Pictorial structures

Laurens van der Maaten

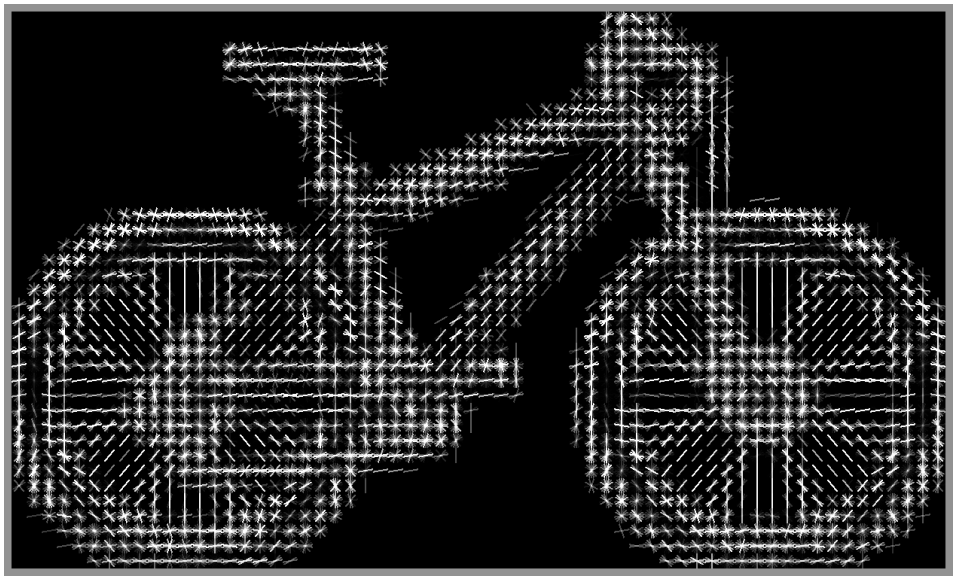
TU Delft

Introduction

- Object detection aims to find a particular object in an image
- Most popular object detectors are based on a *discriminative model*:
  - Gather *annotated image patches* (positive and negative examples)
  - Extract your favorite *image features* from these image patches
  - Train a *classifier* on the features to discriminate object from everything else
  - Classifier is applied on *candidate locations* to determine object presence
- The Dalal-Triggs detector is a commonly used object detector

Dalal-Triggs detector

- Extract *histograms of oriented gradients* (HOG) features from image patch:

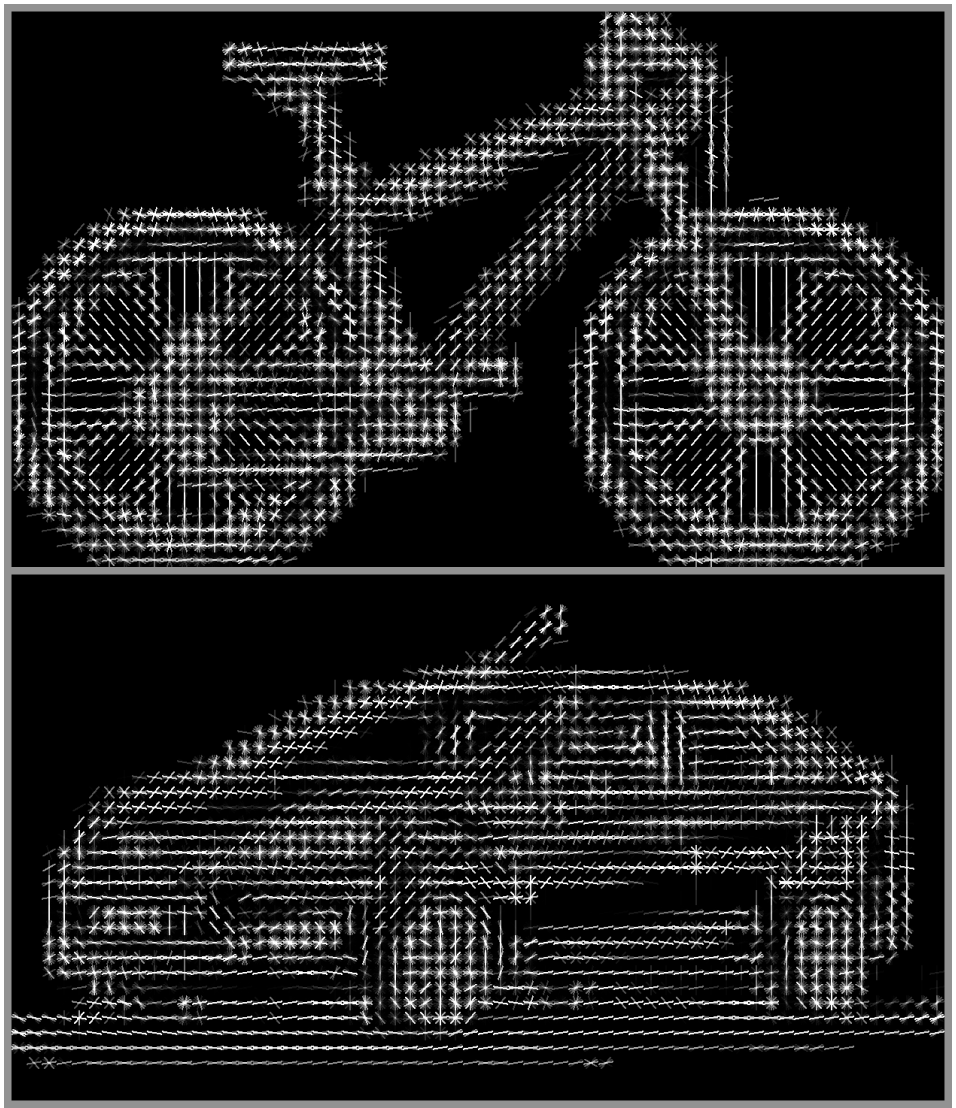


- HOG features divide an image into small (8x8) *blocks*, and measure the *gradient orientations* in each of the blocks using a histogram (almost like SIFT)

* Dalal & Triggs, 2005

Dalal-Triggs detector

- Different objects have different HOG features:

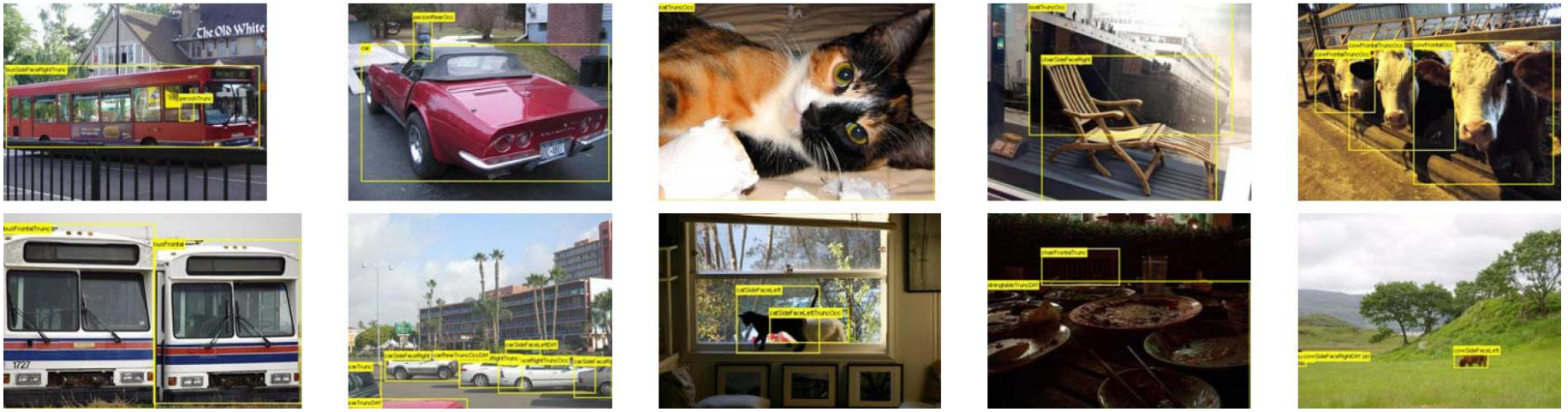


Dalal-Triggs detector

- Train a linear SVM on annotated images to predict object presence:

Training: $\mathbf{w}^* = \underset{\mathbf{w}}{\operatorname{argmin}} \max\left(0, 1 - y\mathbf{w}^{\mathrm{T}}\phi(\mathbf{I}; \mathbf{x})\right)$

Detection: $s(\mathbf{I}; \mathbf{x}) = \mathbf{w}^{*\mathrm{T}}\phi(\mathbf{I}; \mathbf{x})$

