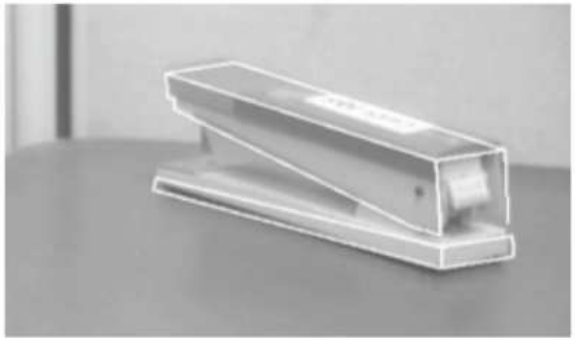
Lecture 17: object detection

Professor Fei-Fei Li

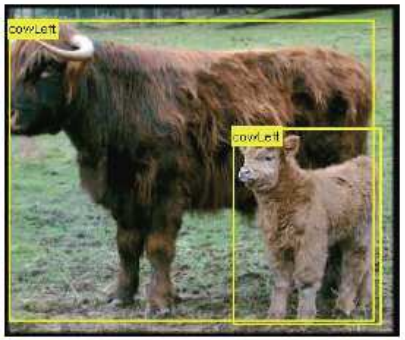
Stanford Vision Lab

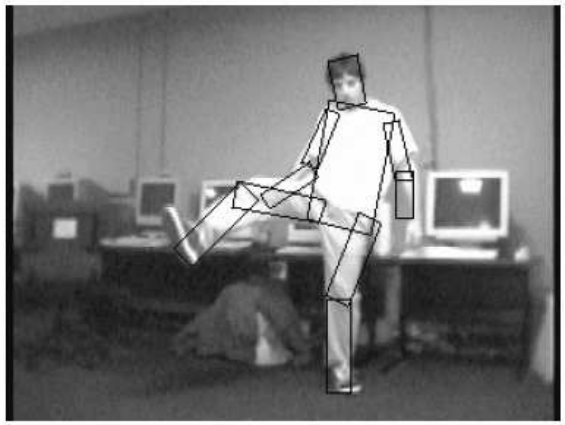
Object detection

Detecting rigid objects

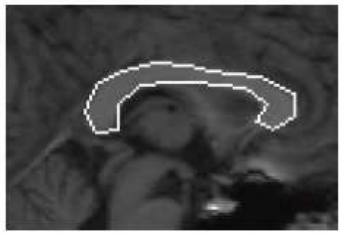


PASCAL challenge

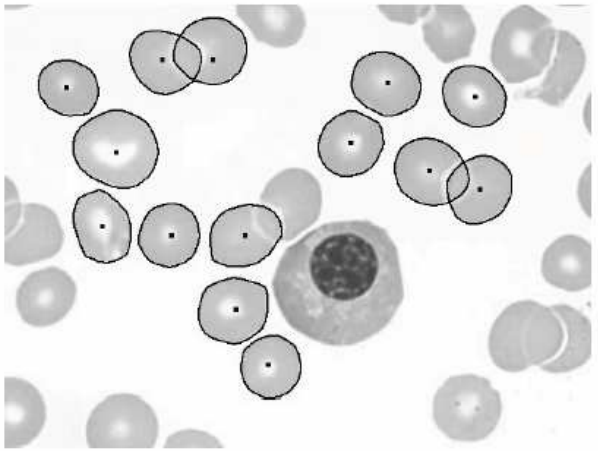




Detecting non-rigid objects



Medical image analysis



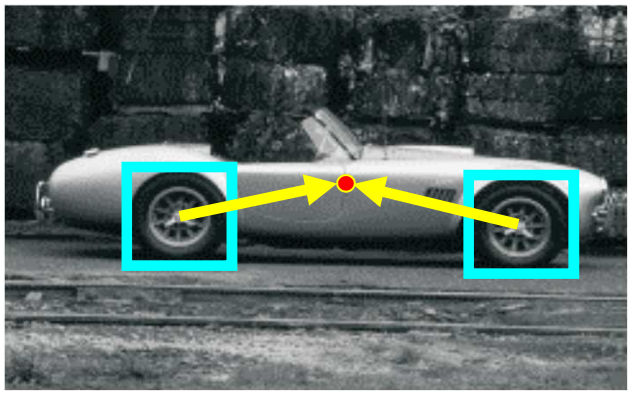
Segmenting cells

What we will learn today?

- Implicit Shape Model
  - Representation
  - Recognition
  - Experiments and results
- Deformable Models
  - The PASCAL challenge
  - Latent SVM Model

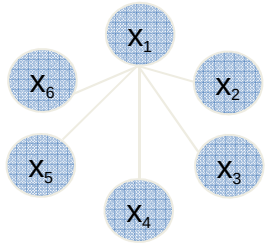
What we will learn today?

- Implicit Shape Model
  - Representation
  - Recognition
  - Experiments and results
- Deformable Models
  - The PASCAL challenge
  - Latent SVM Model



Implicit Shape Model (ISM)

- Basic ideas
  - Learn an appearance codebook
  - Learn a star-topology structural model
    - Features are considered independent given obj. center

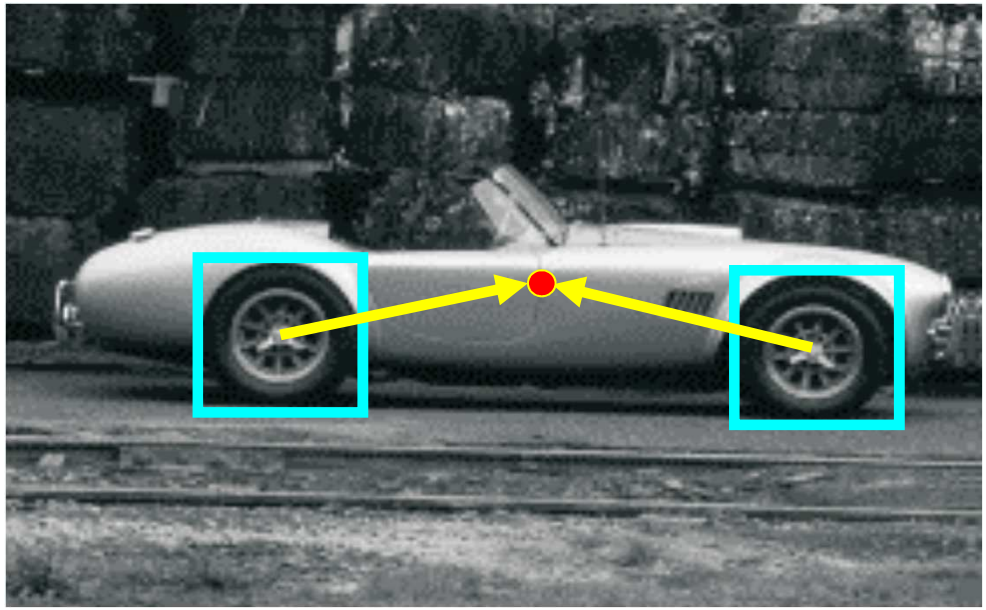


- Algorithm: probabilistic Gen. Hough Transform
  - Exact correspondences → Prob. match to object part
  - NN matching → Soft matching
  - Feature location on obj. → Part location distribution
  - Uniform votes → Probabilistic vote weighting
  - Quantized Hough array → Continuous Hough space

Source: Bastian Leibe

Implicit Shape Model: Basic Idea

- Visual vocabulary is used to index votes for object position [a visual word = "part"].



Training image

Visual codeword with displacement vectors

B. Leibe, A. Leonardis, and B. Schiele, Robust Object Detection with Interleaved Categorization and Segmentation, International Journal of Computer Vision, Vol. 77(1-3), 2008.

Source: Bastian Leibe

Implicit Shape Model: Basic Idea

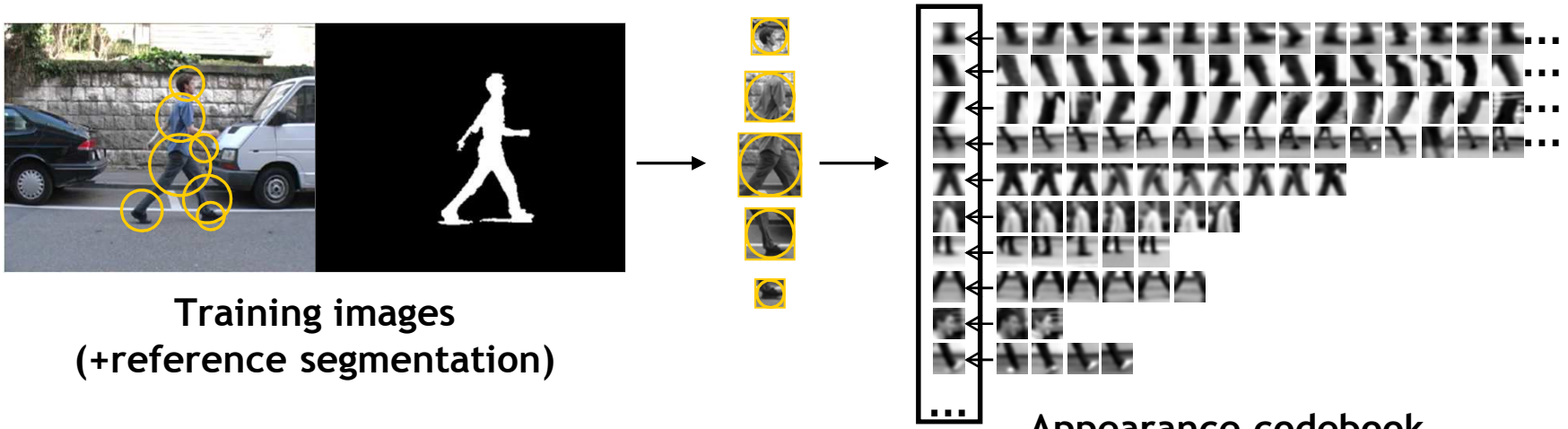
- Objects are detected as consistent configurations of the observed parts (visual words).

Test image

B. Leibe, A. Leonardis, and B. Schiele, Robust Object Detection with Interleaved Categorization and Segmentation, International Journal of Computer Vision, Vol. 77(1-3), 2008.

Source: Bastian Leibe

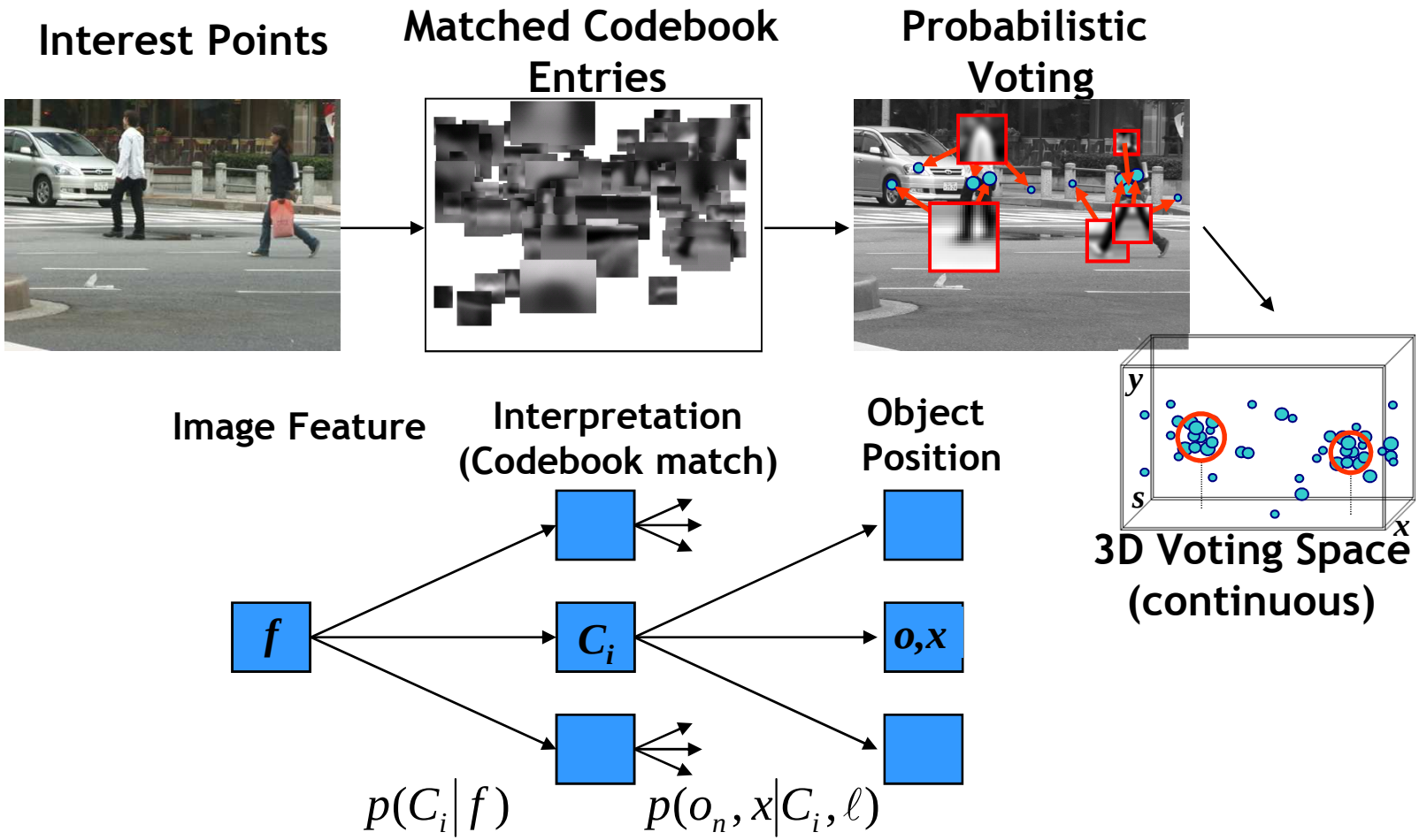
Implicit Shape Model - Representation



- Learn appearance codebook
  - Extract local features at interest points
  - Agglomerative clustering ⇒ codebook
- Learn spatial distributions
  - Match codebook to training images
  - Record matching positions on object

