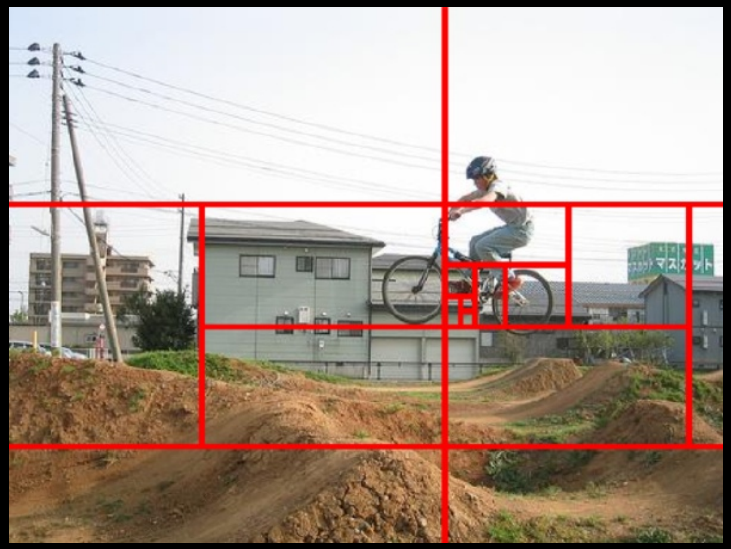
Machine Learning for Computer Vision

21 October, 2013
MVA – ENS Cachan



Rapid Object Detection with Deformable Part Models

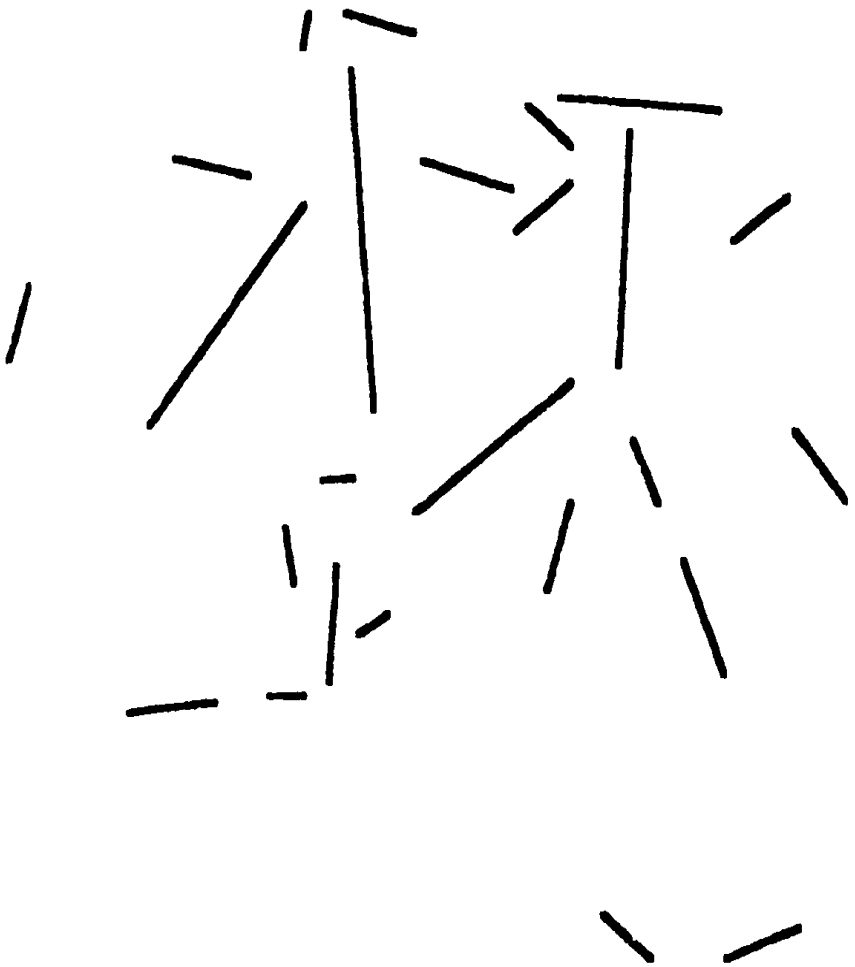
Iasonas Kokkinos

Iasonas.kokkinos@ecp.fr

Center for Visual Computing
Ecole Centrale Paris

Galen Group
INRIA-Saclay

Perceptual Organization

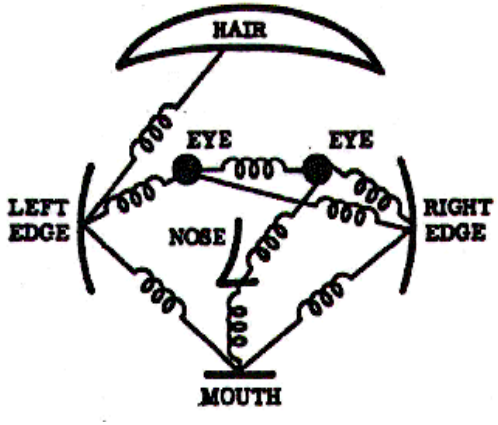


D. Lowe, Perceptual Organization and Visual Recognition, 1985

Perceptual Organization

D. Lowe, Perceptual Organization and Visual Recognition, 1985

Object Detection with Deformable Part Models

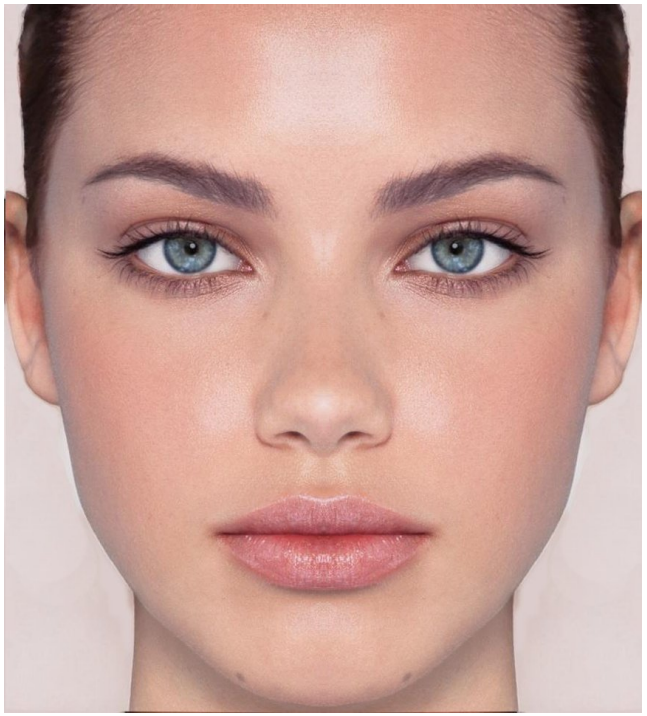





P.F. Felzenszwalb, and D. Huttenlocher, Pictorial Structures for Object Recognition, IJCV 2005
P.F. Felzenszwalb, and D. Huttenlocher, Distance Transforms of Sampled Functions, 2004
P.F. Felzenszwalb, R. B. Girshick, and D. A. McAllester, Cascade object detection with DPMs CVPR 2010

Deformable Part Models (DPMs)

$$\mathbf{S}(\mathbf{x}) = \sum_{p=1}^{P} U_p(x_p) + B_p(x, x_p)$$



Local appearance

$$U_p(x_p) = \langle w_p, H(x_p) \rangle$$

Pairwise compatibility

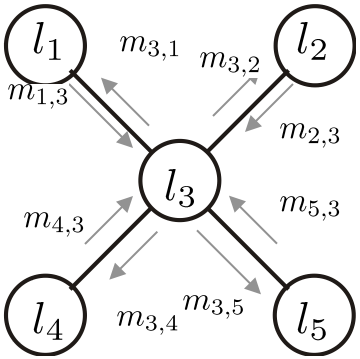
$$B_p(x, x_p) = -(h - h_p - \hat{h}_p)^2 \eta - (v - v_p - \hat{v}_p)^2 \nu$$

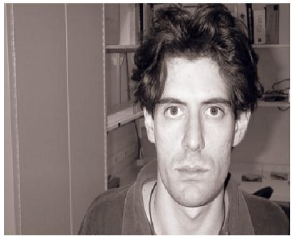
Object score

$$S(x) = \sum_{p=1}^{P} \max_{x'} \left[U_p(x') + B_p(x, x') \right]$$

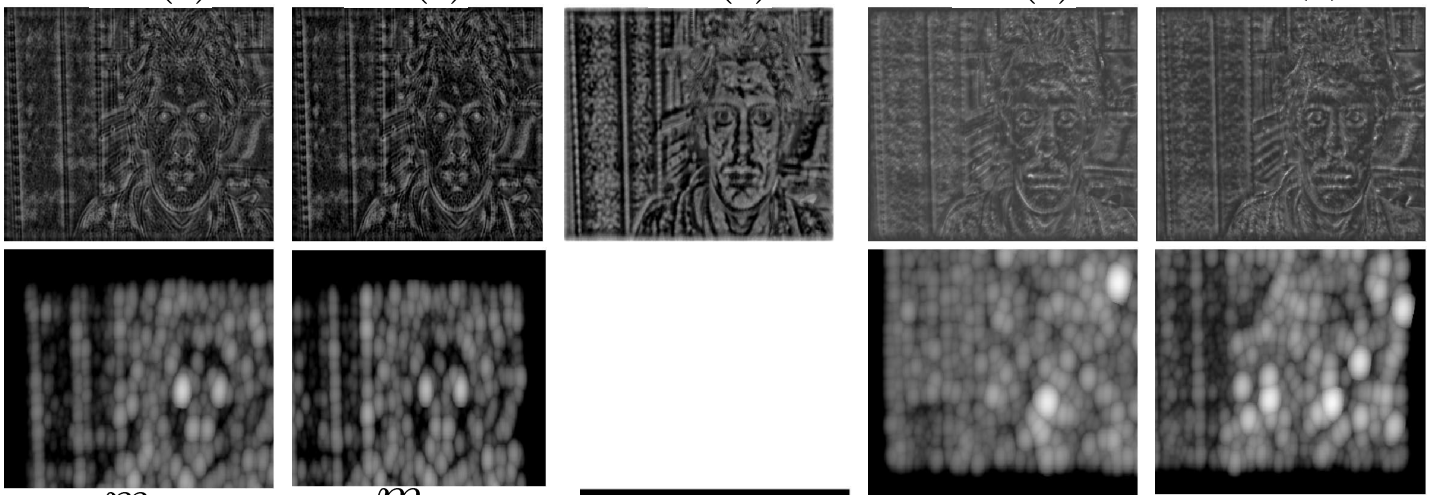
$$\mu_{i\to j}(X_j) = \max_{X_i} \Phi_i(X_i)\Psi_{i,j}(X_i, X_j) \prod_{k\in\mathcal{N}(i)\setminus j} \mu_{k\to j}(X_i)$$

$$B_j(X_j) = \max P(X_j)$$

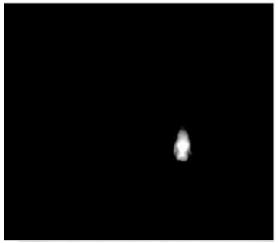




$\Phi_1(l)$ $\Phi_2(l)$ $\Phi_3(l)$ $\Phi_4(l)$ $\Phi_5(l)$

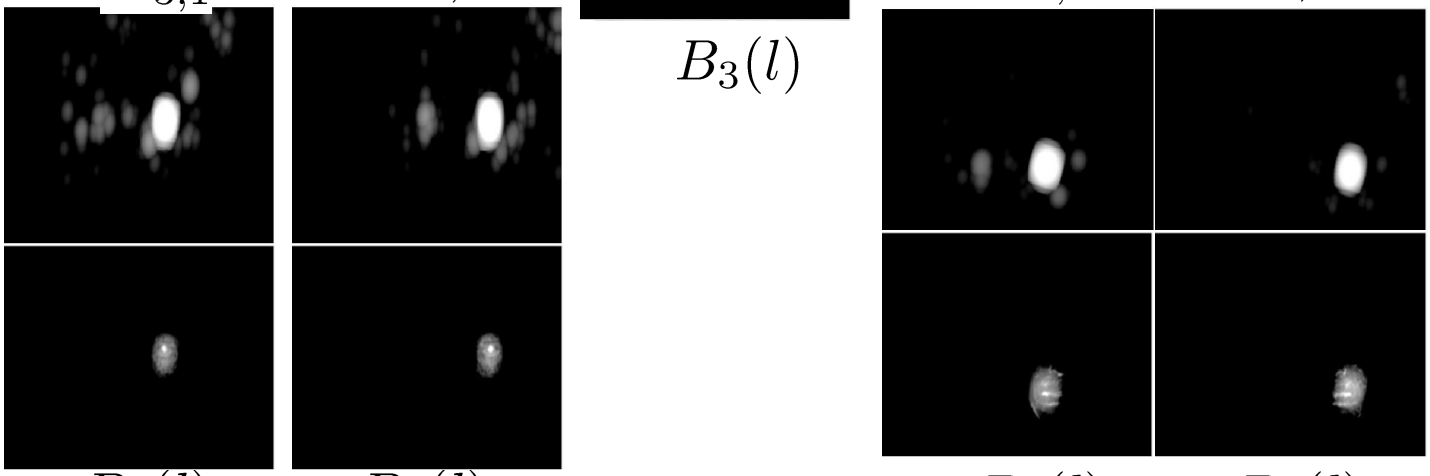


$m_{1,3}$ $m_{2,3}$ $m_{4,3}$ $m_{5,3}$



$B_3(l)$

$m_{3,1}$ $m_{3,2}$ $m_{3,4}$ $m_{3,5}$



$B_1(l)$ $B_2(l)$ $B_4(l)$ $B_5(l)$

Object detection with Deformable Part Models (DPMs)

$$U_p(x') = \langle \mathbf{w}_p, \mathbf{H}(x') \rangle$$

$$\max_{x'} [U_p(x') + B_p(x, x')]$$

PnA 2012

$$p = 1$$

$$\vdots$$

$$p = P$$

DTBB, NIPS 11

$$S(x) = \sum_{p=1}^{P} \max_{x'} [U_p(x') + B_p(x, x')]$$

Object Detection problem

$$S(x) = \sum_{p=1}^{P} \max_{x'} [U_p(x') + B_p(x, x')]$$

Threshold-based detection: recover all locations above threshold

$$M^{\theta} = \{x : S(x) \geq \theta\}$$

1-best detection: recover top-scoring candidate

$$M^{*} = \{\arg\max_x S(x)\}$$

Naïve implementation (Max-Product) : $O(P|x|^2)$

Generalized Distance Transforms (GDT): $O(P|x|)$

Our work (best-case): $O(|M|P \log_2 |x|)$

Key idea: avoid the exact evaluation of S(x)

I. Kokkinos, Rapid DPM detection using Dual Tree Branch-and-Bound, NIPS, 2011

Previous works on acceleration

Sparse image representations:

Y. Chen, L. Zhu, A. Yuille. Rapid inference on a novel and/or graph for object detection, NIPS 2007

I. Kokkinos and A. Yuille. HOP: Hierarchical Object Parsing, CVPR, 2009

Dense image representations:

B. Sapp, A. Toshev, and B. Taskar. Cascaded models for articulated pose estimation, ECCV, 2010

M.Pedersoli, A.Vedaldi, and J.Gonzalez. A coarse-to-fine approach for object detection, CVPR 2011

P F Felzenszwalb R B Girshick and D A McAllester Cascade object detection with DPMs CVPR 2010

`Monolithic' object models:

F. Fleuret and D. Geman. Coarse-to-fine face detection, IJCV 2001

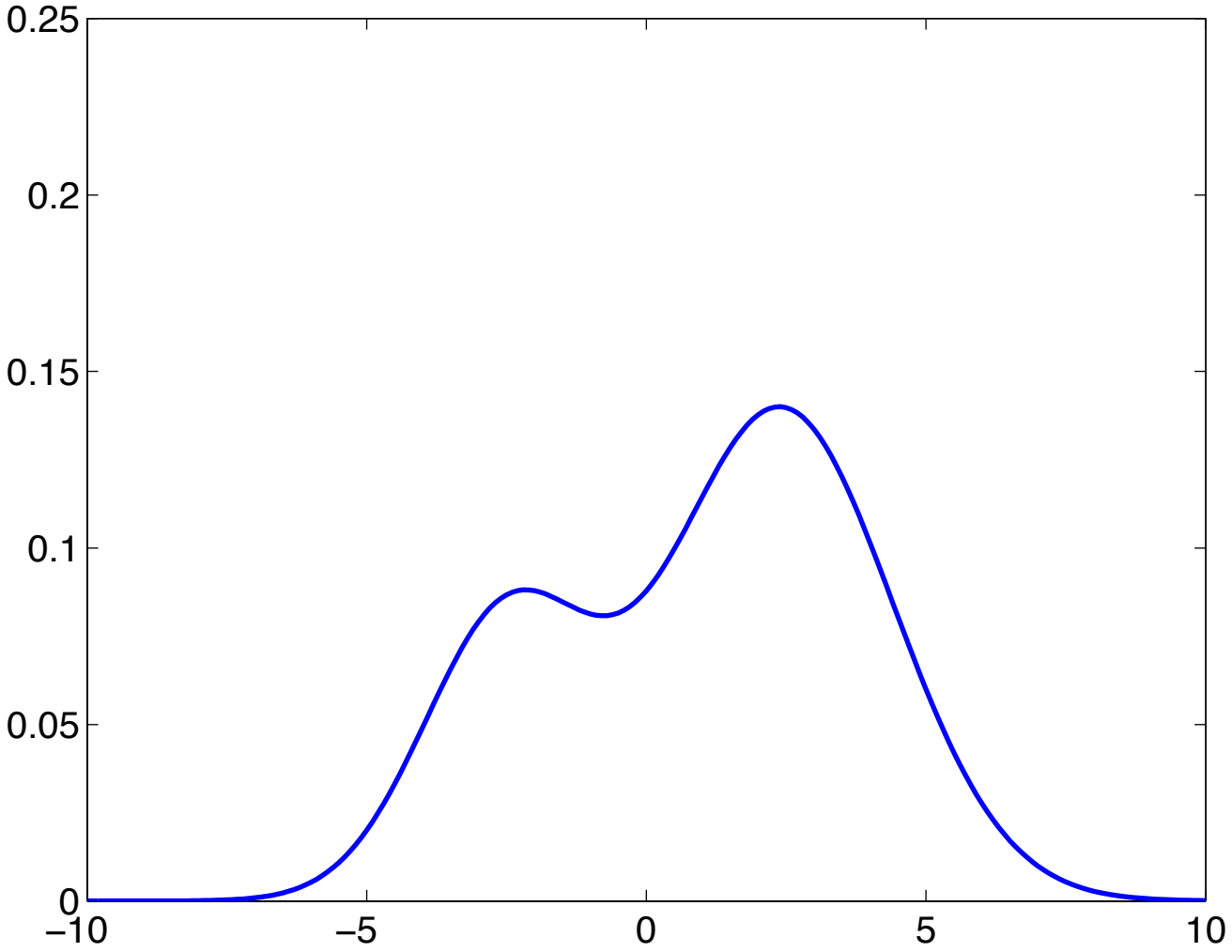
P. Viola and M. Jones. Robust Real-time Object Detection, IJCV 2004

