Reconnaissance d'objets et vision artificielle 2011

Category-level localization and human pose estimation

Ivan Laptev

Slides from Andrew Zisserman
and Deva Ramanan

Also includes slides from: Ondra Chum, Alyosha Efros, Mark Everingham, Pedro Felzenszwalb, Rob Fergus, Kristen Grauman, Bastian Leibe, Fei-Fei Li, Marcin Marszalek, Pietro Perona, Bernt Schiele, Jamie Shotton, Andrea Vedaldi

Announcements

- Assignment 3 is due to next week

http://www.di.ens.fr/willow/teaching/recvis11/assignment3/

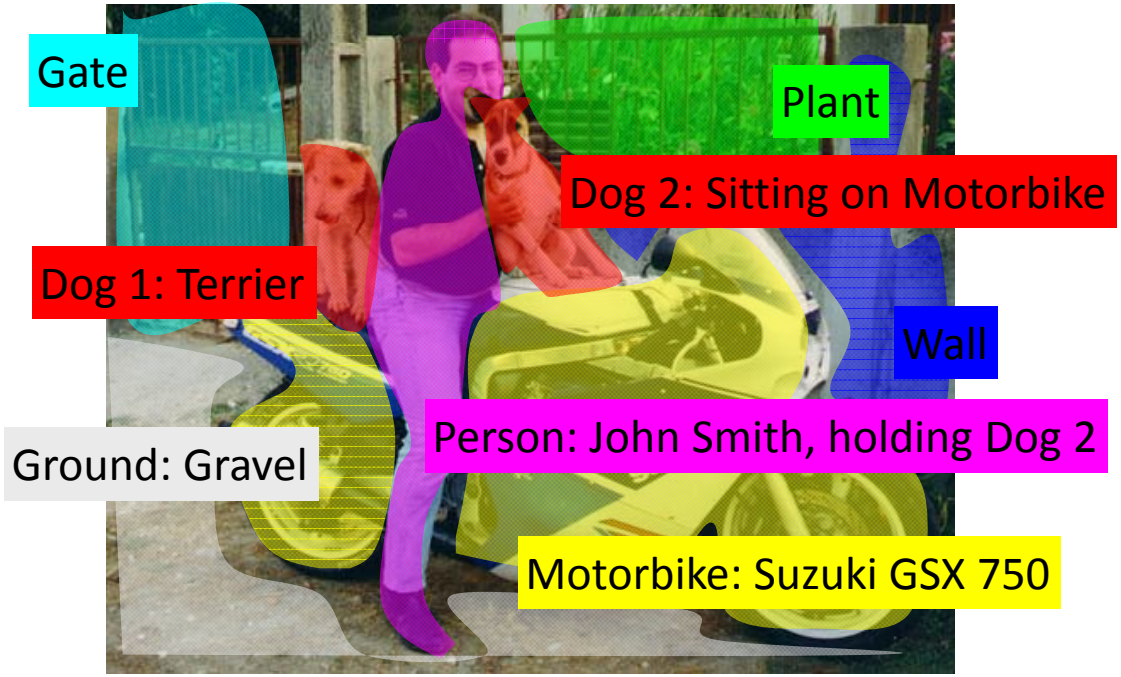
- Final project proposals were due to last week.
  http://www.di.ens.fr/willow/teaching/recvis11/finalproject/

- See what reports we have received from you here:
  https://docs.google.com/spreadsheet/pub?key=0Aso5oi2c4UB5dGVXXzFIRWZoZ24wNzNuQll5c3FsNXc&output=html

Key:	*R=received, L=late (<3days), VL=very late (>3days)*						
Student name	**Email**	**Assignment 1**	**Assignment 2**	**Assignment 3**	**FP proposal**	**FP presentation**	**FP report**
BARRAU Axel	axel.barrau@hotmail.fr	R	R		In Discussion		
BOJANOWSKI Piotr	piotr.bojanowski@m4x.org	R	R		In Discussion		
BOUGHIDA Malik	malikboughida@yahoo.fr	R	R		R		
CADRAN Jerome	jerome.cadran@student.ecp.fr	R	R				
CARLINET Edwin	edwin.carlinet@gmail.com	R	R		R		
CHATELAIN Pierre	pierre.chatelain@ens-cachan.fr	R	R		R		
CHESNEAU Nicolas	nchesneau@gmail.com	R	R		R		
CONEJO Bruno	bconejo@gmail.com	R	R		R		
ERGUN Hilal	hilalergun88@hotmail.com	R	VL		R		
GOMEZ-FERNANDEZ Francisco	fgomezf@gmail.com	R	R		R		
KANTOROV Vadim	vadimkantorov@gmail.com	L			R		
LAURENT Antoine	antoine.laurent@polytechnique.org	R	R		R		
LEFEBVRE de LABOULAYE Gabriel	gabriel.lefebvre-de-laboulaye@student.ecp.fr	R	R		R	Dec 9	
LEGRAND Diego	diego.legrand@student.ecp.fr	R	R		R		
LIU Zhe	liu.zhe.imagine@gmail.com	R			No project - PhD	No project - PhD	No project - PhD
MOULLET Simon	simon.moullet@gmail.com	R	R		In Discussion		
NADAL Pierre-Adrien	panadal@gmail.com	R	R		In Discussion		
POULENARD Raphael	raphael.poulenard@free.fr	(R)			In Discussion		
RAIS Martin	martus@gmail.com	R	R		R		
ROYER Martin	martin.royer@m4x.org	VL	R		R		
SANCHEZ-PEREZ Andres	andres.sanchez-perez@polytechnique.edu	R	R		R		
SCAMAN Kevin	kevin.scaman@polytechnique.edu	R	R		R		
SEICHEPINE Nicolas	nicolas.seichepine@eleves.enpc.fr	R	R		R		
SUZANO-MASSA Francisco Vitor	francisco-vitor.suzano-massa@polytechnique.edu	R	R	R	R		
THOMAS Francois-Xavier	fx.thomas@gmail.com	R	R		In Discussion		
VAROQUAUX Nelle	nelle.varoquaux@gmail.com	R	R		R		
ZHANG Shun	tobbyzh@gmail.com	VL	R				

What we would like to be able to do…

- Visual scene understanding
- **What** is in the image and **where**



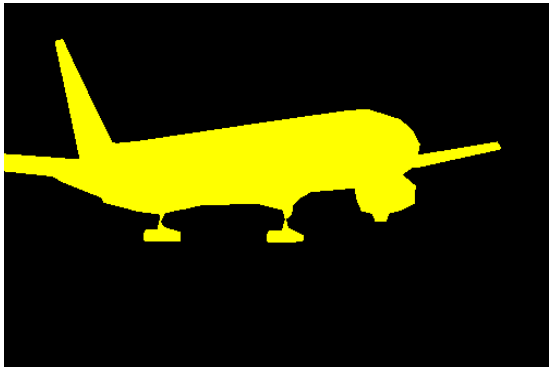
- Object categories, identities, properties, activities, relations, …

Recognition Tasks

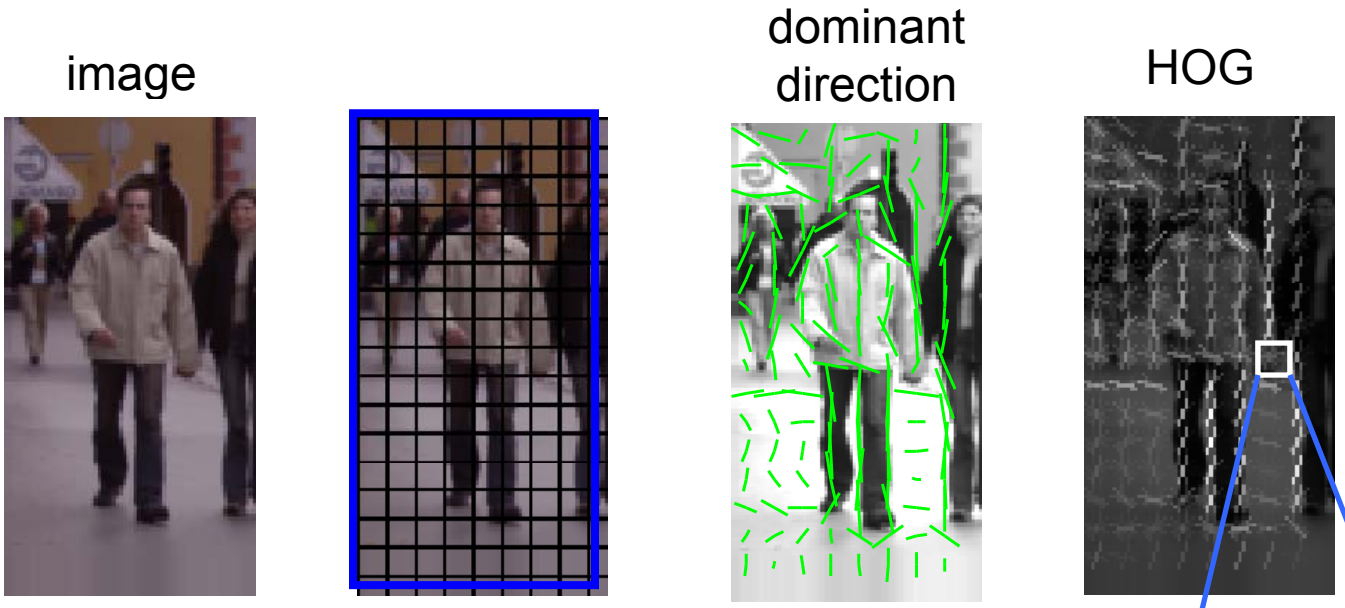
- **Image Classification**
  - Does the image contain an aeroplane?

- **Object Class Detection/Localization**
  - Where are the aeroplanes (if any)?

- **Object Class Segmentation**
  - Which pixels are part of an aeroplane (if any)?

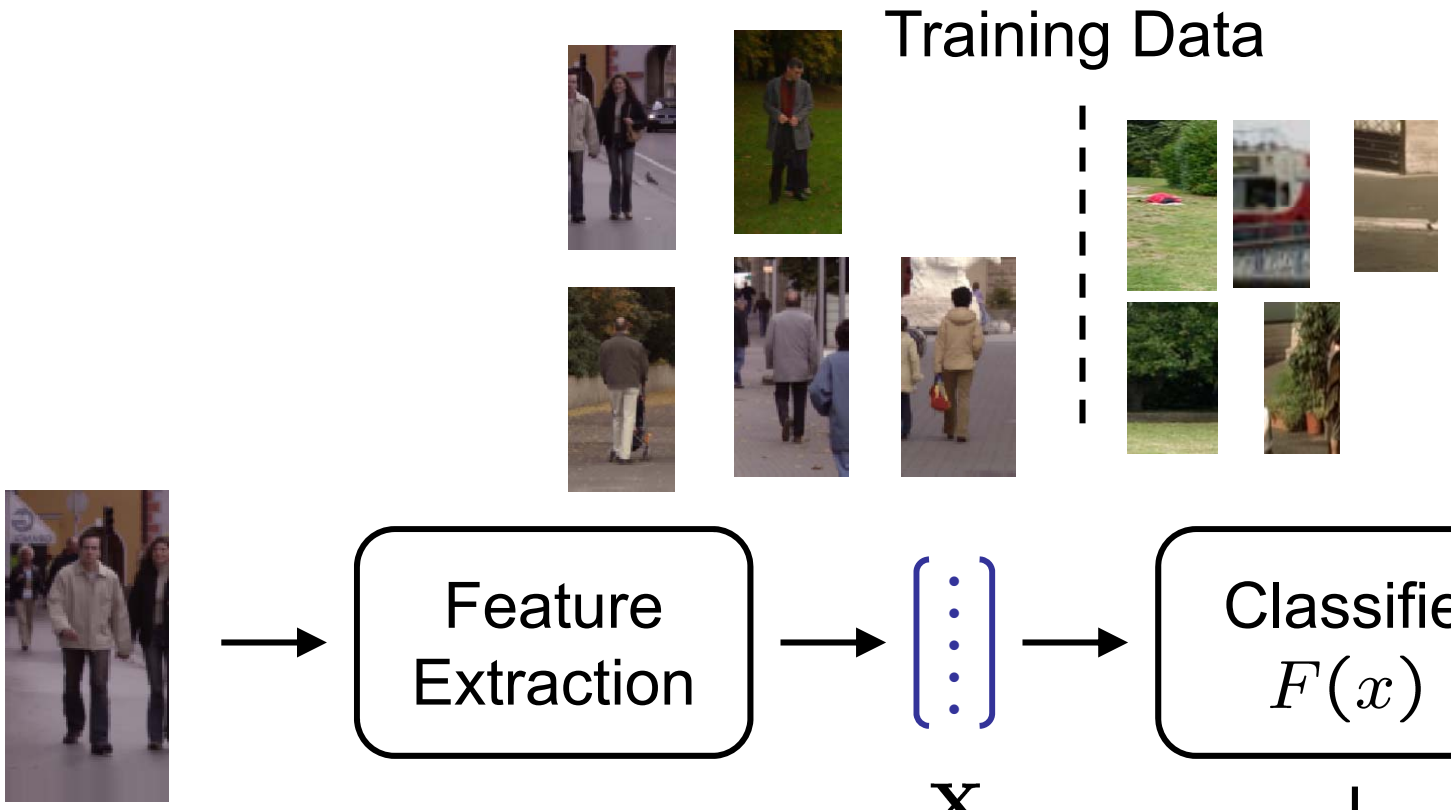


Feature: Histogram of Oriented Gradients (HOG)



- tile 64 x 128 pixel window into 8 x 8 pixel cells
- each cell represented by histogram over 8 orientation bins (i.e. angles in range 0-180 degrees)

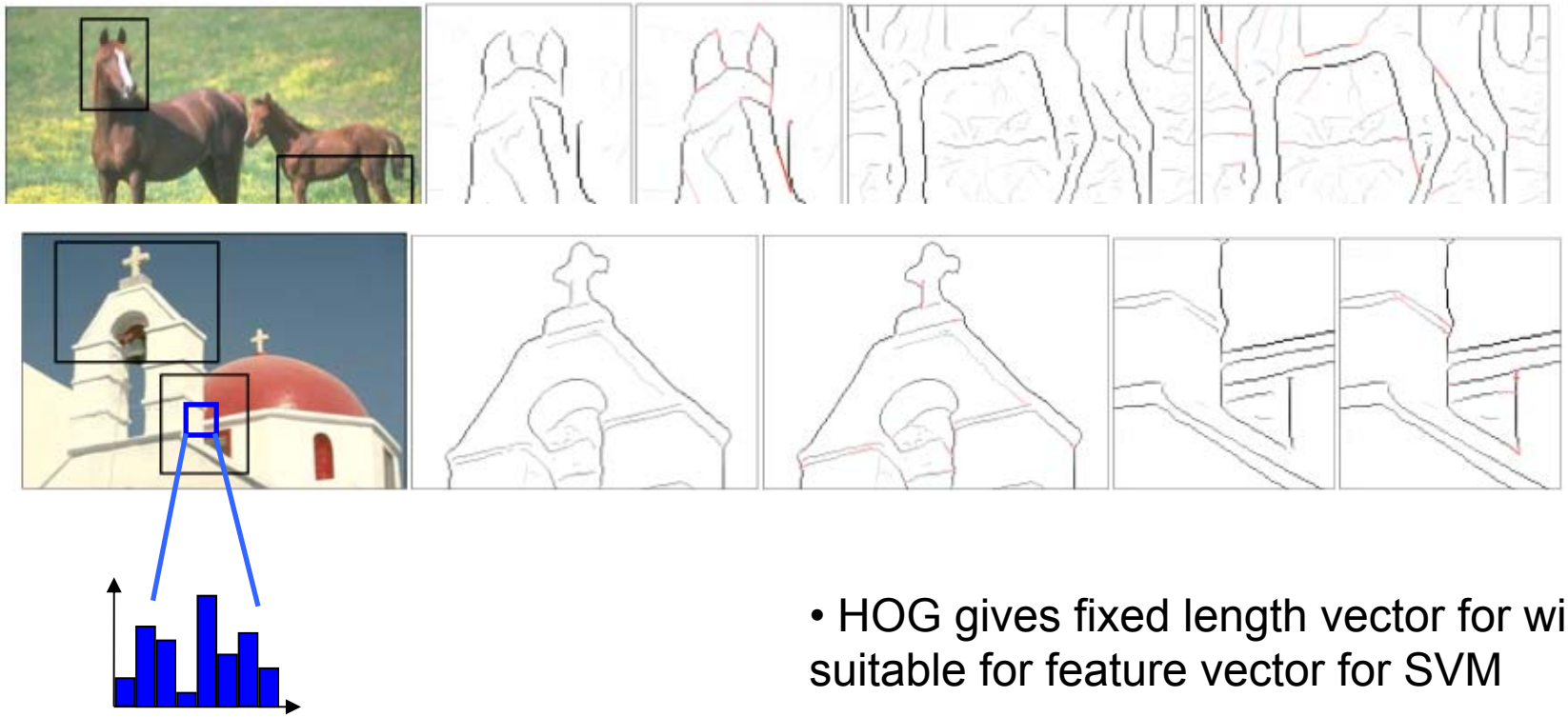
Window (Image) Classification



- HOG Features
- Linear SVM classifier

Why does HOG + SVM work so well?

- Similar to SIFT, records spatial arrangement of histogram orientations
- Compare to learning only edges:
  - Complex junctions can be represented
  - Avoids problem of early thresholding
  - Represents also soft internal gradients
- Older methods based on edges have become largely obsolete

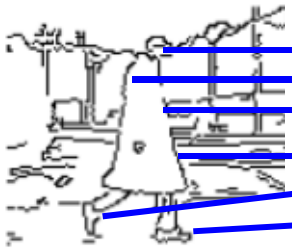


- HOG gives fixed length vector for window, suitable for feature vector for SVM

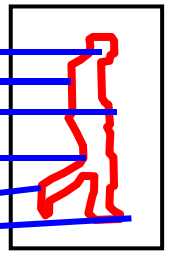
Chamfer Matching

Input

Edges



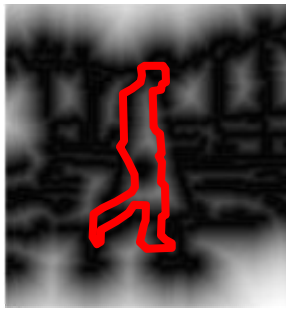
Template



- Match points between template and image
- Measure mean distance
- Template edgel matches **nearest** image edgel

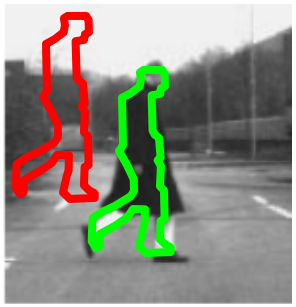
$$D(T, I) = \frac{1}{|T|} \sum_{\mathbf{p} \in T} \min_{\mathbf{q} \in I} d(\mathbf{p}, \mathbf{q})$$

Distance Transform



- Distance transform reduces min operation to array lookup
- Computable in linear time

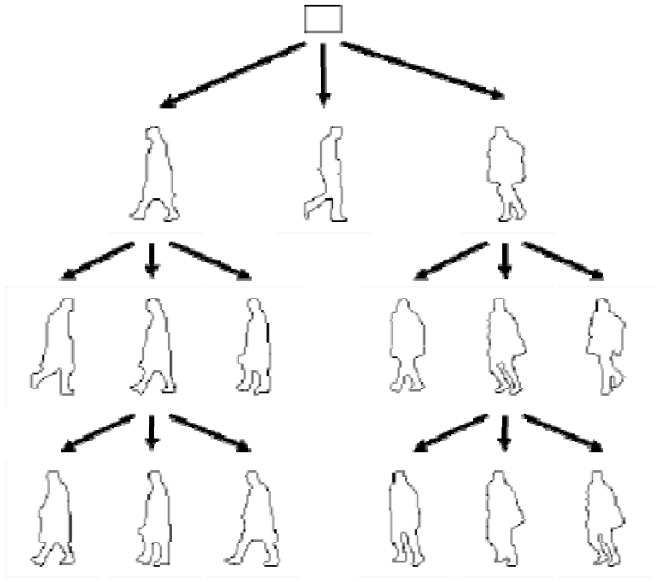
Best match



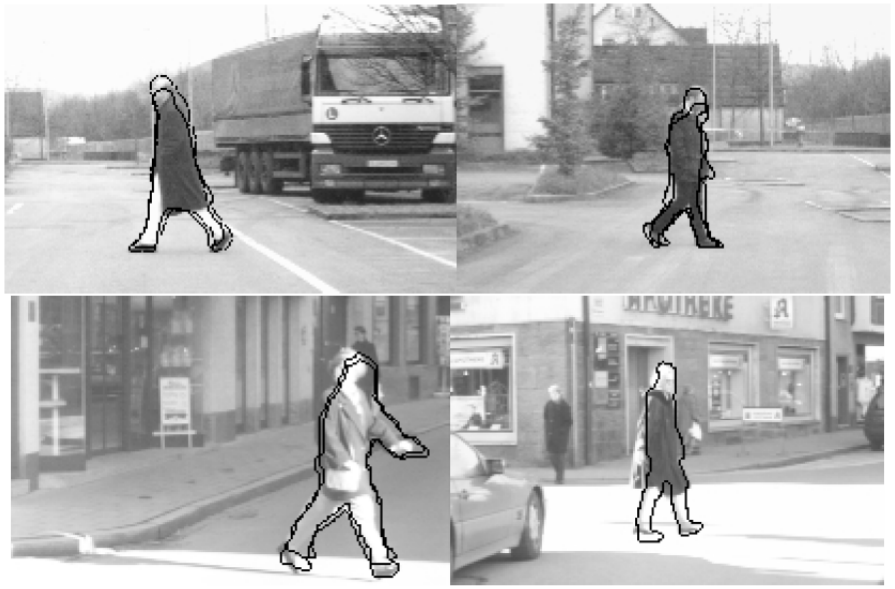
- Localize by sliding window search

[Gavrila & Philomin, 1999]

Chamfer Matching



Hierarchy of Templates



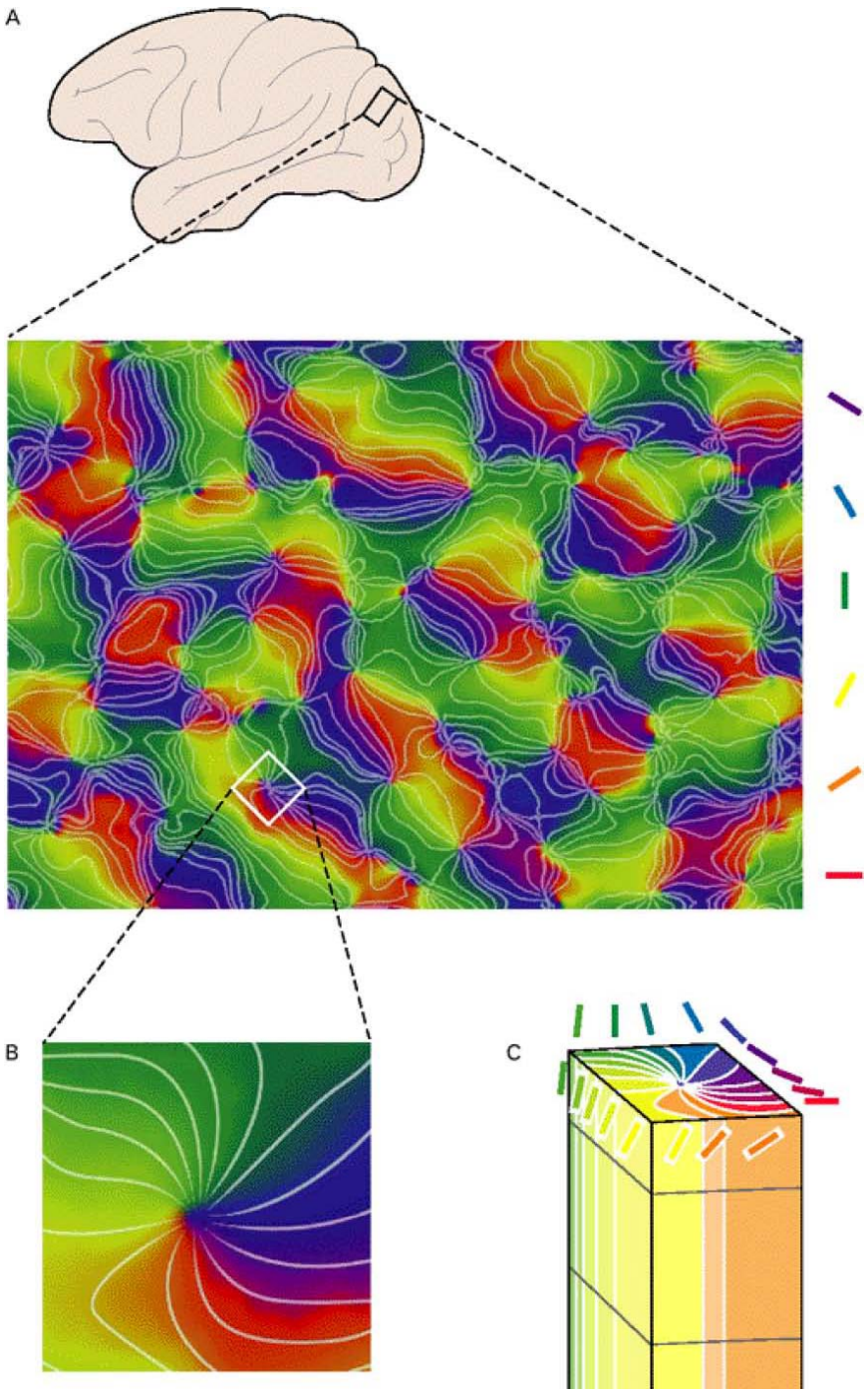
Detections

- In practice performs poorly in clutter
- Unoriented edges are not discriminative enough (too easy to find…)

[Gavrila & Philomin, 1999]

Biologic perspective

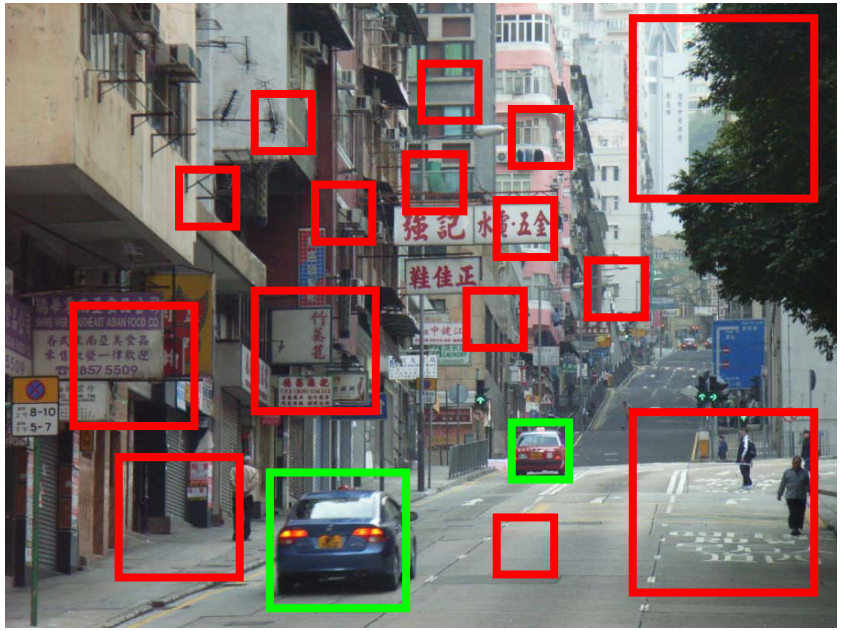
- The function of the brain mostly remains unknown, however, the structure of the primary visual cortex (also known as V1) is quite well understood.
- V1 is organized in orientation- and location-sensitive “columns”.
- Why sampling spatial orientation? Edge directions can be inferred from only two measurements (dI/dx, dI/dy).
- Averaging responses for particular orientation in a local spatial neighborhood would be similar to a particular bin of the HOG/SIFT histogram vectors!



Orientation columns in the visual cortex of the monkey. (Illustration from Gary Blasdel.)

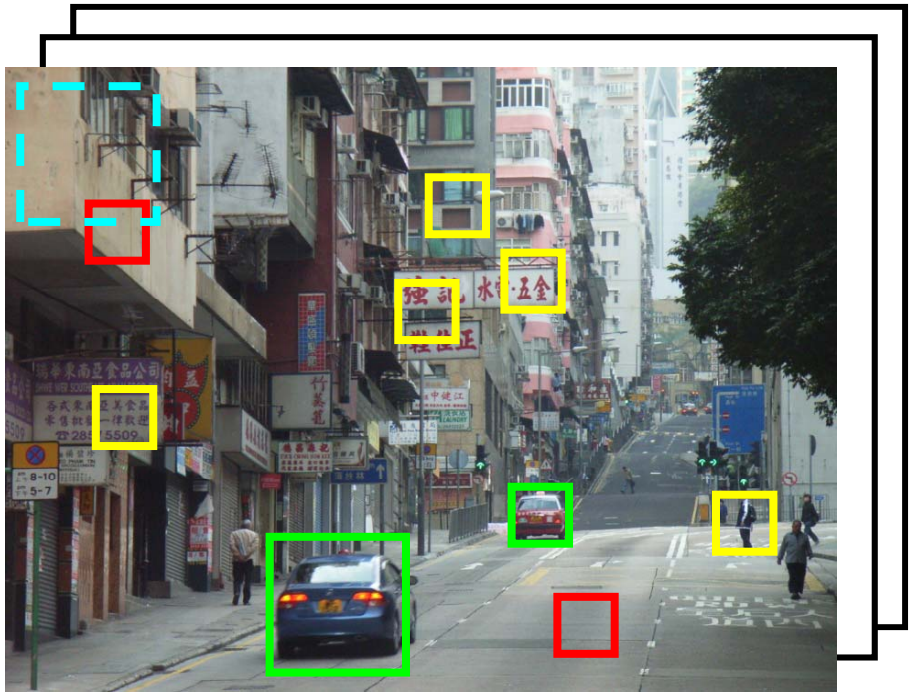
Training a sliding window detector

- Object **detection** is inherently asymmetric: much more "non-object" than "object" data



- Classifier needs to have very low false positive rate
- Non-object category is very complex – need lots of data

Bootstrapping



1. Pick negative training set at random
2. Train classifier
3. Run on training data
4. Add false positives to training set
5. Repeat from 2

- Collect a finite but diverse set of non-object windows
- Force classifier to concentrate on **hard negative** examples

- For some classifiers can ensure equivalence to training on entire data set

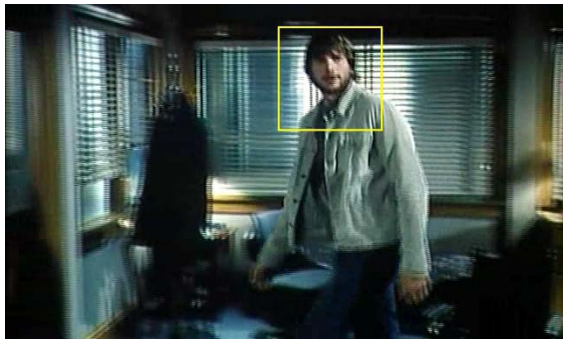
Example: train an upper body detector

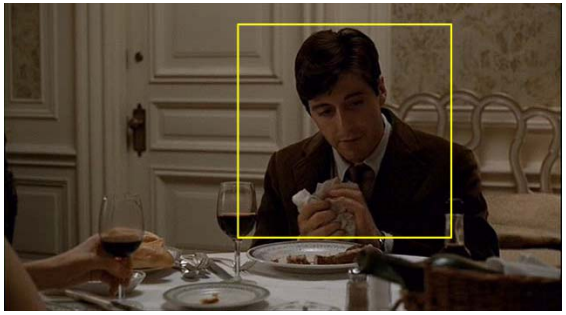
- Training data – used for training and validation sets
  - 33 Hollywood2 training movies
  - 1122 frames with upper bodies marked

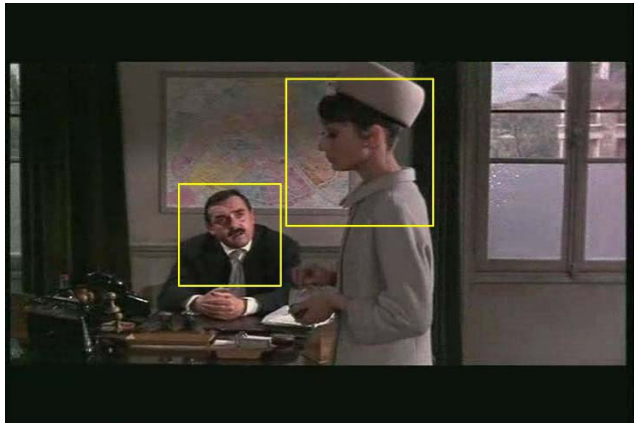
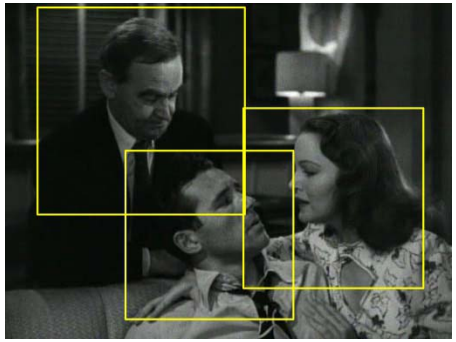
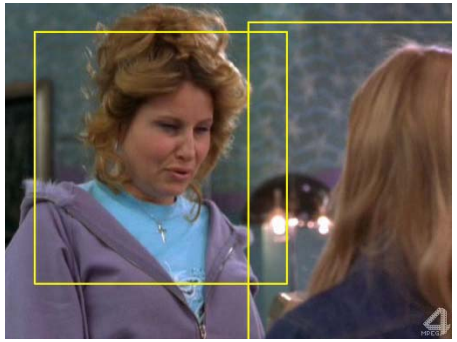
- First stage training (bootstrapping)
  - 1607 upper body annotations jittered to 32k positive samples
  - 55k negatives sampled from the same set of frames

- Second stage training (retraining)
  - 150k hard negatives found in the training data