Source: Bastian Leibe

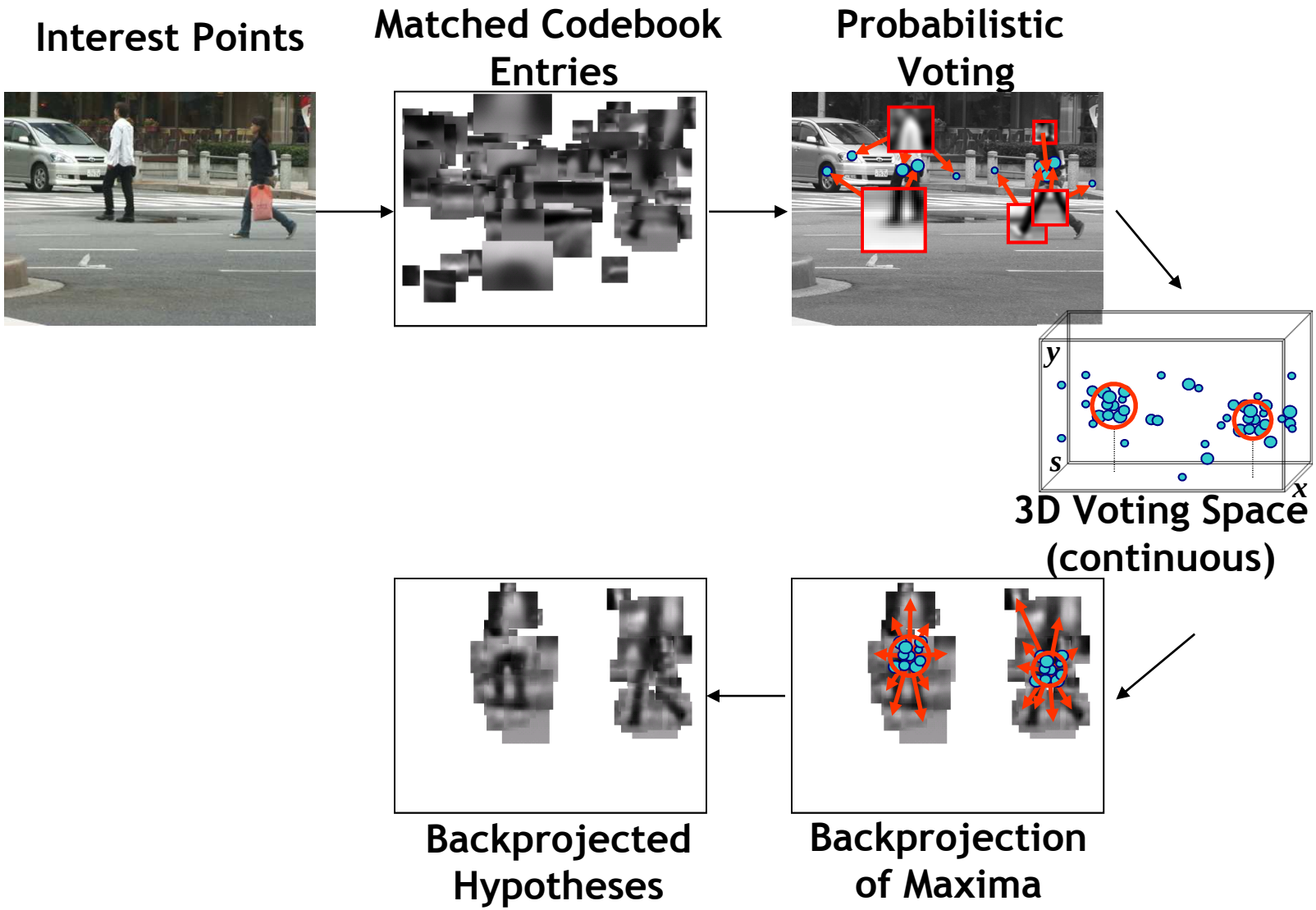
Implicit Shape Model - Recognition



Probabilistic vote weighting (will be explained later in detail)

[Leibe, Leonardis, Schiele, SLCV'04; IJCV'08]

Implicit Shape Model - Recognition



[Leibe, Leonardis, Schiele, SLCV'04; IJCV'08]

Example: Results on Cows

Original image

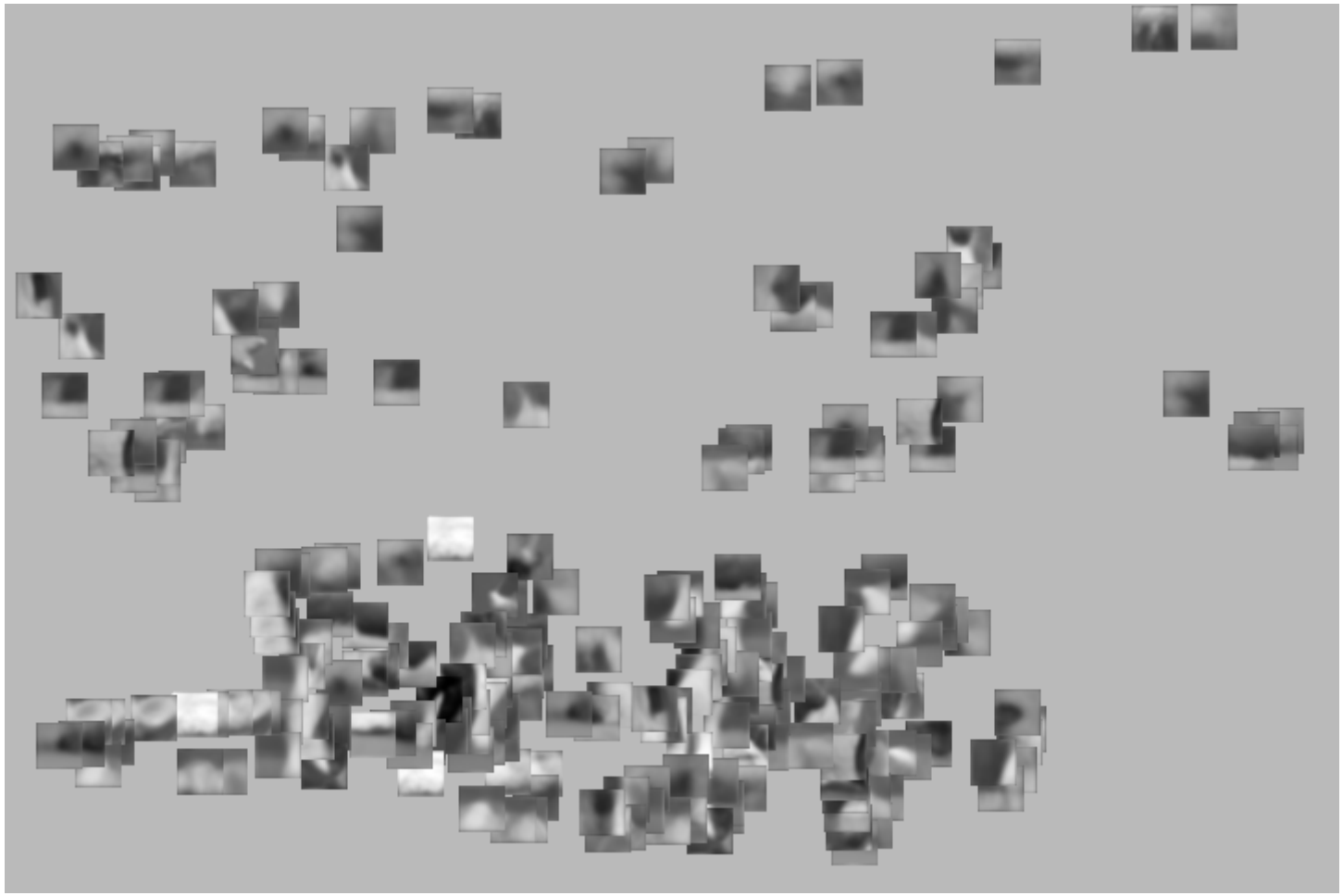
Source: Bastian Leibe

Example: Results on Cows

Interest points

Source: Bastian Leibe

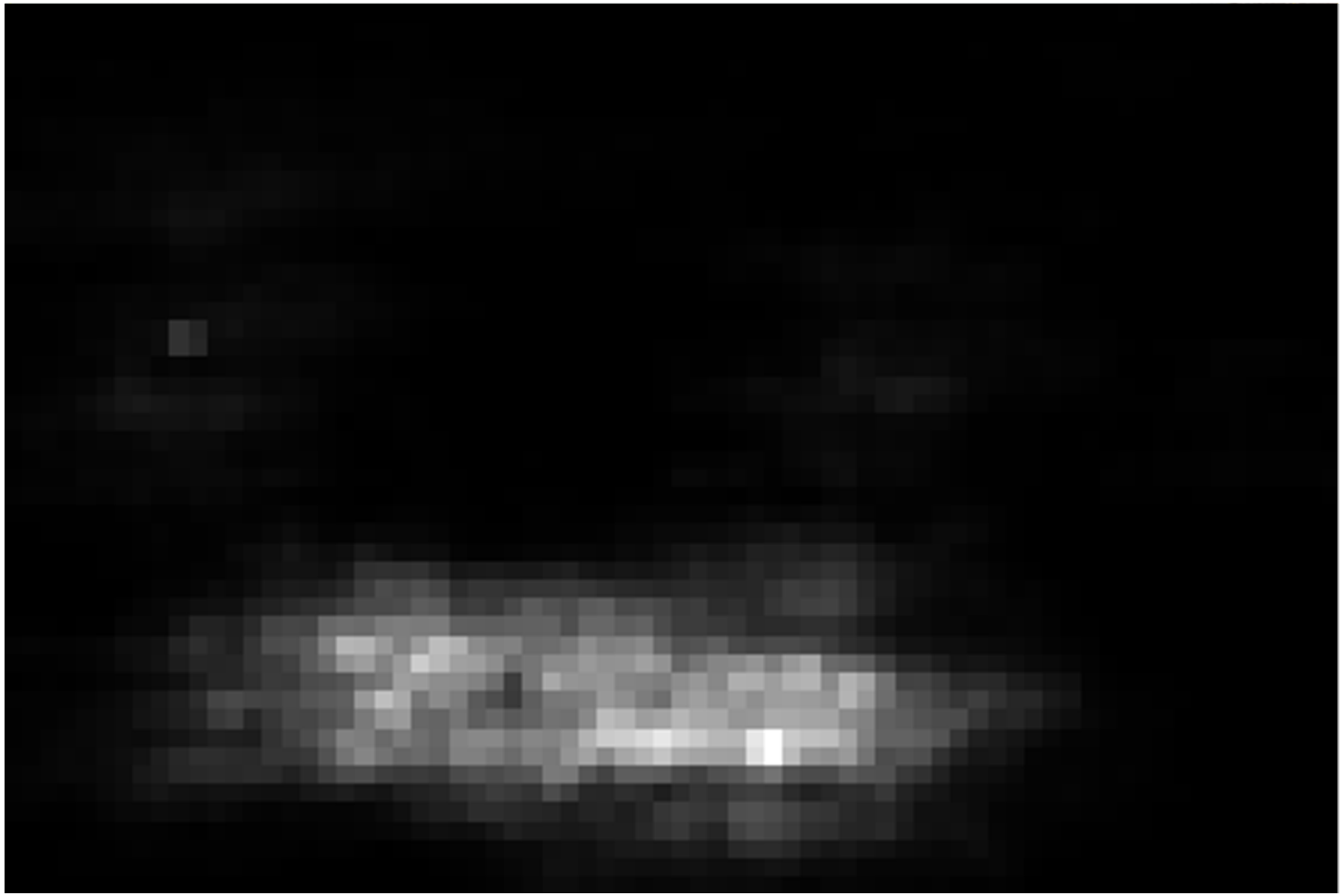
Example: Results on Cows



Matched patches

Source: Bastian Leibe

Example: Results on Cows



Prob. Votes

Source: Bastian Leibe

Example: Results on Cows

1st hypothesis

Source: K. Grauman & B. Leibe

Example: Results on Cows

2nd hypothesis

Source: Bastian Leibe

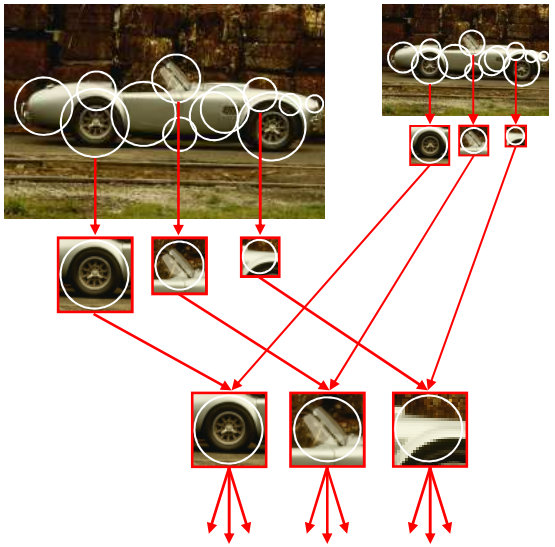
Example: Results on Cows

3rd hypothesis

Source: Bastian Leibe

Scale Invariant Voting

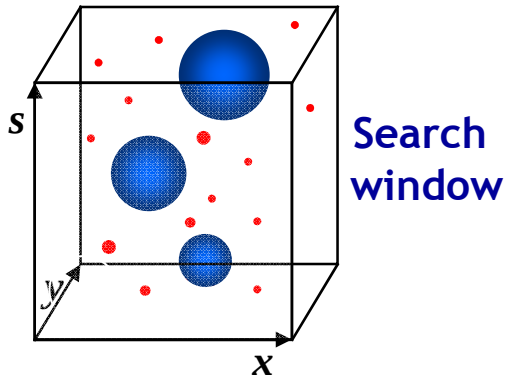
- Scale-invariant feature selection
  - Scale-invariant interest points
  - Rescale extracted patches
  - Match to constant-size codebook



- Generate scale votes
  - Scale as 3rd dimension in voting space

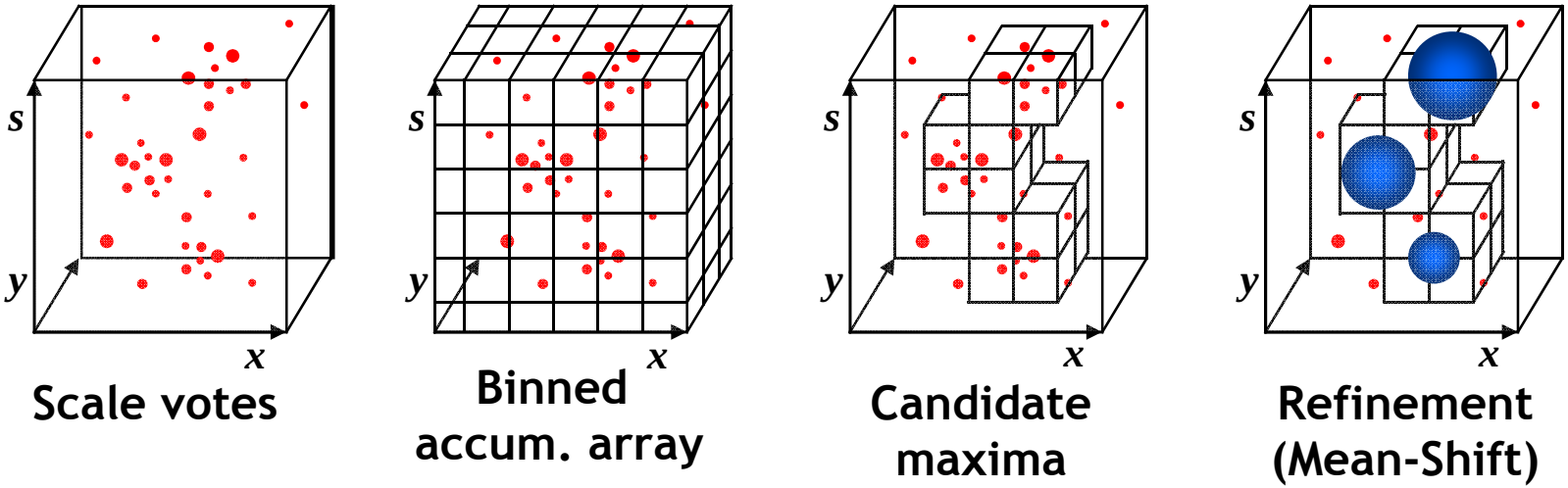
$$
\begin{aligned}
x_{vote} &= x_{img} - x_{occ}(s_{img}/s_{occ}) \\
y_{vote} &= y_{img} - y_{occ}(s_{img}/s_{occ}) \\
s_{vote} &= (s_{img}/s_{occ}).
\end{aligned}
$$

  - Search for maxima in 3D voting space



Source: Bastian Leibe

Scale Voting: Efficient Computation



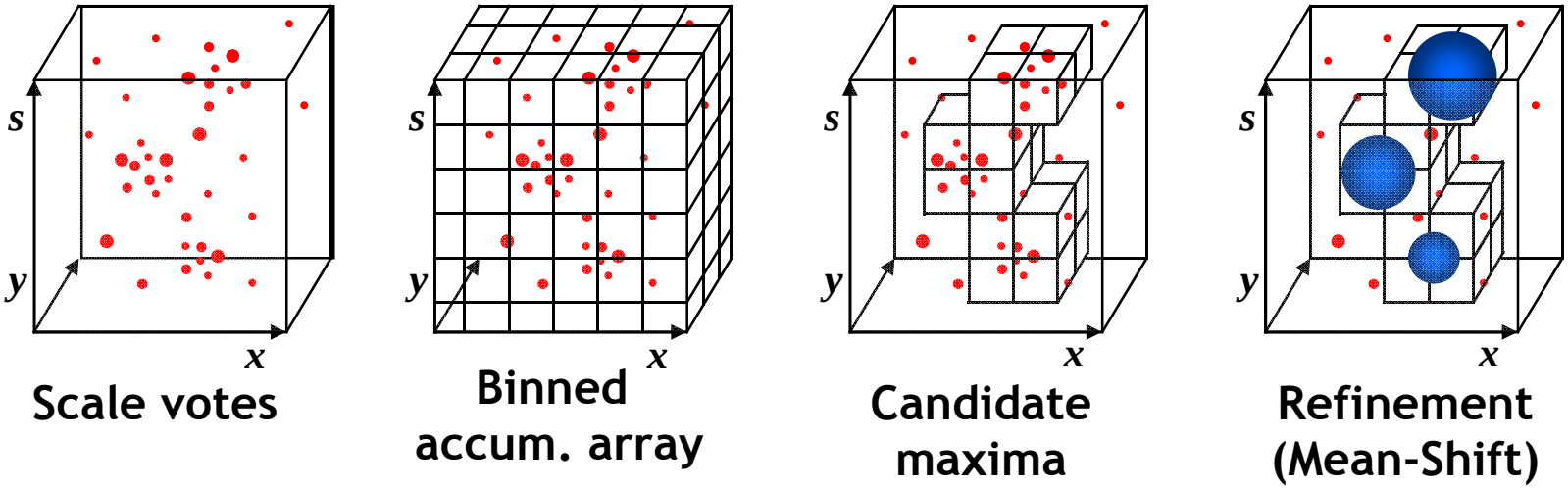
- Continuous Generalized Hough Transform
  - Binned accumulator array similar to standard Gen. Hough Transf.
  - Quickly identify candidate maxima locations
  - Refine locations by Mean-Shift search only around those points

  $\Rightarrow$ Avoid quantization effects by keeping exact vote locations.

