Object Category Detection: Parts-based Models

Computer Vision

CS 543 / ECE 549

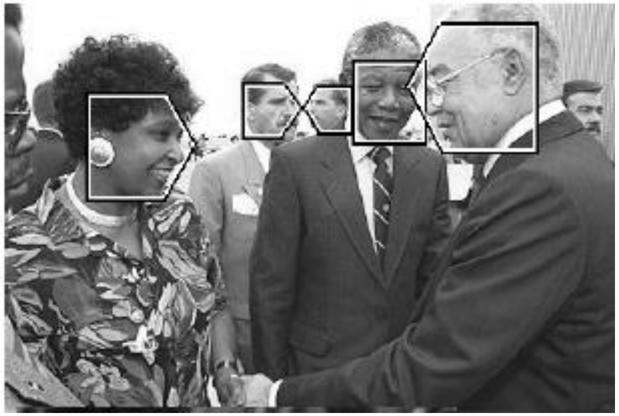
University of Illinois

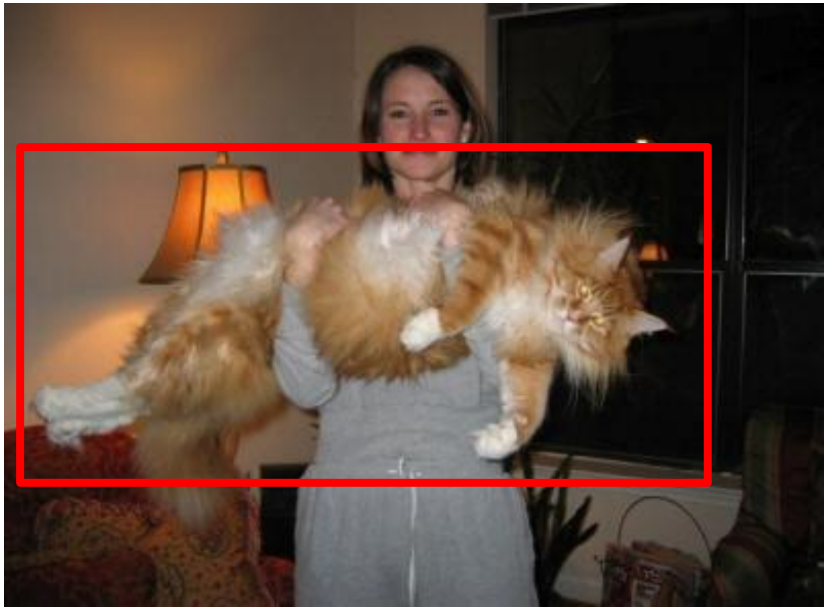
Derek Hoiem

Goal: Detect all instances of objects

Cars

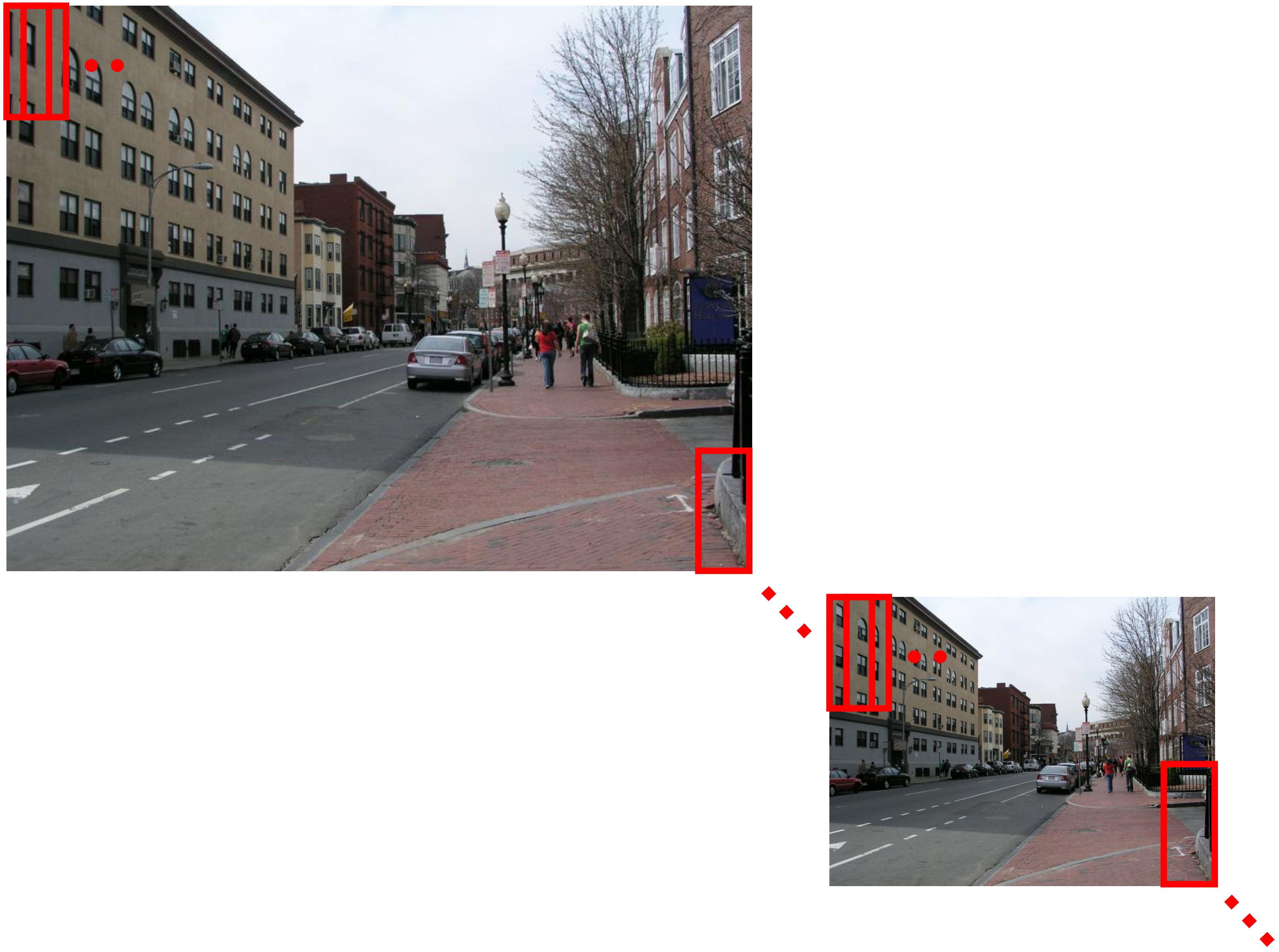
Faces





Cats

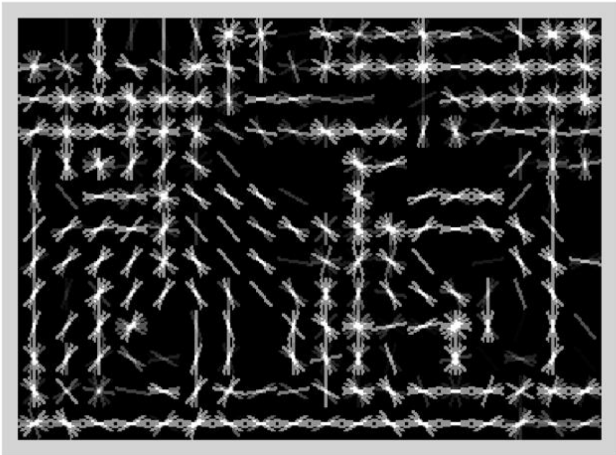
Last class: sliding window detection



Object model: last class

- Statistical Template in Bounding Box
  - Object is some (x,y,w,h) in image
  - Features defined wrt bounding box coordinates

Image



Template Visualization

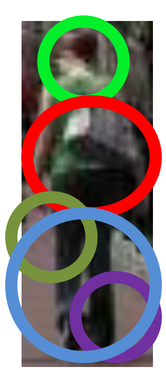
Images from Felzenszwalb

Last class: statistical template

- Object model = log linear model of parts at fixed positions

$$+3 + 2 - 2 - 1 - 2.5 = -0.5 \overset{?}{>} 7.5$$

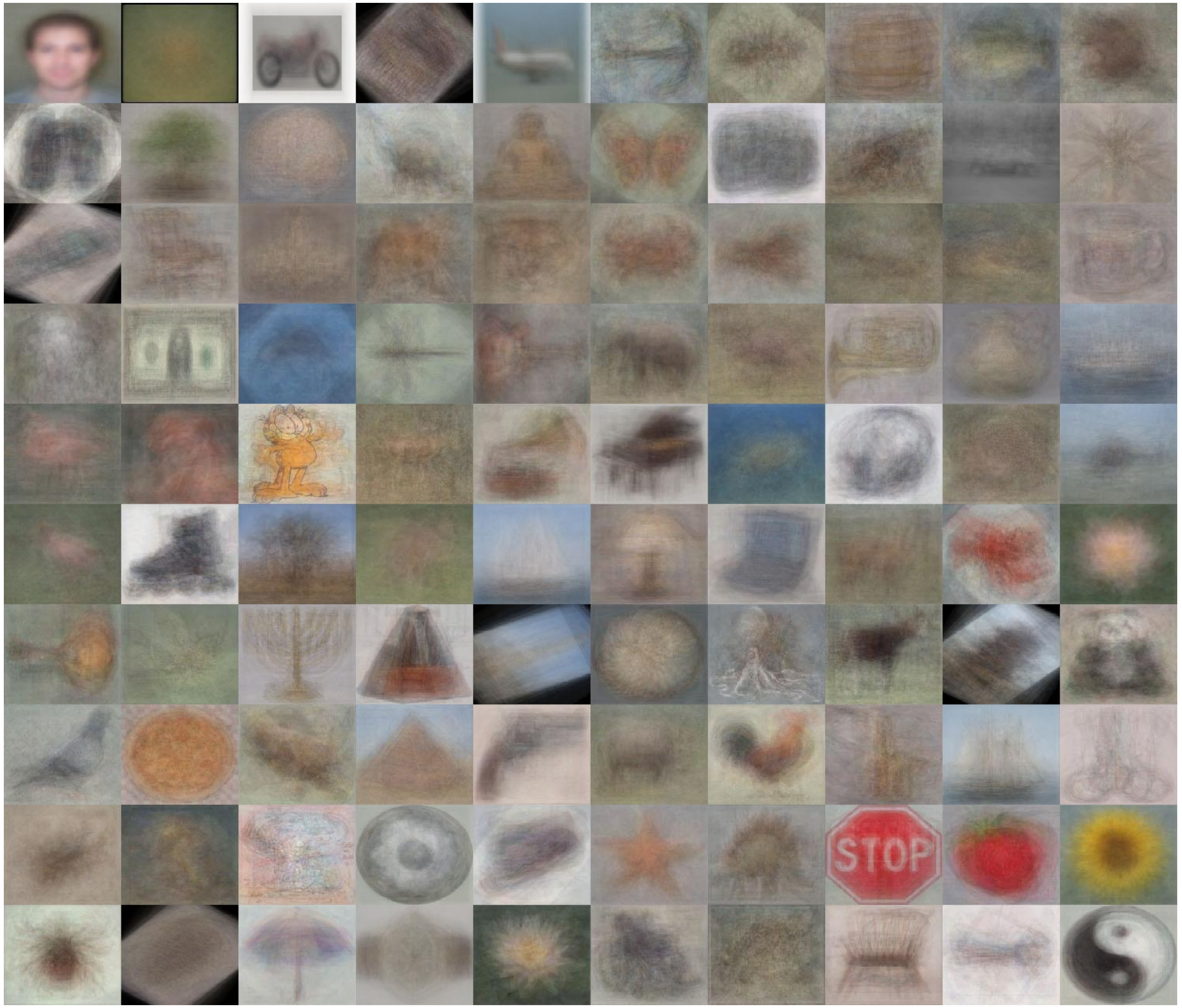
Non-object



$$+4 + 1 + 0.5 + 3 + 0.5 = 10.5 \overset{?}{>} 7.5$$

Object

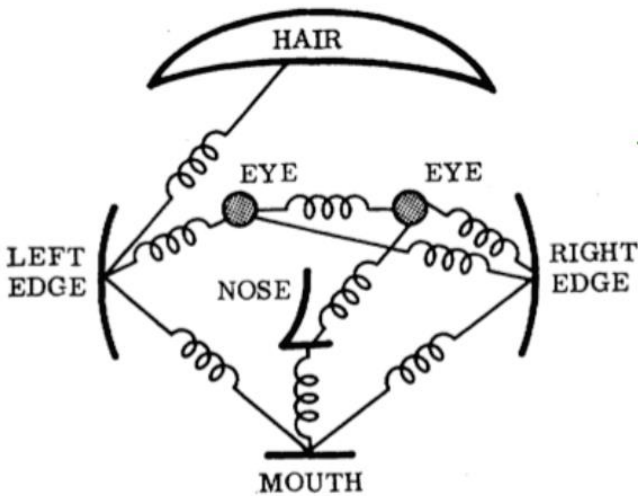
When do statistical templates make sense?