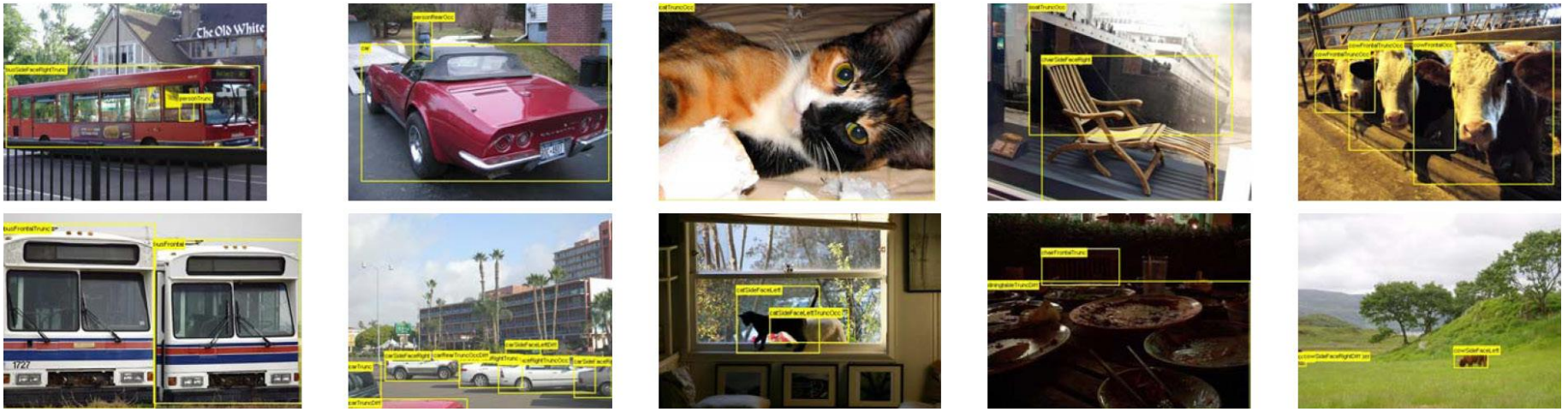


Dalal-Triggs detector

- Train a linear SVM on annotated images to predict object presence:

Training: $\mathbf{w}^* = \underset{\mathbf{w}}{\operatorname{argmin}} \max\left(0, 1 - y\mathbf{w}^{\mathrm{T}}\phi(\mathbf{I};\mathbf{x})\right)$

Detection: $s(\mathbf{I};\mathbf{x}) = \mathbf{w}^{*\mathrm{T}}\phi(\mathbf{I};\mathbf{x})$



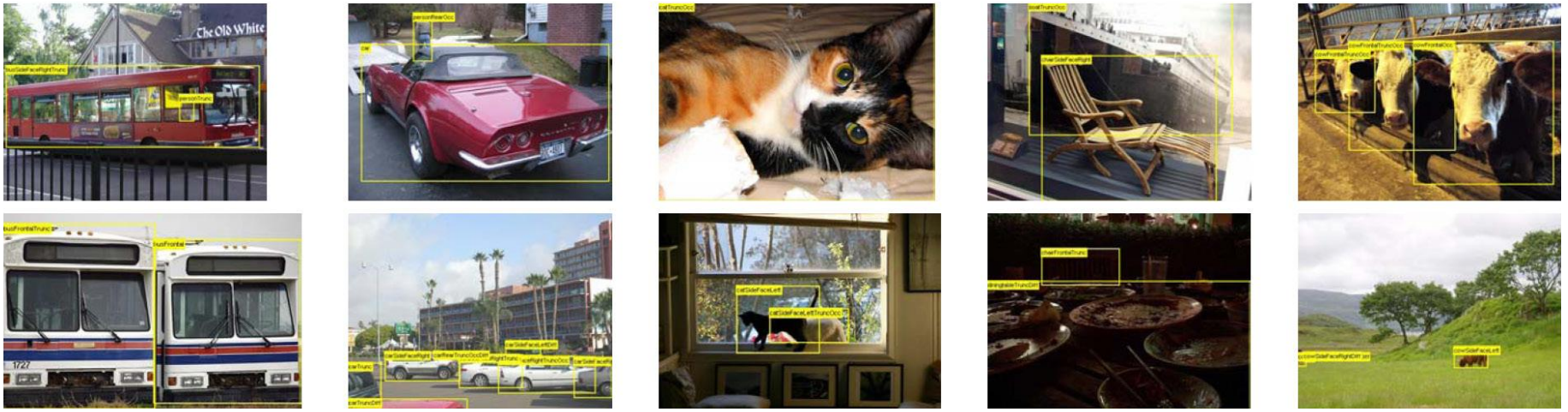
- How do we get the *negative examples* to train the SVM?

Dalal-Triggs detector

- Train a linear SVM on annotated images to predict object presence:

Training: $\mathbf{w}^* = \underset{\mathbf{w}}{\operatorname{argmin}} \max\left(0, 1 - y\mathbf{w}^{\mathrm{T}}\phi(\mathbf{I}; \mathbf{x})\right)$

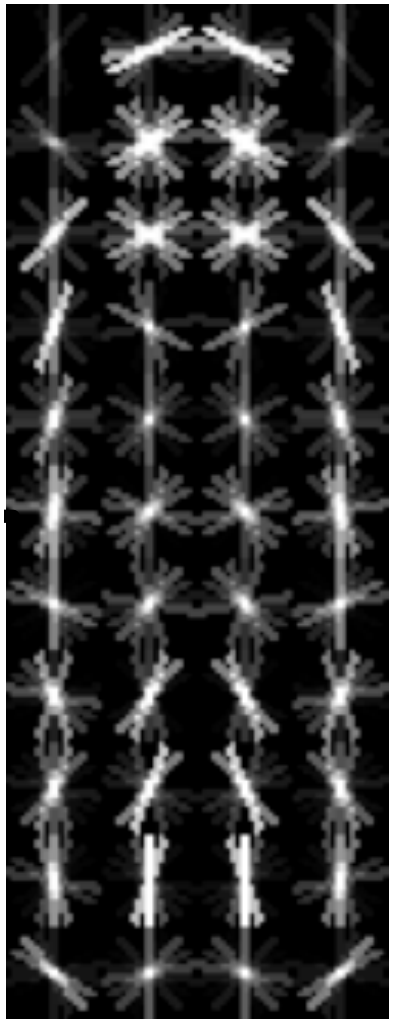
Detection: $s(\mathbf{I}; \mathbf{x}) = \mathbf{w}^{*\mathrm{T}}\phi(\mathbf{I}; \mathbf{x})$



- How do we get the *negative examples* to train the SVM? Random patches!