  $\Rightarrow$ Mean-shift interpretation as kernel prob. density estimation.

Source: Bastian Leibe

Scale Voting: Efficient Computation



- Scale-adaptive Mean-Shift search for refinement
  - Increase search window size with hypothesis scale
  - Scale-adaptive *balloon density estimator*

Source: Bastian Leibe

Detection Results

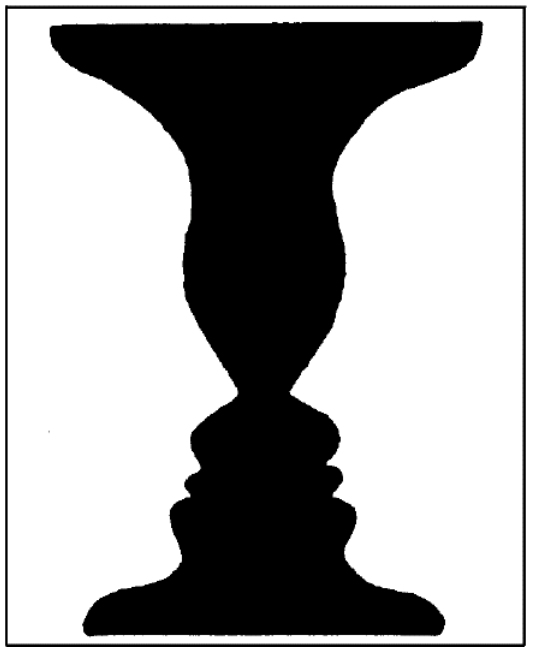
- Qualitative Performance
  - Recognizes different kinds of objects
  - Robust to clutter, occlusion, noise, low contrast

Source: Bastian Leibe

Figure-Ground Segregation

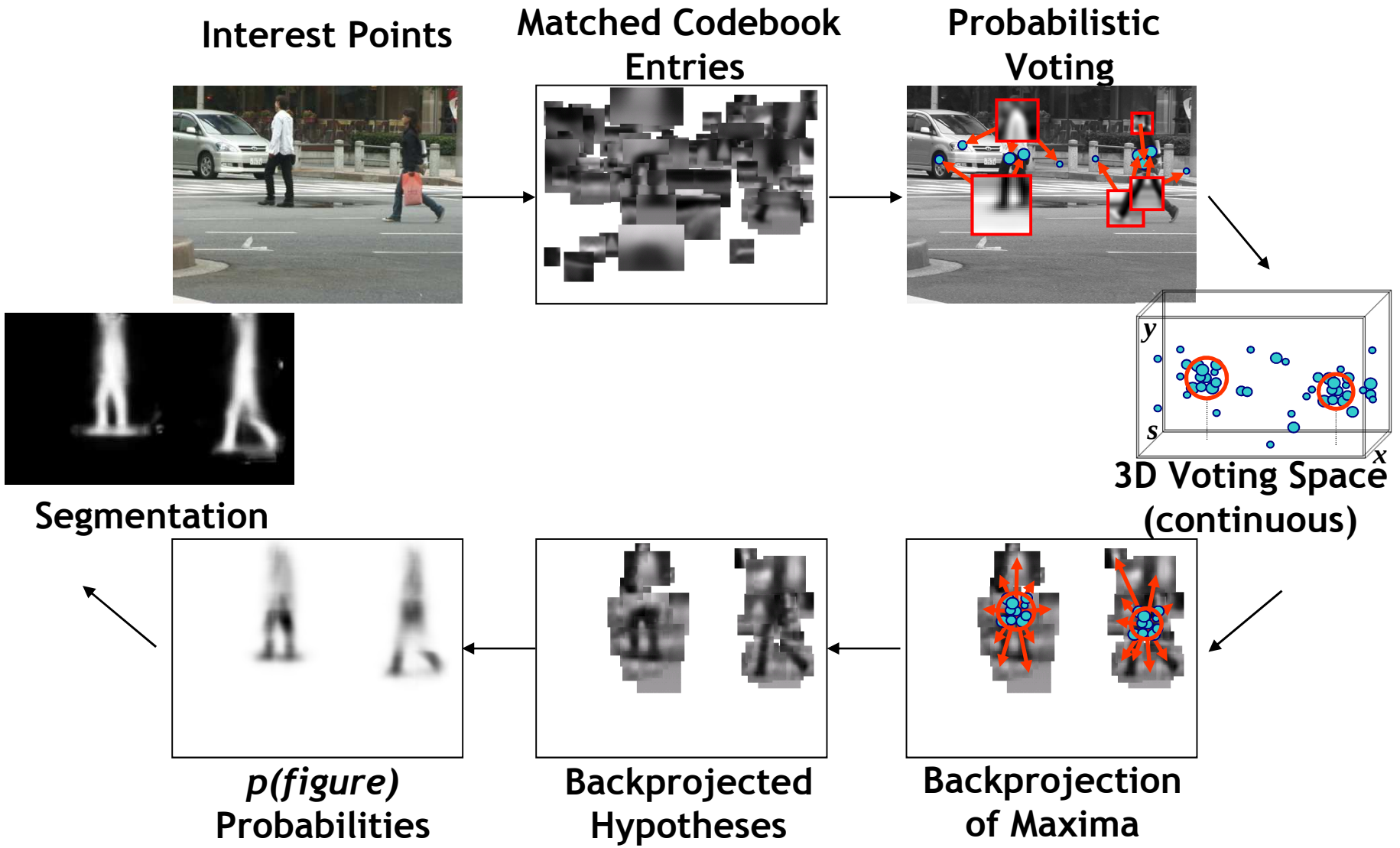
- What happens first – segmentation or recognition?

- Problem extensively studied in Psychophysics
- Experiments with ambiguous figure-ground stimuli
- Results:
  - Evidence that object recognition can and does operate before figure-ground organization
  - Interpreted as Gestalt cue *familiarity*.



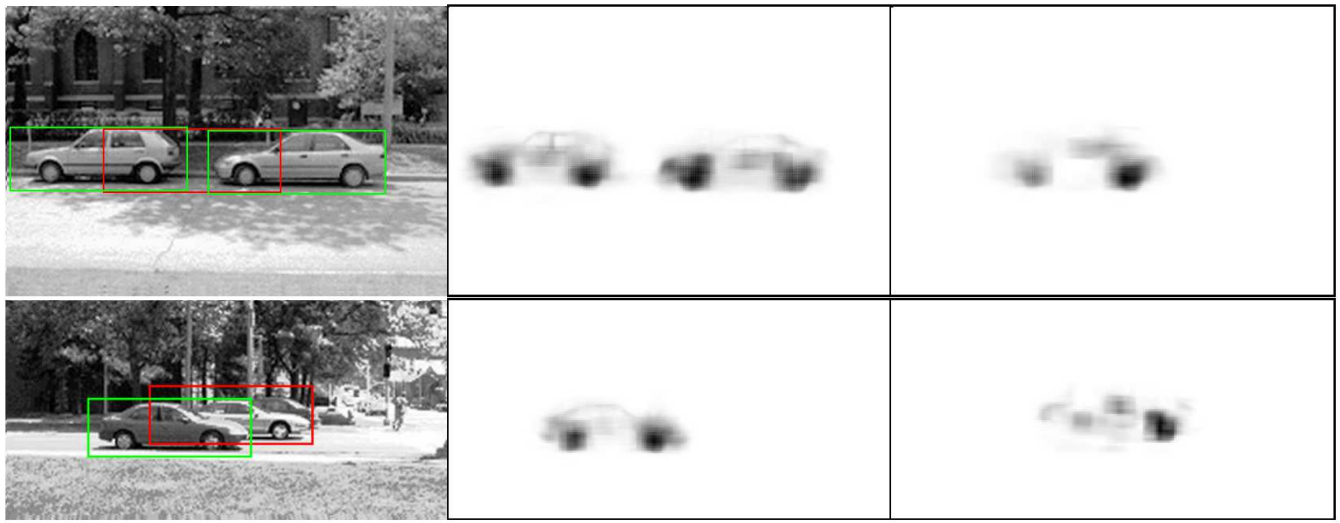
M.A. Peterson, "Object Recognition Processes Can and Do Operate Before Figure-Ground Organization", *Cur. Dir. in Psych. Sc.*, 3:105-111, 1994.

ISM – Top-Down Segmentation



[Leibe, Leonardis, Schiele, SLCV'04; IJCV'08]

Top-Down Segmentation: Motivation



- Secondary hypotheses ("mixtures of cars/cows/etc.")
  - Desired property of algorithm! $\Rightarrow$ robustness to occlusion
  - Standard solution: reject based on bounding box overlap
    $\Rightarrow$ Problematic - may lead to missing detections!

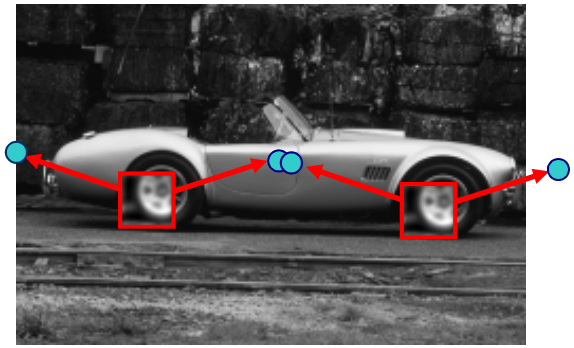
Source: Bastian Leibe

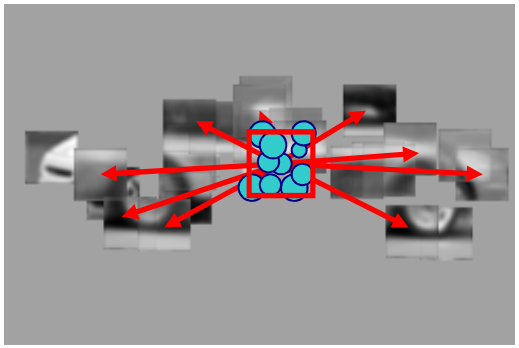
Top-Down Segmentation: Motivation

- Secondary hypotheses ("mixtures of cars/cows/etc.")
  - Desired property of algorithm! $\Rightarrow$ robustness to occlusion
  - Standard solution: reject based on bounding box overlap
    $\Rightarrow$ Problematic - may lead to missing detections!
    $\Rightarrow$ Use segmentations to resolve ambiguities instead.
  - Basic idea: each observed pixel can only be explained by (at most) one detection.

Source: Bastian Leibe

Segmentation: Probabilistic Formulation





- Influence of patch on object hypothesis (vote weight)

$$p(f,\ell|o_n,x)=\frac{\sum_i p(o_n,x|C_i)p(C_i|f)p(f,\ell)}{p(o_n,x)}$$

Backprojection to features *f* and pixels p:

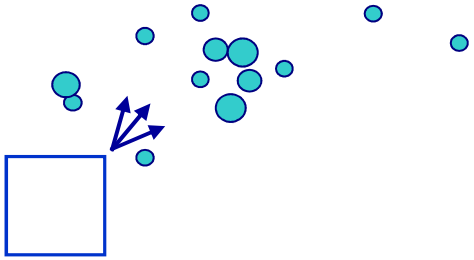
$$p(\mathbf{p}=figure|o_n,x)=\sum_{\mathbf{p}\in(f,\ell)} p(\mathbf{p}=figure|f,\ell,o_n,x)\,p(f,\ell|o_n,x)$$

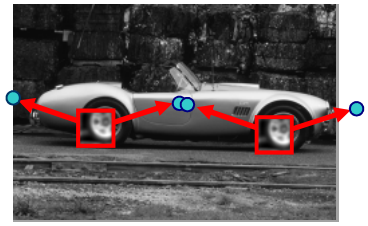
Segmentation information — **Influence on object hypothesis**

[Leibe, Leonardis, Schiele, SLCV'04; IJCV'08]

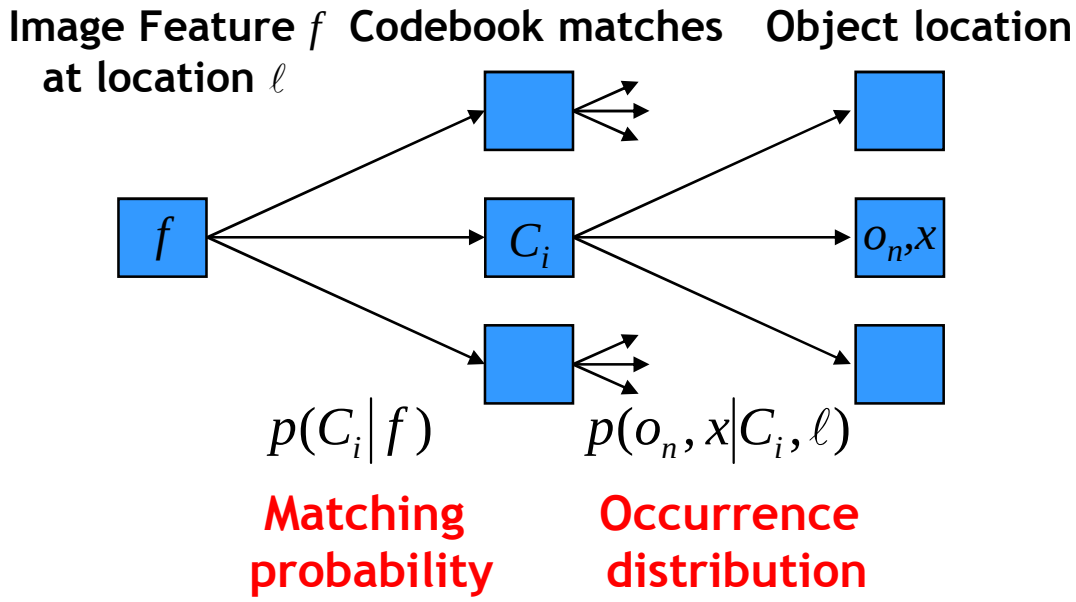
Derivation: ISM Recognition

- Algorithm stages
  1. Voting
  2. Mean-shift search
  3. Backprojection





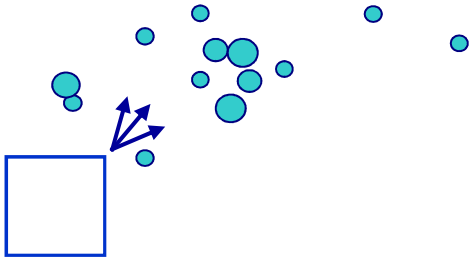
- Vote weights: contribution of a single feature f