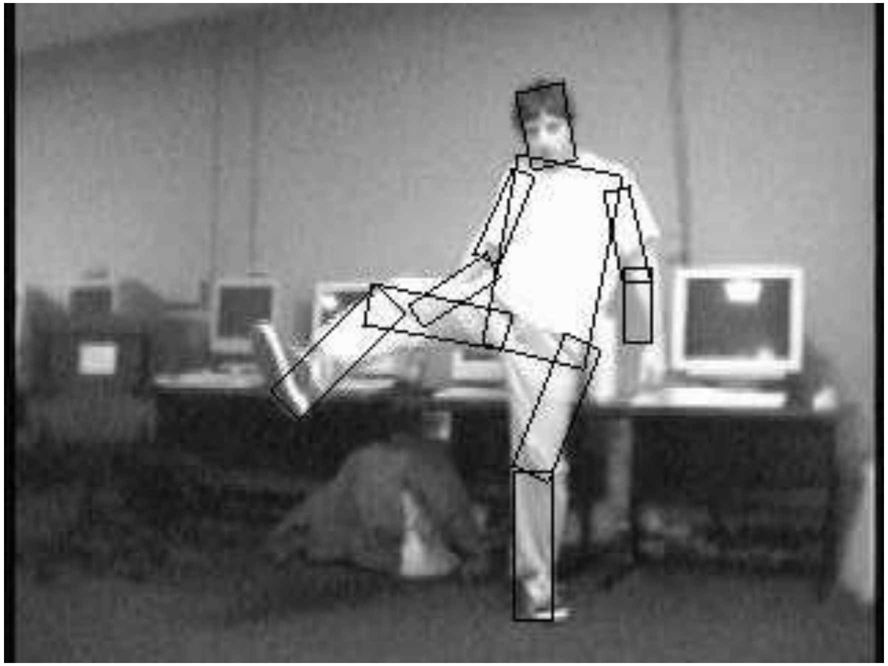


Caltech 101 Average Object Images

Object models: this class

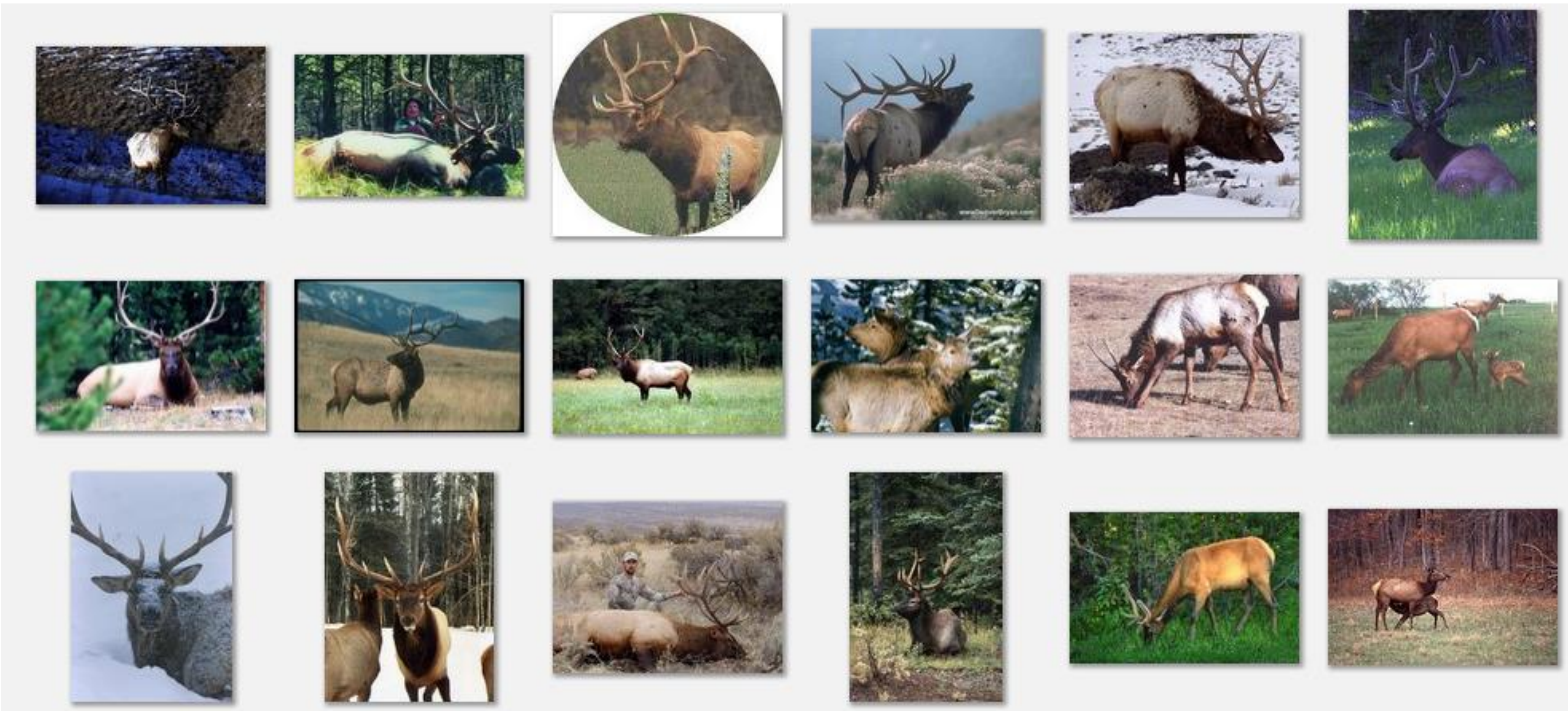
- Articulated parts model
  - Object is configuration of parts
  - Each part is detectable





Images from Felzenszwalb

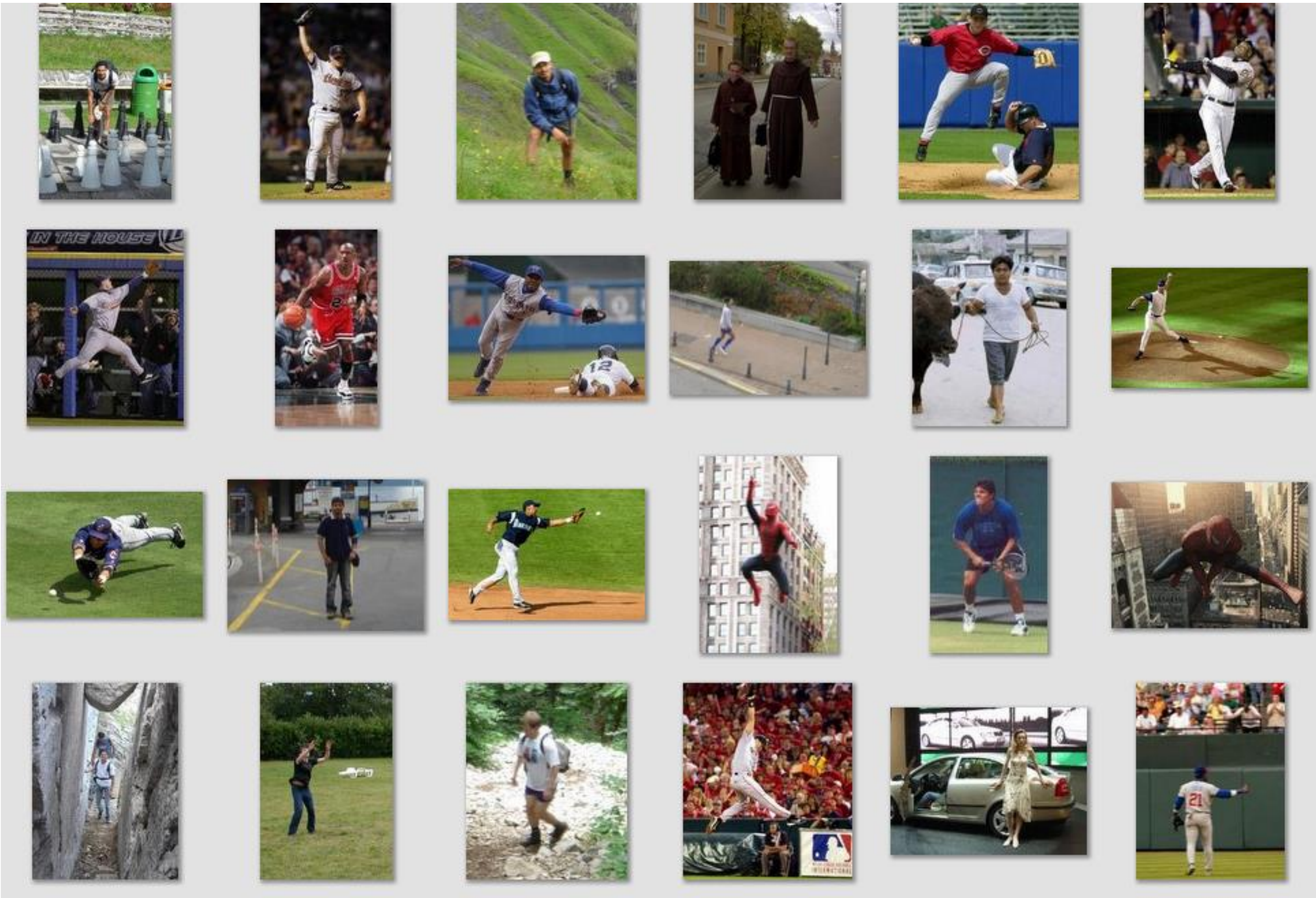
Deformable objects



Images from Caltech-256

Slide Credit: Duan Tran

Deformable objects



Images from D. Ramanan's dataset

Slide Credit: Duan Tran

Compositional objects





Parts-based Models

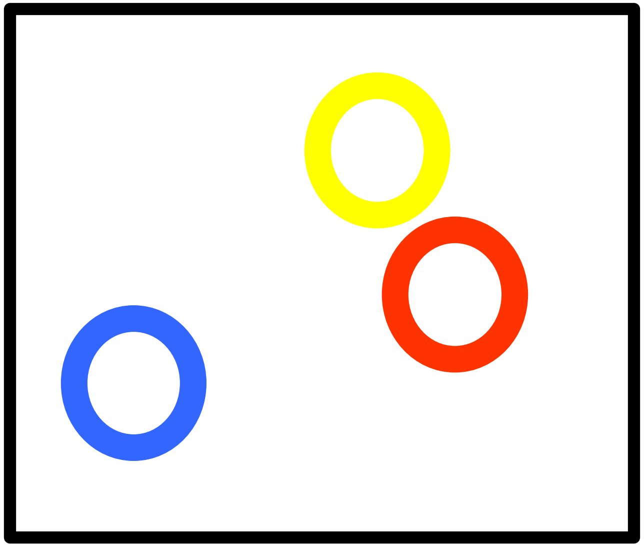
Define object by collection of parts modeled by

1. Appearance
2. Spatial configuration

Slide credit: Rob Fergus

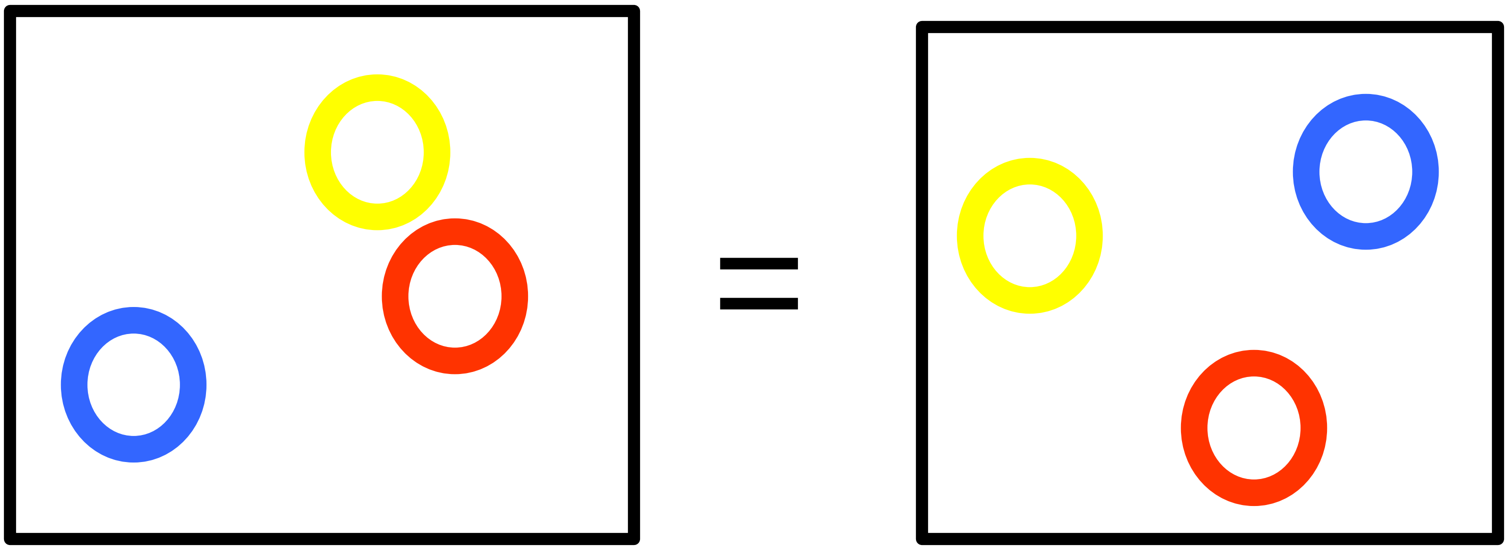
How to model spatial relations?

- One extreme: fixed template



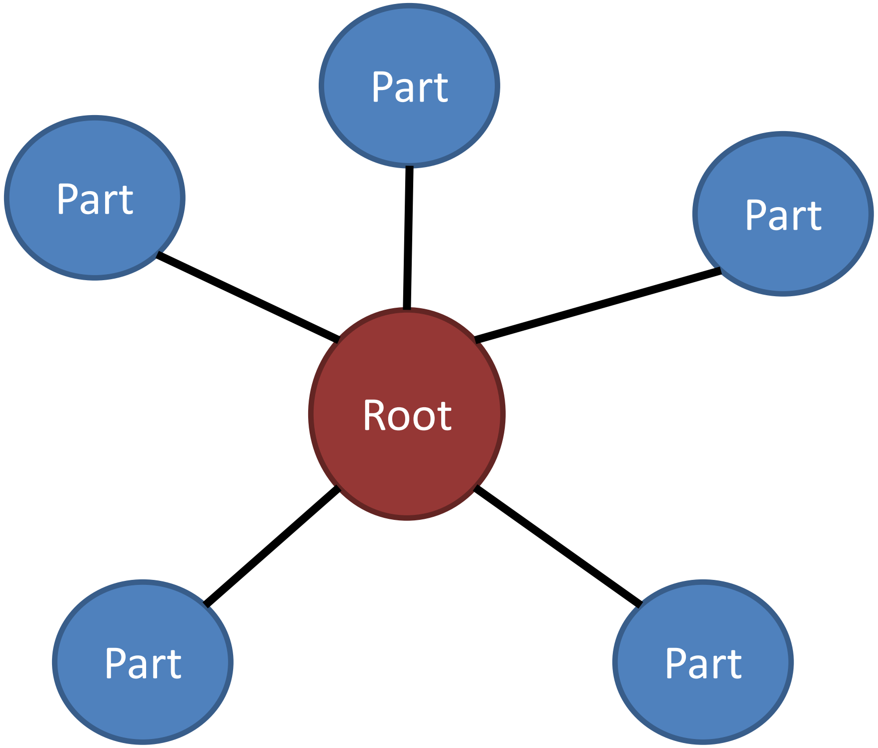
How to model spatial relations?