C. Lampert , M. Blaschko and T. Hoffman. Beyond Sliding Windows- ESS, CVPR 2008

T. Breuel. Fast recognition using adaptive subdivisions of transformation space, CVPR 1992

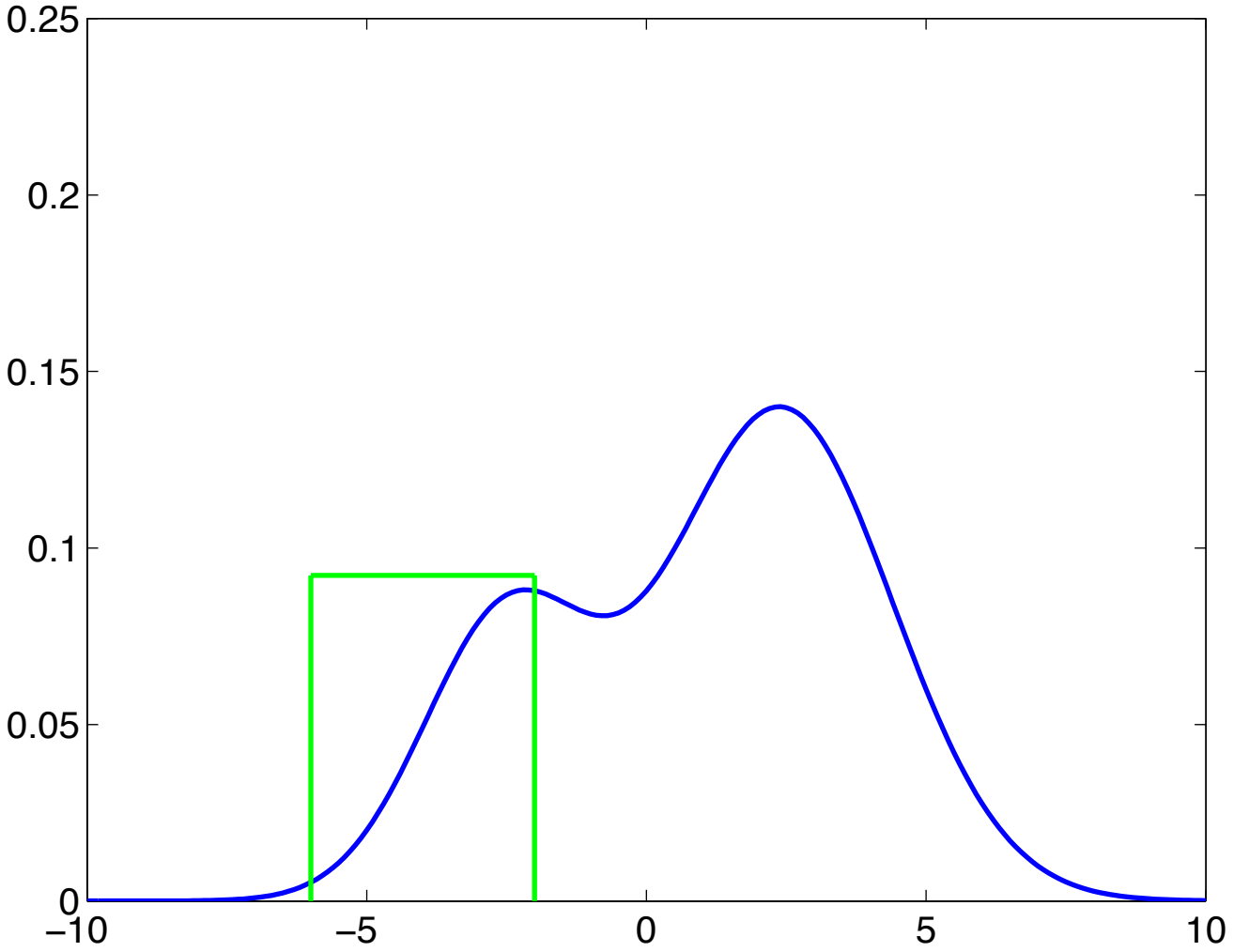
Branch and bound

$$\max_{x \in X} f(x)$$



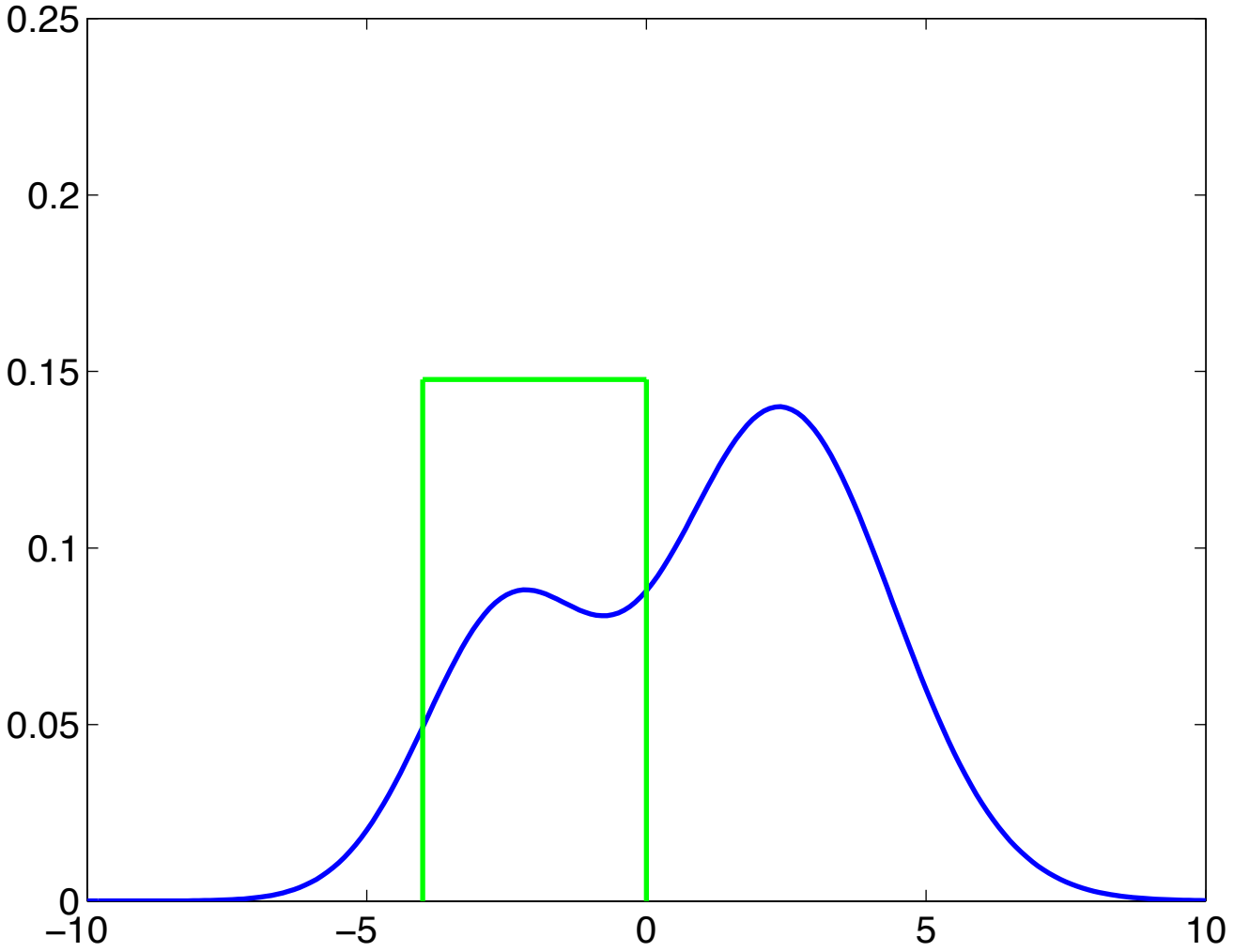
Bounding function

$$\overline{f}(X) \geq \max_{x \in X} f(x), \quad \overline{f}(\{x\}) = f(x)$$



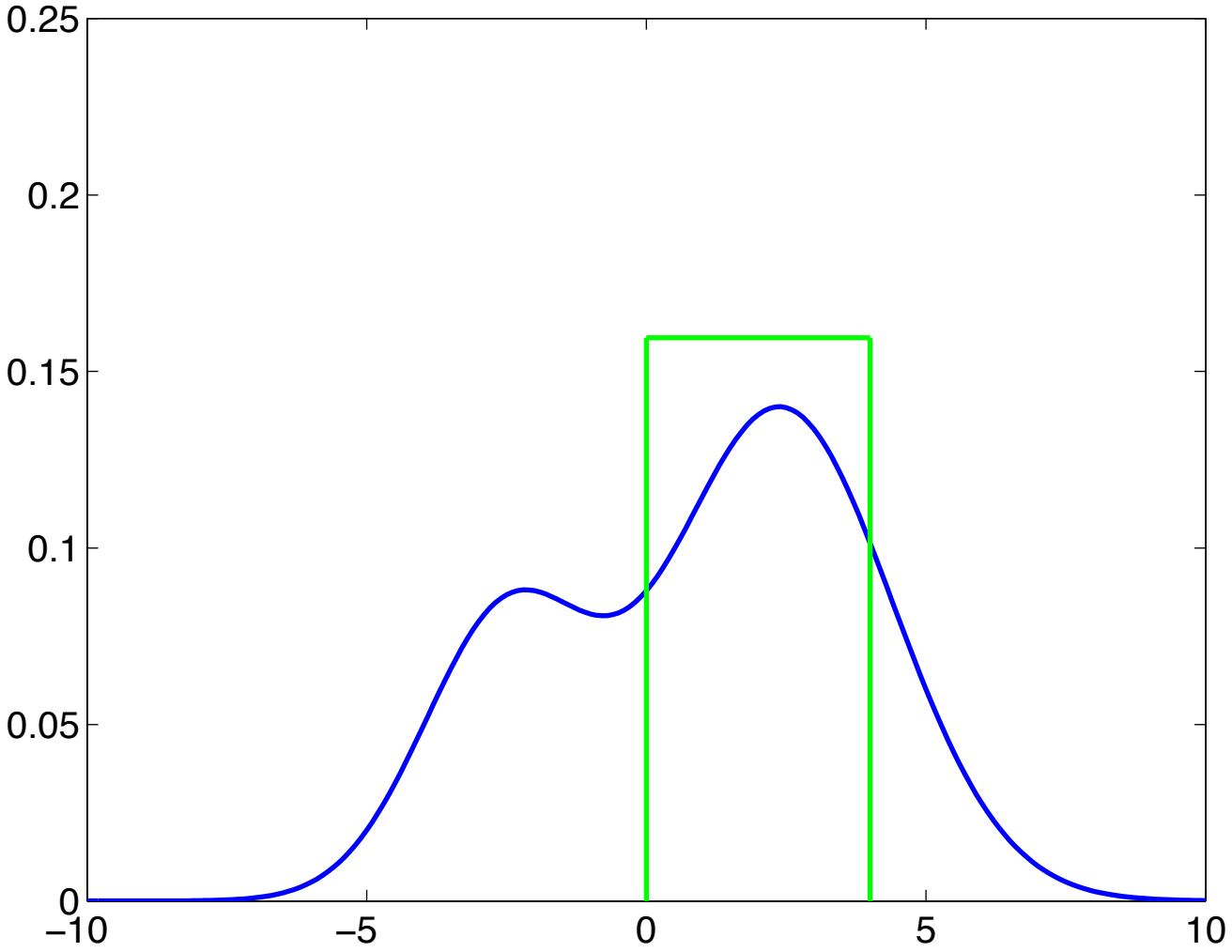
Bounding function

$$\overline{f}(X) \geq \max_{x \in X} f(x), \quad \overline{f}(\{x\}) = f(x)$$



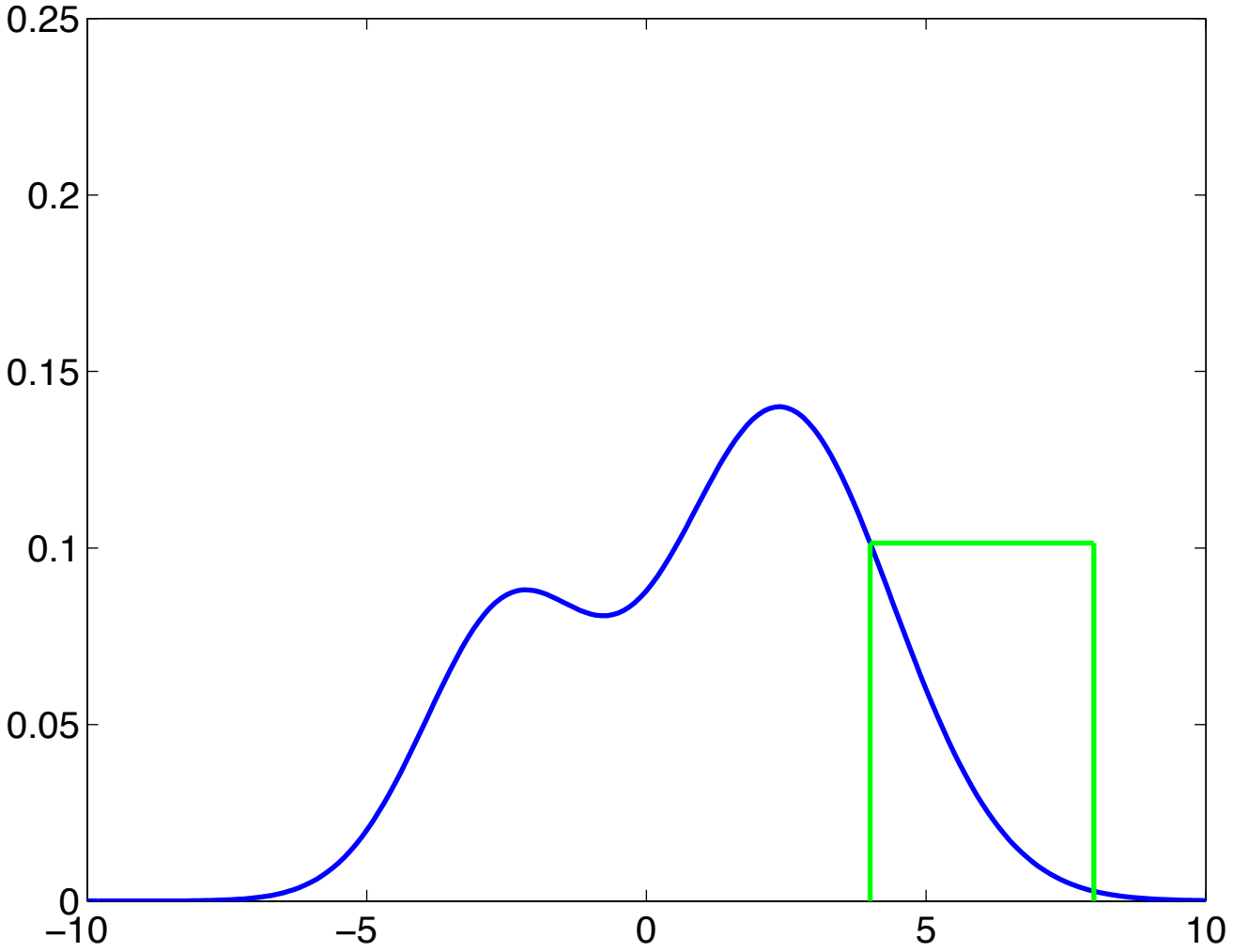
Bounding function

$$\overline{f}(X) \geq \max_{x \in X} f(x), \quad \overline{f}(\{x\}) = f(x)$$

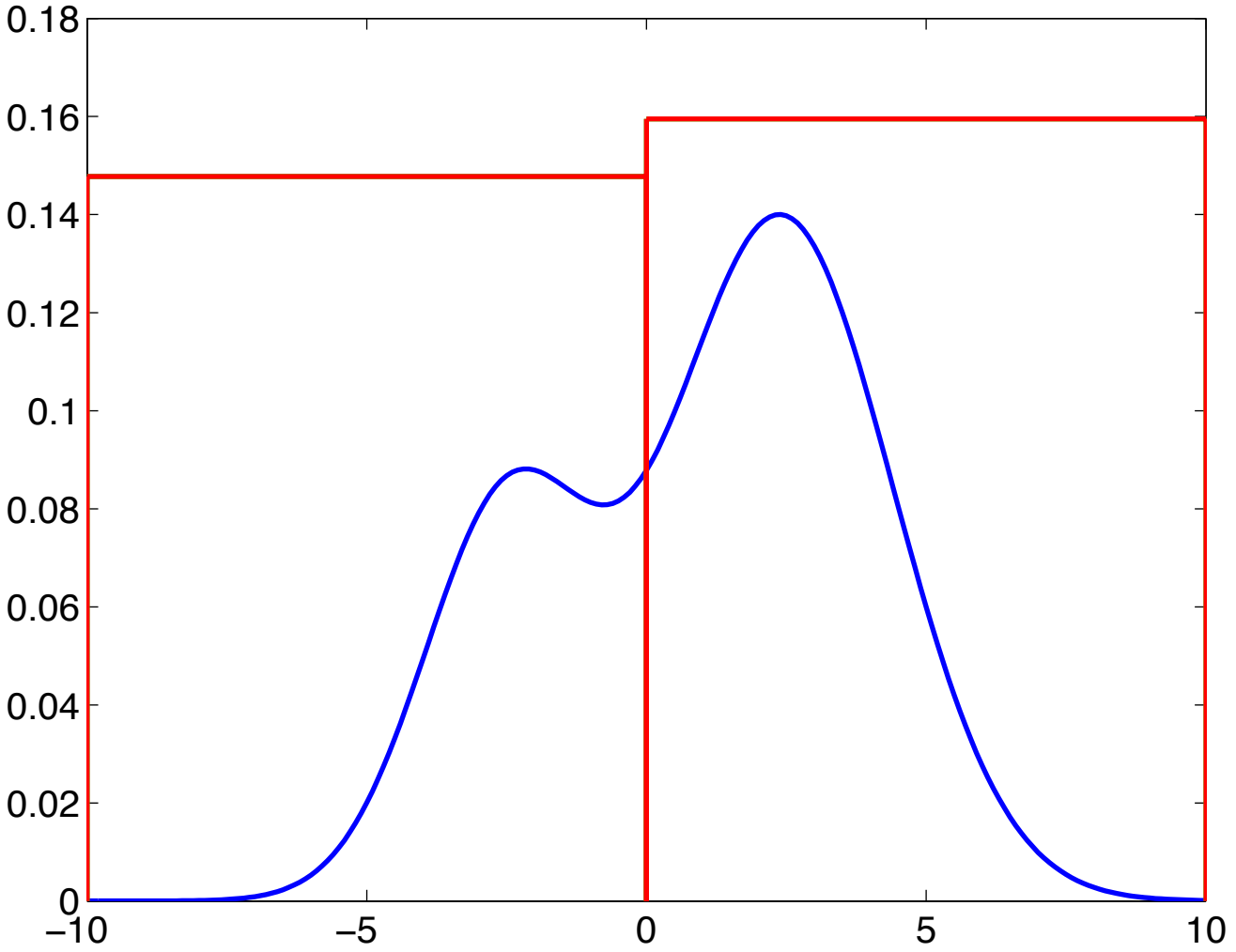


Bounding function

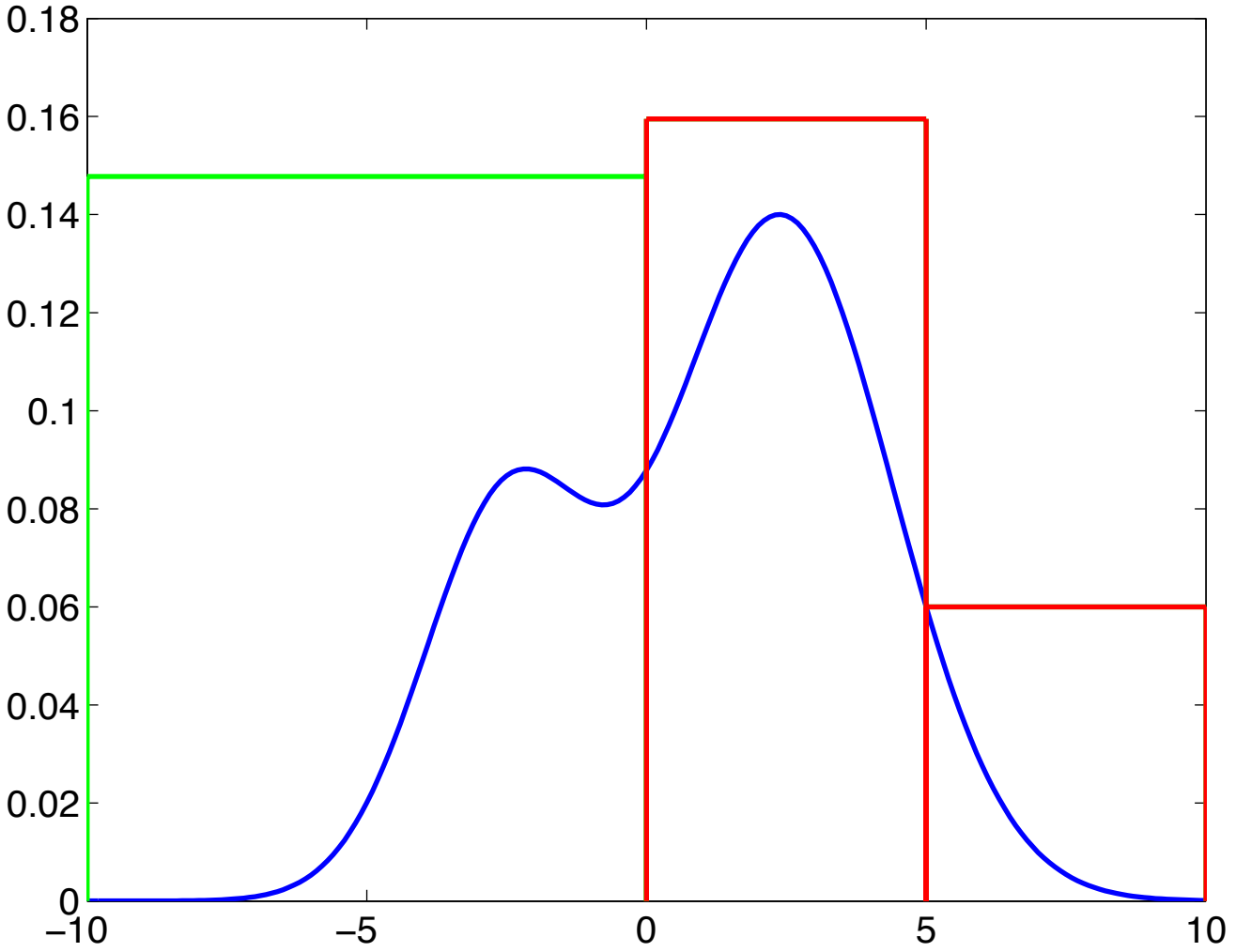
$$\overline{f}(X) \geq \max_{x \in X} f(x), \quad \overline{f}(\{x\}) = f(x)$$



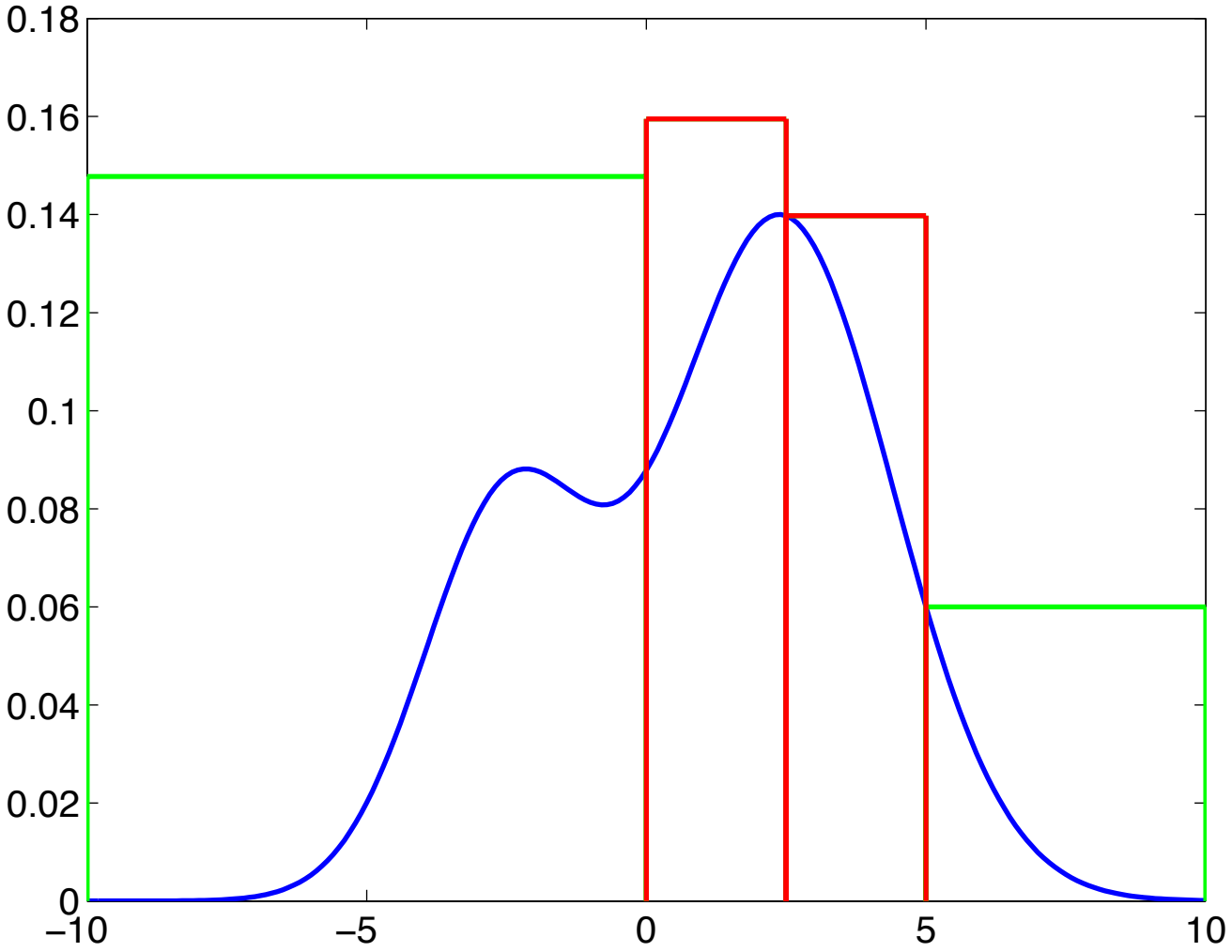
Branch-and-bound: all you need is bounds



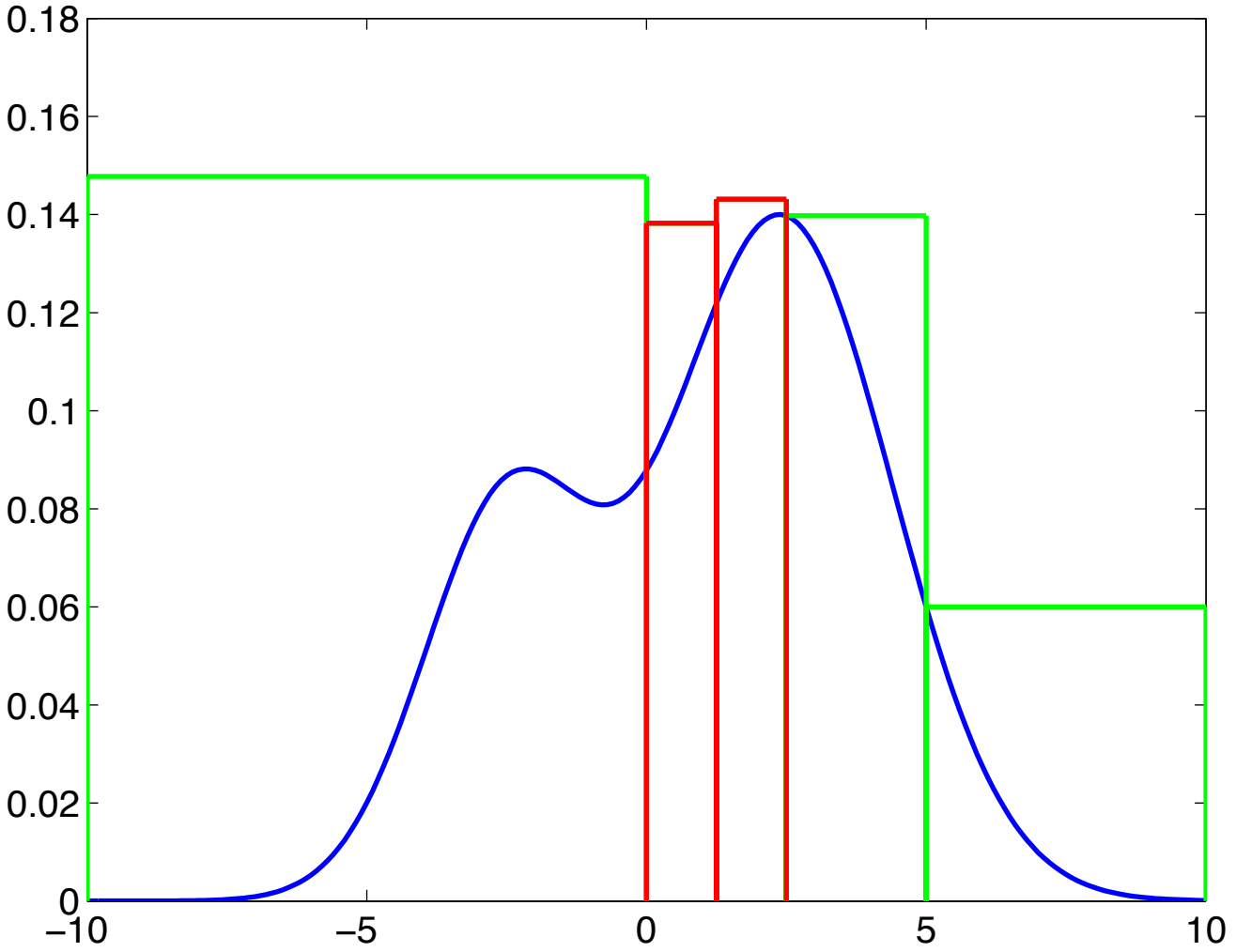
Branch-and-bound: all you need is bounds



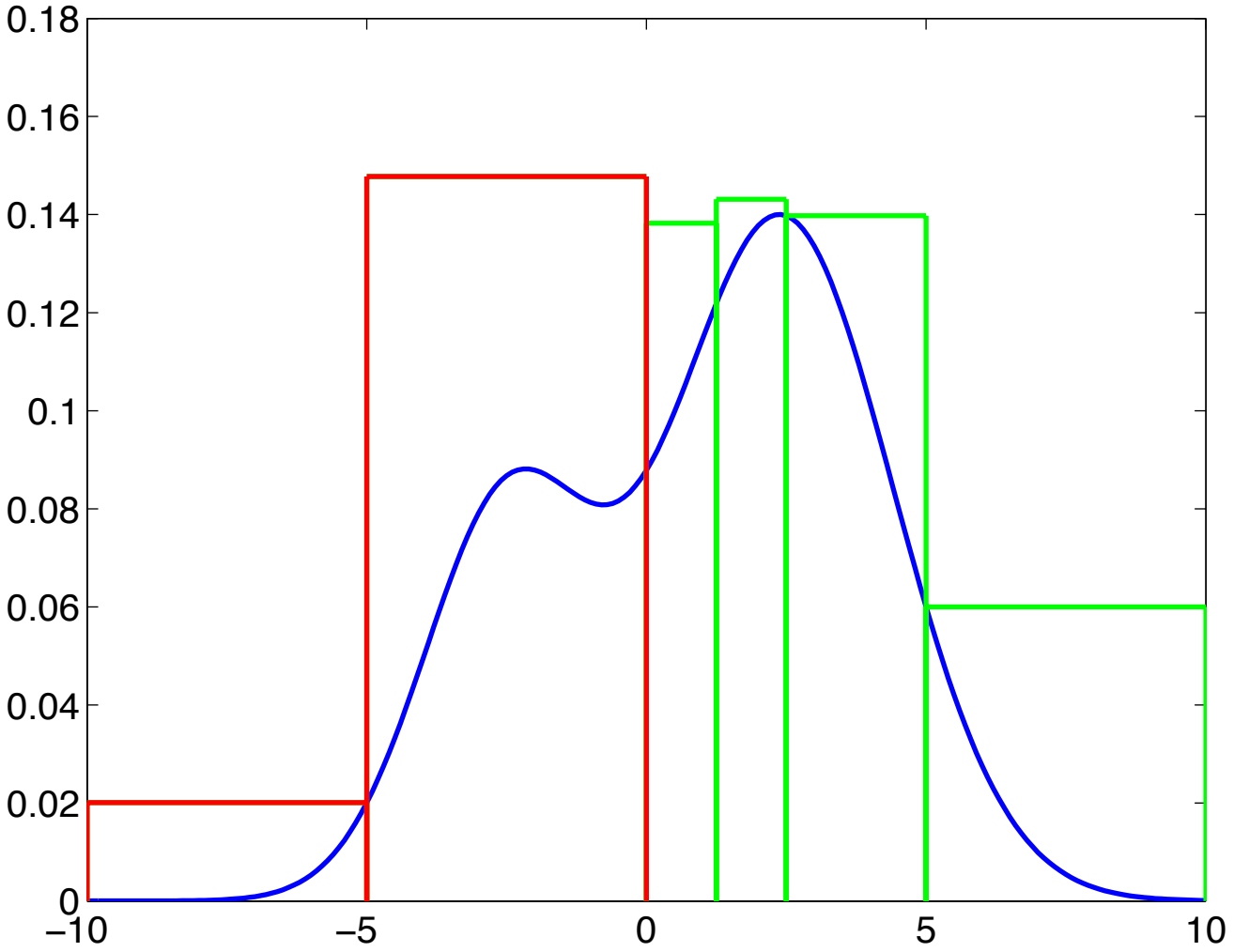
Branch-and-bound: all you need is bounds



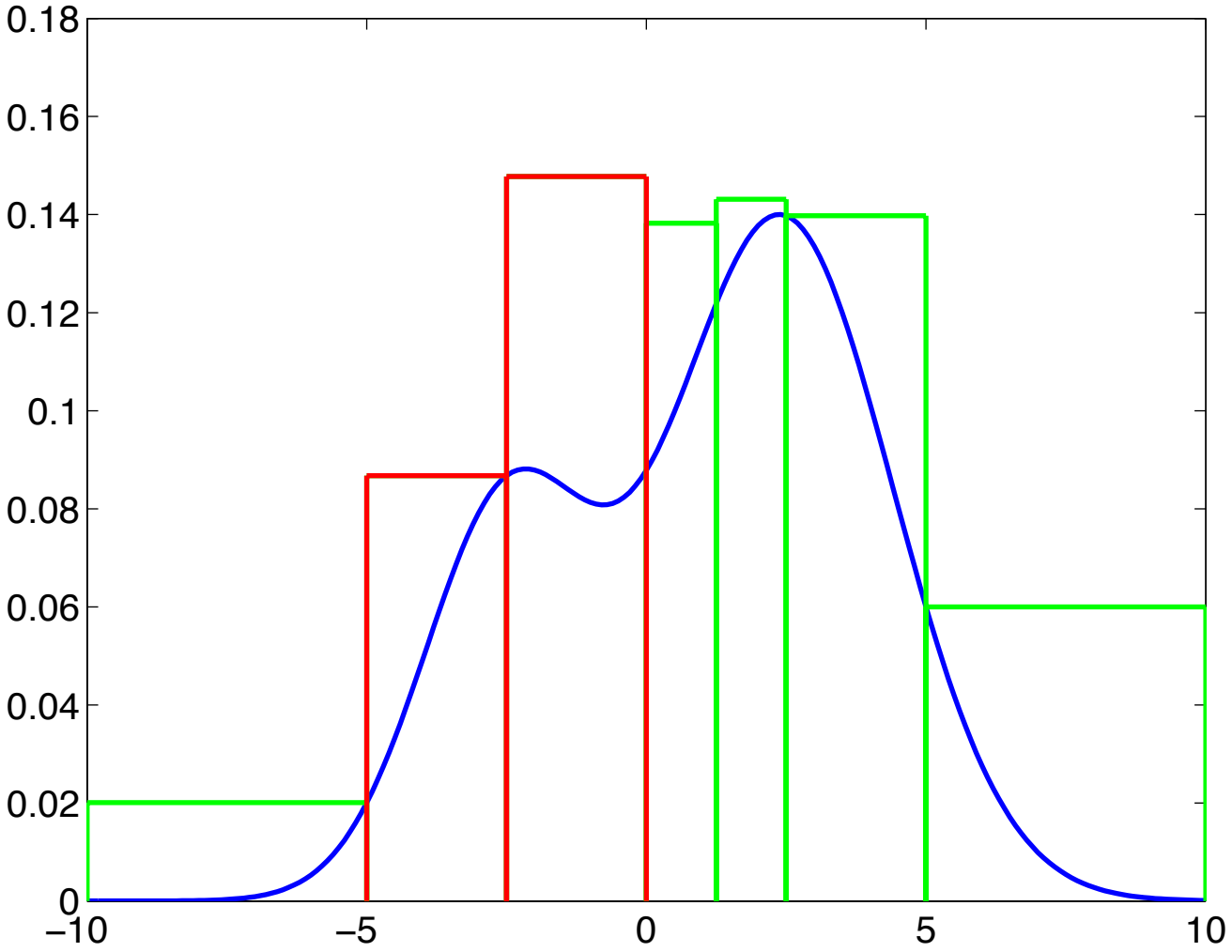
Branch-and-bound: all you need is bounds



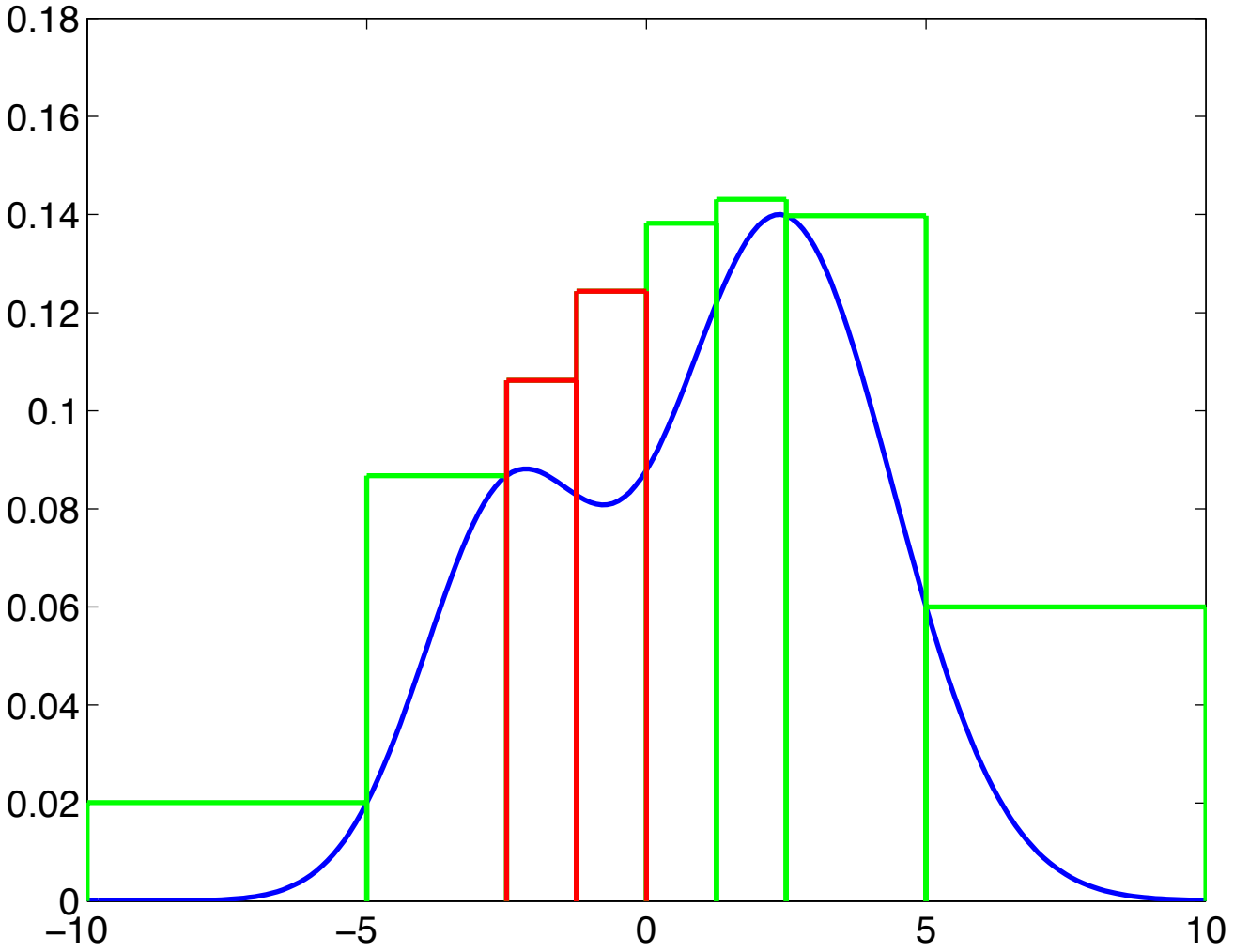
Branch-and-bound: all you need is bounds



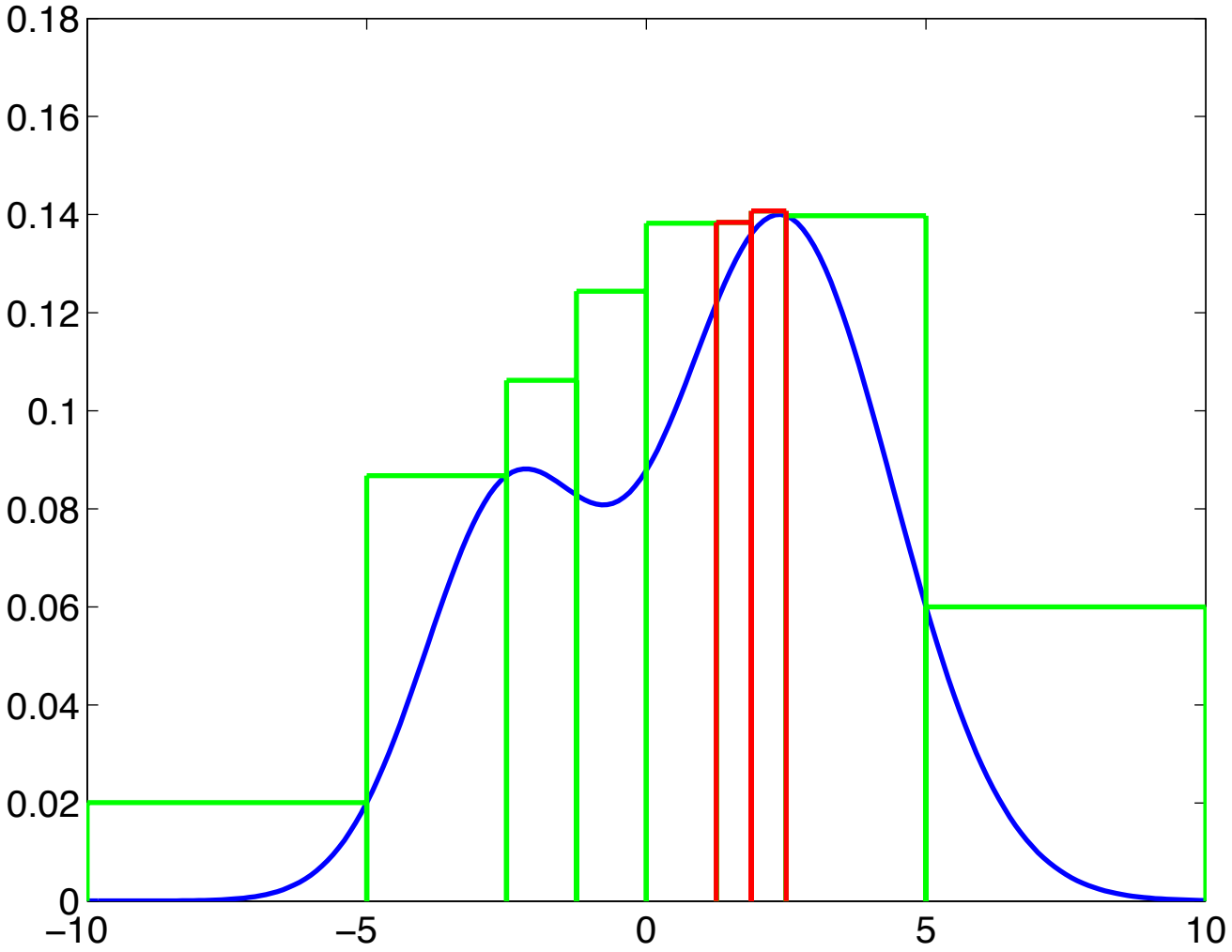
Branch-and-bound: all you need is bounds