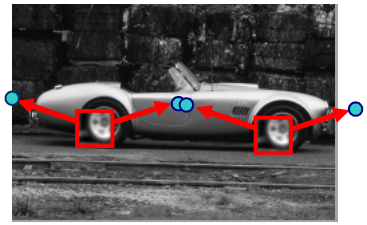


[Leibe, Leonardis, Schiele, SLCV'04; IJCV'08]

Derivation: ISM Recognition

- Algorithm stages
  1. Voting
  2. Mean-shift search
  3. Backprojection





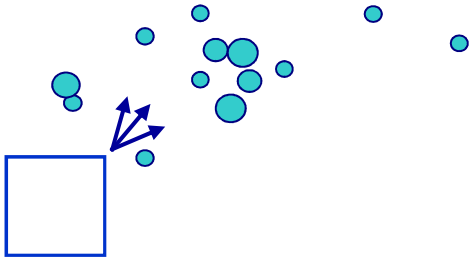
- Vote weights: contribution of a single feature f
  - Probability that object o_n occurs at location x given (f,ℓ)

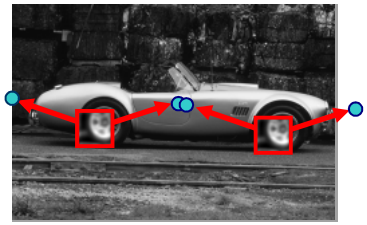
$$p(o_n, x \mid f, \ell) = \sum_i \underbrace{p(C_i \mid f)}_{\text{Matching probability}} \quad \underbrace{p(o_n, x \mid C_i, \ell)}_{\text{Occurrence distribution}}$$

[Leibe, Leonardis, Schiele, SLCV'04; IJCV'08]

Derivation: ISM Recognition

- Algorithm stages
  1. Voting
  2. Mean-shift search
  3. Backprojection





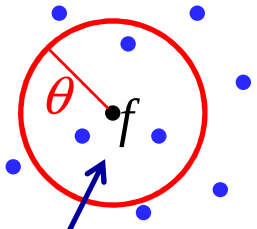
- Vote weights: contribution of a single feature f
  - Probability that object o_n occurs at location x given (f,ℓ)

$$p(o_n, x \mid f, \ell) = \sum_i p(C_i \mid f) \quad p(o_n, x \mid C_i, \ell)$$

  - How to measure those probabilities?

$$p(C_i \mid f) = \frac{1}{|C|}, \quad \text{where} \quad C = \{C_i \mid d(C_i, f) \le \theta\}$$

$$p(o_n, x \mid C_i, \ell) = \frac{1}{\#occurrences(C_i)}$$

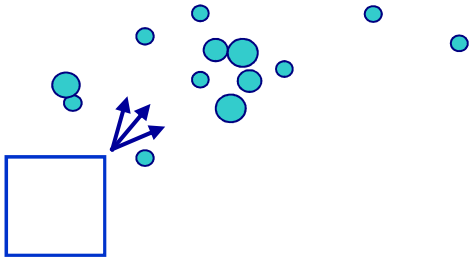


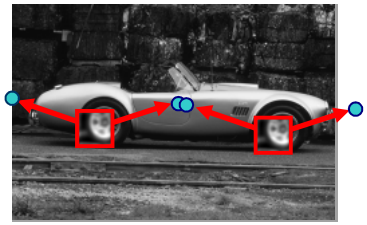
Activated codebook entries

[Leibe, Leonardis, Schiele, SLCV'04; IJCV'08]

Derivation: ISM Recognition

- Algorithm stages
  1. Voting
  2. Mean-shift search
  3. Backprojection





- Vote weights: contribution of a single feature f
  - Probability that object o_n occurs at location x given (f,ℓ)

$$p(o_n, x \mid f, \ell) = \sum_i p(C_i \mid f) \quad p(o_n, x \mid C_i, \ell)$$

  - Likelihood of the observed features given the object hypothesis

$$p(f,\ell \mid o_n, x) = \frac{p(o_n, x \mid f,\ell)\, p(f,\ell)}{p(o_n, x)} = \frac{\sum_i p(o_n, x \mid C_i, \ell)\, p(C_i \mid f)\, p(f,\ell)}{p(o_n, x)}$$

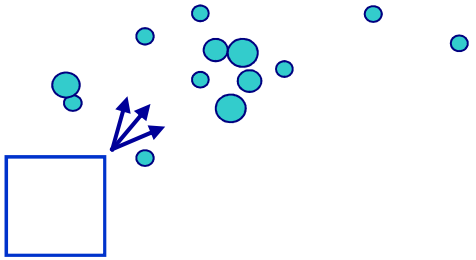
$p(f,\ell)$: **Indicator variable for sampled features**

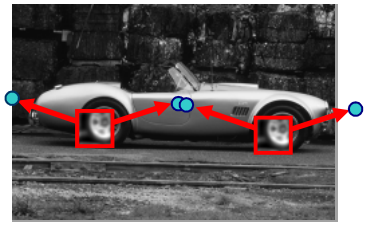
$p(o_n, x)$: **Prior for the object location**

[Leibe, Leonardis, Schiele, SLCV'04; IJCV'08]

Derivation: ISM Recognition

- Algorithm stages
  1. Voting
  2. Mean-shift search
  3. Backprojection





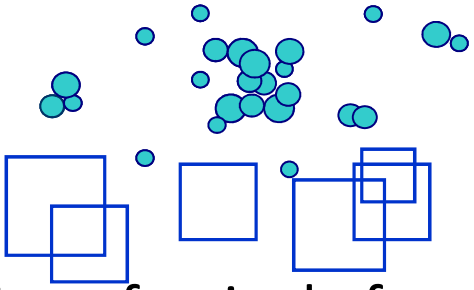
- Vote weights: contribution of a single feature f

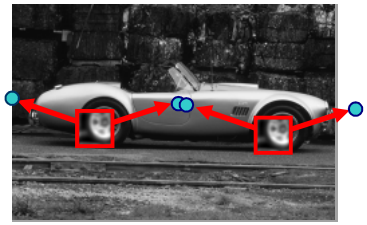
$$p(f,\ell \mid o_n, x) = \frac{p(o_n, x \mid f,\ell)\, p(f,\ell)}{p(o_n, x)} = \frac{\sum_i p(o_n, x \mid C_i, \ell)\, p(C_i \mid f)\, p(f,\ell)}{p(o_n, x)}$$

[Leibe, Leonardis, Schiele, SLCV'04; IJCV'08]

Derivation: ISM Recognition

- Algorithm stages
  1. Voting
  2. Mean-shift search
  3. Backprojection





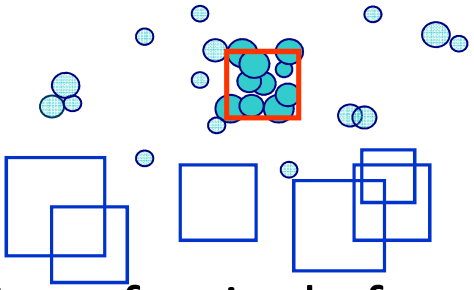
- Vote weights: contribution of a single feature f

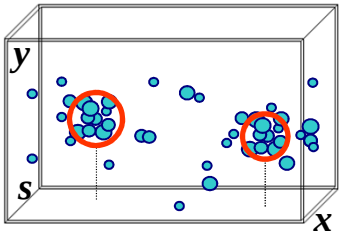
$$p(f,\ell \mid o_n, x) = \frac{p(o_n, x \mid f,\ell)\, p(f,\ell)}{p(o_n, x)} = \frac{\sum_i p(o_n, x \mid C_i, \ell)\, p(C_i \mid f)\, p(f,\ell)}{p(o_n, x)}$$

[Leibe, Leonardis, Schiele, SLCV'04; IJCV'08]

Derivation: ISM Recognition

- Algorithm stages
  1. Voting
  2. Mean-shift search
  3. Backprojection





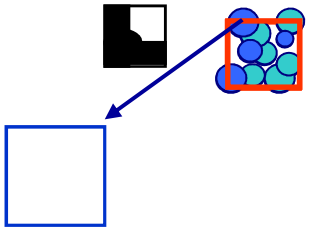
- Vote weights: contribution of a single feature f

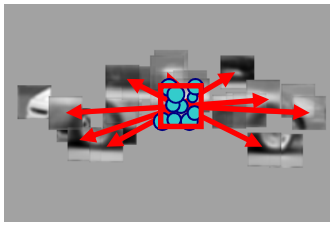
$$p(f,\ell \mid o_n, x) = \frac{p(o_n, x \mid f,\ell)\, p(f,\ell)}{p(o_n, x)} = \frac{\sum_i p(o_n, x \mid C_i, \ell)\, p(C_i \mid f)\, p(f,\ell)}{p(o_n, x)}$$

[Leibe, Leonardis, Schiele, SLCV'04; IJCV'08]

Derivation: ISM Top-Down Segmentation

- Algorithm stages
  1. Voting
  2. Mean-shift search
  3. Backprojection





- Vote weights: contribution of a single feature f

$$p(f,\ell \mid o_n, x) = \frac{p(o_n, x \mid f,\ell)\, p(f,\ell)}{p(o_n, x)} = \sum_i \frac{p(o_n, x \mid C_i, \ell)\, p(C_i \mid f)\, p(f,\ell)}{p(o_n, x)}$$

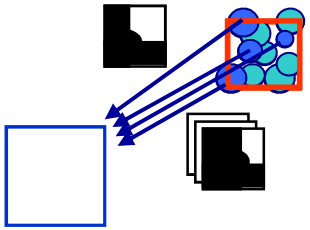
- Figure-ground backprojection

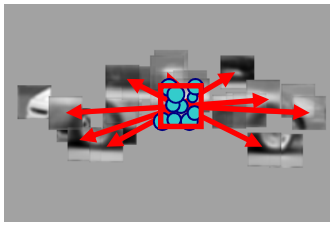
$$p(\mathbf{p} = figure \mid o_n, x, f, C_i, \ell) = \underbrace{p(\mathbf{p} = fig. \mid o_n, x, C_i, \ell)}_{\text{Fig./Gnd. label for each occurrence}} \underbrace{\frac{p(o_n, x \mid C_i, \ell)\, p(C_i \mid f)\, p(f,\ell)}{p(o_n, x)}}_{\text{Influence on object hypothesis}}$$

[Leibe, Leonardis, Schiele, SLCV'04; IJCV'08]

Derivation: ISM Top-Down Segmentation

- Algorithm stages
  1. Voting
  2. Mean-shift search
  3. Backprojection





- Vote weights: contribution of a single feature f

$$p(f,\ell \mid o_n, x) = \frac{p(o_n, x \mid f,\ell)\, p(f,\ell)}{p(o_n, x)} = \sum_i \frac{p(o_n, x \mid C_i, \ell)\, p(C_i \mid f)\, p(f,\ell)}{p(o_n, x)}$$

- Figure-ground backprojection

$$p(\mathbf{p} = figure \mid o_n, x, f, \ell) = \sum_i p(\mathbf{p} = fig. \mid o_n, x, C_i, \ell) \frac{p(o_n, x \mid C_i, \ell) p(C_i \mid f) p(f,\ell)}{p(o_n, x)}$$

Marginalize over all codebook entries matched to f

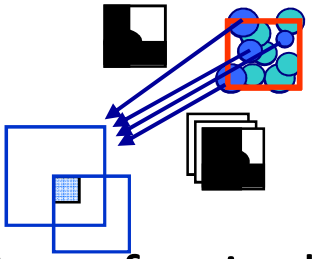
Fig./Gnd. label for each occurrence

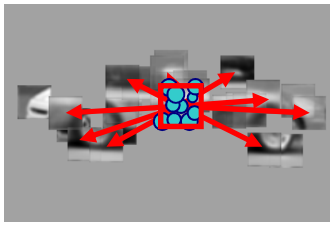
Influence on object hypothesis

[Leibe, Leonardis, Schiele, SLCV'04; IJCV'08]

Derivation: ISM Top-Down Segmentation

- Algorithm stages
  1. Voting
  2. Mean-shift search
  3. Backprojection





- Vote weights: contribution of a single feature f

$$p(f,\ell \mid o_n, x) = \frac{p(o_n, x \mid f,\ell)\, p(f,\ell)}{p(o_n, x)} = \sum_i \frac{p(o_n, x \mid C_i, \ell)\, p(C_i \mid f)\, p(f,\ell)}{p(o_n, x)}$$

- Figure-ground backprojection

$$p(\mathbf{p} = figure \mid o_n, x) = \sum_{\mathbf{p} \in (f,\ell)} \sum_i \underbrace{p(\mathbf{p} = fig. \mid o_n, x, C_i, \ell)}_{\text{Fig./Gnd. label for each occurrence}} \underbrace{\frac{p(o_n, x \mid C_i, \ell)\, p(C_i \mid f)\, p(f,\ell)}{p(o_n, x)}}_{\text{Influence on object hypothesis}}$$

Marginalize over all features containing pixel p

[Leibe, Leonardis, Schiele, SLCV'04; IJCV'08]

Top-Down Segmentation Algorithm

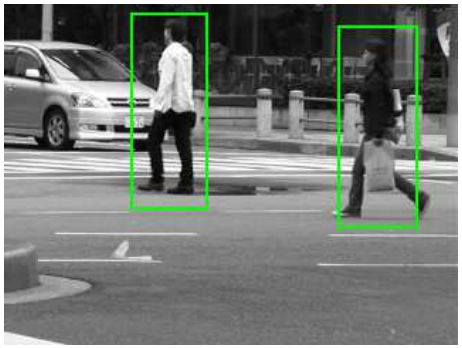
Algorithm 5 The top-segmentation algorithm.