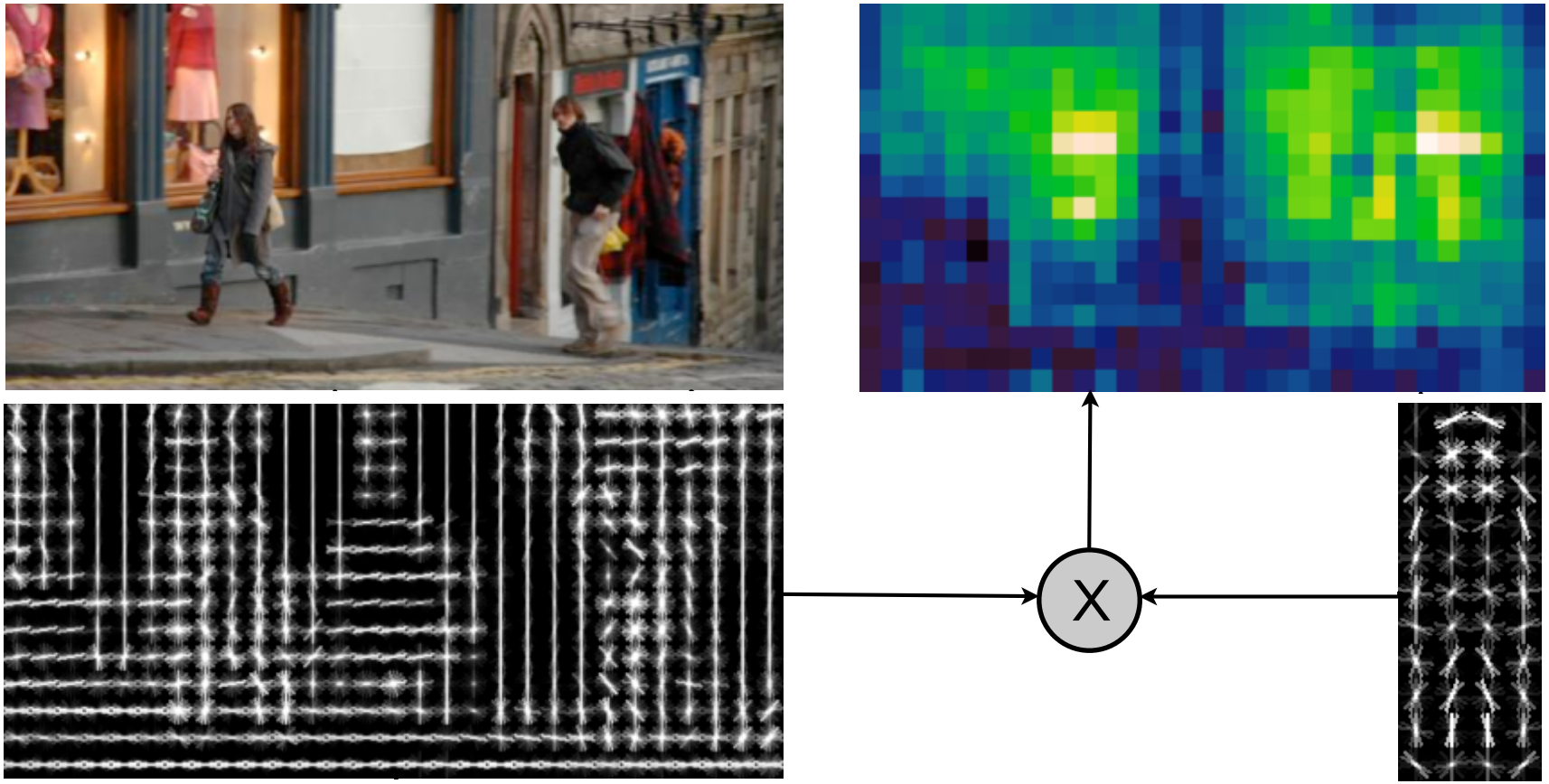
Dalal-Triggs detector

- HOG visualization of the SVM weights for a pedestrian detector:



Dalal-Triggs detector

- Applying the detector at each location leads to a *confidence map*:



- *Non-maxima suppression* can be used to obtain the final detections

Dalal-Triggs detector

- Example of pedestrian detections using Dalal-Triggs detector:

Pictorial structures

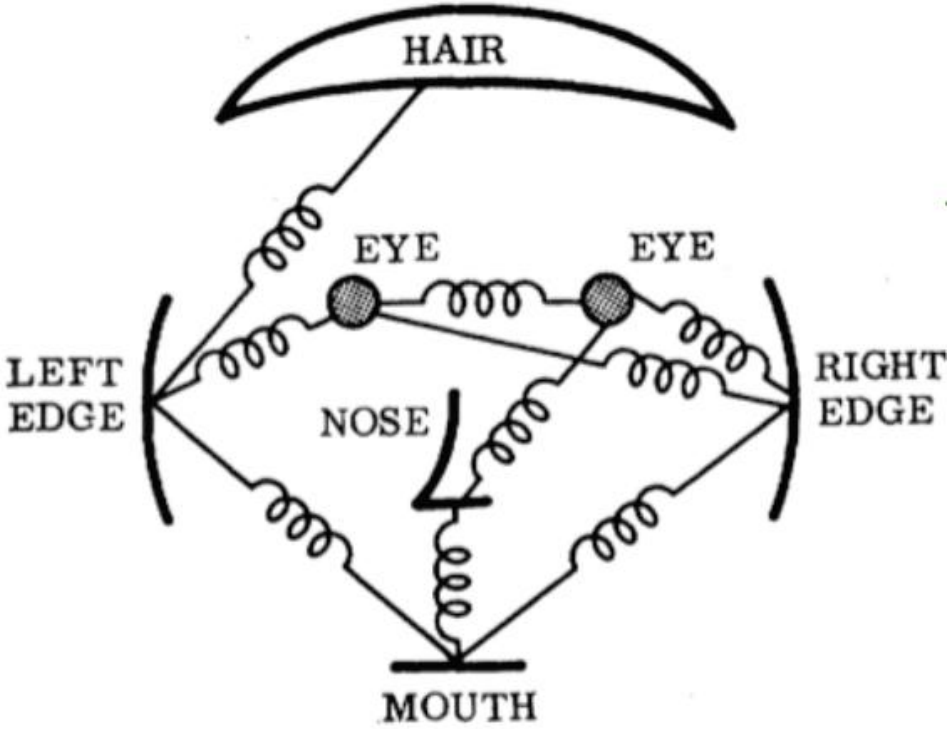
- What can we do when a part of the object to be detected is occluded?

Pictorial structures

- What can we do when a part of the object to be detected is occluded?
- Exploit the fact that other parts of the object are still visible!

Pictorial structures

- What can we do when a part of the object to be detected is occluded?
- Exploit the fact that other parts of the object are still visible!
- *Pictorial structures* does this by modeling objects as a constellation of parts:

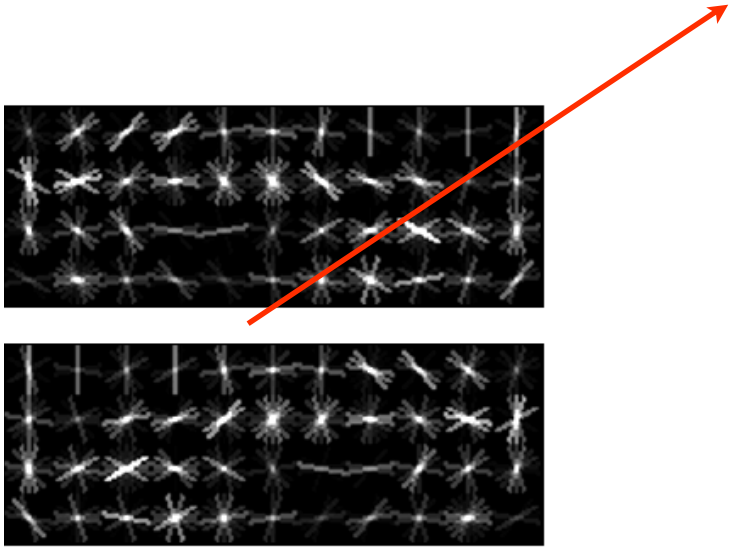


* Fischler & Elschlager, 1973

Deformable template models

- Defines a *score function* that involves *parts* and *part deformations*:

$$s(\mathbf{I}; x_0, y_0, \ldots, x_{|V|}, y_{|V|}) = \mathbf{w}_0^{\mathrm{T}} \phi(\mathbf{I}; x_0, y_0)$$



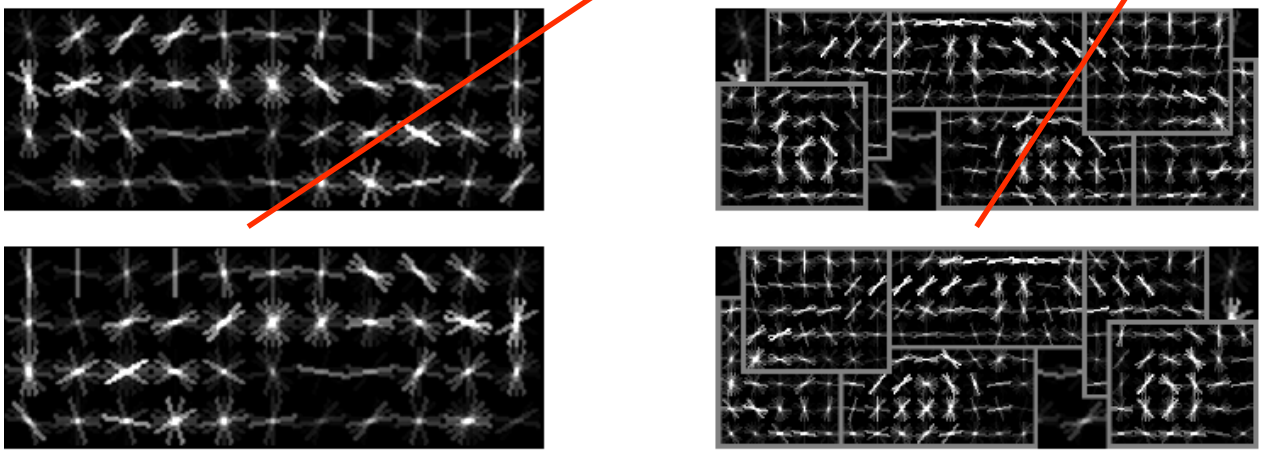
Global object model

* Felzenszwalb *et al.*, 2010

Deformable template models

- Defines a *score function* that involves *parts* and *part deformations*:

$$s(\mathbf{I}; x_0, y_0, \ldots, x_{|V|}, y_{|V|}) = \mathbf{w}_0^{\mathrm{T}} \phi(\mathbf{I}; x_0, y_0) + \sum_{i \in V} \mathbf{w}_i^{\mathrm{T}} \phi(\mathbf{I}; x_i, y_i)$$