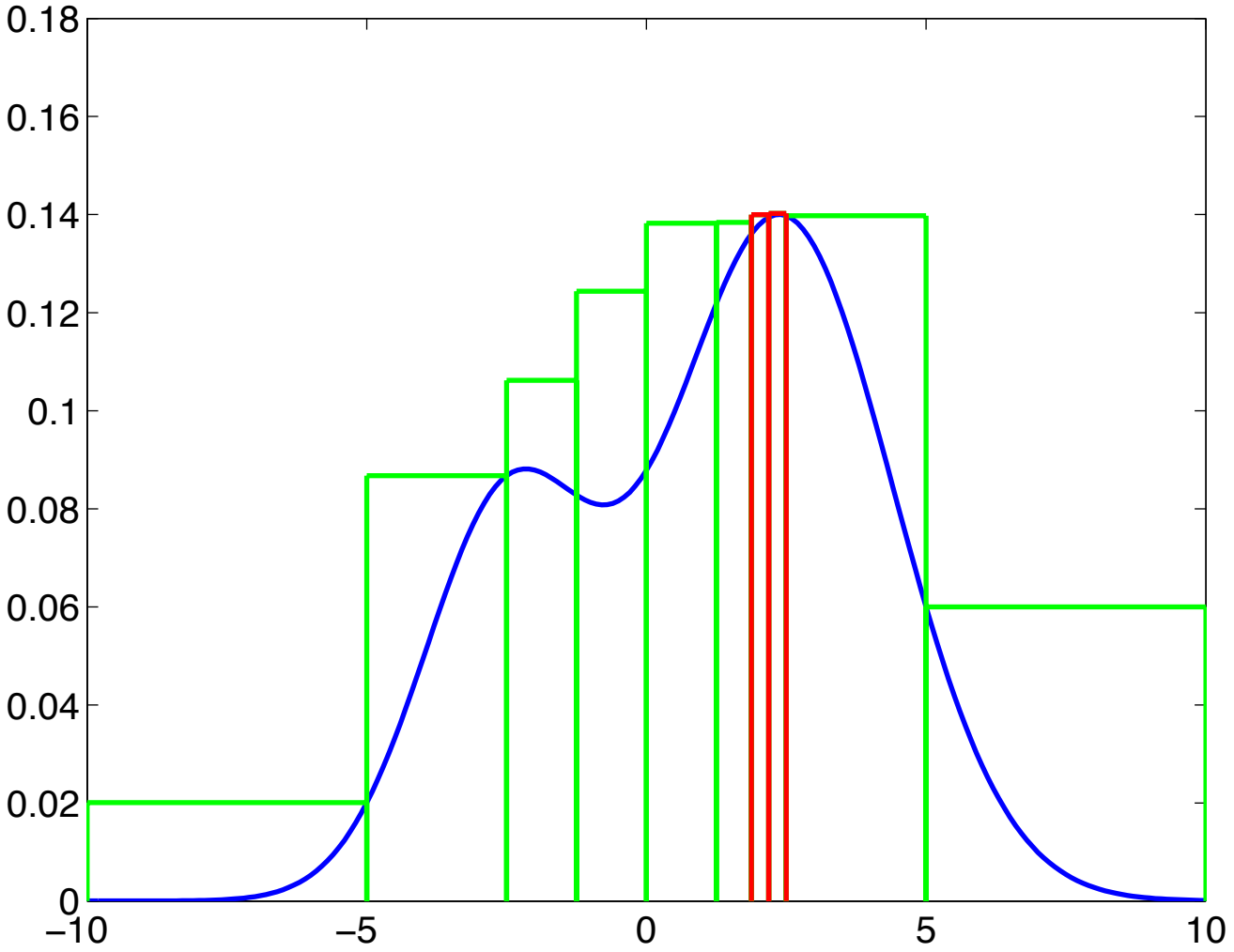
Branch-and-bound: all you need is bounds



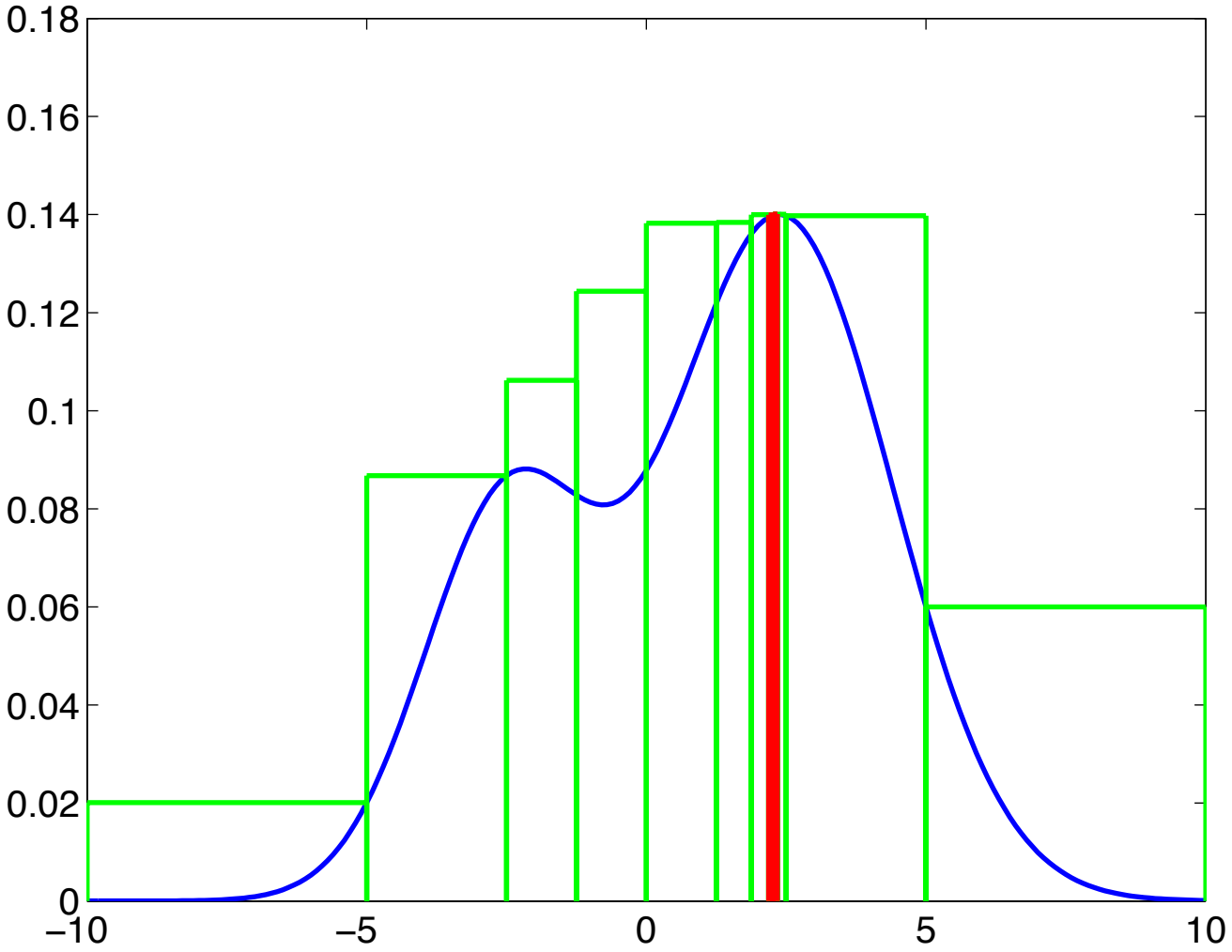
Branch-and-bound: all you need is bounds



Branch-and-bound: all you need is bounds



Branch-and-bound: all you need is bounds



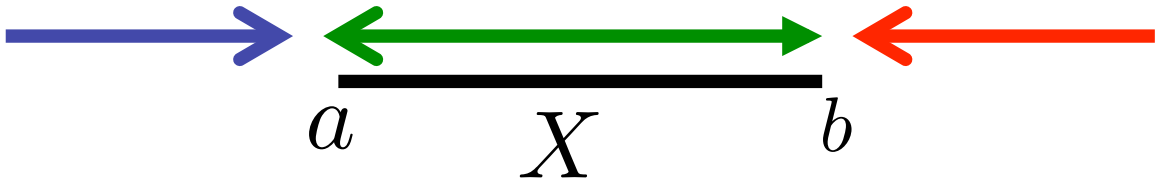
Bounding a mixture-of-gaussians

- Property: $\max_{x \in X} h(x) + g(x) \leq \max_{x \in X} h(x) + \max_{x \in X} g(x)$
- Function: $f(x) = \pi_1 N(x; \mu_1, \sigma_1) + \pi_2 N(x; \mu_2, \sigma_2)$

$$\max_{x \in X} f(x) \leq \max_{x \in X} [\pi_1 N(x; \mu_1, \sigma_1)] + \max_{x \in X} [\pi_2 N(x; \mu_2, \sigma_2)]$$

$$= \pi_1 N(d(X, \mu_1, \sigma_1); 0, 1) + \pi_2 N(d(X, \mu_2, \sigma_2); 0, 1)$$

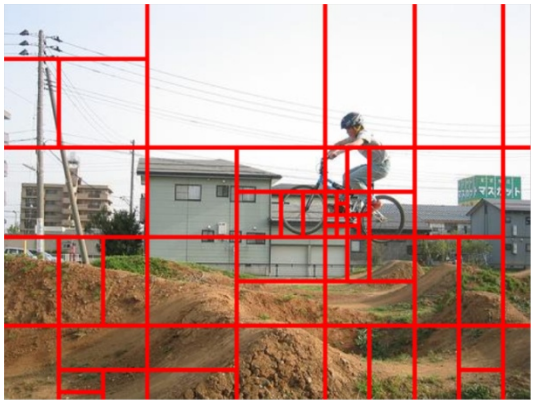
$$d(X, \mu, \sigma) = \begin{cases} 0 & a \leq \mu \leq b \\ \frac{1}{\sigma^2}(\mu - a)^2 & \mu \leq a \\ \frac{1}{\sigma^2}(\mu - b)^2 & b < \mu \end{cases}$$



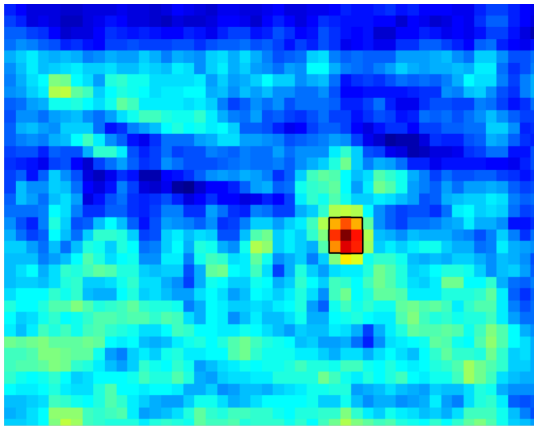
This work: BB for Deformable Part Models

Input & Detection result

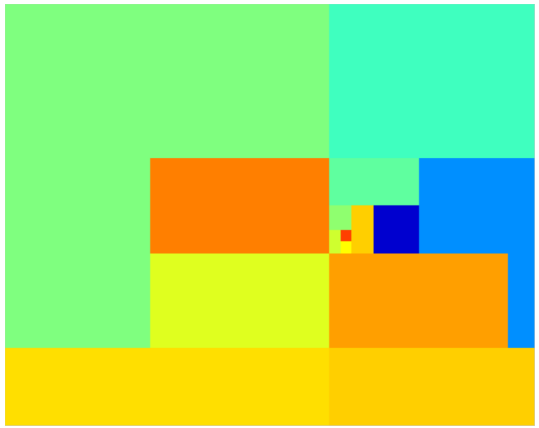
Best-case: $O(P \log N)$



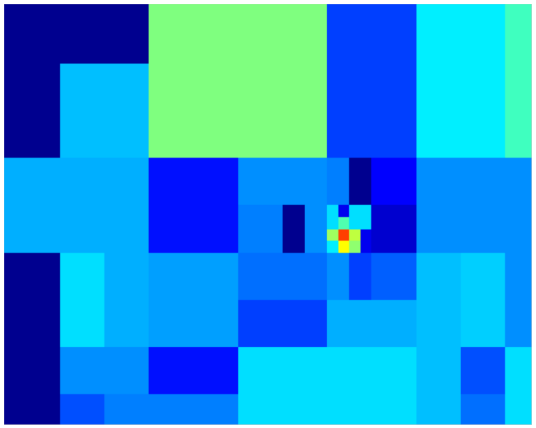
Detector score $S(x)$



BB for $\arg\max_x S(x)$

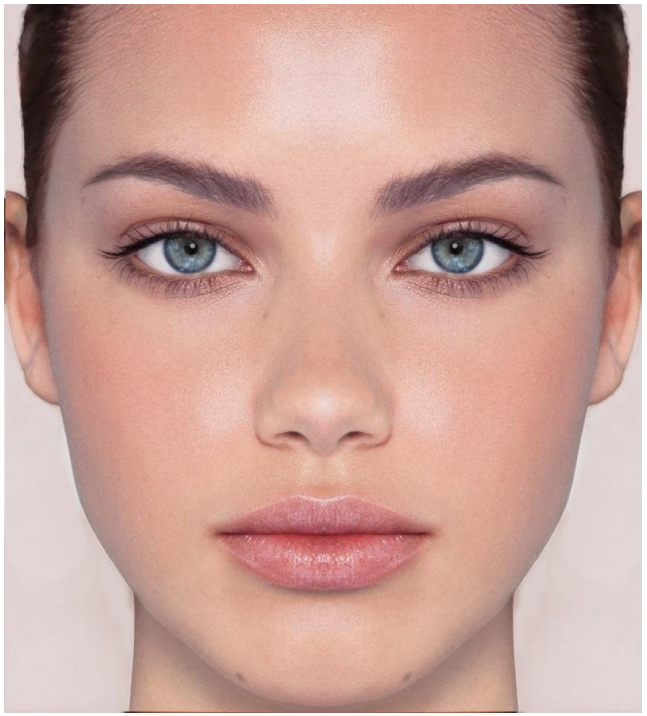


BB for $S(x) \geq -1$



Deformable Part Models (DPMs)

$$\mathbf{S}(\mathbf{x}) = \sum_{p=1}^{P} U_p(x_p) + B_p(x, x_p)$$



Local appearance

$$U_p(x_p) = \langle w_p, H(x_p) \rangle$$

Pairwise compatibility

$$B_p(x, x_p) = -(h - h_p - \hat{h}_p)^2 \eta - (v - v_p - \hat{v}_p)^2 \nu$$

Object score

$$S(x) = \sum_{p=1}^{P} \max_{x'} \left[U_p(x') + B_p(x, x') \right]$$

Bounding the DPM cost function

Wanted: $\max_{x \in X} S(x) \leq \overline{S}(X)$

$$S(x) = \sum_{p=1}^{P} m_p(x) \quad m_p(x) = \max_{x'} \left[U_p(x') + B_p(x, x') \right]$$

Property: $\max_{x \in X} h(x) + g(x) \leq \max_{x \in X} h(x) + \max_{x \in X} g(x)$

$$\max_{x \in X} \sum_{p=1}^{P} m_p(x) \leq \sum_{p=1}^{P} \max_{x \in X} m_p(x)$$

to-do

$$\max_{x \in X} S(x) \leq \sum_{p=1}^{P} \overline{m}_p(X) \doteq \overline{S}(X)$$

Bounding the source-to-domain contributions

$$\mu_d^s \doteq \max_{x \in X_d} \max_{x' \in X_s} U(x') + B(x', x)$$

Upper bound: $\mu_d^s \leq \max_{x' \in X_s} U(x') + \max_{x \in X_d} \max_{x' \in X_s} B(x', x)$

Intuitively: take best source point, put it at best possible source location

Quantities involved: $\overline{u}^s \doteq \max_{x' \in X_s} U(x'), \quad \overline{g}_d^s \doteq \max_{x \in X_d} \max_{x' \in X_s} B(x', x)$

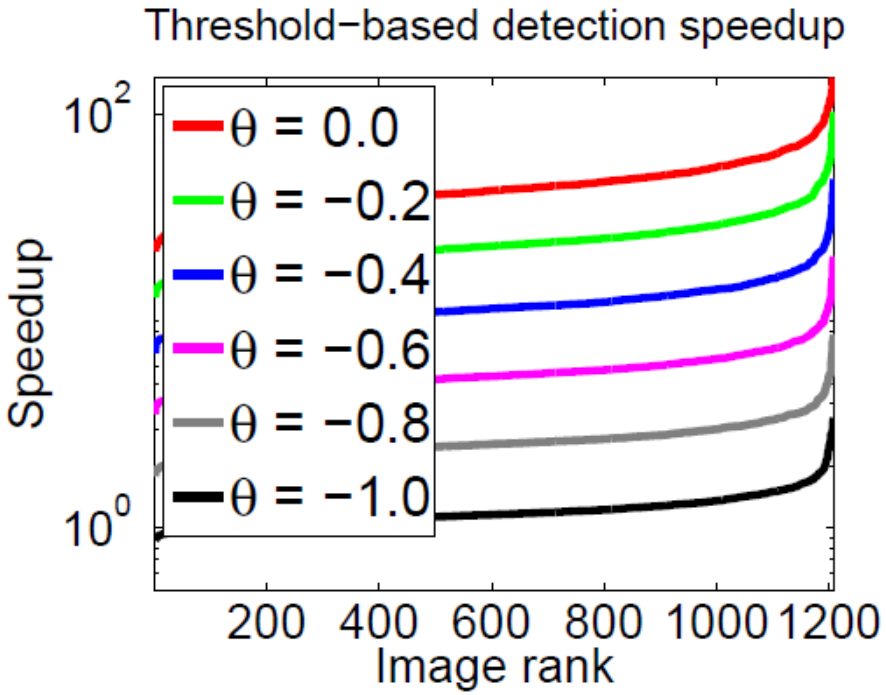
Fine-to-coarse - O(N)

Analytic expression

Pruning over combinations of s and d: Dual Trees

A.G. Gray and A.W. Moore. Nonparametric density estimation: Toward computational tractability. ICDM 2003

Results on Pascal VOC



1200 images, 20 Categories per image

Dual Trees

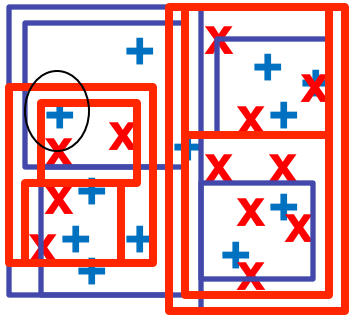
Data structure developed for Kernel Density estimation

$$P(x_j) = \sum_{i=1}^{N} w_i K(x_j, x_i), \quad x_i \in X_S,\ x_j \in X_D \quad j = 1, \ldots, M$$

Naïve solution: $O(|X_S||X_D|)$

Smarter: KD-tree for source points

Smartest: do this in batch mode for domain points



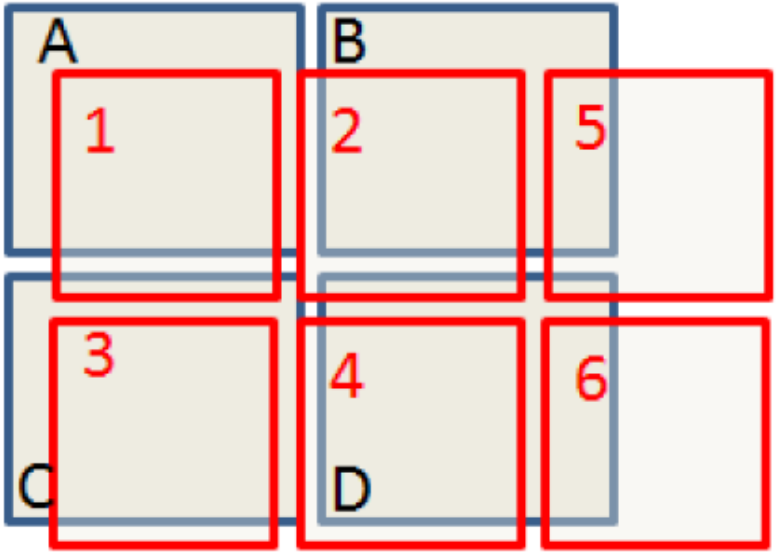
A.G. Gray and A.W. Moore. Nonparametric density estimation: Toward computational tractability. ICDM 2003

A. T. Ihler, E. B. Sudderth,W. T. Freeman, and A. S.Willsky. Efficient multiscale sampling from products of gaussian mixtures (NBP). In *NIPS, 2003.*

Breaking up the message bound

$$m(X) \doteq \max_{x \in X} \max_{x' \in X'} U(x') + B(x', x)$$

Partition domains: $X = \cup_{d \in D} X_d, \quad X' = \cup_{s \in S} X_s$



Define: $\mu_d^s \doteq \max_{x \in X_d} \max_{x' \in X_s} U(x') + B(x', x)$

$$m(X) = \max_d \max_s \mu_d^s$$

Bounding the source-to-domain contributions

$$\mu_d^s \doteq \max_{x \in X_d} \max_{x' \in X_s} U(x') + B(x', x)$$

Constrained optimization: $\max_{x \in X_d} \left[\max_{x' \in X_s} U(x') + \max_{x'' \in X_s} B(x'', x) \right]$

$$\text{s.t. } x'' = x'$$

Removing the constraint gives upper bound:

$$\mu_d^s \leq \max_{x' \in X_s} U(x') + \max_{x \in X_d} \max_{x' \in X_s} B(x', x)$$

Intuitively: take best source point, put it at best possible source location

Quantities involved: $\overline{u}^s \doteq \max_{x' \in X_s} U(x'), \quad \overline{g}_d^s \doteq \max_{x \in X_d} \max_{x' \in X_s} B(x', x)$

Bounding the source-to-domain contributions-II

$$\lambda_d^s = \min_{x \in X_d} \max_{x' \in X_s} U(x') + B(x', x)$$

Lower bound -i $$\underline{\lambda}_{d,1}^s = \max_{x' \in X_s} U(x') + \min_{x \in X_d} \min_{x' \in X_s} B(x', x) \leq \lambda_d^s$$

Intuitively: take best source point, put it at worst possible location

Lower bound -ii $$\underline{\lambda}_{d,2}^s = \min_{x' \in X_s'} U(x') + \min_{x \in X_d} \max_{x' \in X_s} B(x', x) \leq \lambda_d^s$$

Intuitively: take worst source point, put it at best possible location

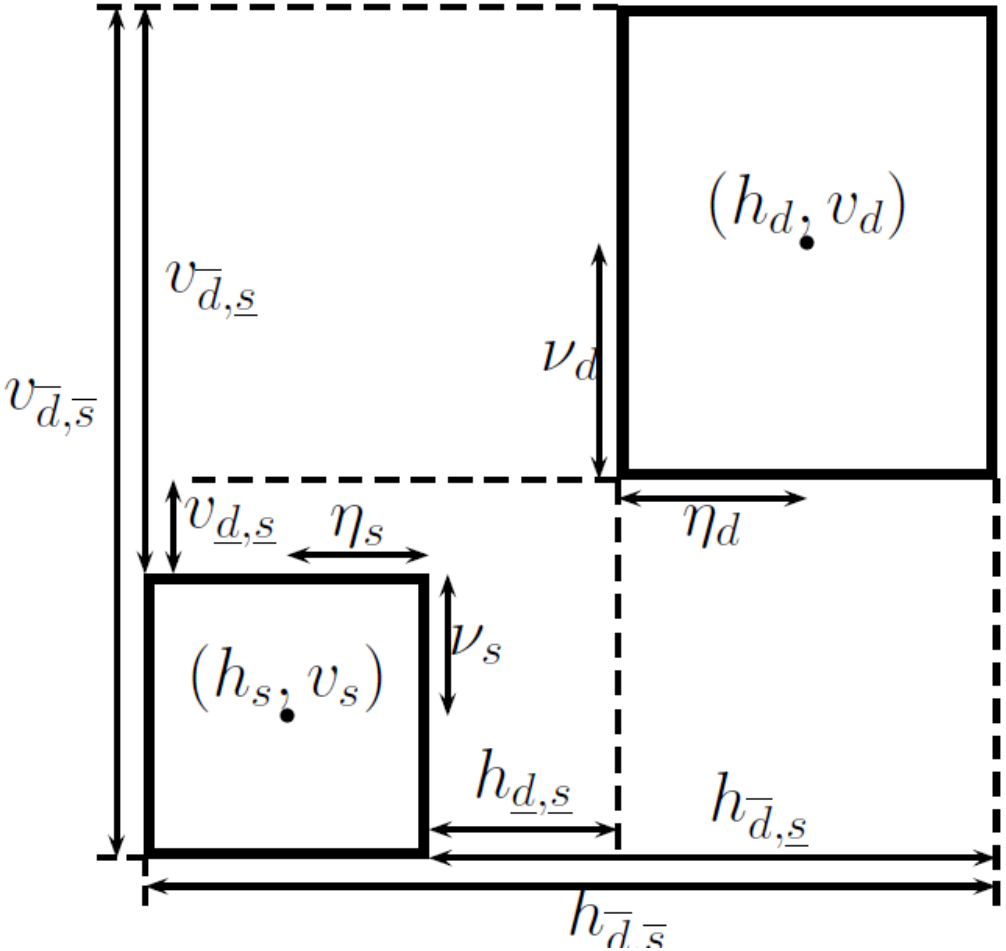
$$\underline{\lambda}_d^s = \max\left(\underline{\lambda}_{d,1}^s, \underline{\lambda}_{d,2}^s\right)$$

Analytical expressions for geometric bounds

Pairwise term: $\mathcal{G}_{x,x'} = -H(h-h')^2 - V(v-v')^2$

$$
\begin{aligned}
\mathcal{G}_{\overline{d},\overline{s}} &\doteq \max_{x\in X_d}\max_{x'\in X_s}\mathcal{G}_{x,x'} \\
&= \max_{h\in X_d^h}\max_{h'\in X_s^h} -H(h-h')^2 + \max_{v\in X_d^v}\max_{v'\in X_s^v} -H(v-v')^2 \\
&= -H h_{\underline{d},\underline{s}}^2 - V v_{\underline{d},\underline{s}}^2, \quad \text{where} \\
h_{\underline{d},\underline{s}} &\doteq \min_{h\in X_d^h}\min_{h'\in X_s^h}|h-h'|, \quad v_{\underline{d},\underline{s}} \doteq \min_{v\in X_d^v}\min_{v'\in X_s^c}|v-v'| \\
\mathcal{G}_{\underline{d},\overline{s}} &\doteq \min_{x\in X_d}\max_{x'\in X_s}\mathcal{G}_{x,x'} = -H h_{\underline{d},\underline{s}}^2 - V v_{\underline{d},\underline{s}}^2, \quad \text{where} \\
h_{\overline{d},\underline{s}} &\doteq \max_{h\in X_d^h}\min_{h'\in X_s^h}|h-h'|, \quad v_{\overline{d},\underline{s}} \doteq \max_{v\in X_d^v}\min_{v'\in X_s^v}|v-v'| \\
\mathcal{G}_{\underline{d},\underline{s}} &\doteq \min_{x\in X_d}\min_{x'\in X_s}\mathcal{G}_{x,x'} = -H h_{\overline{d},\overline{s}}^2 - V v_{\overline{d},\overline{s}}^2, \quad \text{where} \\
h_{\overline{d},\overline{s}} &\doteq \max_{h\in X_d^h}\max_{h'\in X_s^h}|h-h'|, \quad v_{\overline{d},\overline{s}} \doteq \max_{v\in X_d^v}\max_{v'\in X_s^v}|v-v'|
\end{aligned}
$$

Exploit rectangular domain



$$
\begin{aligned}
h_{\overline{d},\overline{s}} &= (h_d + \eta_d) - (h_s + \eta_s) \\
h_{\overline{d},\underline{s}} &= (h_d - \eta_d) - (h_s + \eta_s) \\
h_{\underline{d},\underline{s}} &= (h_d + \eta_d) - (h_s - \eta_s) \\
v_{\overline{d},\overline{s}} &= (v_d + \nu_d) - (v_s + \nu_s) \\
v_{\overline{d},\underline{s}} &= (v_d - \nu_d) - (v_s + \nu_s) \\
v_{\underline{d},\underline{s}} &= (v_d + \nu_d) - (v_s - \nu_s)
\end{aligned}
$$

Done so far

$$m(X) \doteq \max_{x \in X} \max_{x' \in X'} U(x') + B(x', x)$$

$$X = \cup_{d \in D} X_d, \quad X' = \cup_{s \in S} X_s$$

$$\mu_d^s \doteq \max_{x \in X_d} \max_{x' \in X_s} U(x') + B(x', x)$$

$$m(X) = \max_d \max_s \mu_d^s$$

$$\overline{m}(X) = \max_d \max_s \overline{\mu}_d^s$$

Large intervals: loose bounds

Small intervals: tight bounds, but maximization over multiple terms

$$|S| = \frac{|X|}{|X_s|}$$

Need to prune

Dual Recursion

Source-/ Domain-tree: KD-trees for source/domain points, respectively.