// Given: hypothesis h and supporting votes $\mathcal{V}_h$.
for all supporting votes $(x, w, occ, \ell) \in \mathcal{V}_h$ **do**
  Let img_{mask} be the segmentation mask corresponding to occ.
  Let sz be the size at which the interest region ℓ was sampled.
  Rescale img_{mask} to sz.
  $u_0 \leftarrow (\ell_x - \frac{1}{2}\mathrm{sz})$
  $v_0 \leftarrow (\ell_y - \frac{1}{2}\mathrm{sz})$
  for all $u \in [0, \mathrm{sz} - 1]$ **do**
    for all $v \in [0, \mathrm{sz} - 1]$ **do**
      $img_{pfig}(u - u_0, v - v_0) \mathrel{+}= w \cdot img_{mask}(u, v)$
      $img_{pgnd}(u - u_0, v - v_0) \mathrel{+}= w \cdot (1 - img_{mask}(u, v))$
    end for
  end for
end for

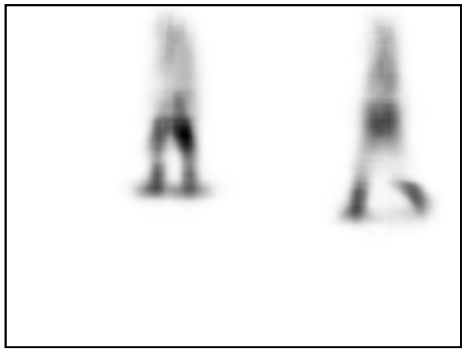
- This may sound quite complicated, but it boils down to a very simple algorithm…

[Leibe, Leonardis, Schiele, SLCV'04; IJCV'08]

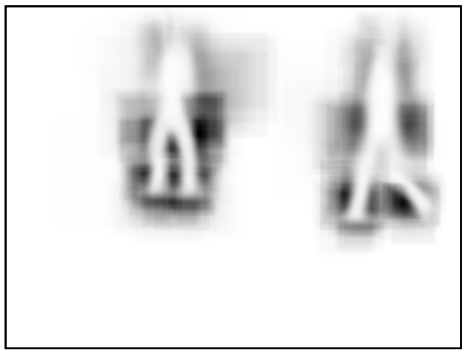
Segmentation



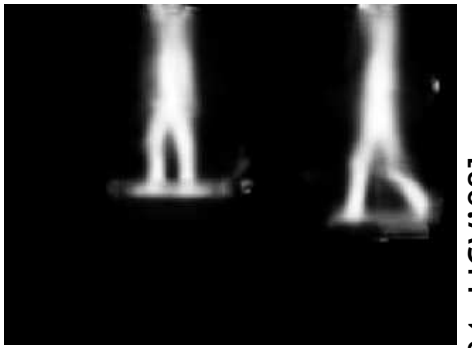
Original image



p(figure)



p(ground)



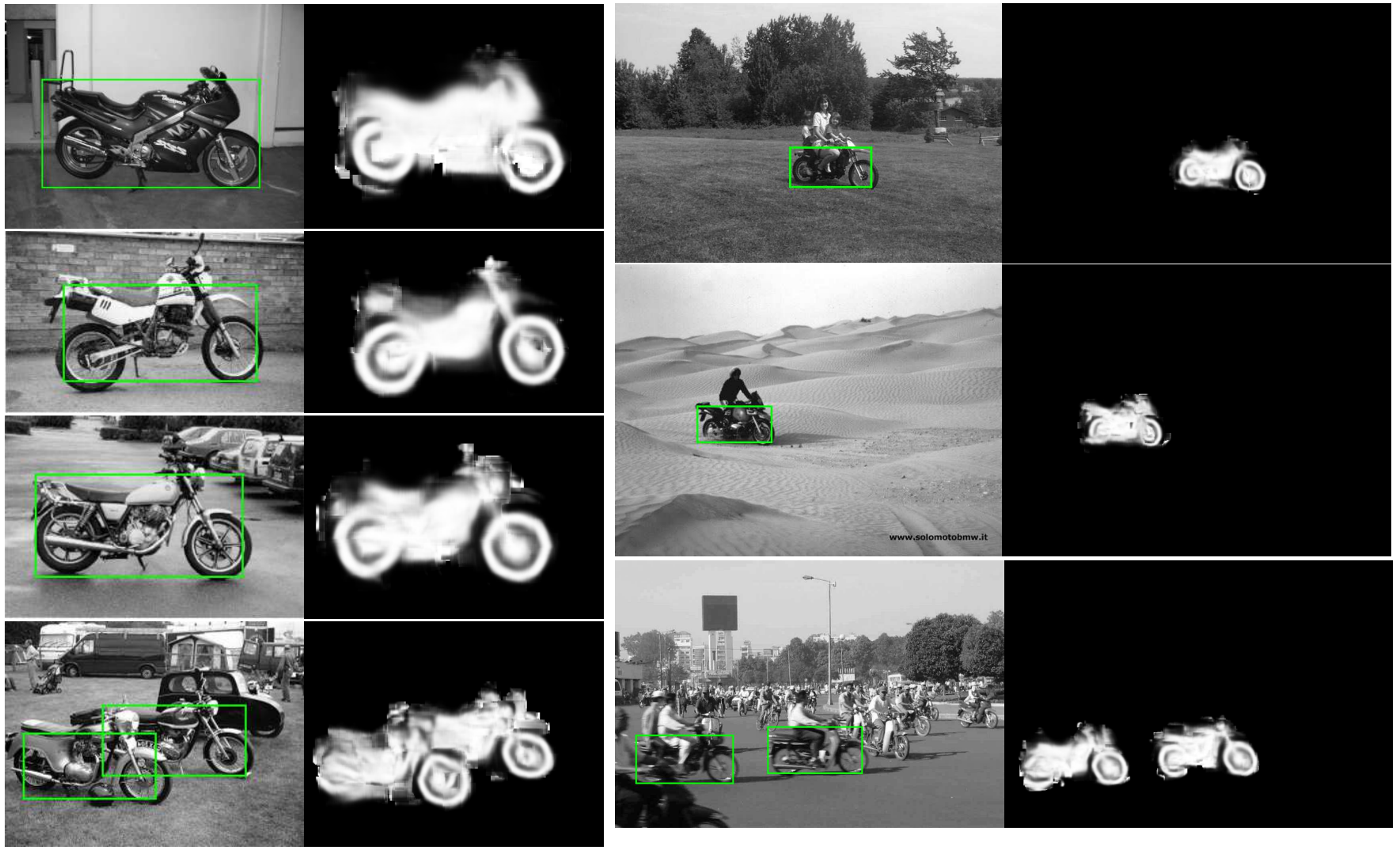
Segmentation

$$\frac{p(figure)}{p(ground)}$$

[Leibe, Leonardis, Schiele, SLCV'04; IJCV'08]

- Interpretation of *p(figure)* map
  - per-pixel confidence in object hypothesis
  - Use for hypothesis verification

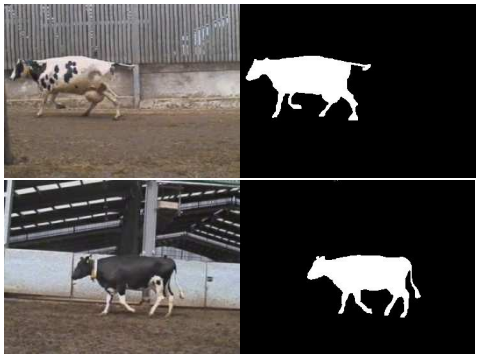
Example Results: Motorbikes



[Leibe, Leonardis, Schiele, SLCV'04; IJCV'08]

Example Results: Cows

- Training
  - 112 hand-segmented images

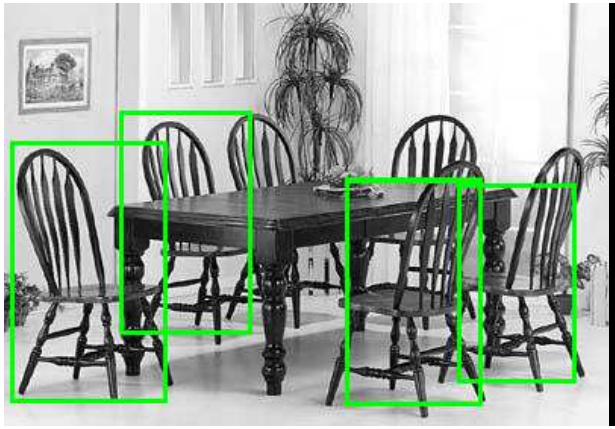


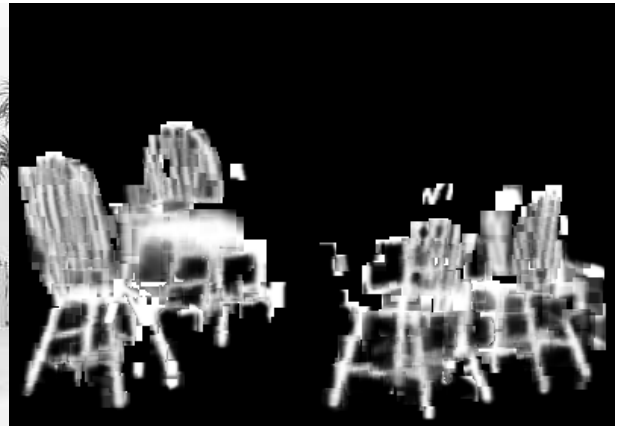
- Results on novel sequences:

Single-frame recognition - No temporal continuity used!

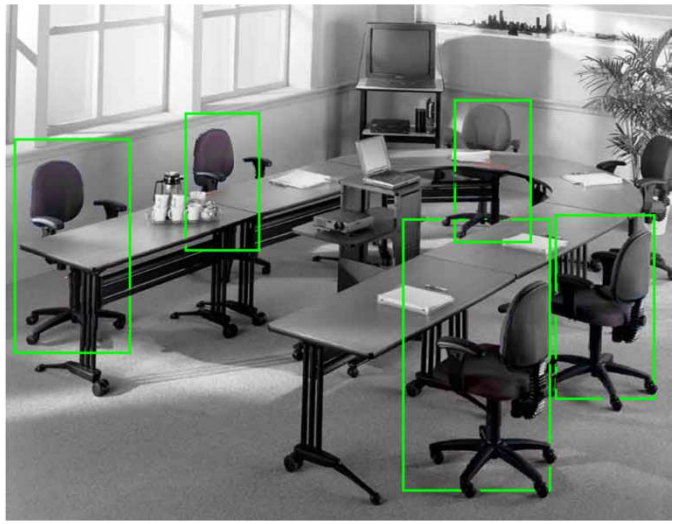
[Leibe, Leonardis, Schiele, SLCV'04; IJCV'08]

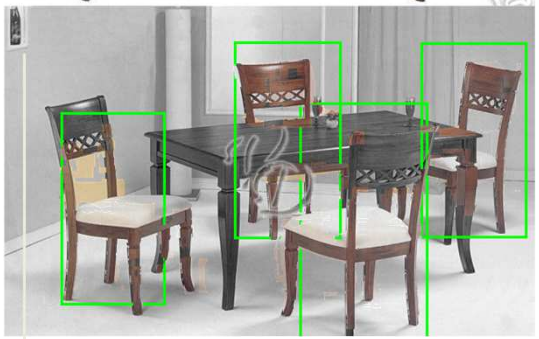
Example Results: Chairs

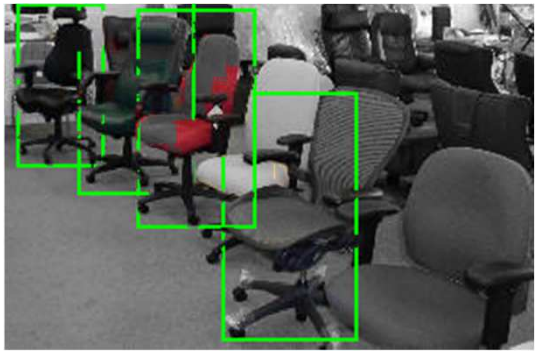


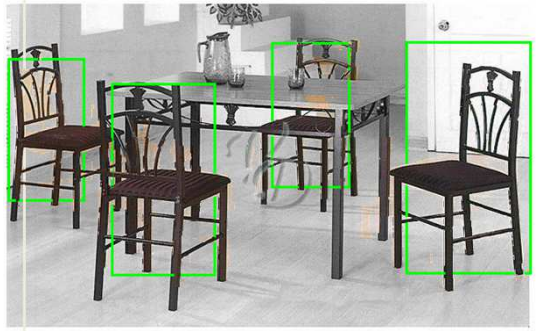


Dining room chairs





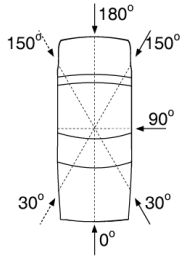




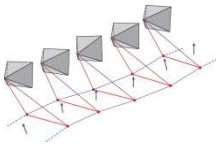
Office chairs

Source: Bastian Leibe

Detections Using Ground Plane Constraints



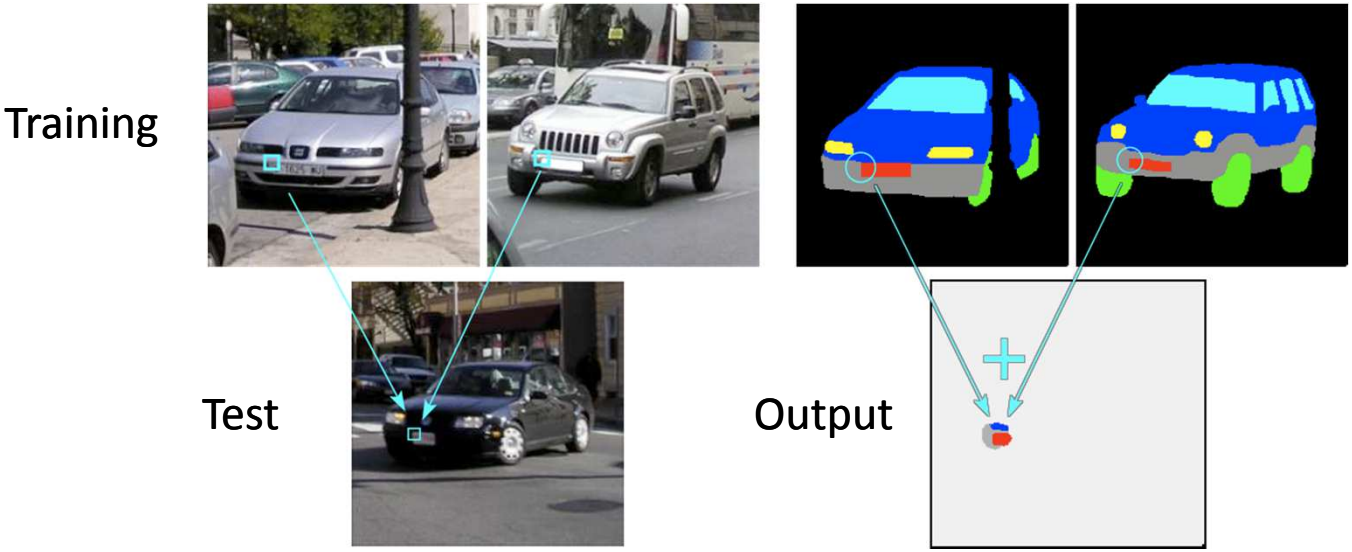
Battery of 5 ISM detectors for different car views



left camera 1175 frames

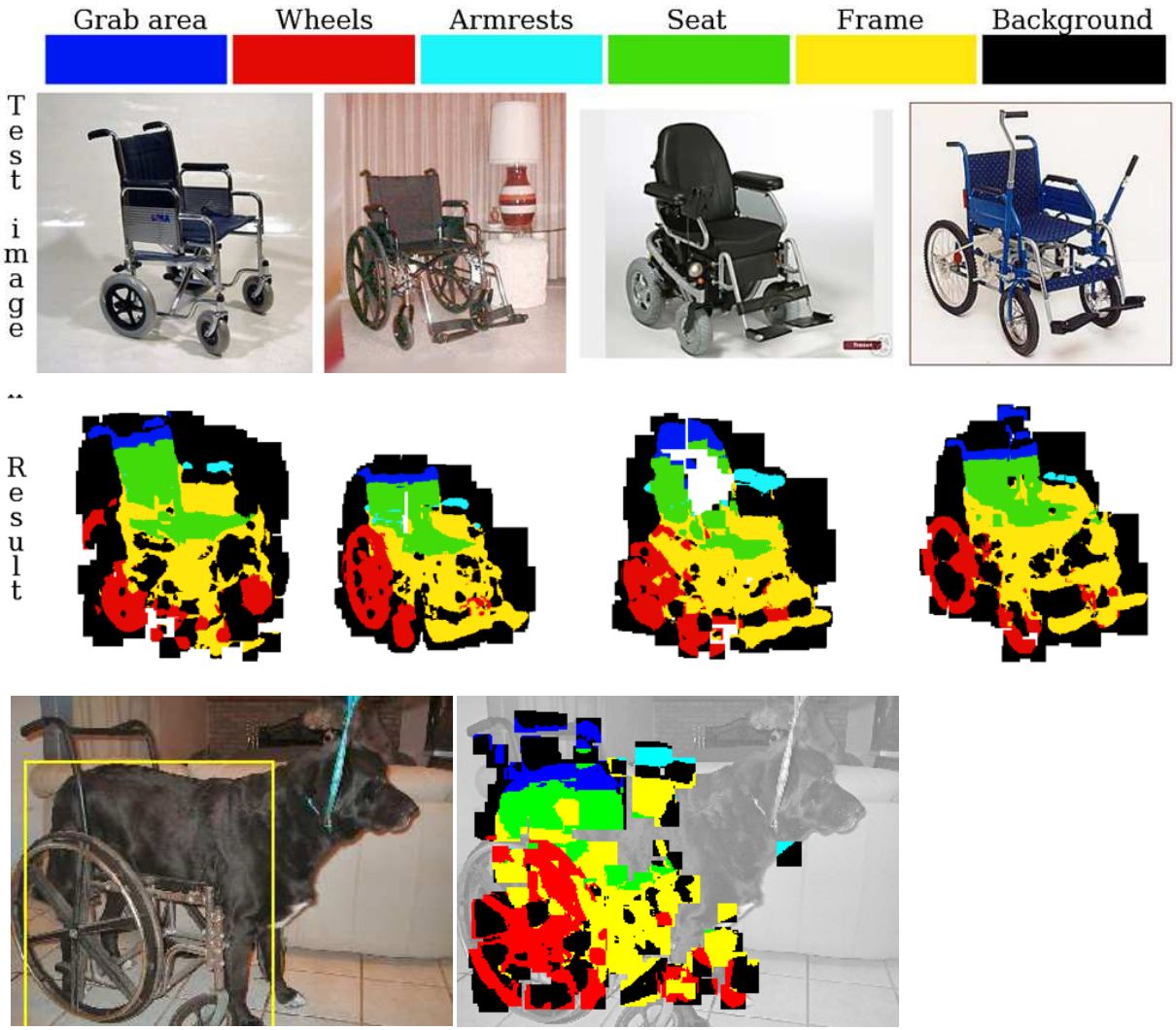
[Leibe, Leonardis, Schiele, SLCV'04; IJCV'08]