Training data – positive annotations

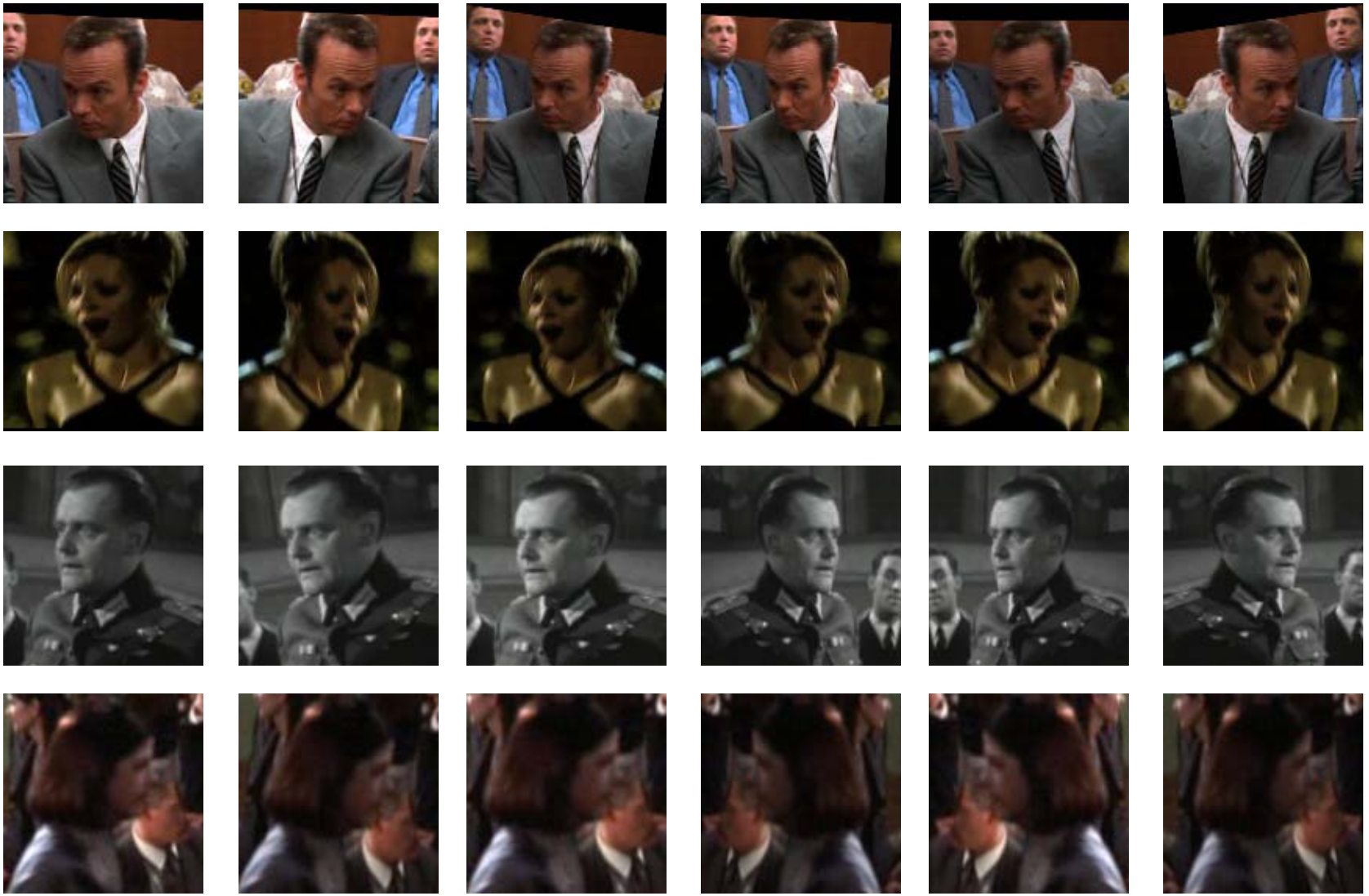



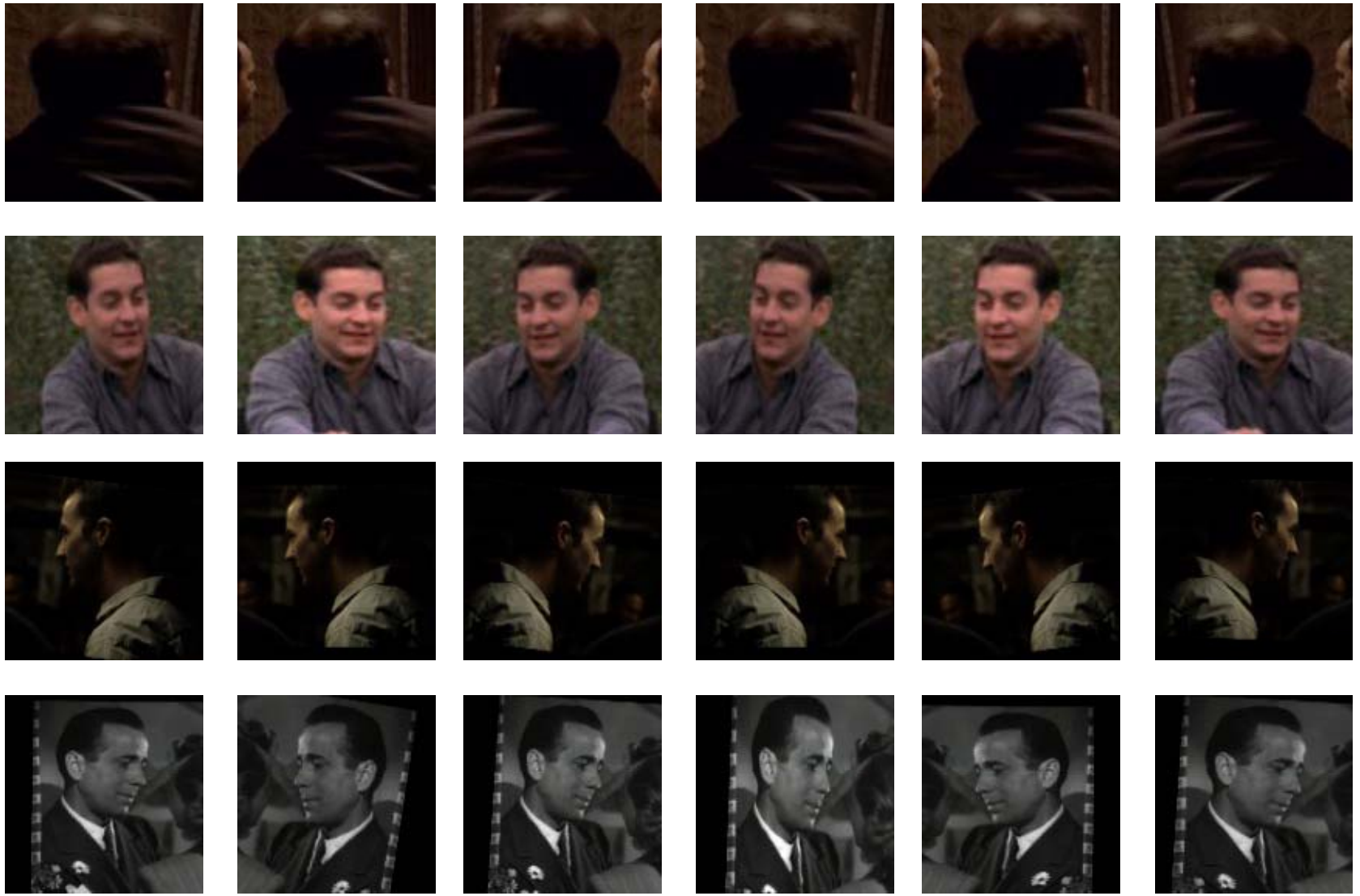
Positive windows

Note: common size and alignment

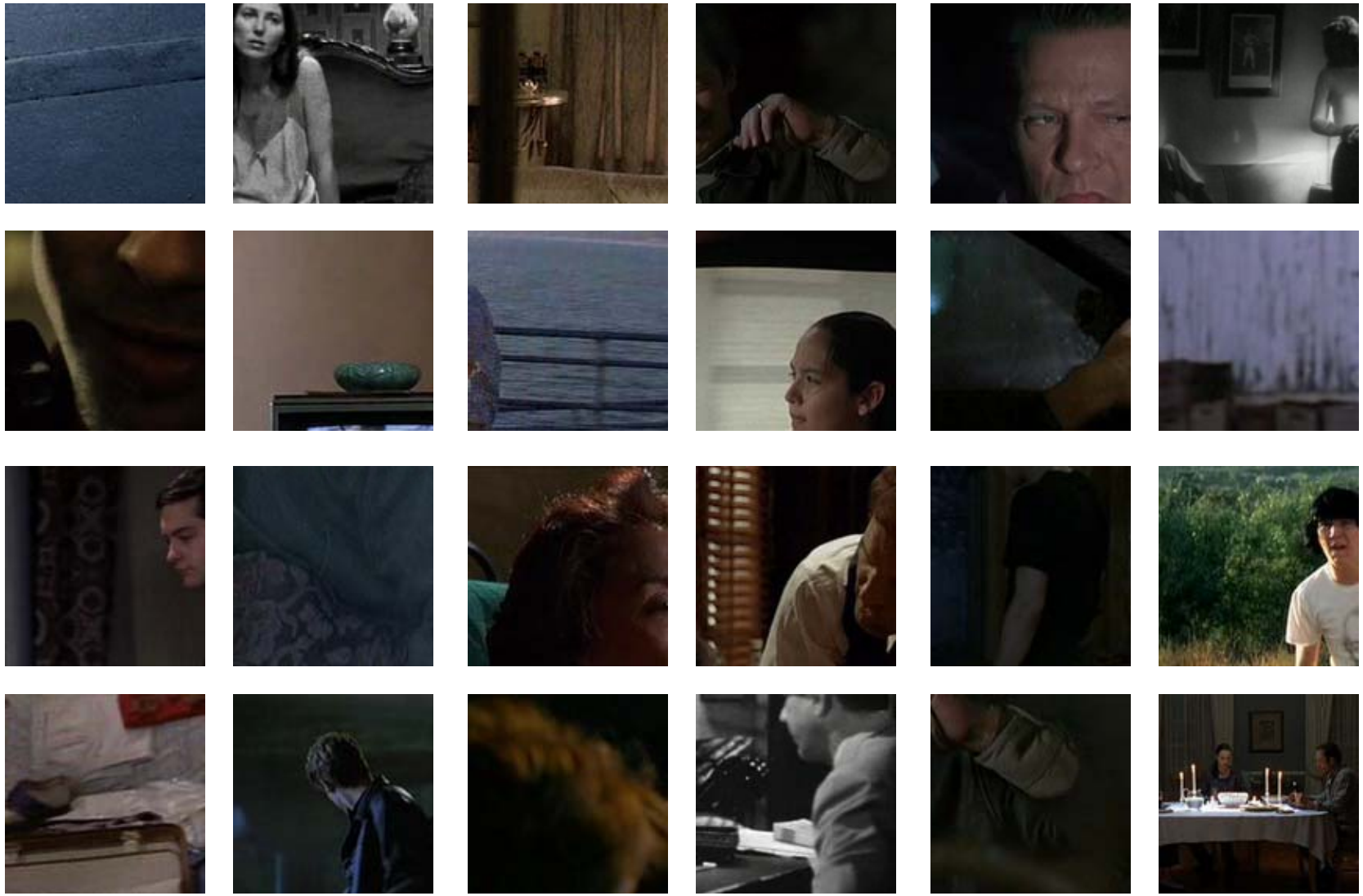
Jittered positives



Jittered positives

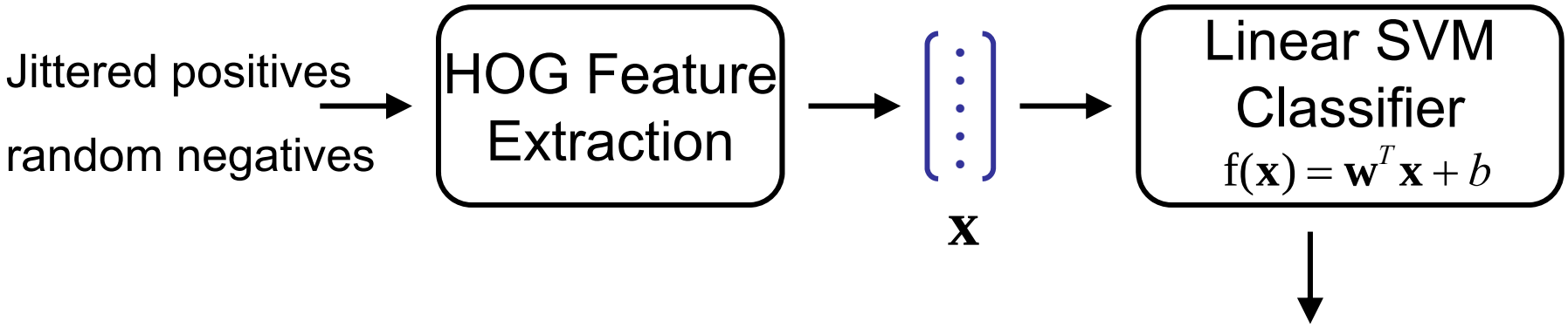


Random negatives



Random negatives

Window (Image) first stage classification

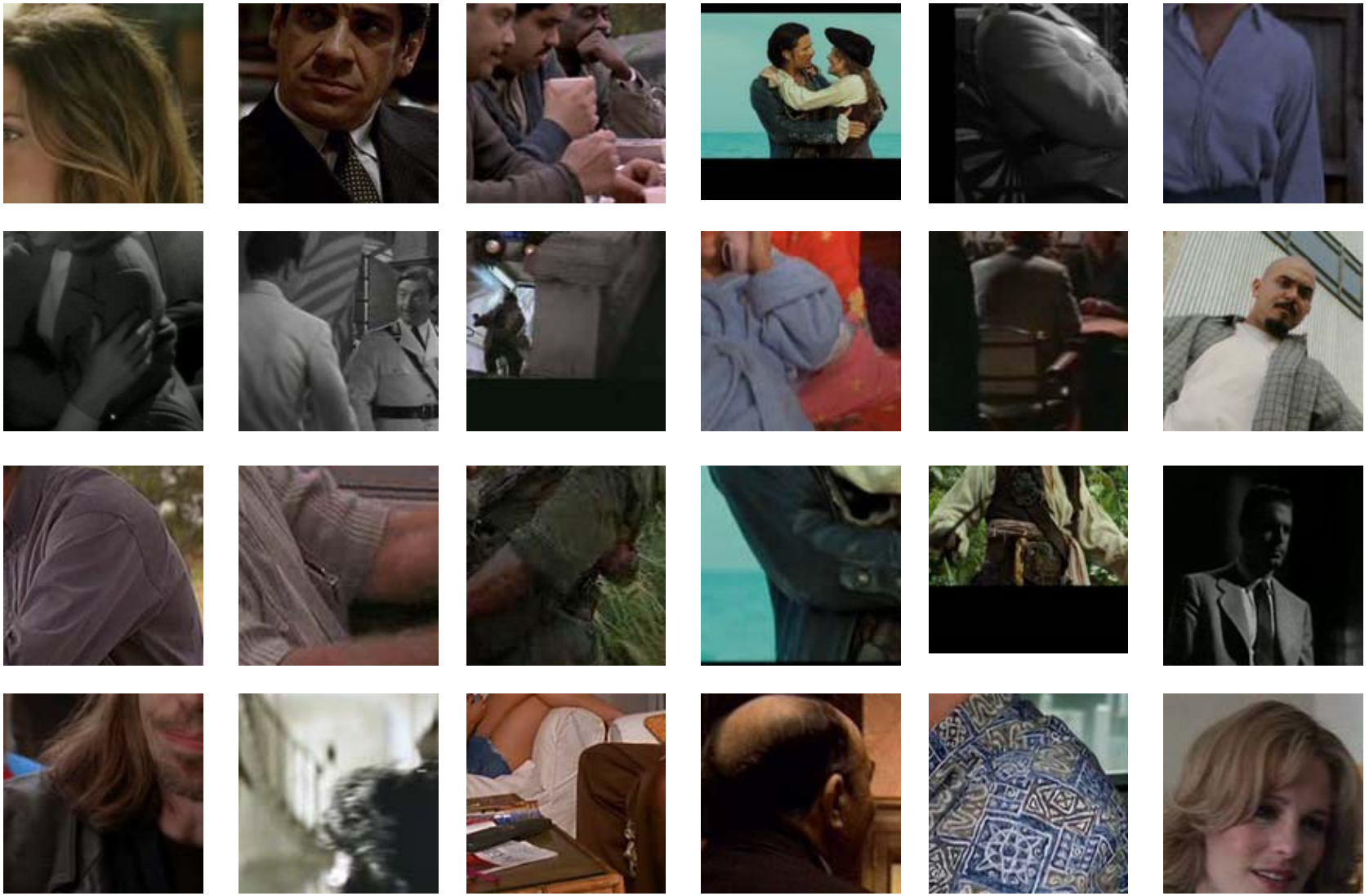


- find high scoring false positives detections

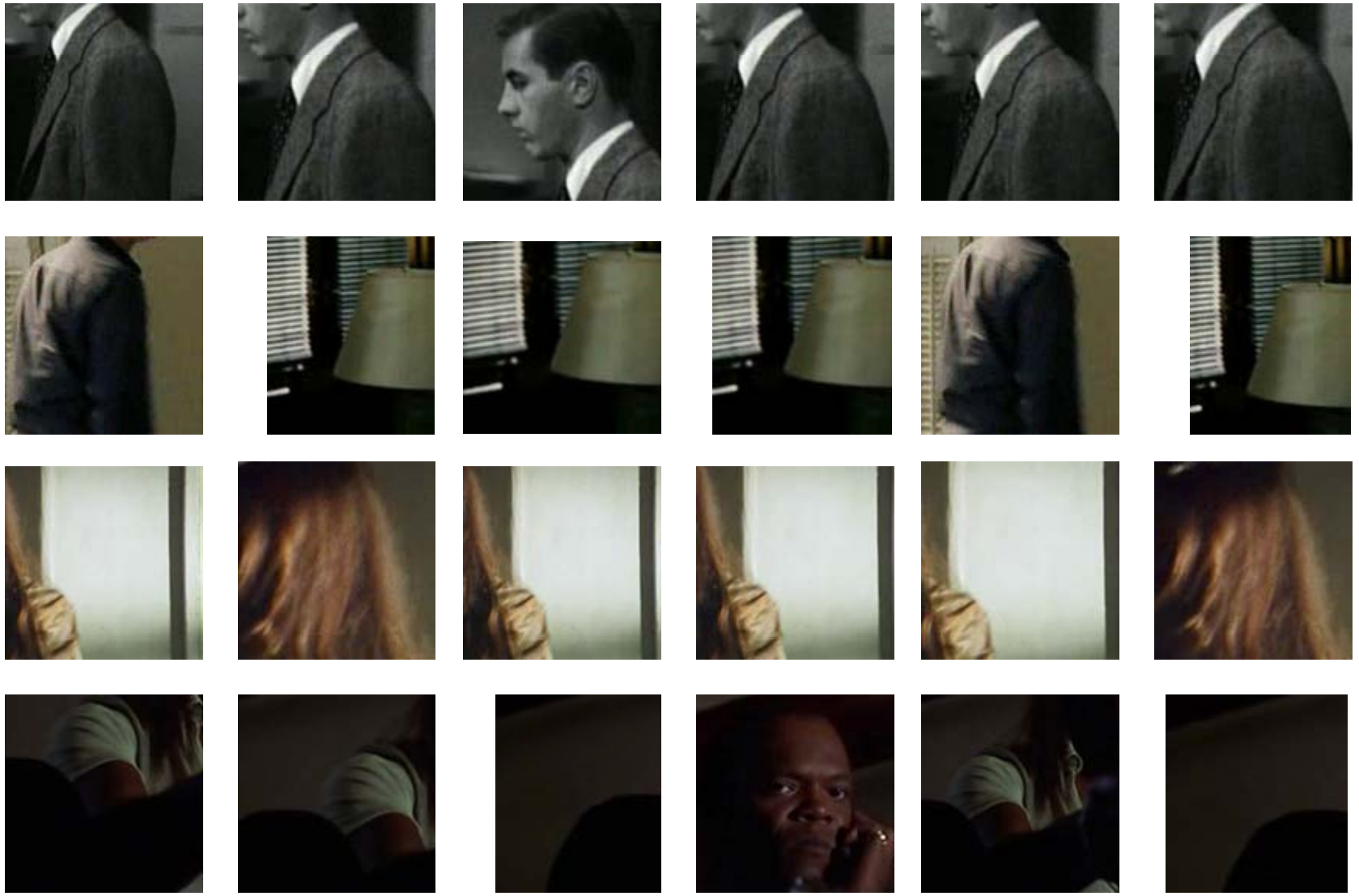
- these are the hard negatives for the next round of training

- cost = # training images x inference on each image

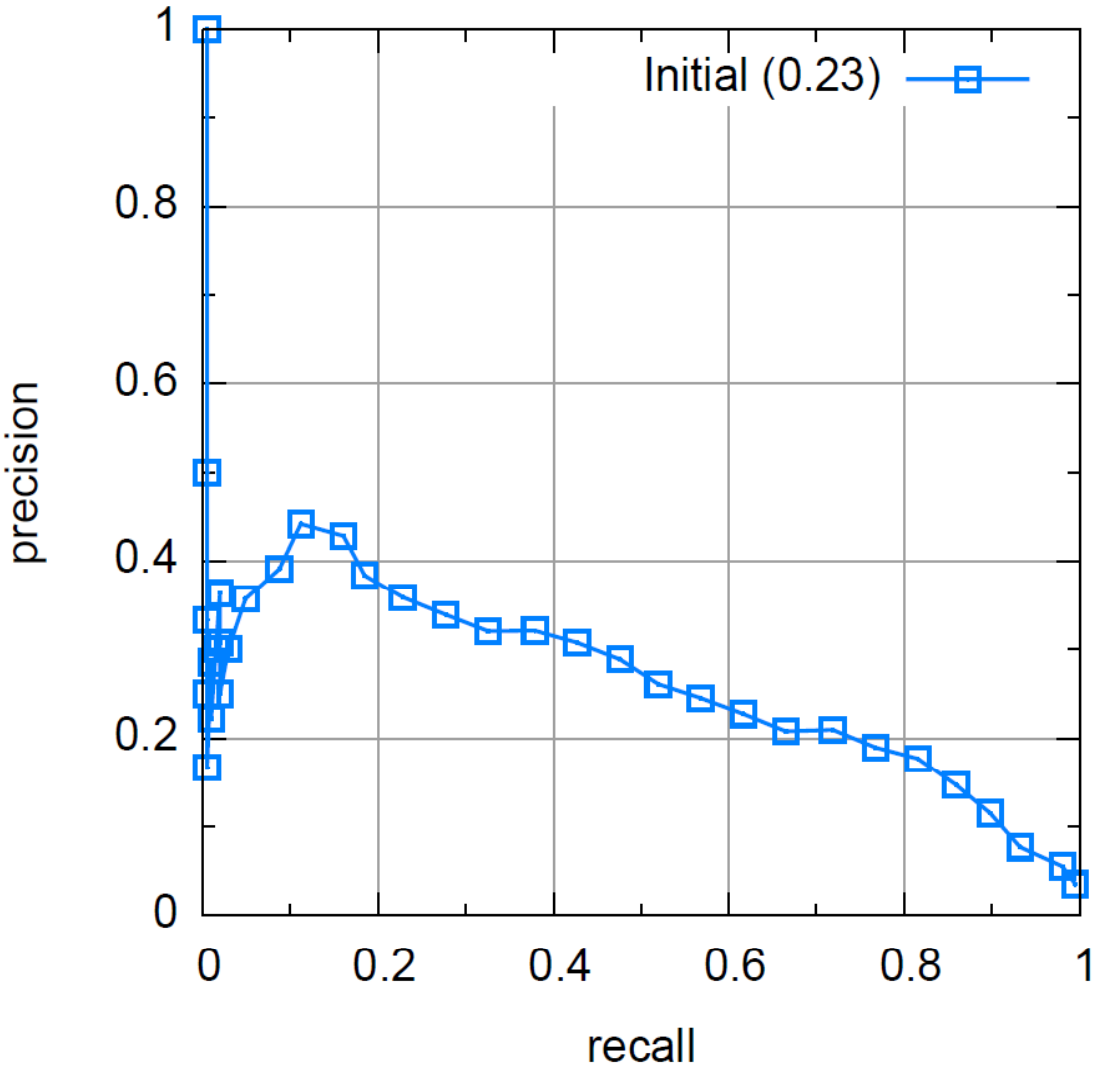
Hard negatives



Hard negatives

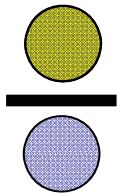


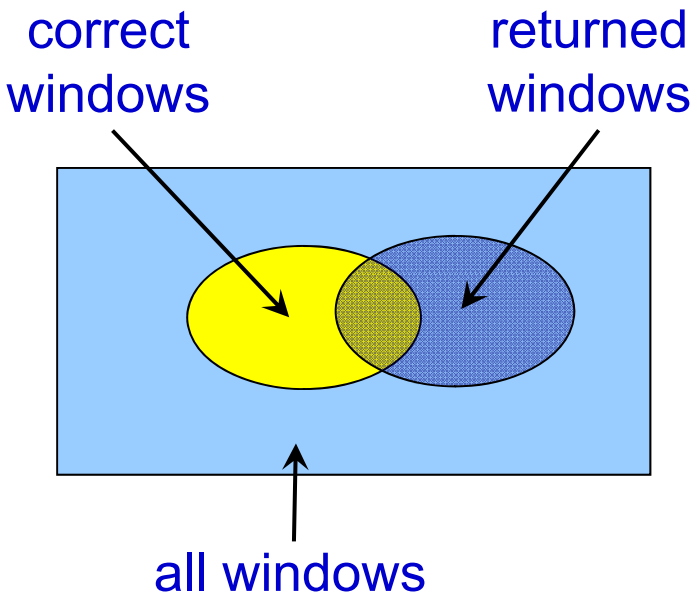
First stage performance on validation set



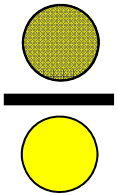
Precision – Recall curve

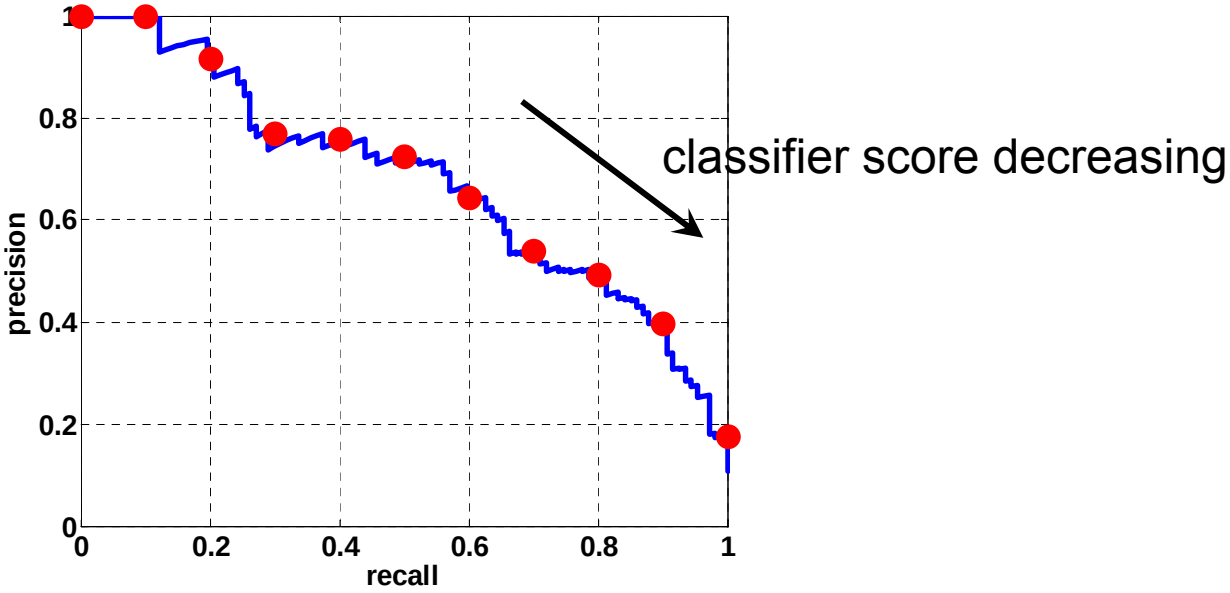
- Precision: % of returned windows that are correct



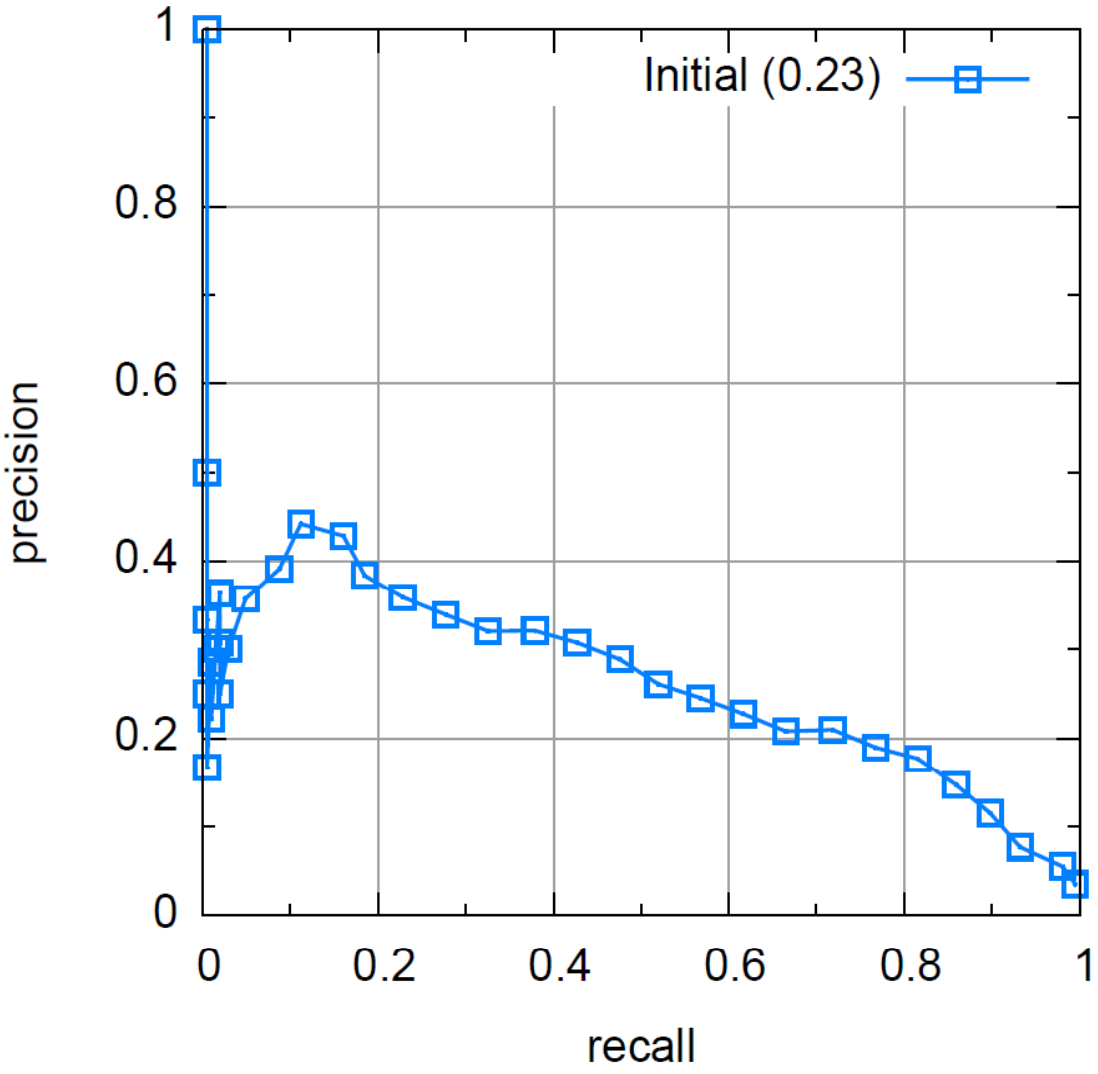


- Recall: % of correct windows that are returned

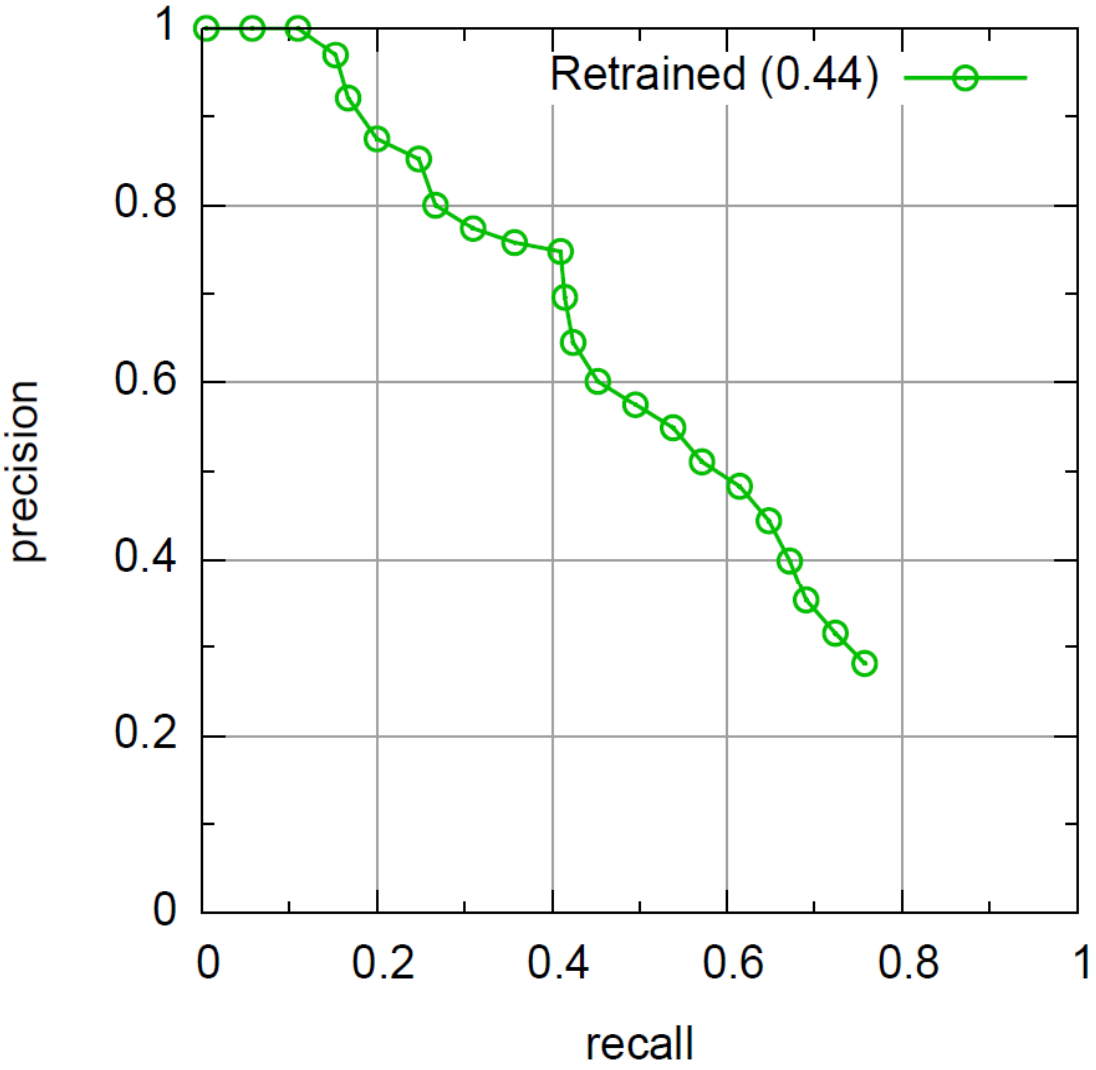




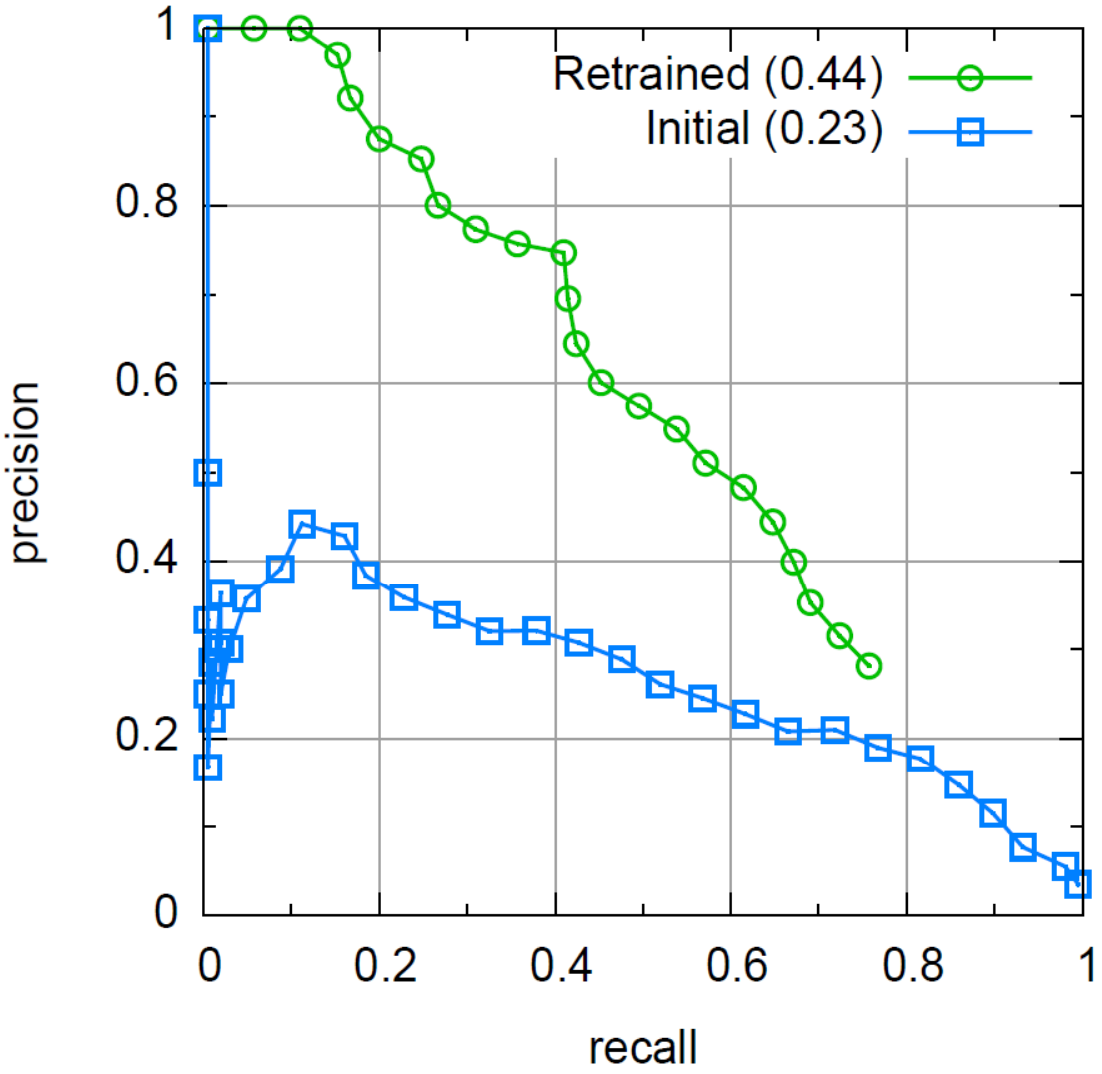
First stage performance on validation set



Performance after retraining

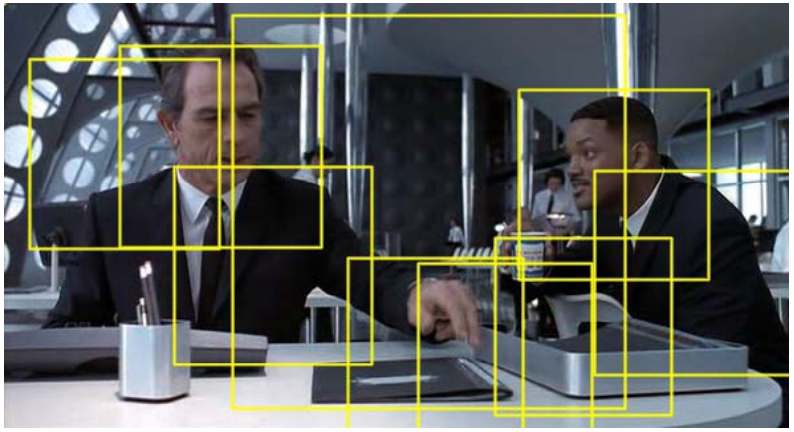


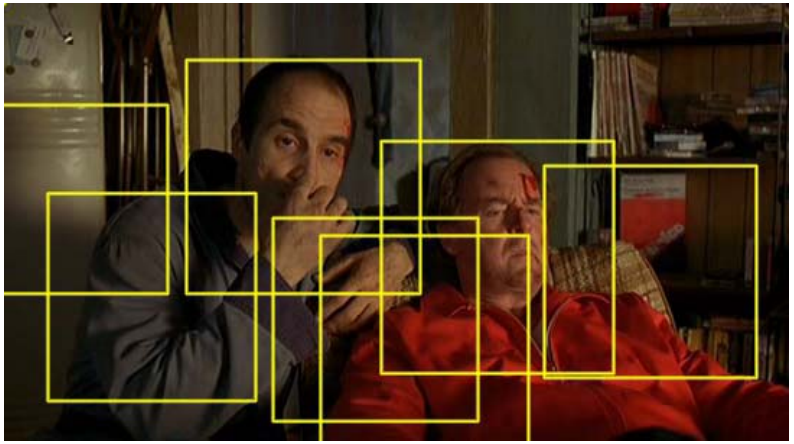
Effects of retraining



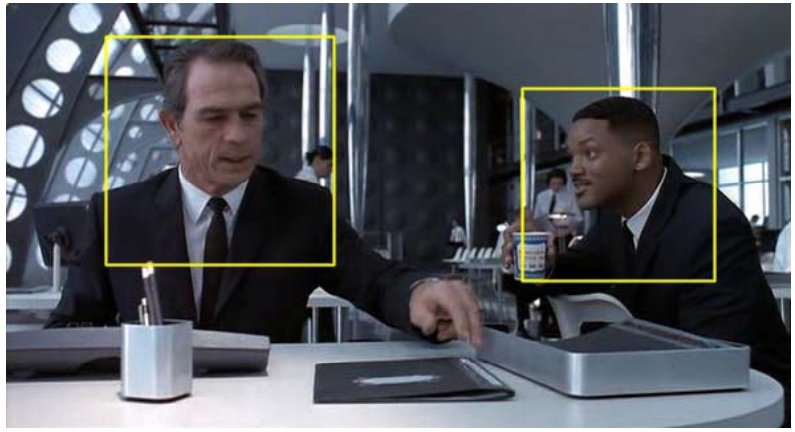
Side by side

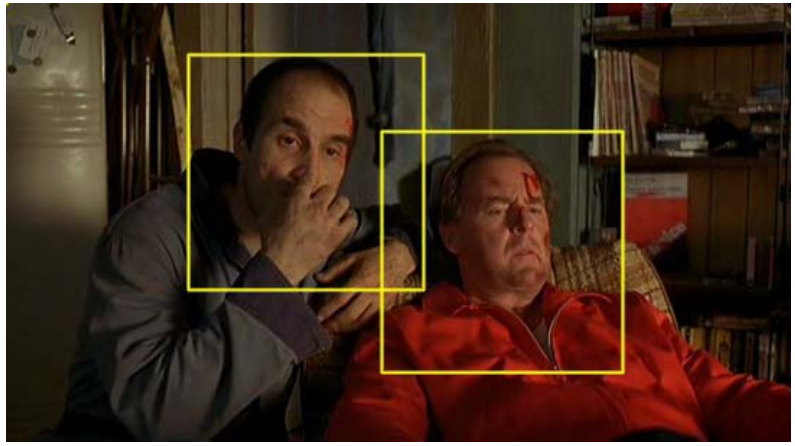
before retraining





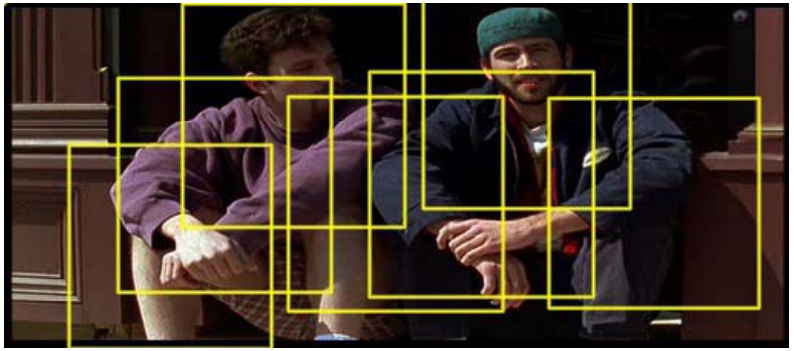
after retraining

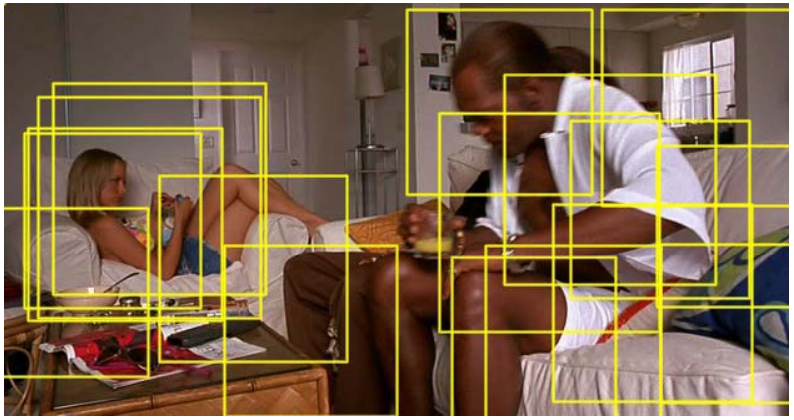




Side by side

before retraining

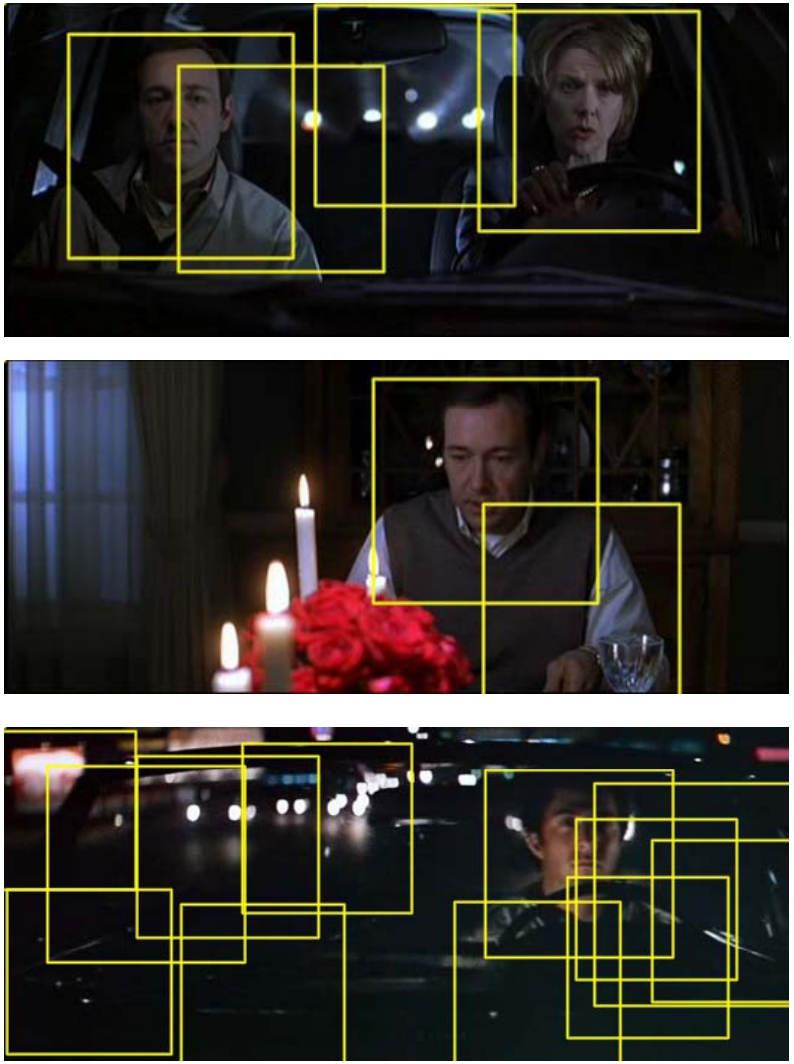




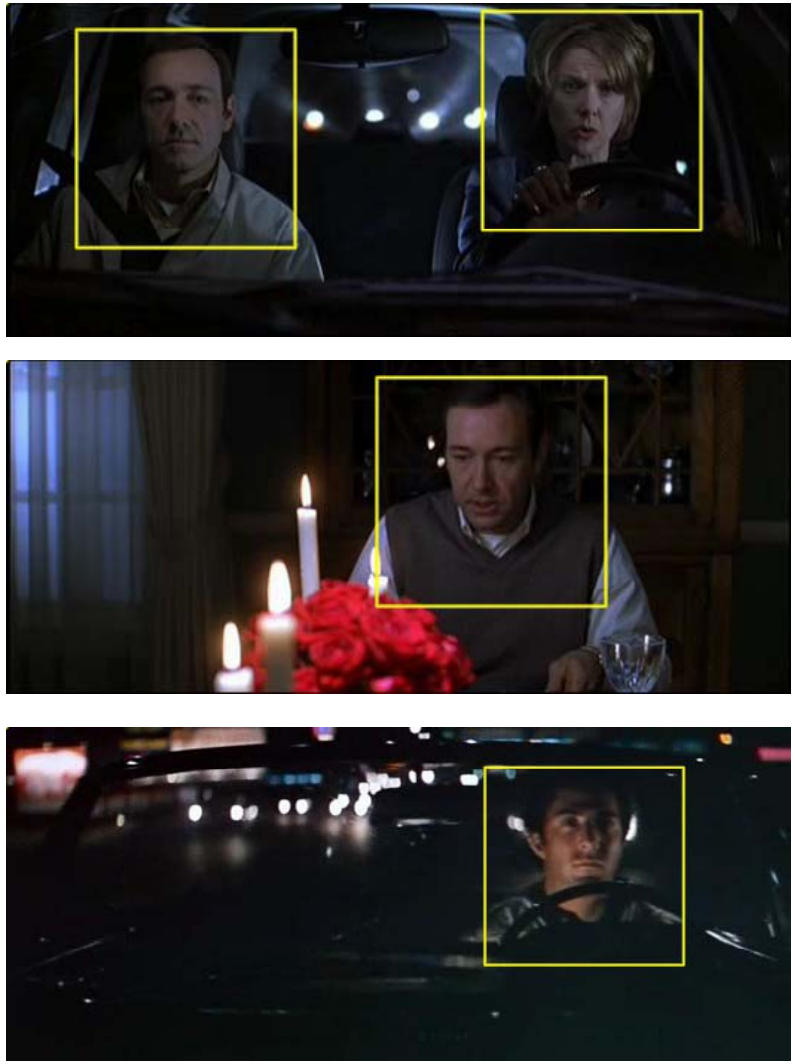
after retraining

Side by side

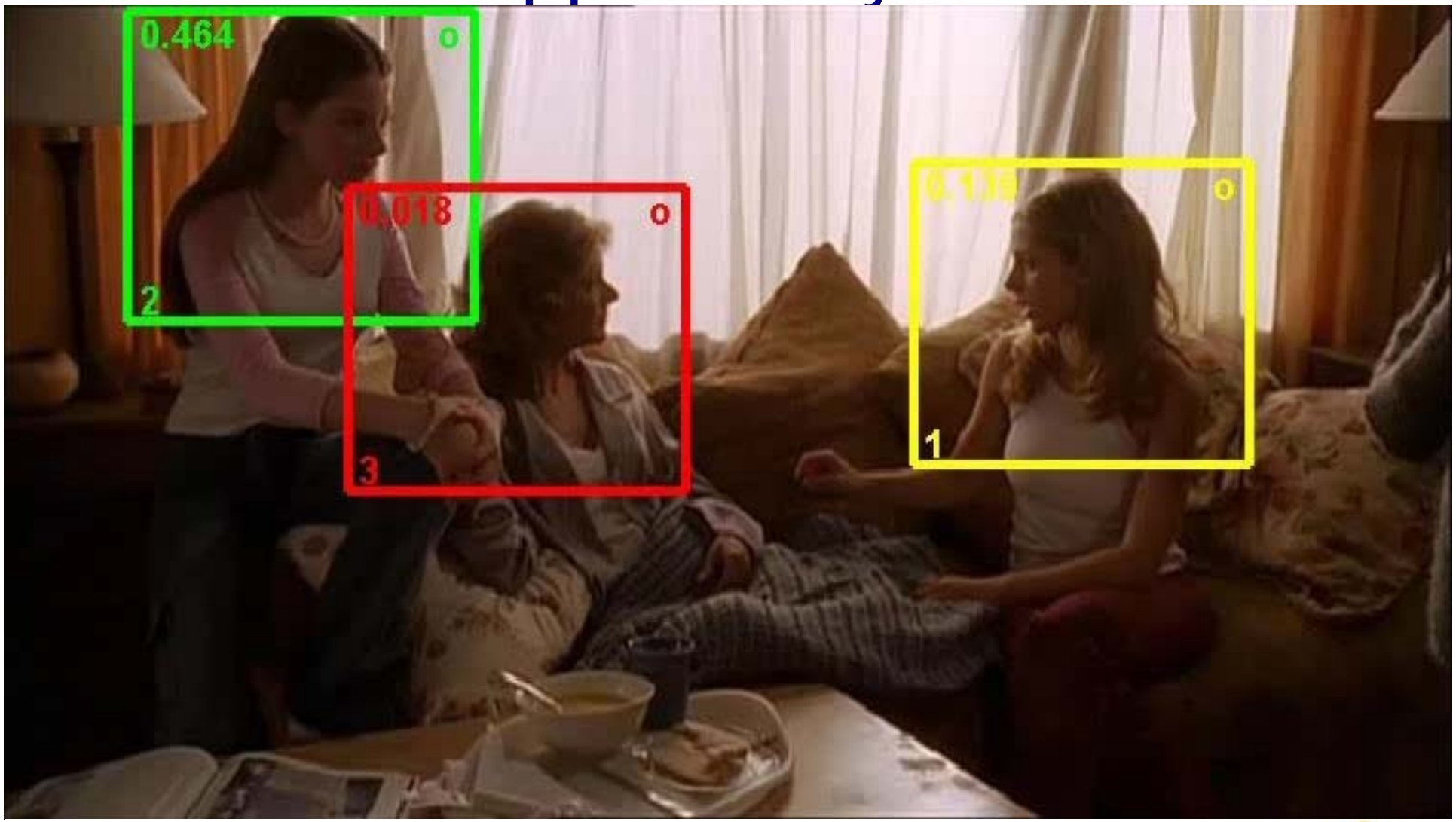
before retraining



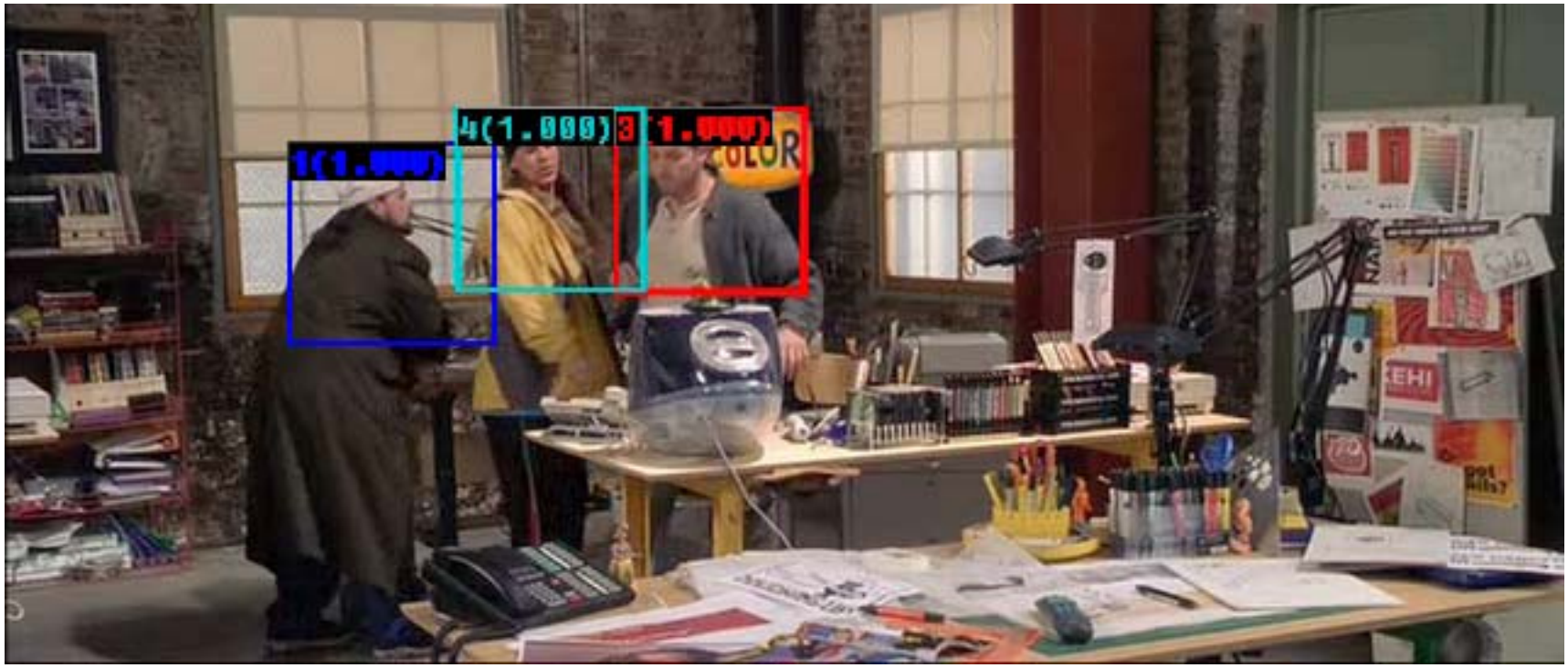
after retraining



Tracked upper body detections



Tracked upper body person detections



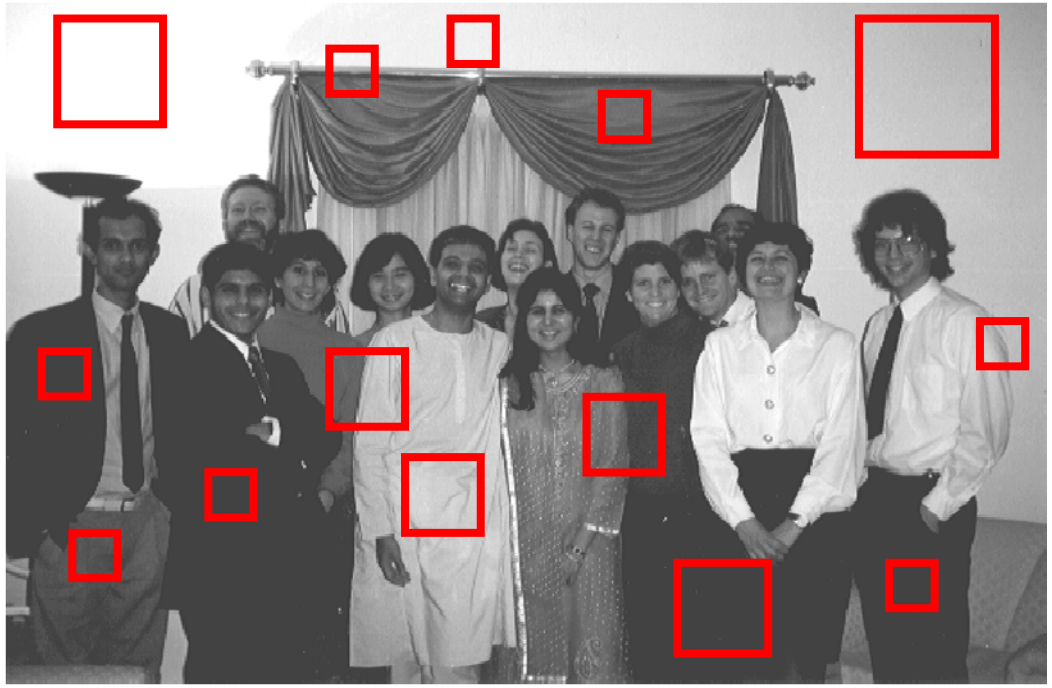
Combined face, upper body and full body detectors “vote” for upper body bounding boxes.