- Another extreme: bag of words



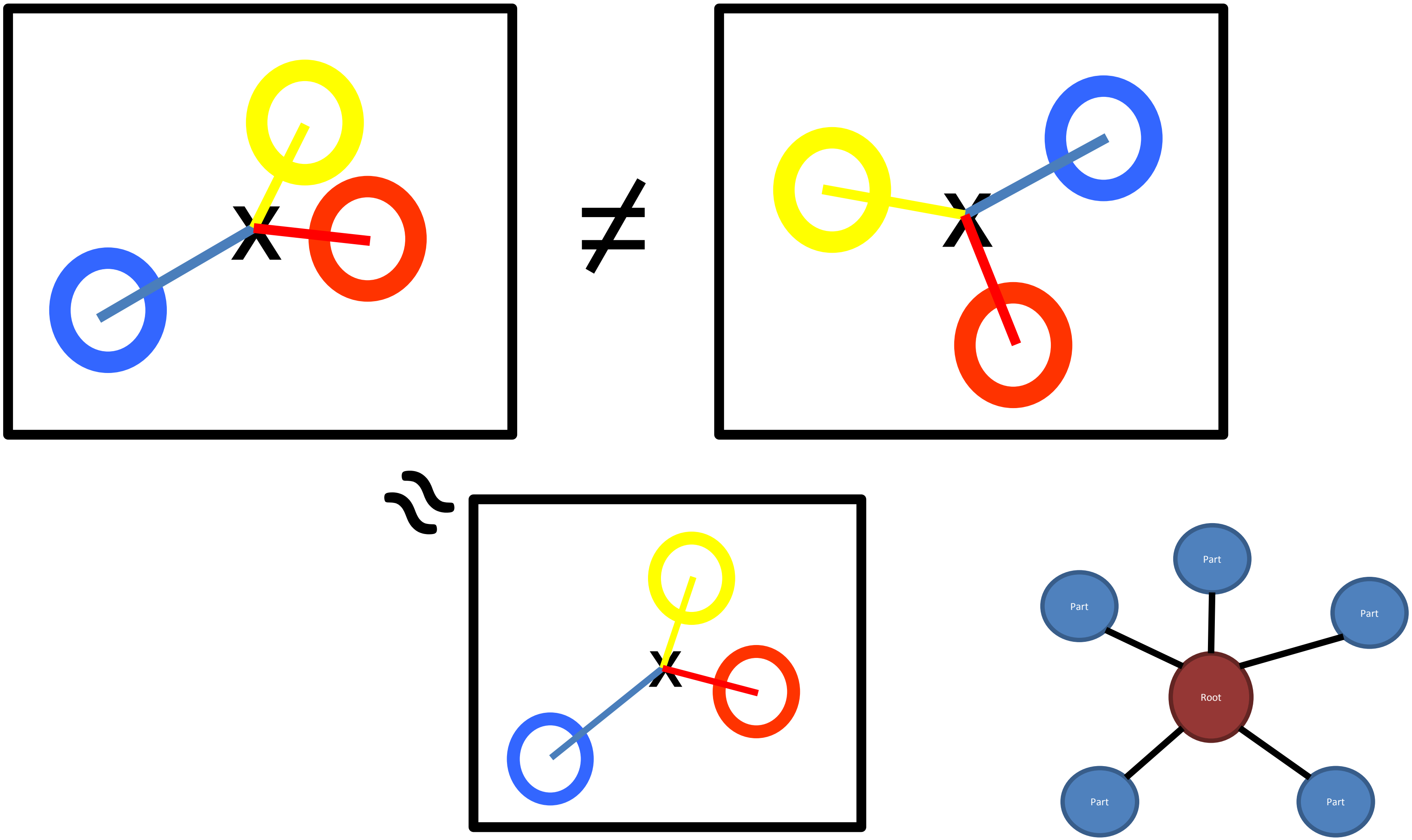
How to model spatial relations?

- Star-shaped model



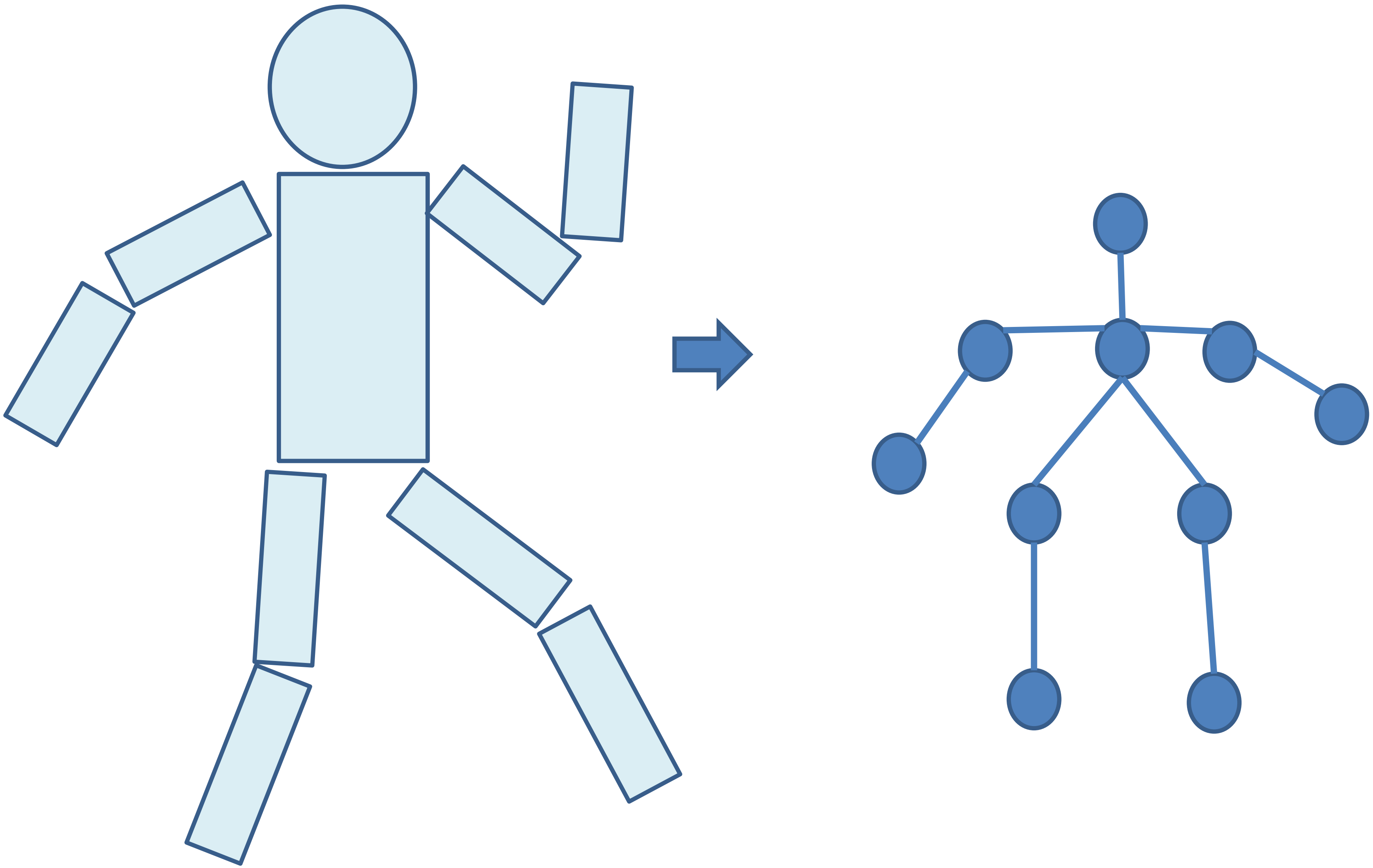
How to model spatial relations?

- Star-shaped model



How to model spatial relations?

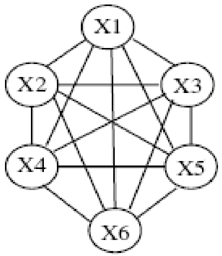
- Tree-shaped model



How to model spatial relations?

- Many others...

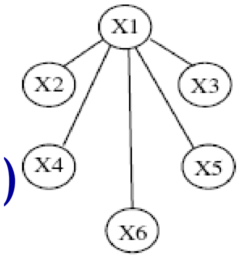
$O(N^6)$



a) Constellation

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

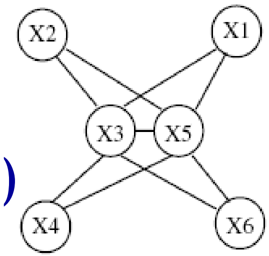
$O(N^2)$



b) Star shape

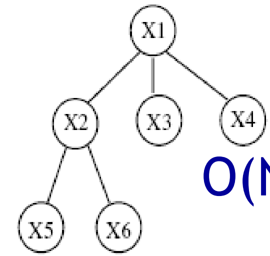
Leibe et al. '04, '08
Crandall et al. '05
Fergus et al. '05

$O(N^3)$



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

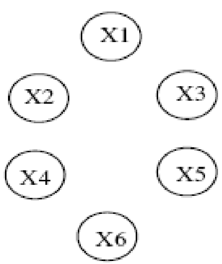
Crandall et al. '05



$O(N^2)$

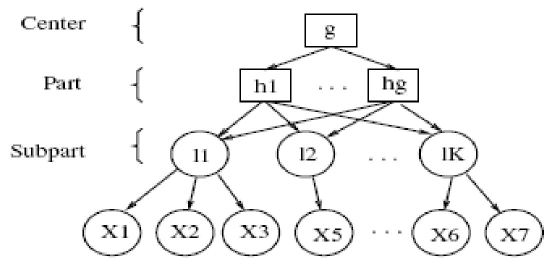
d) Tree

Felzenszwalb &
Huttenlocher '05



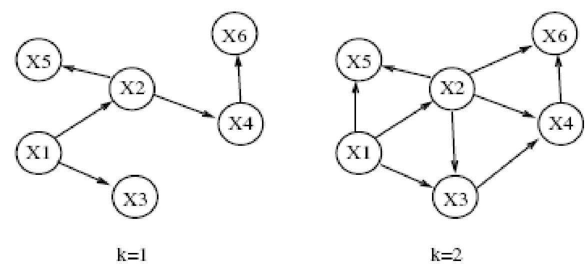
e) Bag of features

Csurka '04
Vasconcelos '00



f) Hierarchy

Bouchard & Triggs '05



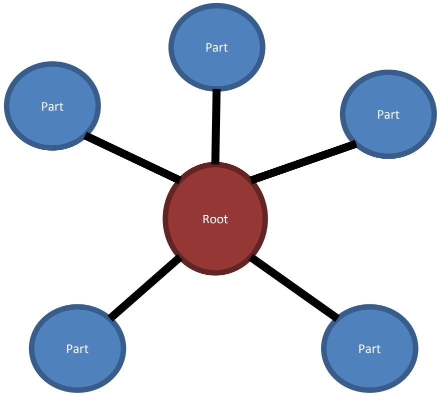
g) Sparse flexible model

Carneiro & Lowe '06

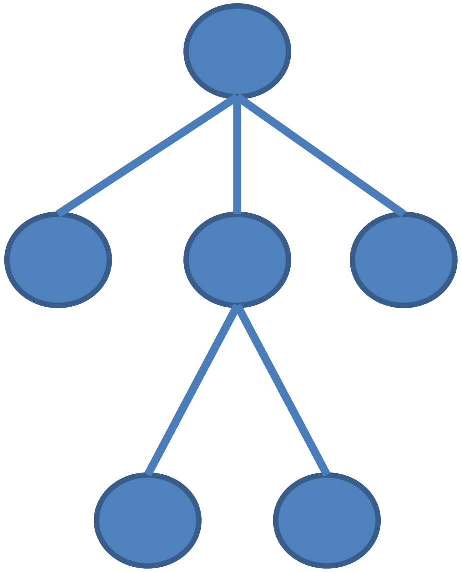
from [Carneiro & Lowe, ECCV'06]

Today's class

1. Star-shaped model
   – Example: Deformable Parts Model
      • [Felzenswalb et al. 2010]



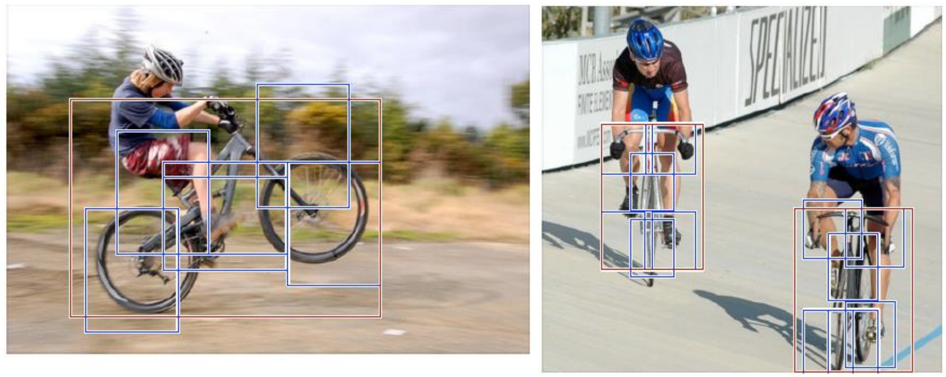
2. Tree-shaped model
   – Example: Pictorial structures
      • [Felzenszwalb Huttenlocher 2005]