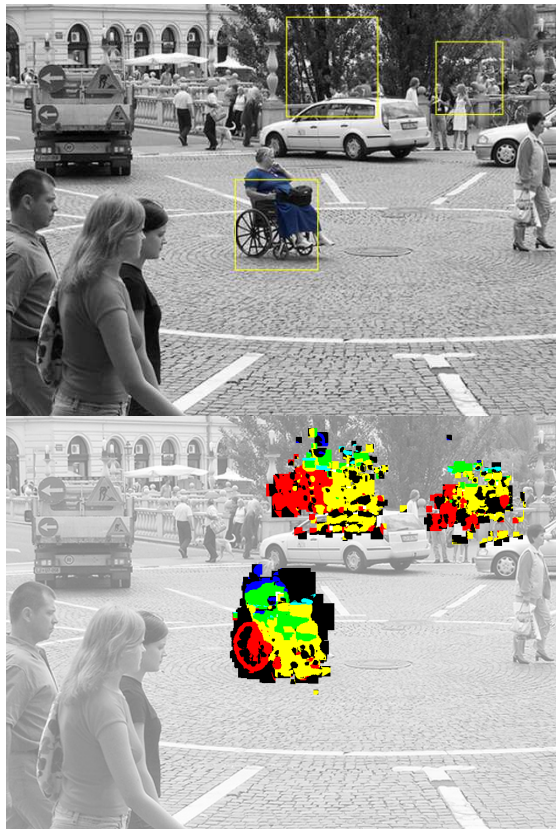
Inferring Other Information: Part Labels (1)



[Thomas, Ferrari, Tuytelaars, Leibe, Van Gool, 3DRR'07; RSS'08]

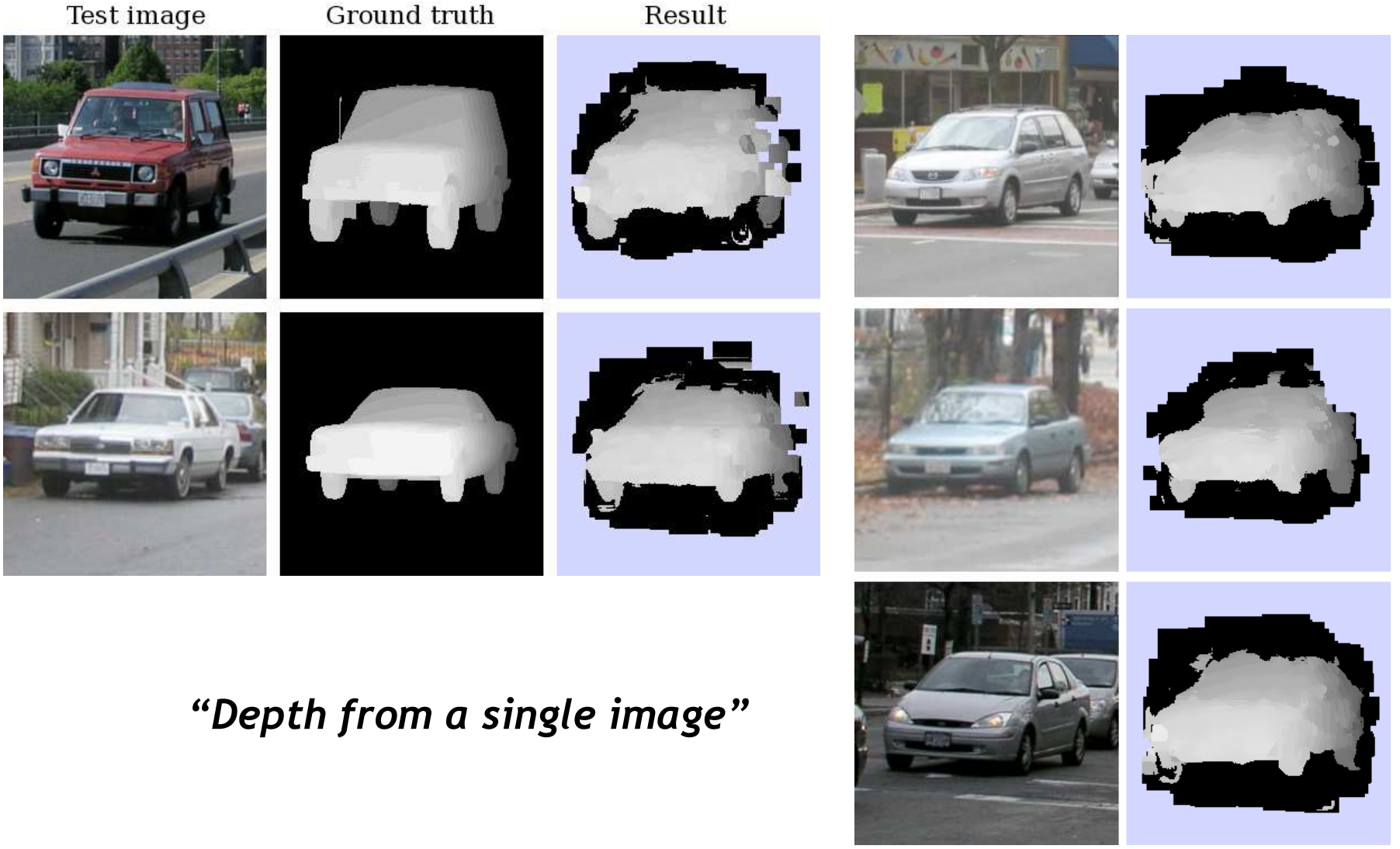
Inferring Other Information: Part Labels (2)





[Thomas, Ferrari, Tuytelaars, Leibe, Van Gool, 3DRR'07; RSS'08]

Inferring Other Information: Depth Maps

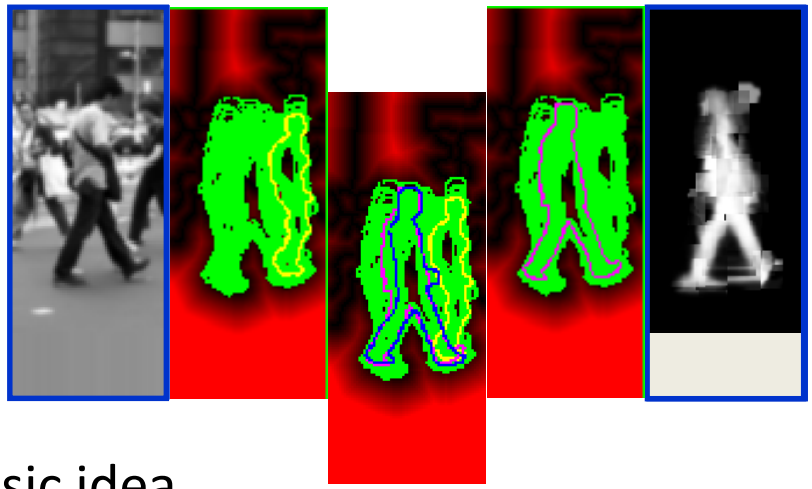


"Depth from a single image"

[Thomas, Ferrari, Tuytelaars, Leibe, Van Gool, 3DRR'07; RSS'08]

Extension: Estimating Articulation

- Try to fit silhouette to detected person

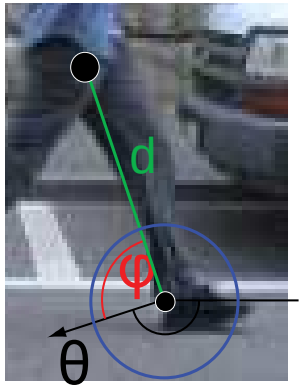


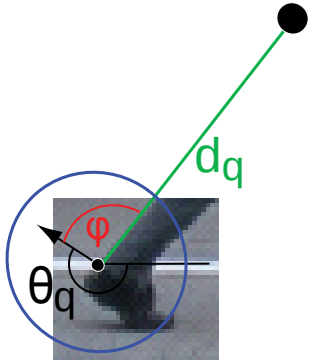
- Basic idea
  - Search for the silhouette that simultaneously optimizes the
    - Chamfer match to the distance-transformed edge image
    - Overlap with the top-down segmentation
  - Enforces global consistency
  - Caveat: introduces again reliance on global model

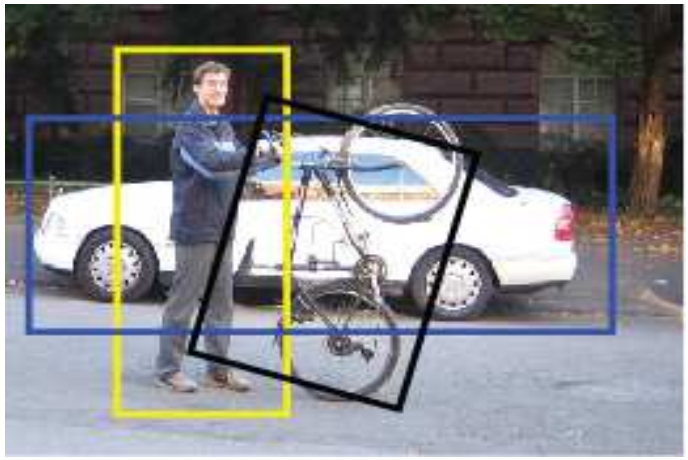
[Leibe, Seemann, Schiele, CVPR'05]

Extension: Rotation-Invariant Detection

- Polar instead of Cartesian voting scheme







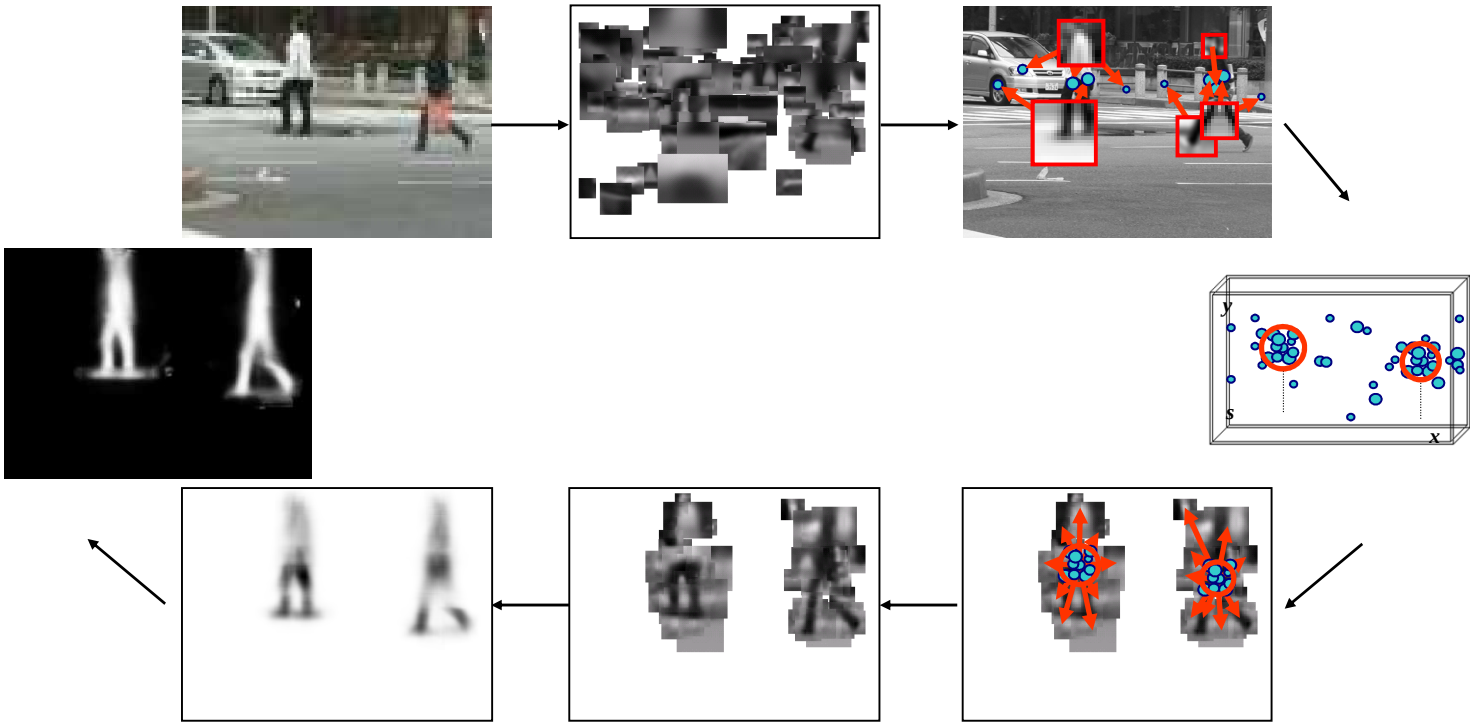
- Benefits:
  - Recognize objects under image-plane rotations
  - Possibility to share parts between articulations.
- Caveats:
  - Rotation invariance should only be used when it's really needed. (Also increases false positive detections)

[Mikolajczyk, Leibe, Schiele, CVPR'06]

Sometimes, Rotation Invariance Is Needed…

[Mikolajczyk et al., CVPR'06]

You Can Try It At Home…

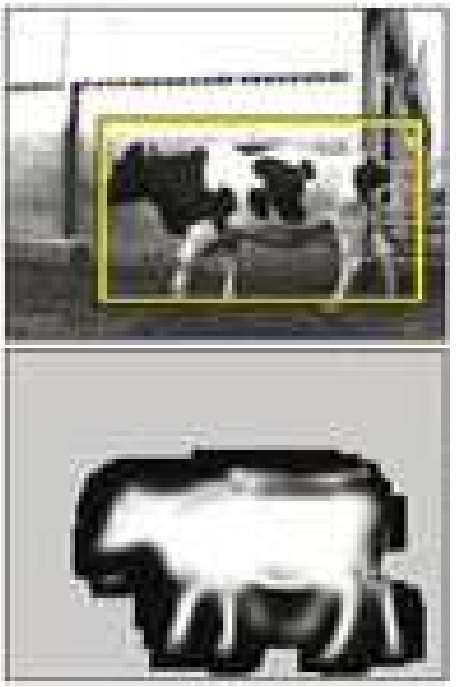


- Linux binaries available
  - Including datasets & several pre-trained detectors
  - http://www.vision.ee.ethz.ch/bleibe/code