Global object model

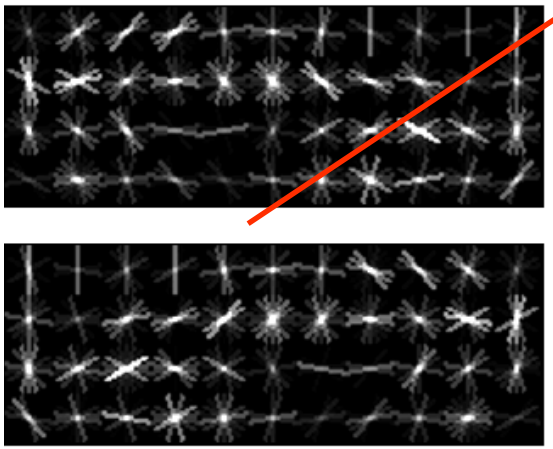
Object part models

* Felzenszwalb *et al.*, 2010

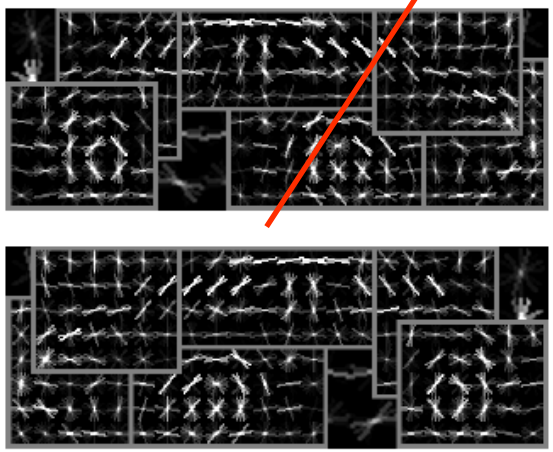
Deformable template models

- Defines a *score function* that involves *parts* and *part deformations*:

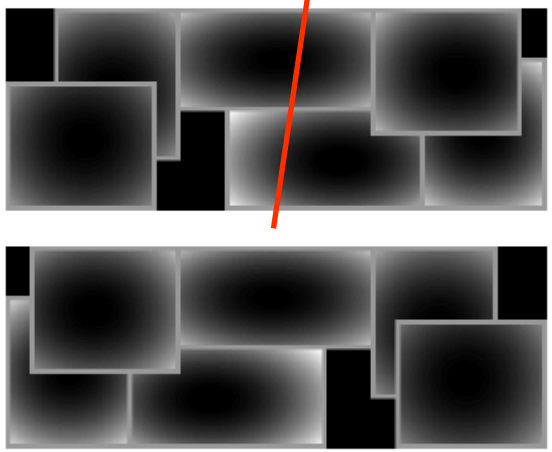
$$s(\mathbf{I}; x_0, y_0, \ldots, x_{|V|}, y_{|V|}) = \mathbf{w}_0^{\mathrm{T}} \phi(\mathbf{I}; x_0, y_0) + \sum_{i \in V} \mathbf{w}_i^{\mathrm{T}} \phi(\mathbf{I}; x_i, y_i) + \sum_{(i,j) \in E} d_{ij} \phi_d(x_i - x_j, y_i - y_j)$$



Global object model



Object part models



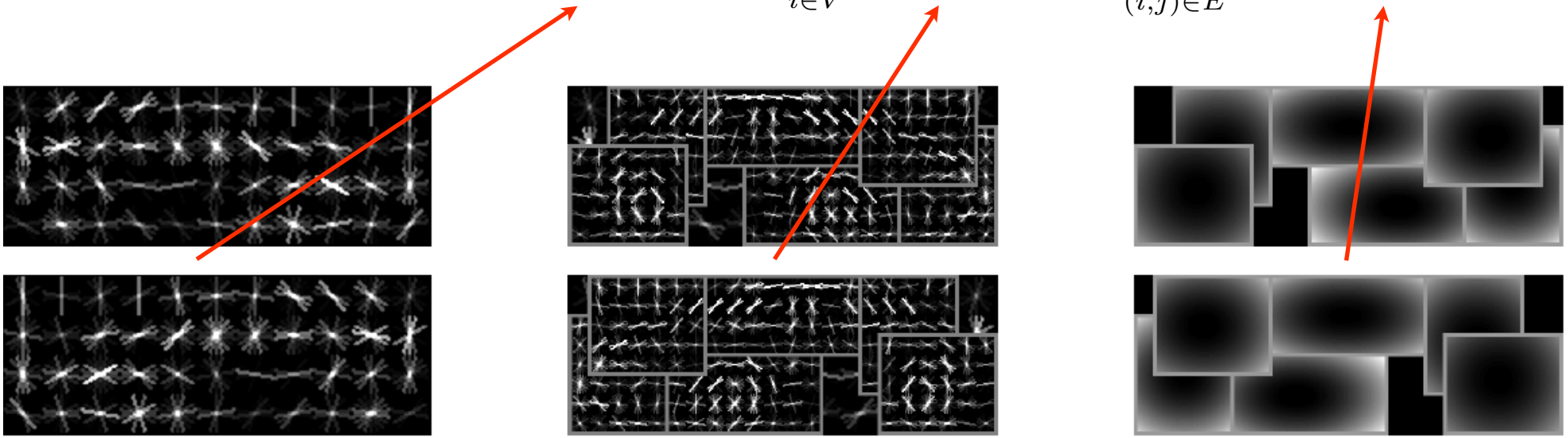
Deformation model

* Felzenszwalb *et al.*, 2010

Deformable template models

- Defines a *score function* that involves *parts* and *part deformations*:

$$s(\mathbf{I}; x_0, y_0, \ldots, x_{|V|}, y_{|V|}) = \mathbf{w}_0^{\mathrm{T}} \phi(\mathbf{I}; x_0, y_0) + \sum_{i \in V} \mathbf{w}_i^{\mathrm{T}} \phi(\mathbf{I}; x_i, y_i) + \sum_{(i,j) \in E} d_{ij} \phi_d(x_i - x_j, y_i - y_j)$$



Global object model | Object part models | Deformation model

- Deformable template models are much more robust against *partial occlusions* and *deformations* of non-rigid objects

* Felzenszwalb *et al.*, 2010

Pictorial structures

- Find the *optimal* configuration of a pictorial structures (detection) as follows:

$$\max_{x_0, y_0, \ldots, x_{|V|}, y_{|V|}} s(\mathbf{I}; x_0, y_0, \ldots, x_{|V|}, y_{|V|})$$

Pictorial structures

- Find the *optimal* configuration of a pictorial structures (detection) as follows:

$$\max_{x_0, y_0, \ldots, x_{|V|}, y_{|V|}} s(\mathbf{I}; x_0, y_0, \ldots, x_{|V|}, y_{|V|})$$

- For *squared-error* deformation models, this can be done very efficiently:

$$g(x_i) = \min_{x_j} (f(x_j) + (x_i - x_j)^2)$$

Pictorial structures

- Find the *optimal* configuration of a pictorial structures (detection) as follows:

$$\max_{x_0, y_0, \ldots, x_{|V|}, y_{|V|}} s(\mathbf{I}; x_0, y_0, \ldots, x_{|V|}, y_{|V|})$$

- For *squared-error* deformation models, this can be done very efficiently:

$$g(x_i) = \min_{x_j} (f(x_j) + (x_i - x_j)^2)$$

final score with deformations

Pictorial structures

- Find configuration of pict. structures model by maximizing over part locations:

$$\max_{x_0, y_0, \ldots, x_{|V|}, y_{|V|}} s(\mathbf{I}; x_0, y_0, \ldots, x_{|V|}, y_{|V|})$$

- For *squared-error* deformation models, this can be done very efficiently:

$$g(x_i) = \min_{x_j} (f(x_j) + (x_i - x_j)^2)$$

final score with deformations

negative part model score

Pictorial structures

- Find the *optimal* configuration of a pictorial structures (detection) as follows:

$$\max_{x_0, y_0, \ldots, x_{|V|}, y_{|V|}} s(\mathbf{I}; x_0, y_0, \ldots, x_{|V|}, y_{|V|})$$

- For *squared-error* deformation models, this can be done very efficiently:

$$g(x_i) = \min_{x_j} (f(x_j) + (x_i - x_j)^2)$$

final score with deformations

negative part model score

deformation penalty

Pictorial structures

- Find the *optimal* configuration of a pictorial structures (detection) as follows:

$$\max_{x_0, y_0, \ldots, x_{|V|}, y_{|V|}} s(\mathbf{I}; x_0, y_0, \ldots, x_{|V|}, y_{|V|})$$