Detections are tracked and smoothed over video.

[Lezama, MVA thesis 2010]

Accelerating Sliding Window Search

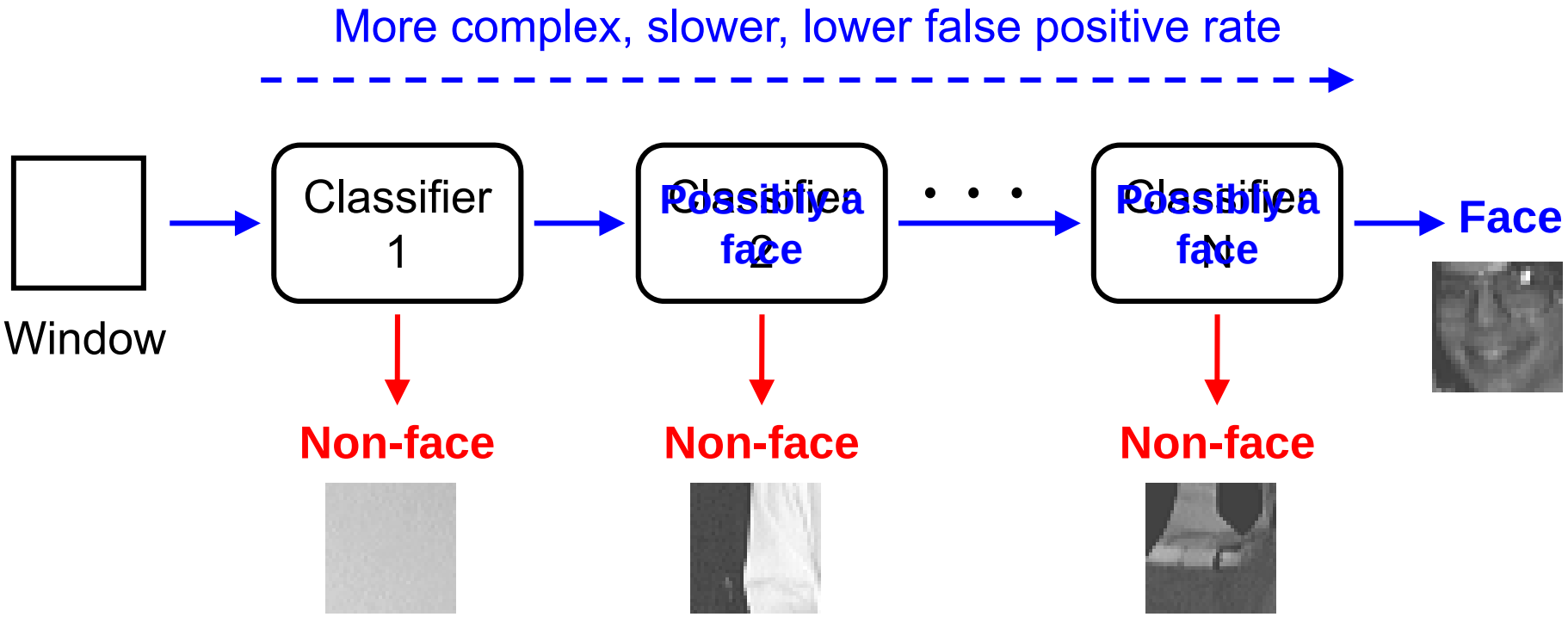
- Sliding window search is slow because so many windows are needed e.g. x × y × scale ≈ 100,000 for a 320×240 image



- Most windows are clearly not the object class of interest
- Can we speed up the search?

Cascaded Classification

• Build a sequence of classifiers with increasing complexity



• Reject easy non-objects using simpler and faster classifiers

Cascaded Classification

- Slow expensive classifiers only applied to a few windows ➔ significant speed-up

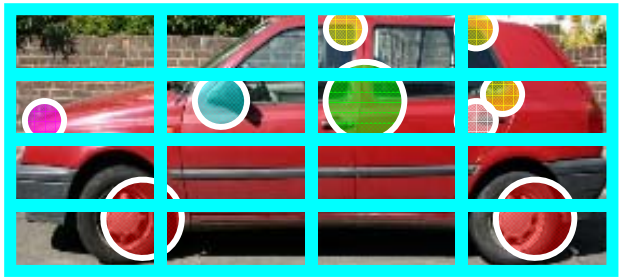
- Controlling classifier complexity/speed:
  - Number of support vectors [Romdhani et al, 2001]
  - Number of features [Viola & Jones, 2001]
  - Type of SVM kernel [Vedaldi et al, 2009]

Summary: Sliding Window Detection

- Can convert any image classifier into an object detector by sliding window. Efficient search methods available.

- Requirements for invariance are reduced by searching over e.g. translation and scale

- Spatial correspondence can be "engineered in" by spatial tiling



Outline

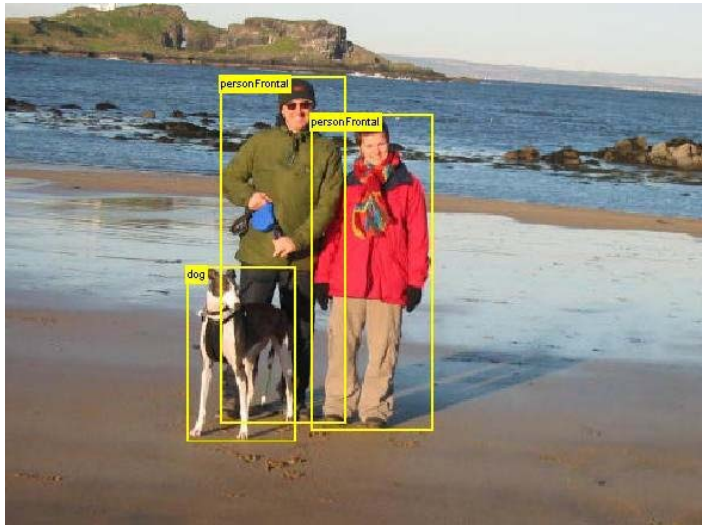
1. Sliding window detectors
2. Features and adding spatial information
3. HOG + linear SVM classifier
4. Two state of the art algorithms and PASCAL VOC
   - VOC challenge
   - Felzenswalb et al – multiple parts, latent SVM
5. The future and challenges

The PASCAL Visual Object Classes (VOC) Dataset and Challenge

Mark Everingham
Luc Van Gool
Chris Williams
John Winn
Andrew Zisserman

The PASCAL VOC Challenge

- Challenge in visual object recognition funded by PASCAL network of excellence

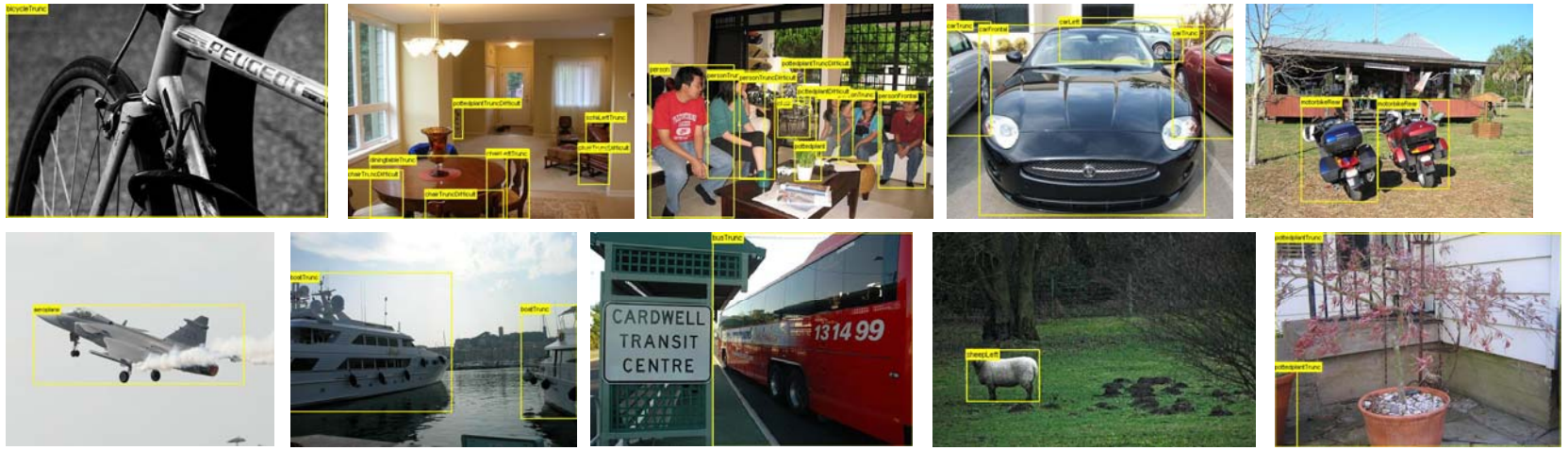


- Publicly available dataset of annotated images
- Main competitions in classification (is there an X in this image), detection (where are the X's), and segmentation (which pixels belong to X)
- "Taster competitions" in 2-D human "pose estimation" (2007-present) and static action classes
- Standard evaluation protocol (software supplied)

Dataset Content

- 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”



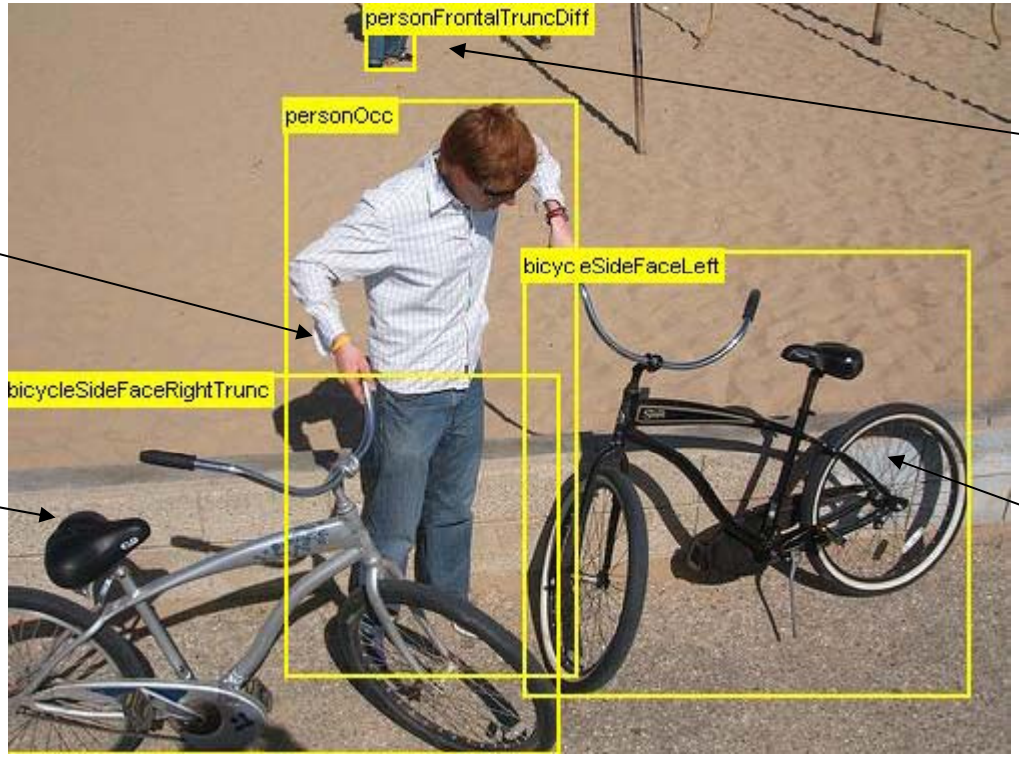
- Complex scenes, scale, pose, lighting, occlusion, ...

Annotation

- Complete annotation of all objects
- Annotated in one session with written guidelines

Occluded
Object is significantly occluded within BB

Truncated
Object extends beyond BB

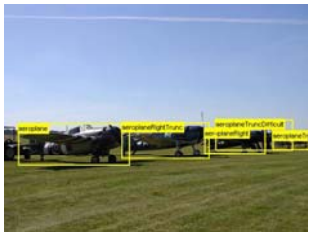


Difficult
Not scored in evaluation

Pose
Facing left

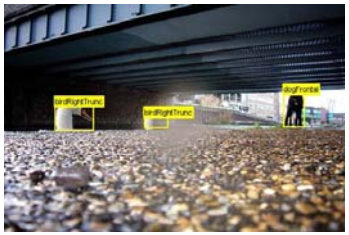
Examples

Aeroplane

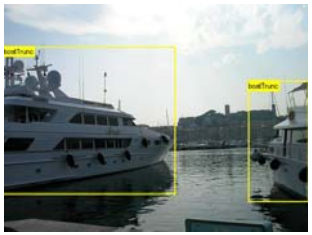


Bicycle

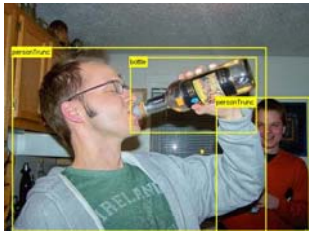
Bird

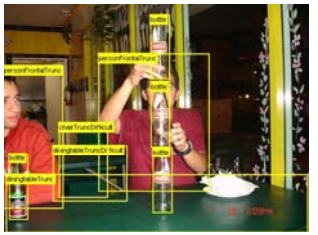


Boat



Bottle

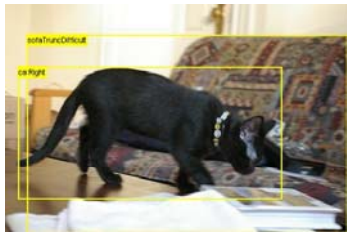




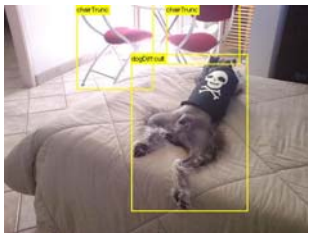
Bus

Car

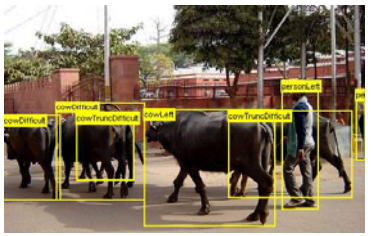
Cat



Chair

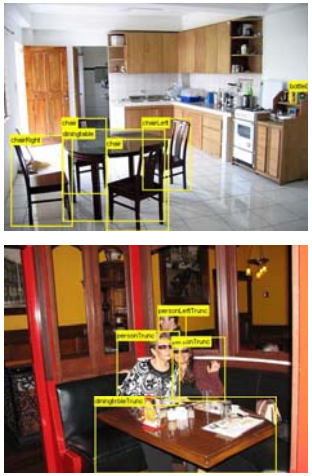


Cow

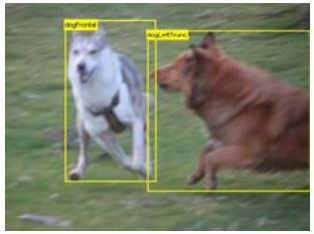


Examples

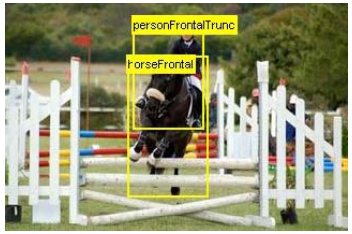
Dining Table

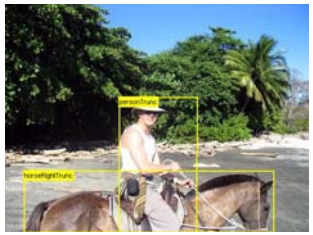


Dog



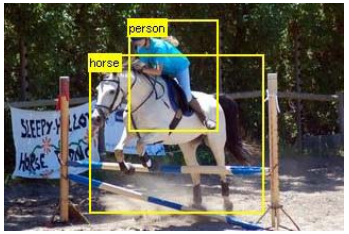
Horse

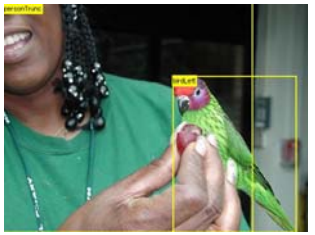




Motorbike

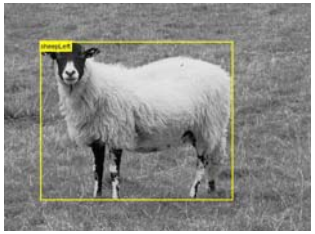
Person



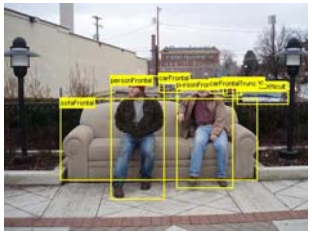


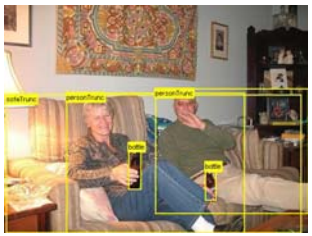
Potted Plant

Sheep



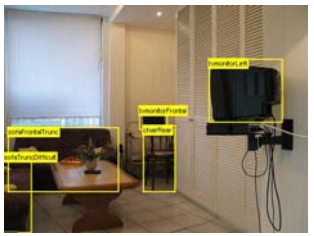
Sofa





Train

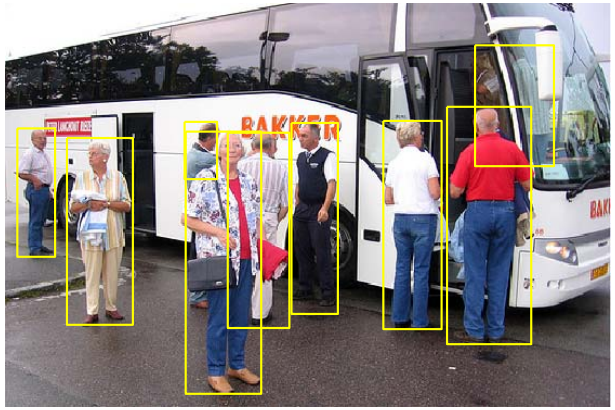
TV/Monitor



Main Challenge Tasks

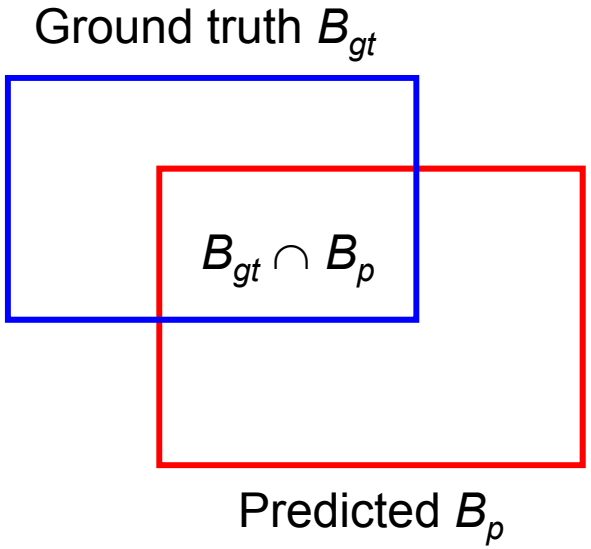
- **Classification**
  - Is there a dog in this image?
  - Evaluation by precision/recall

- **Detection**
  - Localize all the people (if any) in this image
  - Evaluation by precision/recall based on bounding box overlap

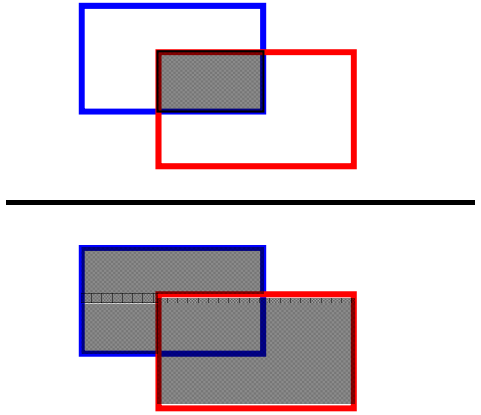


Detection: Evaluation of Bounding Boxes

- Area of Overlap (AO) Measure



$$AO(B_{gt}, B_p) = \frac{|B_{gt} \bigcap B_p|}{|B_{gt} \bigcup B_p|}$$

Detection if  > Threshold

50%

Dataset Statistics

	train		val		trainval		test	
	Images	Objects	Images	Objects	Images	Objects	Images	Objects
Aeroplane	201	267	206	266	407	533		
Bicycle	167	232	181	236	348	468		
Bird	262	381	243	379	505	760		
Boat	170	270	155	267	325	537		
Bottle	220	394	200	393	420	787		
Bus	132	179	126	186	258	365		
Car	372	664	358	653	730	1,317		
Cat	266	308	277	314	543	622		
Chair	338	716	330	713	668	1,429		
Cow	86	164	86	172	172	336		
Diningtable	140	153	131	153	271	306		
Dog	316	391	333	392	649	783		
Horse	161	237	167	245	328	482		
Motorbike	171	235	167	234	338	469		
Person	1,333	2,819	1,446	2,996	2,779	5,815		
Pottedplant	166	311	166	316	332	627		
Sheep	67	163	64	175	131	338		
Sofa	155	172	153	175	308	347		
Train	164	190	160	191	324	381		
Tvmonitor	180	259	173	257	353	516		
Total	3,473	8,505	3,581	8,713	7,054	17,218	6,650	16,829

True Positives - Bicycle

UoCTTI_LSVM-MDPM

OXFORD_MKL

NECUIUC_CLS-DTCT

False Positives - Bicycle

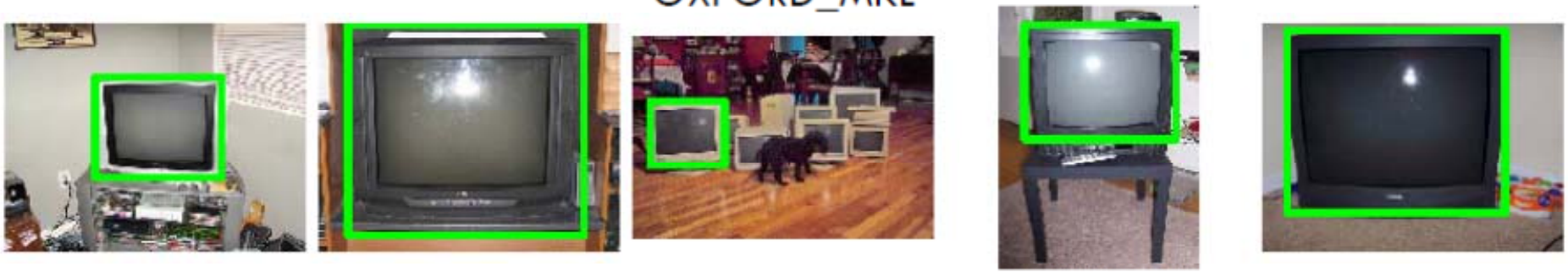
UoCTTI_LSVM-MDPM

OXFORD_MKL

NECUIUC_CLS-DTCT

True Positives – TV/monitor

OXFORD_MKL



UoCTTI_LSVM-MDPM

LEAR_CHI-SVM-SIFT-HOG-CLS

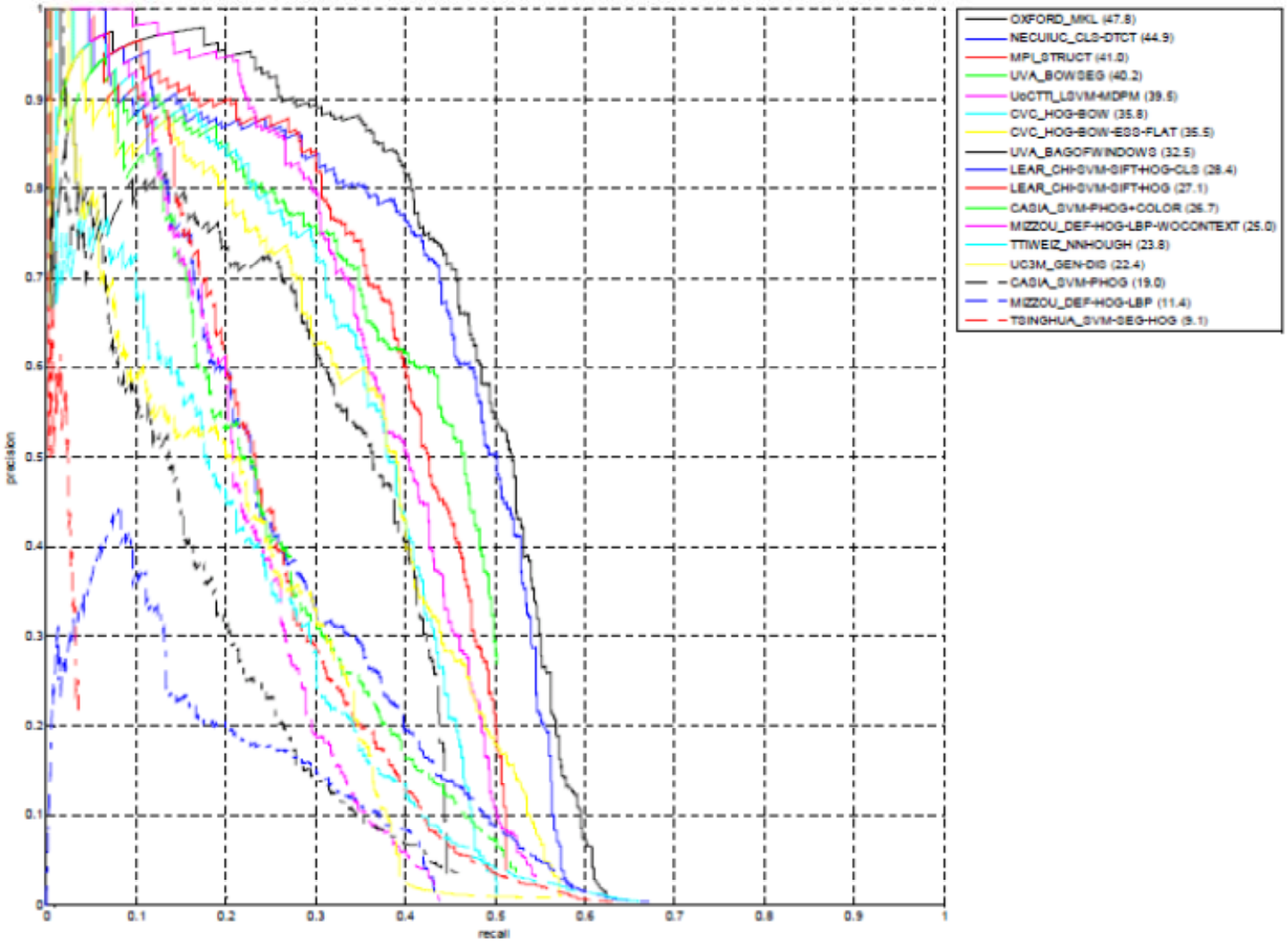
False Positives – TV/monitor

OXFORD_MKL

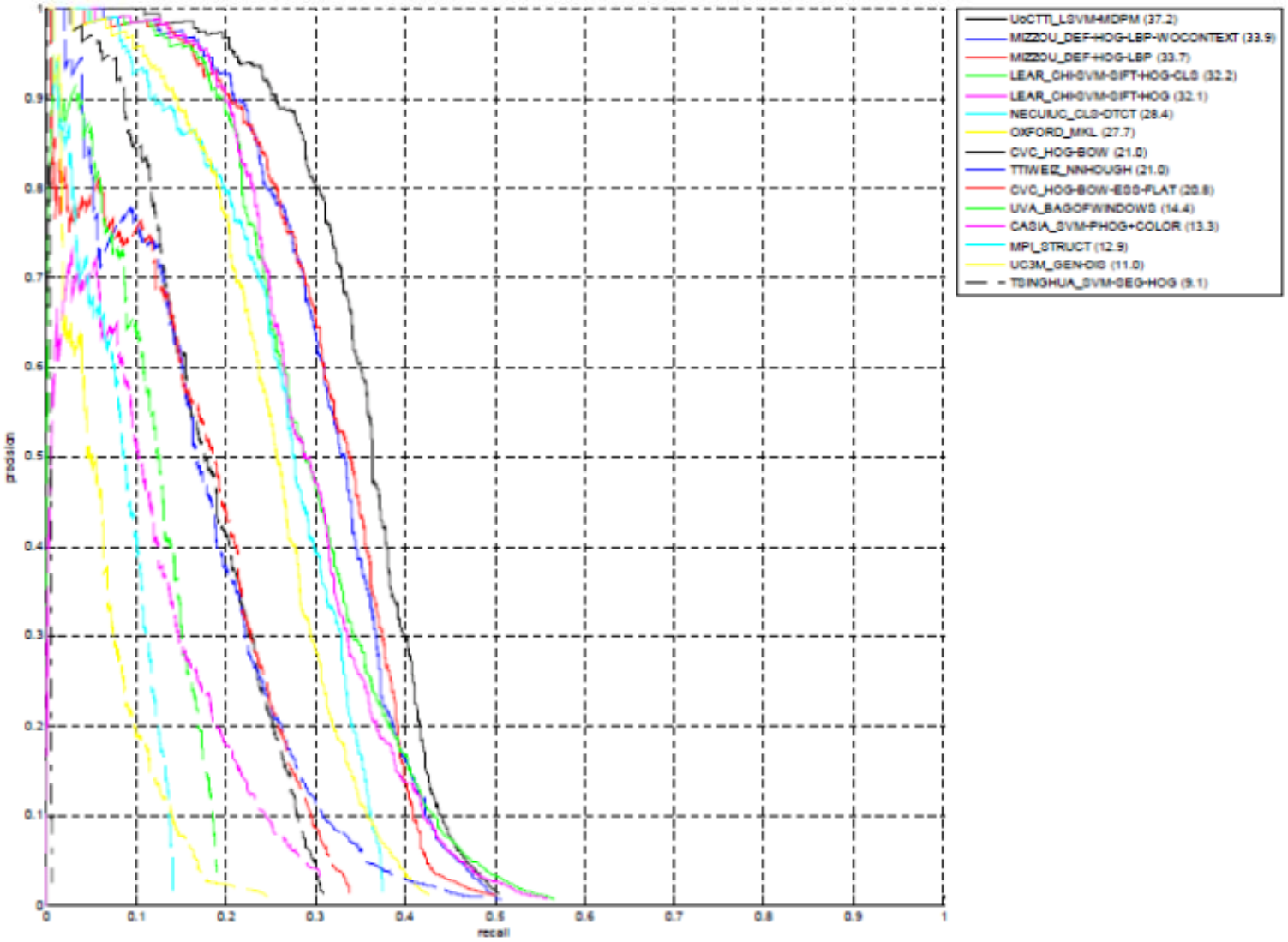
UoCTTI_LSVM-MDPM

LEAR_CHI-SVM-SIFT-HOG-CLS

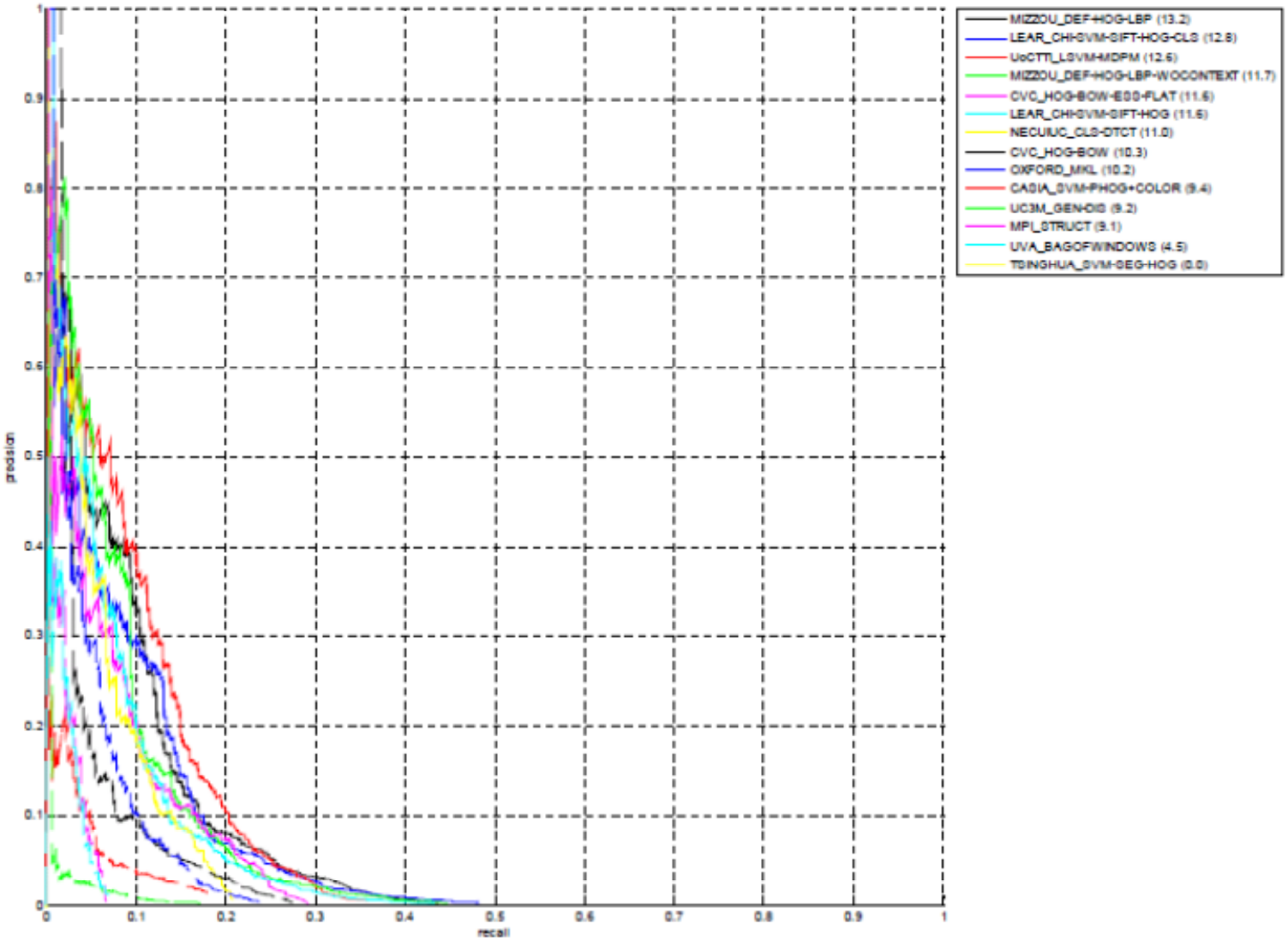
Precision/Recall - Aeroplane



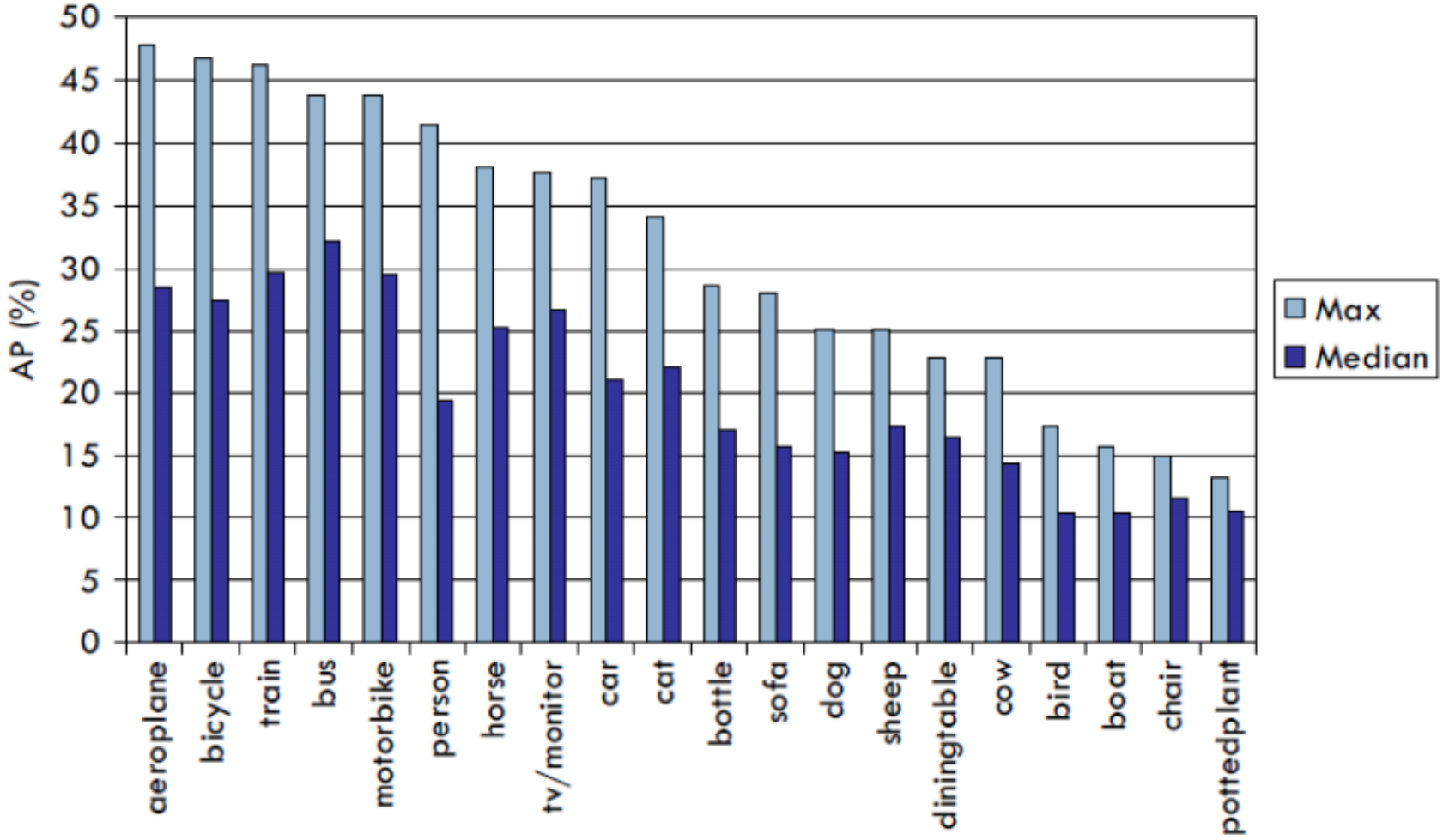
Precision/Recall - Car



Precision/Recall – Potted plant



AP by Class Detection



Wide variety of methods: sliding window, combination with whole image classifiers, segmentation based

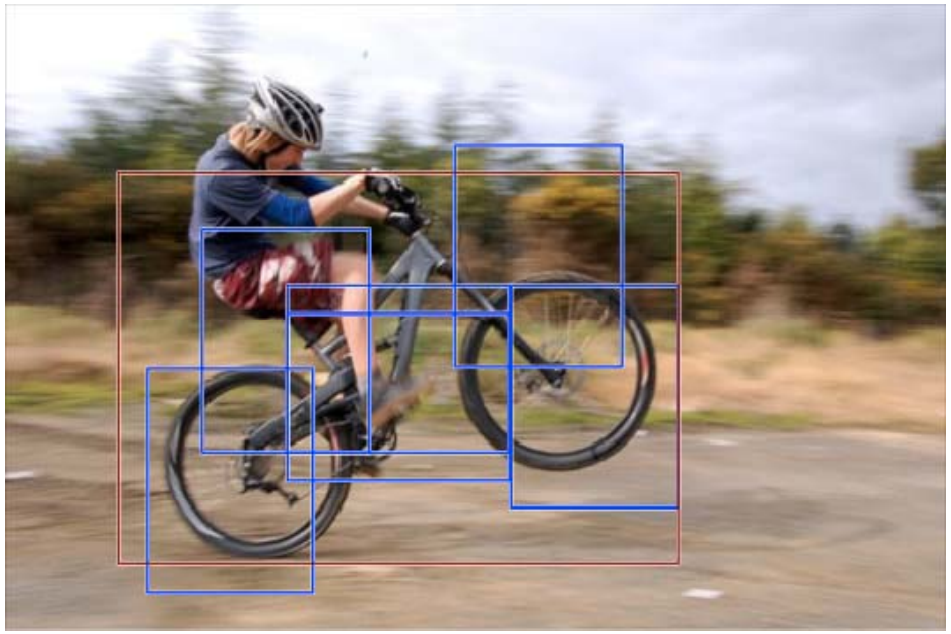
Object Detection with Discriminatively Trained Part Based Models

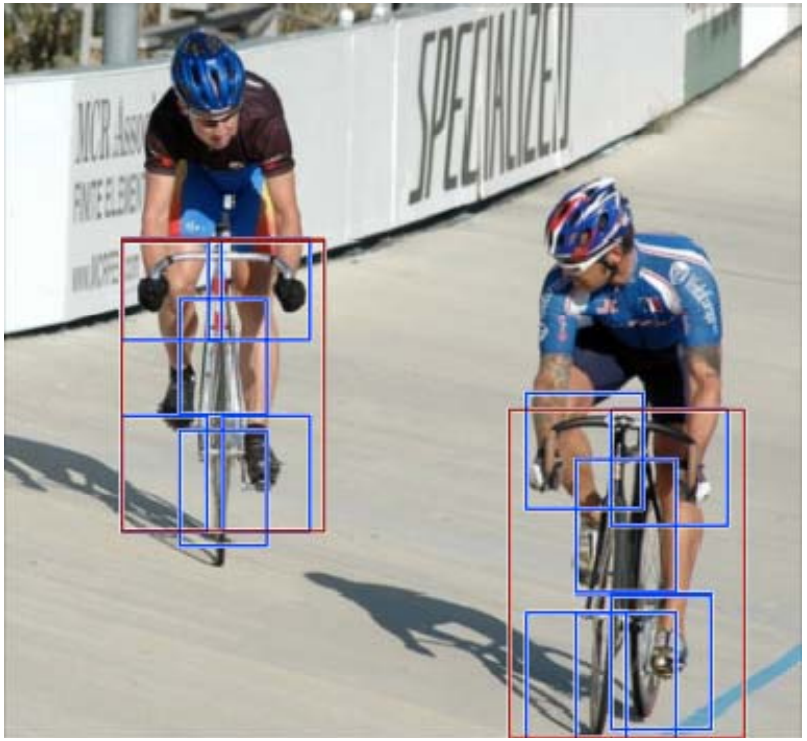
Pedro F. Felzenszwalb, David Mcallester,
Deva Ramanan, Ross Girshick

PAMI 2010

Matlab code available online:
http://www.cs.brown.edu/~pff/latent/

Approach

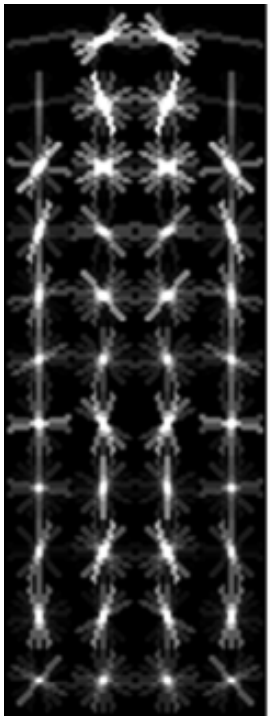




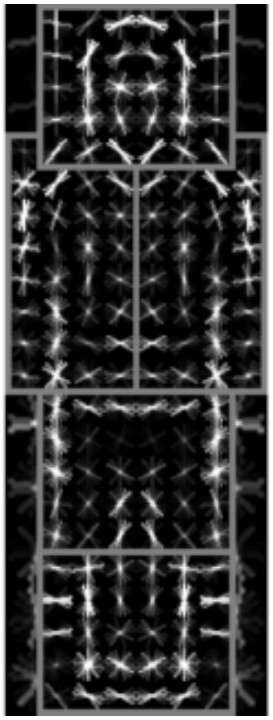
- Mixture of deformable part-based models
  - One component per “aspect” e.g. front/side view
- Each component has global template + deformable parts
- Discriminative training from bounding boxes alone