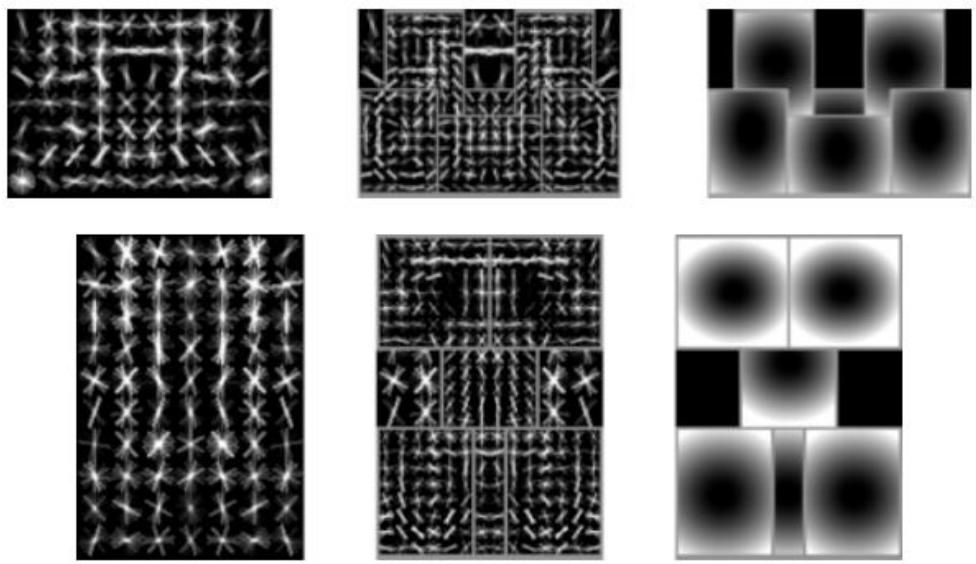
3. Sequential prediction models

Deformable Latent Parts Model (DPM)

Detections



Template Visualization



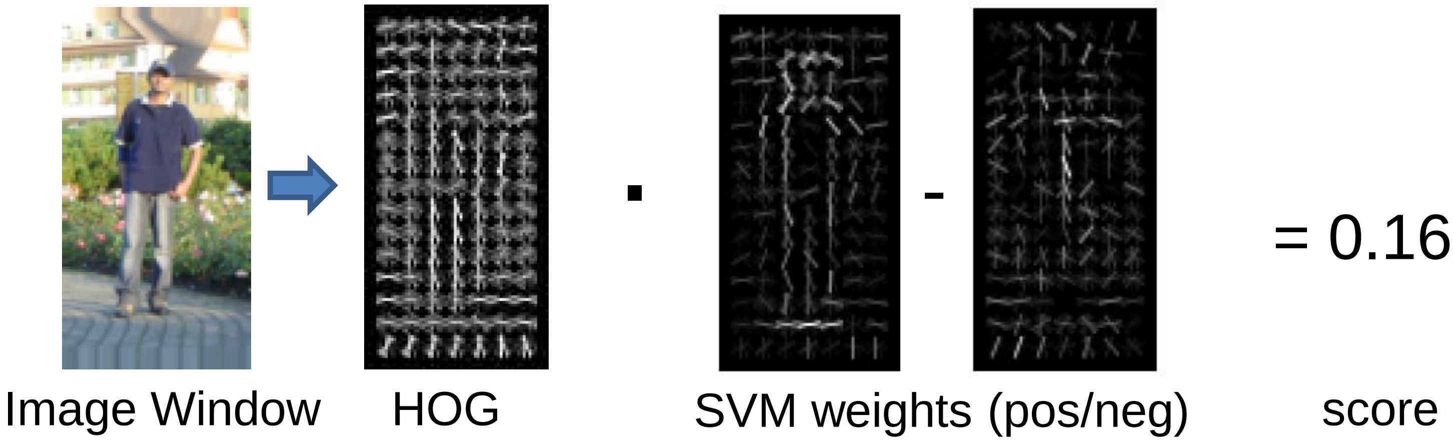
root filters
coarse resolution

part filters
finer resolution

deformation
models

Felzenszwalb et al. 2008, 2010

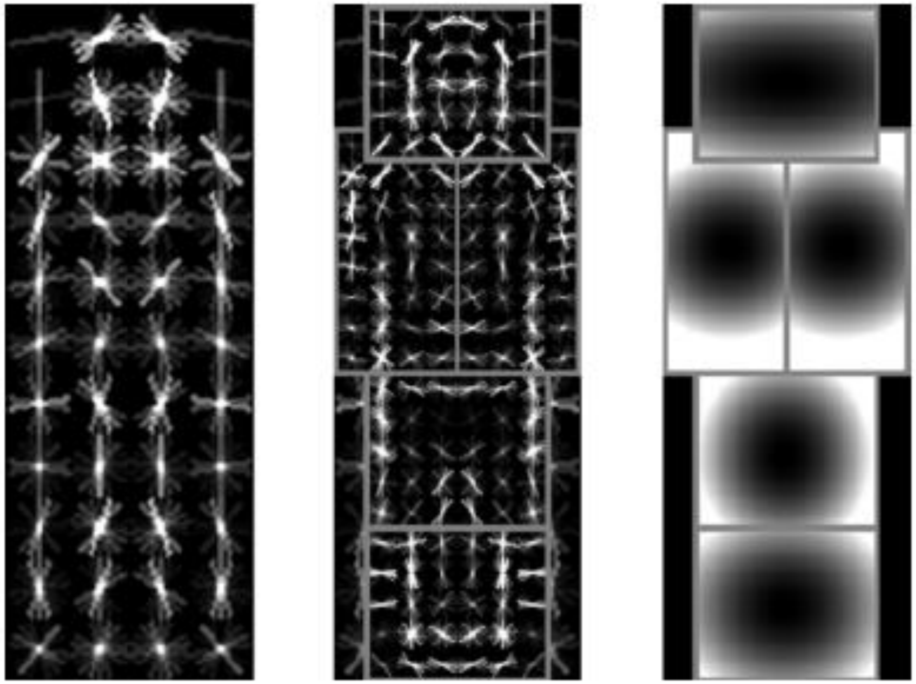
Review: Dalal-Triggs detector



1. Extract fixed-sized (64x128 pixel) window at each position and scale
2. Compute HOG (histogram of gradient) features within each window
3. Score the window with a linear SVM classifier
4. Perform non-maxima suppression to remove overlapping detections with lower scores

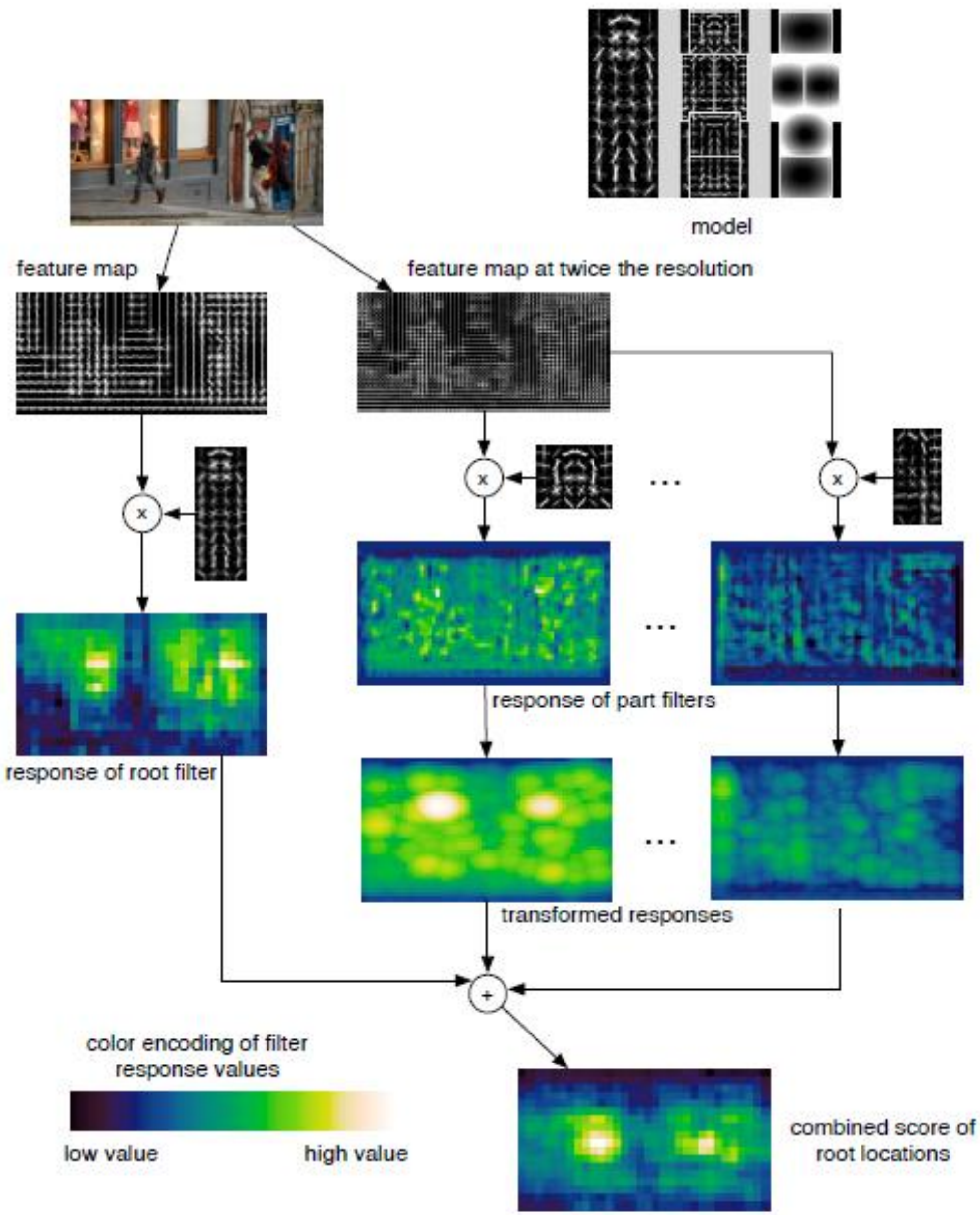
Deformable parts model

- Root filter models coarse whole-object appearance
- Part filters model finer-scale appearance of smaller patches
- For each root window, part positions that maximize appearance score minus spatial cost are found
- Total score is sum of scores of each filter and spatial costs



Root filter Part filters Spatial costs

DPM: computing object score



Scores from root detector

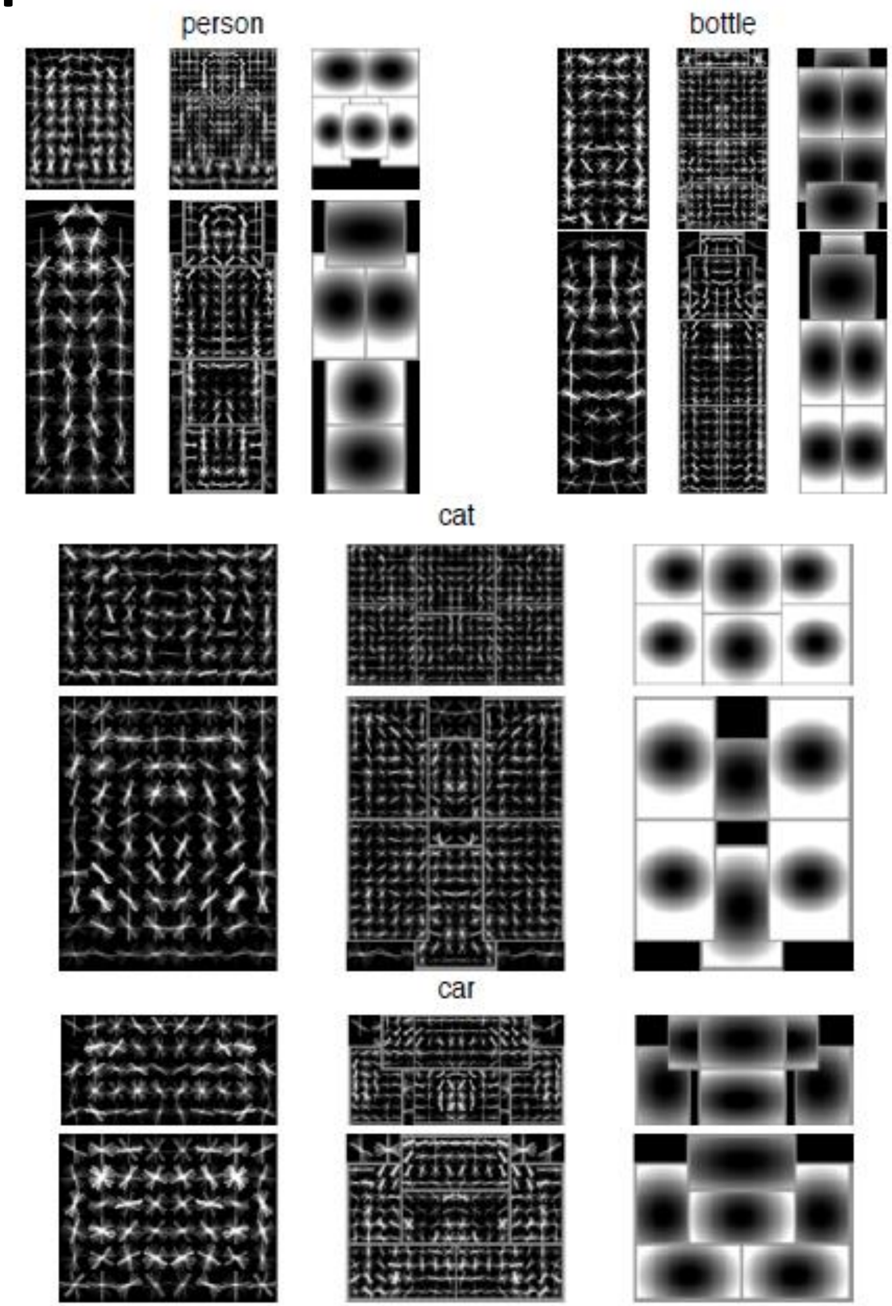
Scores from individual part detectors

With generalized distance transform, compute the maximum part score corresponding to each root position