Goal: keep manageable the computation of $\overline{m}(X) = \max_d \max_s \overline{\mu}_d^s$

Key idea: entertain list of s that can possibly contribute to any d

$$\mathcal{S}_d = \{s_i\}$$

Originally: d is Domain-tree root, its supporter is Source-tree root

$$\mathcal{S}_0 = \{s_0\}$$

When splitting (branching) d, split also its supporters

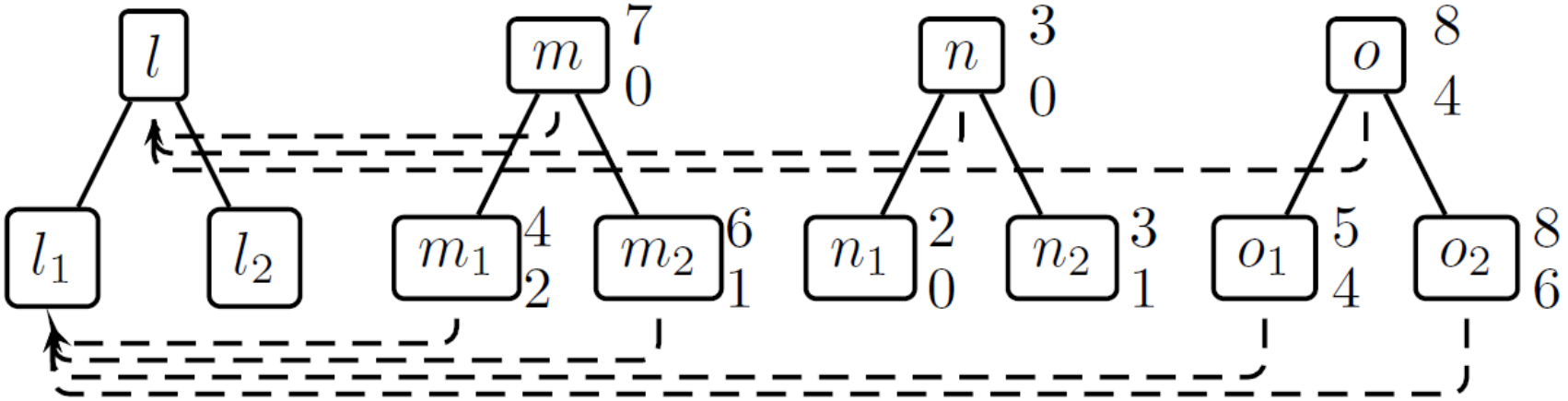
Prune supporters based on upper and lower bounds

Supporter pruning

$\mathcal{S}_l = \{m, n, o\}$ Source nodes m; n; o `support' domain node l

Recurse: $\mathcal{S}_{ch(l)} = \text{Prune}\left[ch(m) \cup ch(n) \cup ch(o)\right]$

Pruning criterion: $\overline{\mu}_d^l < \max_{j \in \mathcal{S}_d} \underline{\lambda}_d^j$

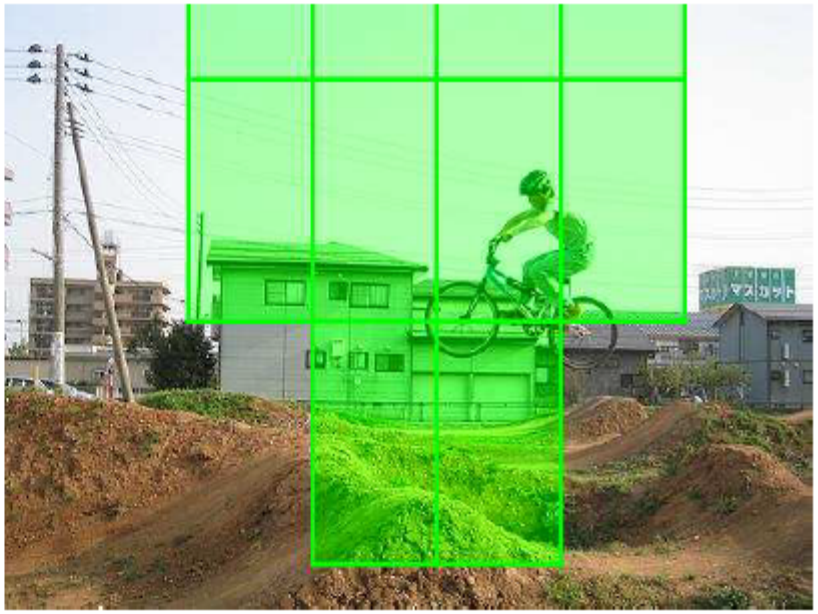


DTBB demonstration

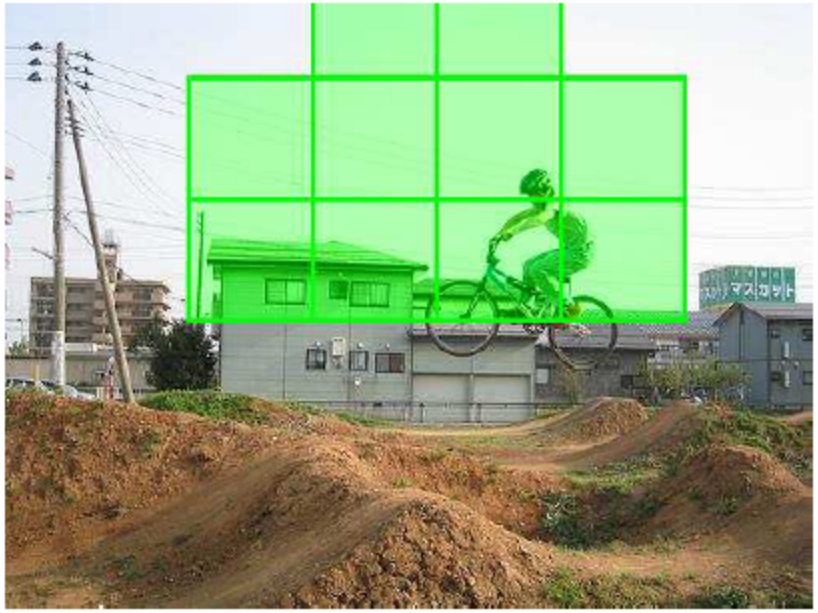
DTBB demonstration

DTBB demonstration

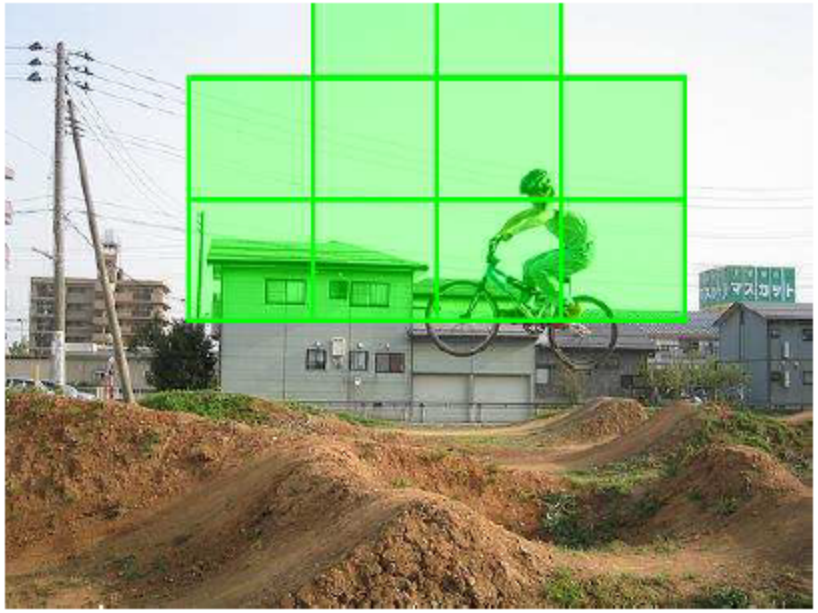
DTBB demonstration



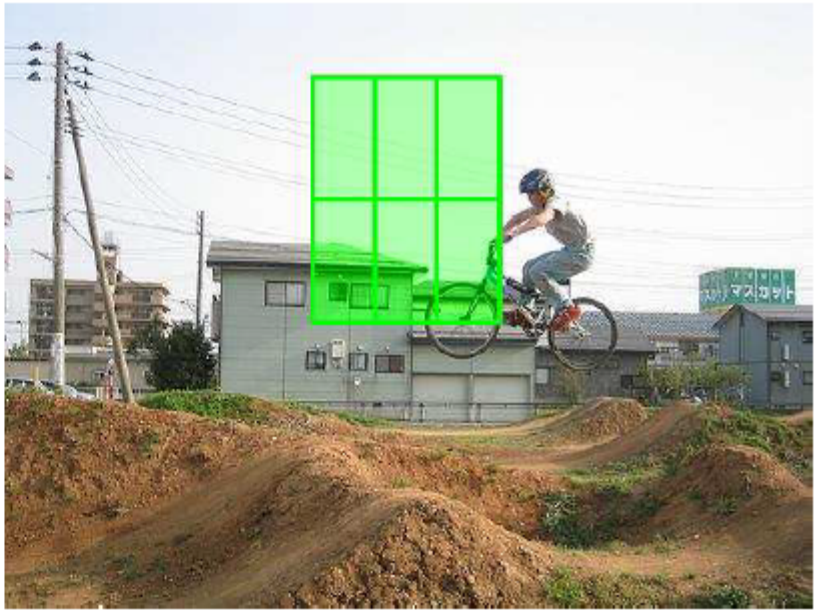
DTBB demonstration



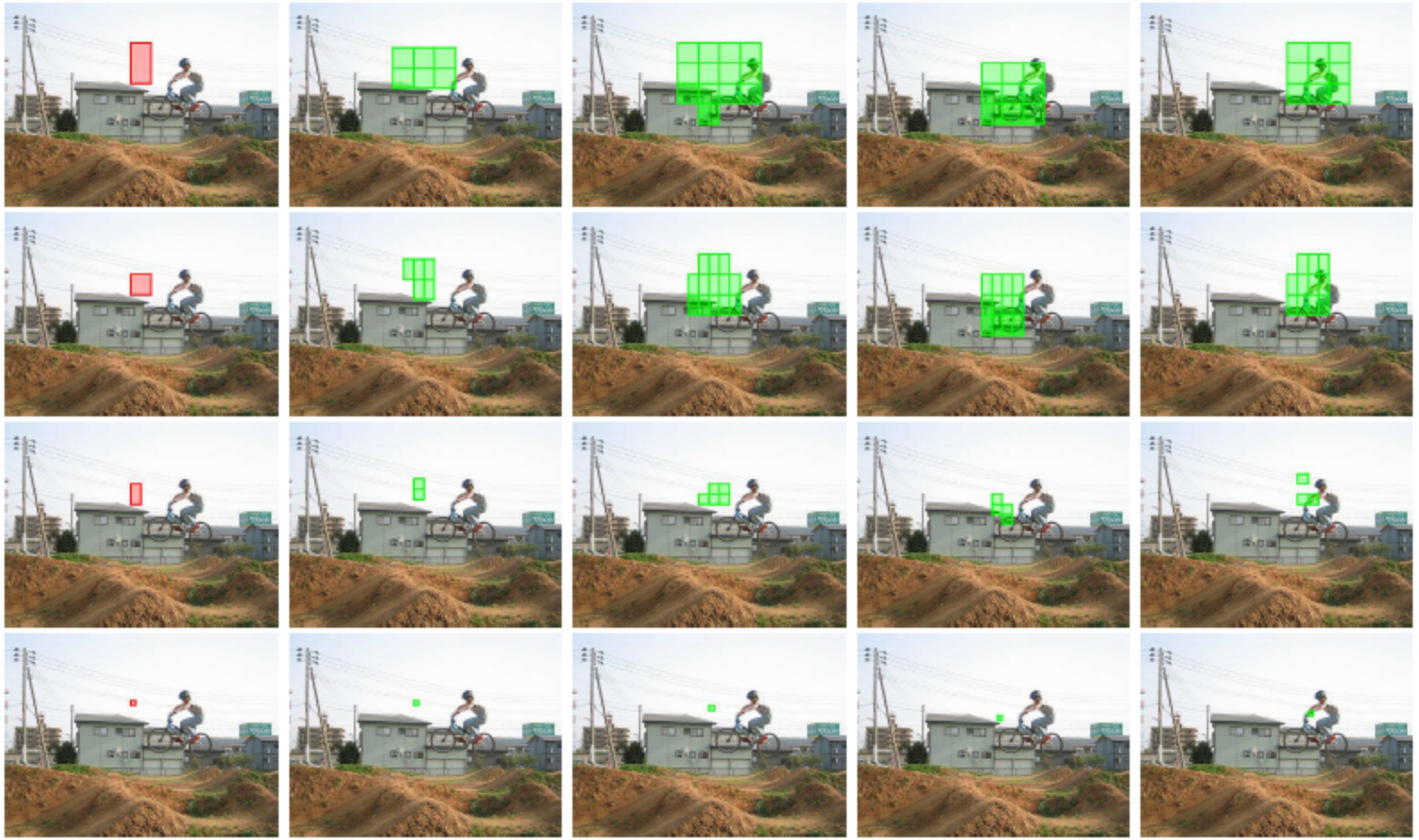
DTBB demonstration



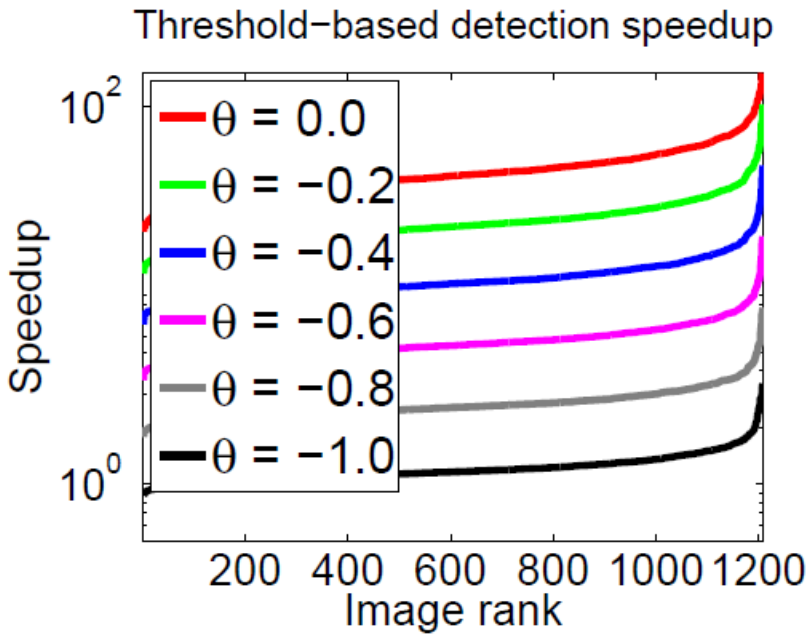
DTBB demonstration

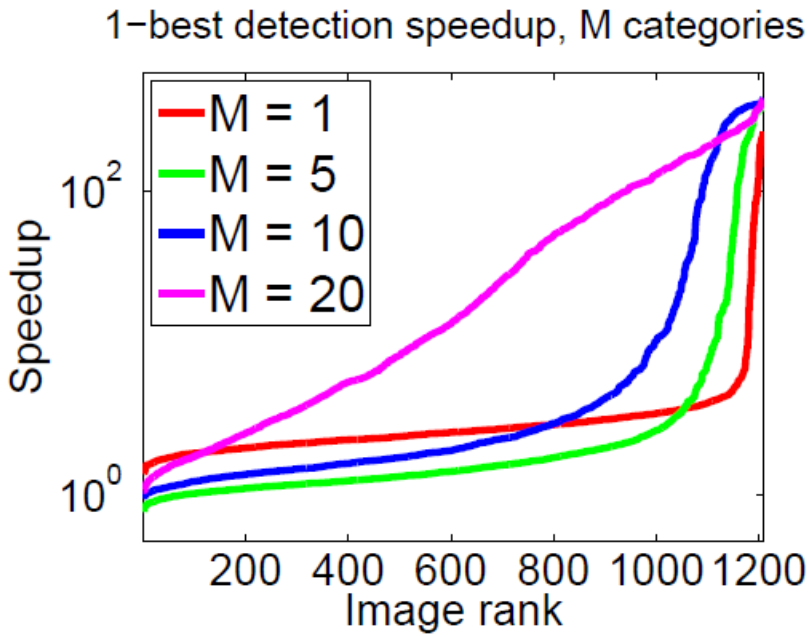


Supporter pruning demonstration



Results on Pascal VOC





1200 images, 20 Categories per image

Results on Pascal VOC

	Our algorithm	[4]
Unary terms	13.20 ± 1.49	159.41 ± 15.82
KD-trees	1.72 ± 0.21	0.00 ± 0.00
Detection, $\theta = 0.0$	0.25 ± 0.07	10.74 ± 1.02
Detection, $\theta = -.2$	0.47 ± 0.12	10.74 ± 1.02
Detection, $\theta = -.4$	0.93 ± 0.22	10.74 ± 1.02
Detection, $\theta = -.6$	1.95 ± 0.42	10.74 ± 1.02
Detection, $\theta = -.8$	4.17 ± 0.84	10.74 ± 1.02
Detection, $\theta = -1$	9.14 ± 1.79	10.74 ± 1.02
Detection, 1-best	0.41 ± 0.08	10.74 ± 1.02
Detection, 5-best	0.47 ± 0.09	10.74 ± 1.02
Detection, 10-best	0.48 ± 0.10	10.74 ± 1.02

1200 images, 20 Categories per image

Object detection with Deformable Part Models (DPMs)

$$U_p(x') = \langle \mathbf{w}_p, \mathbf{H}(x') \rangle$$

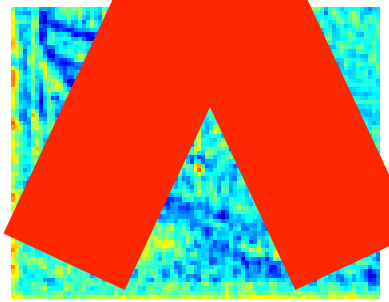
$$\max_{x'} [U_p(x') + B_p(x, x')]$$

This work

$p = 1$

$\vdots$

$p = P$



DTBB, NIPS 11

$$S(x) = \sum_{p=1}^{P} \max_{x'} [U_p(x') + B_p(x, x')]$$

Accelerating detection with DPMs

Efficient detection with DPMs

P F Felzenszwalb R B Girshick and D A McAllester Cascade object detection with DPMs CVPR 2010

B. Sapp, A. Toshev, and B. Taskar. Cascaded models for articulated pose estimation, ECCV, 2010

M. Pedersoli, A.Vedaldi, and J.Gonzalez. A coarse-to-fine approach for object detection, CVPR 2011

I. Kokkinos, Rapid DPM Detection using Dual-Tree Branch-and-Bound, NIPS 2011

Efficient part score computation

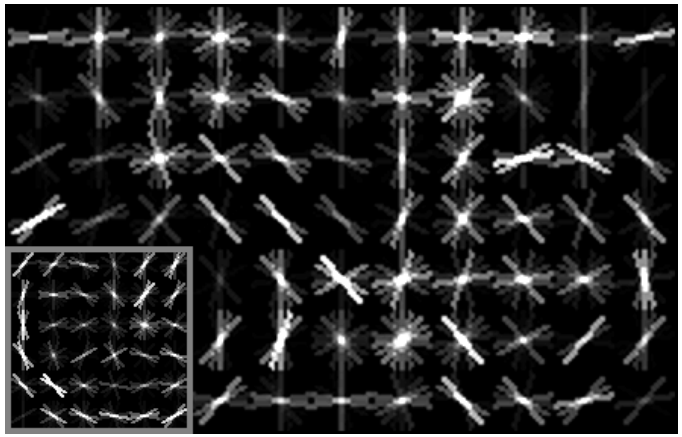
H. Pirsiavash and D. Ramanan, Steerable Part Models, CVPR 2012

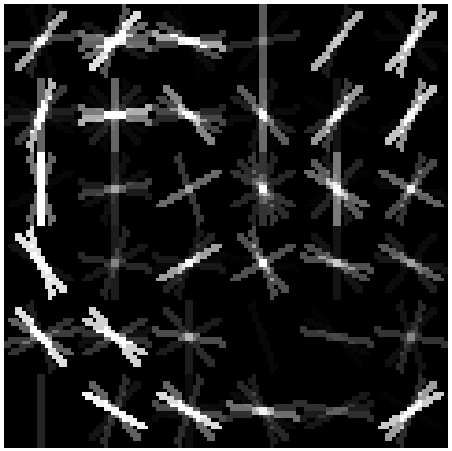
A. Vedaldi and A. Zisserman, Sparse Kernel Maps and Faster Product Quantization Learning, CVPR 2012

H.O. Song, S. Zickler, T. Althoff, R. Girschick, M. Fritz, C. Geyer, P. Felzenszwalb, T. Darrell, Sparselet Models for Efficient Multiclass Object Detection, ECCV 2012

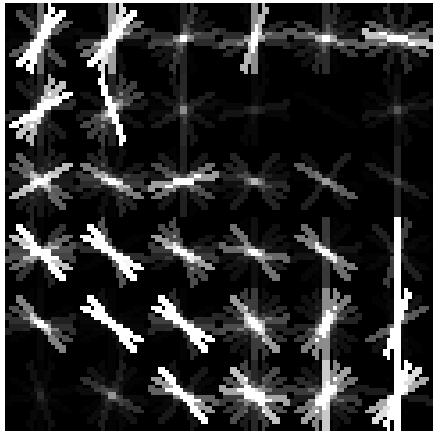
C. Dubout and F. Fleuret. Exact Acceleration of Linear Object Detectors, ECCV 2012

Part score computation





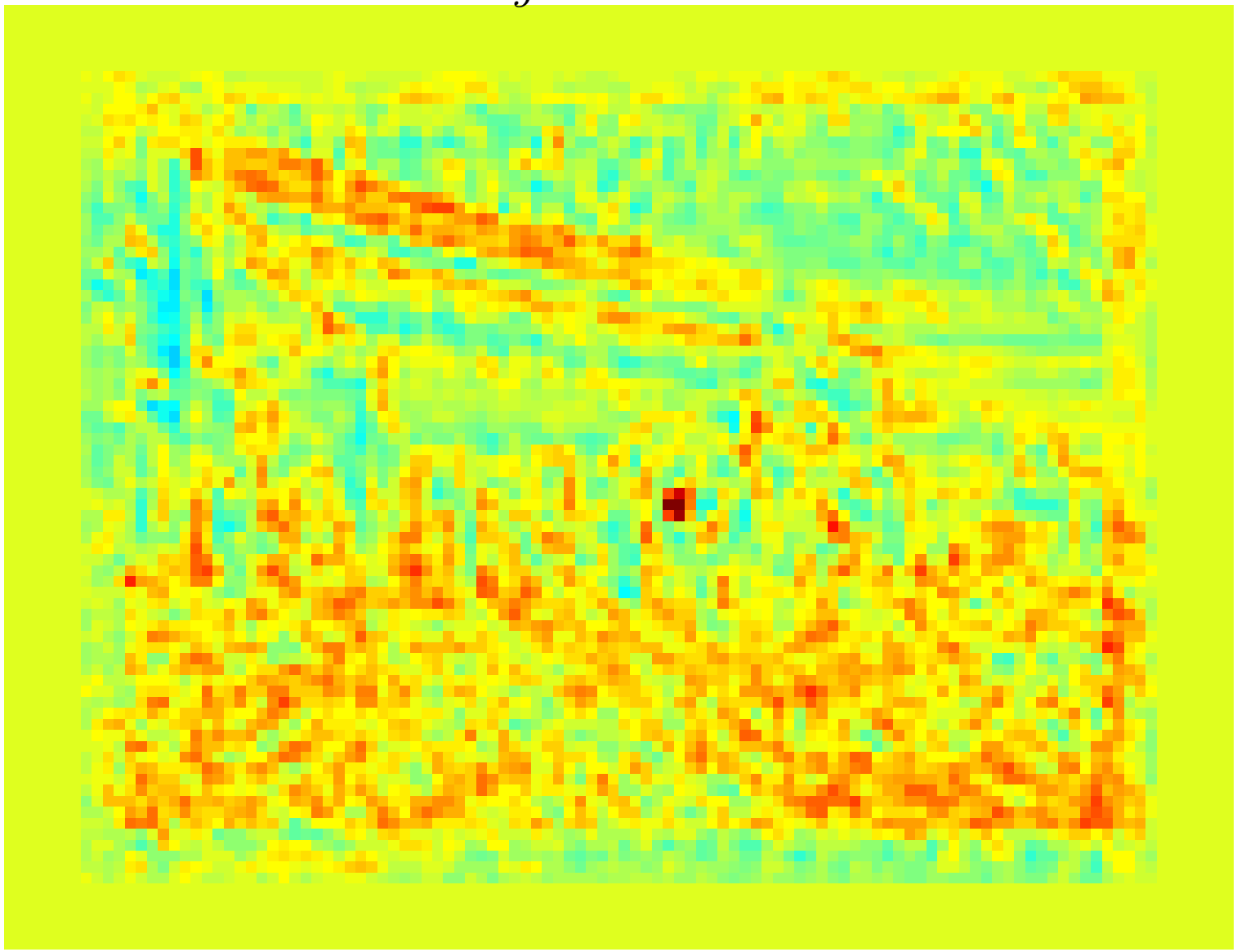
$$\mathbf{w}[y]$$



$$\mathbf{h}[x+y]$$

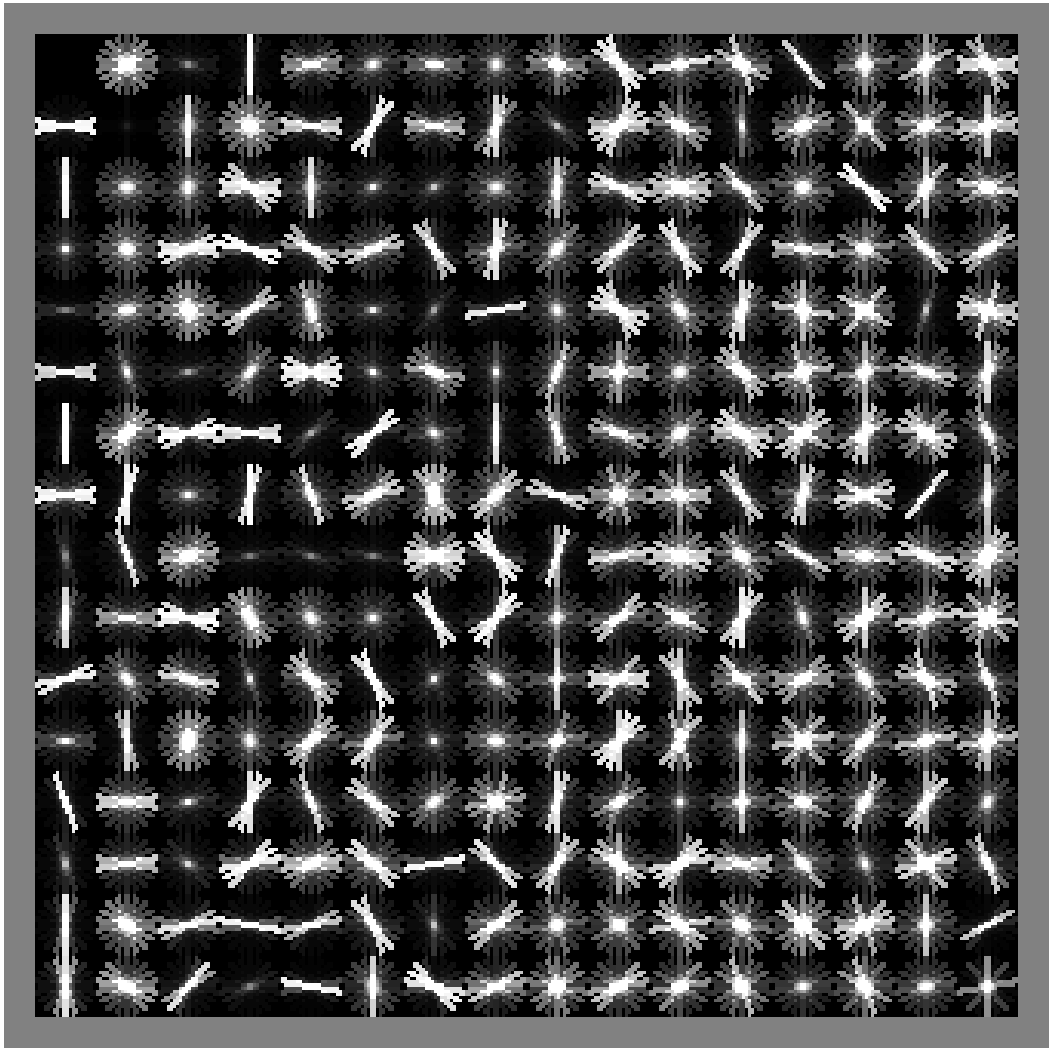
$$s[x] = \sum_{y} \langle \mathbf{h}[x+y], \mathbf{w}[y] \rangle$$

Part scores $s[x] = \sum_y \langle \mathbf{h}[x+y], \mathbf{w}[y] \rangle$



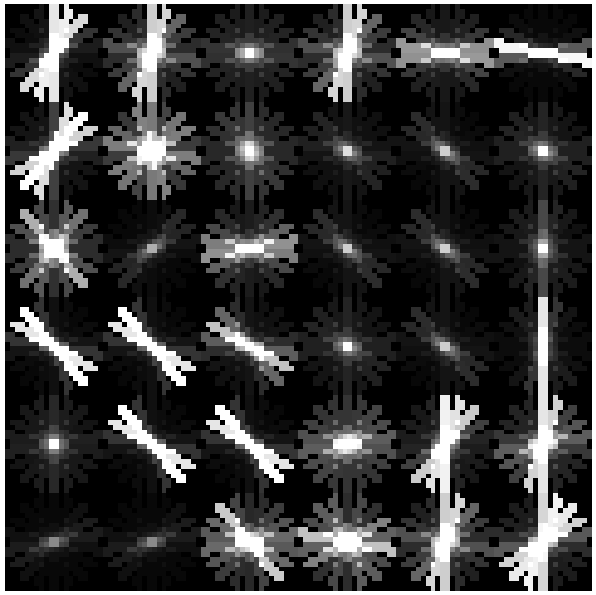
HOG cell quantization: visual ‘letters’

$$\mathcal{C} = \{C_1, \ldots, C_{256}\}$$

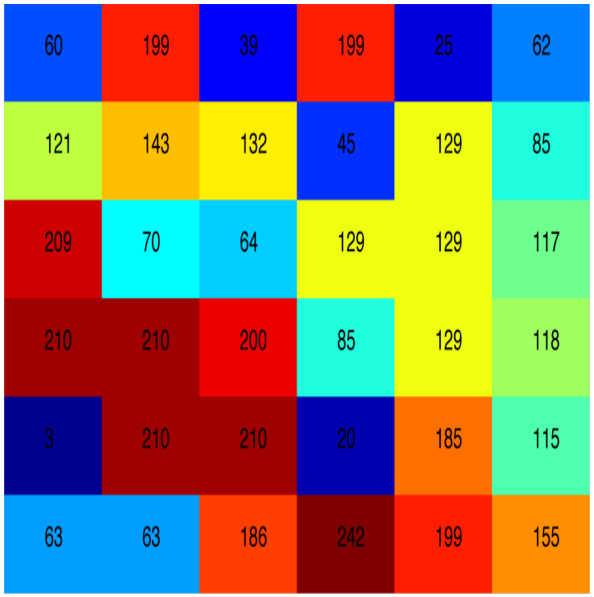


HOG feature quantization

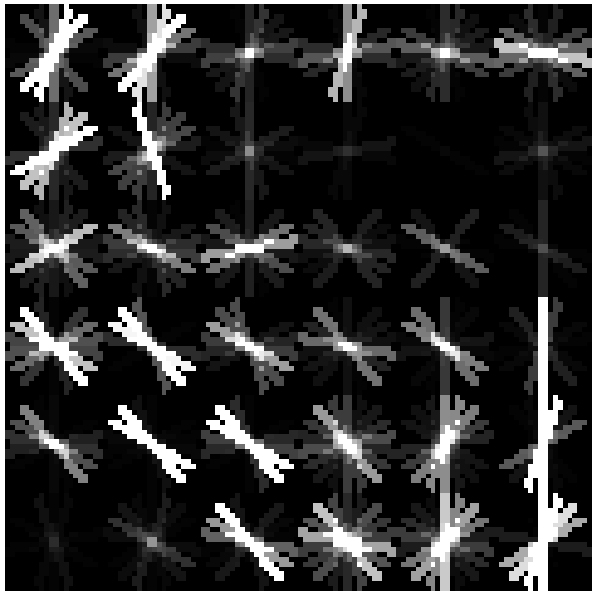
HOG detail



Codebook indices

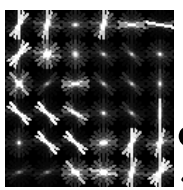
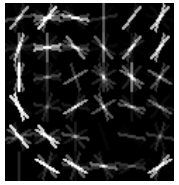
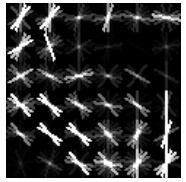


Quantized HOG



$$\mathbf{h}[x] \qquad i[x] = \arg\min_k d(\mathbf{h}[x], C_k) \qquad \hat{\mathbf{h}}[x] = C_{i[x]}$$

Efficient inner product approximation

$$\left\langle \text{  }, \text{  } \right\rangle \simeq \left\langle \text{  }, \text{  } \right\rangle$$

$$s[x] \simeq \hat{s}[x]$$

$$\sum_y \langle \mathbf{h}[x+y], \mathbf{w}[y] \rangle \simeq \sum_y \langle \hat{\mathbf{h}}[x+y], \mathbf{w}[y] \rangle$$

Efficient inner product approximation

$$\langle \mathbf{h}[x+y], \mathbf{w}[y] \rangle \simeq \langle \hat{\mathbf{h}}[x+y], \mathbf{w}[y] \rangle$$