Example Model

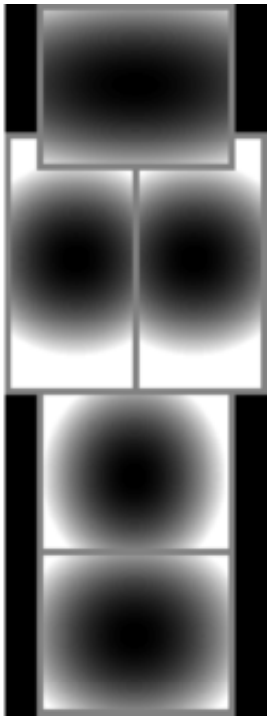
- One component of person model



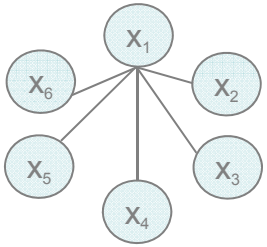
root filters
coarse resolution



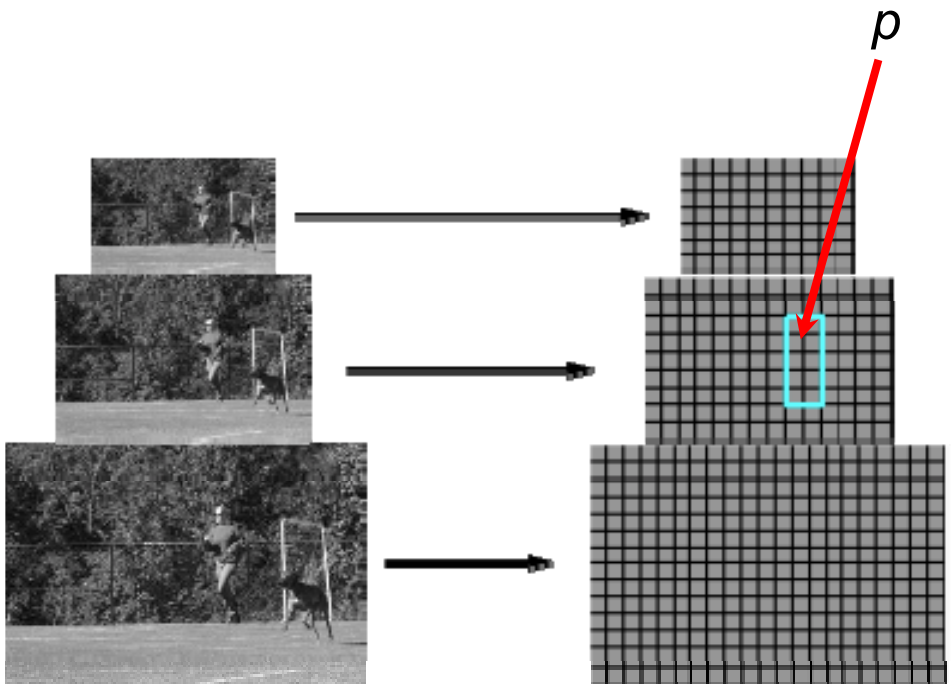
part filters
finer resolution



deformation
models



Starting Point: HOG Filter



HOG pyramid *H*

Filter *F*

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

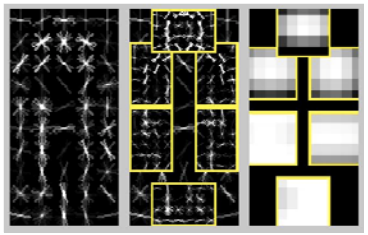
$\varphi(p, H)$ = concatenation of HOG features from subwindow specified by *p*

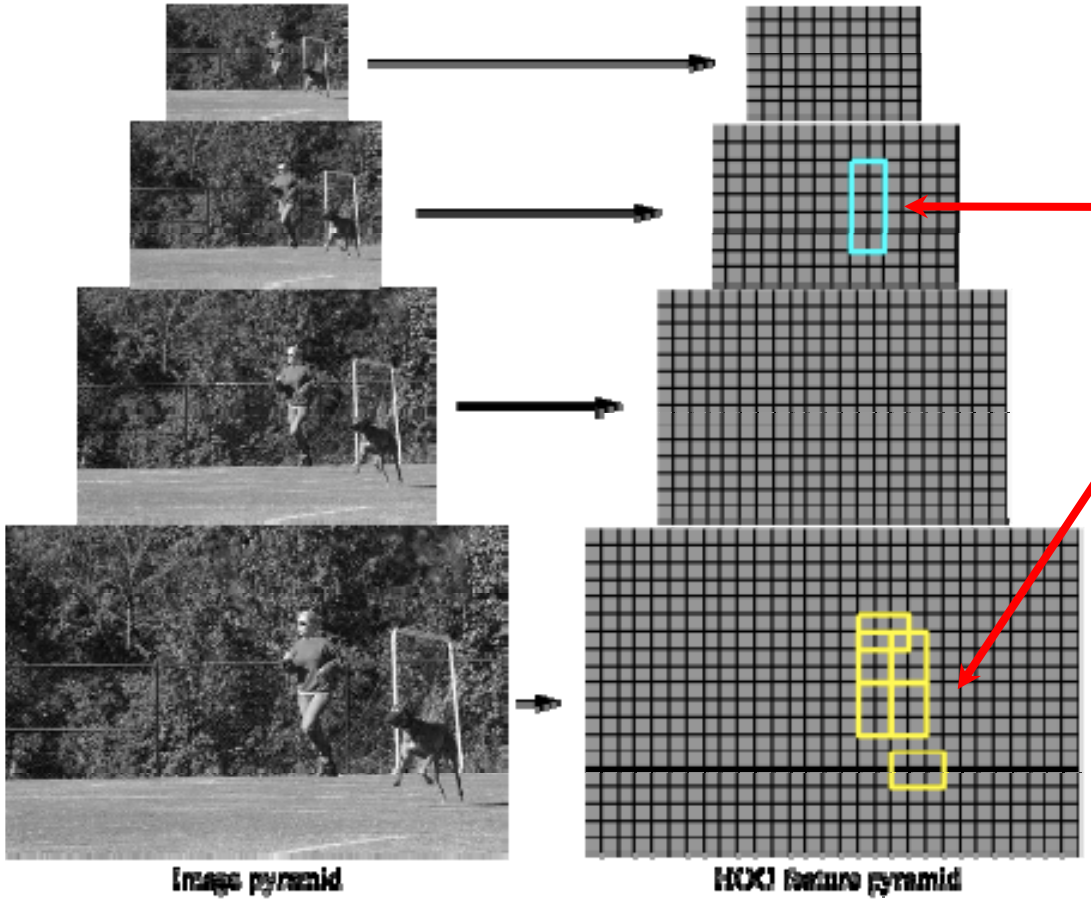
- Search: sliding window over position and scale
- Feature extraction: HOG Descriptor
- Classifier: Linear SVM

Dalal & Triggs [2005]

Object Hypothesis

- Position of root + each part
- Each part: HOG filter (at higher resolution)





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

p_0 : location of root

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

Score is sum of filter scores minus deformation costs

Score of a Hypothesis

Appearance term

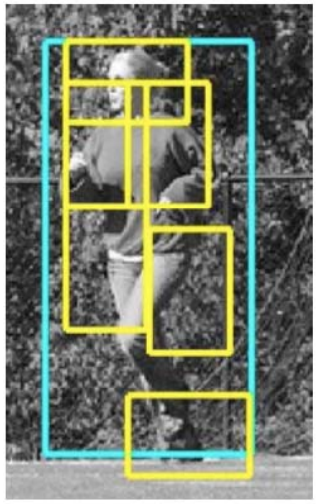
Spatial prior

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

filters

displacements

deformation parameters



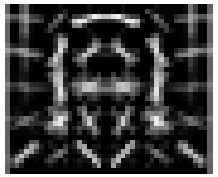
$$\text{score}(z) = \beta \cdot \Psi(H, z)$$

concatenation of filters and deformation parameters

concatenation of HOG features and part displacement features

- Linear classifier applied to feature subset defined by hypothesis

Part Detection



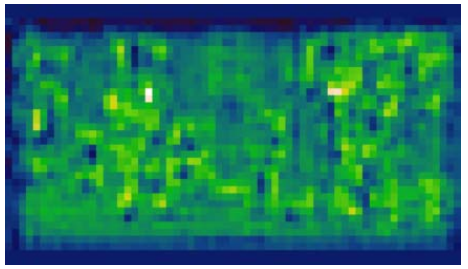
head filter

input image

Response of filter in l-th pyramid level

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

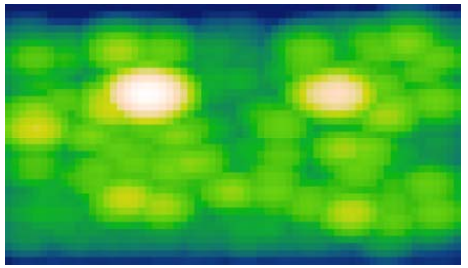
cross-correlation



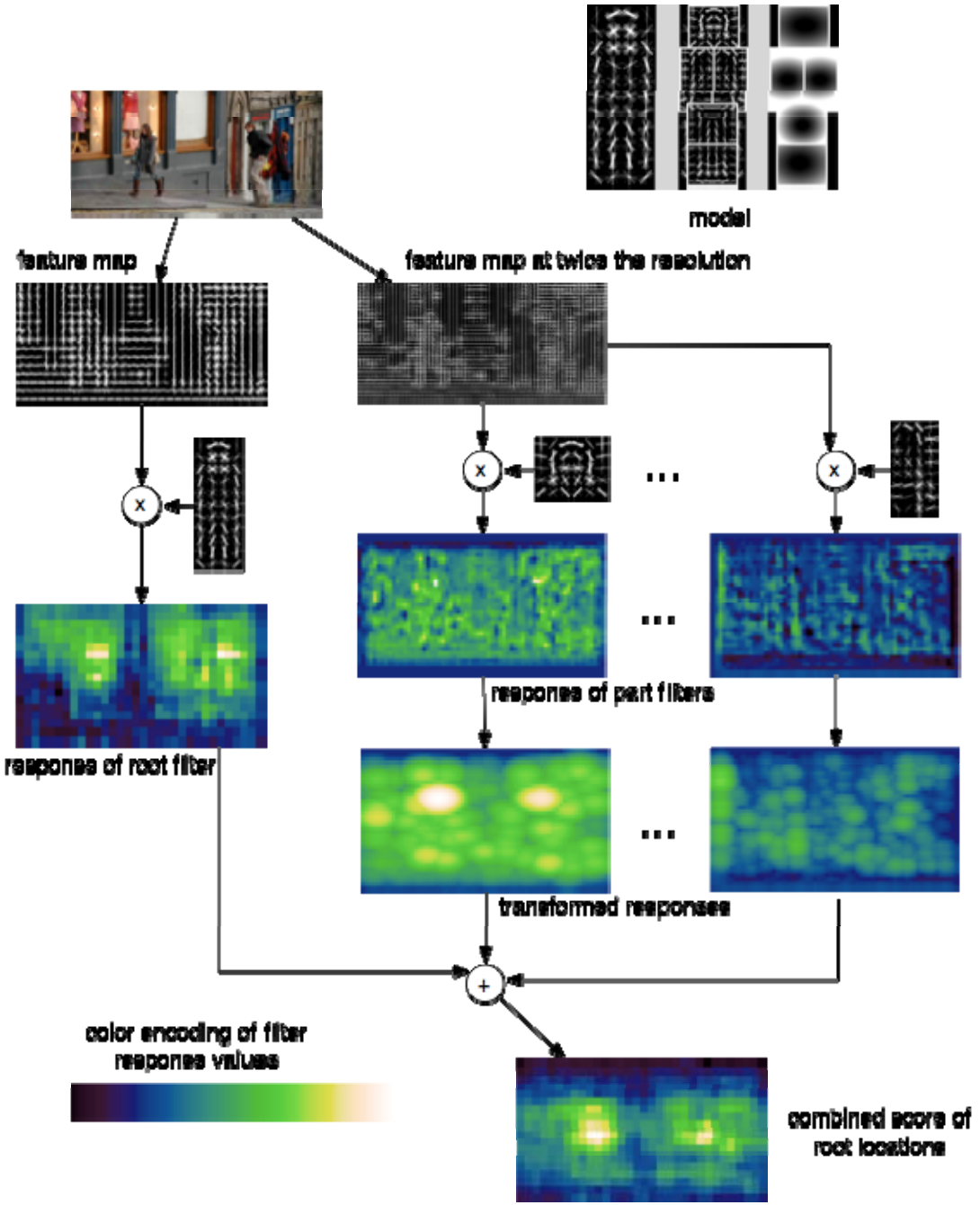
Transformed response

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

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



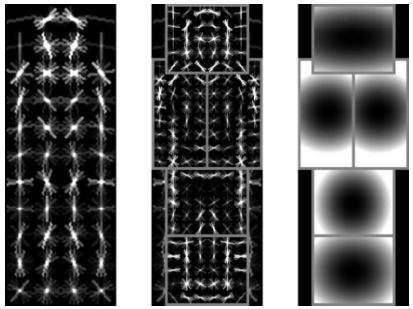
System


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

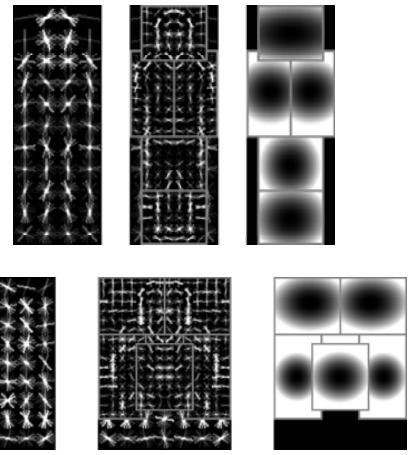
Training

- Training data = images + bounding boxes
- Need to learn: model structure, filters, deformation costs

Training



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

- Which component?
- Where are the parts?

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

Regularizer

"Hinge loss" on one training example

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

SVM objective

Latent SVM Training

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

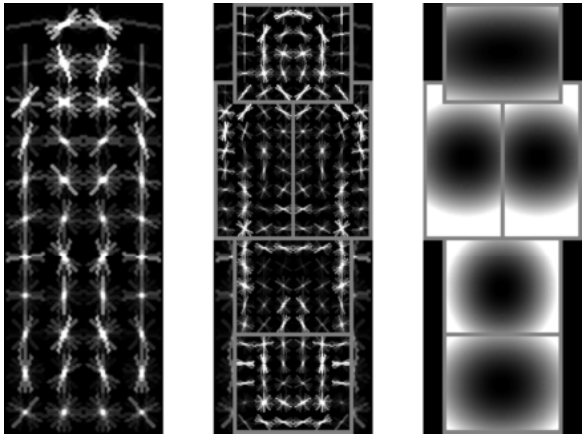
- Convex if we fix z for positive examples

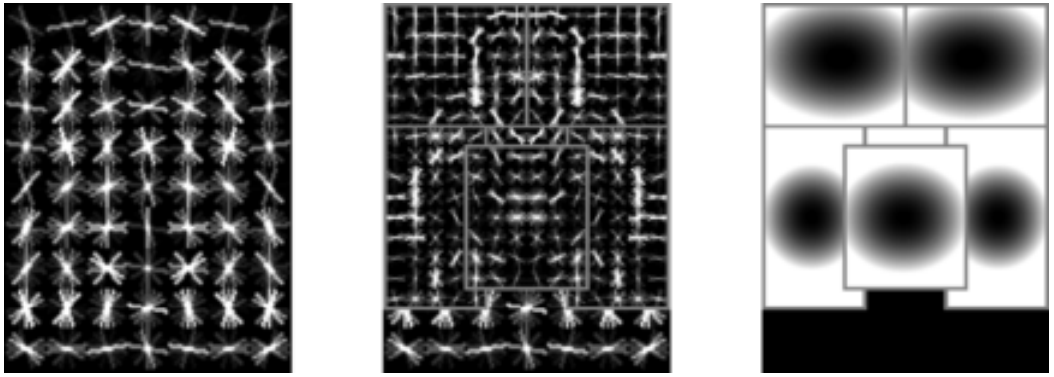
- Optimization:
  - Initialize β and iterate:
    - Pick best z for each positive example
    - Optimize β with z fixed

Alternation strategy

- Local minimum: needs good initialization
  - Parts initialized heuristically from root

Person Model





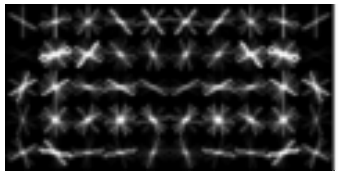
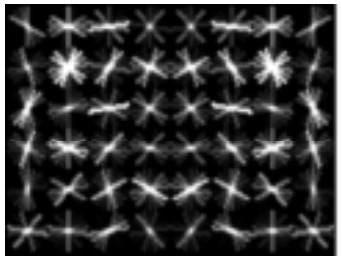
root filters
coarse resolution

part filters
finer resolution

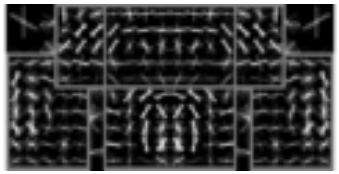
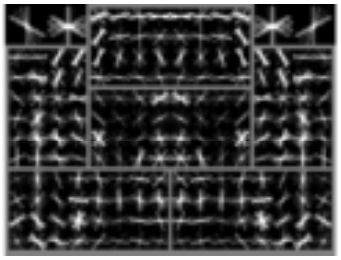
deformation
models

Handles partial occlusion/truncation

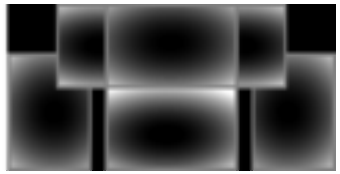
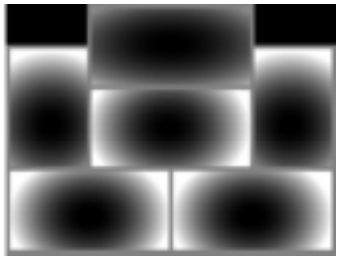
Car Model

root filters
coarse resolution

part filters
finer resolution

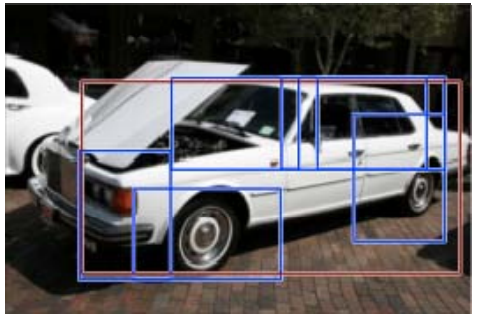



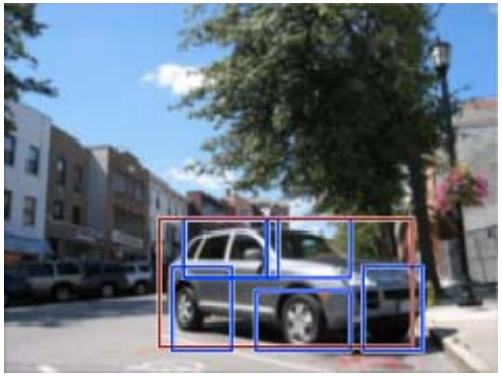
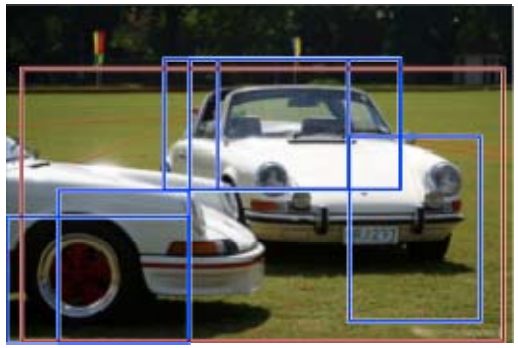
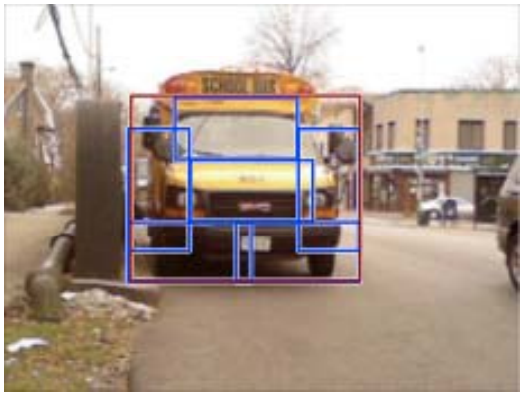
deformation
models

Car Detections

high scoring true positives

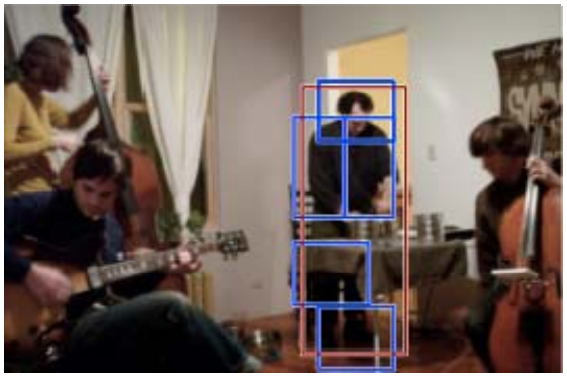
high scoring false positives

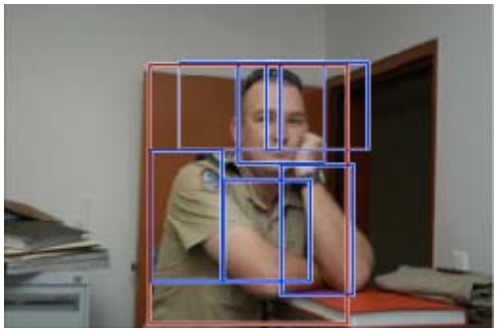



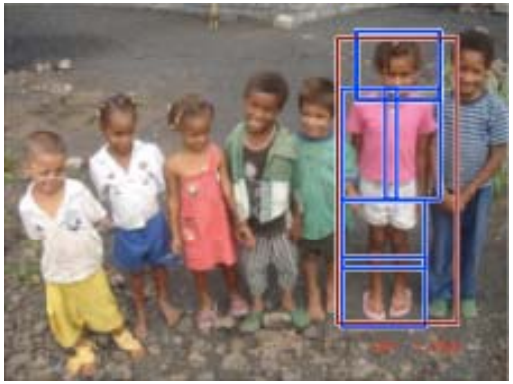




Person Detections

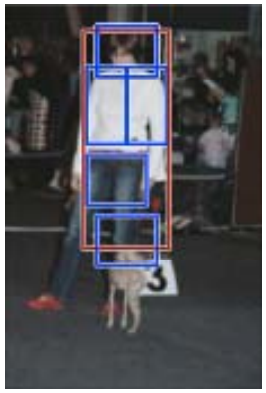
high scoring true positives



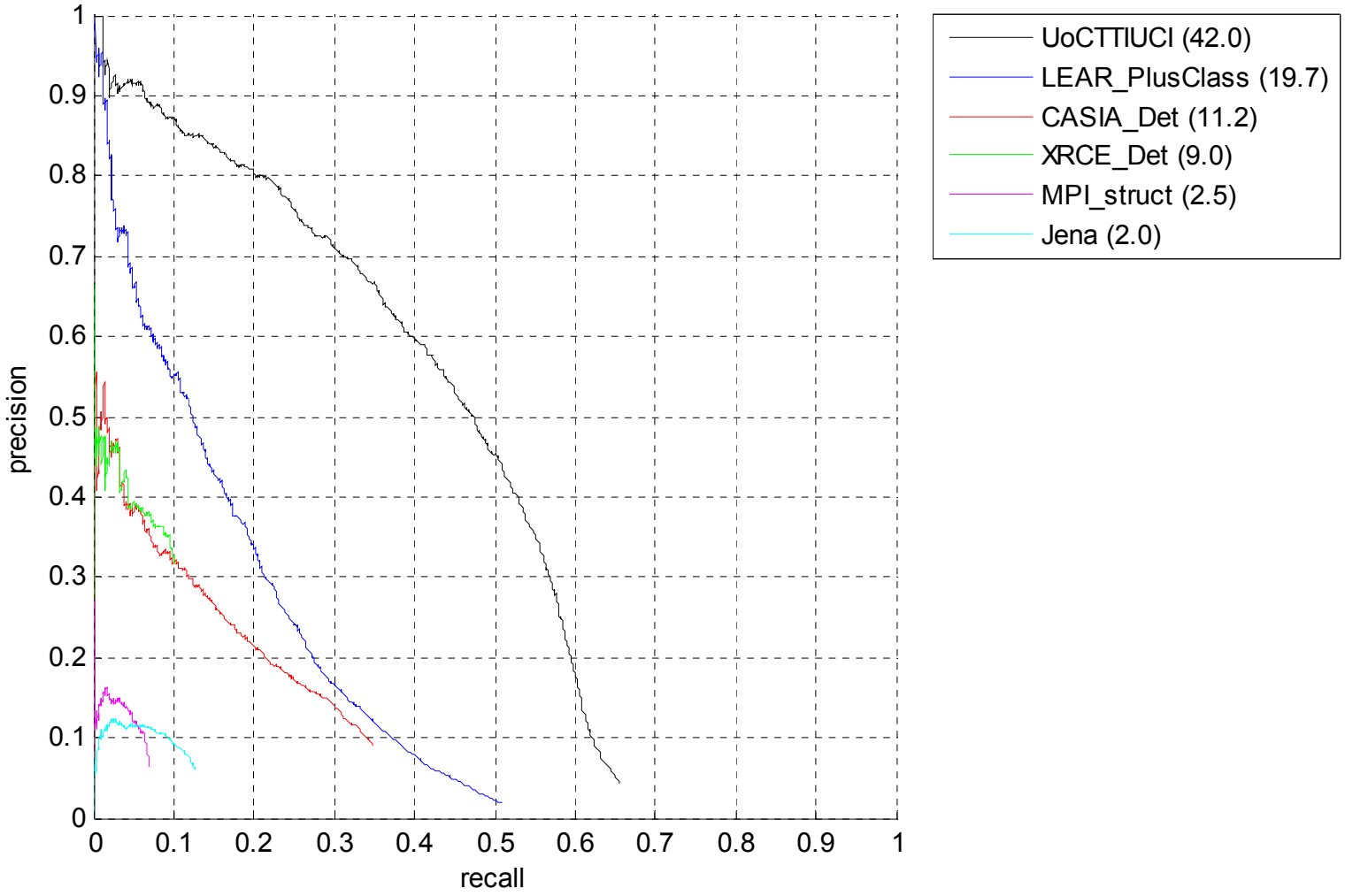




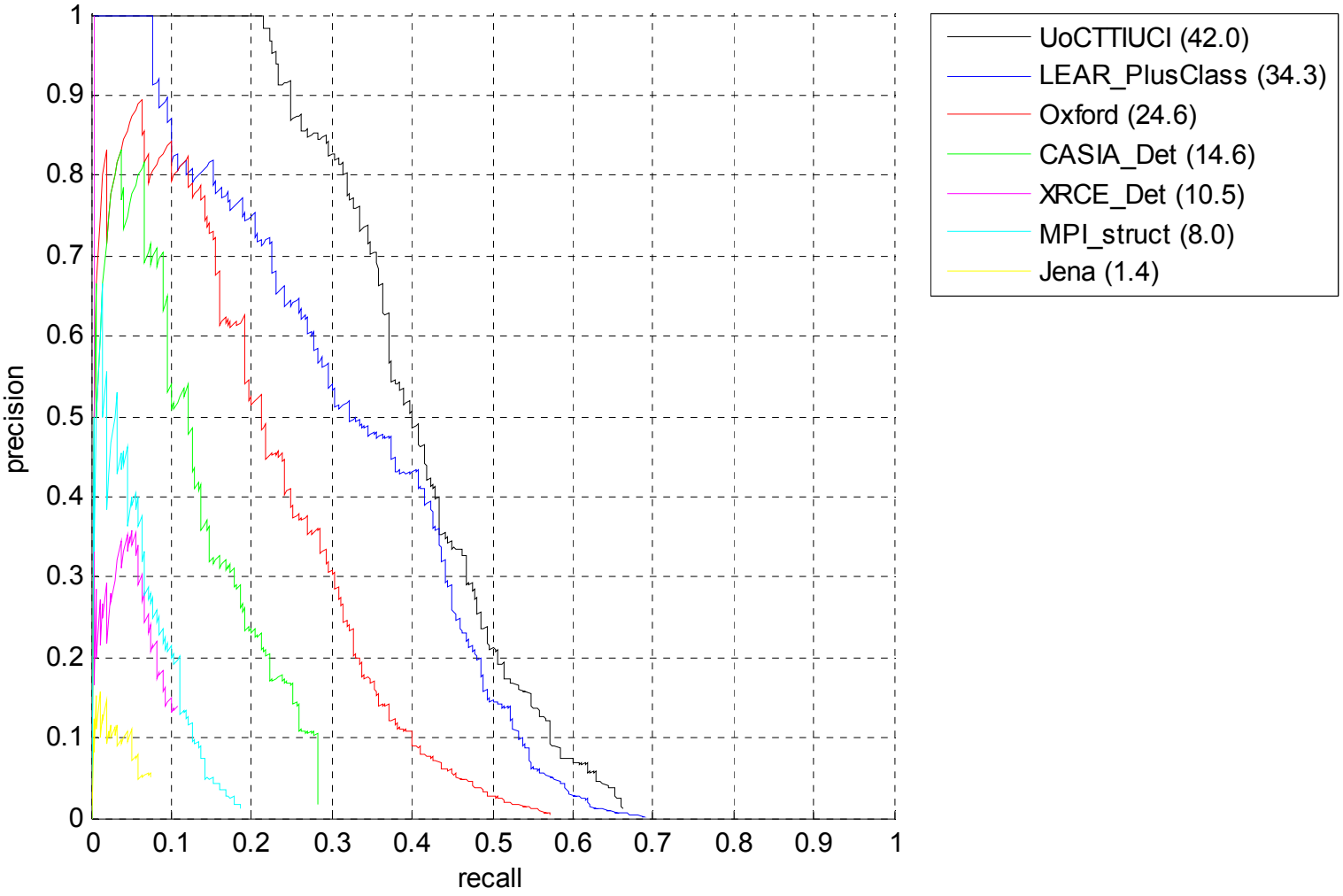
high scoring false positives
(not enough overlap)



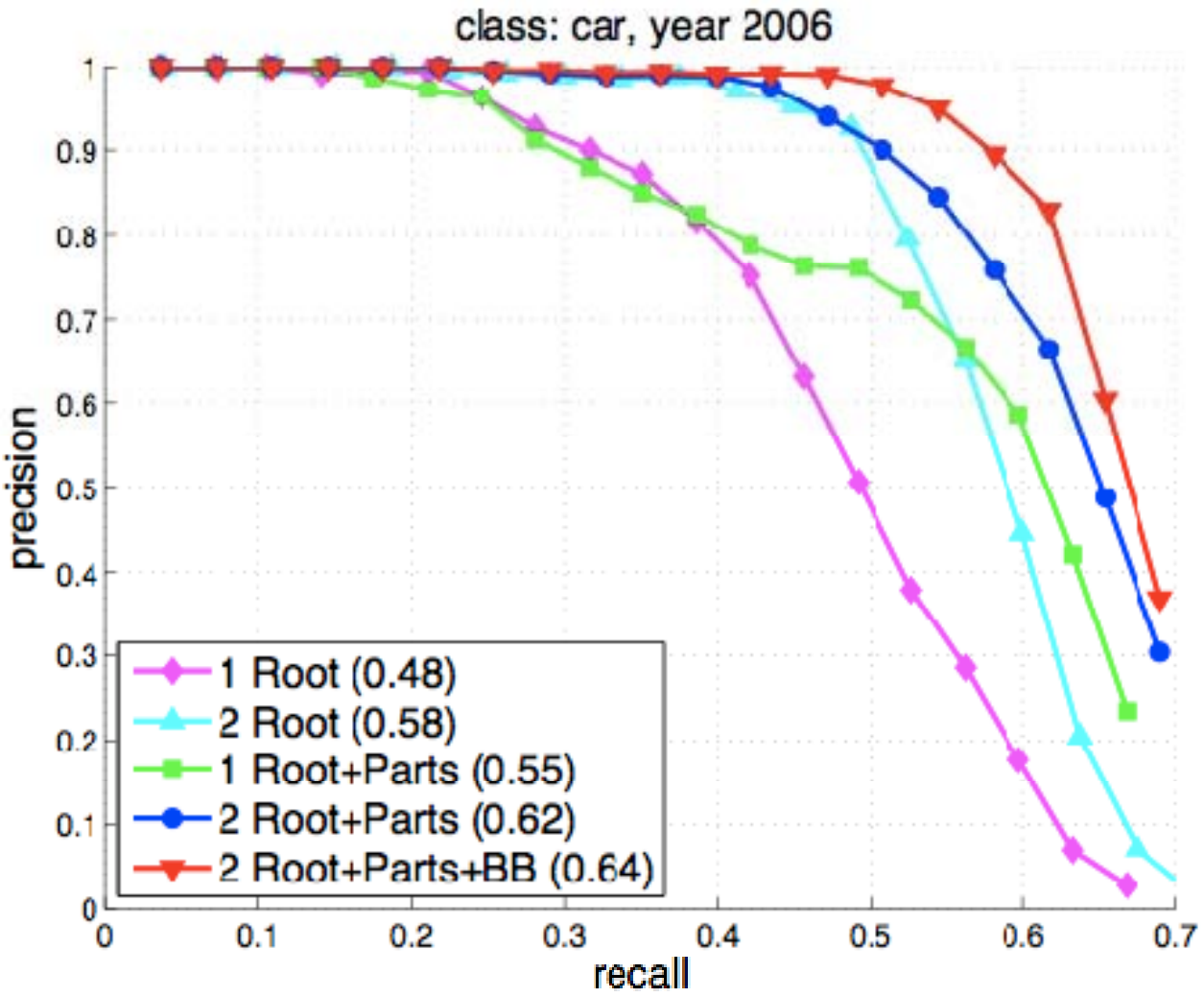
Precision/Recall: VOC2008 Person



Precision/Recall: VOC2008 Bicycle

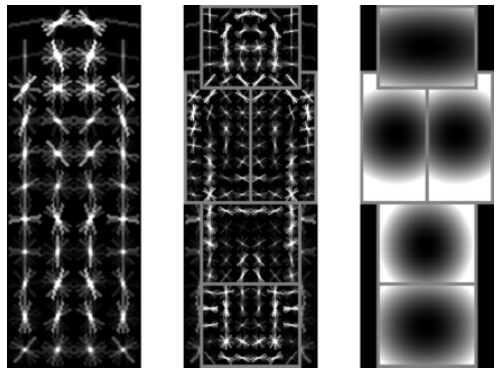


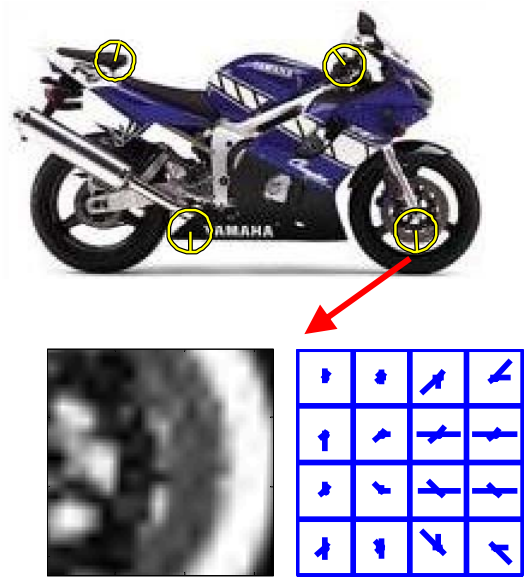
Comparison of Models

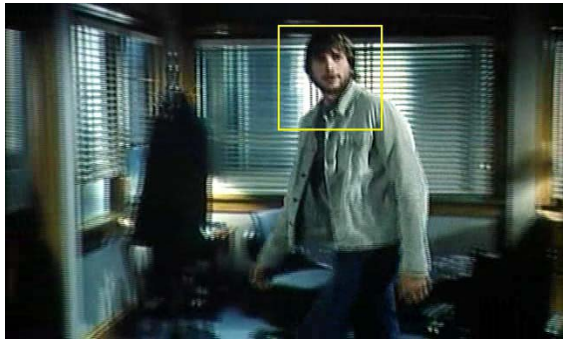


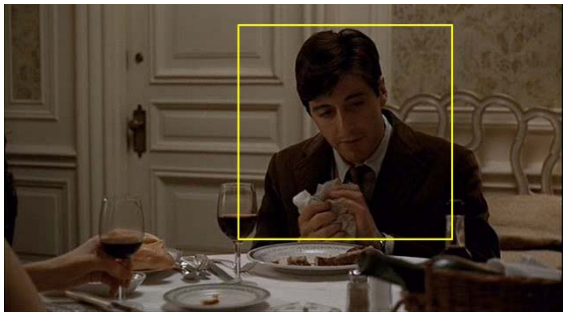
Summary

- Multiple features and multiple kernels boost performance
- Discriminative learning of model with latent variables for single feature (HOG):
  - Latent variables can learn best alignment in the ROI training annotation
  - Parts can be thought of as local SIFT vectors
  - Some similarities to Implicit Shape Model/Constellation models but with discriminative/careful training throughout









NB: Code available for latent model !

Outline

1. Sliding window detectors

2. Features and adding spatial information

3. HOG + linear SVM classifier

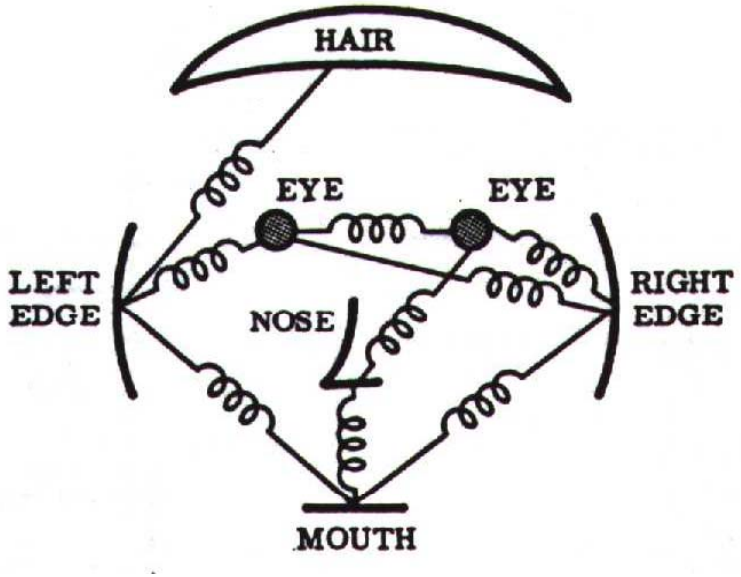
4. Two state of the art algorithms and PASCAL VOC

5. The future and challenges

Current Research Challenges

- Context
  - from scene properties: GIST, BoW, stuff
  - from other objects
  - from geometry of scene, e.g. Hoiem et al CVPR 06
- Occlusion/truncation
  - Winn & Shotton, Layout Consistent Random Field, CVPR 06
  - Vedaldi & Zisserman, NIPS 09
  - Yang et al, Layered Object Detection, CVPR 10
- 3D
- Scaling up – thousands of classes
  - Torralba et al, Feature sharing
  - ImageNet
- Weak and noisy supervision

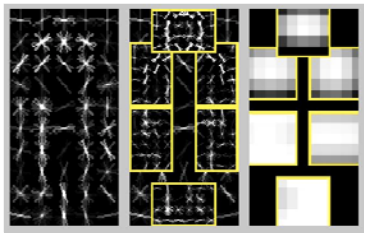
Pictorial structure model re-visited: efficient fitting

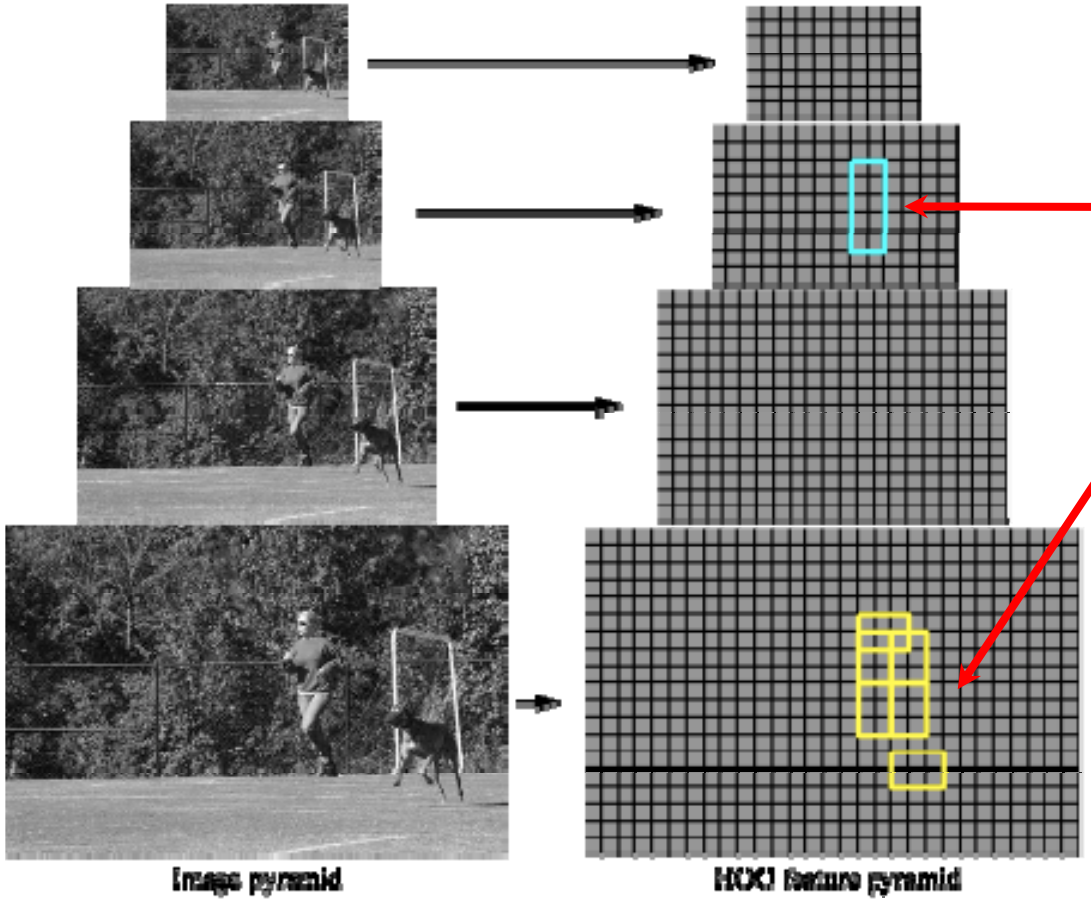


Let’s have a closer look at the LSVM deformable part-based model…

Object Hypothesis

- Position of root + each part
- Each part: HOG filter (at higher resolution)





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

p_0 : location of root

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

Score is sum of filter scores minus deformation costs

What is the cost of fitting the PS model?

- For fixed (learned) F_i and d_i
- For simplicity, consider only single scale of the pyramid
- Parts can appear anywhere in the image (h=number of pixels)

Appearance 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

displacements

deformation parameters

p_0 : location of root

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

$p_i = (x_i, y_i)$

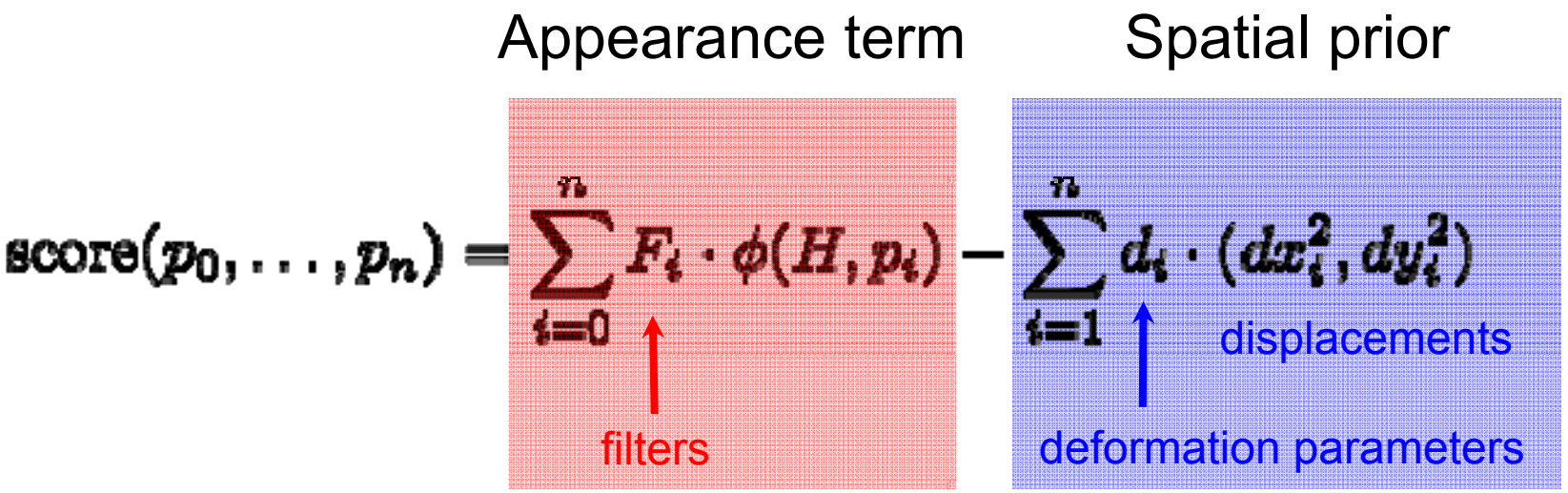
$dx_i = x_i - x_0$

$dy_i = y_i - y_0$

Fitting cost: Naïve search is $O(nh^2)$

What is the cost of fitting the PS model?

- For fixed (learned) F_i and d_i
- For simplicity, consider only single scale of the pyramid
- Parts can appear anywhere in the image (h=number of pixels)



Fitting cost: Naïve search is $O(nh^2)$

Need to evaluate the deformation cost of each part with respect to the root.