Source: Bastian Leibe

Discussion: Implicit Shape Model

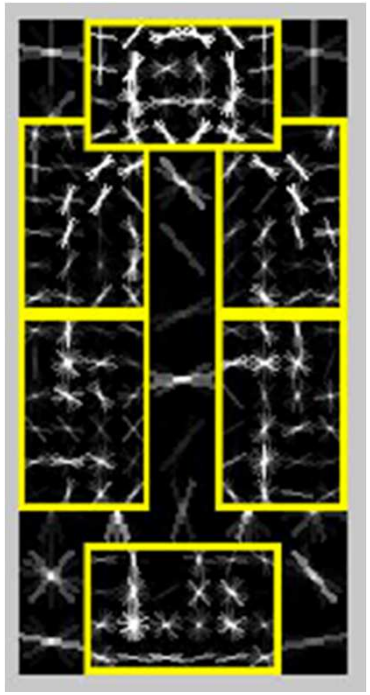
- Pros:
  - Works well for many different object categories
    - Both rigid and articulated objects
  - Flexible geometric model
    - Can recombine parts seen on different training examples
  - Learning from relatively few (50-100) training examples
  - Optimized for detection, good localization properties
- Cons:
  - Needs supervised training data
    - Object bounding boxes for detection
    - Reference segmentations for top-down segm.
  - Only weak geometric constraints
    - Result segmentations may contain superfluous body parts.
  - Purely representative model
    - No discriminative learning



Source: Bastian Leibe

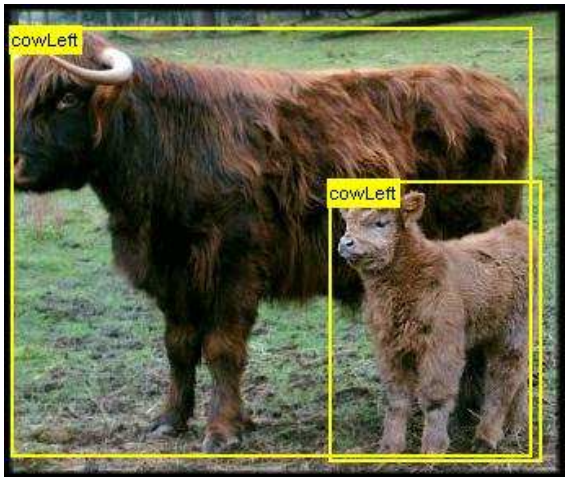
What we will learn today?

- Implicit Shape Model
  - Representation
  - Recognition
  - Experiments and results
- Deformable Models
  - The PASCAL challenge
  - Latent SVM Model



Object Detection – the PASCAL Challenge

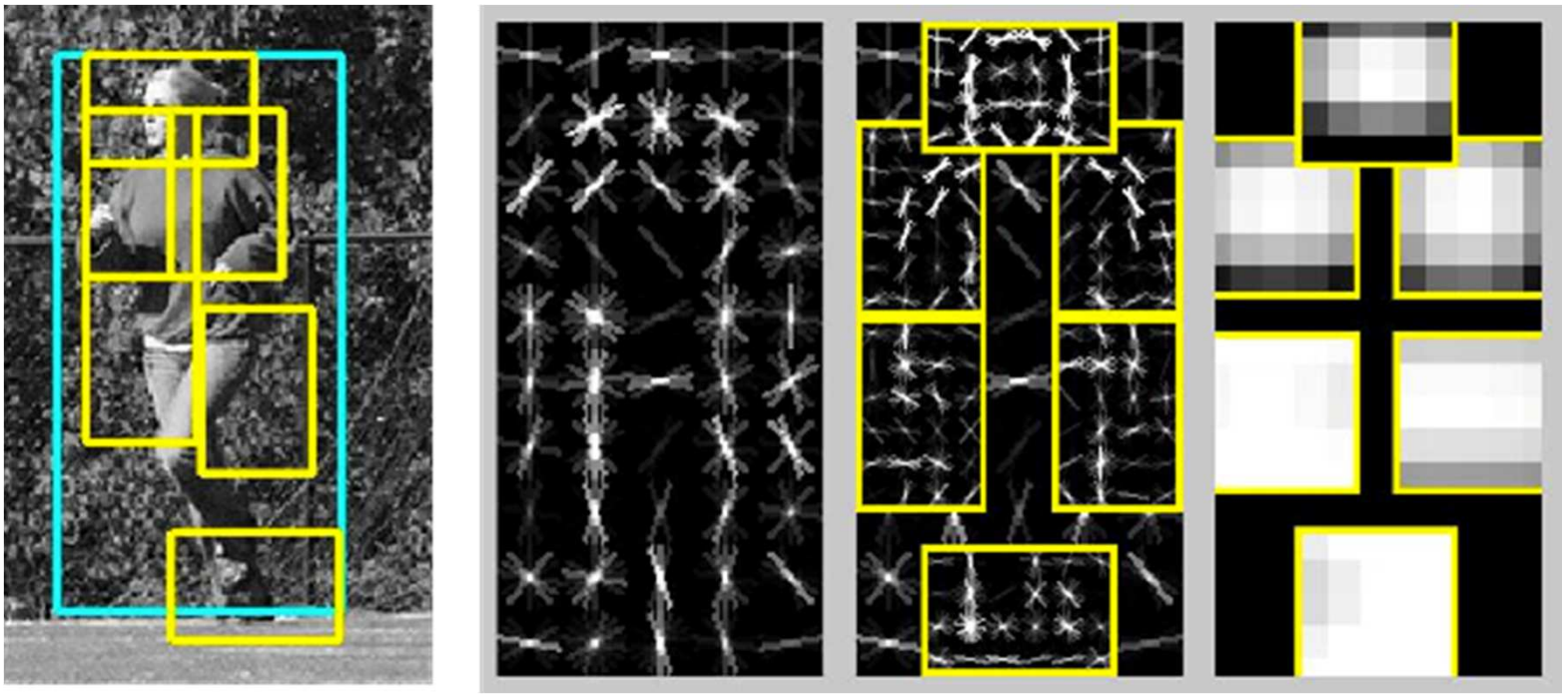
- ~10,000 images, with ~25,000 target objects.
  - Objects from 20 categories (person, car, bicycle, cow, table...).
  - Objects are annotated with labeled bounding boxes.



Source: Pedro Felzenswalb


aeroplane
Bicycle Assembly Area
bicycleRight
personLeftTrunc
bus
carLeftTrunc
car
boatTrunc
personTruncDifficult
boat
personTrunc
horseRightTrunc
Morgan
car
chair
diningtable
chairLeft
chairRight
chair
motorbikeRear
motorbikeRear
dogFrontal
dogLeftTrunc

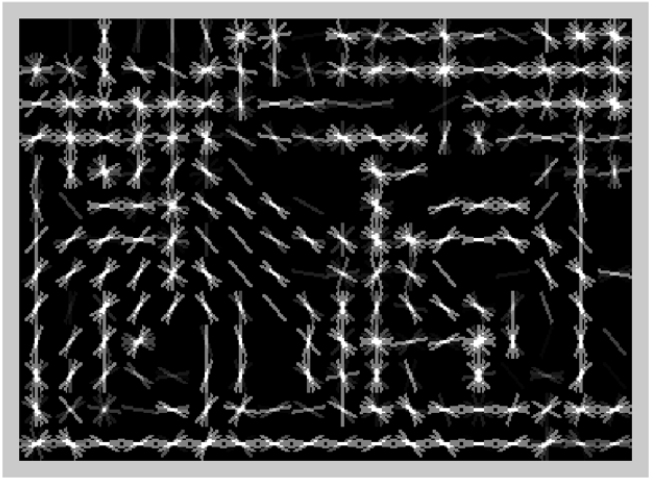
Latent SVM Model: an Overview



detection root filter part filters deformation models

Source: Pedro Felzenswalb

Histogram of Oriented Gradient (HOG) Features

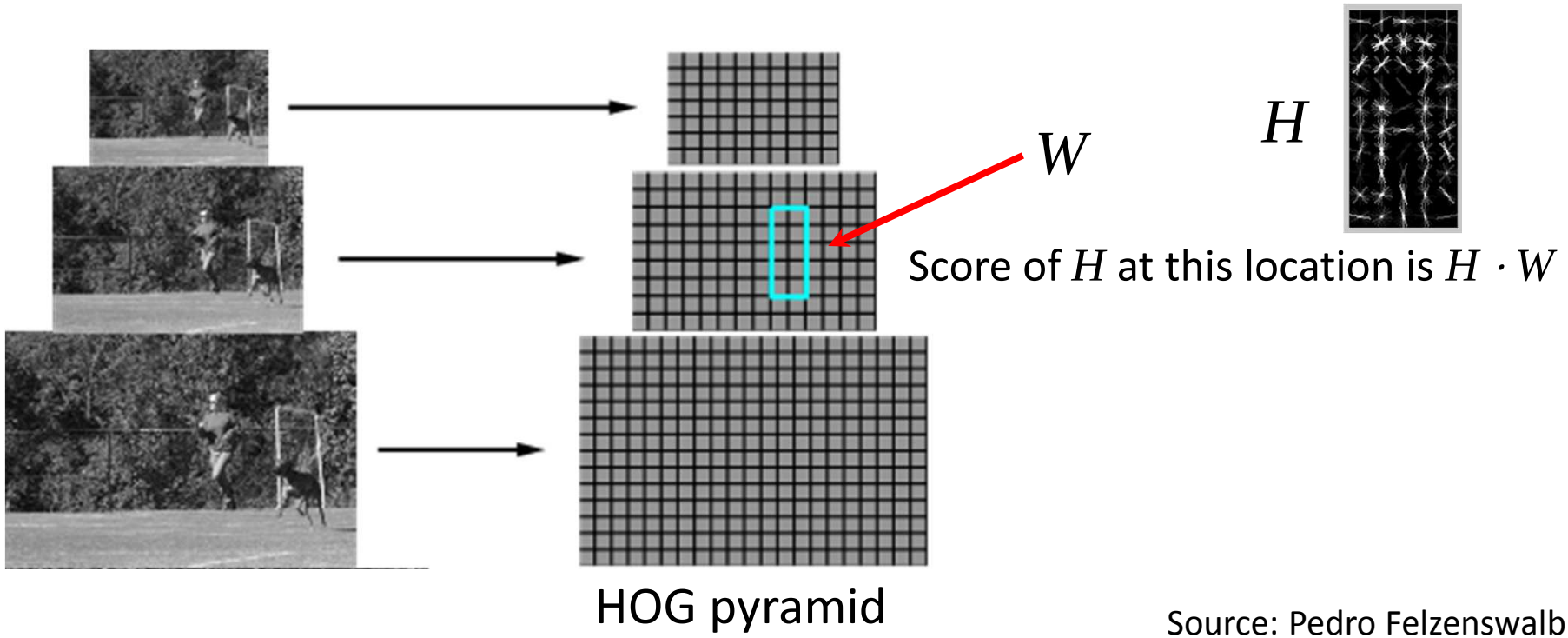


- Image is partitioned into 8x8 pixel blocks.
- In each block we compute a histogram of gradient orientations.
  - **Invariant** to changes in lighting, small deformations, etc.
- We compute features at different resolutions (pyramid).

Source: Pedro Felzenswalb

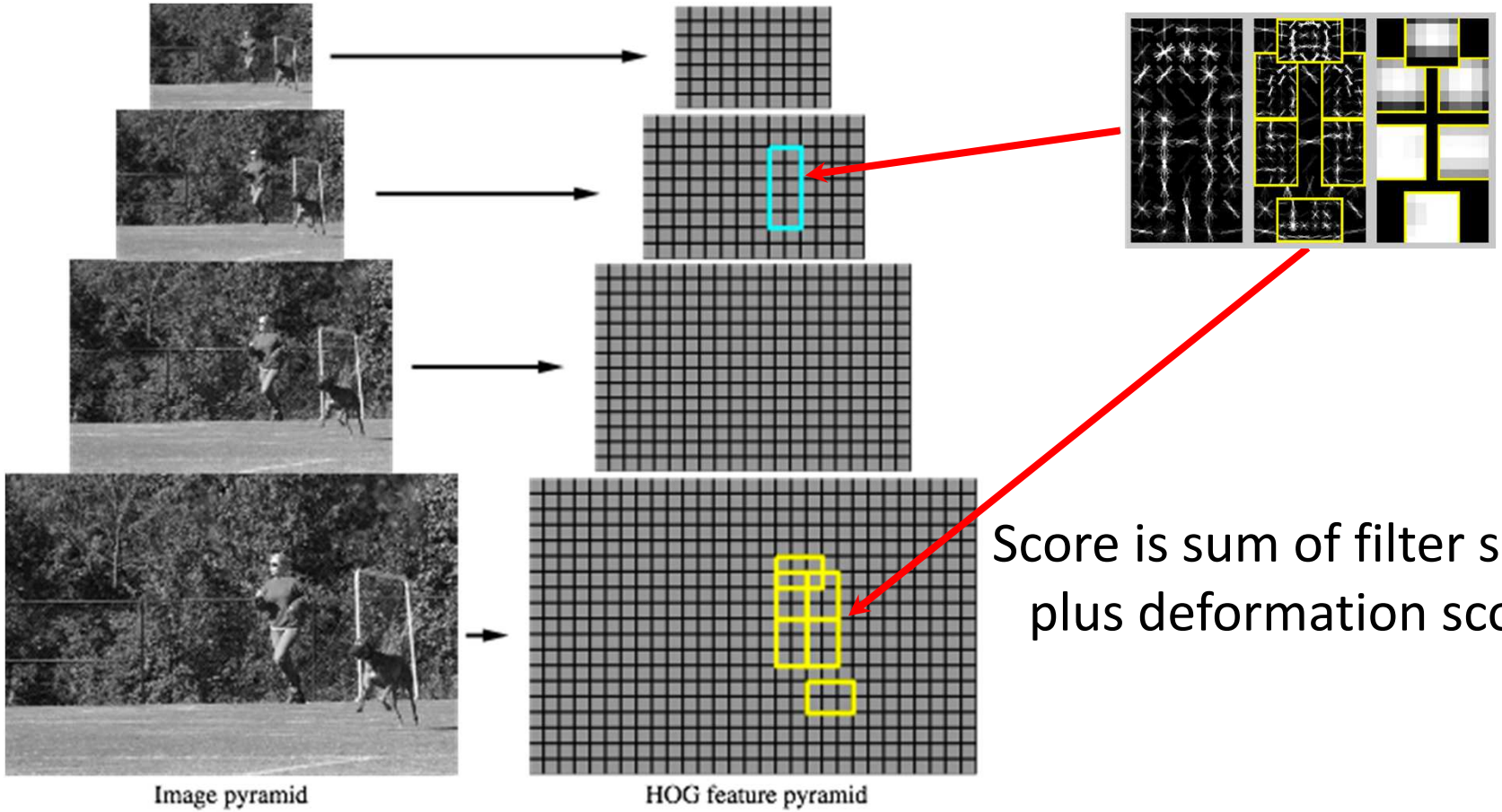
Filters

- Filters are rectangular templates defining weights for features.
- Score is dot product of filter and subwindow of HOG pyramid.