DPM: mixture model

- Each positive example is modeled by one of M detectors

- In testing, all detectors are applied with non-max suppression



DPM: Training

$F_n := \emptyset$
for *relabel* := 1 **to** *num-relabel* **do**
  $F_p := \emptyset$
  for $i := 1$ **to** n **do**
    Add `detect-best` (β, I_i, B_i) to F_p
  end
  for *datamine* := 1 **to** *num-datamine* **do**
    for $j := 1$ **to** m **do**
      if $|F_n| \geq$ *memory-limit* **then break**
      Add `detect-all` $(\beta, J_j, -(1 + \delta))$ to F_n
    end
    $\beta :=$ `gradient-descent` $(F_p \cup F_n)$
    Remove (i, v) with $\beta \cdot v < -(1 + \delta)$ from F_n
  end
end

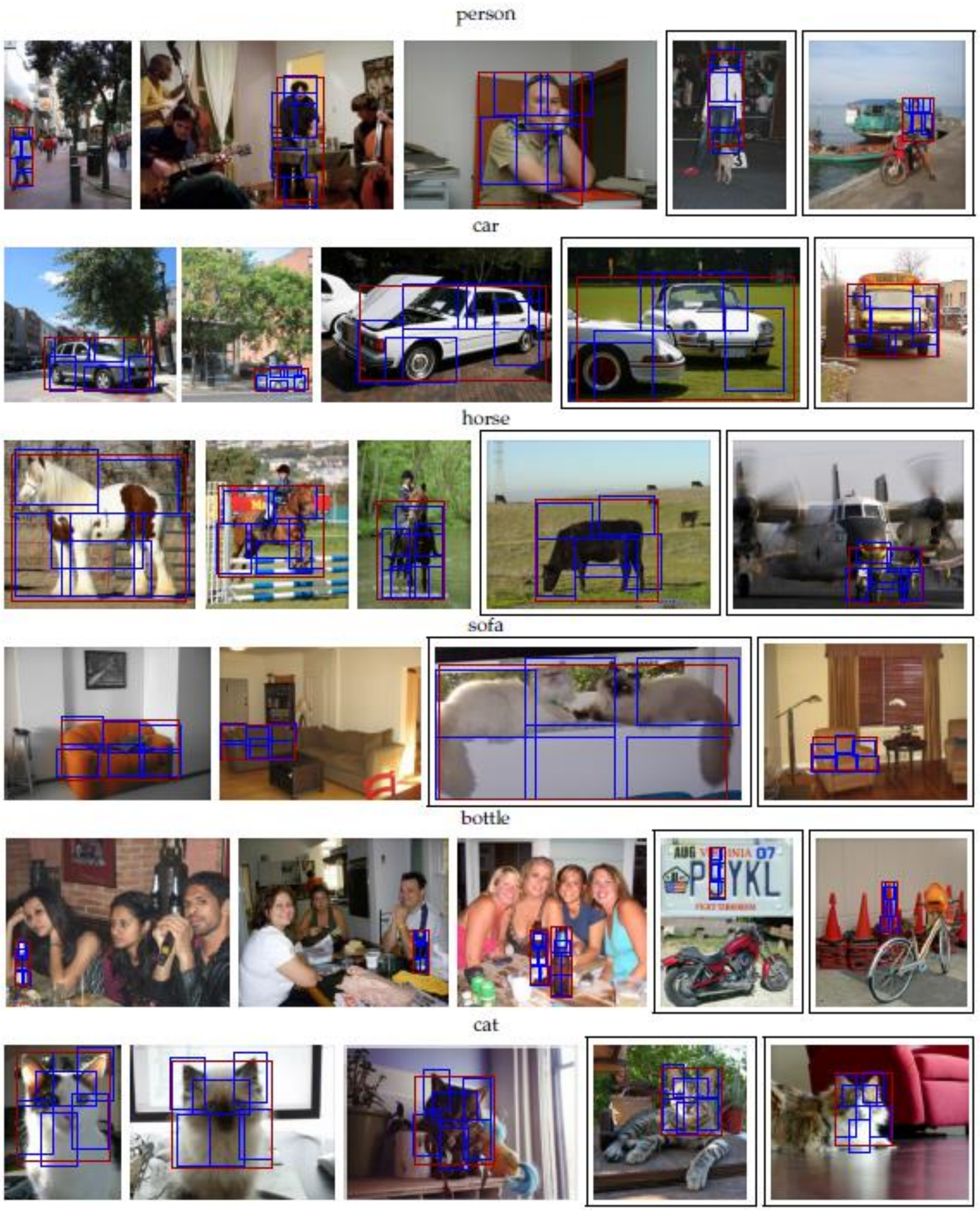
Procedure `Train`

Solve for latent parameters (root/part positions, mixture component) that maximize score and are consistent with ground truth bounding box

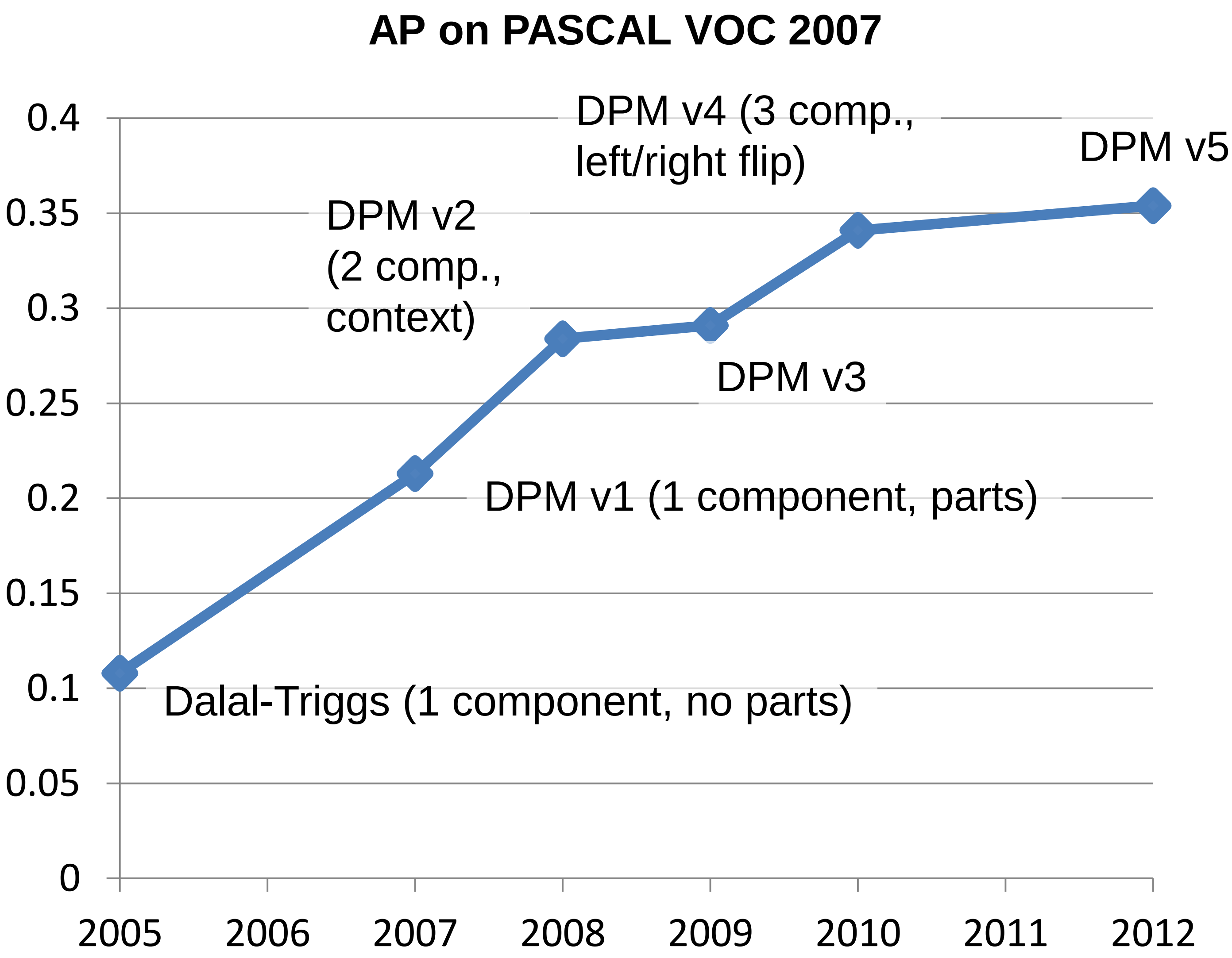
Add negative examples that achieve some minimum score (> 1 – delta)

Solve for SVM weights given current latent parameters and negative examples

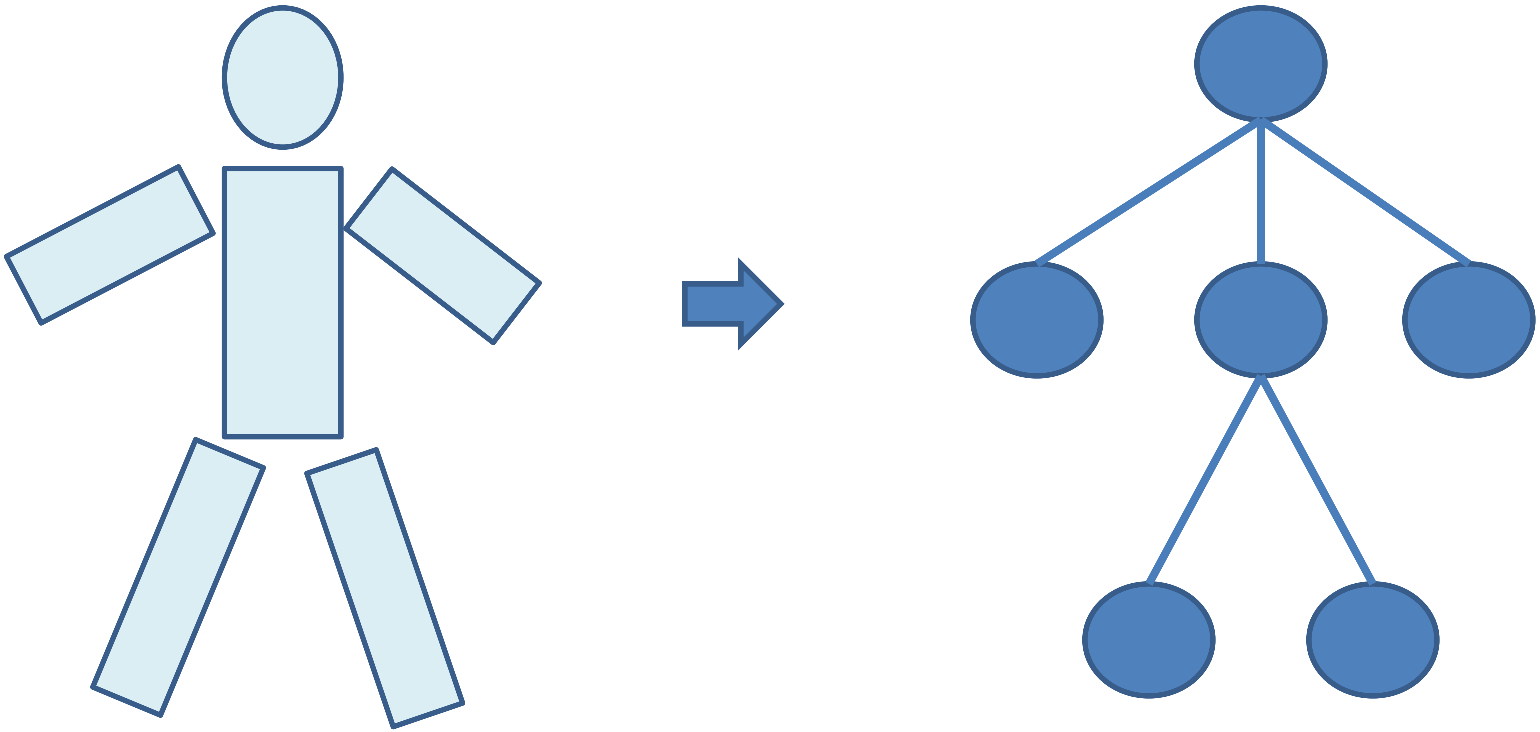
Results



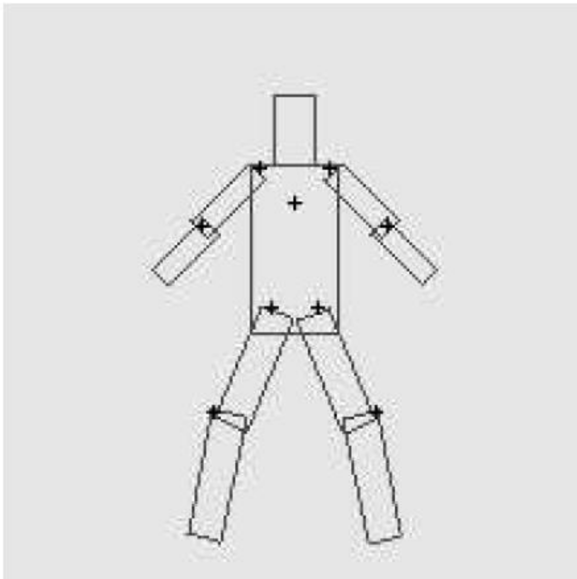
Improvement over time for HOG-based detectors