- For *squared-error* deformation models, this can be done very efficiently:

$$g(x_i) = \min_{x_j} (f(x_j) + (x_i - x_j)^2)$$

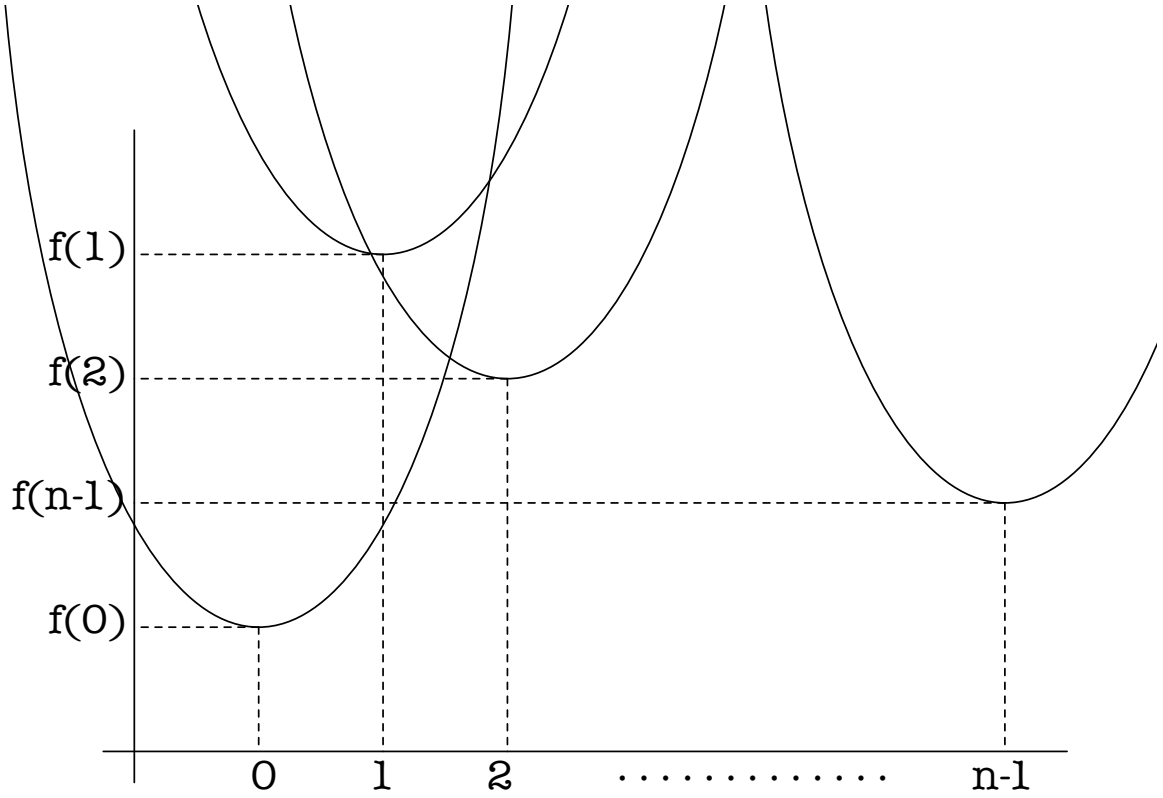
final score with deformations

negative part model score

deformation penalty

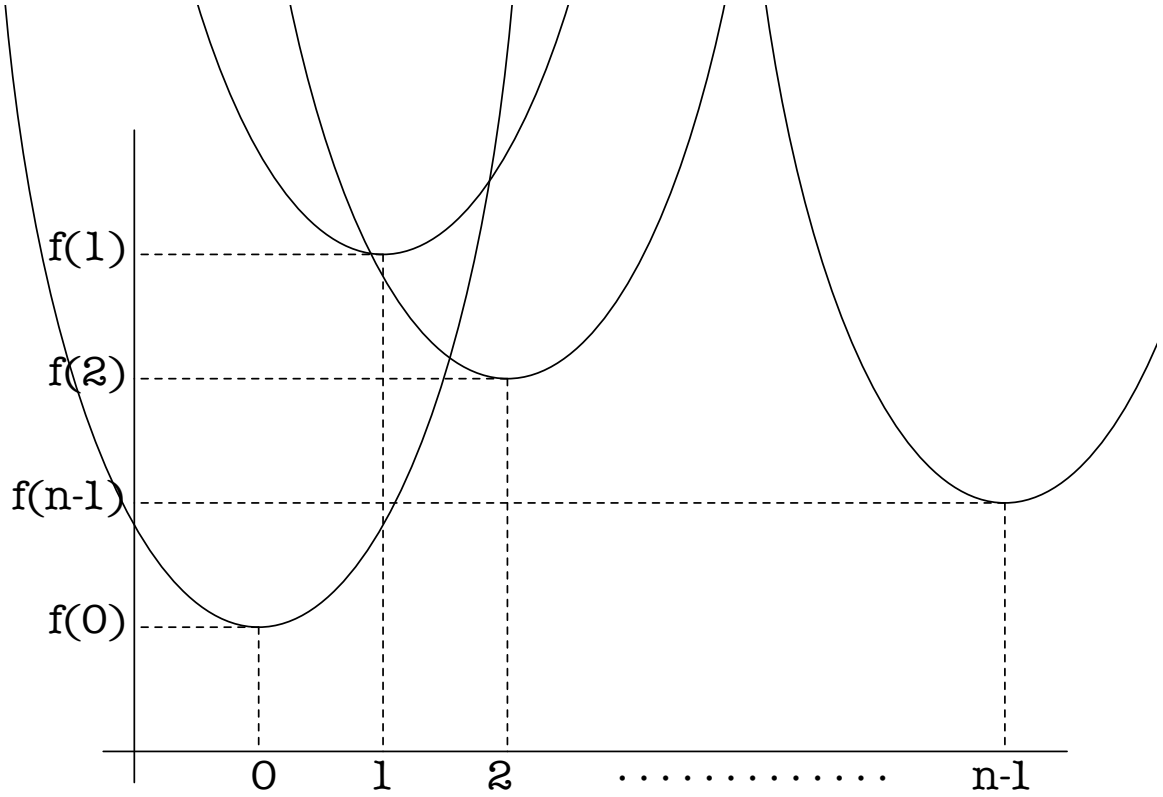
- Hence, we have a parabola for every pixel x_j rooted at $(x_j, f(x_j))$

Pictorial structures



* Felzenszwalb & Huttenlocher, 2004

Pictorial structures



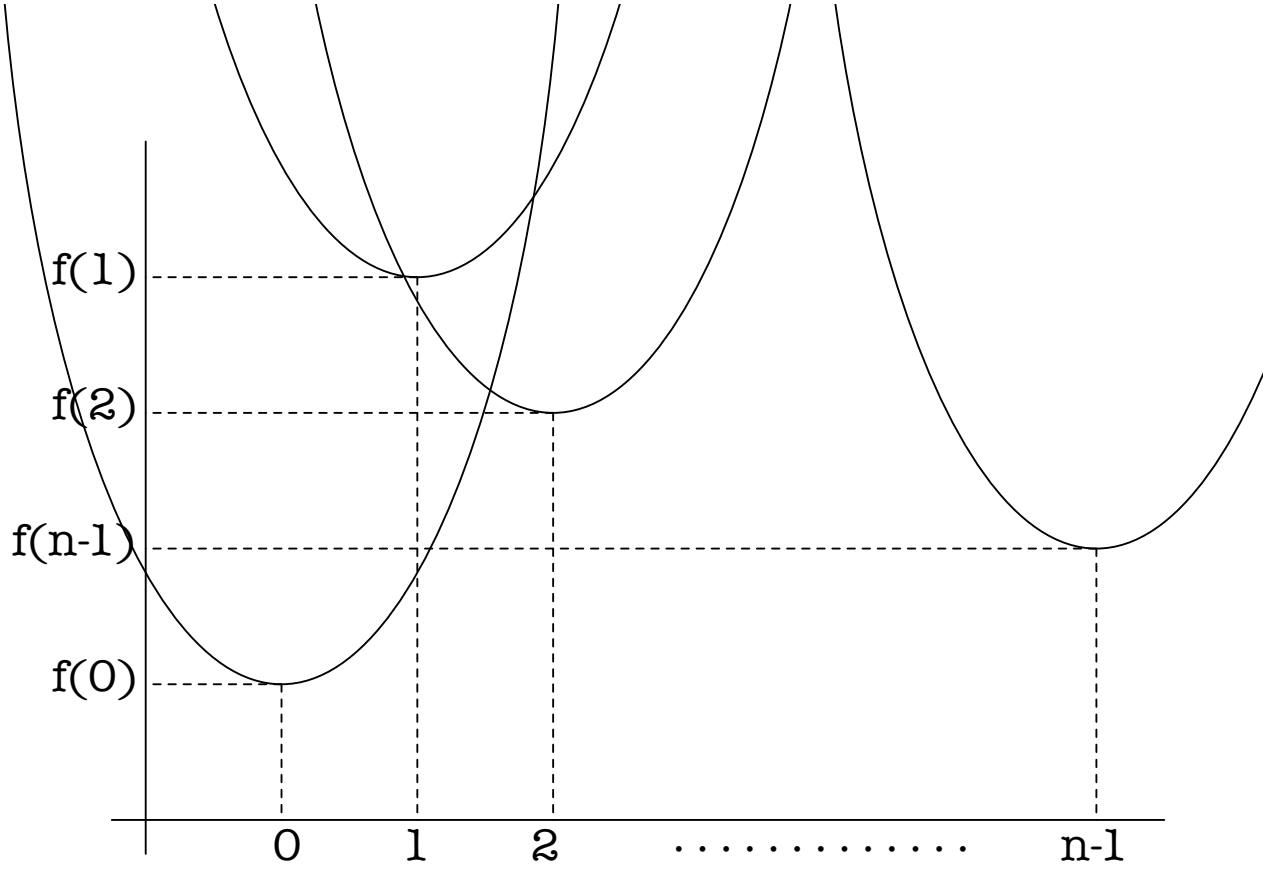
- It is straightforward to compute the *intersection* between two parabolas:

$$\frac{(f(x_i) + x_i^2) - (f(x_j) + x_j^2)}{2x_i - 2x_j}$$

* Felzenszwalb & Huttenlocher, 2004

Pictorial structures

- If $x_j < x_i$: parabola corresponding to x_j is *below* that of x_i *left* of the intersection, and above it right of the intersection

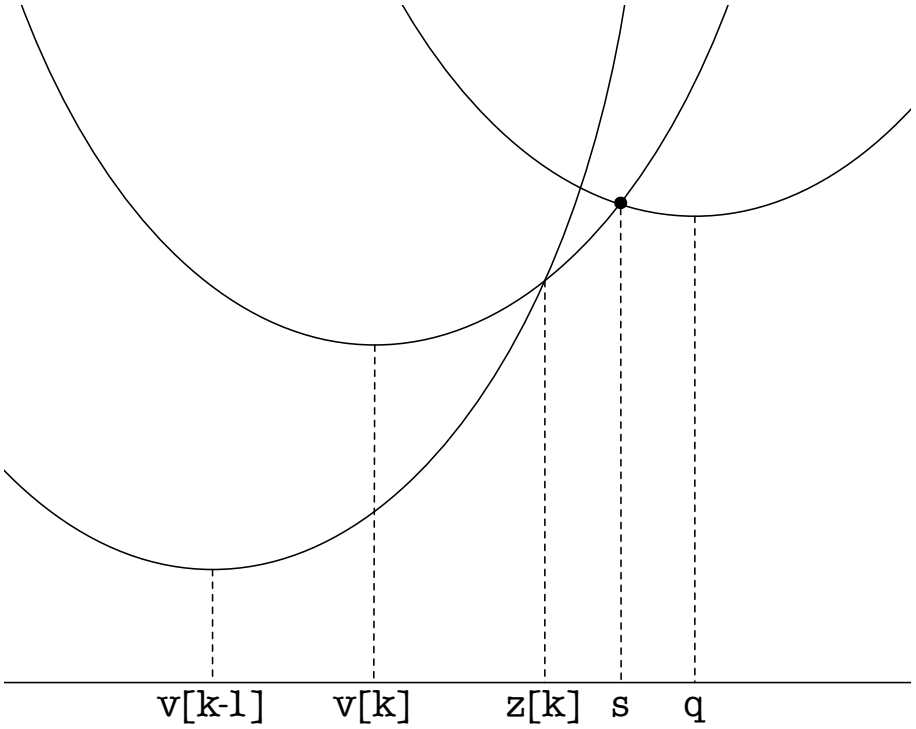


* Felzenszwalb & Huttenlocher, 2004

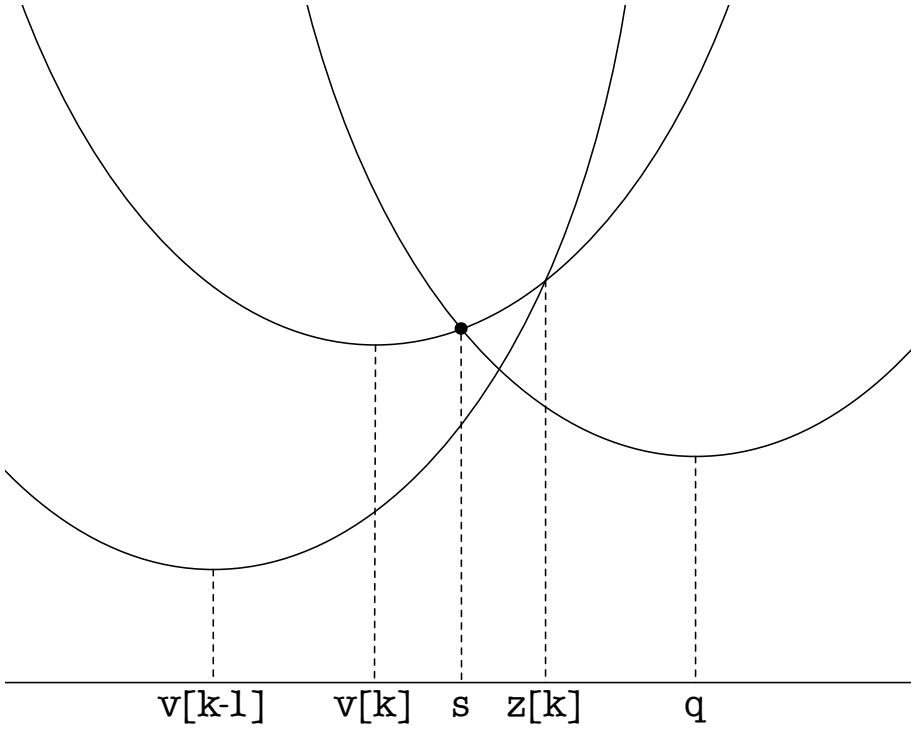
Pictorial structures

- Maintain the *lower envelope* of the parabolas (parabolas and intersections)

- When adding a new parabola, there are two possibilities:



new intersection right of last intersection:
maintain last parabola in the envelope