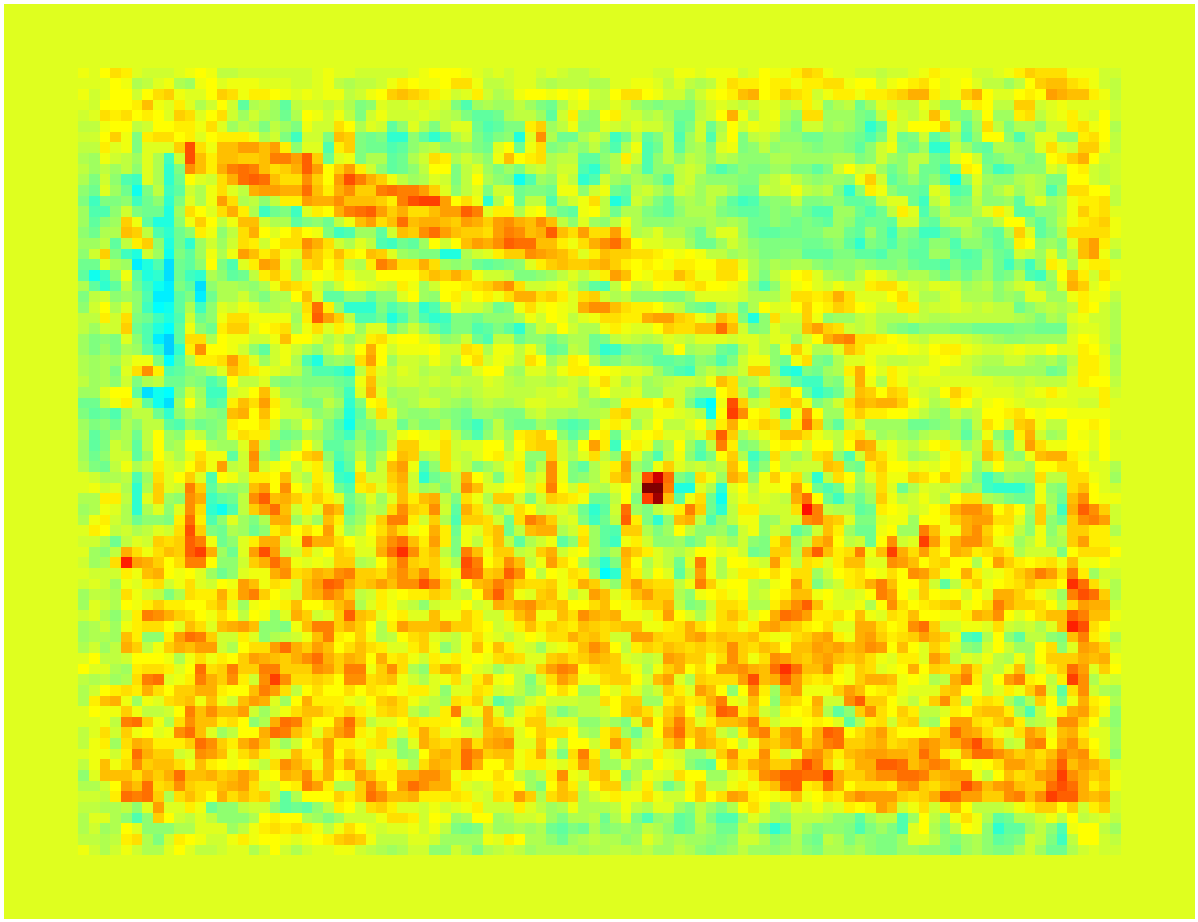
$$= \langle C_{I[x+y]}, \mathbf{w}[y] \rangle$$

$$= \Pi[I[x+y], y]$$

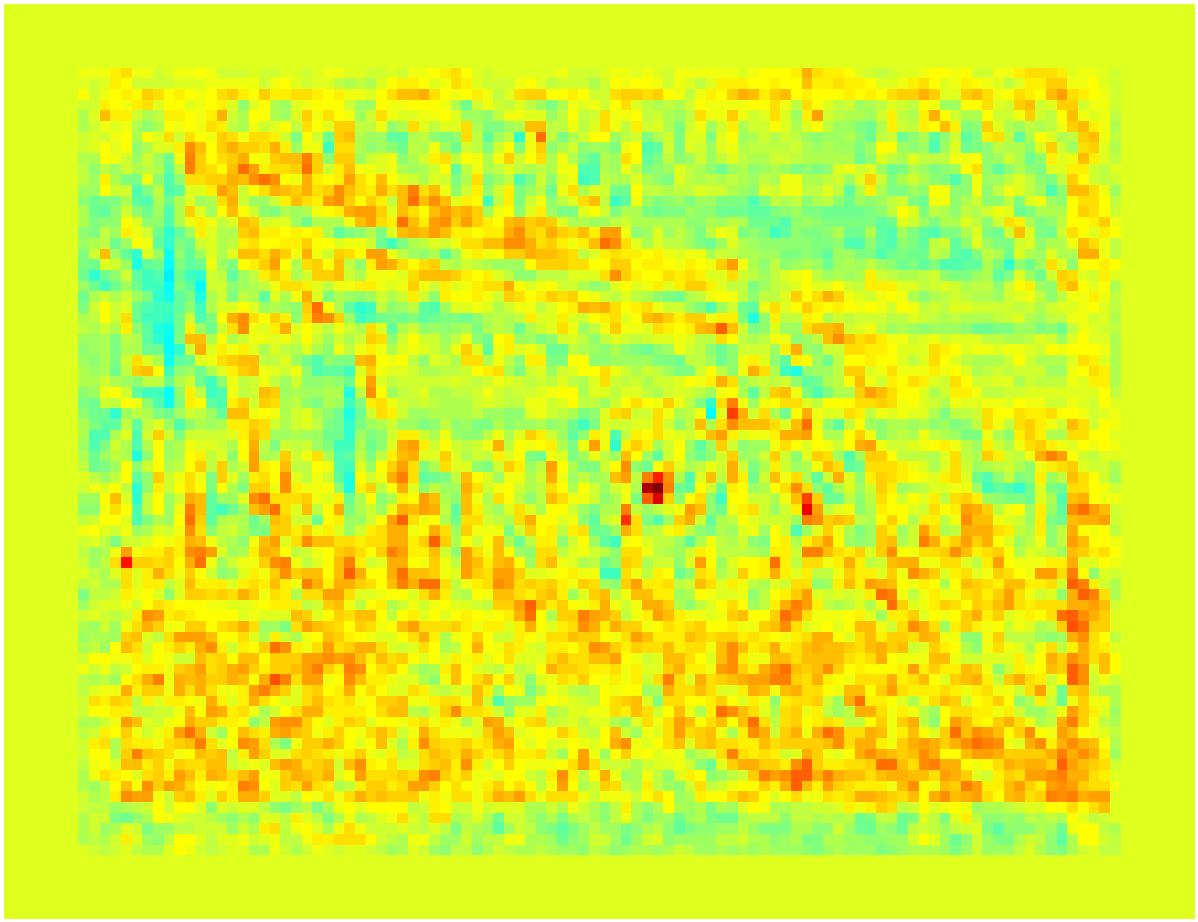
$$\Pi[k, y] = \langle C_k, \mathbf{w}_y \rangle$$

$$s[x] \simeq \hat{s}[x] = \sum_y \Pi[I[x+y], y]$$

Lookup-based estimate demonstration: $s[x]$



Lookup-based estimate demonstration: $\hat{s}[x]$



Part-level approximation error

$$\epsilon \doteq s - \hat{s} = \left\langle \quad - \quad , \quad \right\rangle$$

$$= \sum_{y} \underbrace{\langle \mathbf{h}[y] - \hat{\mathbf{h}}[y], \mathbf{w}[y] \rangle}_{e[y]}$$

Cell-level approximation error

$$e[y] = \langle \mathbf{h}[y] - \hat{\mathbf{h}}[y], \mathbf{w}[y] \rangle = \langle$$  $-$  $,$  $\rangle$

$$= \langle \mathbf{e}[y], \mathbf{w}[y] \rangle$$

$$= \sum_{f=1}^{32} \mathbf{e}_y[f] \mathbf{w}_y[f]$$

Chebyshev inequality

For any zero-mean random variable, and any value of α:

$$P(|X| > \alpha) \leq \frac{E\{X^2\}}{\alpha^2}$$

Equivalently, with probability of error smaller than p_e:

$$X \in \left[-\sqrt{\frac{E\{X^2\}}{p_e}}, \sqrt{\frac{E\{X^2\}}{p_e}} \right]$$

Chebyshev inequality-II

For a weighted sum of i.i.d. zero-mean random variables:

$$X' = \sum_{k=1}^{K} w_k X_k$$

with probability of error smaller than p_e :

$$X' \in \left[-\sqrt{\frac{\left(\sum_k w_k^2\right) E\{X^2\}}{p_e}}, \sqrt{\frac{\left(\sum_k w_k^2\right) E\{X^2\}}{p_e}} \right]$$

Chebyshev inequality for cell-level error

$$e[y] = \sum_{f=1}^{F} \mathbf{e}_y[f]\mathbf{w}_y[f]$$

with probability of error smaller than p_e :

$$e_y \in \left[-\sqrt{\frac{\|\mathbf{w}[y]\|^2 \|\mathbf{e}[y]\|^2}{p_e F}}, \sqrt{\frac{\|\mathbf{w}[y]\|^2 \|\mathbf{e}[y]\|^2}{p_e F}} \right]$$

Chebyshev inequality for part-level error

$$\boxed{\epsilon = \hat{s} - s} = \sum_y e[y]$$

with probability of error smaller than p_e :

$$\epsilon \in \left[-\sqrt{\frac{\sum_y \|\mathbf{w}[y]\|^2 \|\mathbf{e}[y]\|^2}{p_e F}}, \sqrt{\frac{\sum_y \|\mathbf{w}[y]\|^2 \|\mathbf{e}[y]\|^2}{p_e F}} \right]$$

→ with probability of error smaller than p_e:

$$s \in \left[\hat{s} - \sqrt{\frac{\sum_y \|\mathbf{w}[y]\|^2 \|\mathbf{e}[y]\|^2}{p_e F}}, \hat{s} + \sqrt{\frac{\sum_y \|\mathbf{w}[y]\|^2 \|\mathbf{e}[y]\|^2}{p_e F}} \right]$$

Recap

Lookup-based approximation:

$$s[x] \simeq \hat{s}[x] = \sum_y \Pi[I[x+y], y]$$

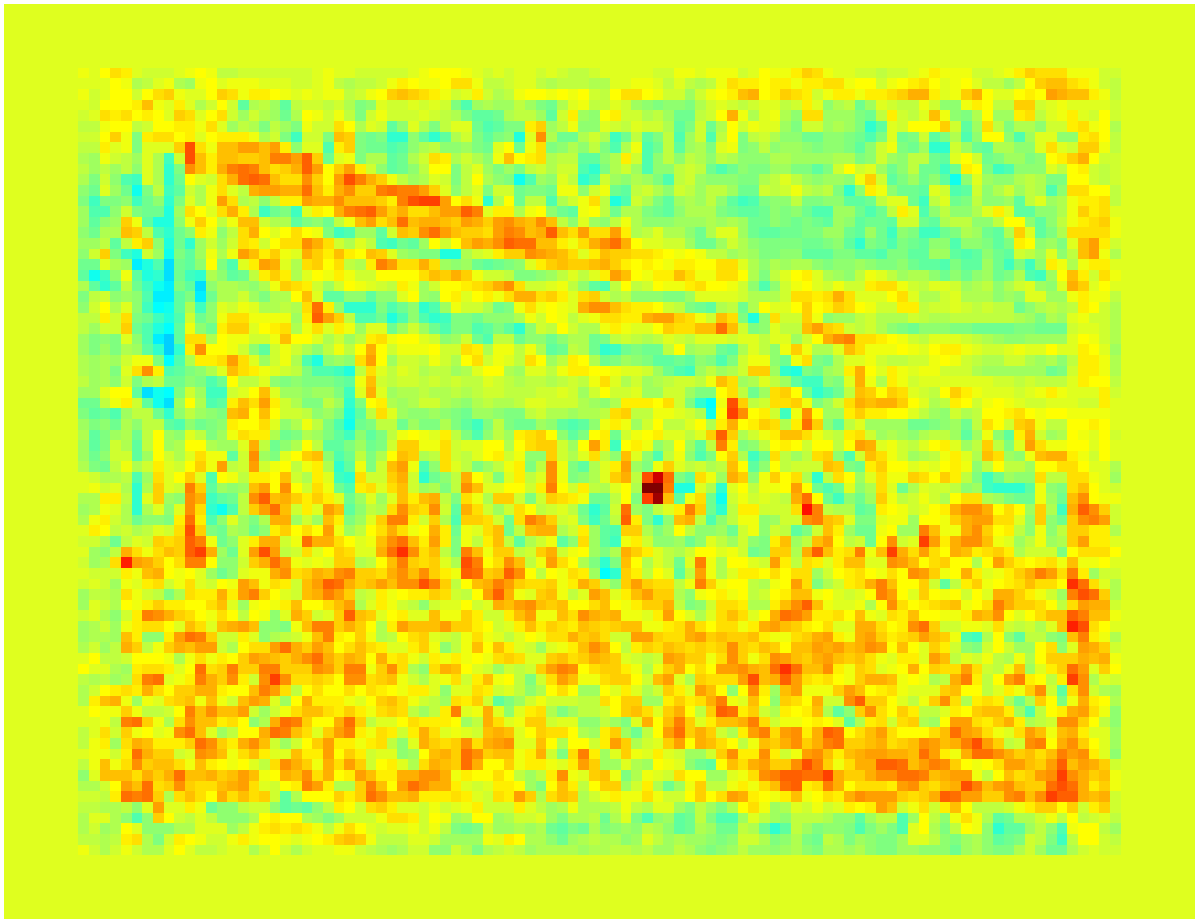
With probability of error at most p_e:

$$\underline{s}[x] \leq s[x] \leq \overline{s}[x]$$

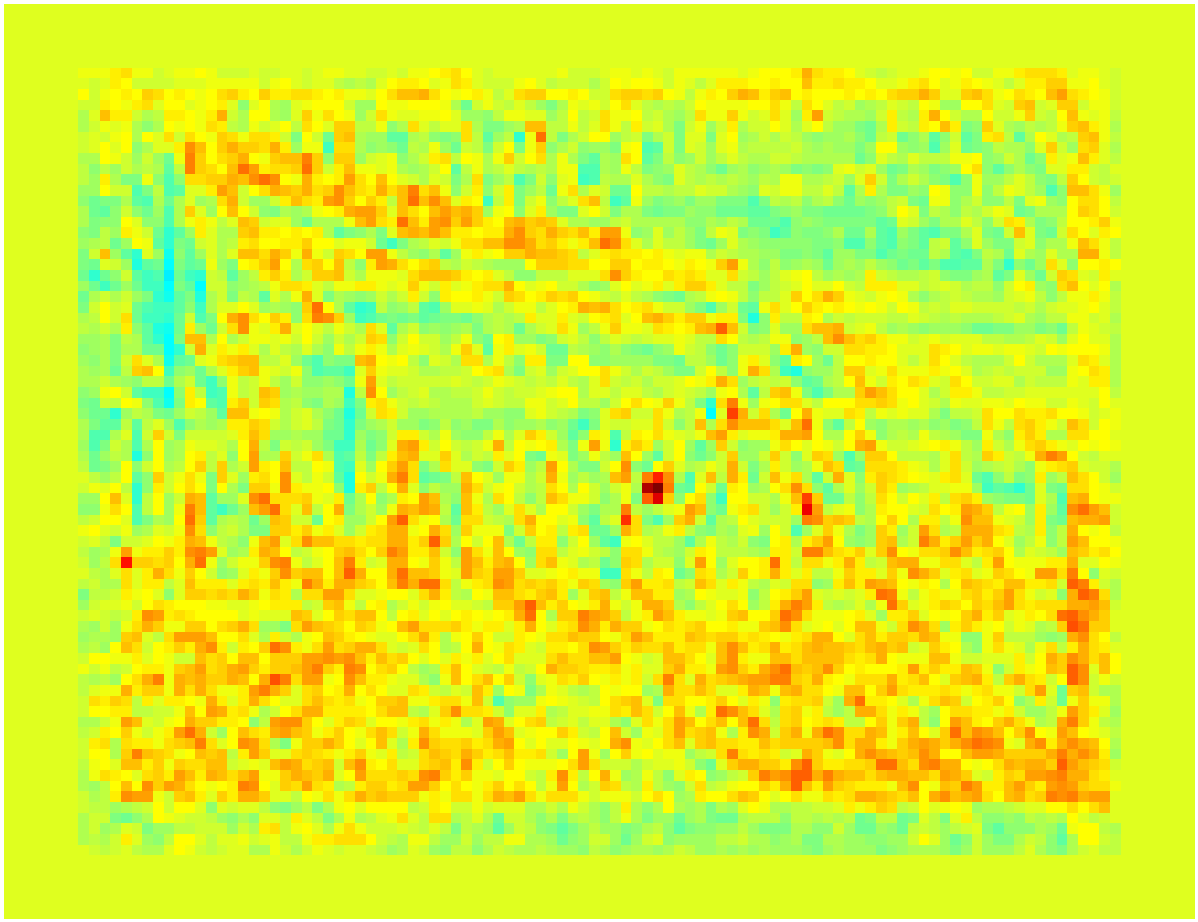
$$\underline{s}[x] = \hat{s}[x] - \sqrt{\frac{\sum_y \|\mathbf{w}[y]\|^2 \|\mathbf{e}[x+y]\|^2}{p_e F}}$$

$$\overline{s}[x] = \hat{s}[x] + \sqrt{\frac{\sum_y \|\mathbf{w}[y]\|^2 \|\mathbf{e}[x+y]\|^2}{p_e F}}$$

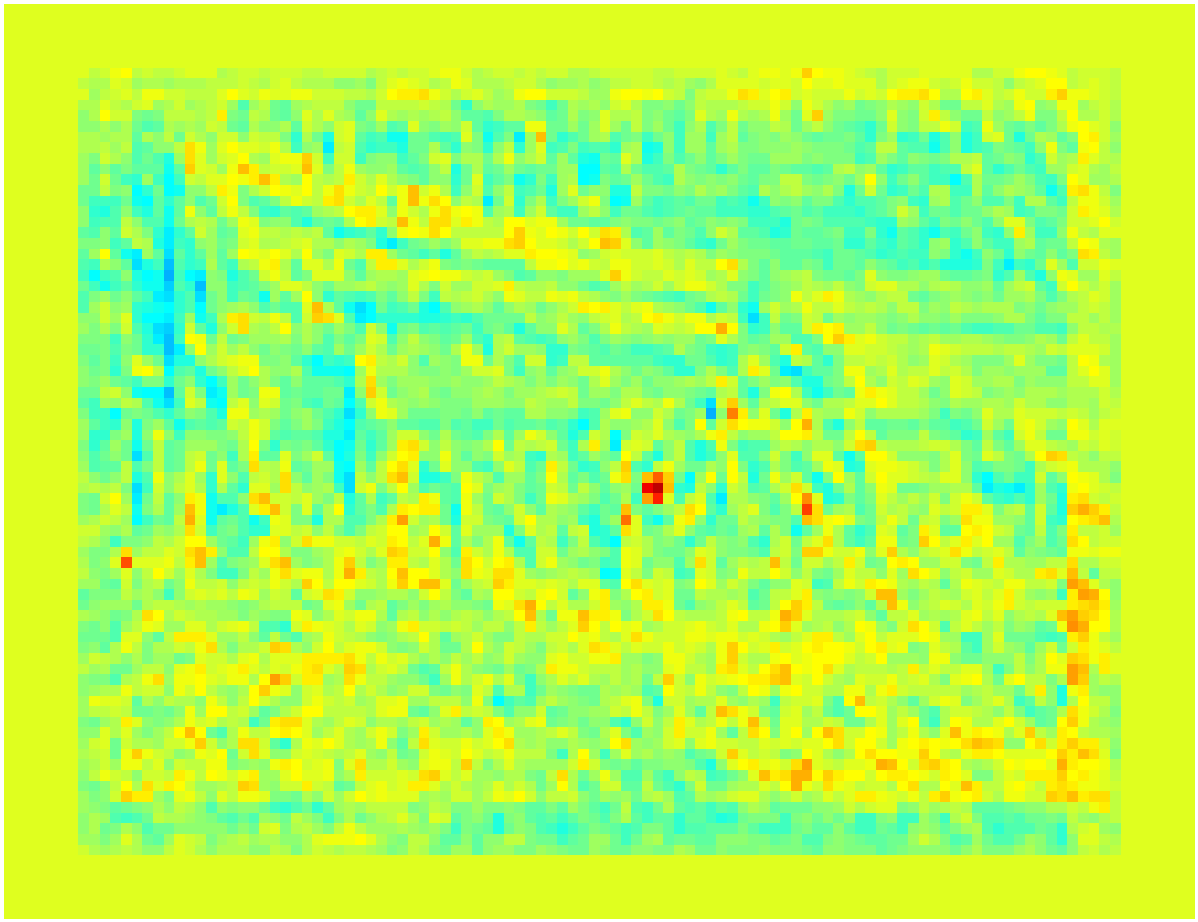
Bound demonstration: $s[x]$



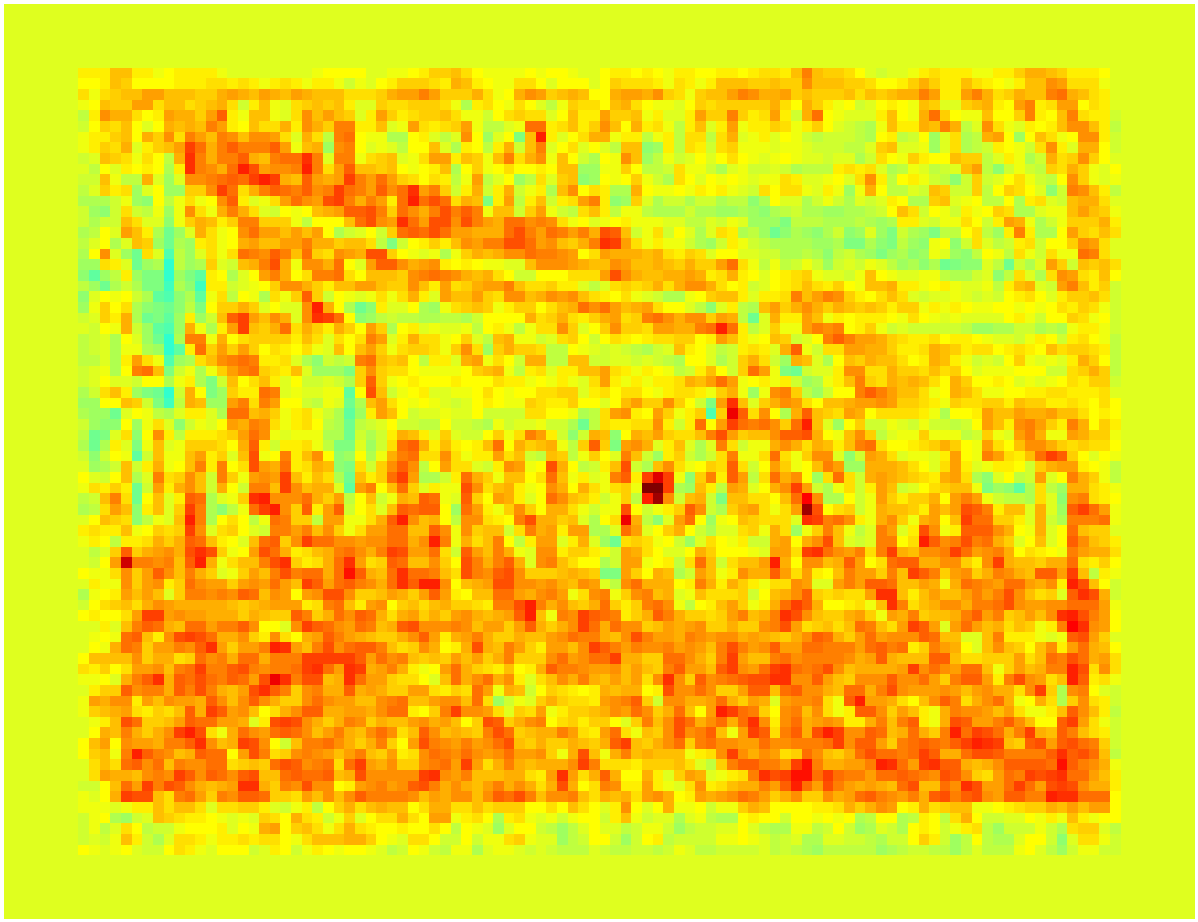
Bound demonstration: $\hat{s}[x]$



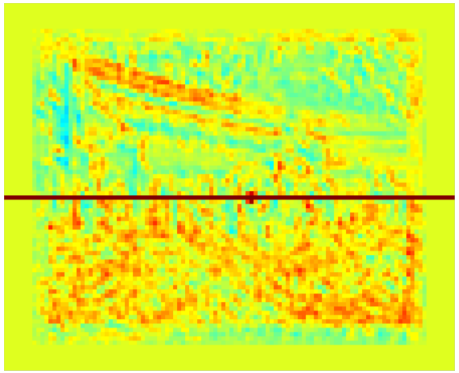
Bound demonstration: $\underline{s}[x],\ p_e = .05$

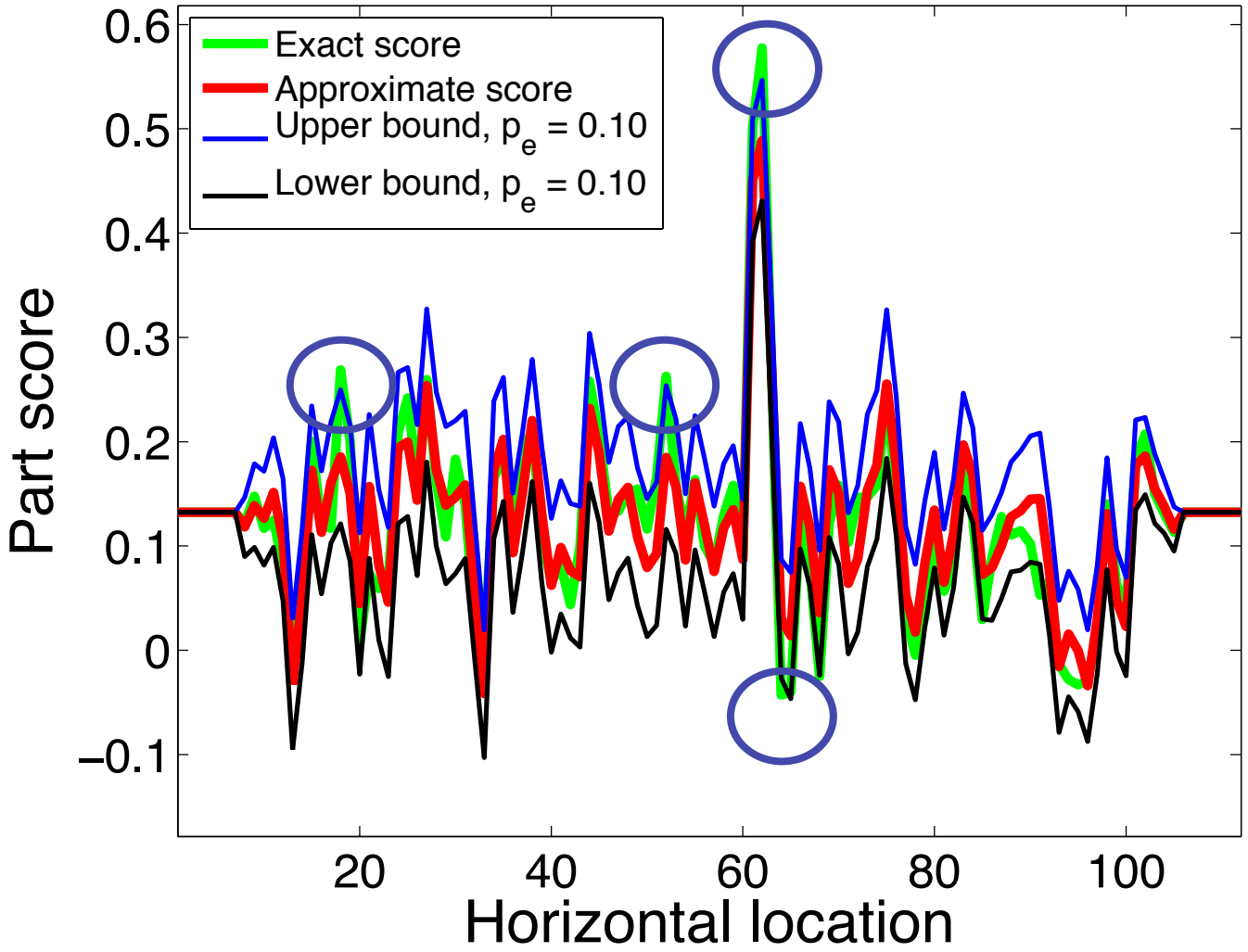


Bound demonstration: $\overline{s}[x],\ p_e = .05$

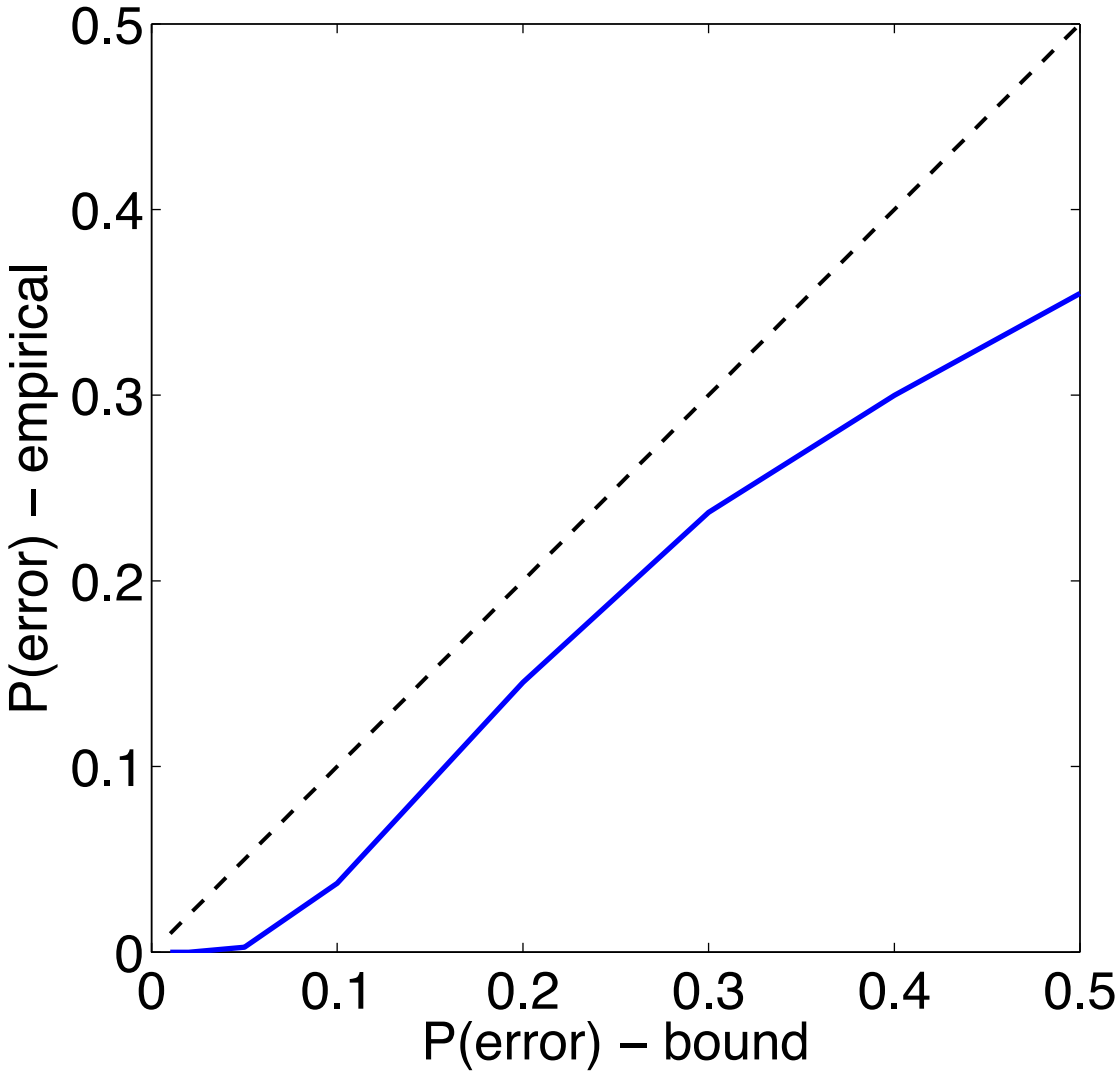


Bound demonstration for varying confidence



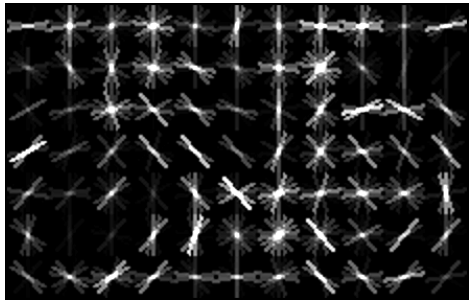


Bound tightness

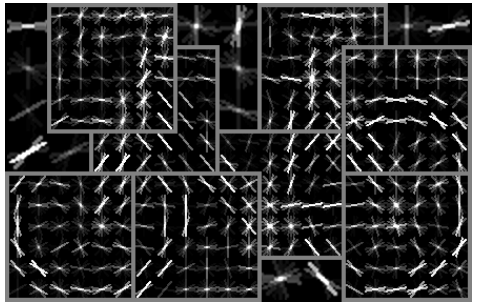


Accelerating detection with DPMs

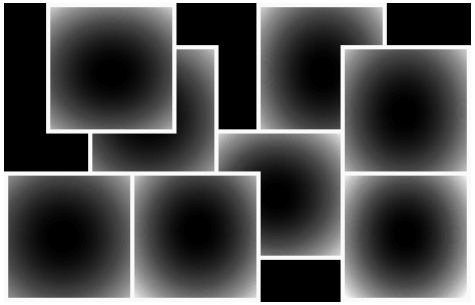
This work



w_1

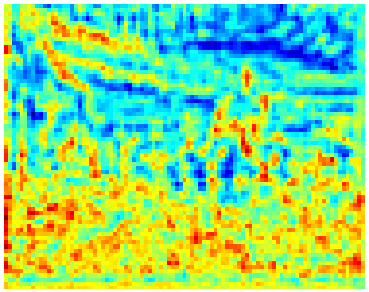


$w_2 \dots w_P$

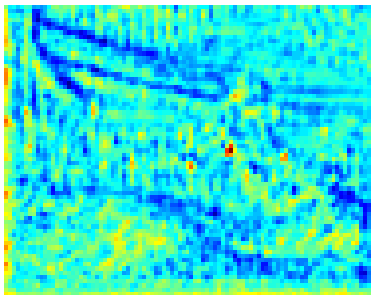


$B_2 \dots B_P$

$$U_p(x) = \langle w_p, H(x) \rangle$$



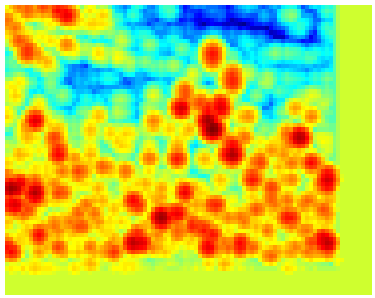
$p = 3$

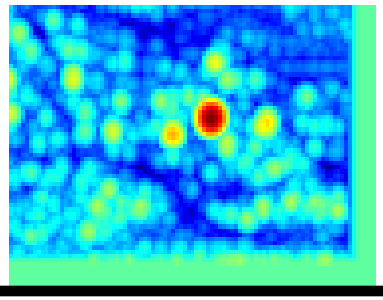


$p = 5$

GDT

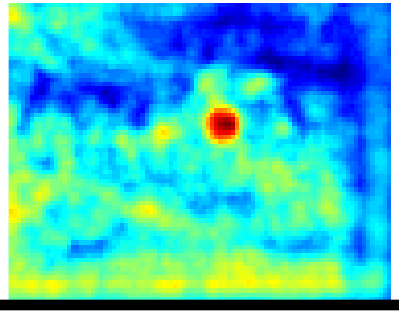
$$\mu_p(x) = \max_{x'}[U_p(x') + B_p(x, x')]$$