Can be done in O(nh)

Special case of a more general problem

Appearance term Spatial prior

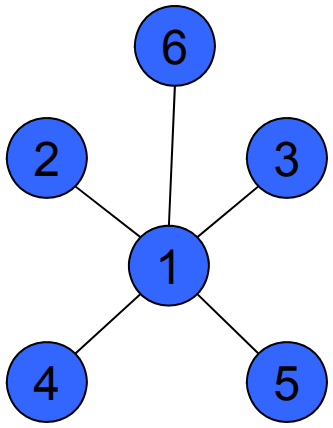
$$\text{score}(p_0, \ldots, p_n) = \sum_{i=0}^{n} F_i \cdot \phi(H, p_i) - \sum_{i=1}^{n} d_i \cdot (dx_i^2, dy_i^2)$$

filters

displacements

deformation parameters

Maximization of the PS score can be re-written as a **minimization** of the following cost function on a "star" graph:



$$f(\mathbf{x}) = \sum_{v_i \in V} m_i(v_i) + \sum_{e_{ij} \in E} \phi(v_i, v_j)$$

- Graph (V, E)
- Vertices v_i for $i = 1, \ldots, n$
- Edges e_{ij} connect v_i to other vertices v_j

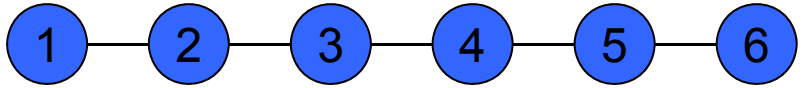
Dynamic programming on graphs

- Graph (V, E)
- Vertices v_i for $i = 1, \ldots, n$
- Edges e_{ij} connect v_i to other vertices v_j

$$f(\mathbf{x}) = \sum_{v_i \in V} m_i(v_i) + \sum_{e_{ij} \in E} \phi(v_i, v_j)$$

Dynamic programming - review

- Discrete optimization
- Each variable x has a finite number of possible states
- Applies to problems that can be decomposed into a sequence of stages
- Each stage expressed in terms of results of fixed number of previous stages
- The cost function need not be convex
- The name “dynamic” is historical
- Also called the “Viterbi” algorithm
- Let’s first consider a chain:

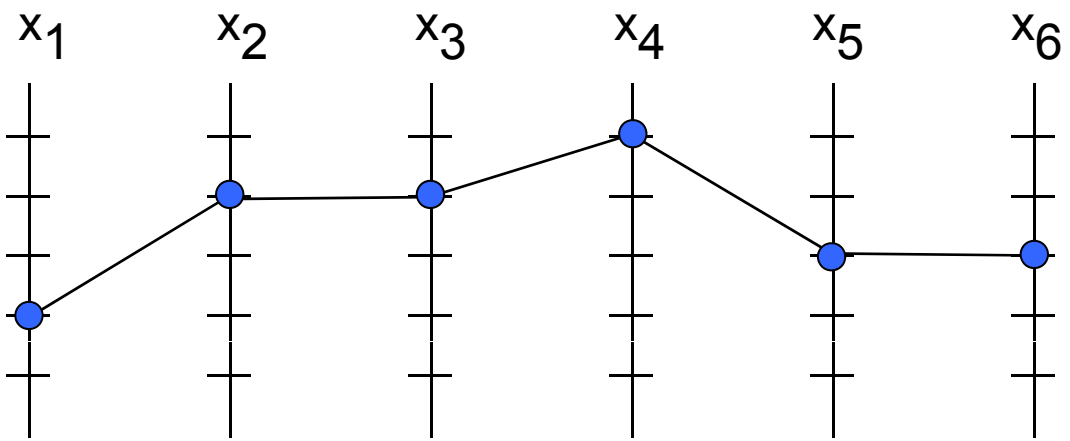


Consider a cost function $f(\mathbf{x}) : \mathbb{R}^n \rightarrow \mathbb{R}$ of the form

$$f(\mathbf{x}) = \sum_{i=1}^{n} m_i(x_i) + \sum_{i=2}^{n} \phi_i(x_{i-1}, x_i)$$

where x_i can take one of h values

e.g. h=5, n=6

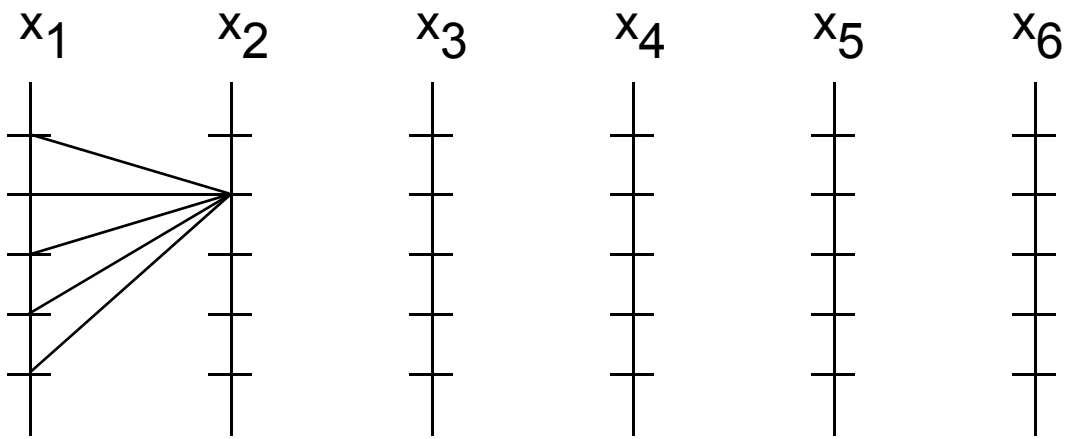


$$f(\mathbf{x}) = \begin{cases} m_1(x_1) + m_2(x_2) + m_3(x_3) + m_4(x_4) + m_5(x_5) + m_6(x_6) \\ \phi(x_1, x_2) + \phi(x_2, x_3) + \phi(x_3, x_4) + \phi(x_4, x_5) + \phi(x_5, x_6) \end{cases}$$

Complexity of minimization:

- exhaustive search $O(h^n)$
- dynamic programming $O(nh^2)$

$$f(\mathbf{x}) = \sum_{i=1}^{n} m_i(x_i) + \sum_{i=2}^{n} \phi(x_{i-1}, x_i)$$



Key idea: the optimization can be broken down into n sub-optimizations

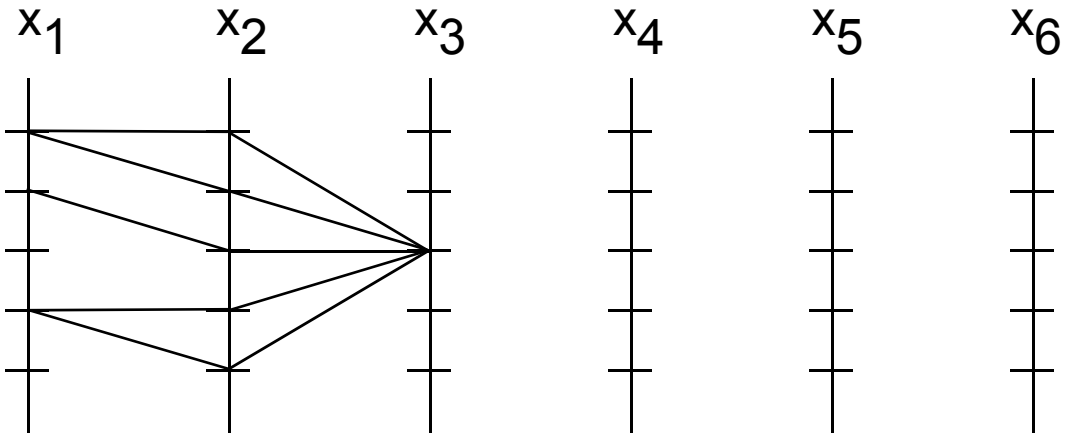
Step 1: For each value of x_2 determine the best value of x_1

- Compute

$$\begin{aligned} S_2(x_2) &= \min_{x_1}\{m_2(x_2) + m_1(x_1) + \phi(x_1, x_2)\} \\ &= m_2(x_2) + \min_{x_1}\{m_1(x_1) + \phi(x_1, x_2)\} \end{aligned}$$

- Record the value of x_1 for which $S_2(x_2)$ is a minimum

To compute this minimum for all x_2 involves $O(h^2)$ operations



Step 2: For each value of x_3 determine the best value of x_2 **and** x_1

- Compute

$$S_3(x_3) = m_3(x_3) + \min_{x_2}\{S_2(x_2) + \phi(x_2, x_3)\}$$

- Record the value of x_2 for which $S_3(x_3)$ is a minimum

Again, to compute this minimum for all x_3 involves $O(h^2)$ operations Note $S_k(x_k)$ encodes the lowest cost partial sum for all nodes up to k which have the value x_k at node k, i.e.

$$S_k(x_k) = \min_{x_1, x_2, \ldots, x_k} \sum_{i=1}^{k} m_i(x_i) + \sum_{i=2}^{k} \phi(x_{i-1}, x_i)$$

Viterbi Algorithm

- Initialize $S_1(x_1) = m_1(x_1)$
- For $k = 2 : n$

$$S_k(x_k) = m_k(x_k) + \min_{x_{k-1}} \{S_{k-1}(x_{k-1}) + \phi(x_{k-1}, x_k)\}$$
$$b_k(x_k) = \arg\min_{x_{k-1}} \{S_{k-1}(x_{k-1}) + \phi(x_{k-1}, x_k)\}$$

- Terminate

$$x_n^* = \arg\min_{x_n} S_n(x_n)$$

- Backtrack

$$x_{i-1} = b_i(x_i)$$

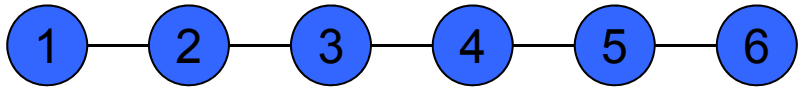
Complexity $O(nh^2)$

Dynamic programming on graphs

- Graph (V, E)
- Vertices v_i for $i = 1, \ldots, n$
- Edges e_{ij} connect v_i to other vertices v_j

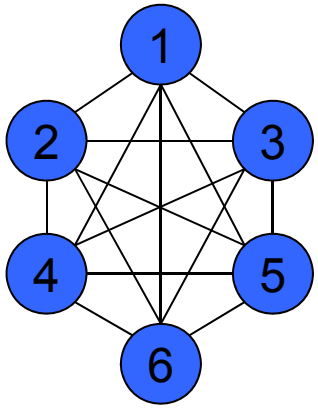
$$f(\mathbf{x}) = \sum_{v_i \in V} m_i(v_i) + \sum_{e_{ij} \in E} \phi(v_i, v_j)$$

So far have considered chains



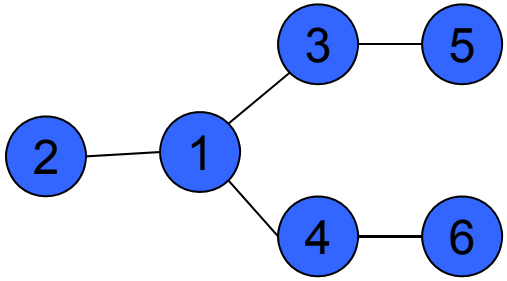
Different graph structures

Can use dynamic programming



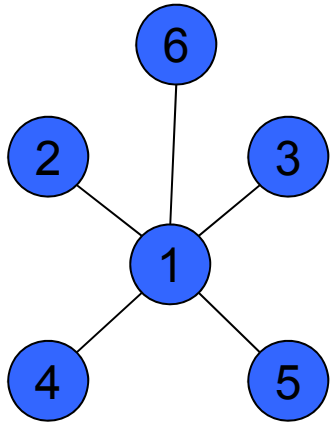
Fully connected

$O(h^n)$



Tree structure

$O(nh^2)$



Star structure

$O(nh^2)$

n parts

h positions (e.g. every pixel for translation)

Coming back to fitting pictorial structures

Appearance term Spatial prior

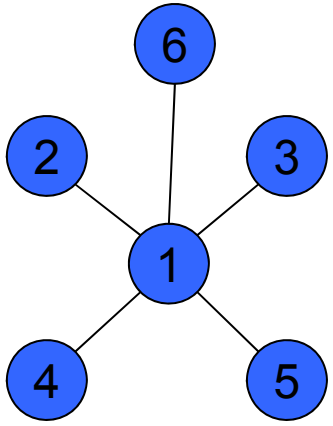
$$\text{score}(p_0, \ldots, p_n) = \sum_{i=0}^{n} F_i \cdot \phi(H, p_i) - \sum_{i=1}^{n} d_i \cdot (dx_i^2, dy_i^2)$$

filters

displacements

deformation parameters

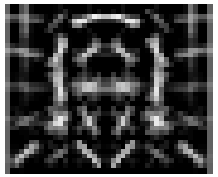
Maximization of the PS score can be re-written as a **minimization** of the following cost function on a "star" graph:



$$f(\mathbf{x}) = \sum_{v_i \in V} m_i(v_i) + \sum_{e_{ij} \in E} \phi(v_i, v_j)$$

As the spatial prior is a quadratic function of part positions, (x_i, y_i), finding the optimal configuration of parts **can be done in O(nh) time**, instead of naïve $O(nh^2)$.

Part Detection



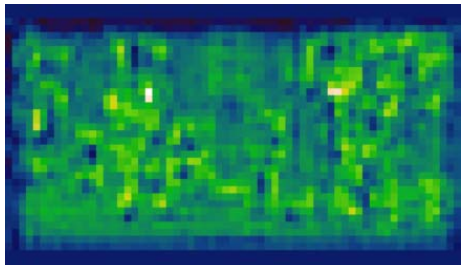
head filter

input image

Response of filter in l-th pyramid level

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

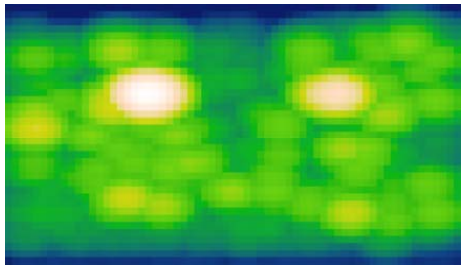
cross-correlation



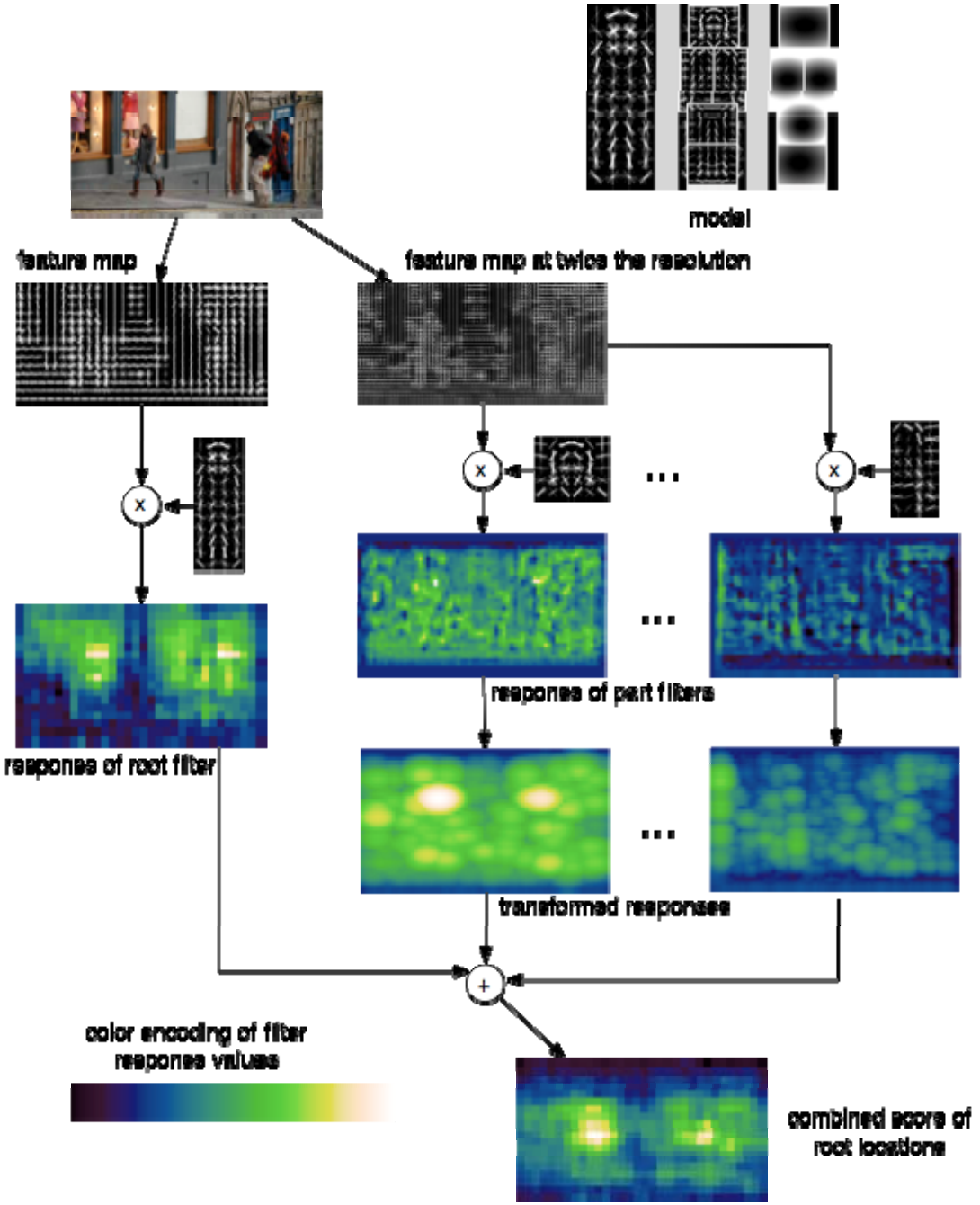
Transformed response

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

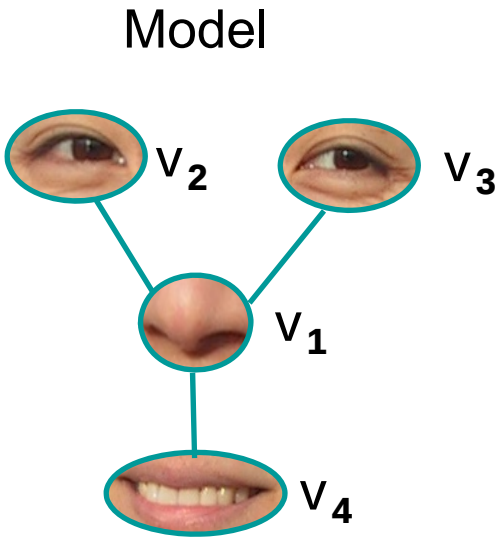
Distance transform computed in linear time
(spreading, local max, etc)



System

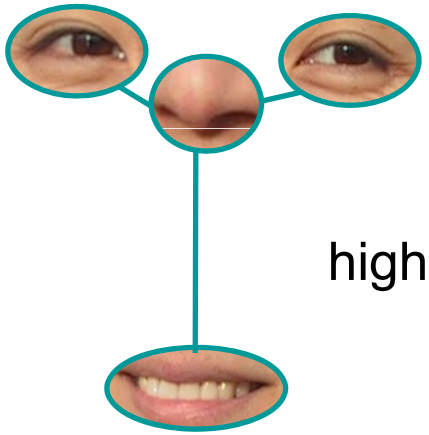

model
feature map
feature map at twice the resolution
response of root filter
response of part filters
transformed responses
color encoding of filter
response values
combined score of
root locations

Other applications of PS models: facial feature detection in images



The goal: Localize facial features in faces output by face detector

- Parts V= $\{v_1, \ldots v_n\}$
- Connected by springs in a star configuration to nose (can be a tree)
- Quadratic cost for springs



high spring cost

Example part localizations in video

Example of a model with 9 parts

Support parts-based face descriptors

Provide initialization for global face descriptors

Code available online: http://www.robots.ox.ac.uk/~vgg/research/nface/index.html

Summary

- Pictorial structure models with tree configuration of parts can be fitted in **$O(nh^2)$**. {n=number of parts, h=number of pixels}

- For quadratic pair-wise terms this can be reduced to **O(nh)**.

- This can lead to significant speed-ups if h is large (e.g. number of pixels).

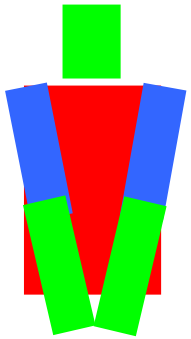
Other applications:

- Facial feature finding
- Fitting articulated models

Human Pose Estimation

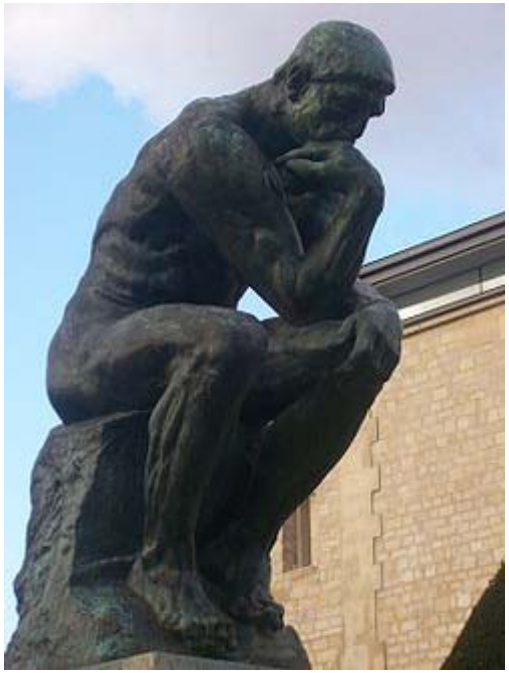
Objective and motivation

Determine human body pose (layout)



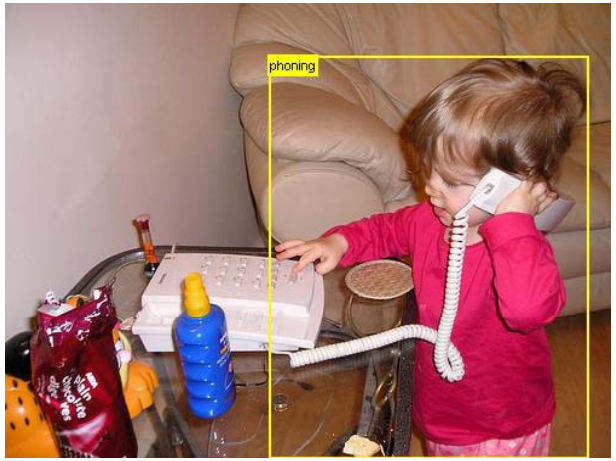
Why? To recognize poses, gestures, actions

Activities characterized by a pose



Activities characterized by a pose

Activities characterized by a pose

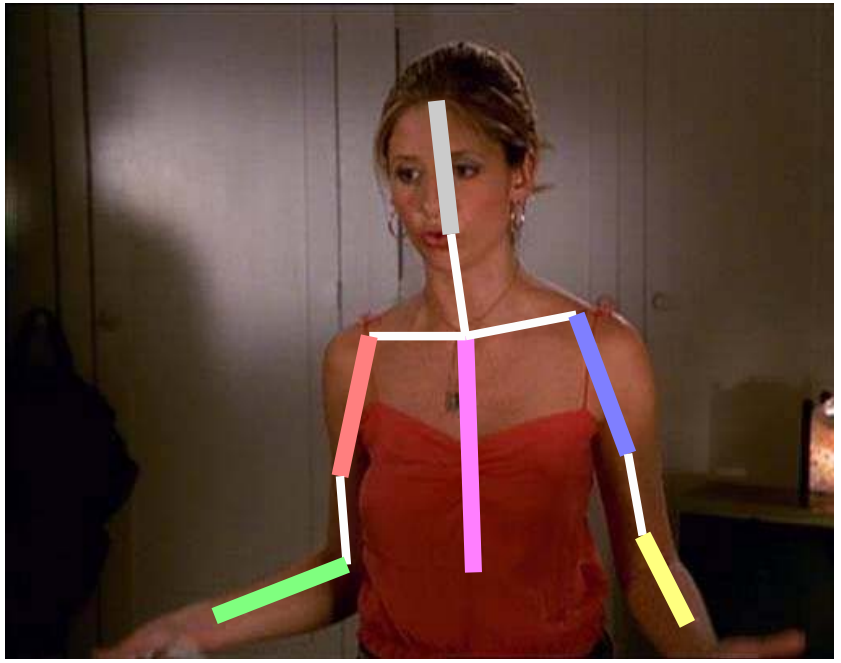


Challenges: articulations and deformations

Challenges: of (almost) unconstrained images




varying illumination and low contrast; moving camera and background;
multiple people; scale changes; extensive clutter; any clothing



Outline

Review of pictorial structures for articulated models

Inference given the model: Strong supervision, full generative model – “Gold-standard model”

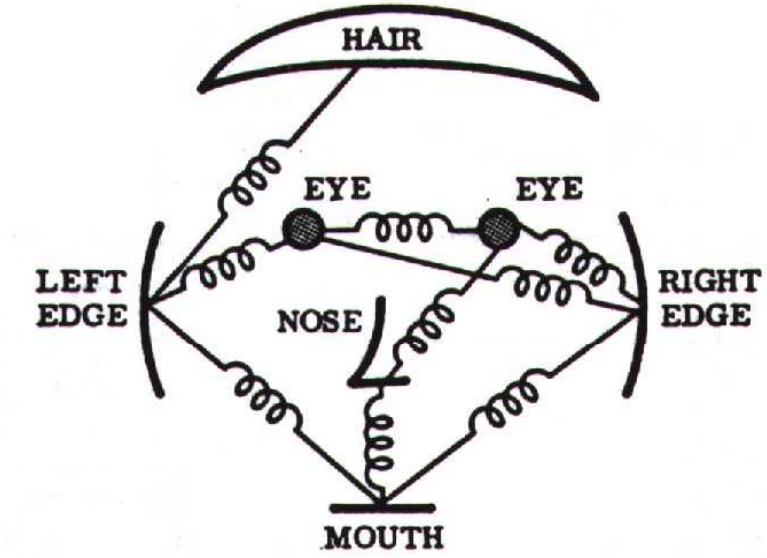
Image parsing: learning the model for a specific image

Recent advances

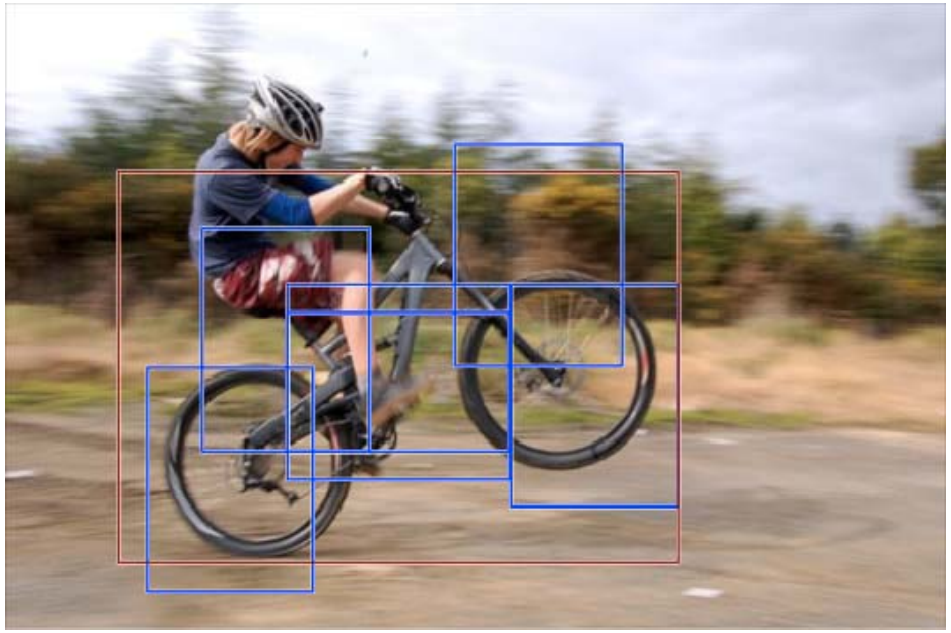
Datasets and challenges

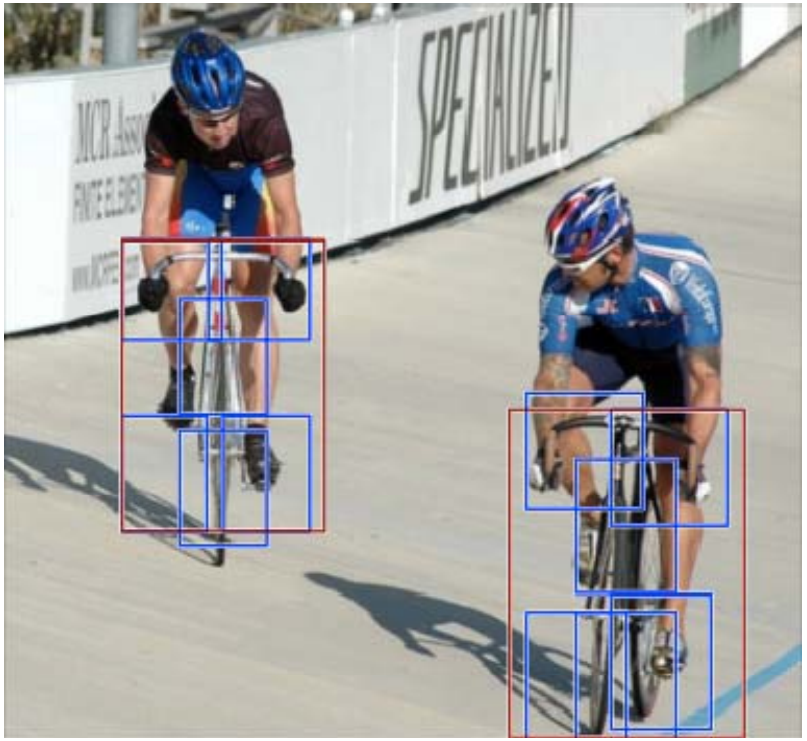
Pictorial Structures

- Intuitive model of an object
- Model has two components
  1. parts (2D image fragments)
  2. structure (configuration of parts)
- Dates back to Fischler & Elschlager 1973



From earlier: objects





Mixture of deformable part-based models

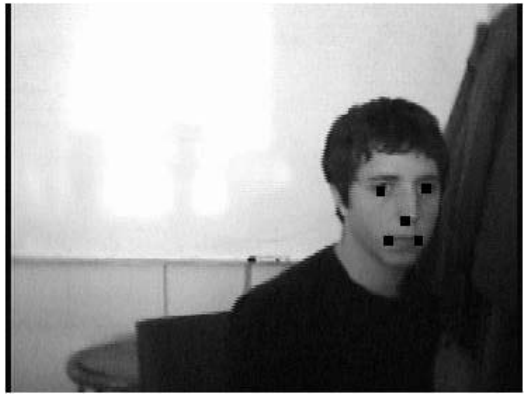
- One component per “aspect” e.g. front/side view

Each component has global template + deformable parts

Discriminative training from bounding boxes alone

Localize multi-part objects at arbitrary locations in an image

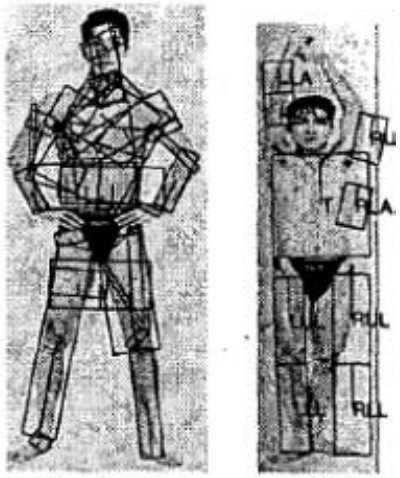
- Generic object models such as person or car
- Allow for articulated objects
- Simultaneous use of appearance and spatial information
- Provide efficient and practical algorithms



To fit model to image: minimize an energy (or cost) function that reflects both

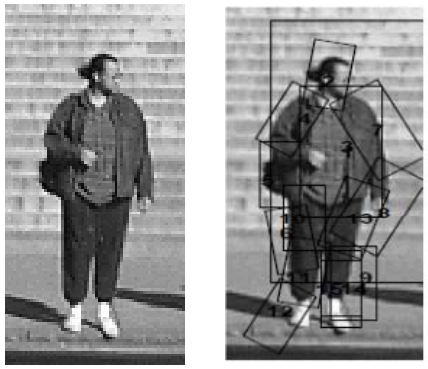
- Appearance: how well each part matches at given location
- Configuration: degree to which parts match 2D spatial layout

Long tradition of using pictorial structures for humans



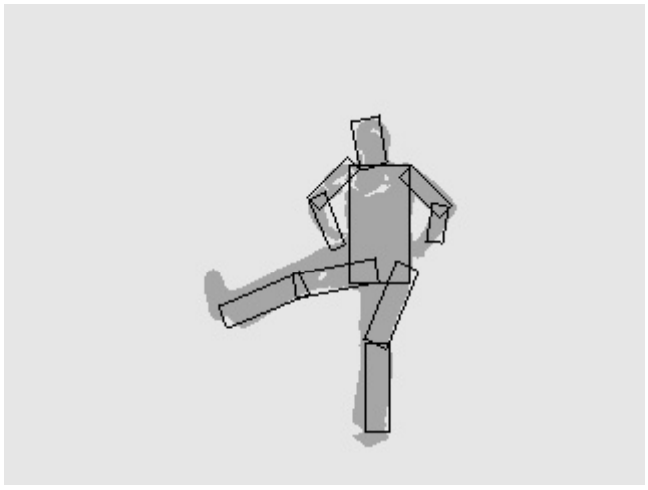
Finding People by Sampling
Ioffe & Forsyth, ICCV 1999

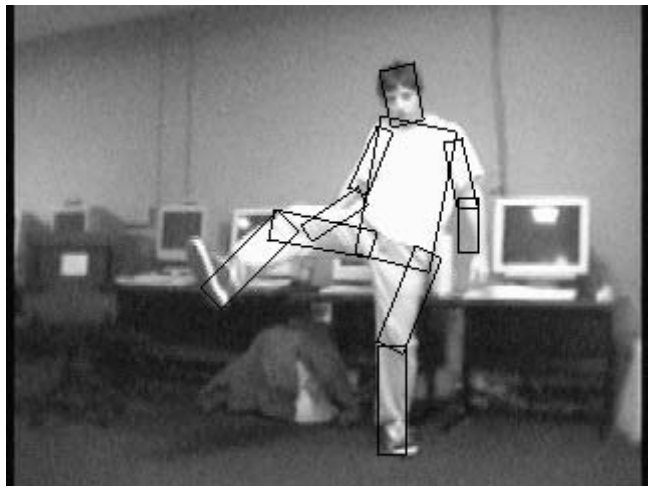
Pictorial Structure Models for Object Recognition
Felzenszwalb & Huttenlocher, 2000

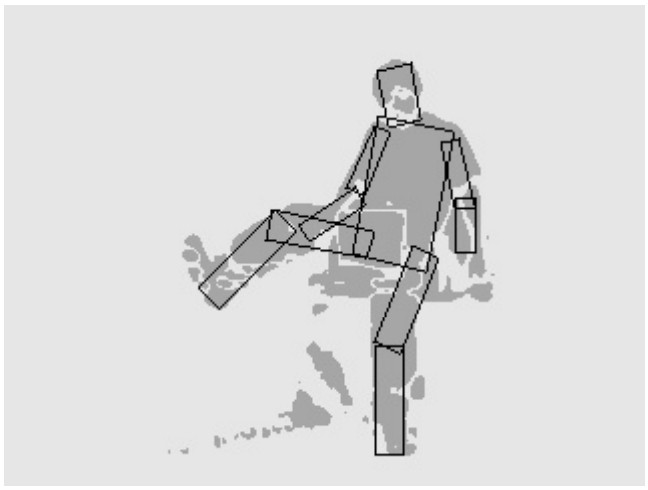


Learning to Parse Pictures of People
Ronfard, Schmid & Triggs, ECCV 2002

Felzenszwalb & Huttenlocher

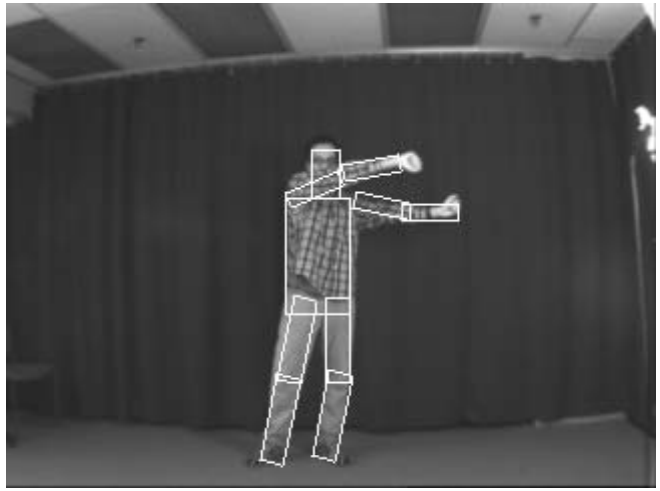
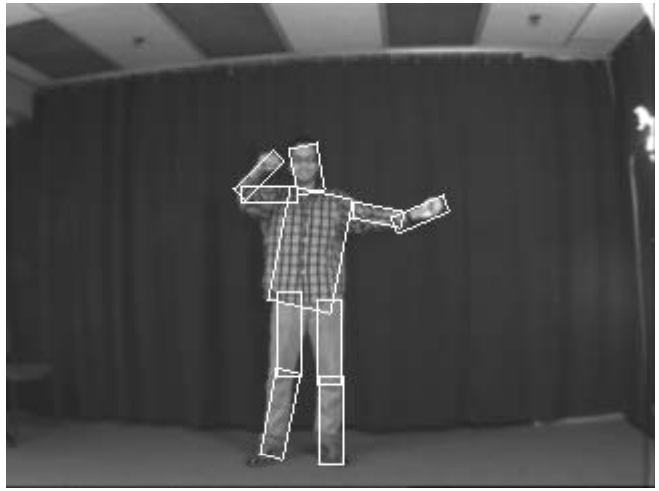
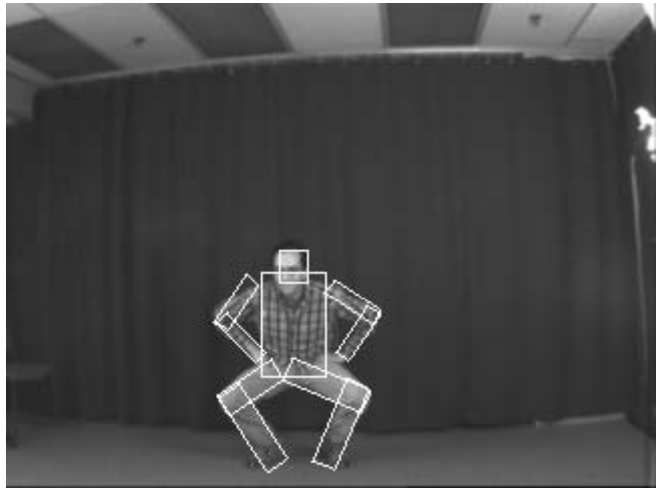
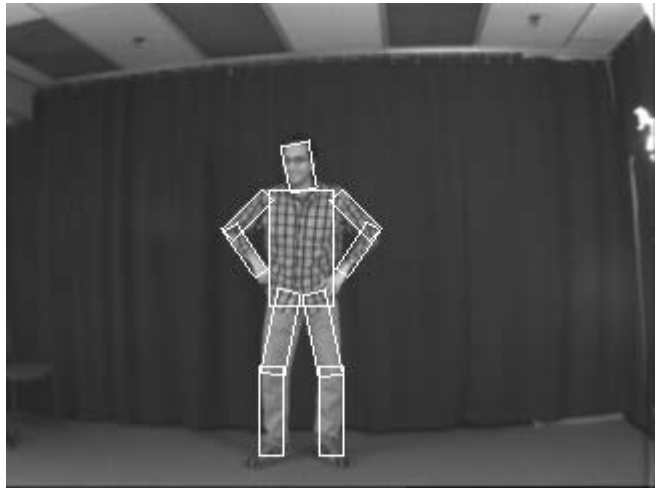




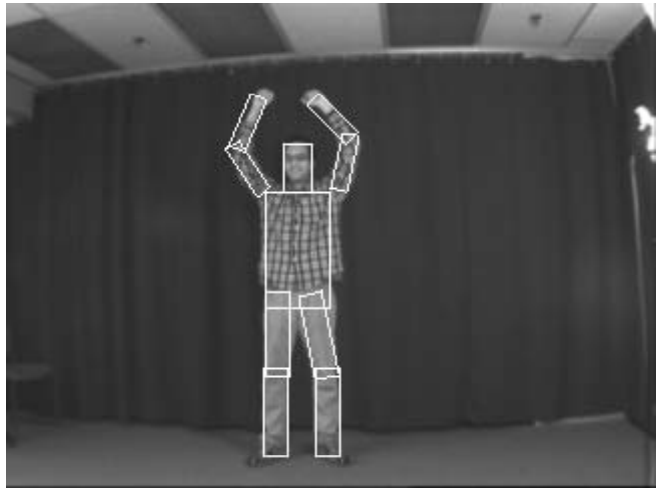
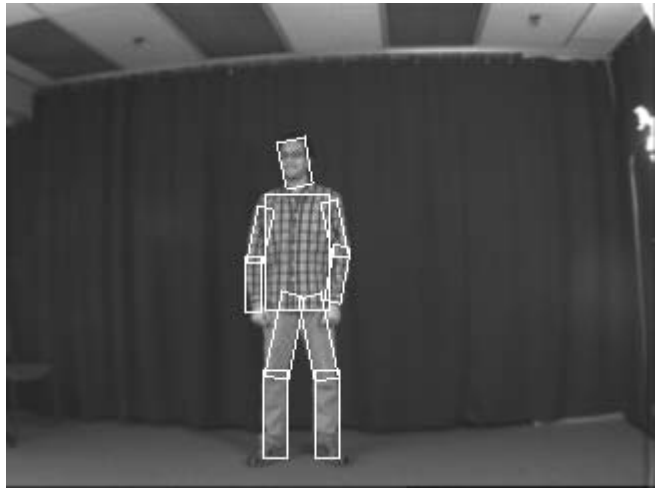
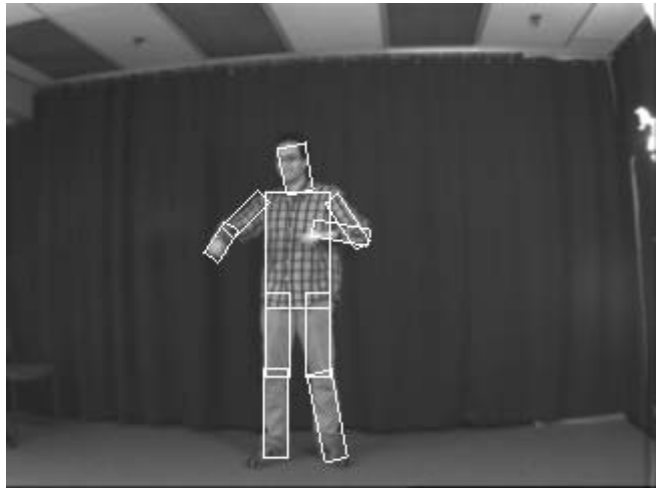
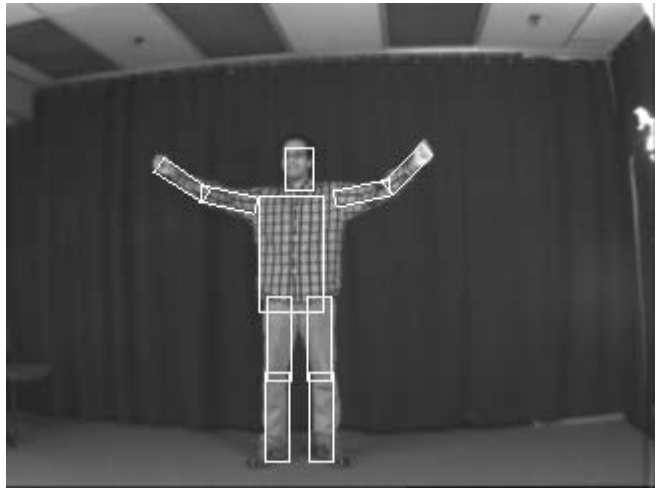


NB: requires background subtraction

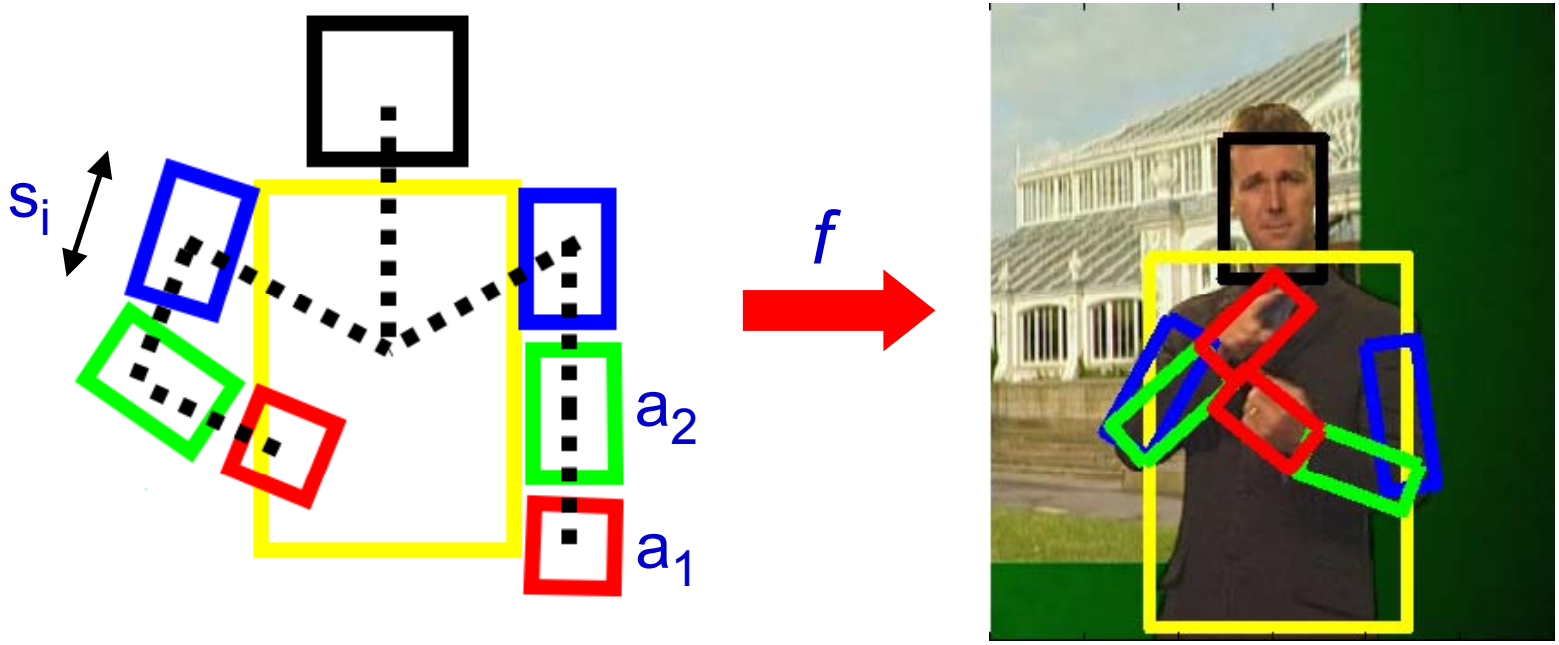
Variety of Poses

Variety of Poses

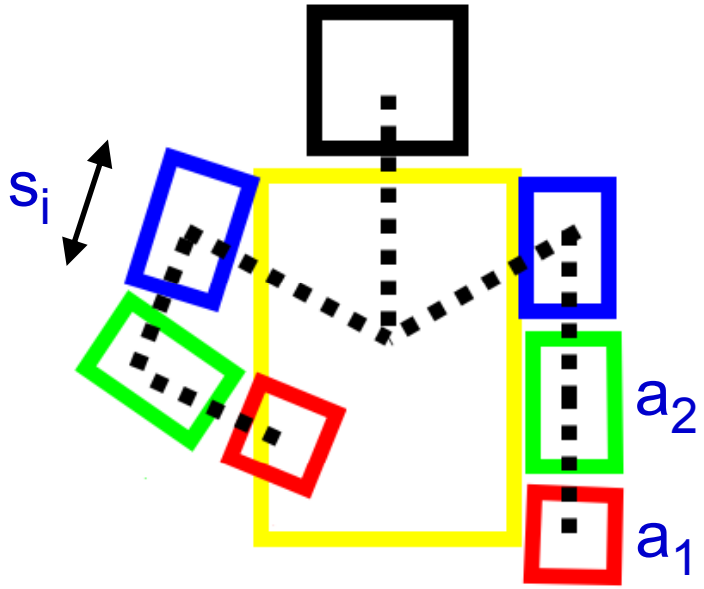
Objective: detect human and determine upper body pose (layout)



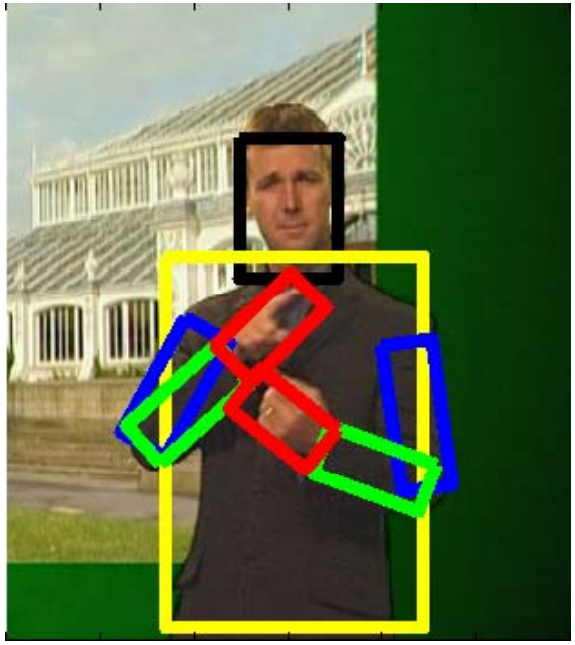
Model as a graph labelling problem

- Vertices $\mathcal{V}$ are parts, $a_i, i = 1, \cdots, n$
- Edges $\mathcal{E}$ are pairwise linkages between parts
- For each part there are h possible poses $\mathbf{p}_j = (x_j, y_j, \phi_j, s_j)$
- Label each part by its pose: $f : \mathcal{V} \longrightarrow \{1, \cdots, h\}$, i.e. part a takes pose $\mathbf{p}_{f(a)}$.