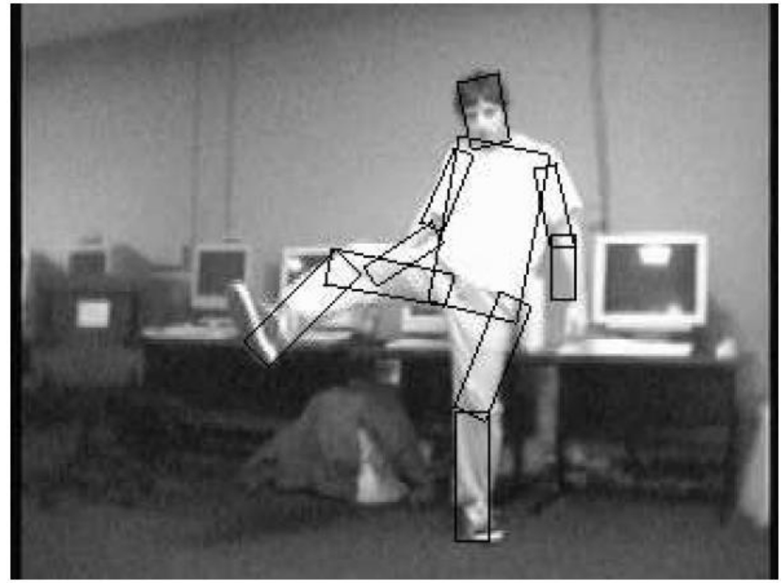


Tree-shaped model



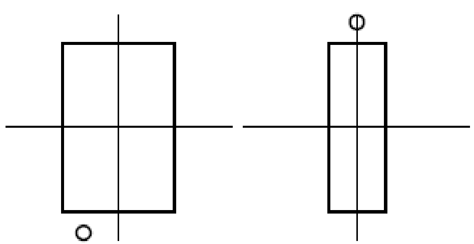
Pictorial Structures Model



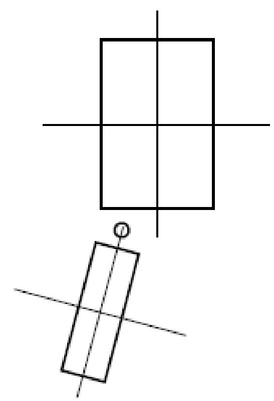


Part = oriented rectangle

Spatial model = relative size/orientation



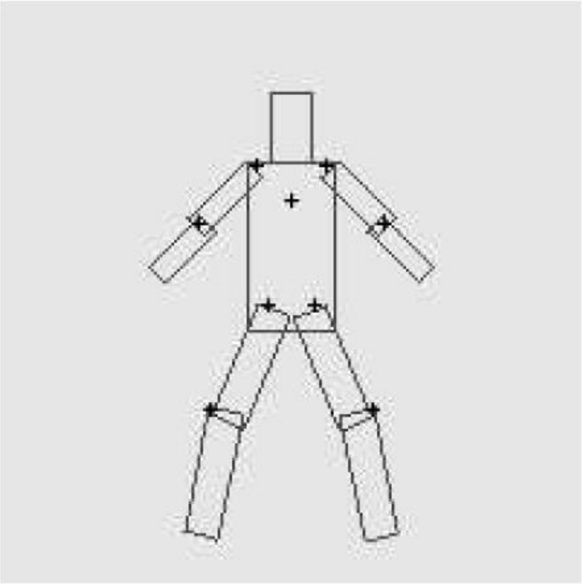
a



b

Felzenszwalb and Huttenlocher 2005

Pictorial Structures Model



$$P(L|I,\theta) \propto \left(\prod_{i=1}^{n} p(I|l_i, u_i) \prod_{(v_i,v_j)\in E} p(l_i, l_j|c_{ij}) \right)$$

Appearance likelihood

Geometry likelihood

Modeling the Appearance

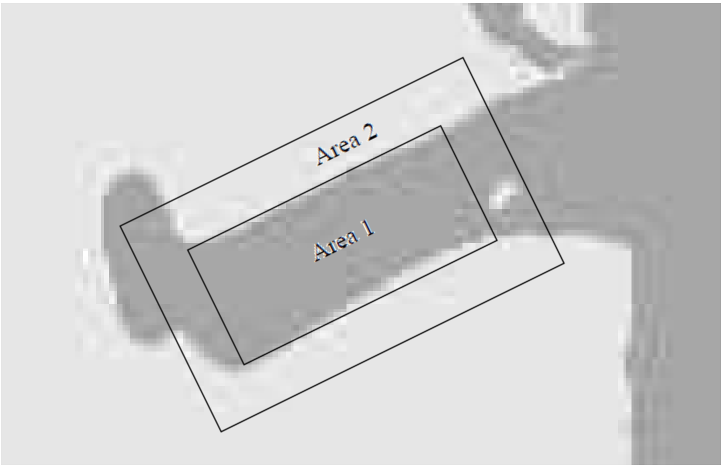
- Any appearance model could be used
  - HOG Templates, etc.
  - Here: rectangles fit to background subtracted binary map

- Can train appearance models independently (easy, not as good) or jointly (more complicated but better)

$$P(L|I,\theta) \propto \left(\prod_{i=1}^{n} \underbrace{p(I|l_i, u_i)}_{\text{Appearance likelihood}} \prod_{(v_i,v_j)\in E} \underbrace{p(l_i, l_j|c_{ij})}_{\text{Geometry likelihood}} \right)$$

Part representation

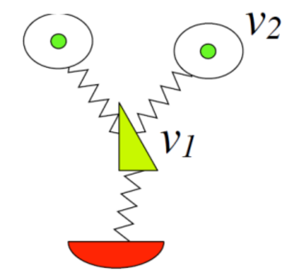
- Background subtraction



Pictorial structures model

Optimization is tricky but can be efficient

$$L^* = \arg\min_L \left(\sum_{i=1}^{n} m_i(l_i) + \sum_{(v_i,v_j)\in E} d_{ij}(l_i, l_j) \right)$$



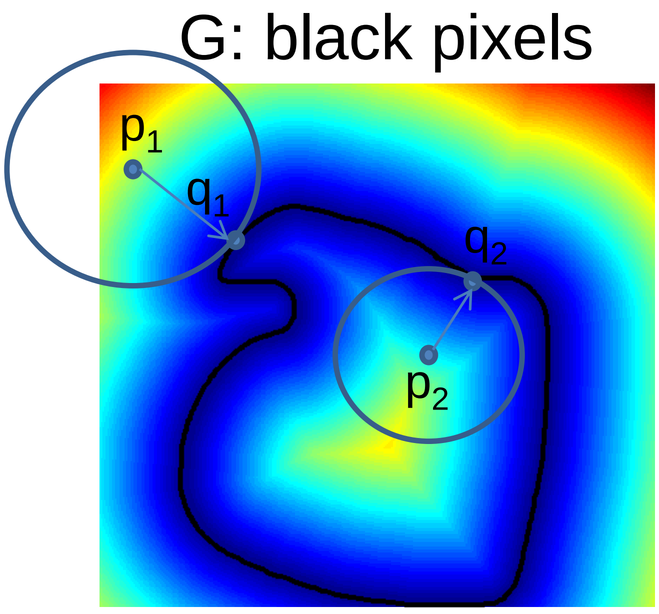
- For each l_1, find best l_2:

$$\text{Best}_2(l_1) = \min_{l_2} \left[m_2(l_2) + d_{12}(l_1, l_2) \right]$$

- Remove v_2, and repeat with smaller tree, until only a single part
- For k parts, n locations per part, this has complexity of $O(kn^2)$, but can be solved in ~$O(kn)$ using generalized distance transform

Distance Transform

- For each pixel p, how far away is the nearest pixel q of set G
  - $f(p) = \min_{q \in G} \; d(p, q)$
  - G is often the set of edge pixels



Distance Transform - Applications

- Set distances – e.g. Hausdorff Distance
- Image processing – e.g. Blurring
- Robotics – Motion Planning
- Alignment
  - Edge images
  - Motion tracks
  - Audio warping
- *Deformable Part Models*

Generalized Distance Transform

- Original form: $f(p) = \min_{q \in G} \; d(p,q)$
- General form: $f(p) = \min_{q \in [1,N]} \; m(q) + d(p,q)$

- For many deformation costs, $O(N^2) \rightarrow O(N)$

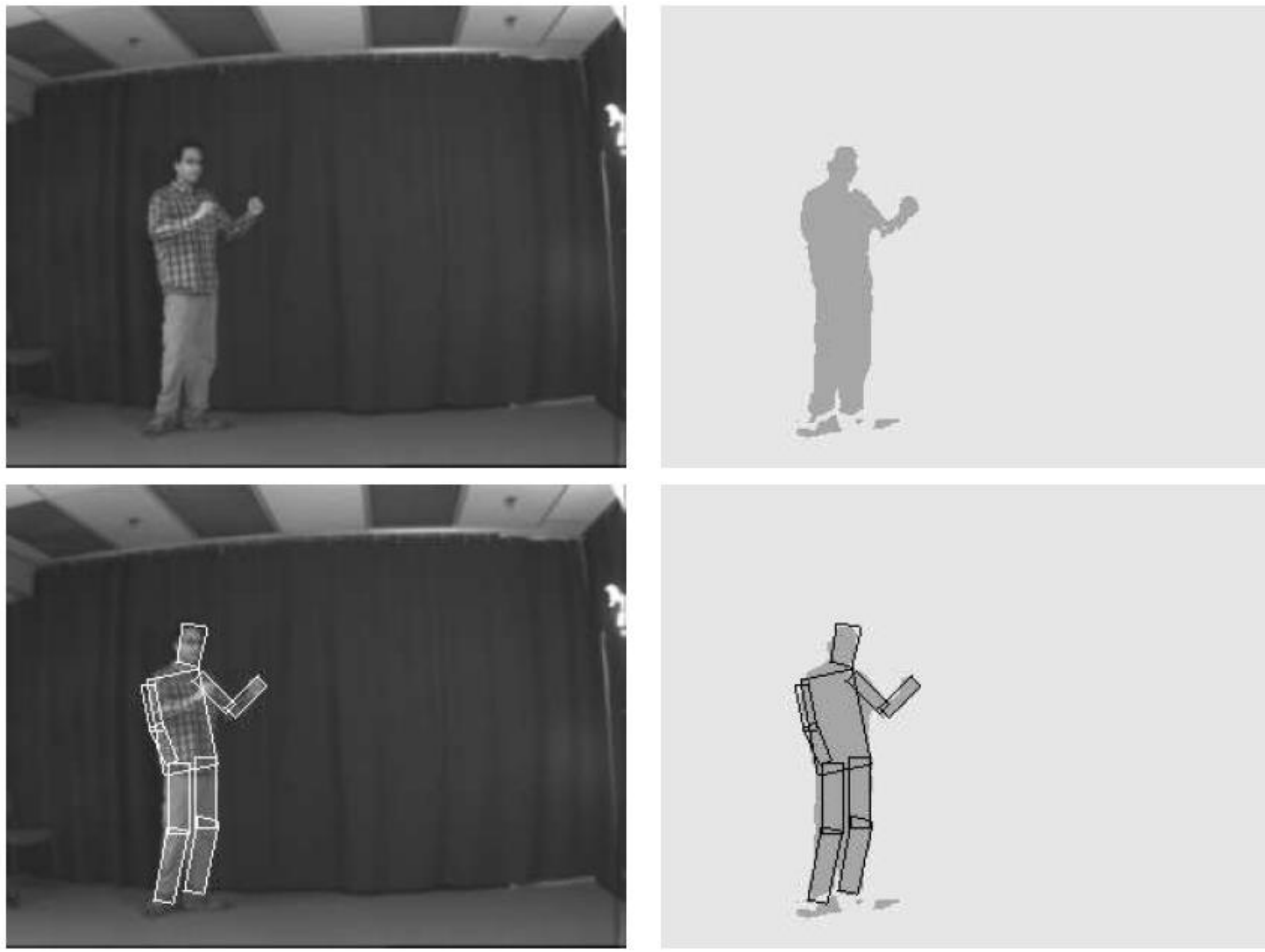
Quadratic $$d(p,q) = \alpha(p-q)^2 + \beta(p-q)$$

Abs Diff $$d(p,q) = \alpha|p-q|$$

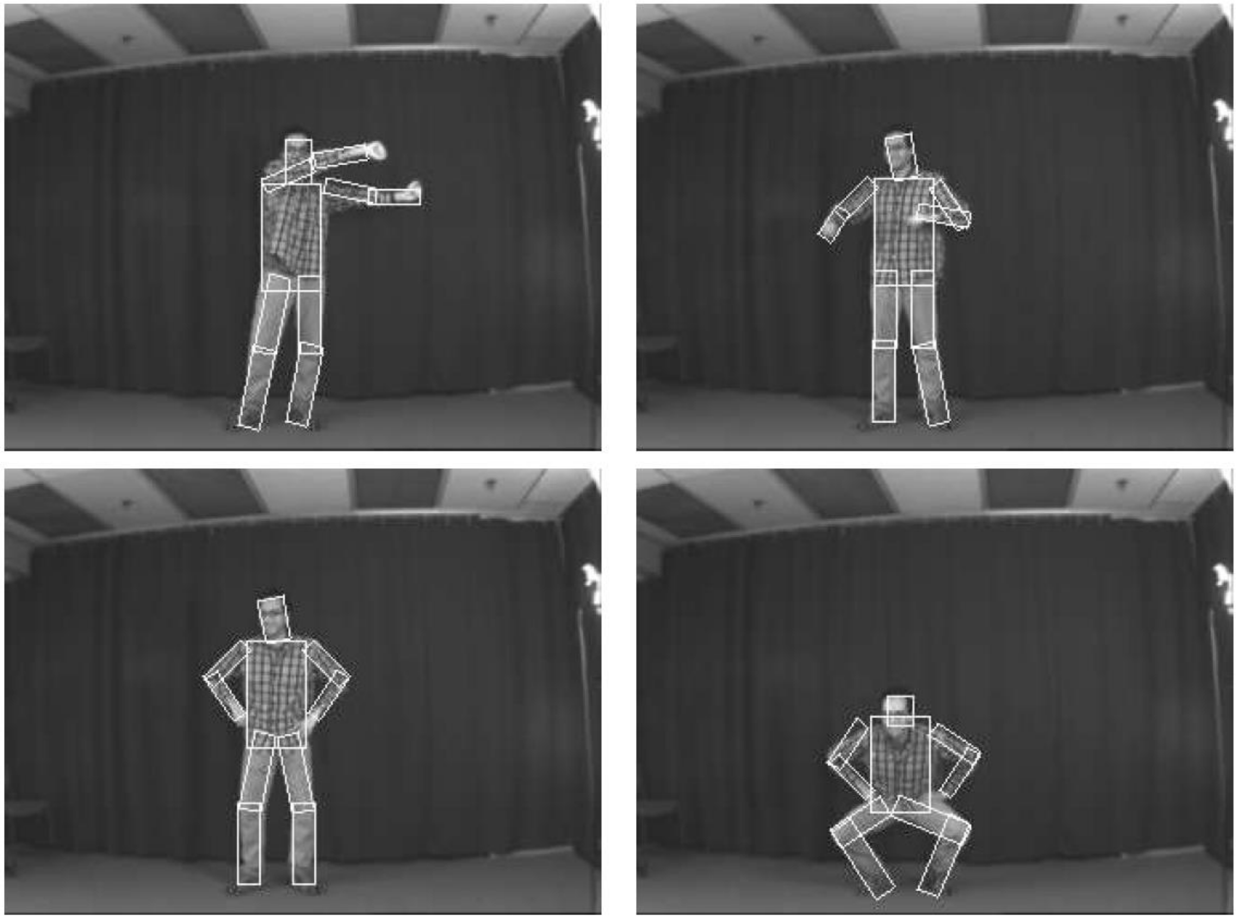
Min Composition $$d(p,q) = \min(d_1(p,q), d_2(p,q))$$