Source: Pedro Felzenswalb

Object Hypothesis



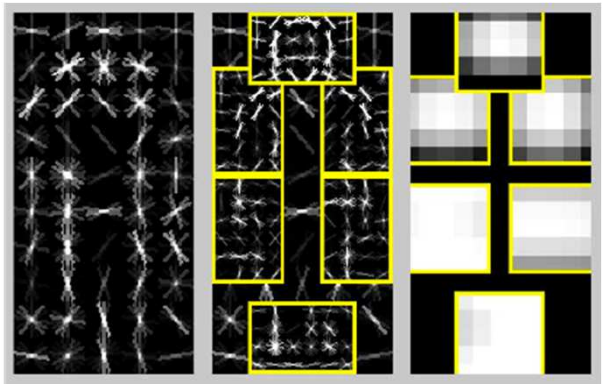
Score is sum of filter scores plus deformation scores

Multiscale model captures features at two-resolutions

Training the Latent SVM Model

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

Training



Source: Pedro Felzenswalb

Connection with Linear Classifiers

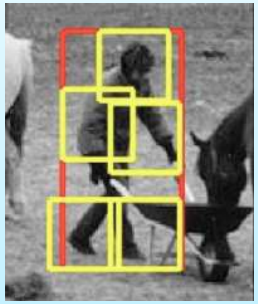
- Score of model is sum of filter scores plus deformation scores
  - Bounding box in training data specifies that the score should be high for some placement in a range

Standard SVM

$$f_w(x) = w \cdot \Phi(x)$$

Weight vector | Features

Latent SVM



w is a model
x is a detection window
z are filter placements

$$f_w(x) = \max_z w \cdot \Phi(x, z)$$

Concatenation of filters and deformation parameters

Concatenation of features and part displacements

Latent SVMs

$$f_w(x) = \max_z w \cdot \Phi(x, z)$$

Linear in *w* if *z* is fixed

Observed variables

Latent variables

$$\text{Training data: } (x_1, y_1), \ldots, (x_n, y_n) \text{ with } y_i \in \{-1, 1\}$$

$$\text{Learning: find } w \text{ such that } y_i f_w(x_i) > 0$$

$$w^* = \operatorname*{argmin}_w \lambda ||w||^2 + \sum_{i=1}^{n} \max(0, 1 - y_i f_w(x_i))$$

Regularization

Hinge Loss

Latent SVM Training

$$w^* = \underset{w}{\operatorname{argmin}}\, \lambda||w||^2 + \sum_{i=1}^{n} \max(0, 1 - y_i f_w(x_i))$$

- Semi-convex optimization problem
  - Maximum of convex functions is convex
  - $f_w(x) = \max_z w \cdot \Phi(x, z)$ is convex in *w*

$$\max(0, 1 - y_i f_w(x_i)) = \begin{cases} \max(0, 1 + f_w(x_i)) & \text{if } y_i = -1 \\ \max(0, 1 - f_w(x_i)) & \text{if } y_i = 1 \end{cases}$$

Convex!

Not convex

Latent SVM Training

$$w^* = \underset{w}{\operatorname{argmin}}\, \lambda||w||^2 + \sum_{i=1}^{n} \max(0, 1 - y_i f_w(x_i))$$

- Convex if we fix z for positive examples:

Affine!

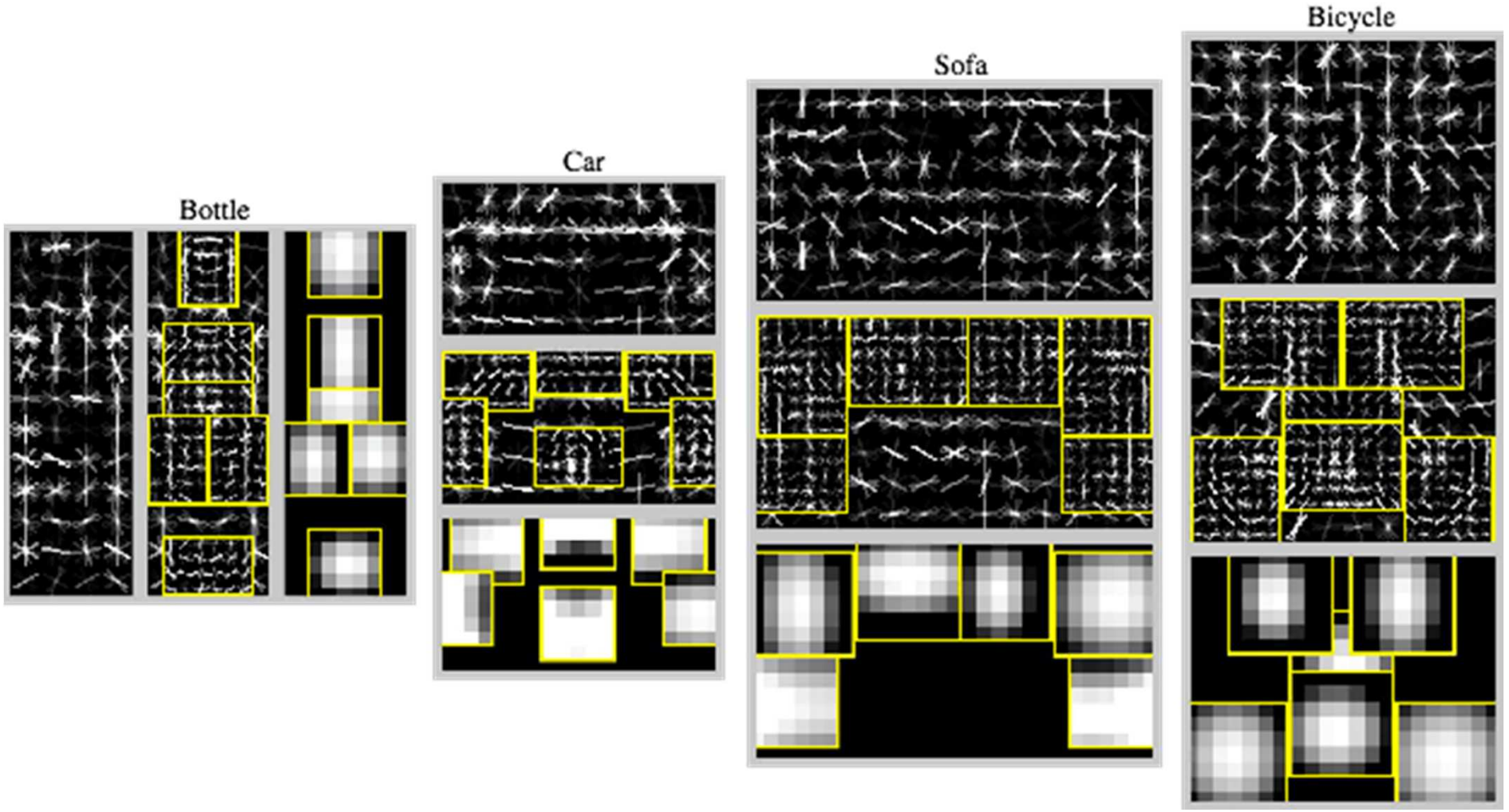
$$f_w(x) = w \cdot \Phi(x) \Rightarrow \max(0, 1 - w \cdot \Phi(x)) \text{ if } y_i = 1$$

- Iterative optimization procedure:
  - Initialize w and iterate:
    - Pick best z for each positive example
    - Optimize w via gradient descent with data mining

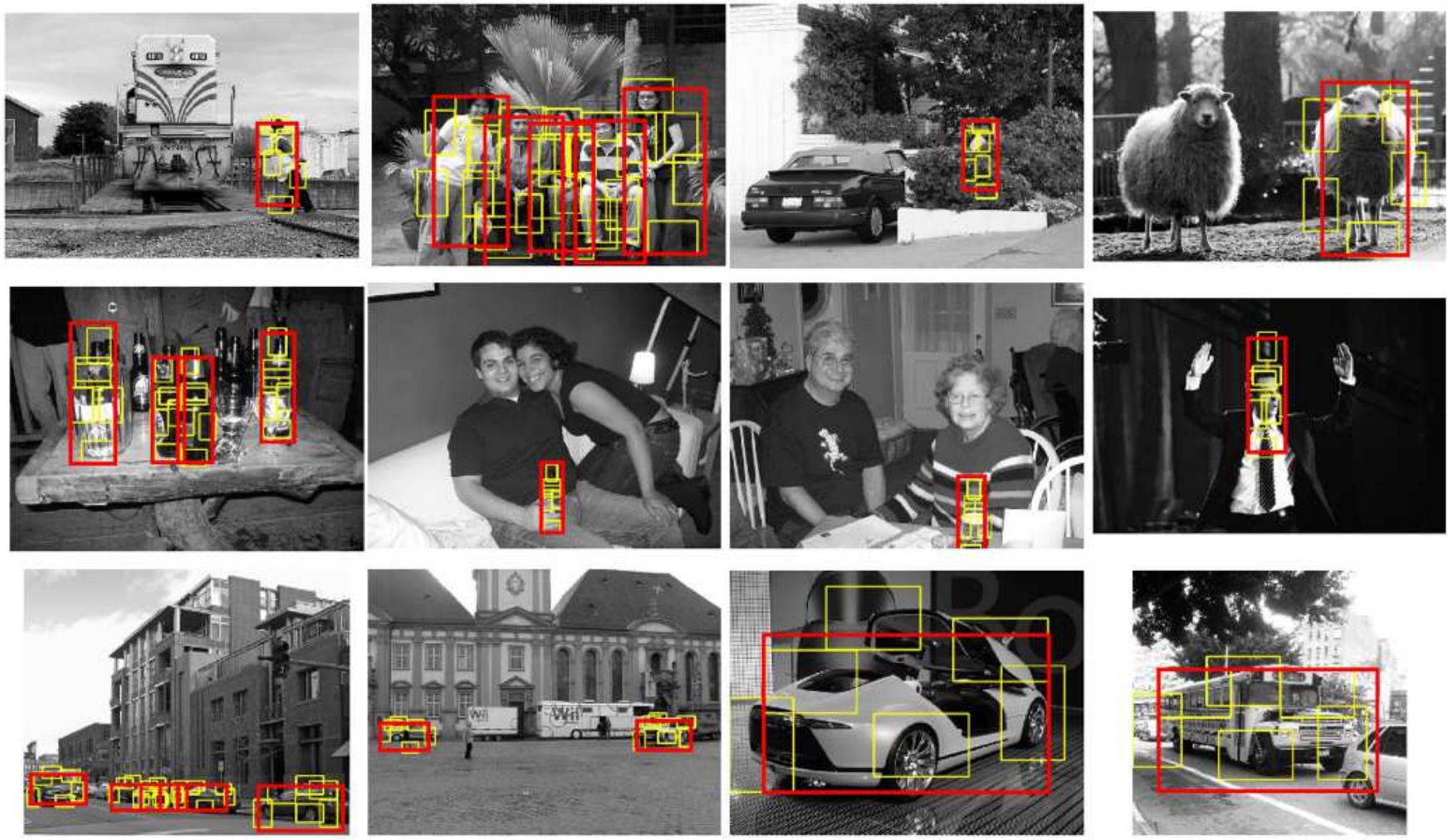
Latent SVM Training: Initializing w

- For k component mixture model:
  - Split examples into k sets based on bounding box aspect ratio
- Learn k root filters using standard SVM
  - Training data: Warped positive examples and random windows from negative images (Dalal & Triggs)
- Initialize parts by selecting patches from root filters:
  - Sub-windows with strong coefficients
  - Interpolate to get higher resolution filters
  - Initialize spatial model using fixed spring constants

Learned Models

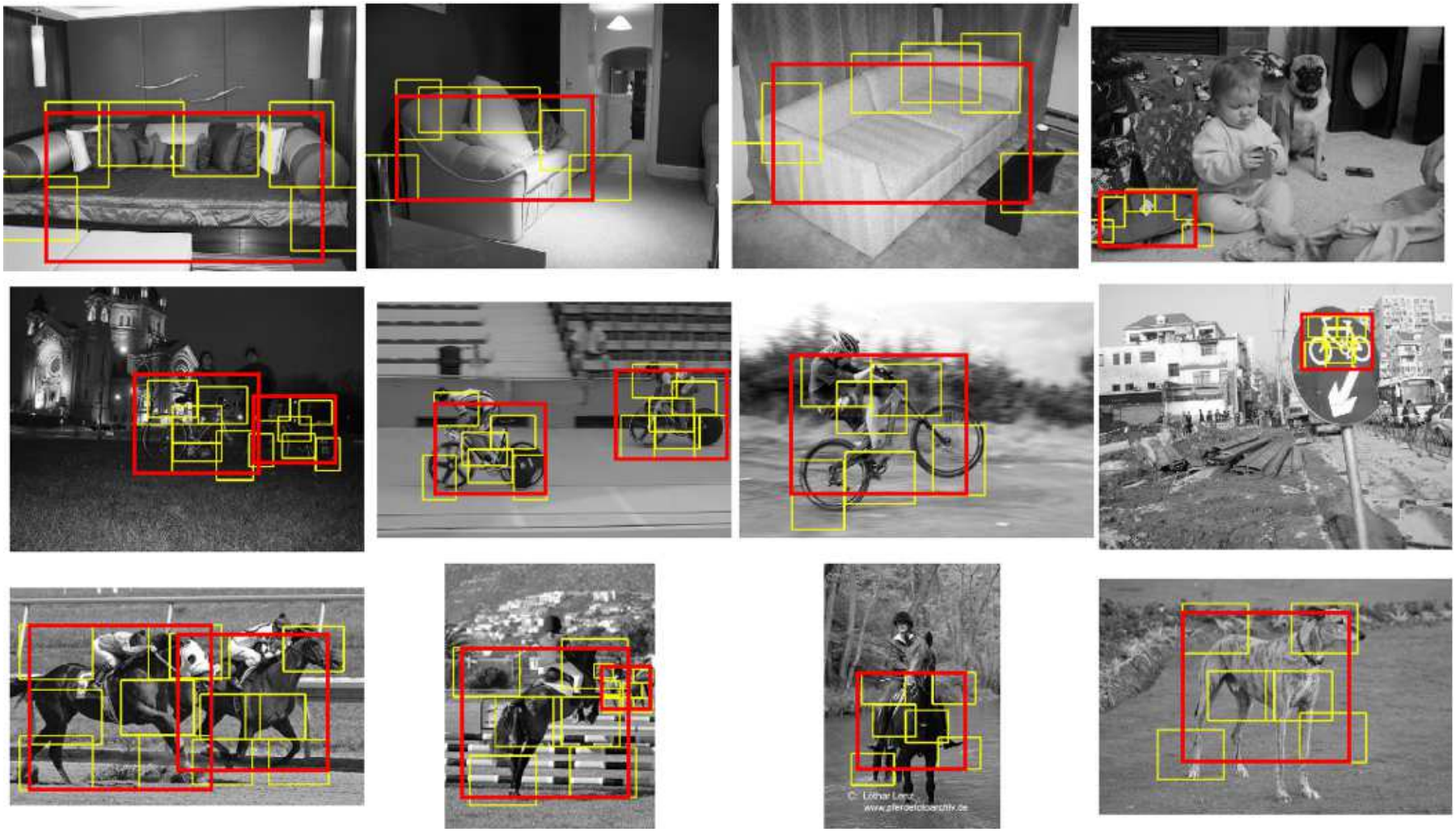


Example Results



Source: Pedro Felzenswalb

More Results



Quantitative Results

- 9 systems competed in the 2007 challenge.
- Out of 20 classes:
  - First place in 10 classes
  - Second place in 6 classes
- Some statistics:
  - It takes ~2 seconds to evaluate a model in one image.
  - It takes ~3 hours to train a model.
  - MUCH faster than most systems.

Source: Pedro Felzenswalb

Code for Latent SVM

Source code for the system and models trained on PASCAL 2006, 2007 and 2008 data are available at:

http://www.cs.uchicago.edu/~pff/latent

Source: Pedro Felzenswalb

Summary

- Deformable models provide an elegant framework for object detection and recognition.
  - Efficient algorithms for matching models to images.
  - Applications: pose estimation, medical image analysis, object recognition, etc.
- We can learn models from partially labeled data.
  - Generalized standard ideas from machine learning.
  - Leads to state-of-the-art results in PASCAL challenge.
- Future work: hierarchical models, grammars, 3D objects.

Source: Pedro Felzenswalb

What we have learned today

- Implicit Shape Model
  - Representation
  - Recognition
  - Experiments and results
- Deformable Models
  - The PASCAL challenge
  - Latent SVM Model