo S60
- Mobileye

Conclusion & Future Works

- Deformable models for object detection
  - Fast matching algorithms
  - Learning from weakly-labeled data
  - Overperform state-of-the-art results in PASCAL challenge
  - GMM based segmentation
  - Background modeling
  - Next steps
- Future work:
  - Hierarchical models
  - Visual grammars
  - To predict the people's movement trajectory



Future work: Hybrid Methods



- Combine different approaches
  - Silhouette information
  - Appearance information
  - Motion