Pictorial structure model – CRF



f



- Each labelling has an energy (cost):

$$E(f) = \sum_{a \in \mathcal{V}} \theta_{a;f(a)} + \sum_{(a,b) \in \mathcal{E}} \theta_{ab;f(a)f(b)}$$

unary terms (appearance)

pairwise terms (configuration)

Features for unary:

- colour
- HOG

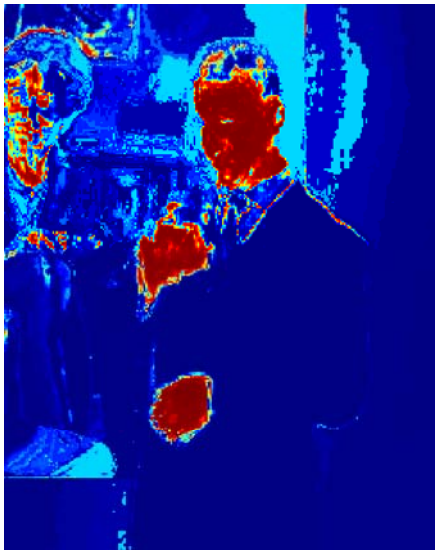
for limbs/torso

- Fit model (inference) as labelling with lowest energy

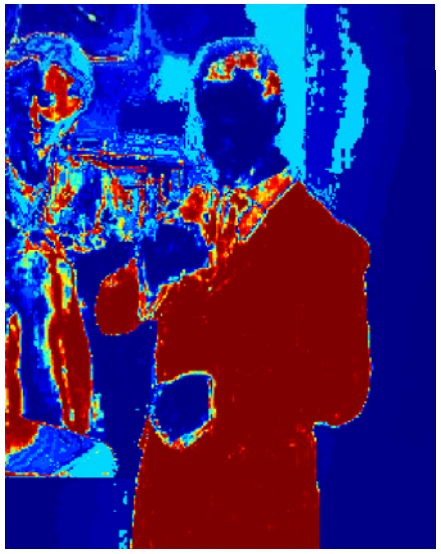
Unary term: appearance feature I - colour

input image

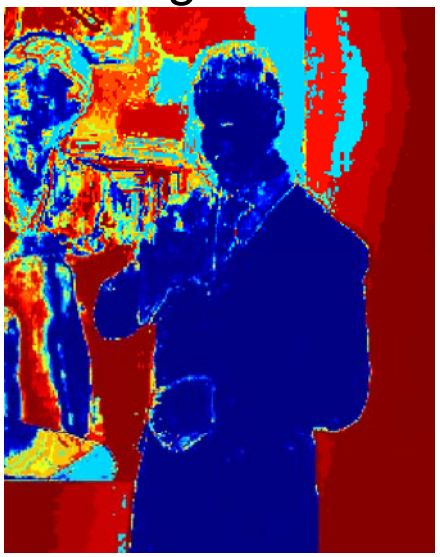
skin

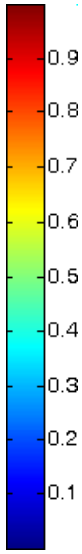


torso



background



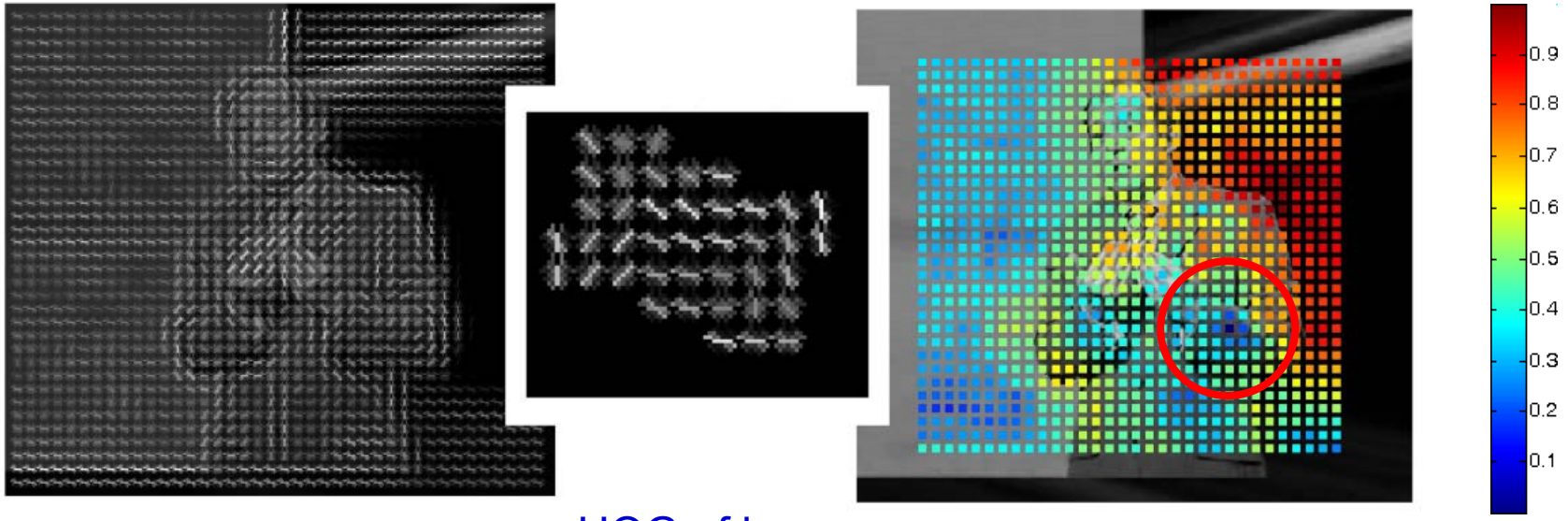


colour posteriors

Unary term: appearance feature II - HOG

Dalal & Triggs, CVPR 2005

Histogram of oriented gradients (HOG)



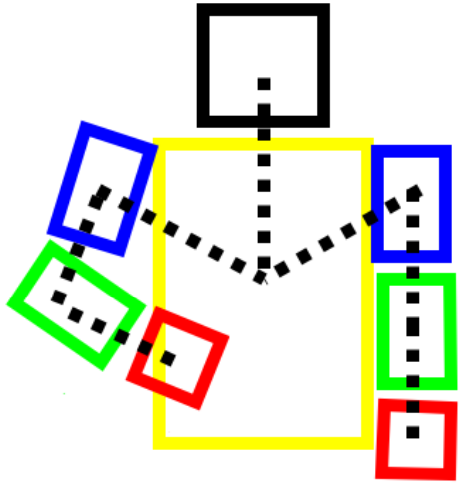
HOG of image

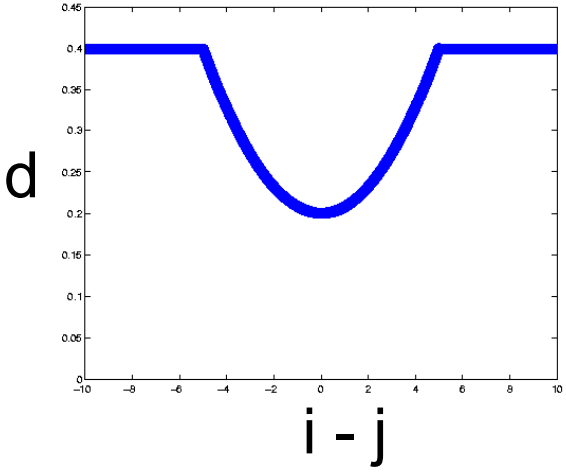
HOG of lower arm template (learned)

L2 Distance

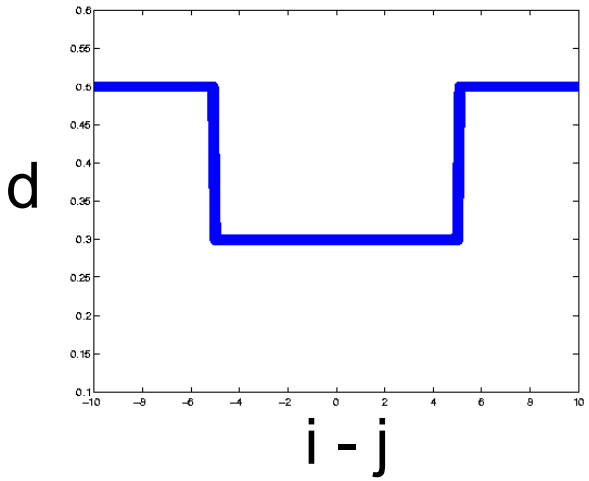
Pairwise terms: kinematic layout

$$\theta_{ab;ij} = w_{ab} d(|i-j|)$$



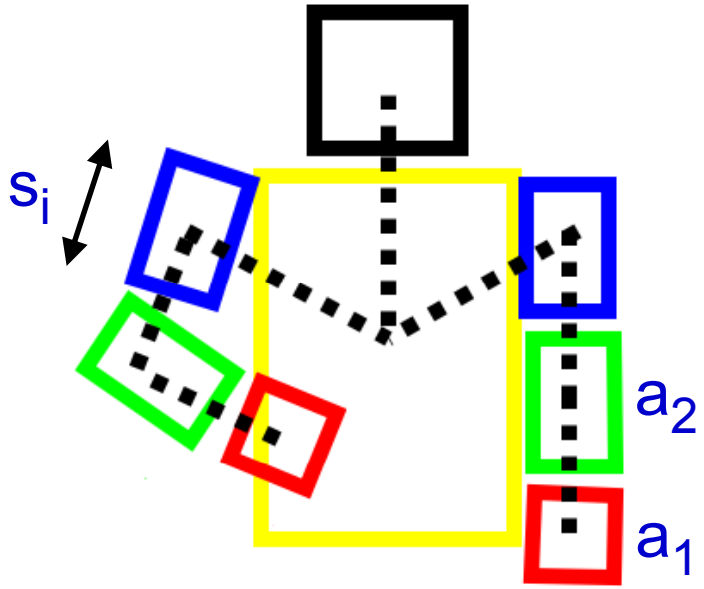


Truncated Quadratic

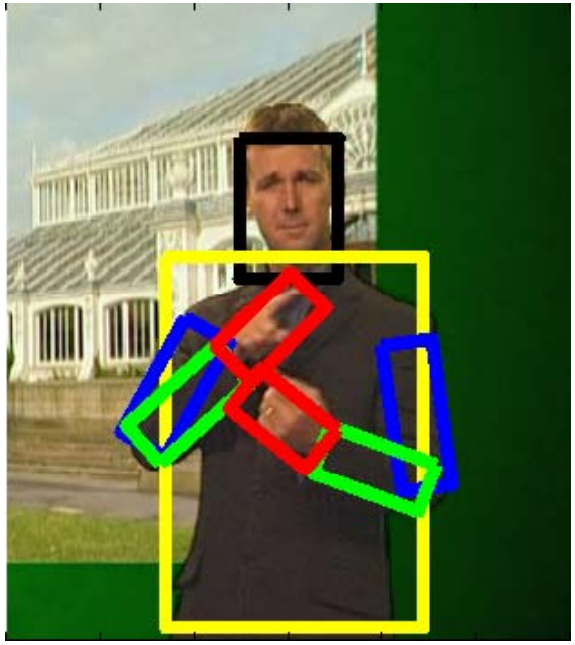


Potts

Pictorial structure model – CRF



f



- Each labelling has an energy (cost):

$$E(f) = \underbrace{\sum_{a \in \mathcal{V}} \theta_{a;f(a)}}_{\text{unary terms (appearance)}} + \underbrace{\sum_{(a,b) \in \mathcal{E}} \theta_{ab;f(a)f(b)}}_{\text{pairwise terms (configuration)}}$$

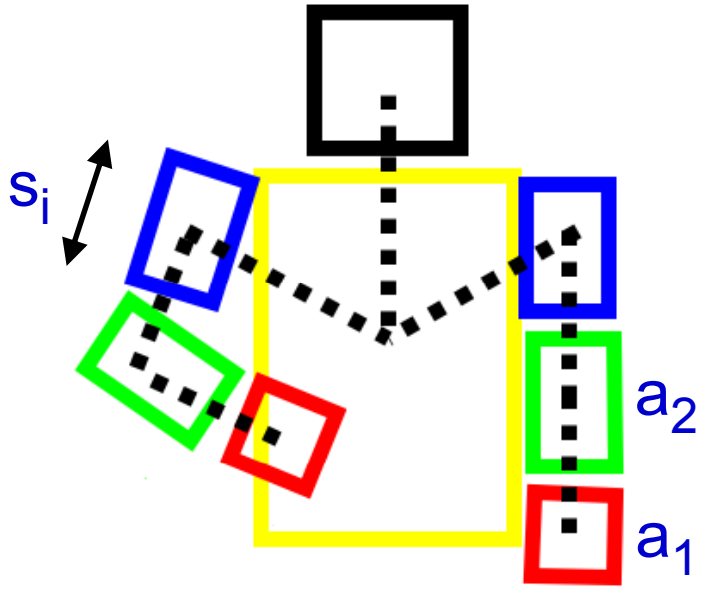
Features for unary:

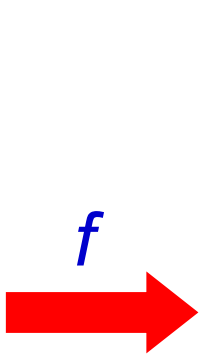
- colour
- HOG

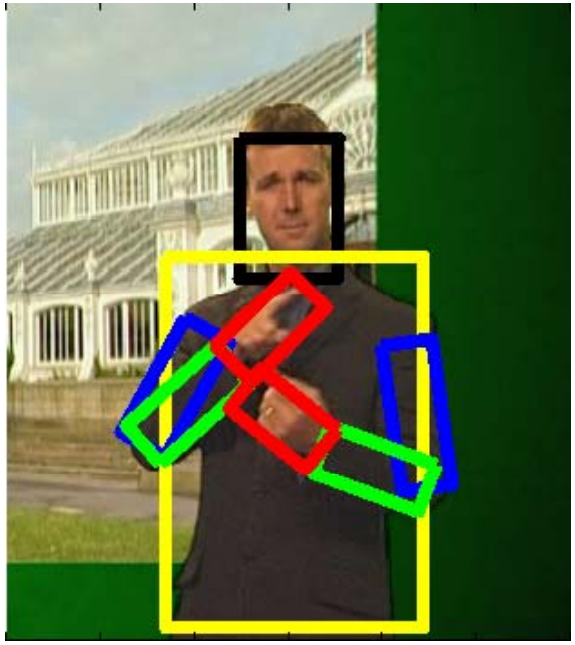
for limbs/torso

- Fit model (inference) as labelling with lowest energy

Complexity





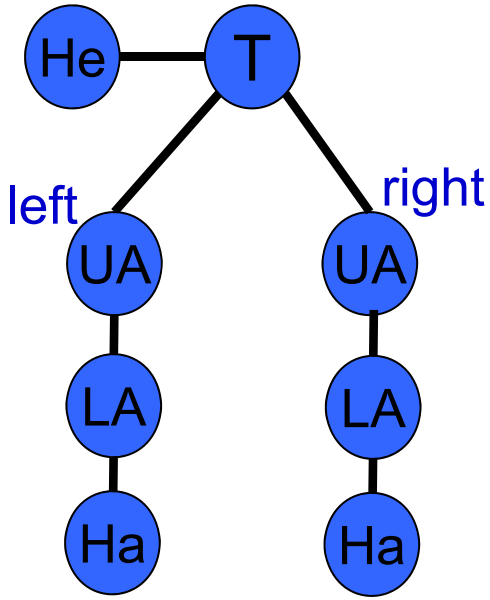


- n parts
- For each part there are h possible poses $\mathbf{p}_j = (x_j, y_j, \phi_j, s_j)$
- There are h^n possible labellings

Problem: any reasonable discretization (e.g. 12 scales and 36 angles for upper and lower arm, etc) gives a number of configurations 10^12 – 10^14

→ Brute force search not feasible

Are trees the answer?



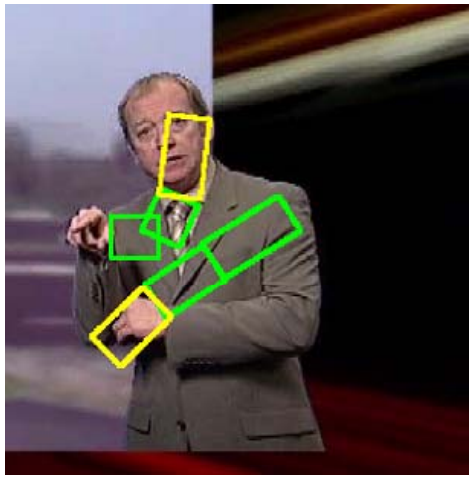
- With n parts and h possible discrete locations per part, $O(h^n)$
- For a tree, using dynamic programming this reduces to $O(nh^2)$
- If model is a tree and has certain edge costs, then complexity reduces to O(nh) using a distance transform [Felzenszwalb & Huttenlocher, 2000, 2005]

Problems with tree structured pictorial structures

- Layout model defines the foreground,

i.e. it chooses the pixels to “explain”

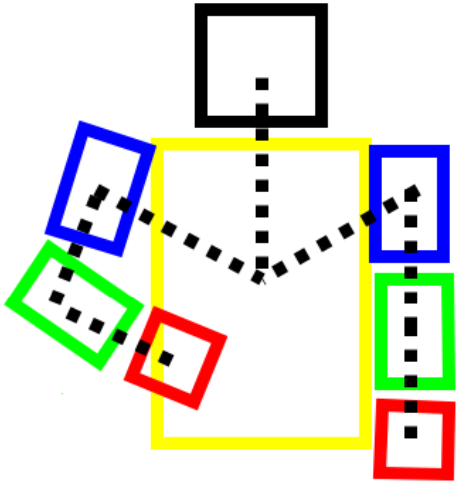
- ignores skin and strong edge in background



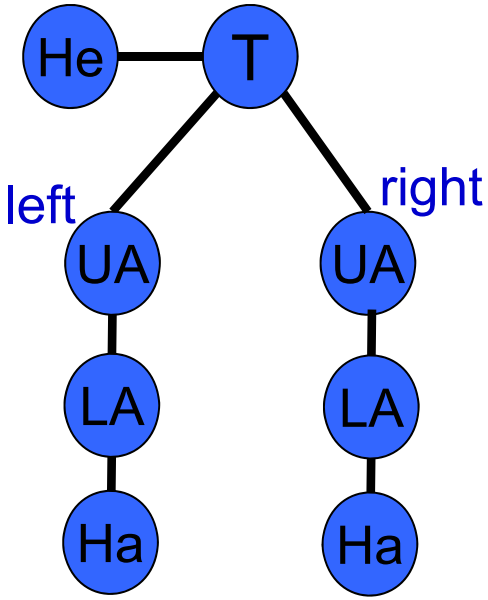
- “double counting”

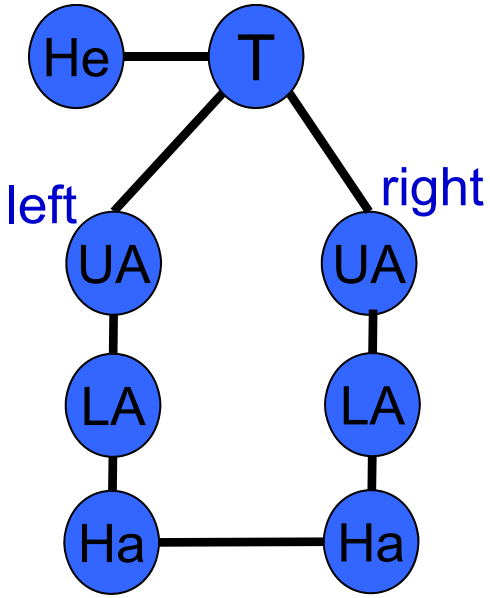
Generative model of foreground only

Kinematic structure vs graphical (independence) structure



Graph G = (V,E)





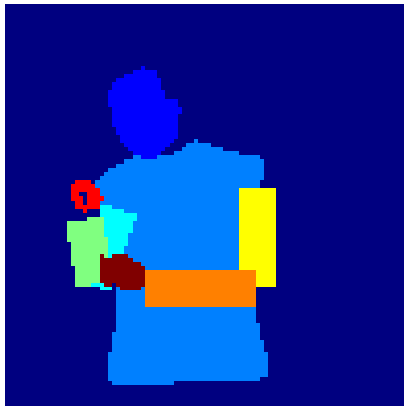
Requires more connections than a tree

And for the background problem

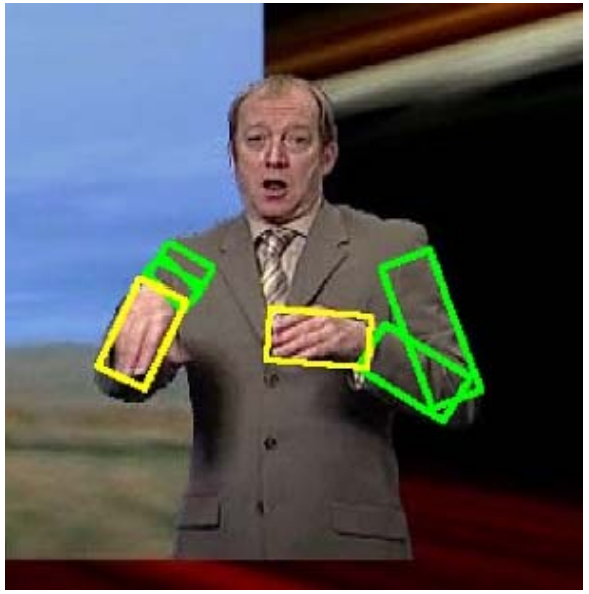
1. Add background model so that every pixel in region explained

$$E_{\text{full}} = E(f) + \sum_{\text{pixels } \mathbf{x}_i \text{ not in } f} E(\mathbf{x}_i|\text{bgcol})$$

2. *f* lays out parts in back-to-front depth order (painter's algorithm)



Colour is pixel-wise labelling
by parts (back-to-front)



Generative model of entire region

Outline

Review of pictorial structures for articulated models

Inference given the model: Strong supervision, full generative model – “Gold-standard model”

Image parsing: learning the model for a specific image

Recent advances

Datasets and challenges

Long Term Arm and Hand Tracking for Continuous Sign Language TV Broadcasts

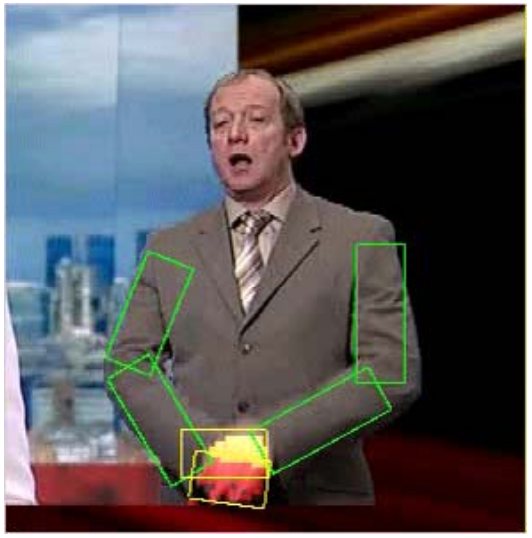
Patrick Buehler, Mark Everingham,

Daniel Huttenlocher, Andrew Zisserman

British Machine Vision Conference 2008

Objective

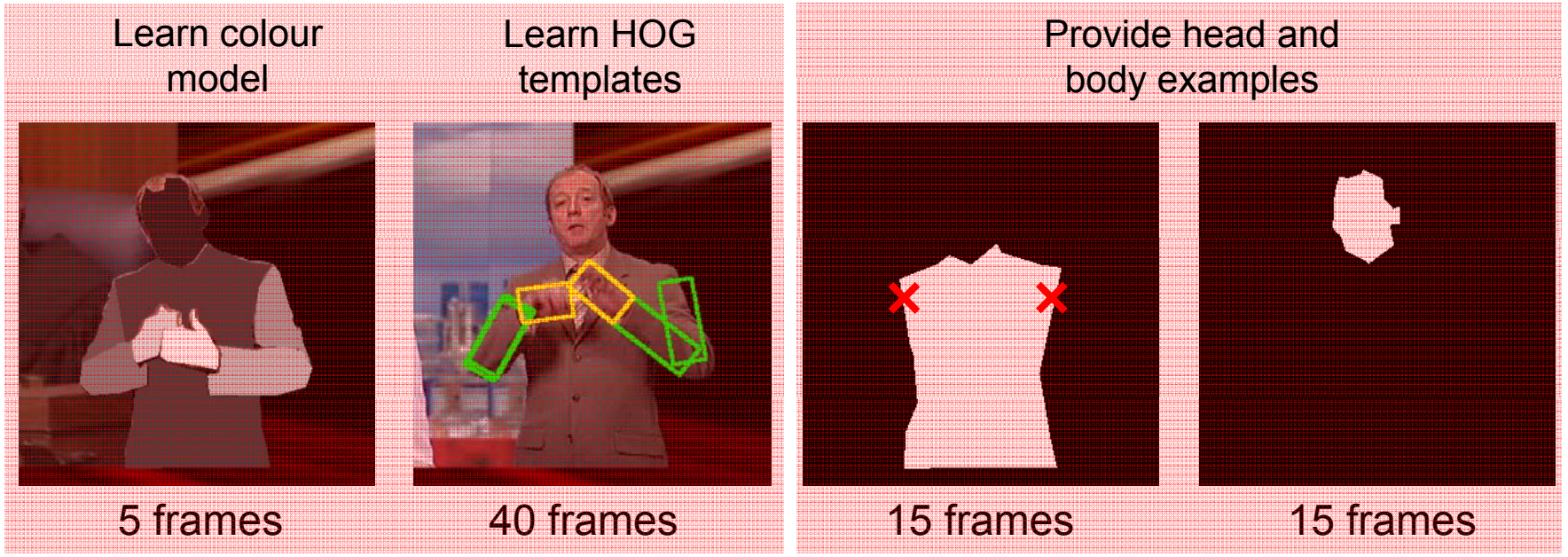
- Detect hands and arms of person signing British Sign Language
- Hour long sequences



- Strong but minimal supervision

Learning the model

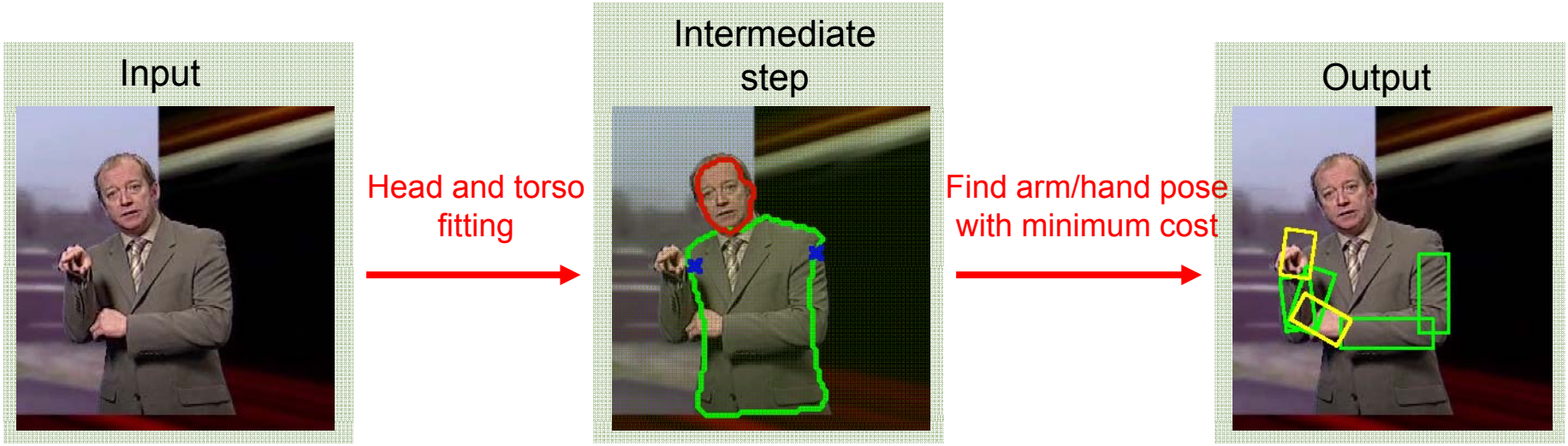
Strong supervision: manual input



40 annotated frames per video, used for pose estimation in > 50,000 frames

Inference (model fitting)

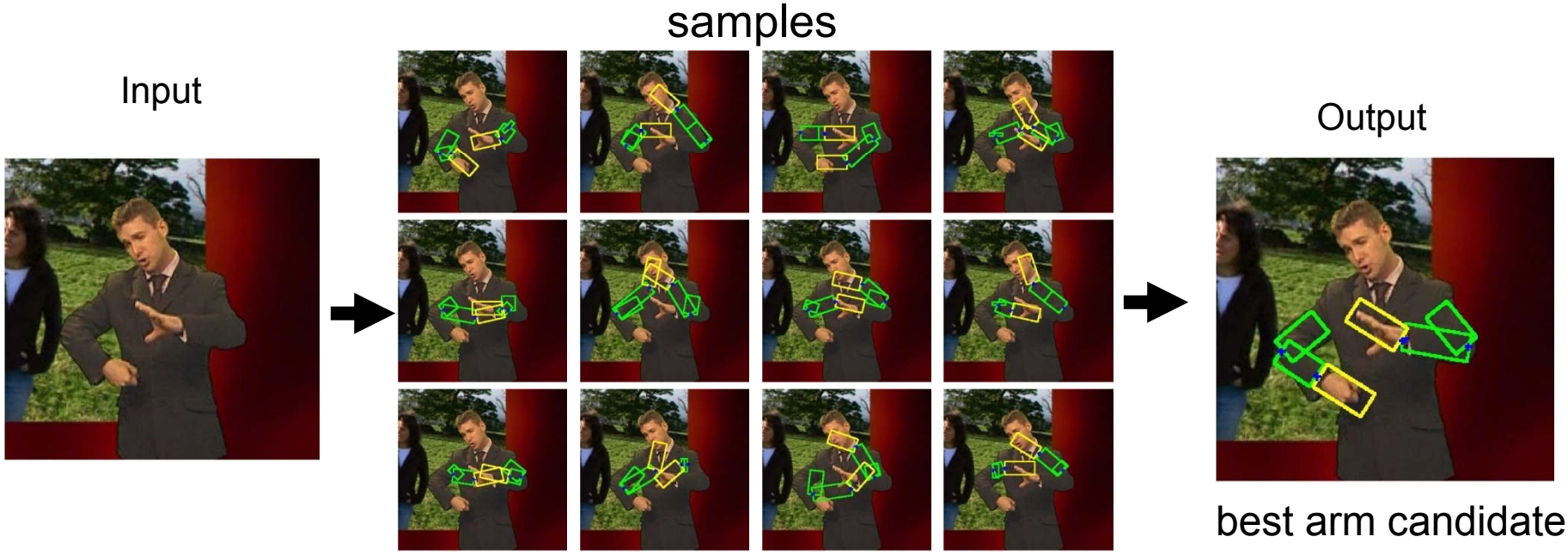
- Fit head and torso *[Navaratnam et al. 2005]*
- Then: arms and hands



Problem: Brute force search is still not feasible

Model fitting by sampling

- Sample configurations from inexpensive model
- Evaluate configuration using full model

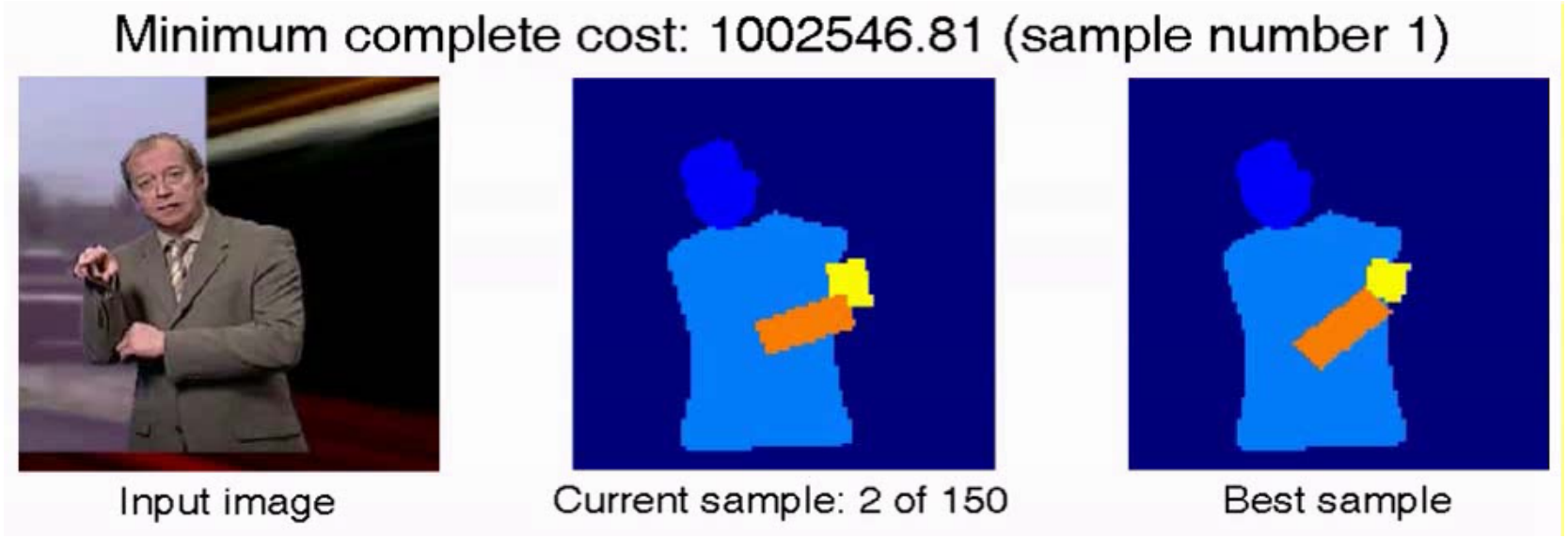


For sampling use tree structured pictorial Structures:

- [Felzenszwalb & Huttenlocher 2000, 2005]
- Complexity linear in the number of parts → O(nh)
- Pr(f | data): Sample from max-marginal with heuristics 1000 times
- cf Felzenszwalb & Huttenlocher 2005 sampled from marginal

Model fitting by sampling

- Sample configurations from inexpensive tree structured model
- Evaluate configuration using full model



Example results

Pose estimation results

Application

Learning sign language by watching TV (using weakly aligned subtitles)

Patrick Buehler

Mark Everingham

Andrew Zisserman

CVPR 2009

Objective

Learn signs in British Sign Language (BSL) corresponding to text words:

- Training data from TV broadcasts with simultaneous signing
- Supervision solely from sub-titles

Input: video + subtitle

Output: automatically
learned signs (4x slow motion)

Office

Government

Use subtitles to find video sequences containing word. These are the positive training sequences. Use other sequences as negative training sequences.

Overview

Given an English word e.g. “tree” what is the corresponding British Sign Language sign?

positive sequences

negative set

Use sliding window to choose sub-sequence of poses in one positive sequence and determine if

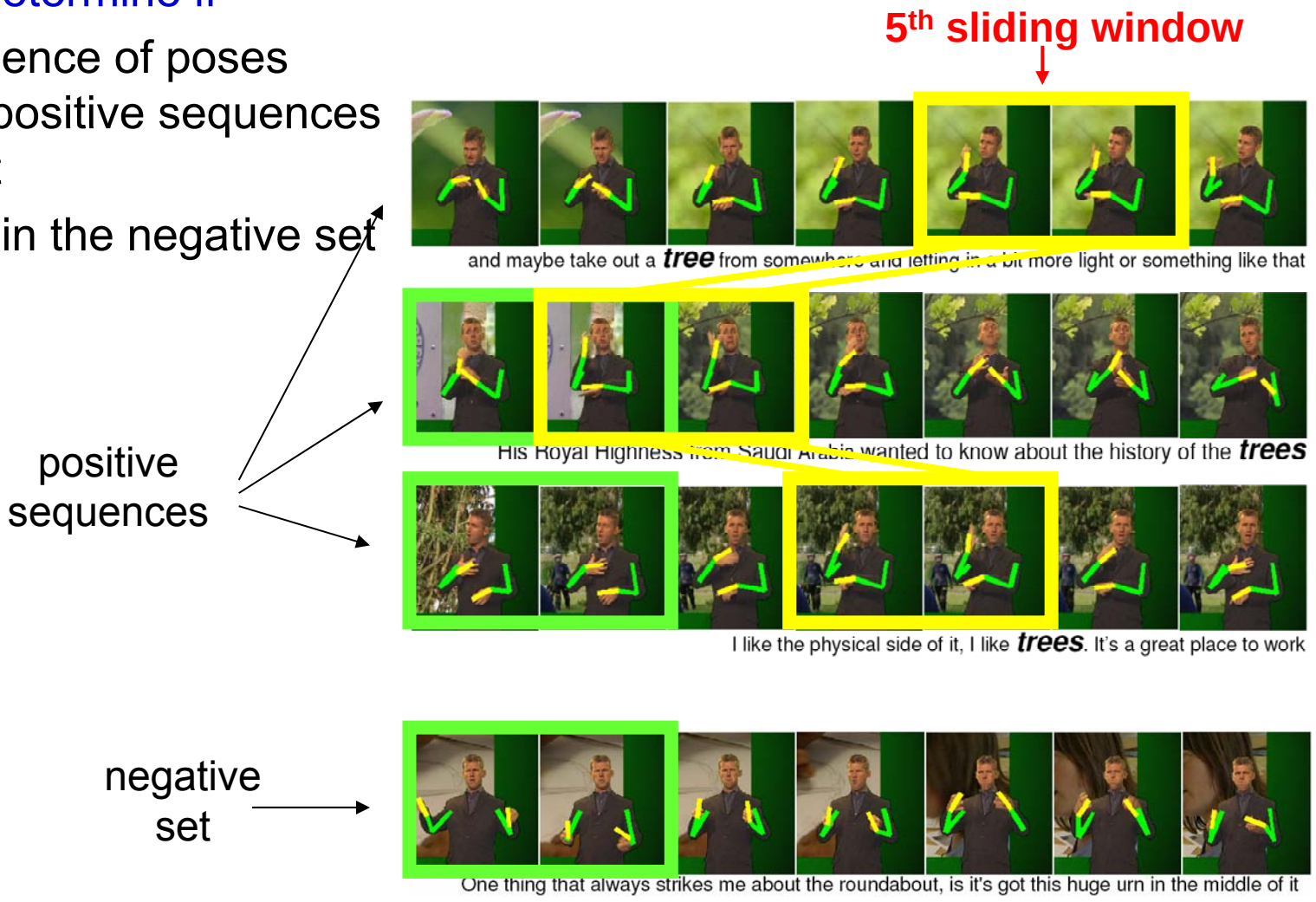
same sub-sequence of poses occurs in other positive sequences somewhere, but

does not occur in the negative set

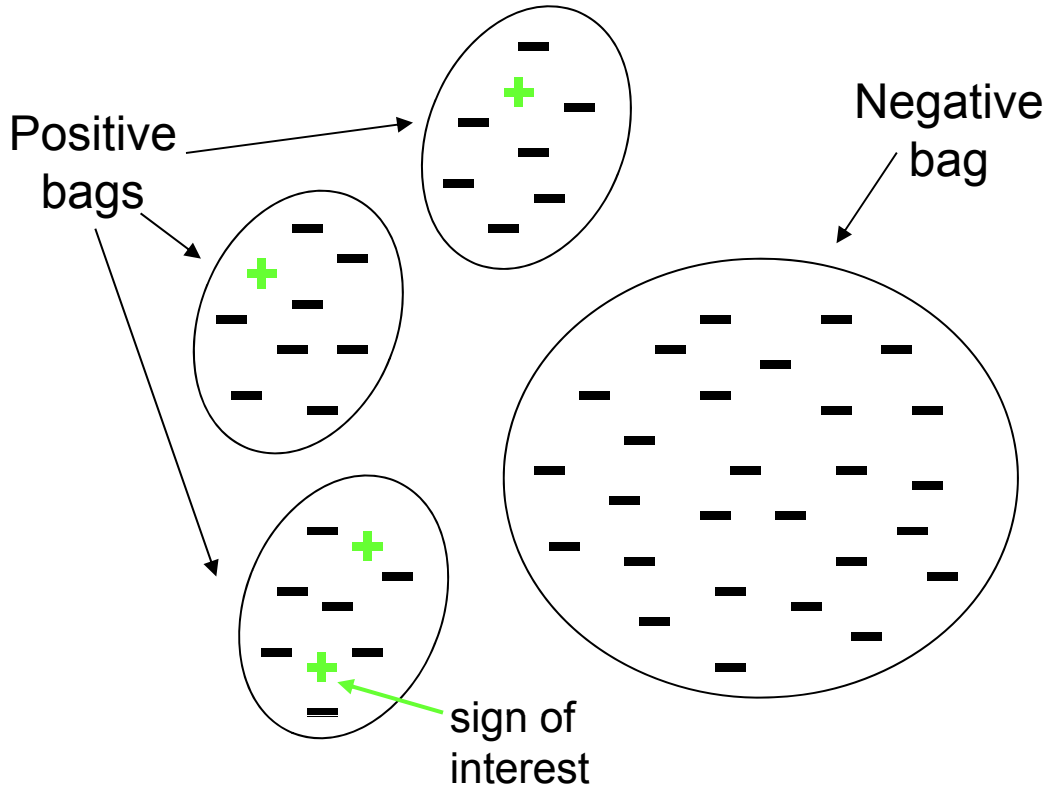
Use sliding window to choose sub-sequence of poses in one positive sequence and determine if

same sub-sequence of poses occurs in other positive sequences somewhere, but

does not occur in the negative set



Multiple instance learning



Evaluation

Good results for a variety of signs:

Signs where hand movement is important

Navy

Prince

Signs where hand shape is important

Lung

Garden

Signs where both hands are together

Fungi

Golf

Signs which are finger--spelled

Kew

Bob

Signs which are perfomed in front of the face

Whale

Rose

Summary

Given a good appearance model and proper account of foreground and background, then problems such as occlusion and ordering can be resolved. The cost of inference still remains though.

Next:

How to obtain models automatically in videos and images

If the appearance features are discriminative, how far can one go with foreground only pictorial structures and tree based inference?