Bounded $$d_\tau(p,q) = \begin{cases} d(p,q) & : |p-q| < \tau \\ \infty & : |p-q| \geq \tau \end{cases}$$

Results for person matching



Results for person matching



Enhanced pictorial structures

- Learn spatial prior
- Color models from soft segmentation (initialized by location priors of each part)

BMVC 2009

2 minute break

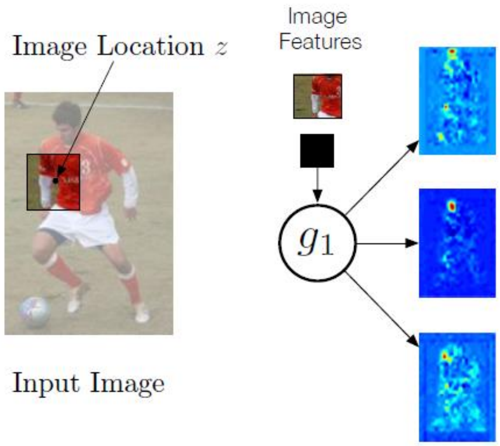
Which patch corresponds to a body part?

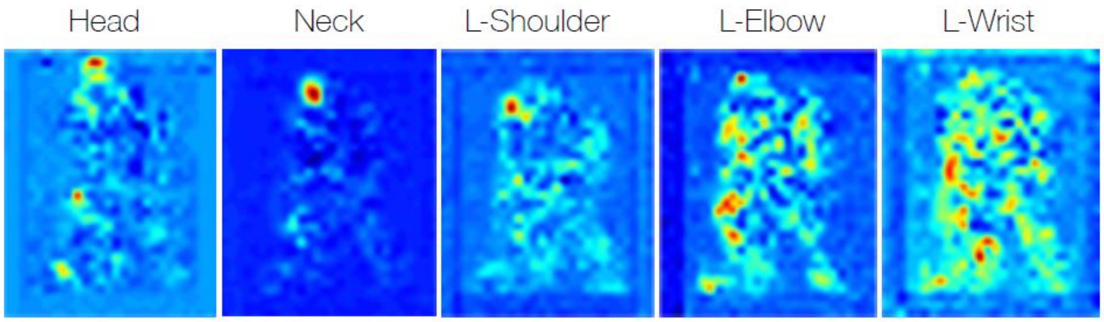
Example from Ramakrishna

Sequential structured prediction

- Can consider pose estimation as predicting a set of related variables (called structured prediction)
  - Some parts easy to find (head), some are hard (wrists)
- One solution: jointly solve for most likely variables (DPM, pictorial structures)
- Another solution: iteratively predict each variable based in part on previous predictions

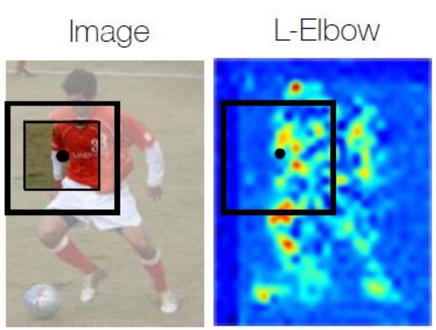
Pose machines

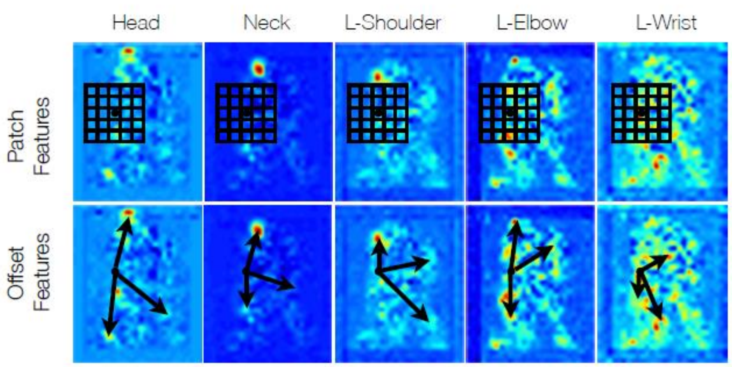




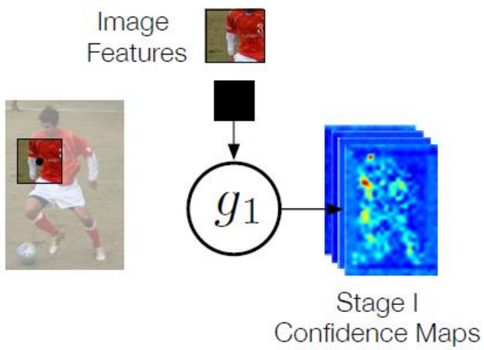
Local image evidence is weak

Certain parts are easier to detect than others

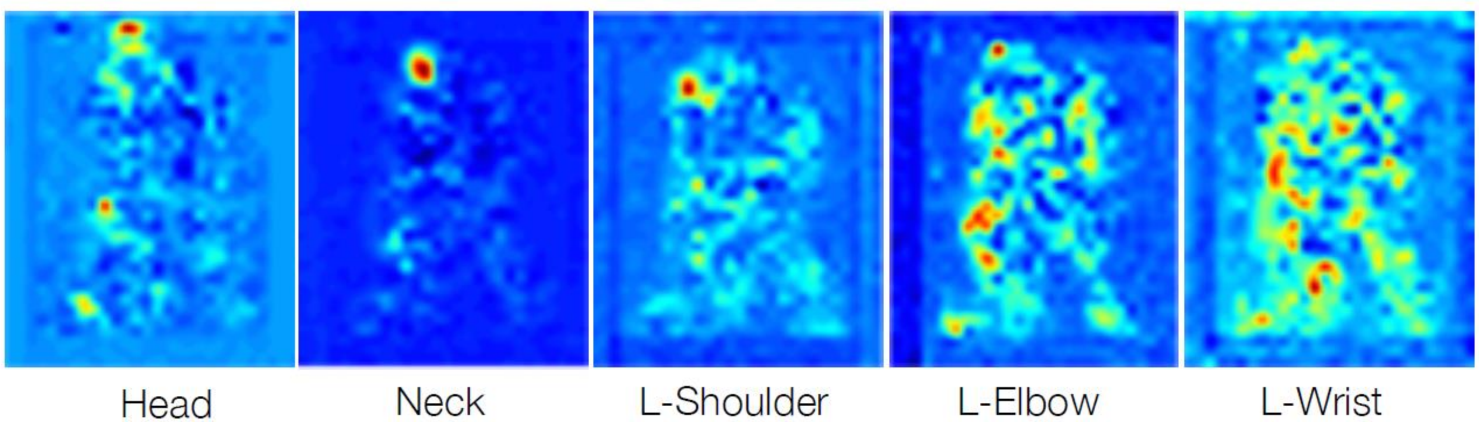


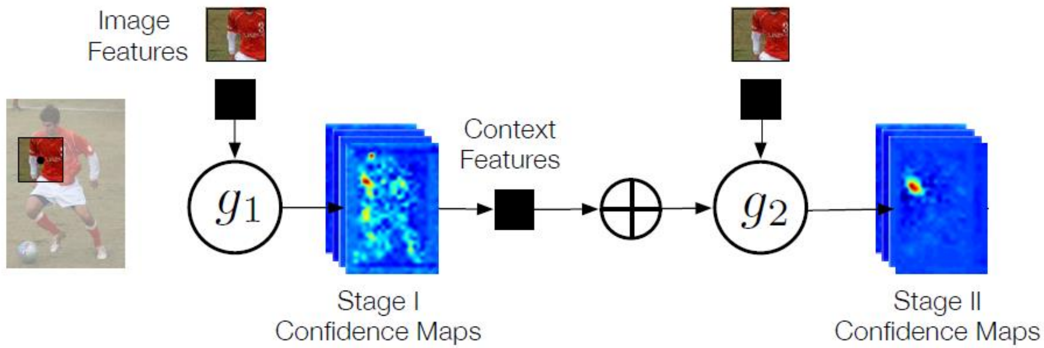


Ramakrishna et al. ECCV 2014

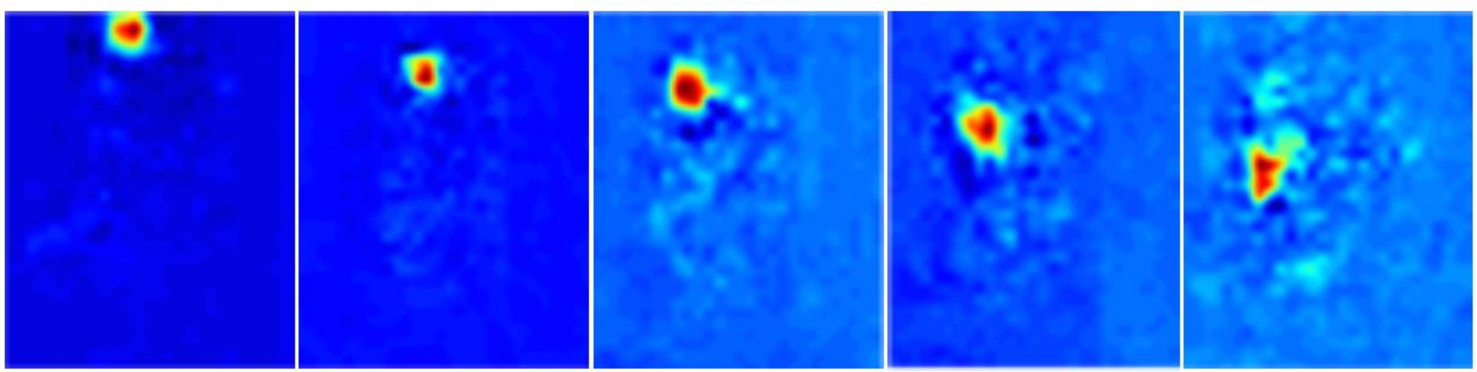


Stage I Confidence





Stage II Confidence



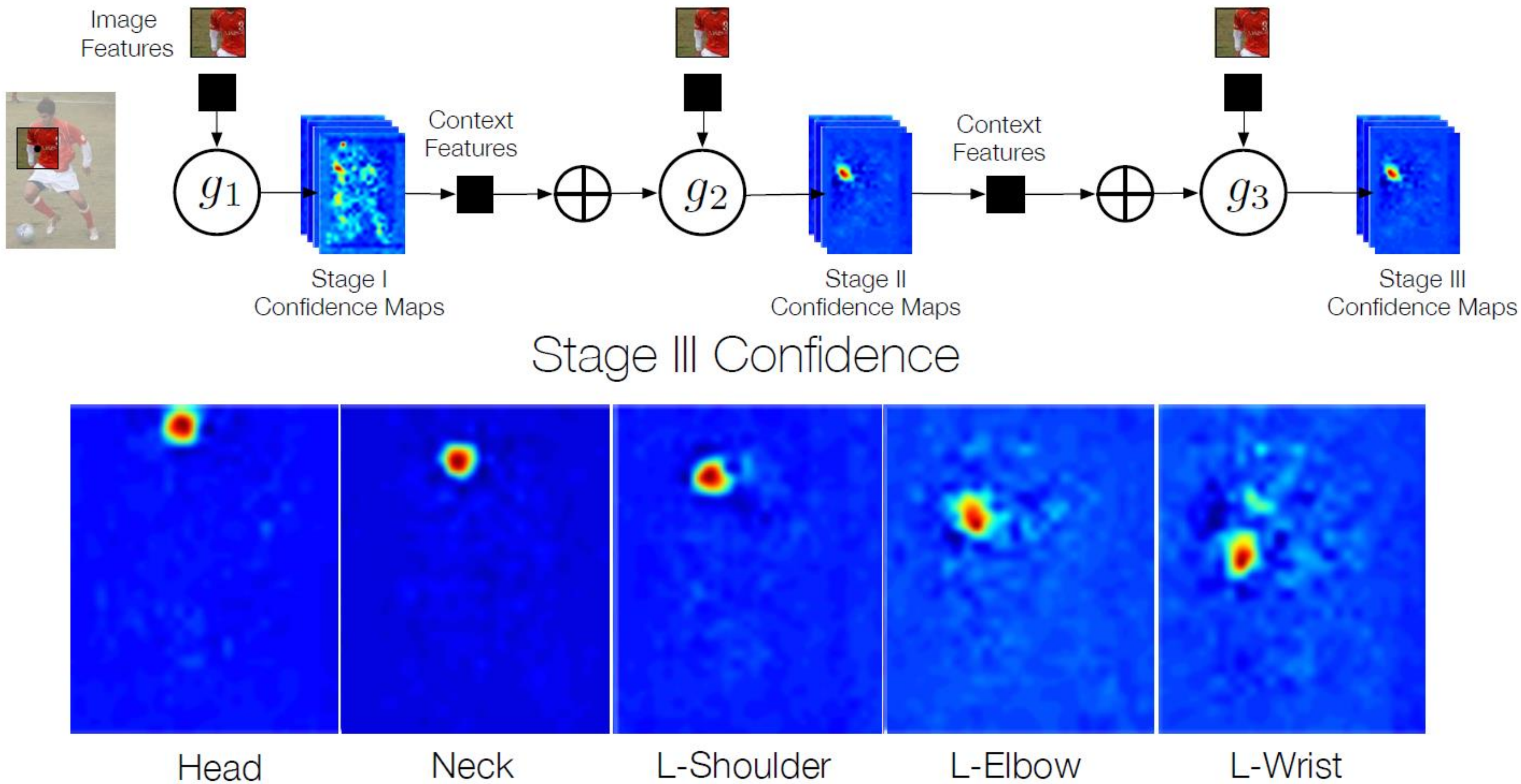


Image
Features
g_1
Stage I
Confidence Maps
Context
Features
g_2
Stage II
Confidence Maps
Context
Features
g_3
Stage III
Confidence Maps
Stage III Confidence
Head
Neck
L-Shoulder
L-Elbow
L-Wrist