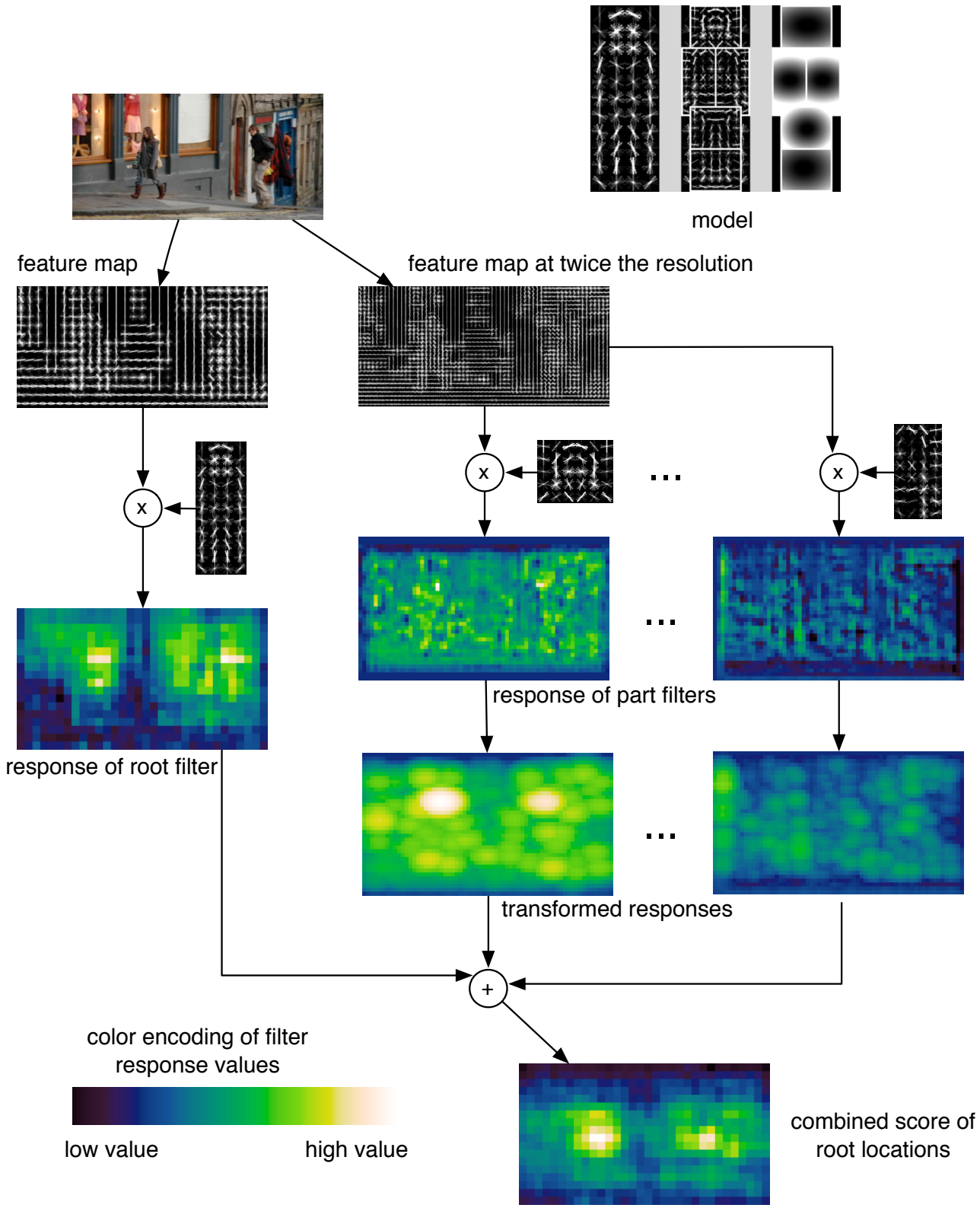


new intersection left of last intersection:
remove last parabola from the envelope

Pictorial structures

- This suggests a simple algorithm that is *linear* in the number of pixels:
  - Maintain list with the *lower envelope* of the parabolas (indices and intersections)
  - Move from *left to right* through all parabolas; and do for each parabola:
    - Find intersection of parabola with the last parabola in lower envelope
    - If intersection is left of last intersection in lower envelope: remove last parabola from lower envelope, and go back one step
    - Add parabola to lower envelope, starting from intersection

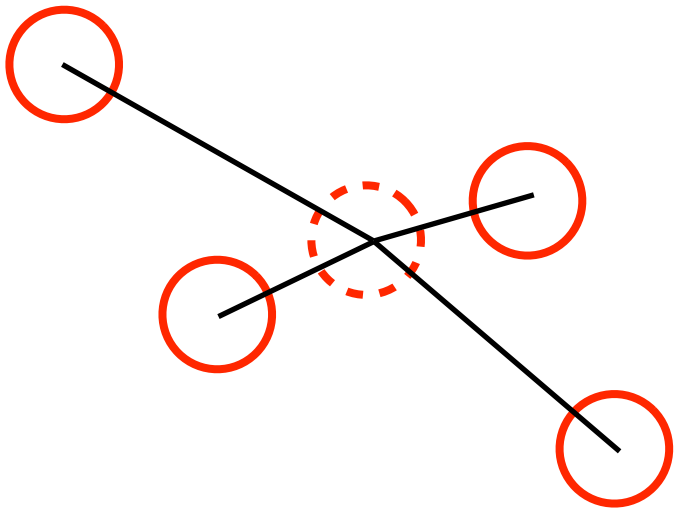
* Felzenszwalb & Huttenlocher, 2004


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

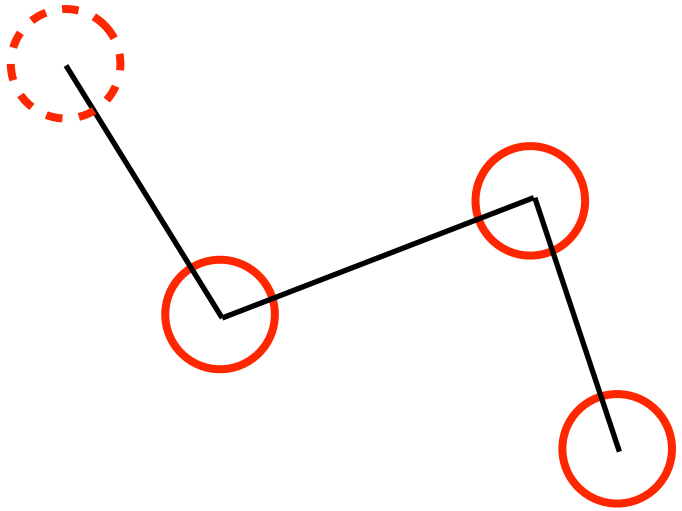
Graph structure

- One can define different graph structures, as long as they are trees:

Star-shaped tree



Minimum spanning tree



- The tree structure is fixed, but edge lengths and directions are learned

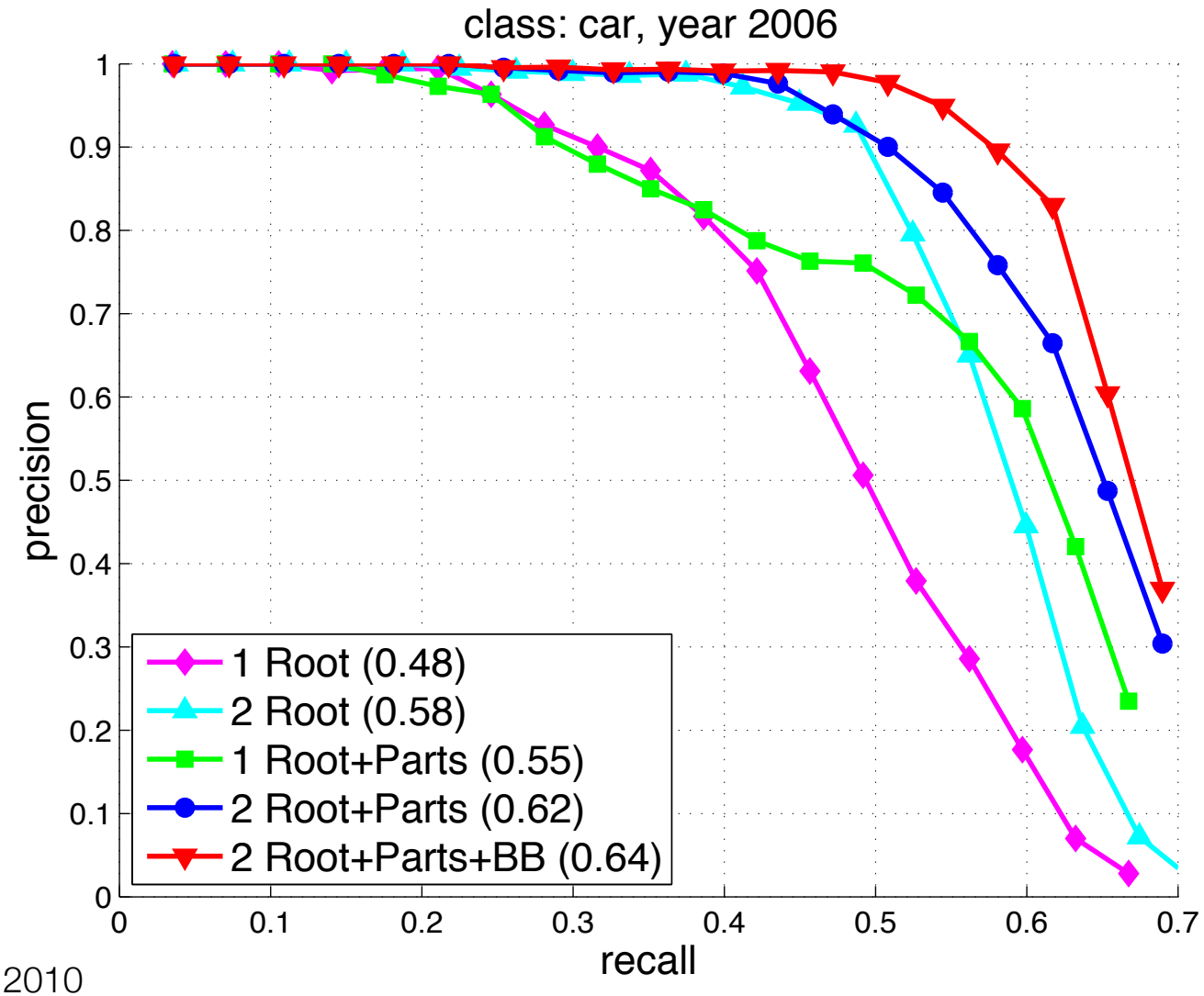
Pictorial structures

- Examples of object detections by pictorial-structures models:

* Felzenszwalb *et al.*, 2010

Results

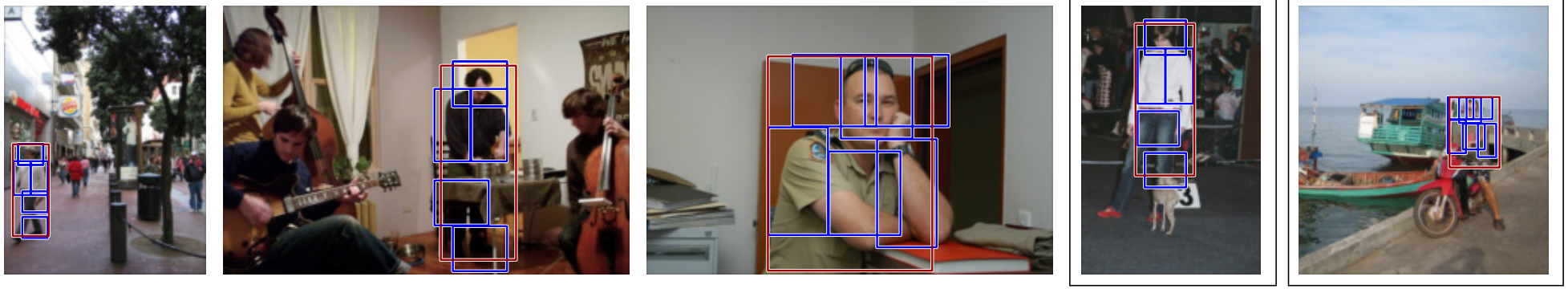
- Precision / recall curves for car detector on Pascal VOC:



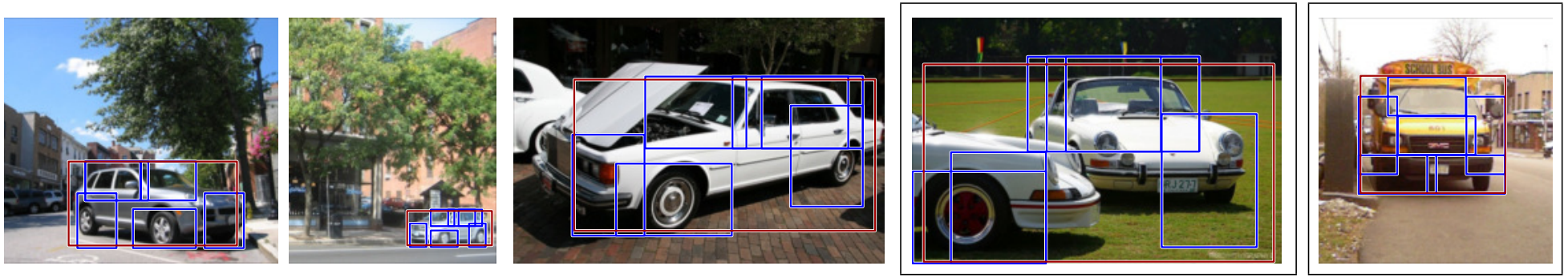
* Felzenszwalb *et al.*, 2010

Example detections

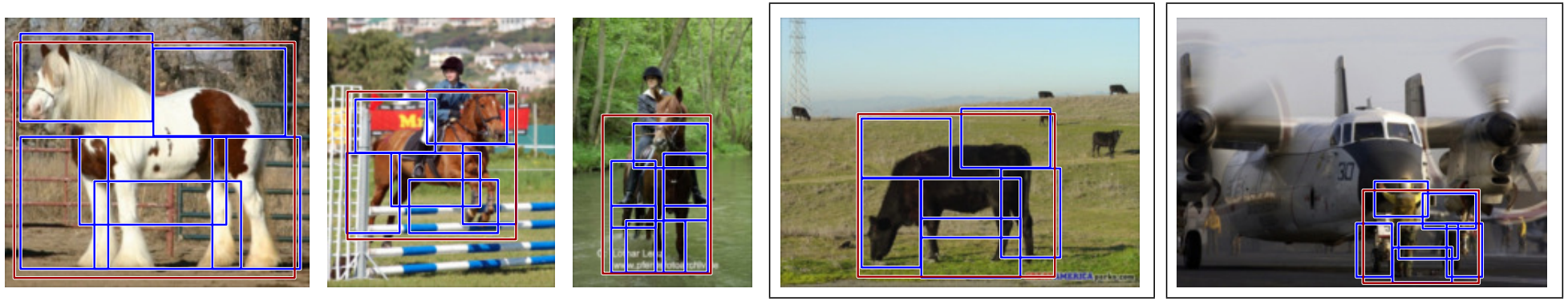
person



car

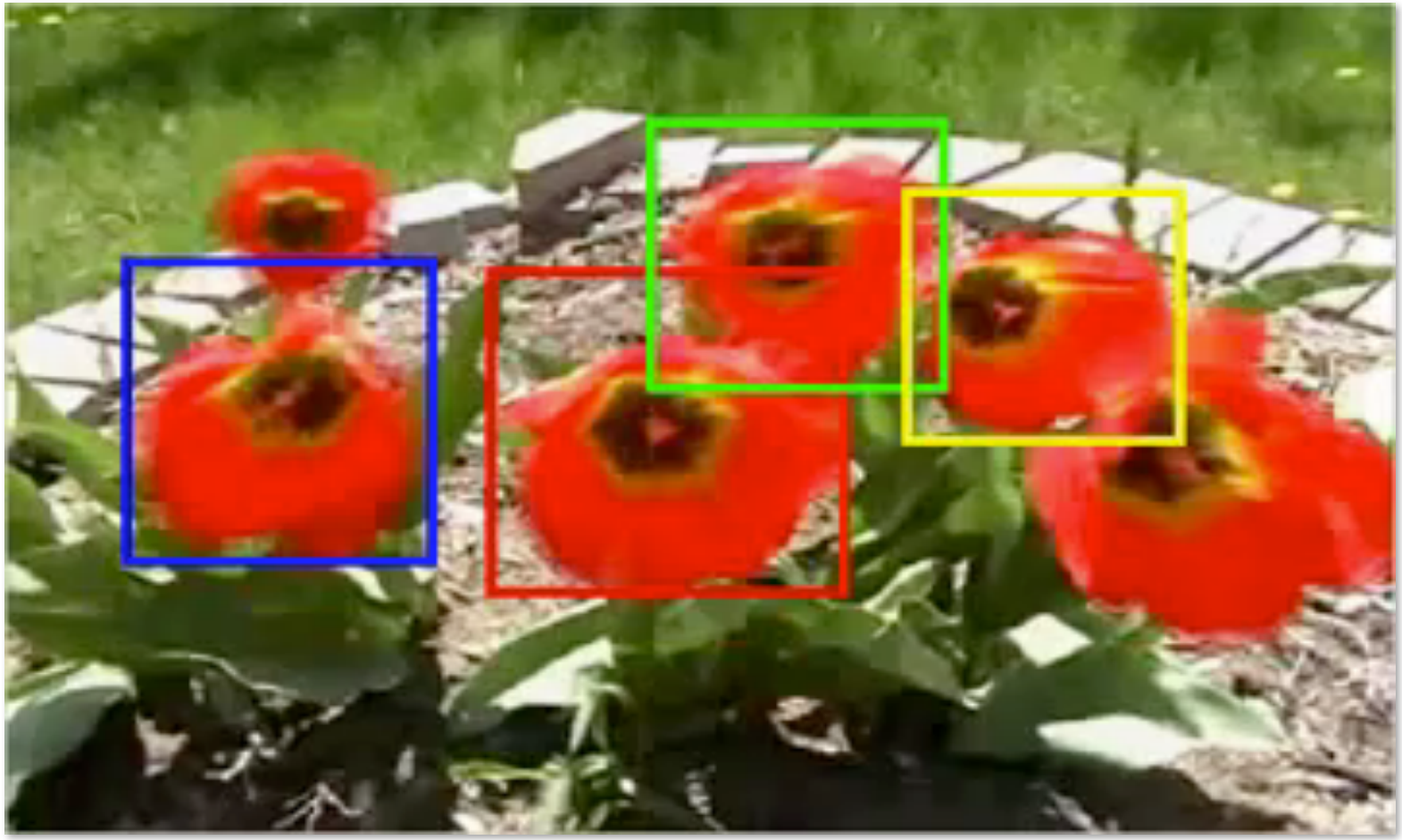


horse



Pictorial structures

- Use pictorial structures to prevent trackers from “switching” objects:



* Zhang & van der Maaten, 2013

Questions?