Outline

Review of pictorial structures for articulated models

Inference given the model: Strong supervision, full generative model – “Gold-standard model”

Image parsing: learning the model for a specific image

Recent advances

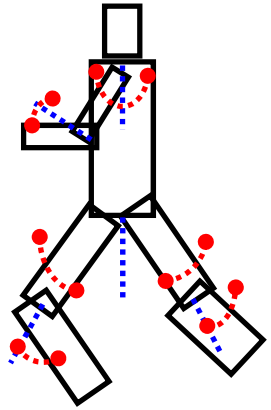
Datasets and challenges

Learning appearance models in videos

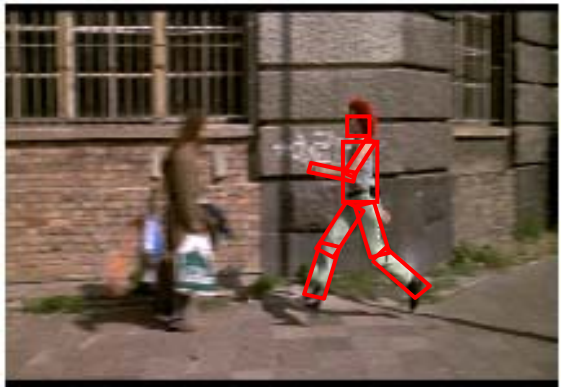
Strike a Pose: Tracking People by Finding Stylized Poses
Deva Ramanan, David Forsyth and Andrew Zisserman, CVPR 2005

edges

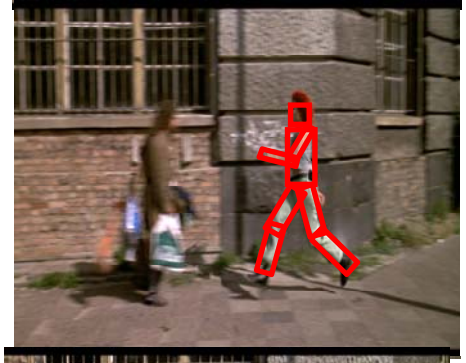
walking pose pictorial structure



efficient matching



Build Model

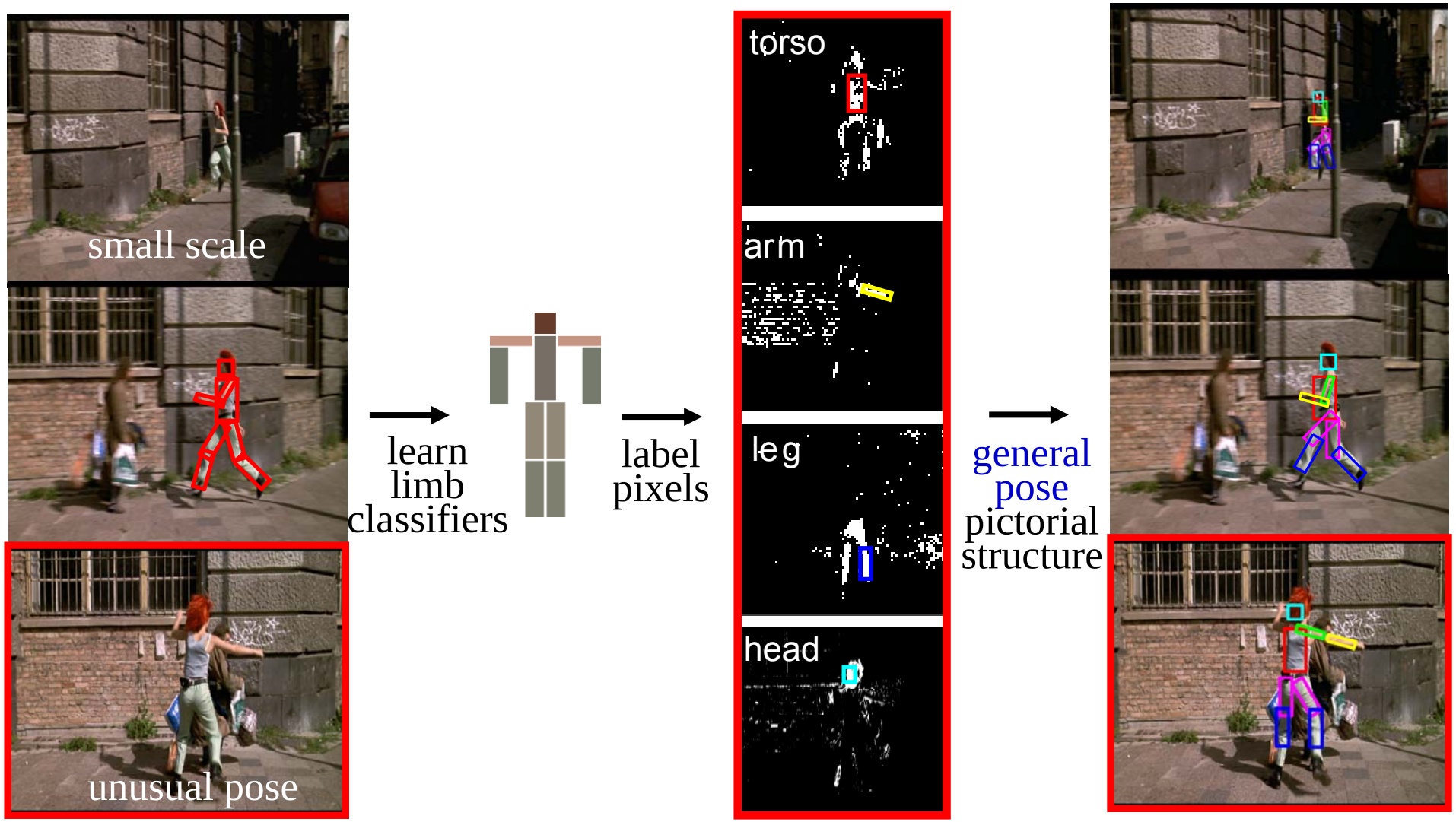


find
discriminative
features

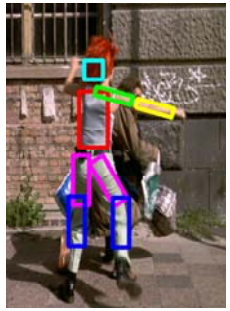
learn
limb
classifiers

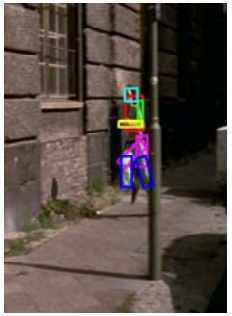
(limb pixels alone
are poor model)

Build Model & Detect

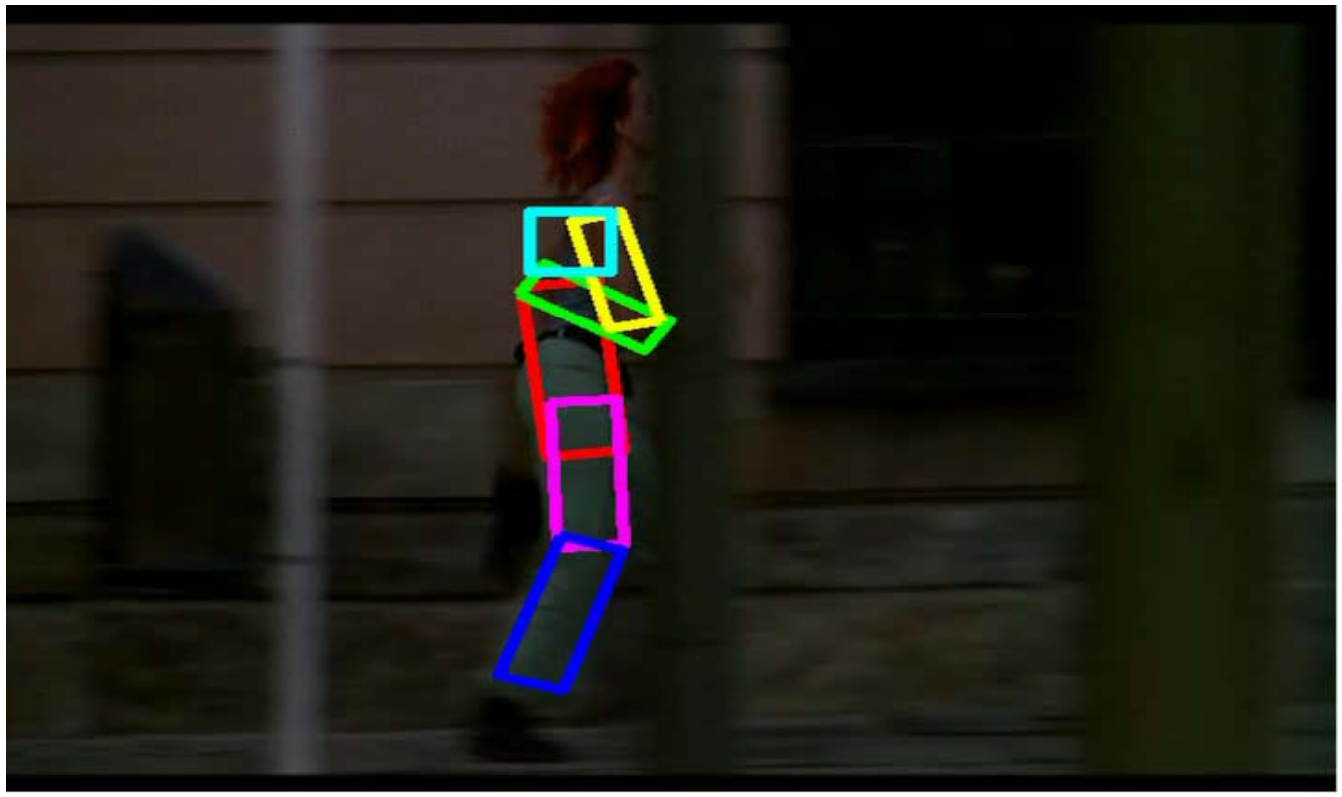


Running Example





How well do classifiers generalize?



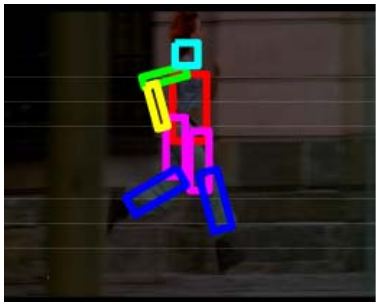
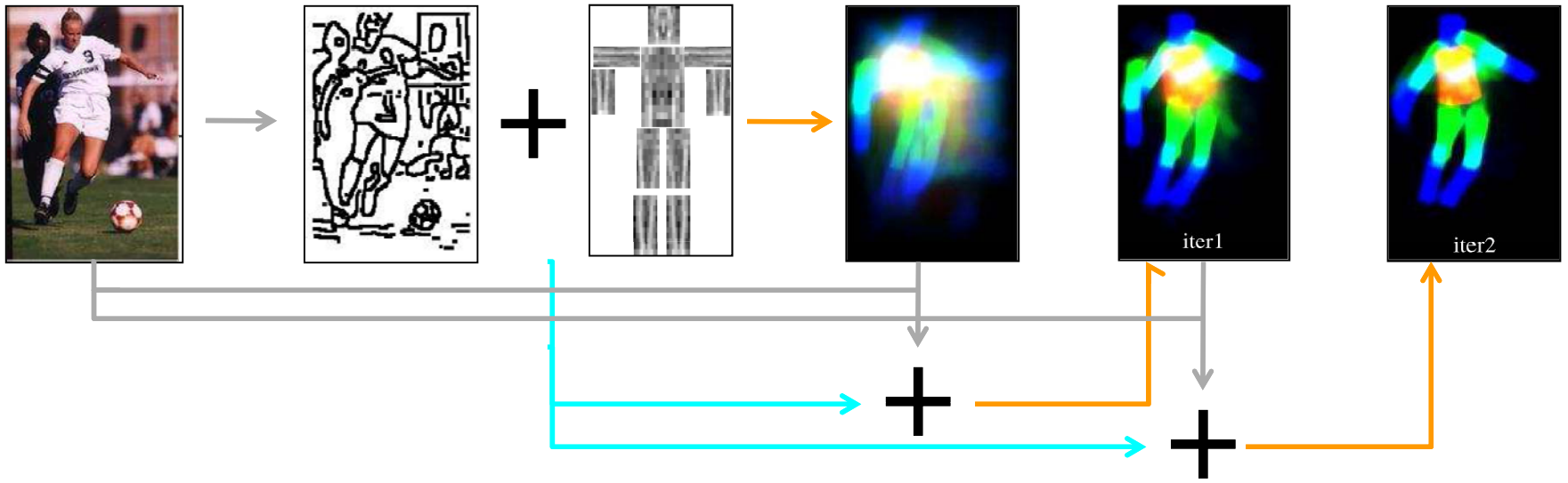


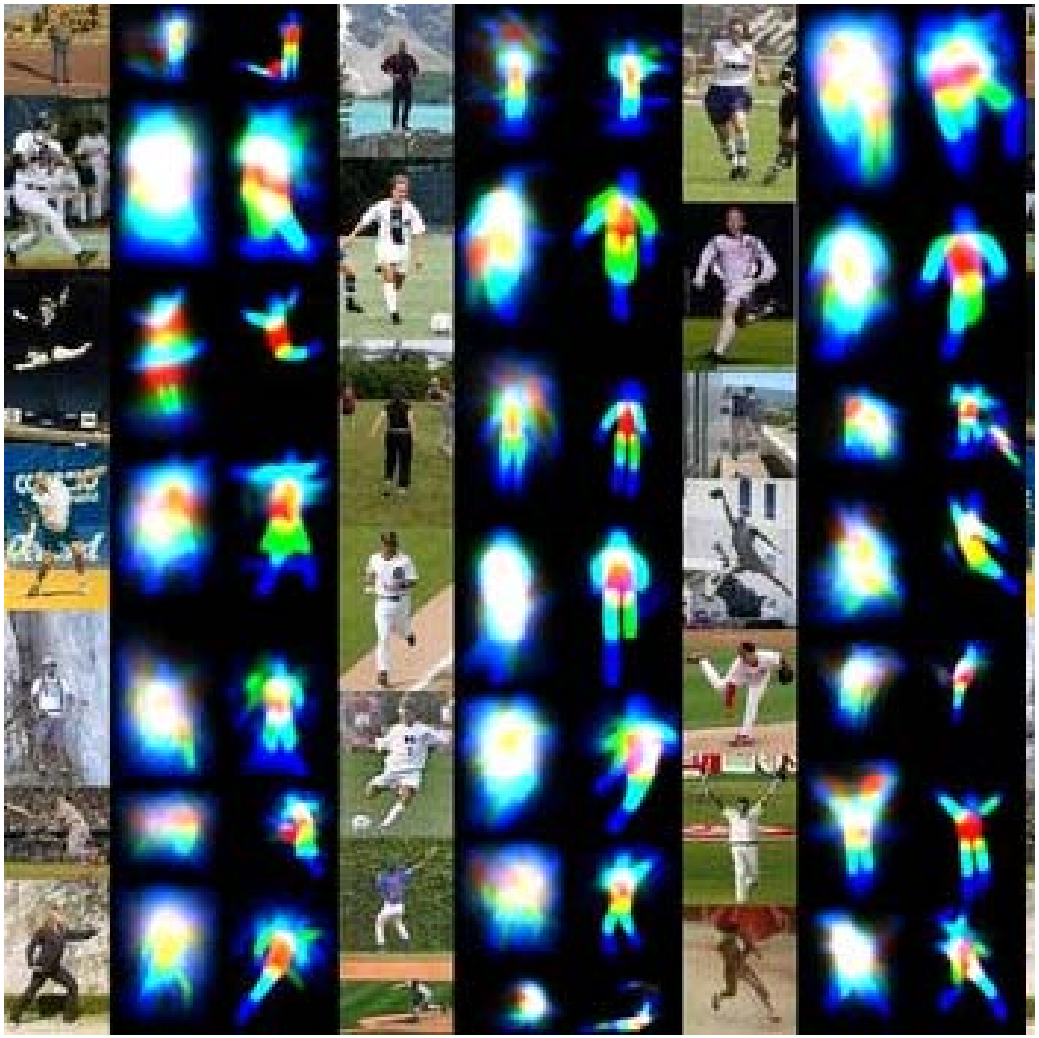
Image Parsing – Ramanan NIPS 06

$$E(f) = \underbrace{\sum_{a \in \mathcal{V}} \theta_{a;f(a)}}_{\text{unary terms (edges/colour)}} + \underbrace{\sum_{(a,b) \in \mathcal{E}} \theta_{ab;f(a)f(b)}}_{\text{pairwise terms (configuration)}}$$

Learn image and person specific unary terms

- initial iteration → edges
- following iterations → edges & colour



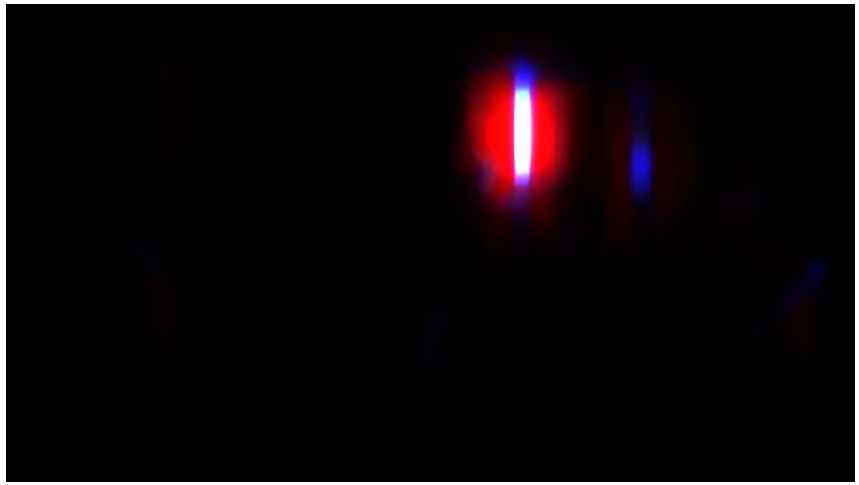


(Almost) unconstrained images

Extremely difficult when knowing nothing about appearance/pose/location

Failure of direct pose estimation

Ramanan NIPS 2006 unaided



Not powerful enough for a cluttered image where size is not given

Progressive search space reduction for human pose estimation

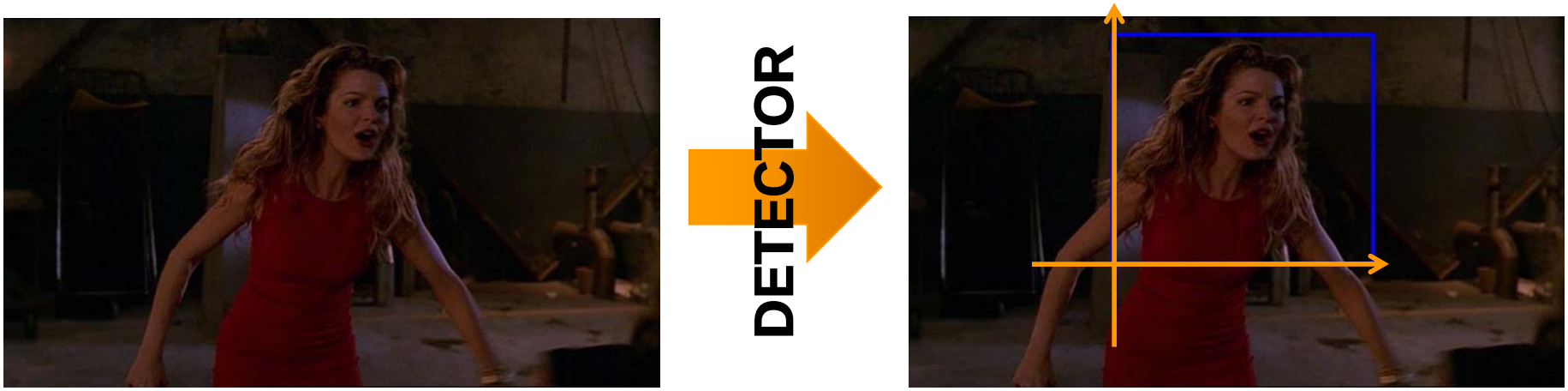
Vitto Ferrari, Manuel Marin-Jimenez, Andrew Zisserman

CVPR 2008/2009

Restrict search space using detector

Find (x,y,s) coordinate frame for a person

detection window (upper-body, face etc.)



Ferrari et al. 08, Andriluka et al. 09, Gammeter et al. 08

Learn an image and person specific model

Supervision

- None

Weaker model

- Tree structured graphical model
- Overlap not modelled
- Single scale parameter
- No background model

Inference

- **Detect person** – use upper body detector
- Use upper body region to restrict search
- Use colour segmentation to restrict search further
- Parsing pictorial structure by Ramanan NIPS 06

Search space reduction by upper body human detection

(1) detect human; (2) reduce search from h^n

Train

Idea

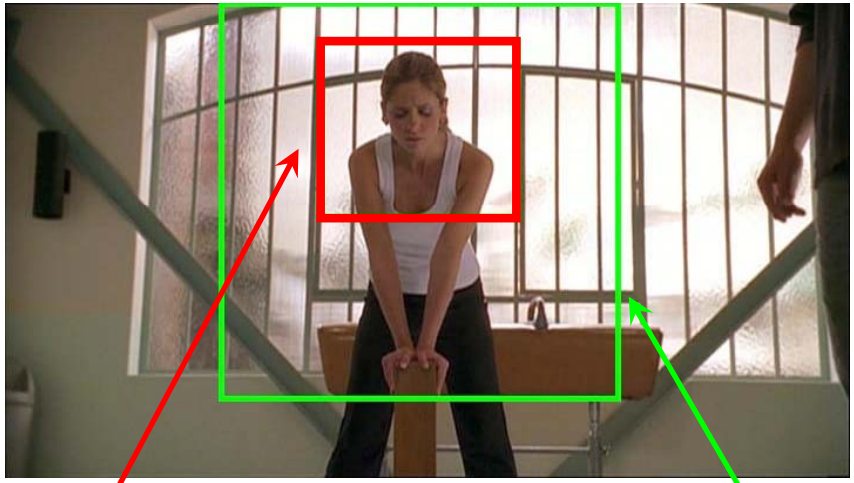
get approximate location and scale with a detector generic over pose and appearance

Building an upper-body detector

- based on Dalal and Triggs CVPR 2005

- train = 96 frames X 12 perturbations

Test



Benefits for pose estimation

+ fixes scale of body parts

+ sets bounds on x,y locations

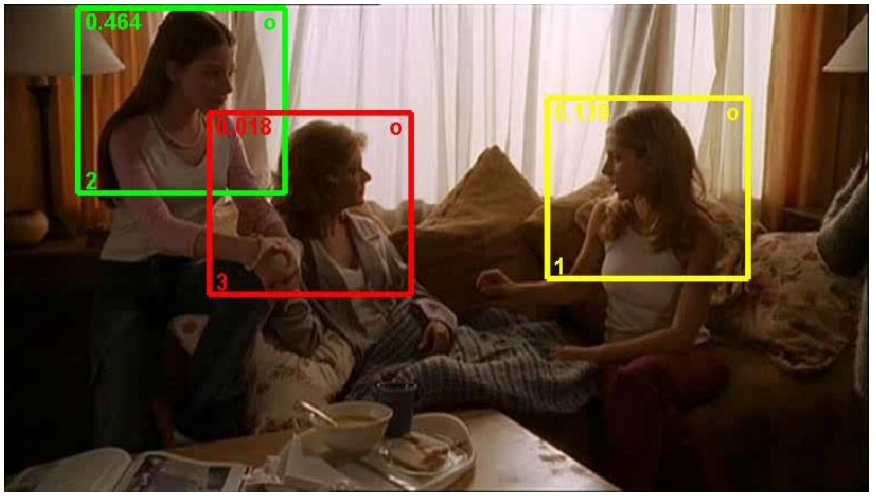
+ detects also back views

+ fast

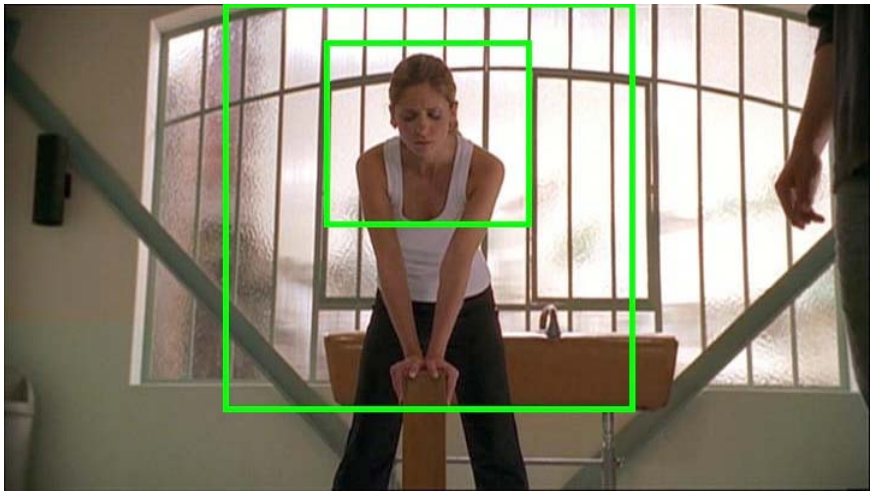
- little info about pose (arms)

Upper body detector – using HOGs

average training data



Search space reduction by foreground highlighting



initialization

output

Idea

exploit knowledge about structure of search area to initialize Grabcut

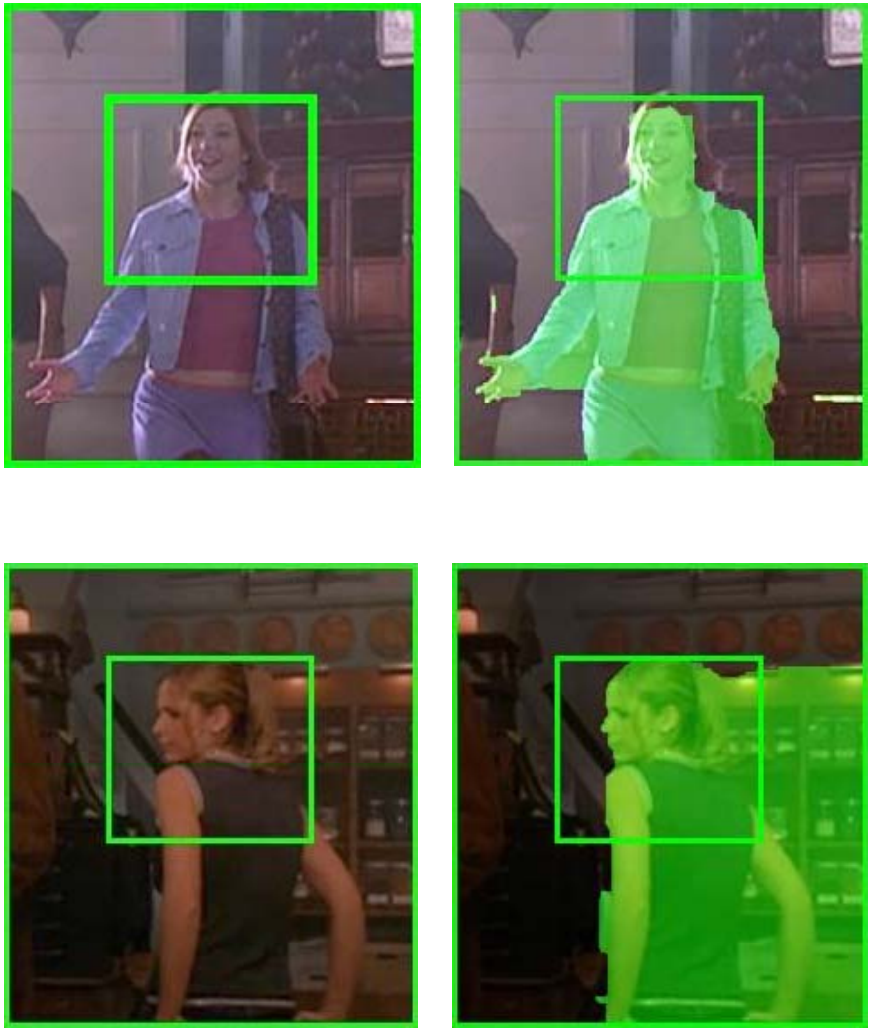
Initialization

- learn fg/bg models from regions where person likely present/absent
- clamp central strip to fg
- don't clamp bg (arms can be anywhere)

Benefits for pose estimation

+ further reduce clutter
+ conservative (no loss 95.5% times)
+ needs no knowledge of background
+ allows for moving background

Search space reduction by foreground highlighting



Idea

exploit knowledge about structure of search area to initialize Grabcut

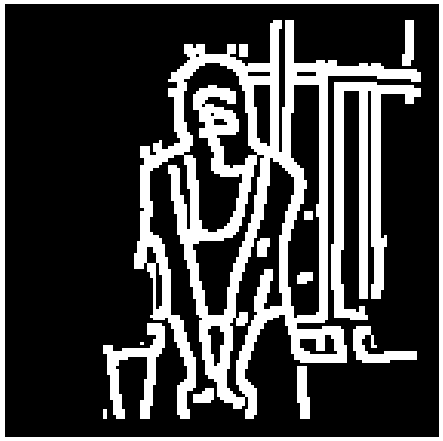
Initialization

- learn fg/bg models from regions where person likely present/absent
- clamp central strip to fg
- don’t clamp bg (arms can be anywhere)

Benefits for pose estimation

+ further reduce clutter
+ conservative (no loss 95.5% times)
+ needs no knowledge of background
+ allows for moving background

Pose estimation by image parsing - Ramanan NIPS 06



edge parse

appearance

edge + col parse

Goal

estimate posterior of part configuration

$$E(f) = \underbrace{\sum_{a \in \mathcal{V}} \theta_{a;f(a)}}_{\text{unary terms (edges/colour)}} + \underbrace{\sum_{(a,b) \in \mathcal{E}} \theta_{ab;f(a)f(b)}}_{\text{pairwise terms (configuration)}}$$

Algorithm

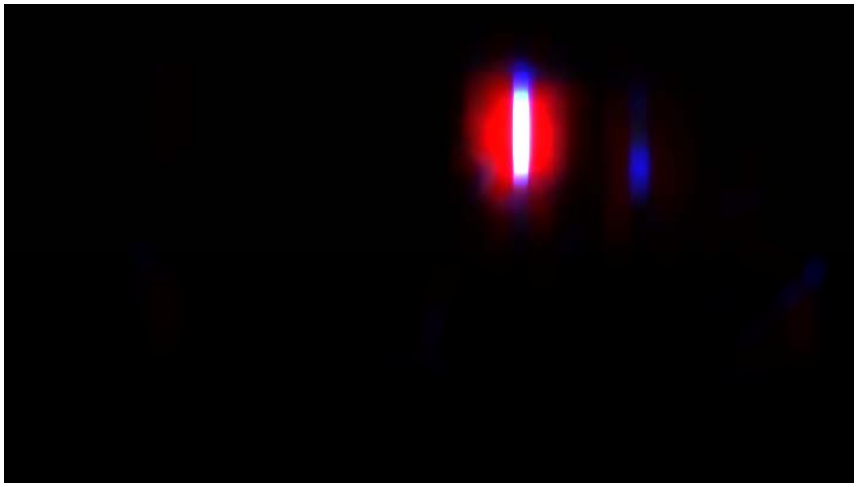
1. inference with edges unary
2. learn appearance models of body parts and background
3. inference with edges + colour unary

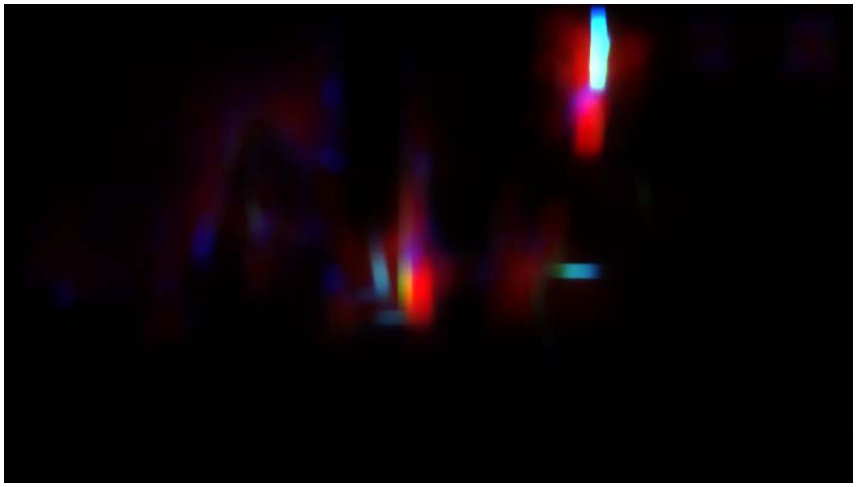
Advantages of space reduction

+ much more robust
+ much faster (10x-100x)

Failure of direct pose estimation

Ramanan NIPS 2006 unaided



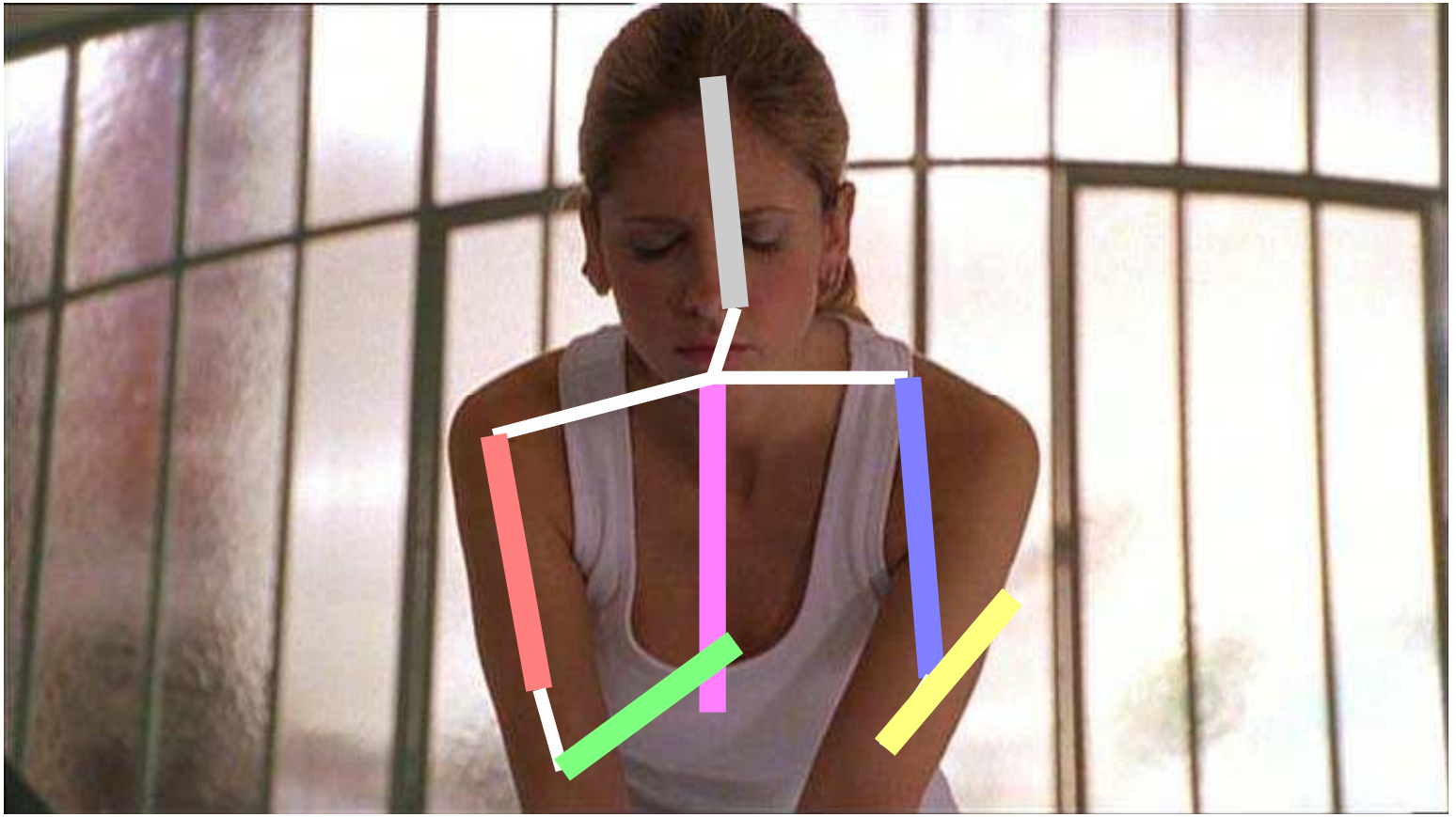


Results on Buffy frames

Results on PASCAL flickr images

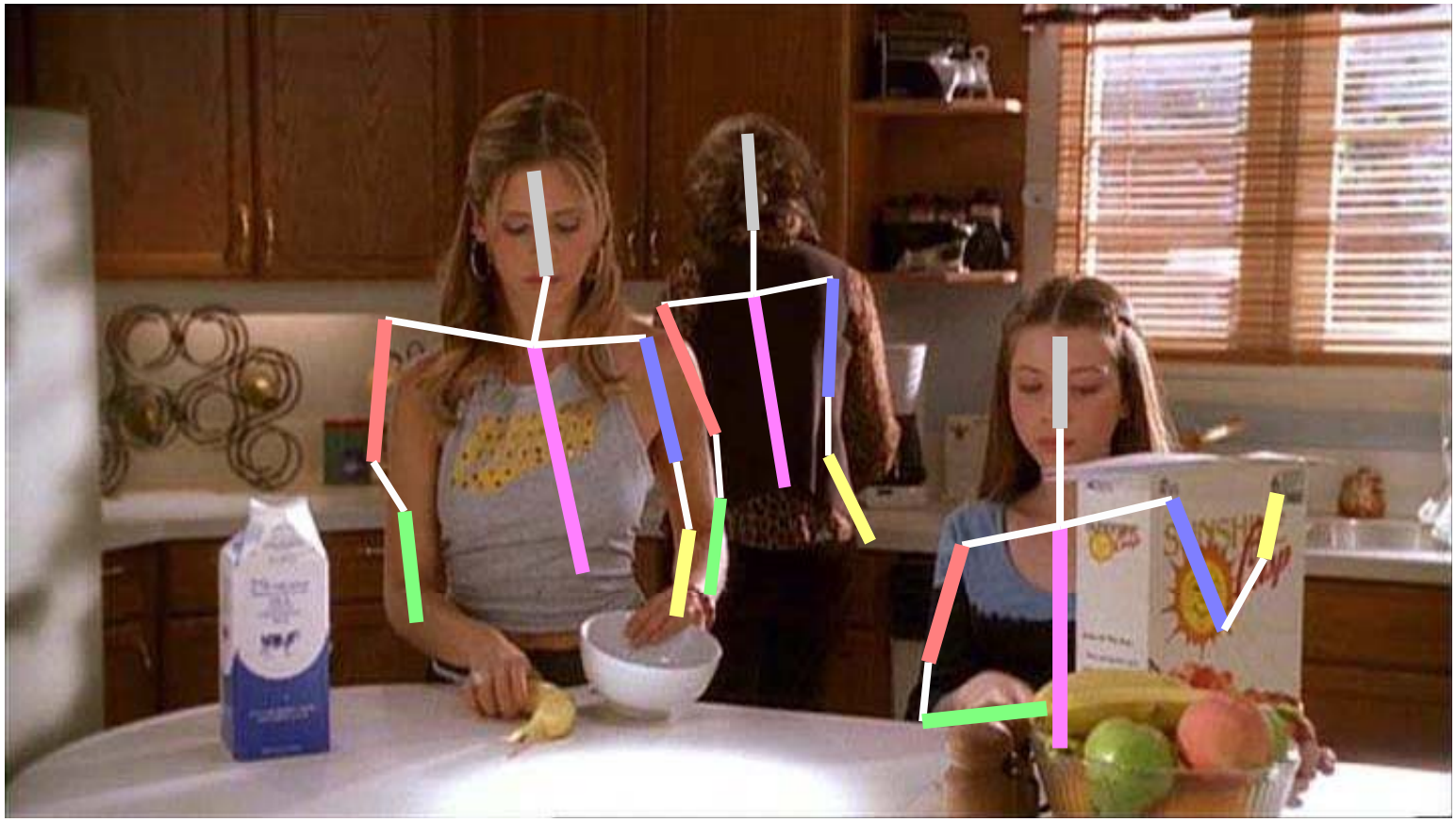
What is missed?

What is missed?



truncation is not modelled

What is missed?



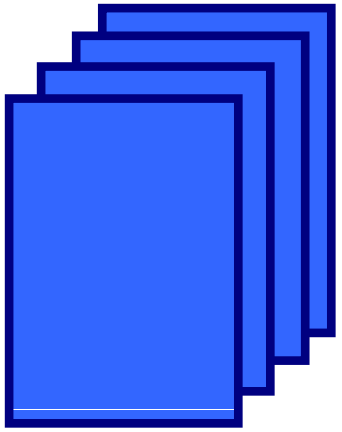
occlusion is not modelled

Application: Pose Search

Given user-selected
query frame+person …

query

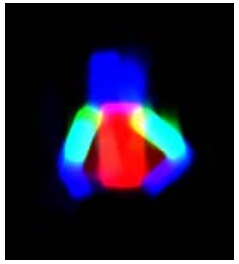
… retrieve shots with persons
in the same pose from video database

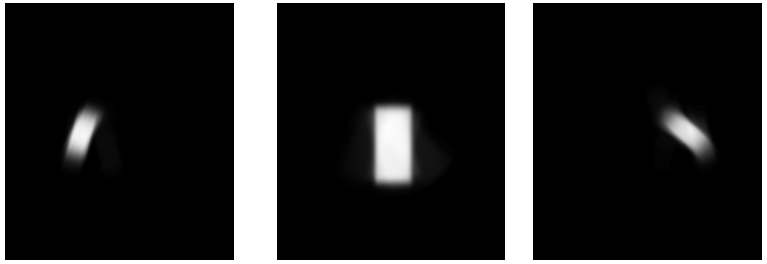


video database

CVPR 2009

Pose Search





Pose descriptors

- soft-segmentations of body parts

- distributions over orient+location for parts and pairs of parts

Similarity measures

- dot-product (= soft intersection)

- Batthacharrya / Chi-square

Processing

Off-line:

- Detect upper bodies in every frame
- Link (track) upper body detections
- Estimate upper body pose for each frame of track
- Compute descriptor (vector) for each upper body pose

Run-time:

- Rank each track by its similarity to the query pose

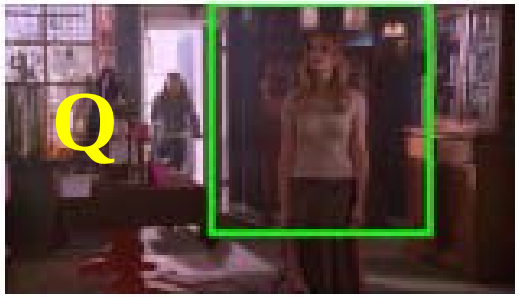
Pose Search

"hips pose"

Pose Search

"rest pose"

Pose Search



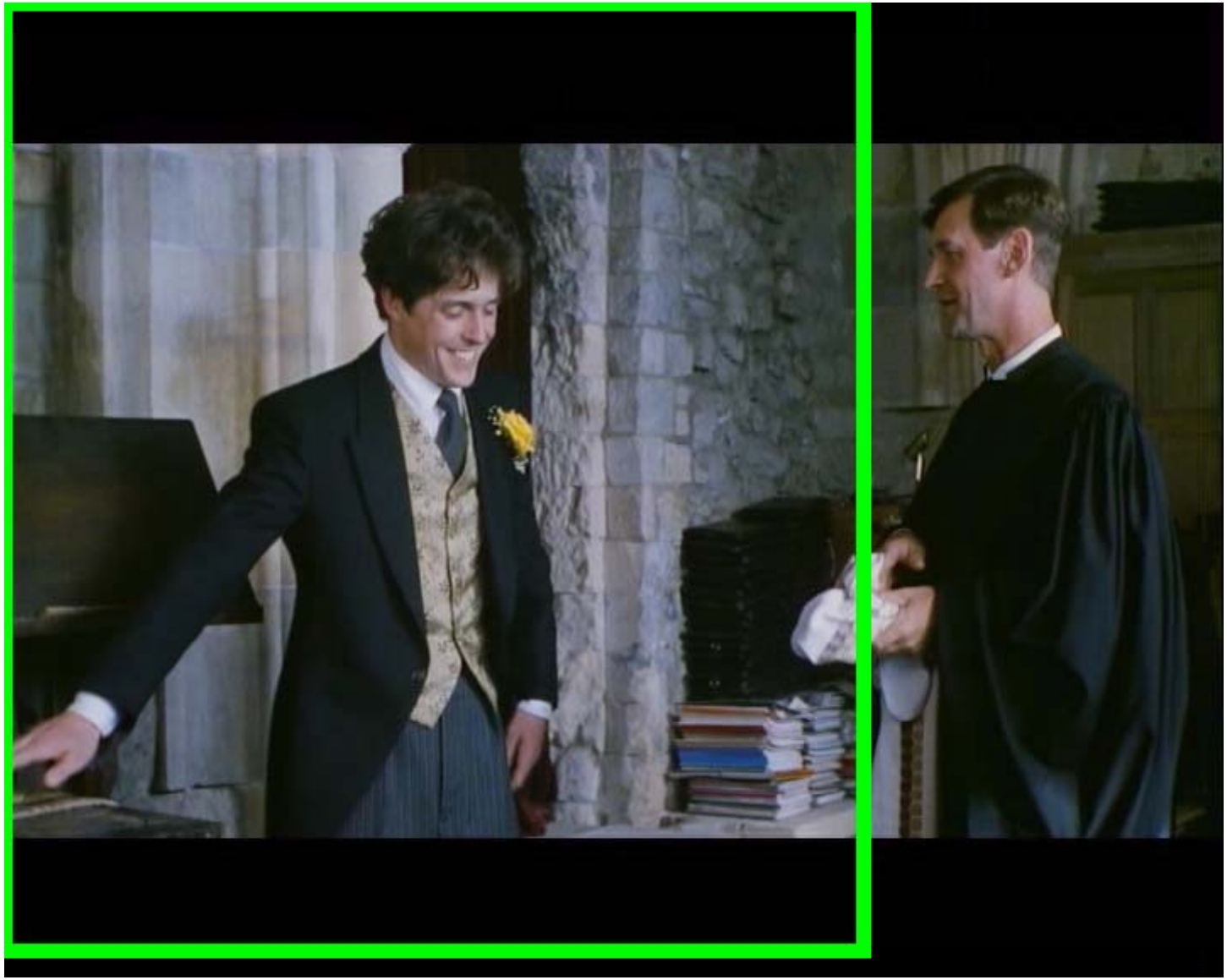
"rest pose"

Other poses – query interesting pose

Hollywood movies – Query on Gandhi, Search Hugh Grant opus

Other poses – query interesting pose

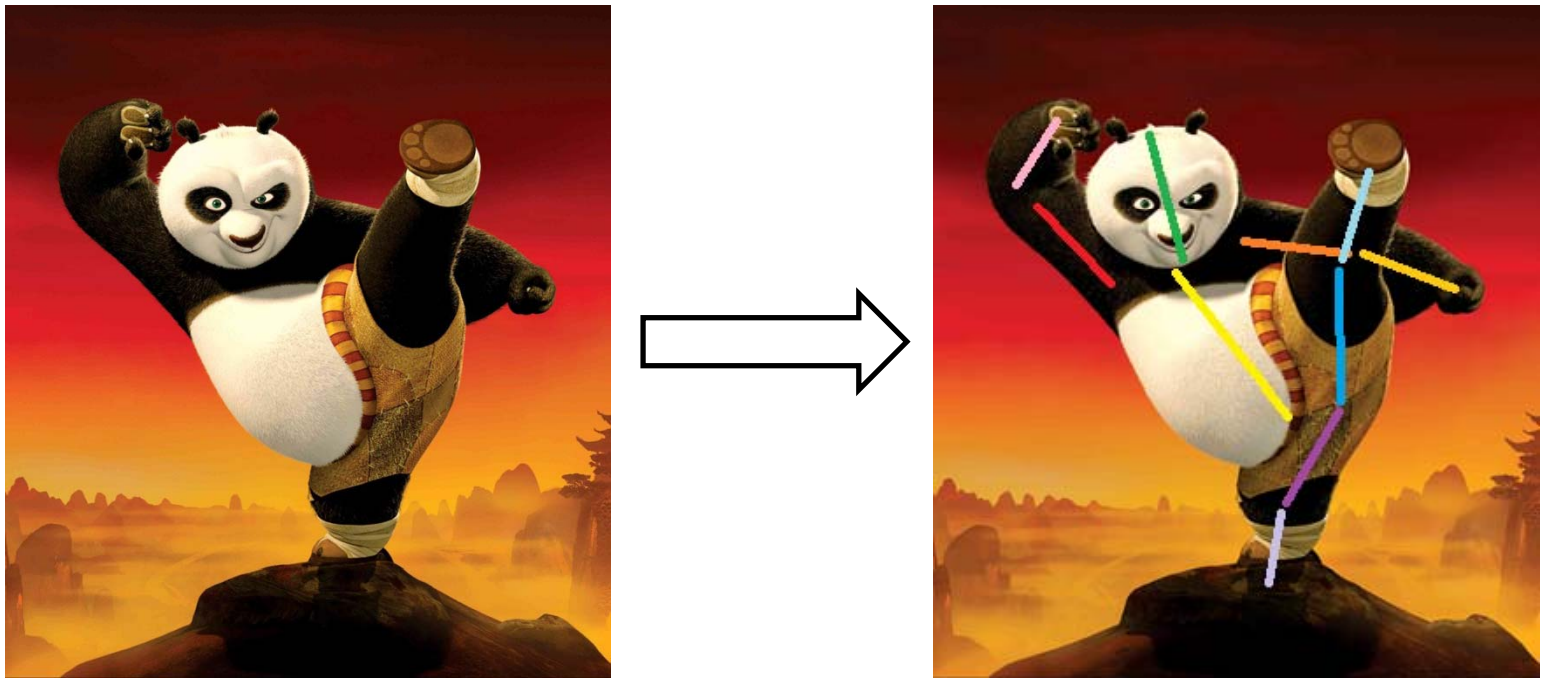
Hollywood movies – Query on Gandhi, Search Hugh Grant opus



Articulated Pose Estimation with Flexible Mixtures of Parts

Yi Yang & Deva Ramanan

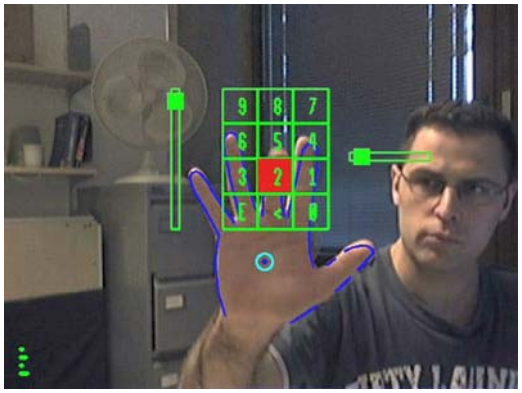
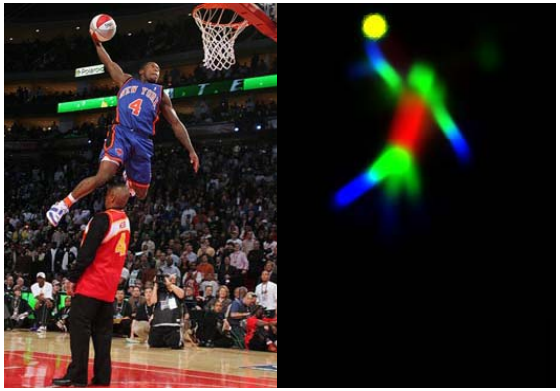
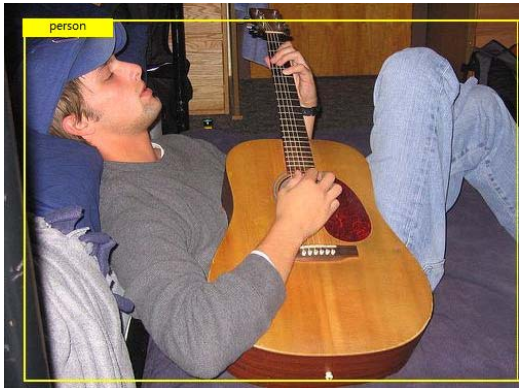
Goal



Articulated pose estimation (by Wikipedia)

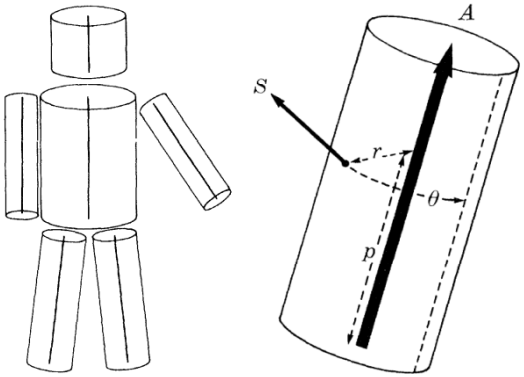
recovers the pose of an articulated object which consists of joints and rigid parts

Applications

Action 	HCI 	Gaming 
Segmentation 	Object Detection 	.....