DTBB, NIPS 2011

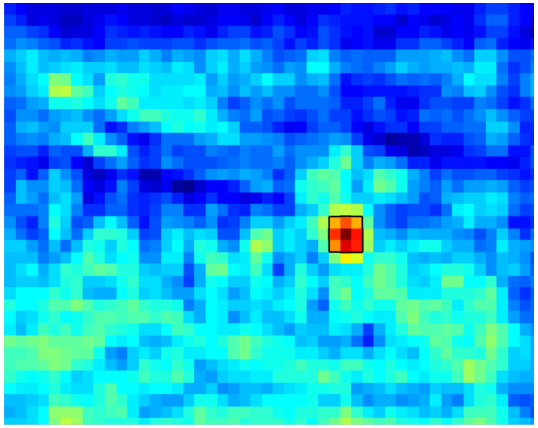
$$S(x) = \sum_{p=1}^{P} \mu_p(x)$$



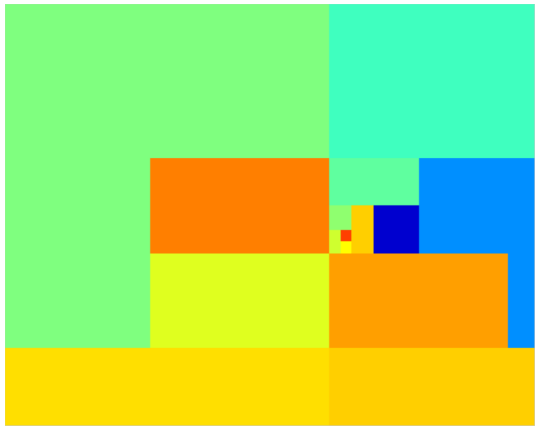
Dual-Tree Brand-and-Bound

Input & Detection result

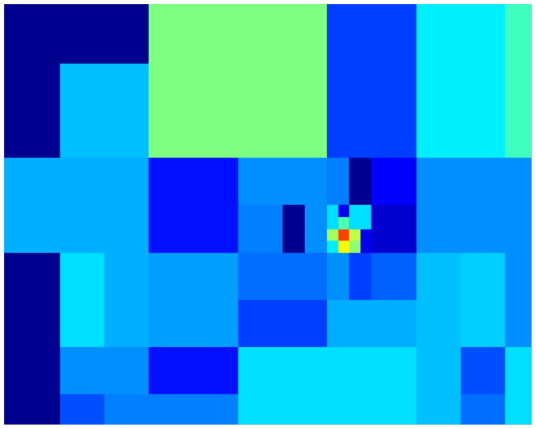
Detector score $S(x)$



BB for $\arg\max_x S(x)$

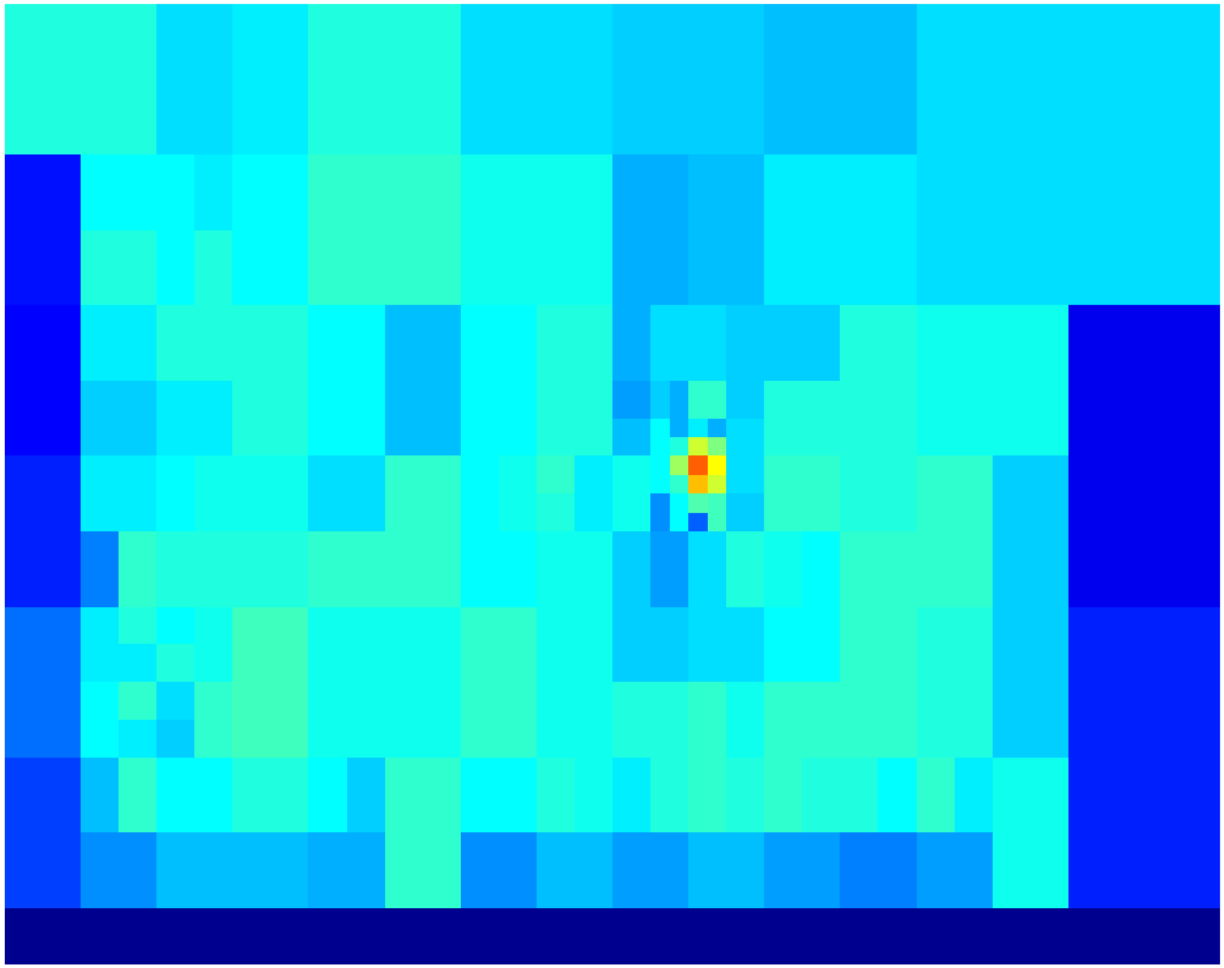


BB for $S(x) \geq -1$

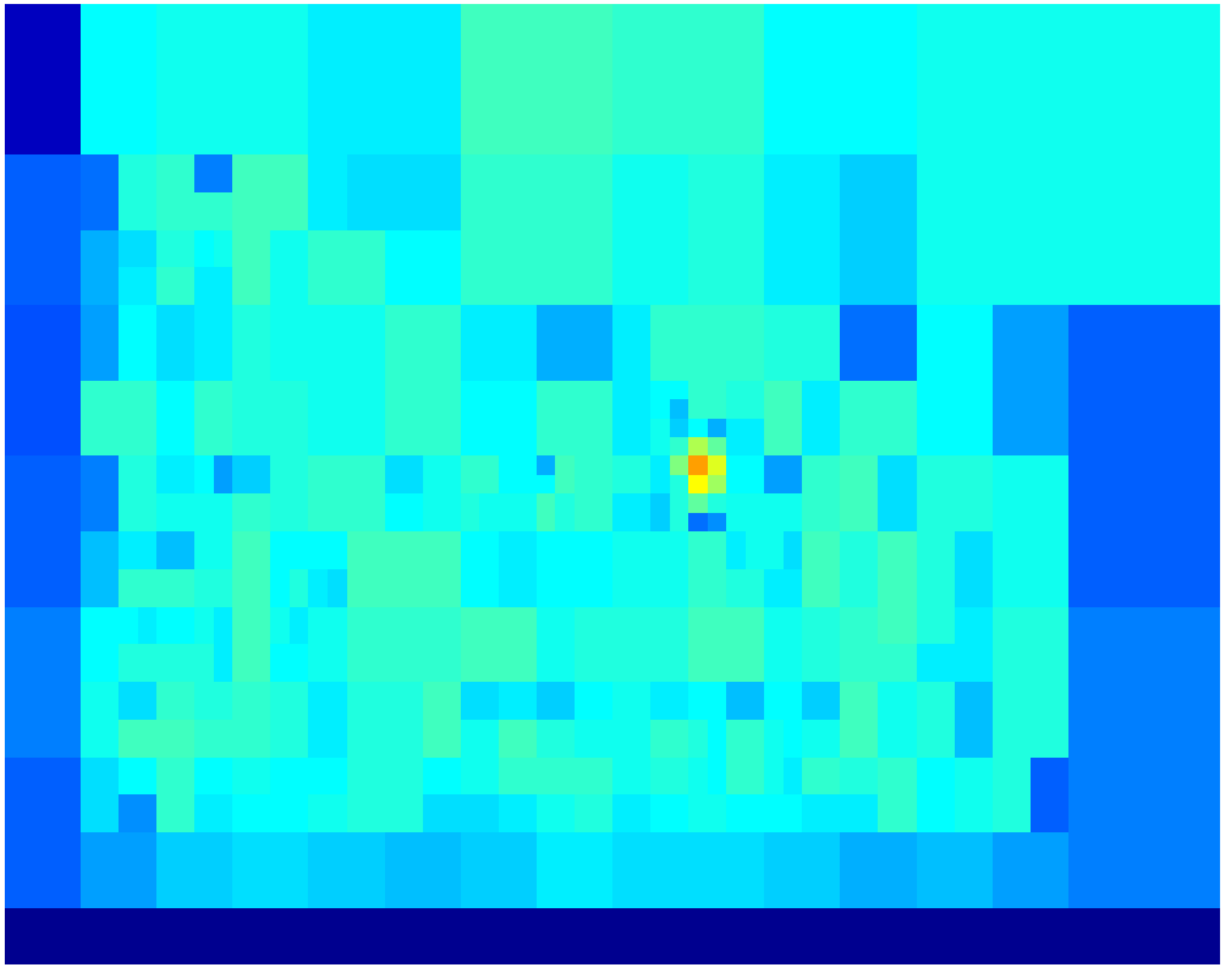


I. Kokkinos. Rapid Deformable Object Detection using DTBB, NIPS 2011

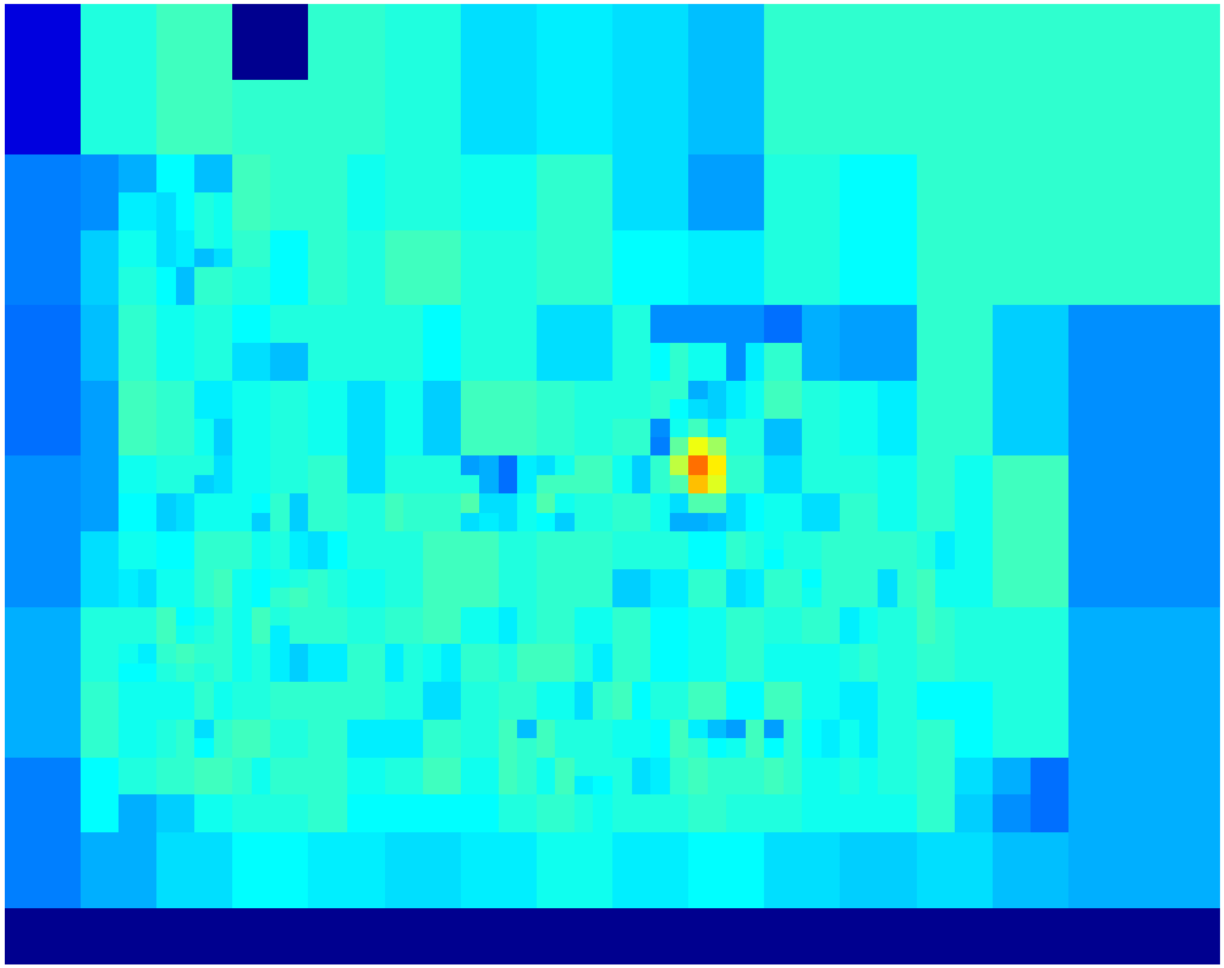
DTBB results: exact part scores



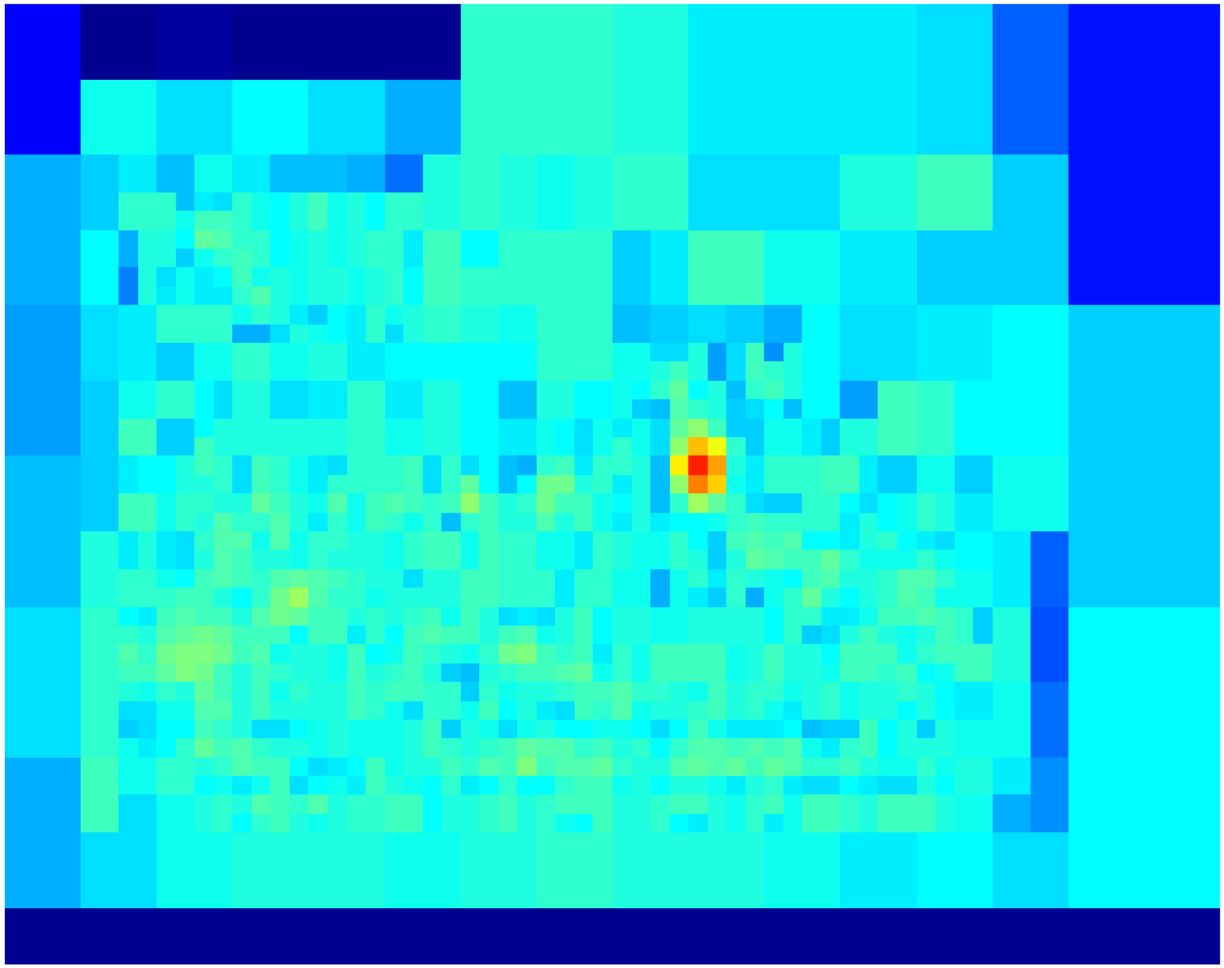
DTBB results, part score bounds @ $p_e = 0.2$



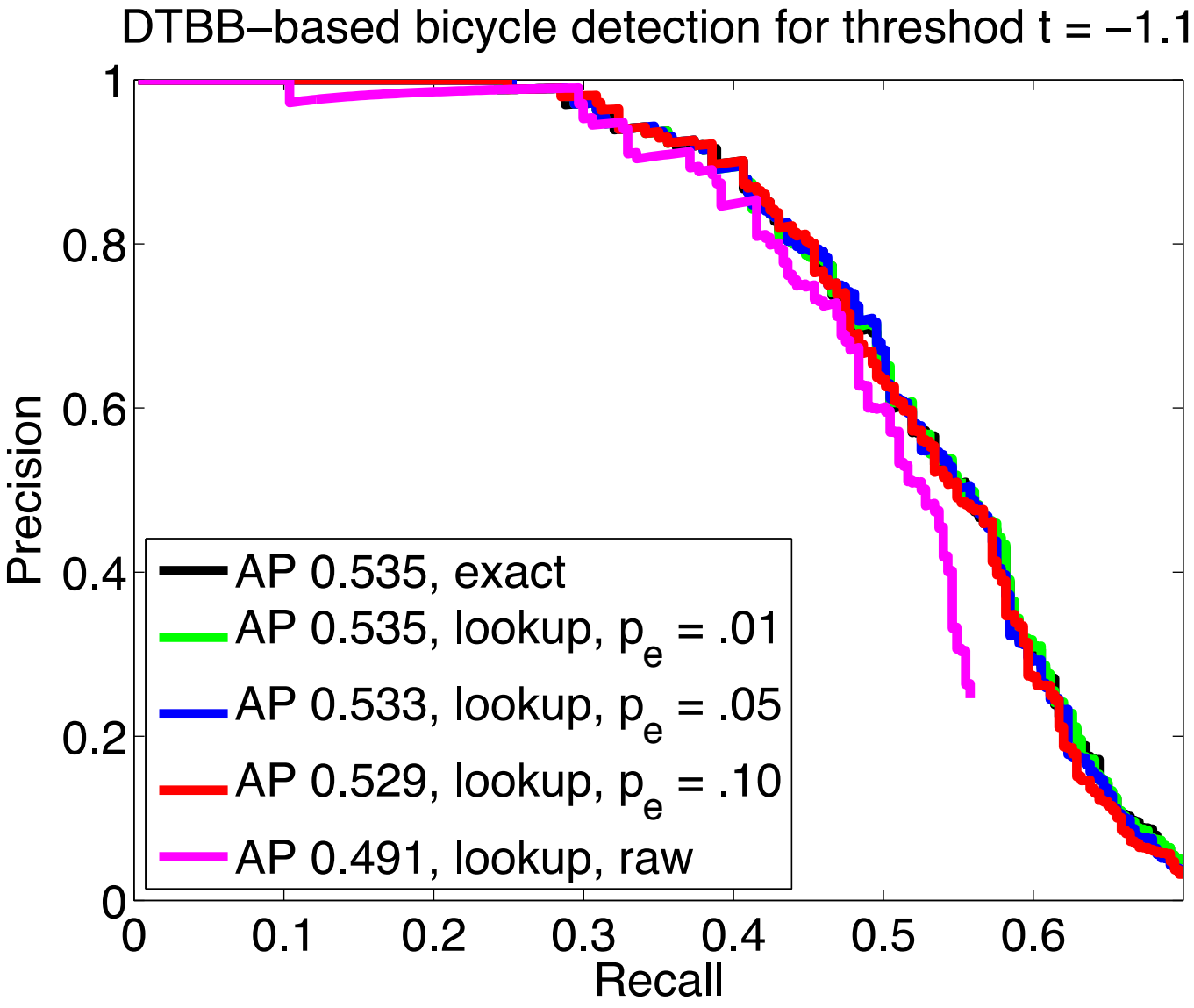
DTBB results, part score bounds @ $p_e = 0.1$



DTBB results, part score bounds @ $p_e = 0.05$



Impact on performance



Speedup results

	GDTs	DTBB	$p_e = 0.05$	$p_e = 0.01$
Part terms	8.35 ± 0.77	1.69 ± 0.18	0.69 ± 0.03	0.69 ± 0.06

Detection with Cascade DPMs (C-DPMs)

$$S_0(x) = 0,\ \mathcal{I}_0 = [1, N] \times [1, M]$$

$$S_k(x) = S_{k-1}(x) + \max_{x'} \left(U_p(x') + B_p(x', x)\right)$$

$$\mathcal{I}_k = \{x \in \mathcal{I}_{k-1} : S_{k-1}(x) \geq \theta_k\}$$

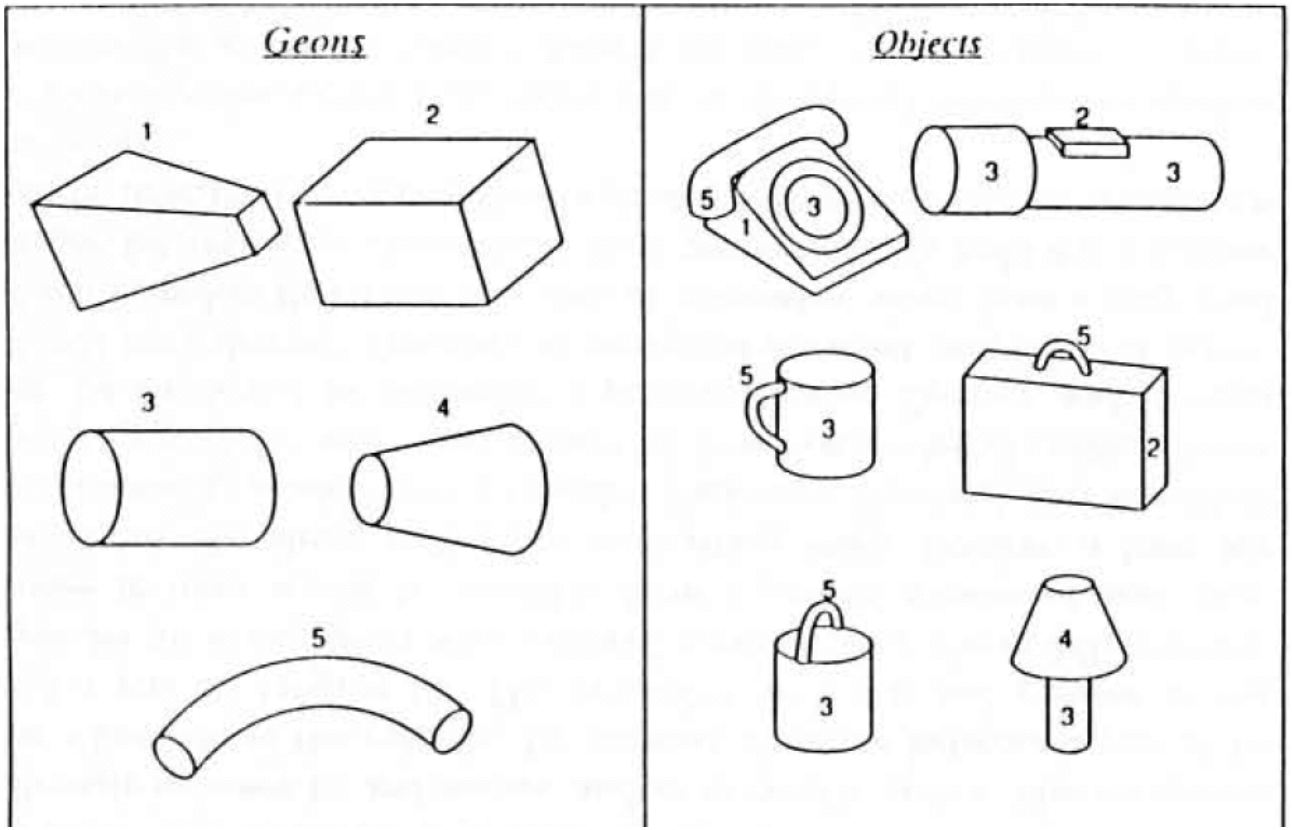
Felzenszwalb, Girschick, et al: use PCA-projection of $\mathbf{h}, \mathbf{w}$

Our work: use quick upper bounds, thresholds for full HOG

	GDTs	C-DPM	$p_e = 0.05$	$p_e = 0.01$
$\theta = -0.5$	8.95 ± 0.82	0.56 ± 0.07	0.19 ± 0.03	0.23 ± 0.04
$\theta = -0.7$	8.95 ± 0.82	0.72 ± 0.09	0.29 ± 0.04	0.36 ± 0.06
$\theta = -1.0$	8.95 ± 0.82	1.04 ± 0.16	0.51 ± 0.10	1.07 ± 0.29


Future work
~10,000 to 30,000

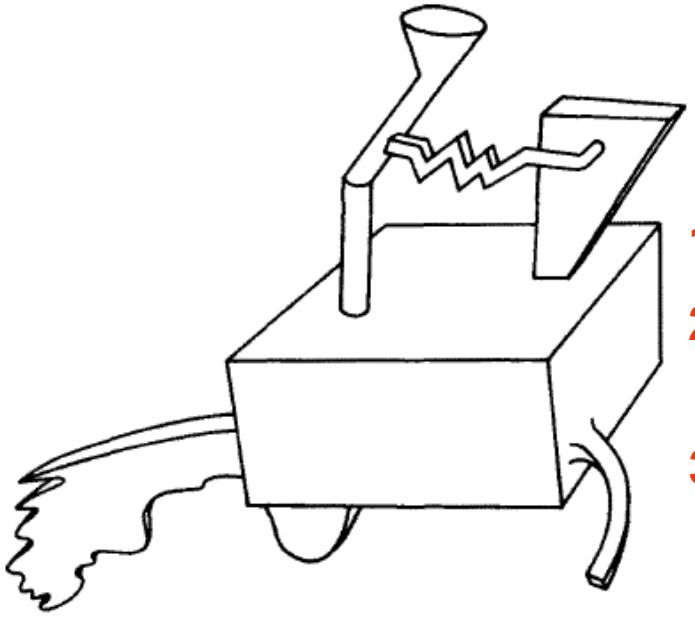
Scalability: objects and their geons



Hypothesis: there is a small number of geometric components that constitute the primitive elements of the object recognition system

Analogy: using letters to form words (compare with Ideograms)

Scalability: Recognition-by-components



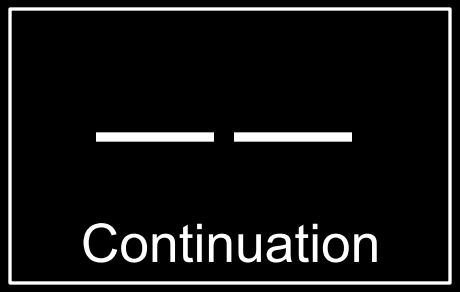
1) We know that this object is nothing we know
2) We can split this objects into parts that everybody will agree
3) We can see how it resembles something familiar: “a hot dog cart”

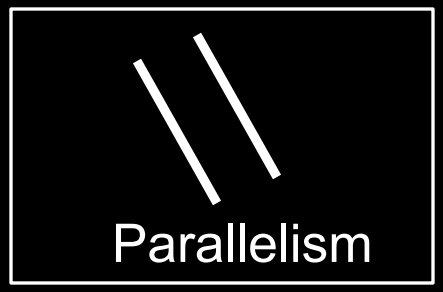
Mid-level vision

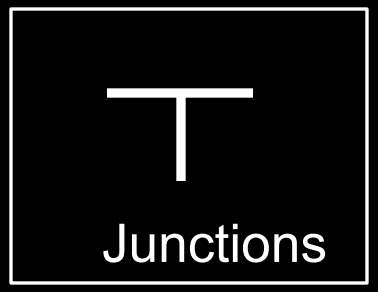
Object-like, but not an object

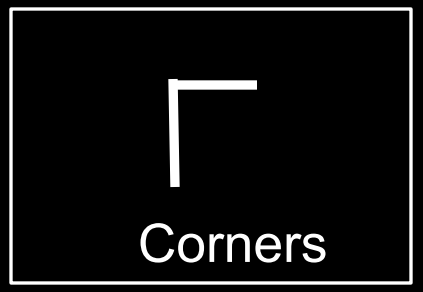
Mid-Level Representations

- Mid-level cues









"Tokens" from Vision by D.Marr:

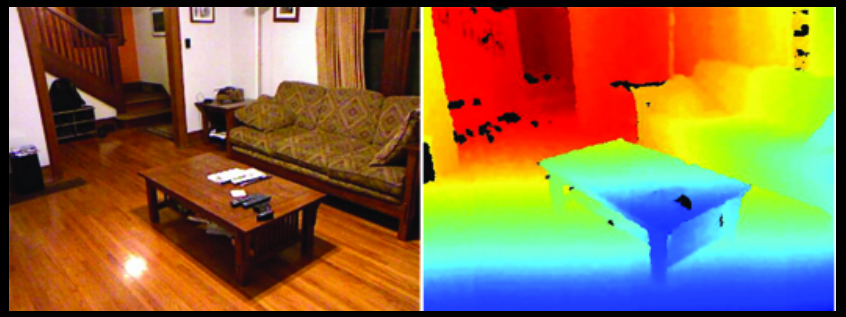
- Object parts:

- Difficult to hand-engineer → What about learning them?