Example results

General principle

- "Auto-context" (Tu CVPR 2008): instead of fancy graphical models, create feature from past predictions and repredict

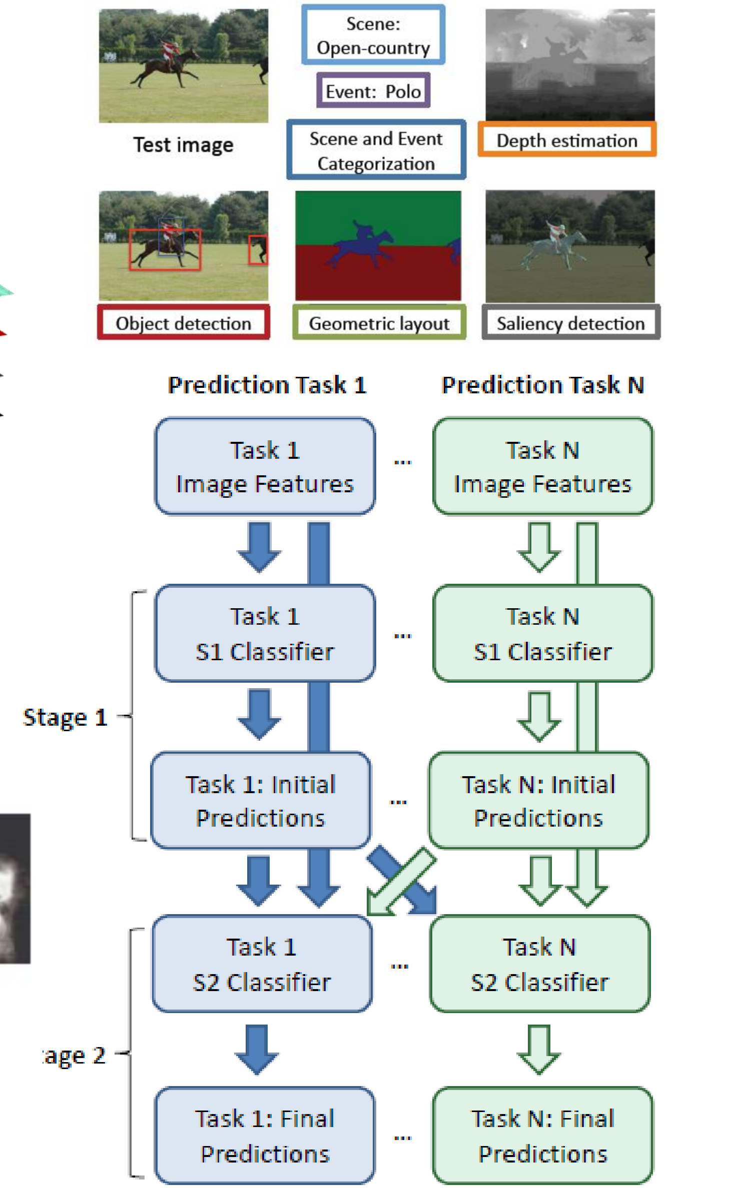
- Can view this as an "unrolled belief propagation" (Ross et al. 2011)

Tu Bai 2010: Auto-context

Ross Munoz Hebert Bagnell 2011: Message-Passing Inference Machines

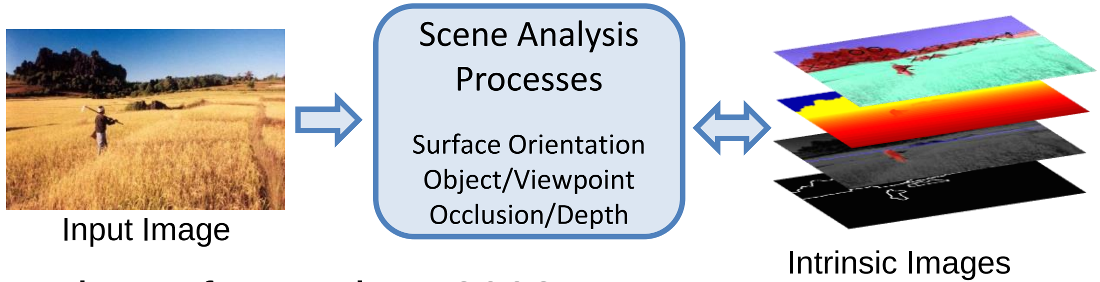
Many uses and variations on sequential structured prediction

Cascaded Classification Model



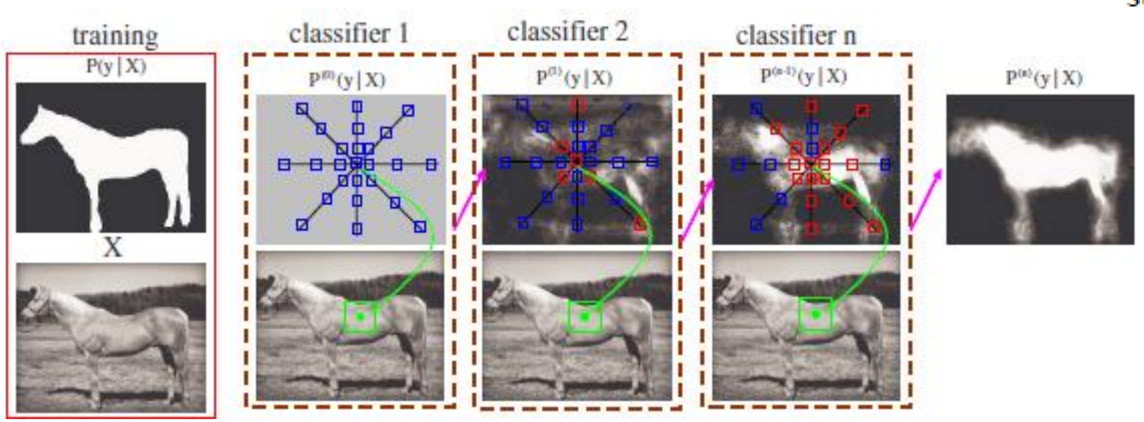
Heitz Gould Saxena Koller 2008
Li Kowdle Saxena Chen 2010

Closing the Loop



Hoiem Efros Hebert 2008

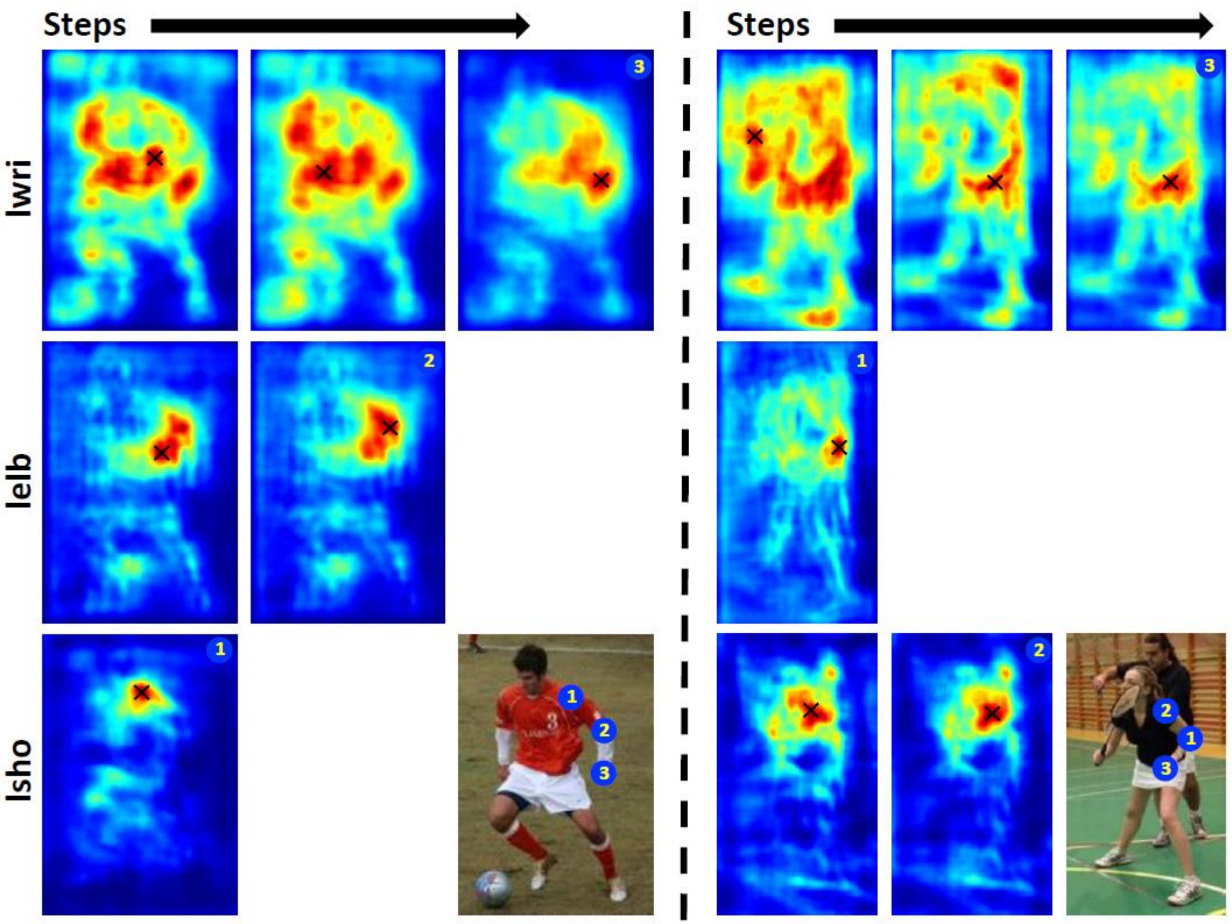
Autocontext



Tu 2008
Tu Bai 2010

Learning to search for landmarks

- Learn to find easy landmarks (body joints) first and use them as context for harder ones



Singh et al. CVPR 2015

Results: best (top) to worst (bottom)

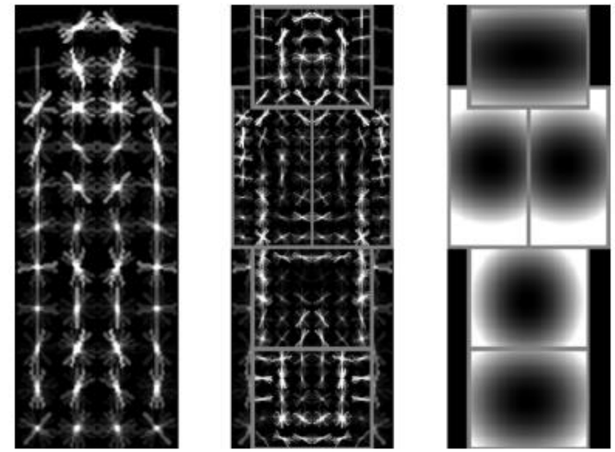
Graphical models vs. structured prediction

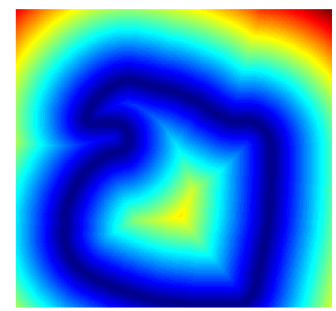
- Advantages of sequential prediction
  - Simple procedures for training and inference
  - Learns how much to rely on each prediction
  - Can model very complex relations

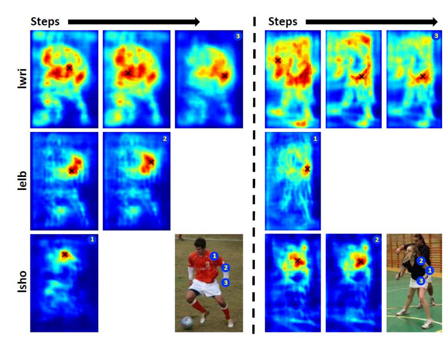
- Advantages of BP/graphcut/etc
  - Elegant
  - Relations are explicitly modeled
  - Exact inference in some cases

Things to remember

- Models can be broken down into part appearance and spatial configuration
  - Wide variety of models
- Efficient optimization can be tricky but usually possible
  - Generalized distance transform is a useful trick
- Rather than explicitly modeling contextual relations, can encode through features/classifiers







Next classes

- HW 5 due Monday (last one!!)
- Tues: Object tracking with Kalman Filters
- Thurs: Action Recognition