Unconstrained Images

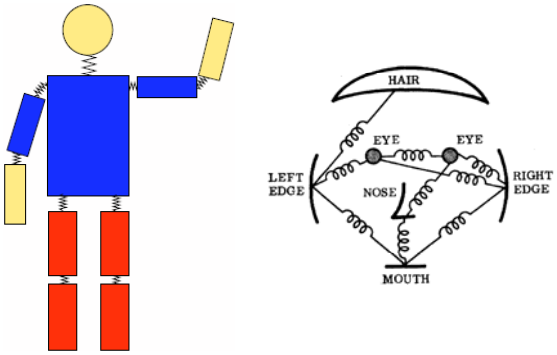
Classic Approach



Marr & Nishihara 1978

Part Representation

- Head, Torso, Arm, Leg
- Location, Rotation, Scale

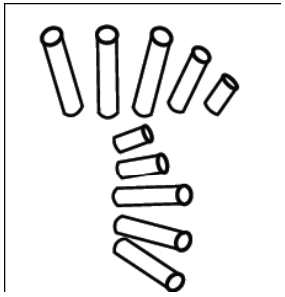


Fischler & Elschlager 1973

Felzenszwalb & Huttenlocher 2005

Pictorial Structure

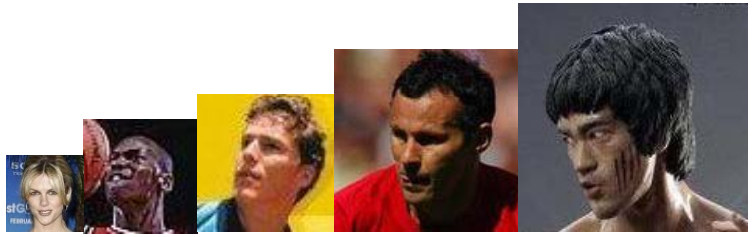
- Unary Templates
- Pairwise Springs



Lan & Huttenlocher 2005
Sigal & Black 2006
Ramanan 2007
Epshteian & Ullman 2007
Wang & Mori 2008
Ferrari etc. 2008
Andriluka etc. 2009
Eichner etc. 2009
Singh etc. 2010
Johnson & Everingham 2010
Sapp etc. 2010
Tran & Forsyth 2010

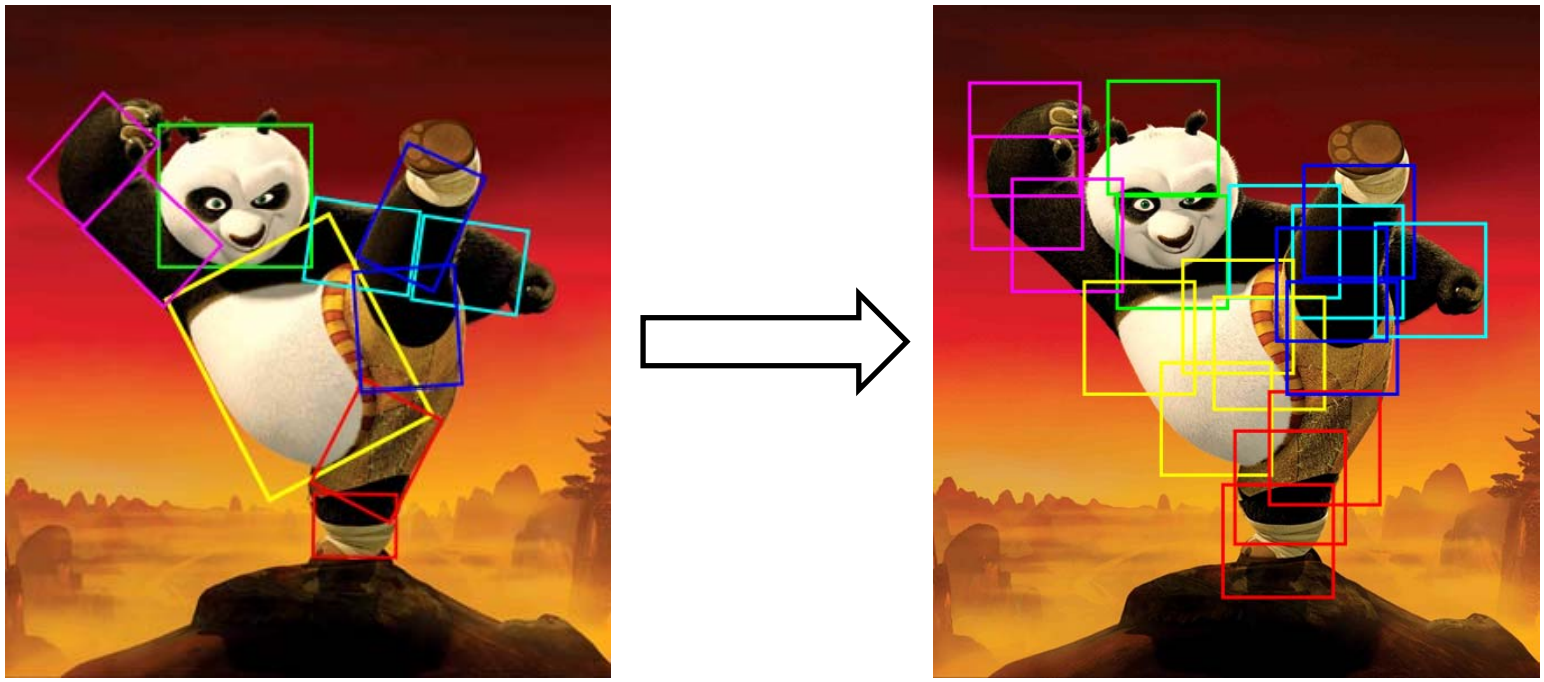
Problem

How to capture **affine** deformations of limbs?

In plane rotation	Out plane rotation
	
Foreshortening	
	

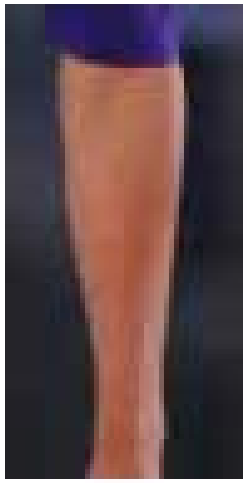
Naïve brute-force evaluation is expensive

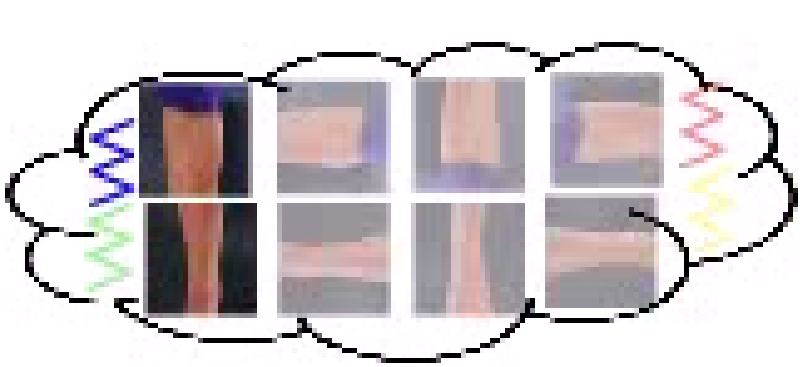
Our Approach – “Mini” Parts

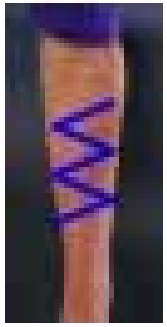


Capture affine deformations with “mini” part model

Example: Arm Approximation



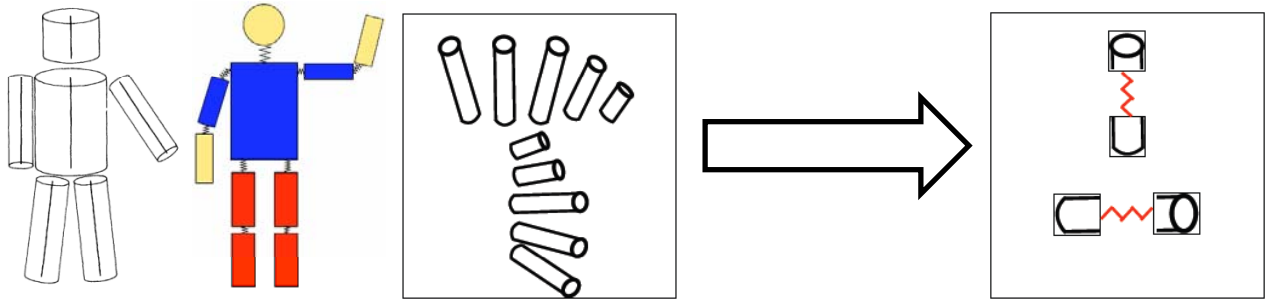




Example: Torso Approximation

Our Approach

- Extension of Pictorial Structure Model

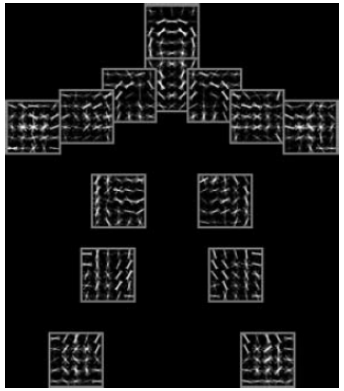


- Why?

Flexibility: General affine warps (orientation, foreshortening, ...)

Speed: Mixtures of local templates + dynamic programming

Linear-Parameterized Pictorial Structure Model



$S(I, L)$

- I: Image;
- K: Number of parts
- $L = \{l_i \mid i = 1, \ldots, K\}$: Locations of parts

Linear-Parameterized Pictorial Structure Model

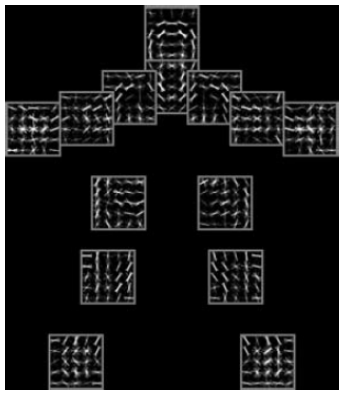
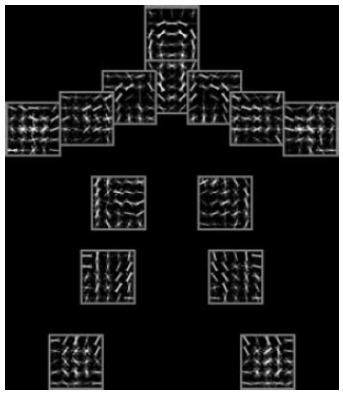
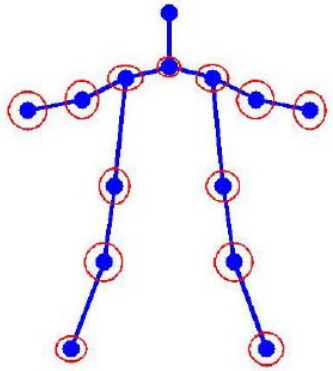
- V: Vertices
- α_i : Unary template for part i
- $\phi(I, l_i)$: Local image features at location l_i

- V: Vertices
- α_i : Unary template for part i
- $\phi(I, l_i)$: Local image features at location l_i

$$S(I, L) = \sum_{i \in V} \alpha_i \cdot \phi(I, l_i)$$

- V: Vertices
- α_i : Unary template for part i
- $\phi(I, l_i)$: Local image features at location l_i

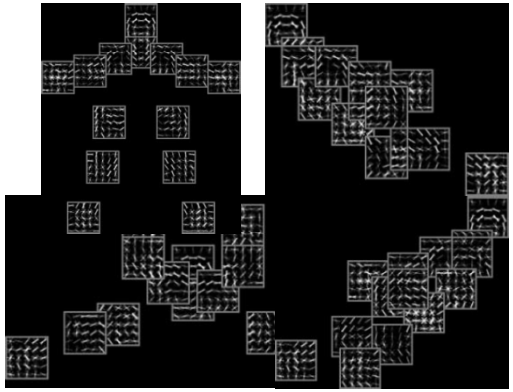
Linear-Parameterized Pictorial Structure Model

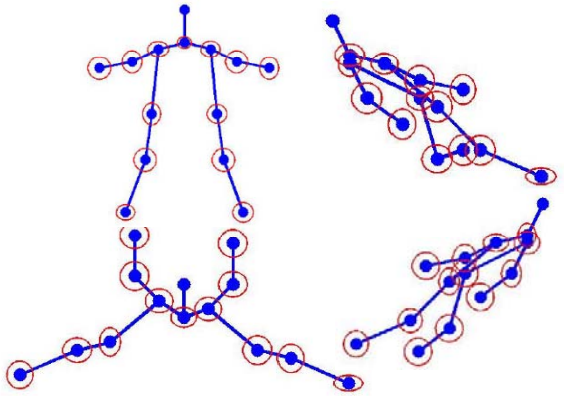




$$S(I,L) = \sum_{i \in V} \alpha_i \cdot \phi(I, l_i) + \sum_{ij \in E} \beta_{ij} \cdot \psi(l_i, l_j)$$

- E: Edges
- $\psi(l_i, l_j)$: Spatial features between l_i and l_j
- β_{ij}: Pairwise springs between part i and part j

Our Flexible Mixture Model

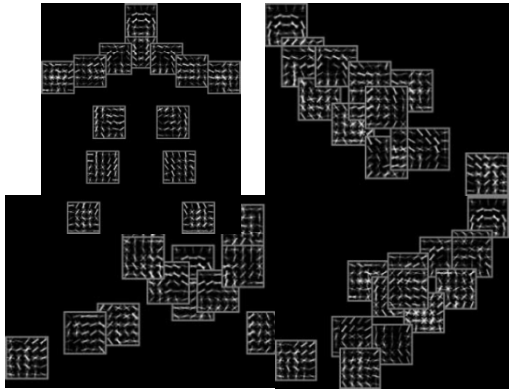


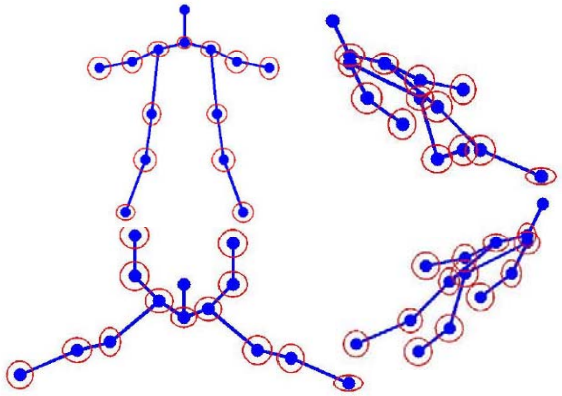


$$S(I, L, M) = \sum_{i \in V} \alpha_i^{m_i} \cdot \phi(I, l_i) + \sum_{ij \in E} \beta_{ij}^{m_i m_j} \cdot \psi(l_i, l_j)$$

- M= $\{m_i \mid i = 1, \ldots, K\}$: Mixtures of parts
- $\alpha_i^{m_i}$: Unary template for part i with mixture m_i
- $\beta_{ij}^{m_i m_j}$: Pairwise springs between part i with mixture m_i and part j with mixture m_j

Our Flexible Mixture Model

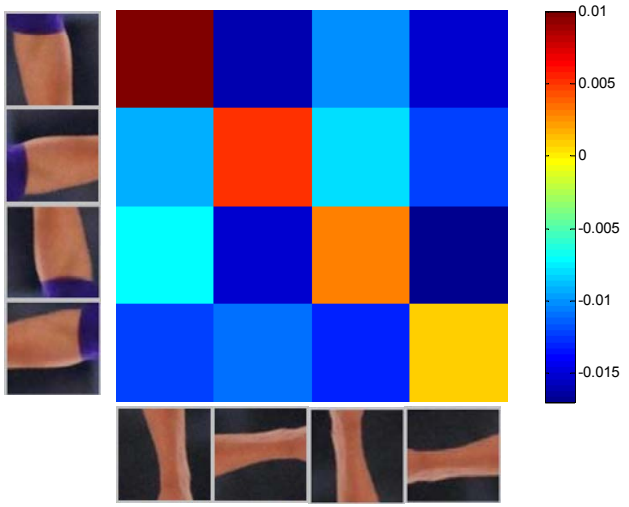


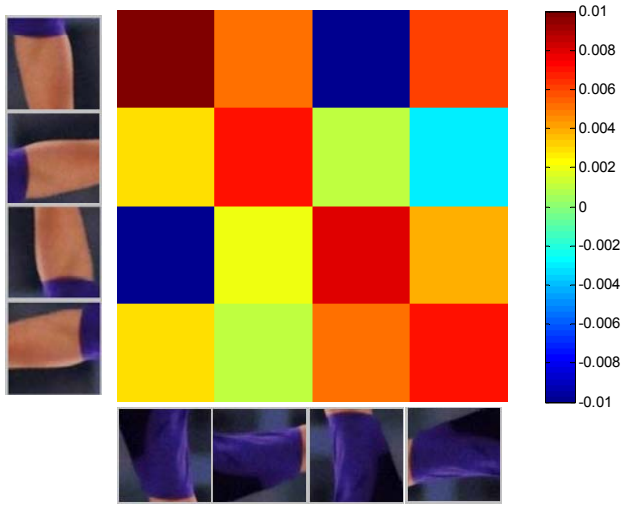


$$S(I,L,M) = \sum_{i \in V} \alpha_i^{m_i} \cdot \phi(I, l_i) + \sum_{ij \in E} \beta_{ij}^{m_i m_j} \cdot \psi(l_i, l_j) + S(M)$$

- M= $\{m_i \mid i = 1, \ldots, K\}$: Mixtures of parts
- $\alpha_i^{m_i}$: Unary template for part i with mixture m_i
- $\beta_{ij}^{m_i m_j}$: Pairwise springs between part i with mixture m_i and part j with mixture m_j

Co-occurrence “Prior”





$$S(M) = \sum_{ij \in E} b_{ij}^{m_i m_j}$$

- $b_{ij}^{m_i m_j}$: Pairwise co-occurrence prior between part i with mixture m_i and part j with mixture m_j

Inference & Learning

Inference

$$\max_{L,M} S(I, L, M)$$

For a tree graph (V,E): dynamic programming

Inference & Learning

Inference

$$\max_{L,M} S(I, L, M)$$

For a tree graph (V,E): dynamic programming

Given labeled positive $\{I_n, L_n, M_n\}$ and negative $\{I_n\}$,
write $z_n = (L_n, M_n)$, and $S(I, z) = w \cdot \phi(I, z)$

Learning

$$\min_w \frac{1}{2}\|w\|$$

$$\text{s.t.} \quad \forall n \in \text{pos} \;\; w \cdot \phi(I_n, z_n) \geq 1$$

$$\forall n \in \text{neg}. \forall z \;\; w \cdot \phi(I_n, z) \leq -1$$

Benchmark Datasets

PARSE Full-body

http://www.ics.uci.edu/~dramanan/papers/parse/index.html

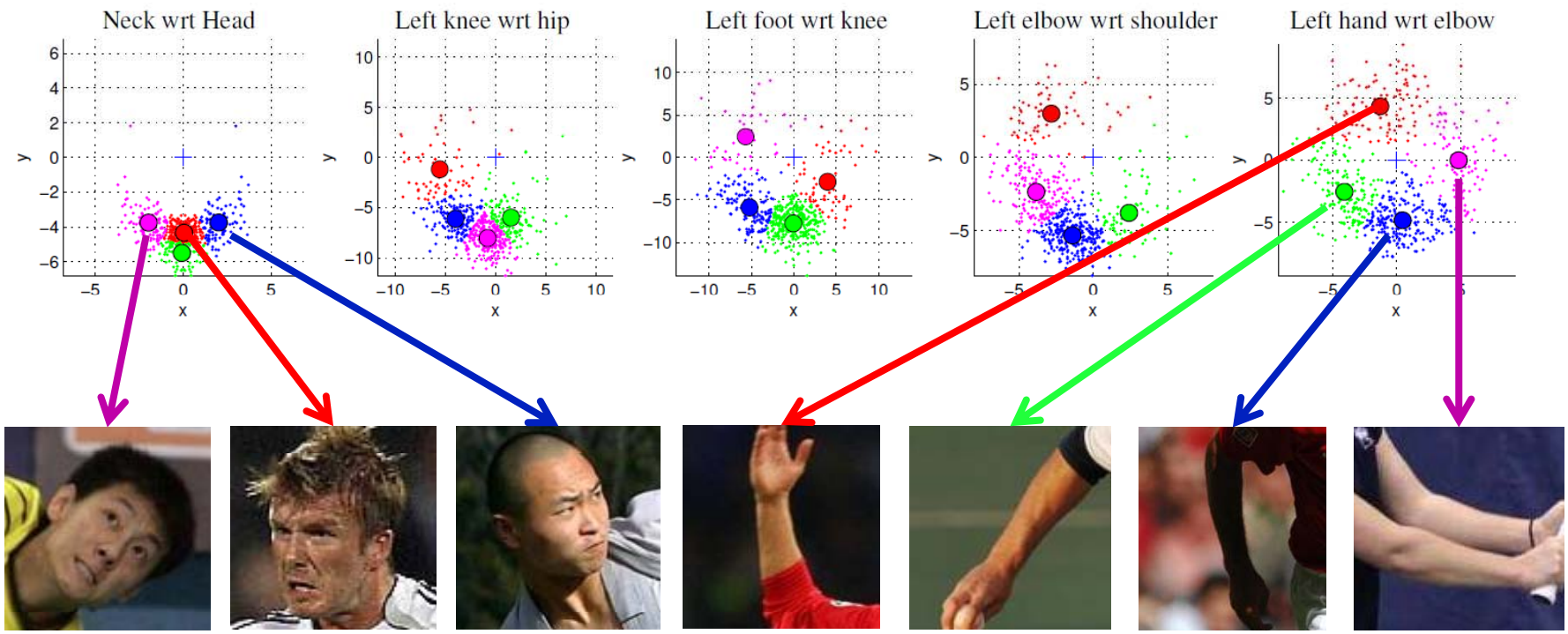
BUFFY Upper-body

http://www.robots.ox.ac.uk/~vgg/data/stickmen/index.html

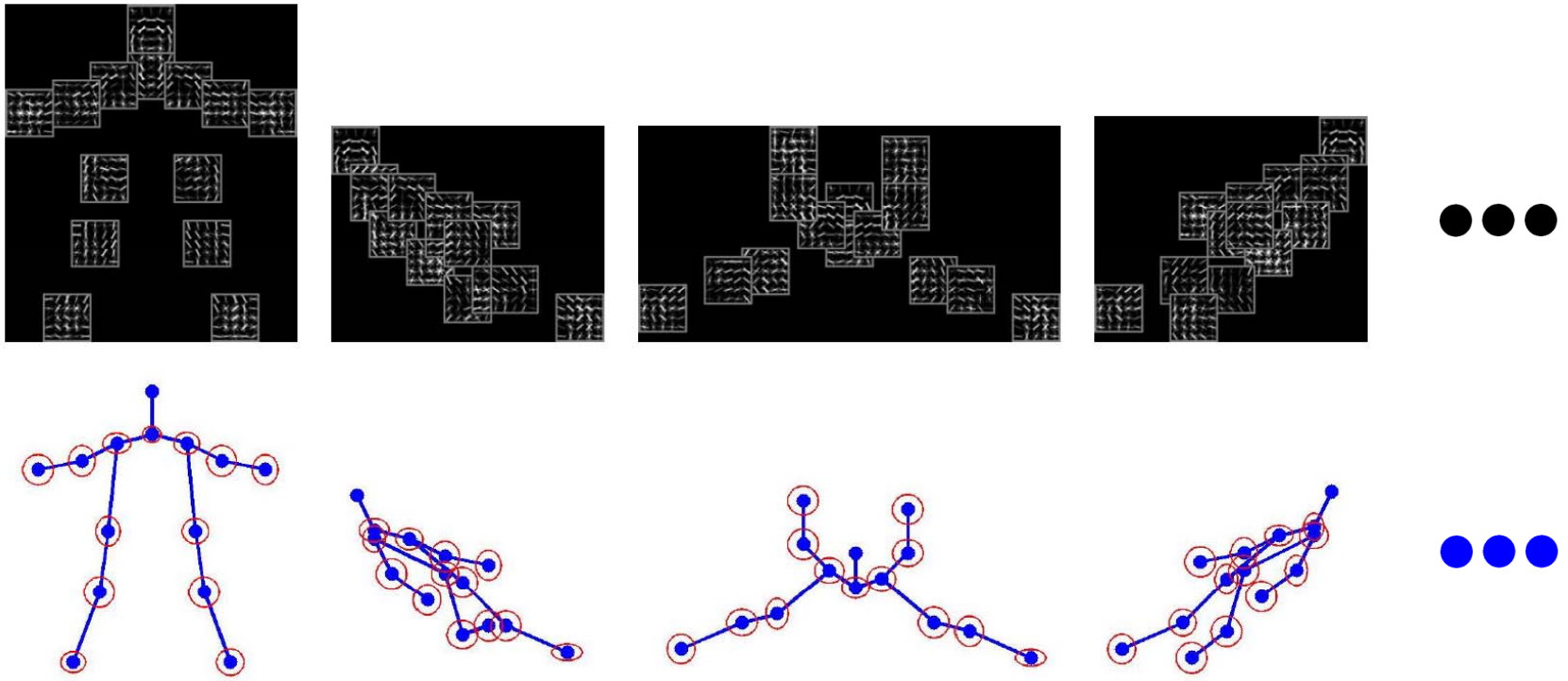
How to Get Part Mixtures?

Solution:

Cluster relative locations of joints w.r.t. parents



Articulation

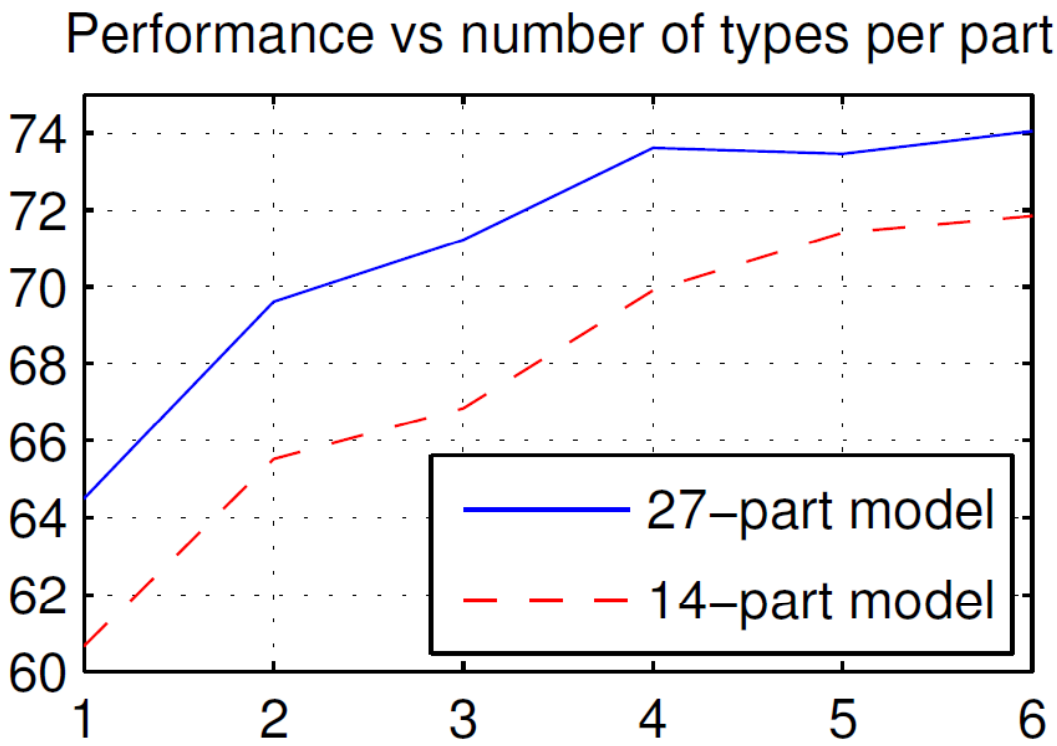


K parts, M mixtures ⇒ K^M unique pictorial structures

Not all are equally likely --- "prior" given by $S(M)$

Qualitative Results

Diagnostic



- 14 parts (joints) vs 27 parts (joints + midpoints)
- More parts and types/mixtures help

Quantitative Results

% of correctly localized limbs

Image Parse Testset

Method							Total
Ramanan 2007							27.2
Andrikluka 2009							55.2
Johnson 2010a							56.4
Singh 2010							60.9
Johnson 2010b							66.2
Our Model							**74.9**

All previous work use explicitly articulated models

Quantitative Results

% of correctly localized limbs

Image Parse Testset

Method	Head	Torso	U. Legs	L. Legs	U. Arms	L. Arms	Total
Ramanan 2007	52.1	37.5	31.0	29.0	17.5	13.6	27.2
Andrikluka 2009	81.4	75.6	63.2	55.1	47.6	31.7	55.2
Johnson 2010a	77.6	68.8	61.5	54.9	53.2	39.3	56.4
Singh 2010	91.2	76.6	71.5	64.9	50.0	34.2	60.9
Johnson 2010b	85.4	76.1	73.4	65.4	64.7	46.9	66.2
Our Model	**97.6**	**93.2**	**83.9**	**75.1**	**72.0**	**48.3**	**74.9**

1 second per image

Quantitative Results

% of correctly localized limbs

Subset of Buffy Testset

Method					Total
Tran 2010					62.3
Andrikluka 2009					73.5
Eichner 2009					80.1
Sapp 2010a					85.9
Sapp 2010b					85.5
Our Model					**89.1**

All previous work use explicitly articulated models

Quantitative Results

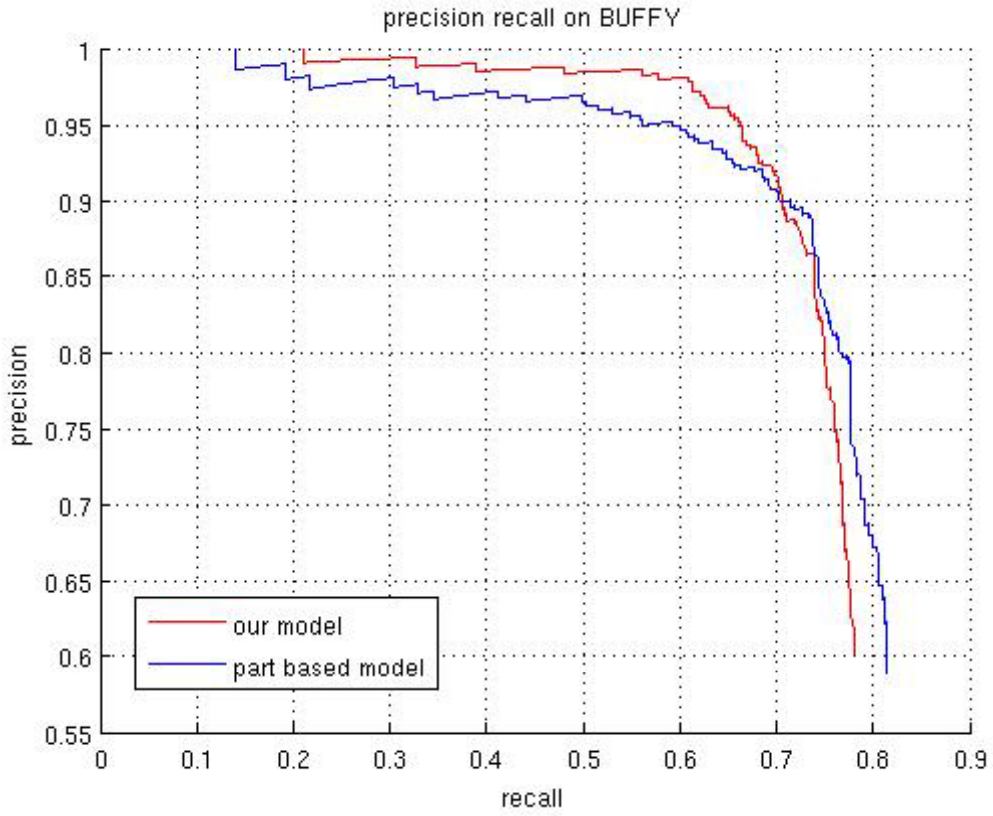
% of correctly localized limbs

Subset of Buffy Testset

Method	Head	Torso	U. Arms	L. Arms	Total
Tran 2010	---	---	---	---	62.3
Andrikluka 2009	90.7	95.5	79.3	41.2	73.5
Eichner 2009	98.7	97.9	82.8	59.8	80.1
Sapp 2010a	100	**100**	91.1	65.7	85.9
Sapp 2010b	100	96.2	95.3	63.0	85.5
Our Model	**100**	99.6	**96.6**	**70.9**	**89.1**

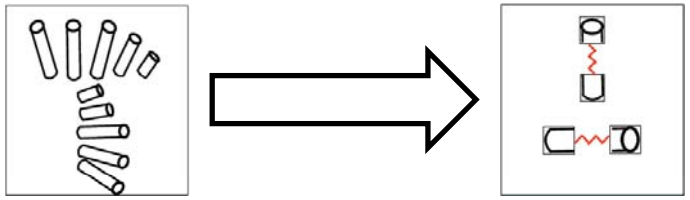
Ours | 5 seconds VS 5 minutes | next best

Human Detection



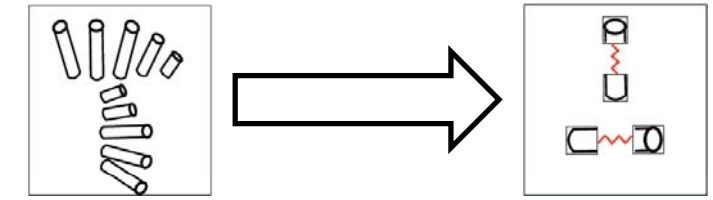
Conclusion

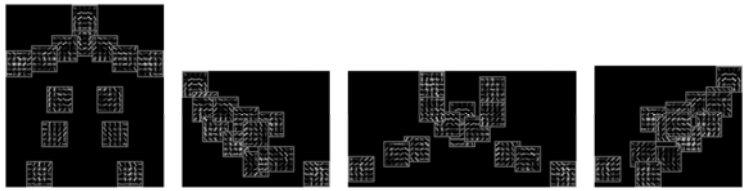
- Model affine warps with a part-based model



Conclusion

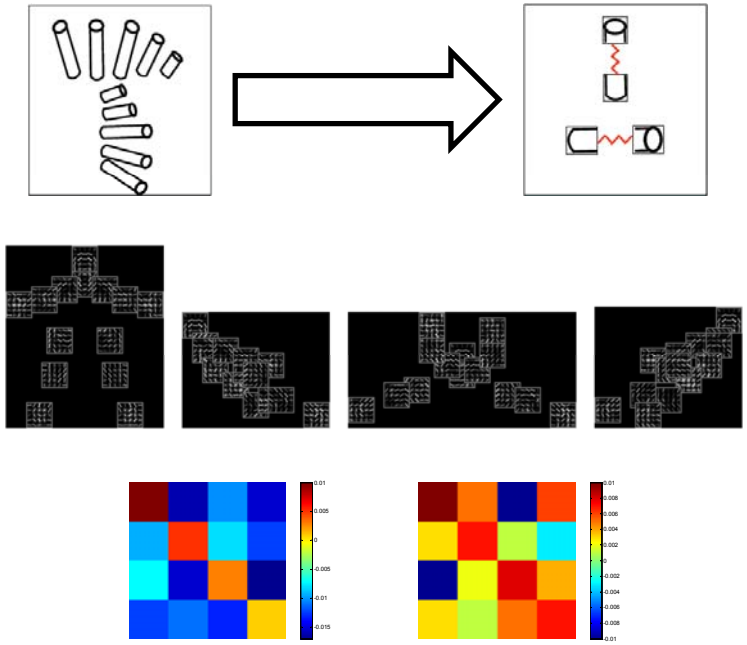
- Model affine warps with a part-based model
- Exponential set of pictorial structures





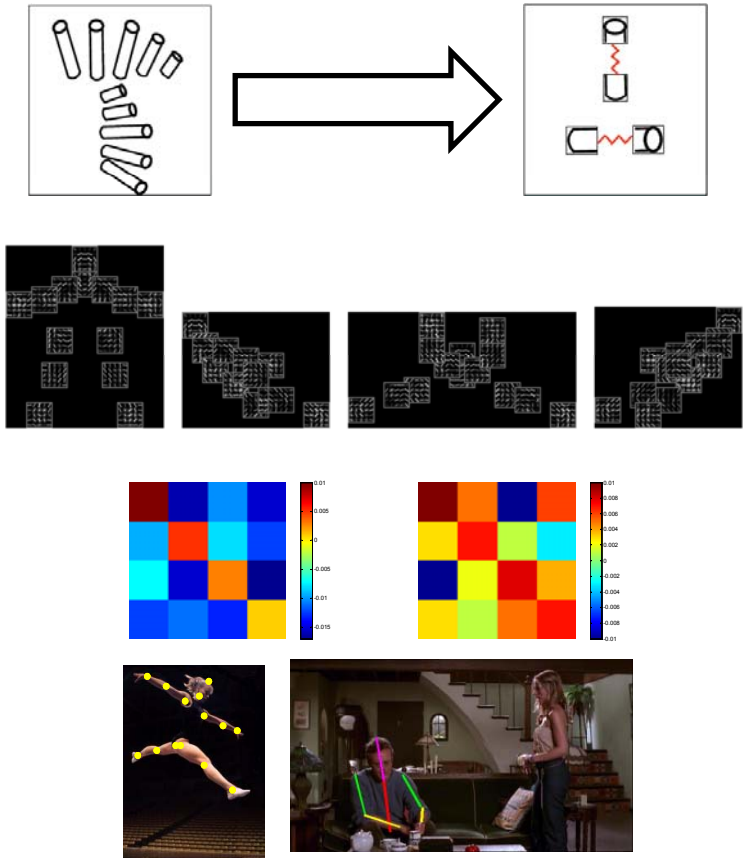
Conclusion

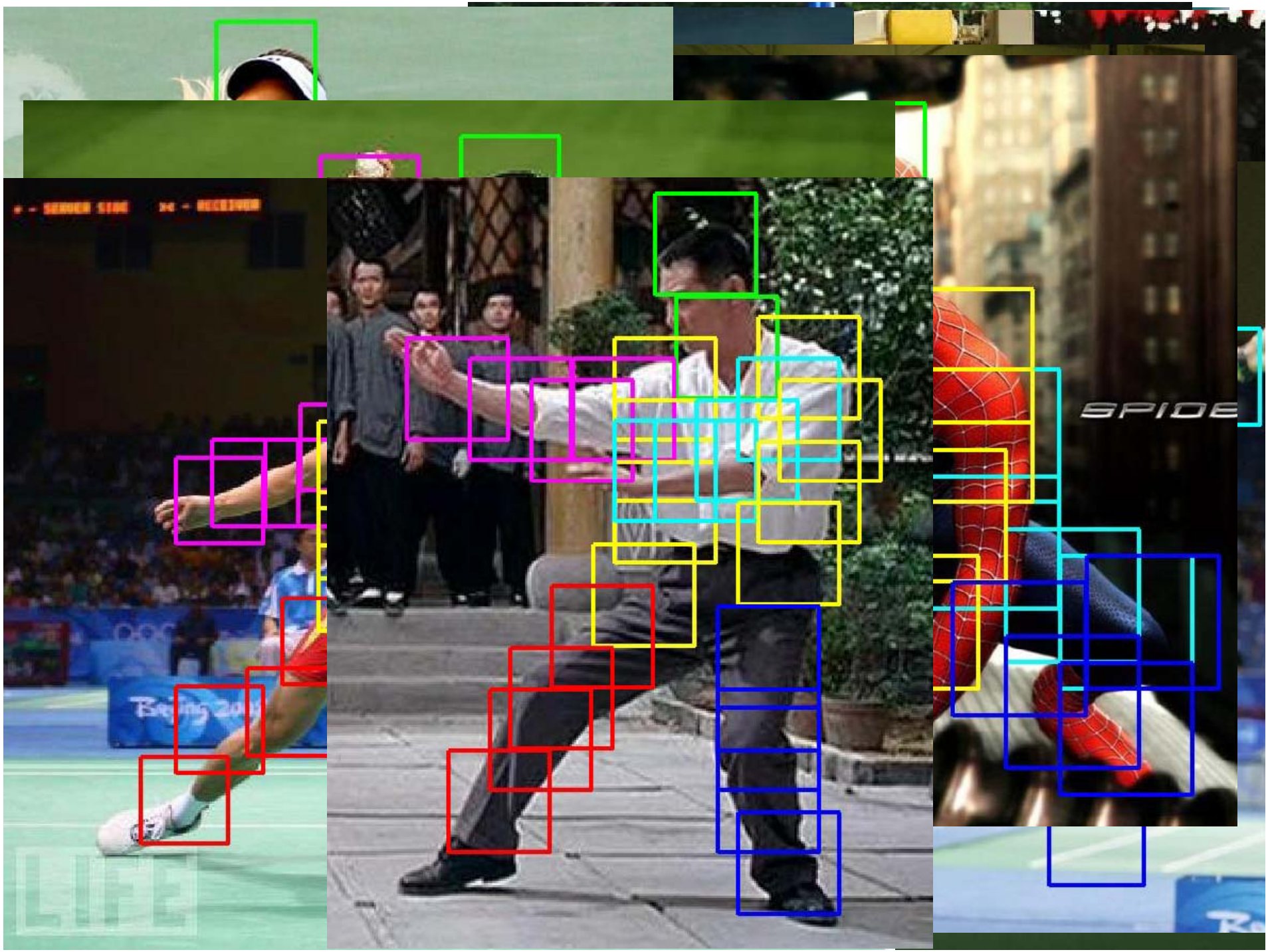
- Model affine warps with a part-based model
- Exponential set of pictorial structures
- Rigid vs flexible relations



Conclusion

- Model affine warps with a part-based model
- Exponential set of pictorial structures
- Rigid vs flexible relations
- Supervision helps





Further ideas:

Human Pose Estimation Using Consistent Max-Covering, Hao Jiang, ICCV 09

Max-margin hidden conditional random fields for human action recognition, Yang Wang and Greg Mori, CVPR 09

Adaptive pose priors for pictorial structures, B. Sapp, C. Jordan, and B. Taskar, CVPR 10