Slide credit: R. Fergus

Why Learn Features?

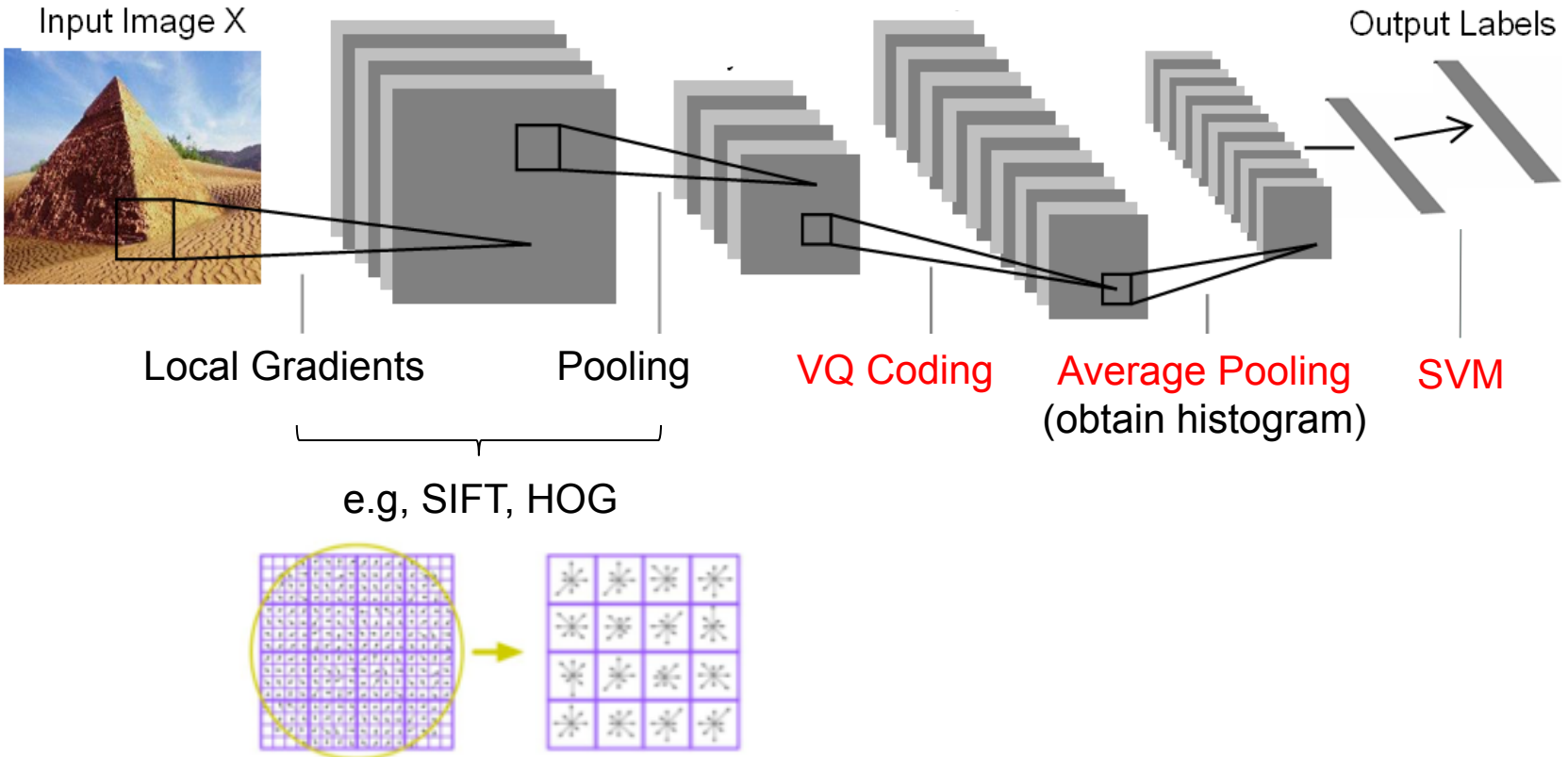
- Better performance
- Other domains (unclear how to hand engineer):
  - Kinect
  - Video
  - Multi spectral



- Feature computation time
  - Dozens of features now regularly used
  - Getting prohibitive for large datasets (10's sec /image)

Slide credit: R. Fergus

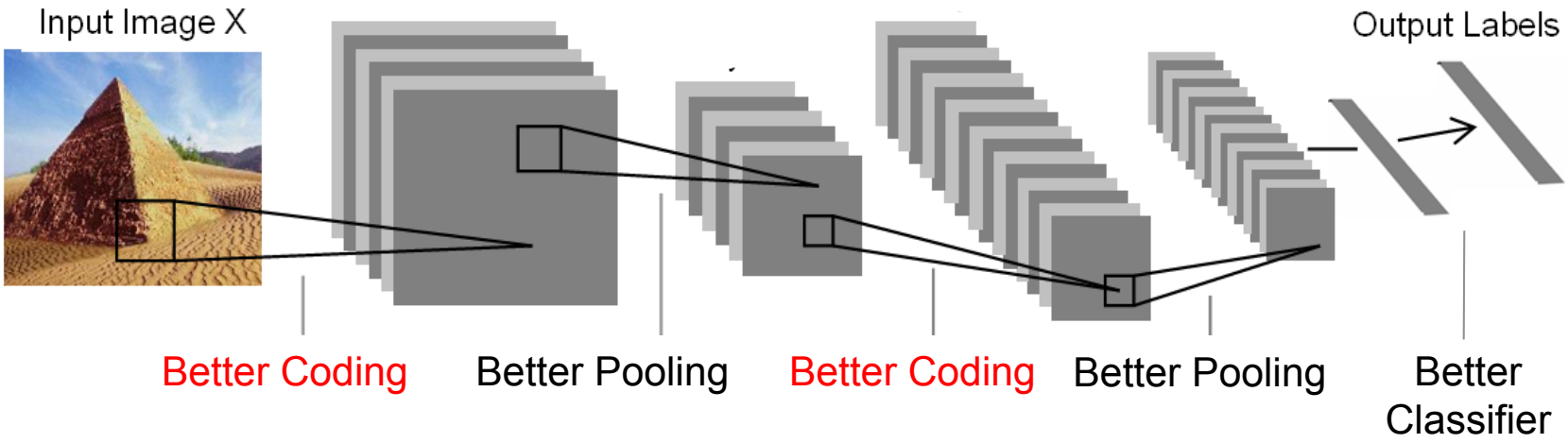
“BoW+SPM” has two coding+pooling layers



SIFT feature itself follows a coding+pooling operation

Slide credit: K. Yu

Develop better coding methods



- Coding: nonlinear mapping data into another feature space
- Better coding methods: sparse coding, RBMs, auto-encoders

Slide credit: K. Yu

What is sparse coding

Sparse coding (Olshausen & Field,1996). Originally developed to explain early visual processing in the brain (edge detection).

Training: given a set of random patches x, learning a dictionary of bases [$\Phi_1, \Phi_2, \ldots$]

Coding: for data vector x, solve LASSO to find the sparse coefficient vector a

$$\min_{a,\phi} \sum_{i=1}^{m} \left\| x_i - \sum_{j=1}^{k} a_{i,j}\phi_j \right\|^2 + \lambda \sum_{i=1}^{m} \sum_{j=1}^{k} |a_{i,j}|$$

Slide credit: K. Yu

Sparse coding: training time

Input: Images $x_1, x_2, \ldots, x_m$ (each in R^d)
Learn: Dictionary of bases $\phi_1, \phi_2, \ldots, \phi_k$ (also R^d).

$$\min_{a,\phi} \sum_{i=1}^{m} \left\| x_i - \sum_{j=1}^{k} a_{i,j}\phi_j \right\|^2 + \lambda \sum_{i=1}^{m}\sum_{j=1}^{k} |a_{i,j}|$$

Alternating optimization:

1. Fix dictionary $\phi_1, \phi_2, \ldots, \phi_k$, optimize a (a standard LASSO problem)

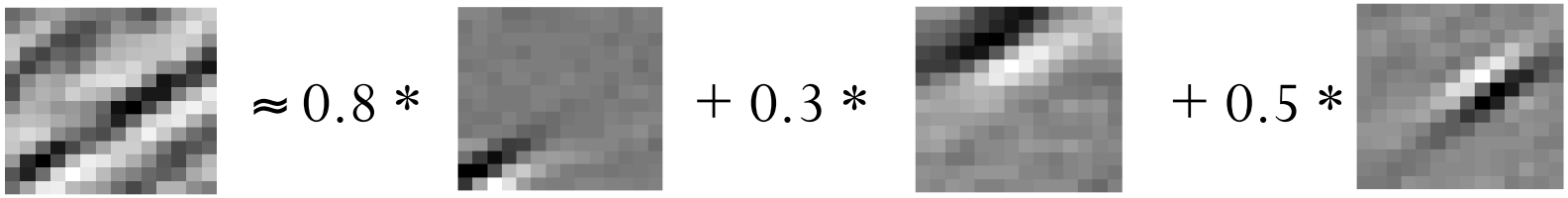
2. Fix activations a, optimize dictionary $\phi_1, \phi_2, \ldots, \phi_k$ (a convex QP problem)

Slide credit: K. Yu

Sparse coding: testing time

Input: Novel image patch x (in R^d) and previously learned ϕ_i's
Output: Representation $[a_{i,1}, a_{i,2}, \ldots, a_{i,\kappa}]$ of image patch x_i.

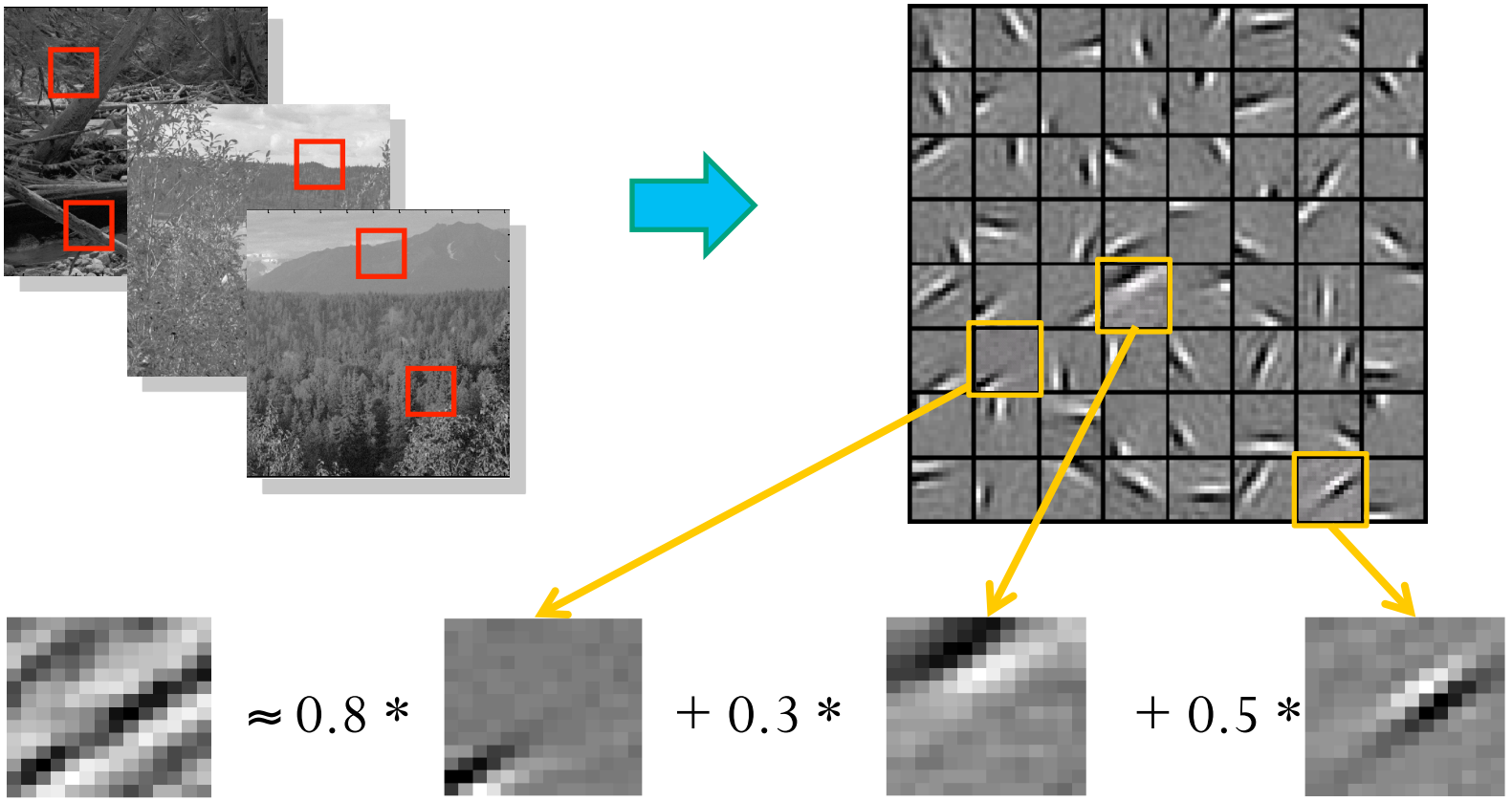
$$\min_a \sum_{i=1}^{m} \left\| x_i - \sum_{j=1}^{k} a_{i,j}\phi_j \right\|^2 + \lambda \sum_{i=1}^{m} \sum_{j=1}^{k} |a_{i,j}|$$



Represent x_i as: a_i = [0, 0, …, 0, **0.8**, 0, …, 0, **0.3**, 0, …, 0, **0.5**, …]

Slide credit: A. Ng

Sparse coding illustration



$[a_1, \ldots, a_{64}] = [0, 0, \ldots, 0, \mathbf{0.8}, 0, \ldots, 0, \mathbf{0.3}, 0, \ldots, 0, \mathbf{0.5}, 0]$
(feature representation)

Slide credit: A. Ng

Compact & easily interpretable

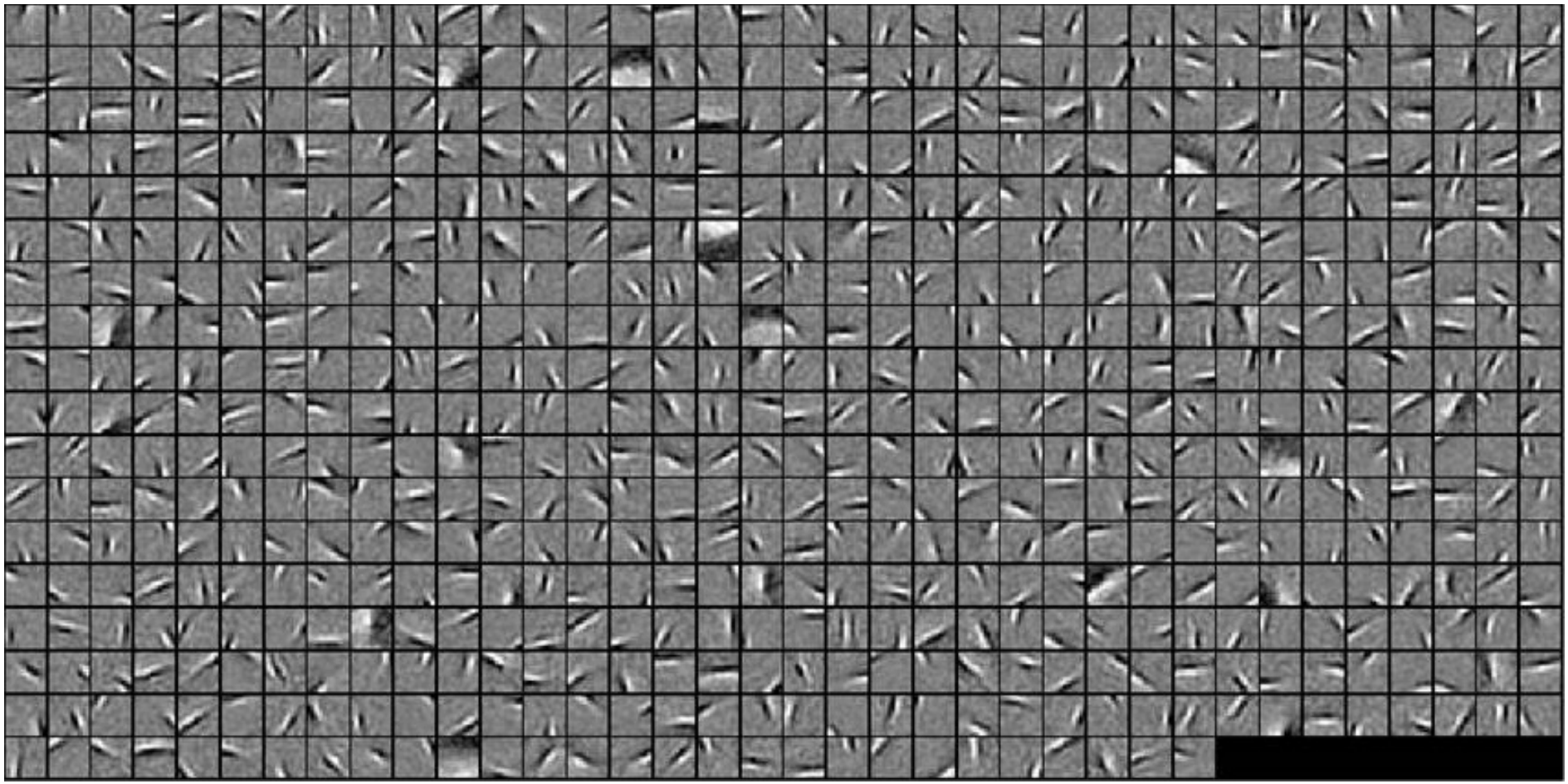
More examples

Represent as: [0, 0, ..., 0, 0.6, 0, ..., 0, 0.8, 0, ..., 0, 0.4, ...]

Represent as: [0, 0, ..., 0, 1.3, 0, ..., 0, 0.9, 0, ..., 0, 0.3, ...]

Slide credit: A. Ng

Sparse coding basis



Slide credit: A. Ng

Recap of sparse coding for feature learning

Training time

Input: Images $x^{(1)}, x^{(2)}, \ldots, x^{(m)}$ (each in $R^{n \times n}$)
Learn: Dictionary of bases $\phi_1, \phi_2, \ldots, \phi_k$ (also $R^{n \times n}$).

$$\min_{a,\phi} \sum_{i=1}^{m} \left(\left\| x^{(i)} - \sum_{j=1}^{k} a_j^{(i)} \phi_j \right\|^2 - \lambda \sum_{j=1}^{k} |a_j^{(i)}| \right)$$

Test time

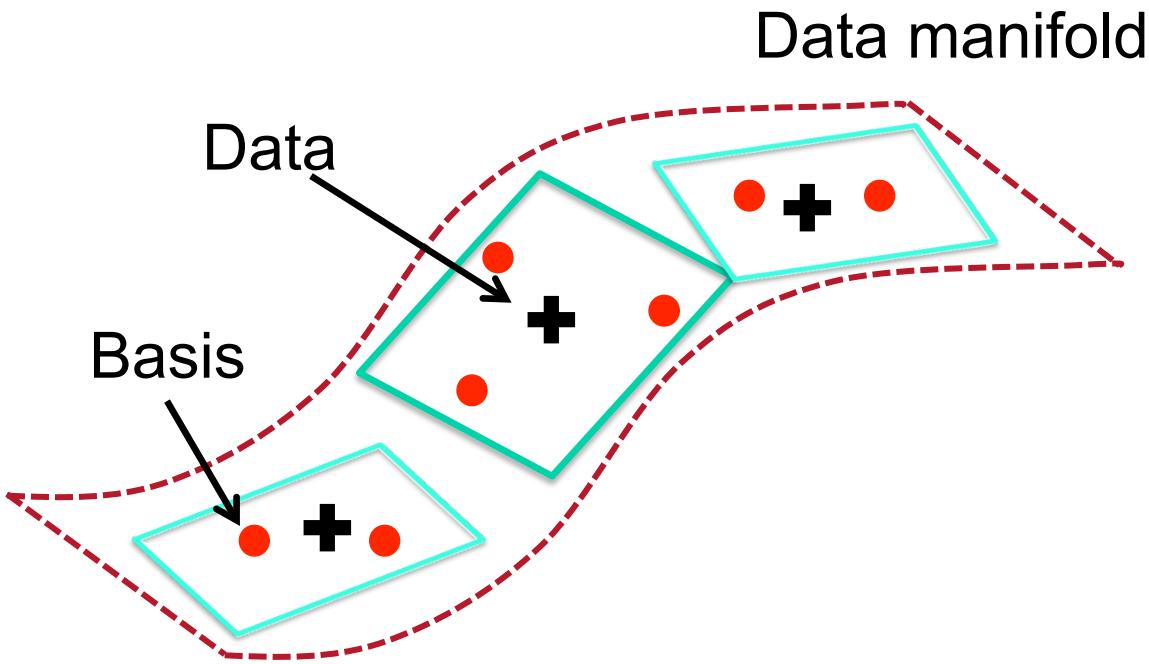
Input: Novel image x (in $R^{n \times n}$) and previously learned ϕ_i's.
Output: Representation $[a_1, a_2, \ldots, a_k]$ of image x.

$$\min_{a} \left\| x - \sum_{j=1}^{k} a_j \phi_j \right\|^2 + \lambda \sum_{j=1}^{k} |a_j|$$

$x \approx 0.8 * \phi_{36} + 0.3 * \phi_{42} + 0.5 * \phi_{63}$

Represent as: [0, 0, …, 0, **0.8**, 0, …, 0, **0.3**, 0, …, 0, **0.5**, …]

A geometric view to sparse coding



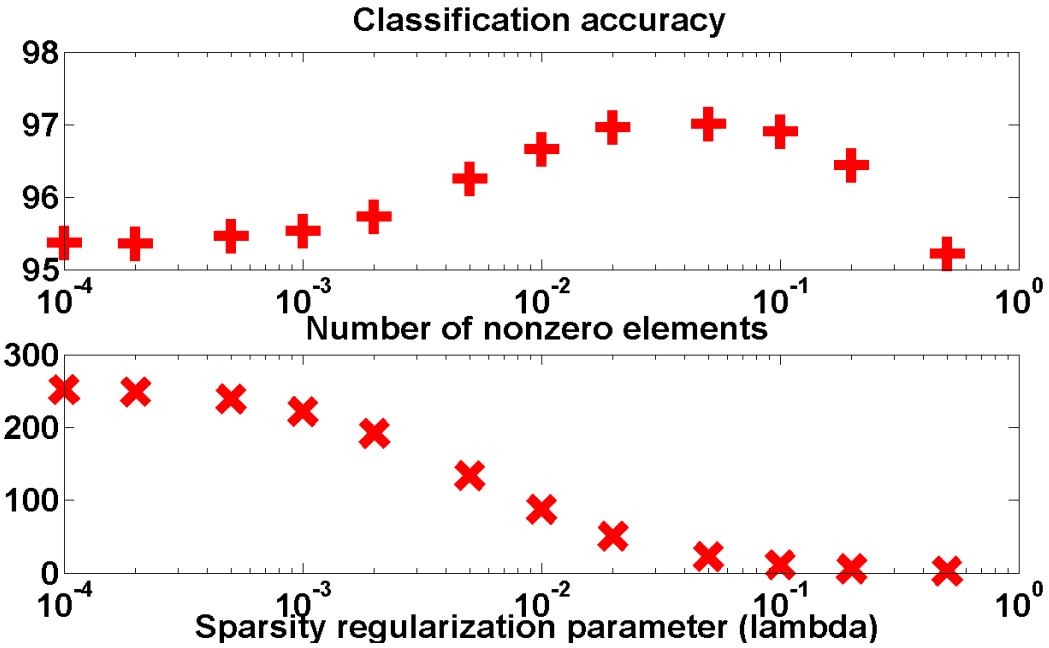
- Each basis is somewhat like a pseudo data point – "anchor point"
- Sparsity: each datum is a sparse combination of neighbor anchors.
- The coding scheme explores the manifold structure of data.

Slide credit: K. Yu

MNIST Experiment: Classification using SC

$$\min_{a,\phi} \sum_{i=1}^{m} \left\| x_i - \sum_{j=1}^{k} a_{i,j}\phi_j \right\|^2 + \lambda \sum_{i=1}^{m} \sum_{j=1}^{k} |a_{i,j}|$$

Try different values

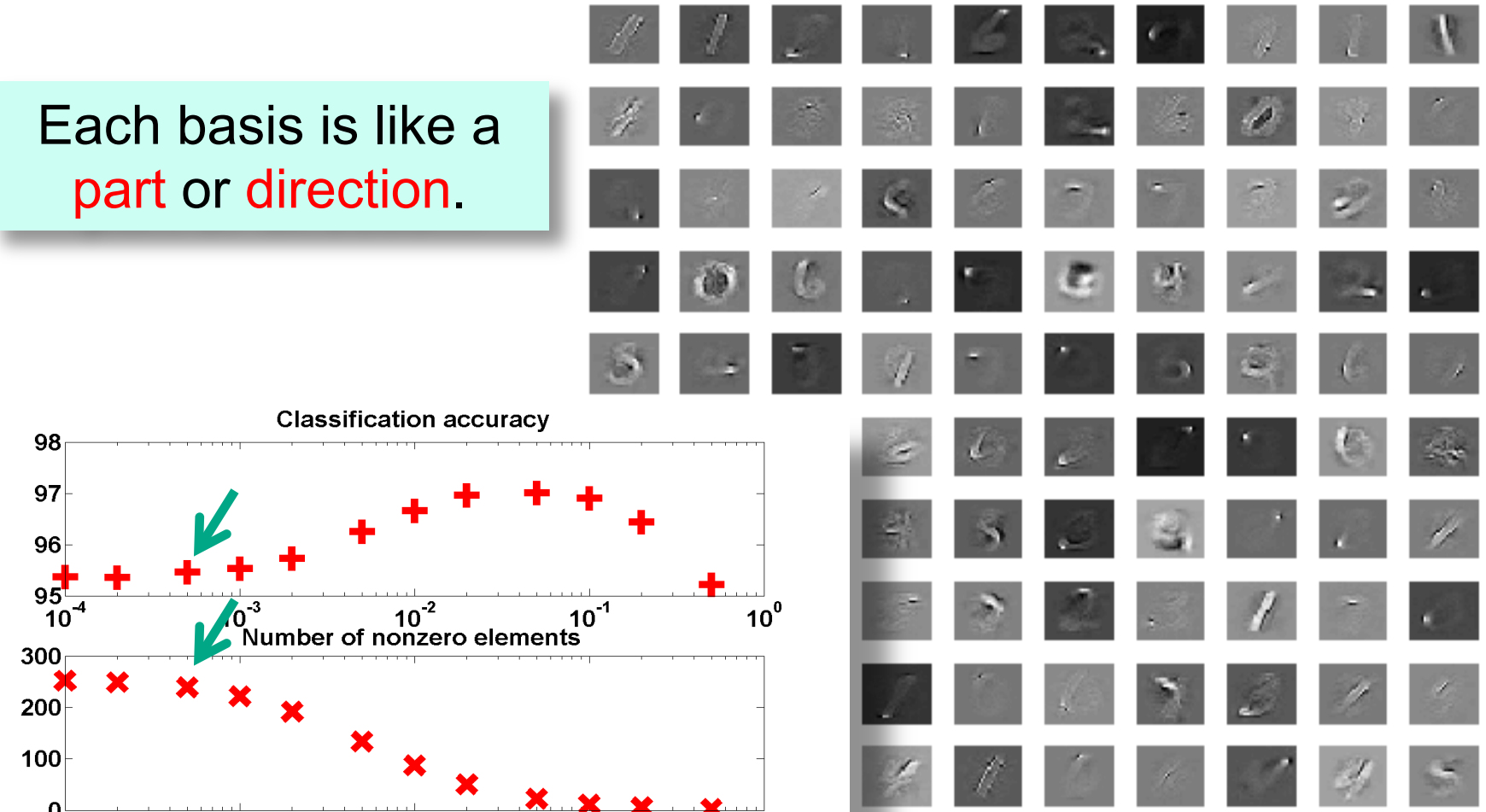


- 60K training, 10K for test
- Let k=512
- Linear SVM on sparse codes

Slide credit: K. Yu

MNIST Experiment: Lambda = 0.0005

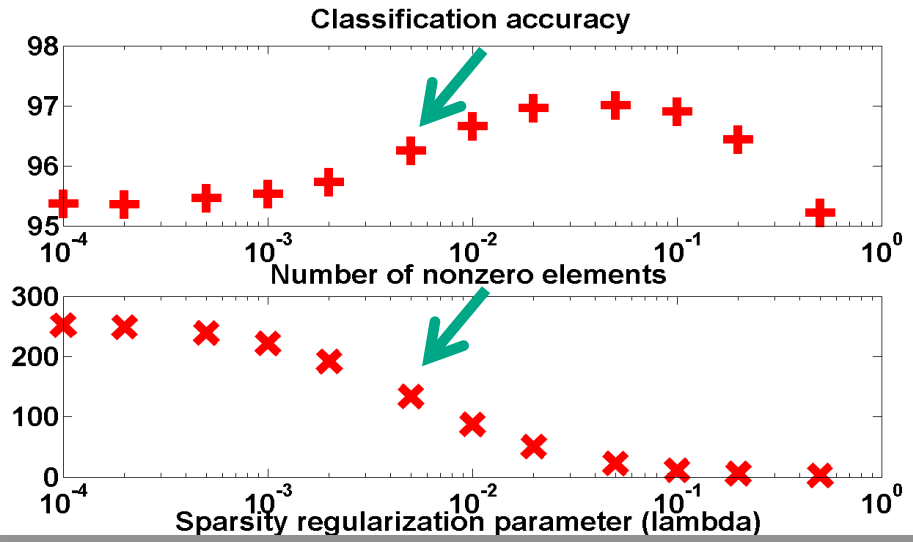
Each basis is like a part or direction.

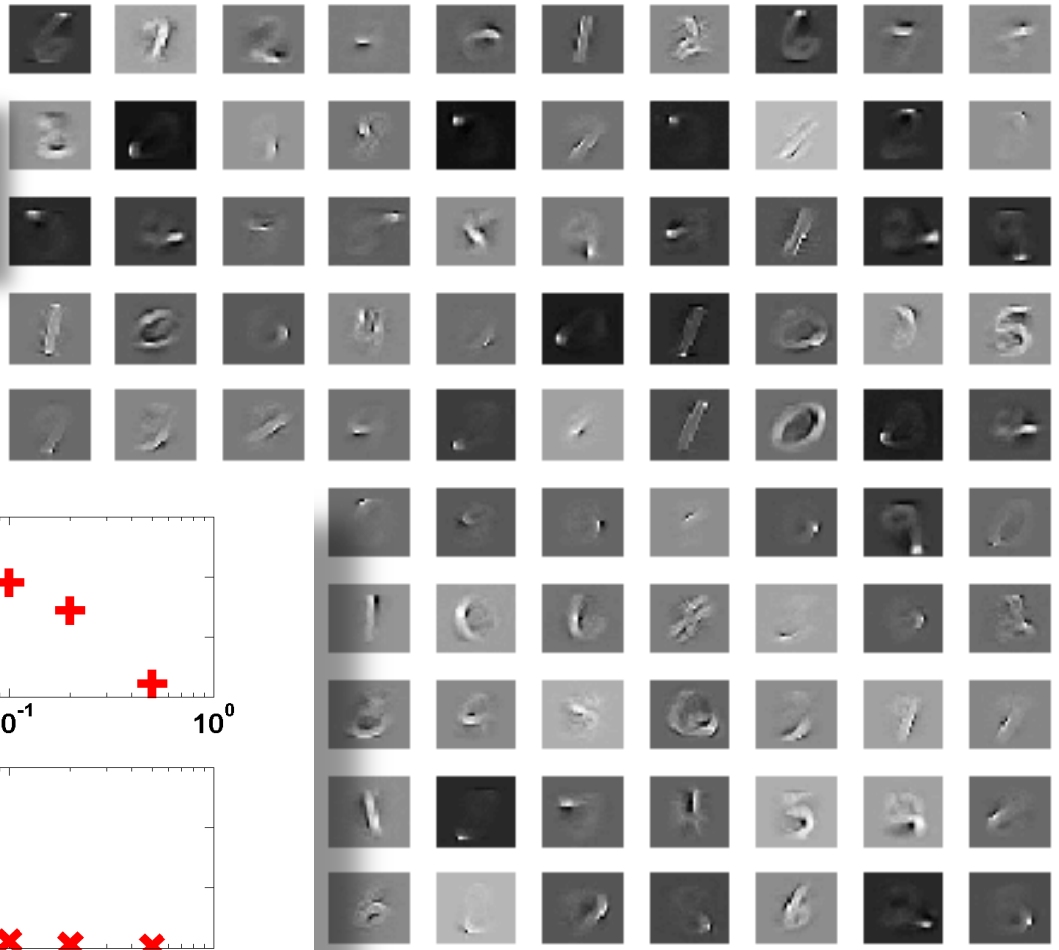


Slide credit: K. Yu

MNIST Experiment: Lambda = 0.005

Again, each basis is like a part or direction.

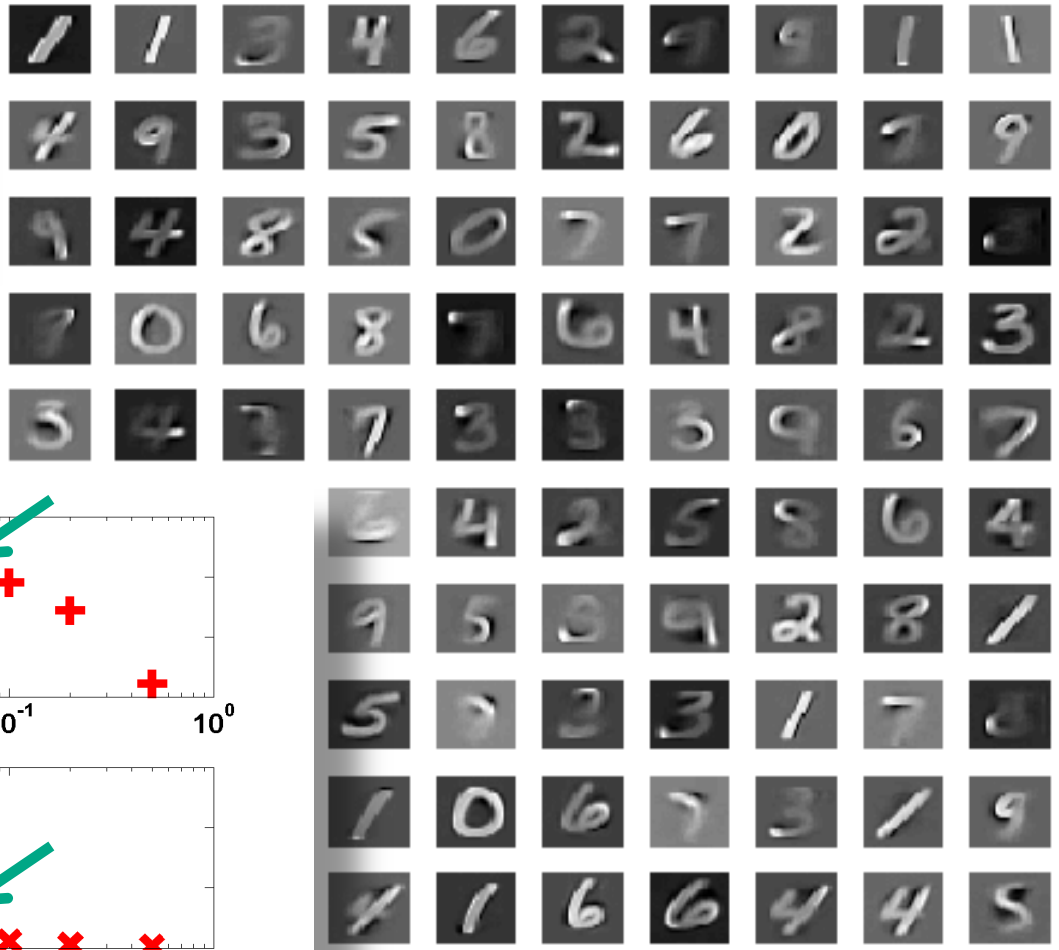


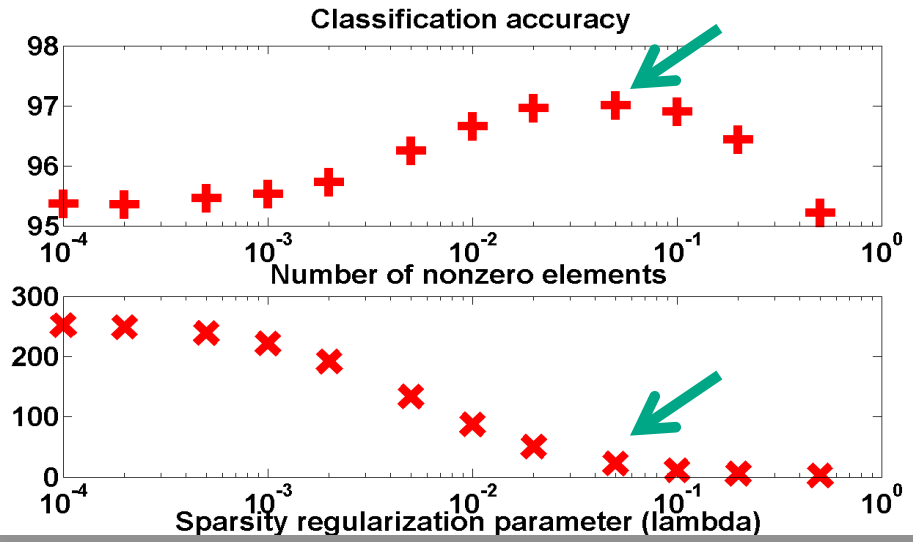


Slide credit: K. Yu

MNIST Experiment: Lambda = 0.05

Now, each basis is more like a digit !

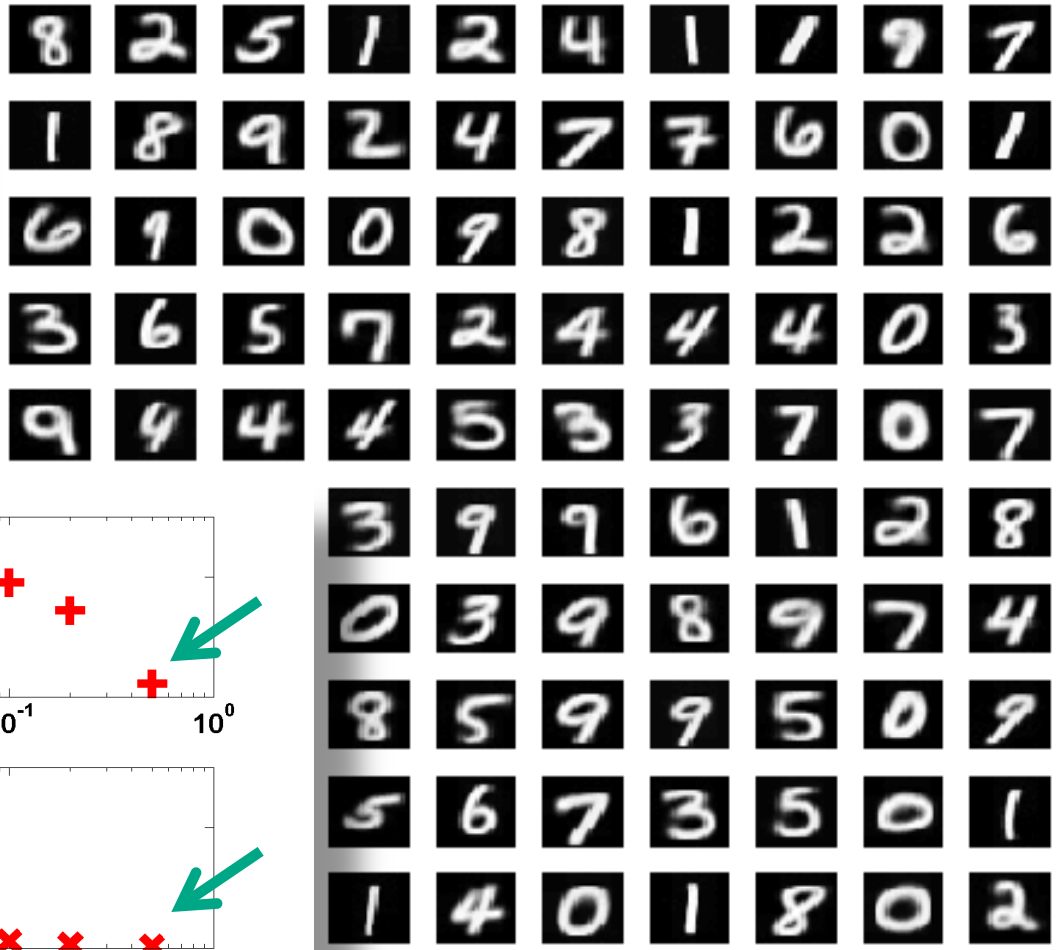


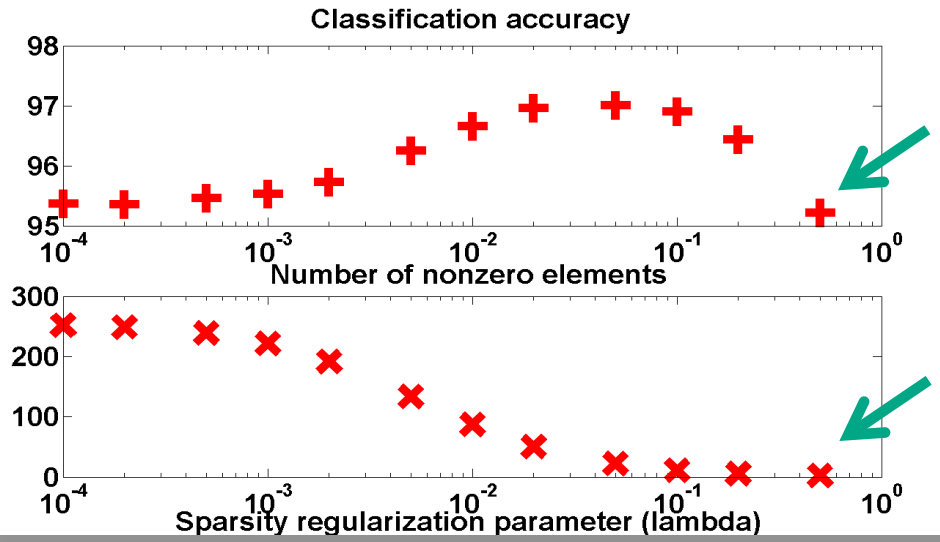


Slide credit: K. Yu

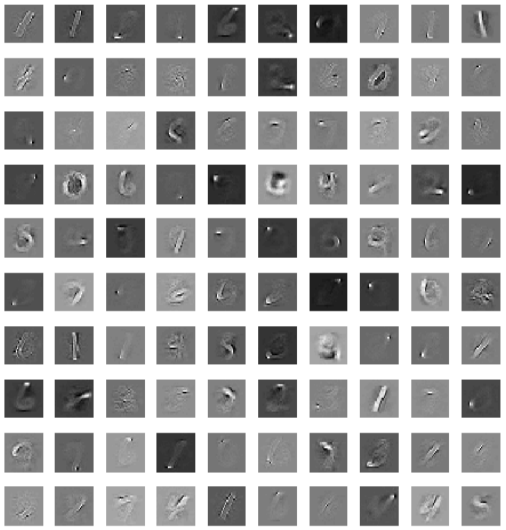
MNIST Experiment: Lambda = 0.5

Like clustering now!

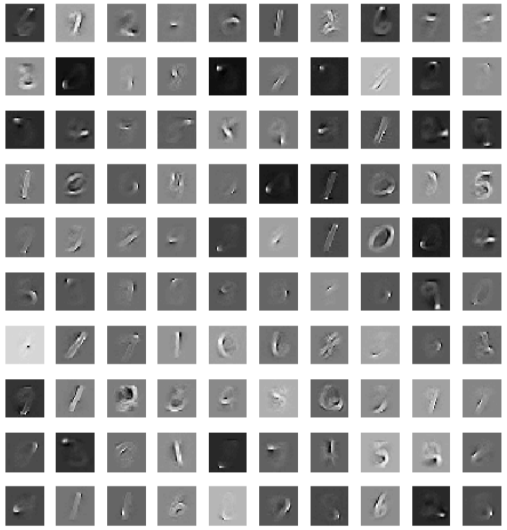




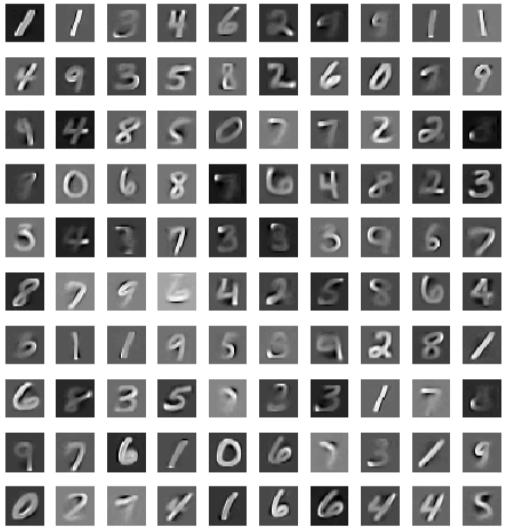
Geometric view of sparse coding



Error: 4.54%



Error: 3.75%



Error: 2.64%

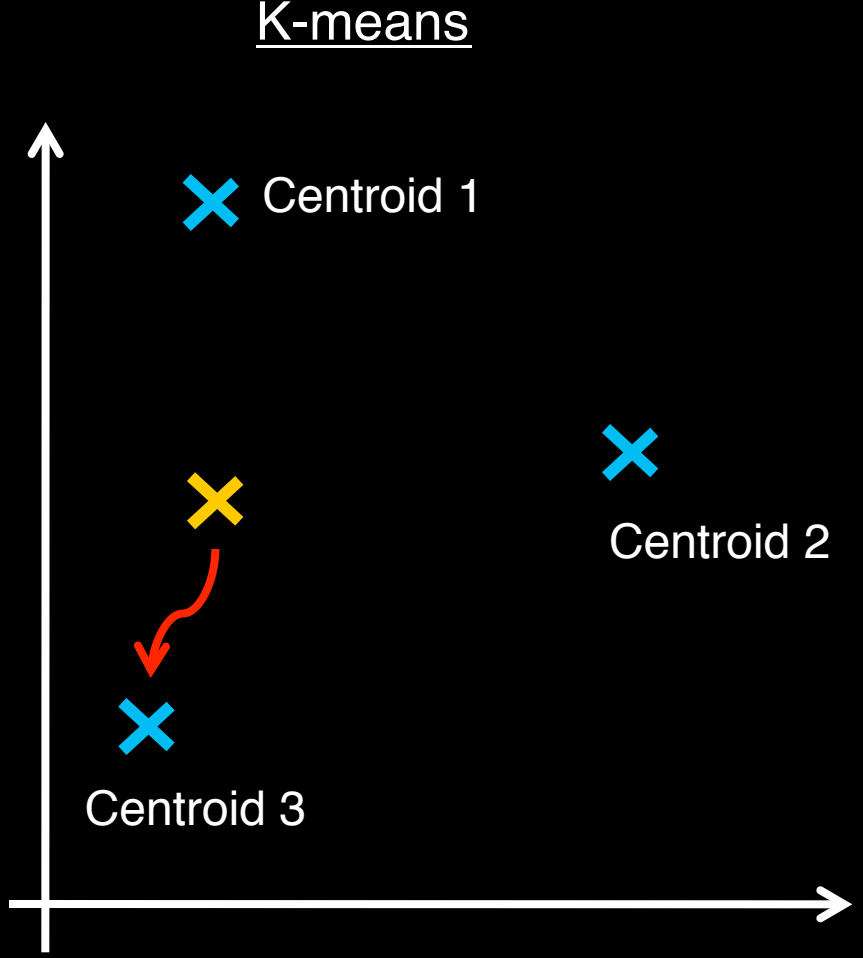
- When SC achieves the best classification accuracy, the learned bases are like digits – each basis has a clear local class association.

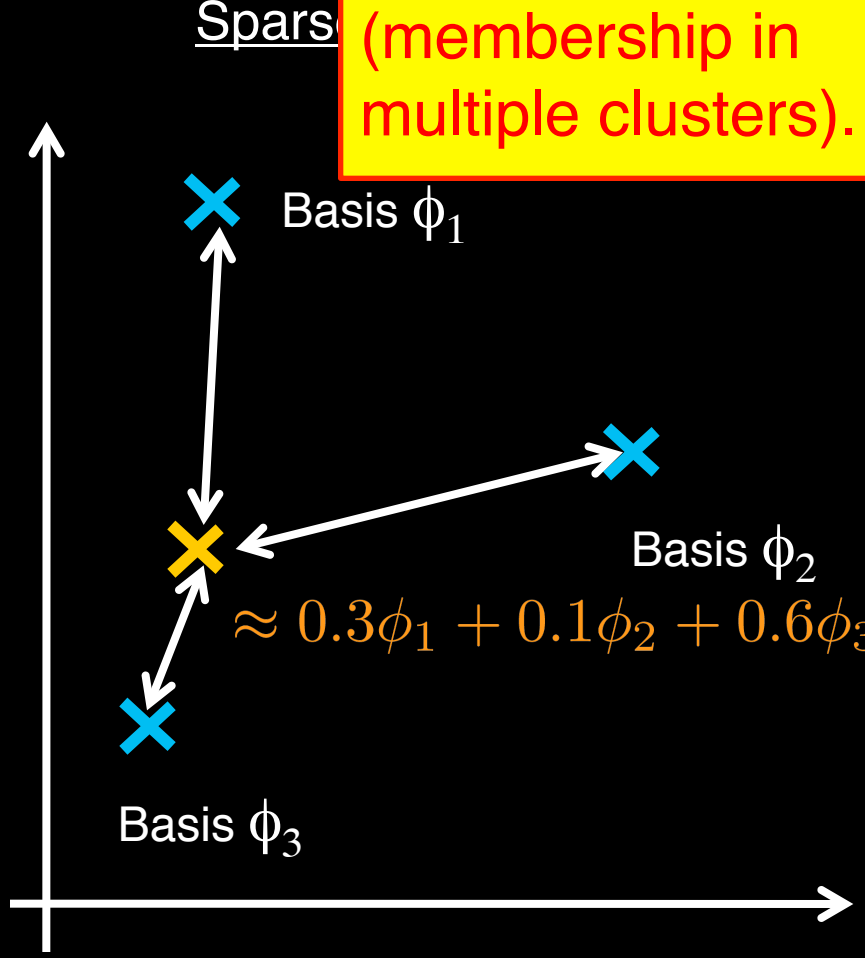
- Implication: exploring data geometry may be useful for classification.

Slide credit: K. Yu

K-means vs. sparse codin

Intuition: "Soft" version of k-means (membership in multiple clusters).





Represent as: $\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$

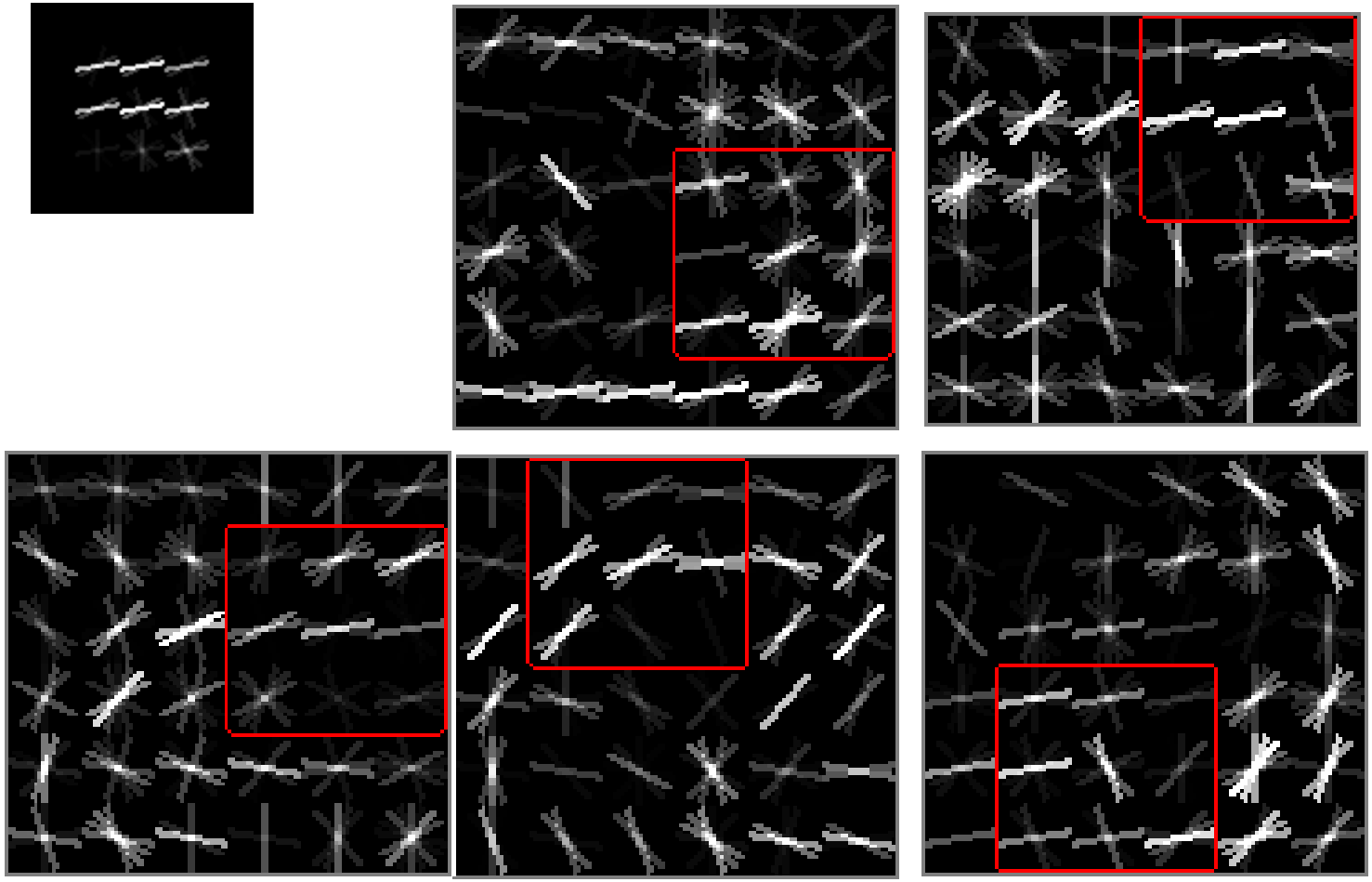
Represent as: $\begin{bmatrix} 0.3 \\ 0.1 \\ 0.6 \end{bmatrix}$

Slide credit: A. Ng

Andrew Ng

Dealing with translations

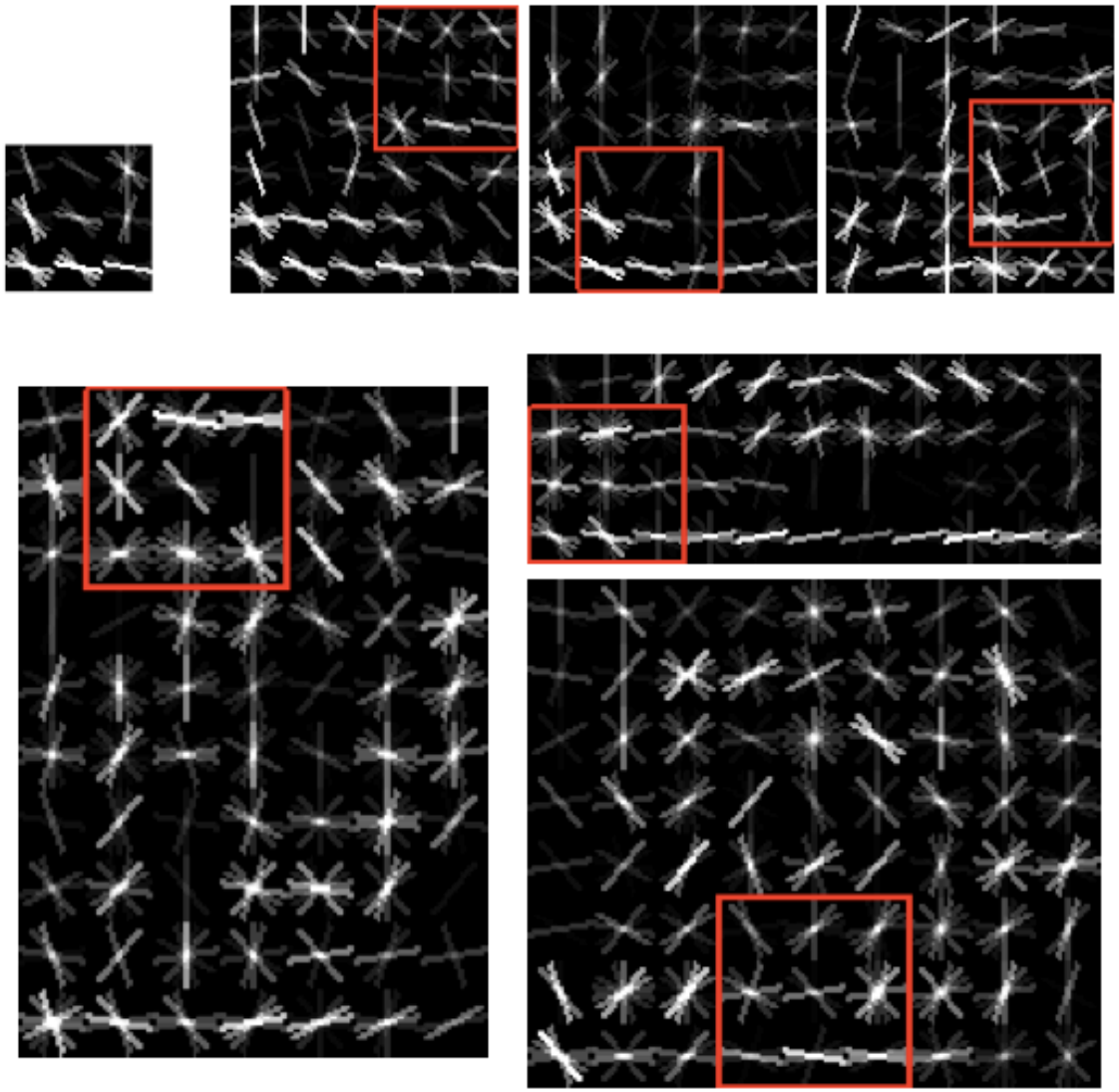
Motivation



I.K. Shufflets: Shared mid level parts for fast detection, ICCV 2013

Dealing with translations

Motivation



I.K. Shufflets: Shared mid level parts for fast detection, ICCV 2013

Sparse Coding

$$(P1) \qquad \min_{\mathbf{b},\alpha_1 \ldots \alpha_P} \sum_{p=1}^{P} \|\mathbf{w}_p - \mathbf{b}\alpha_p\|_2$$

$$\text{s.t.} \qquad \|\alpha_p\|_0 \leq L, \quad p = 1, \ldots P$$

$$\|\mathbf{b}_b\|_2 \leq 1, \quad b = 1, \ldots B$$

Notice the l0 norm

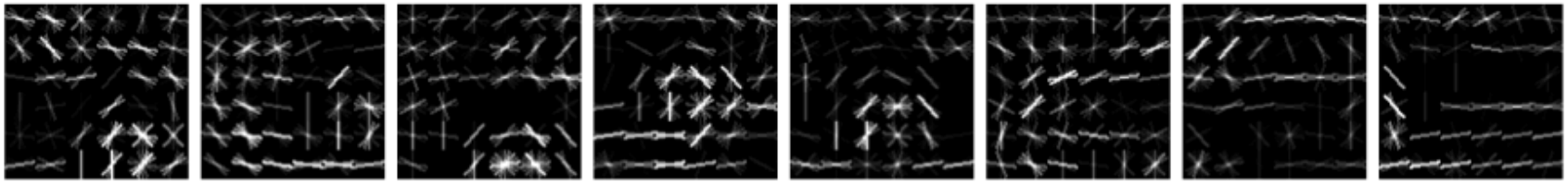
Olshausen and Field, Sparse coding with an overcomplete basis, Vision Research 1997
Mallat and Zhang, 'Matching Pursuits with time-frequency dictionaries, IEEE Trans. Sig. Proc, 1993

Shift Invariant Sparse Coding

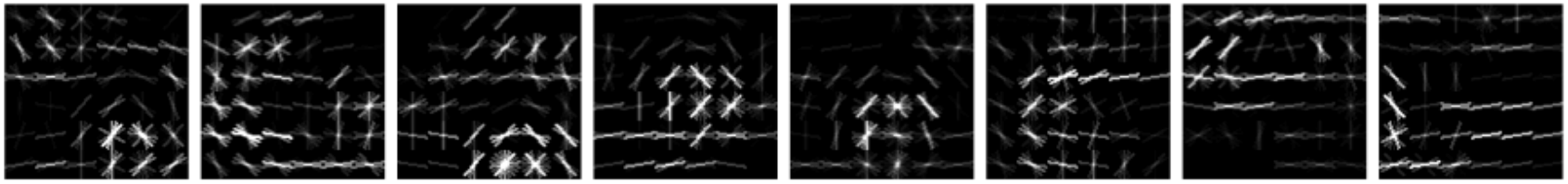
$$
\begin{aligned}
(P2) \qquad & \min_{\mathbf{k},\alpha_1 \ldots \alpha_P} \sum_{p=1}^{P} \|\mathbf{w}_p - \mathbf{b}\alpha_p\|_2 \\
\text{s.t.} \qquad & \|\alpha_p\|_0 \leq L, \quad p = 1, \ldots, P \\
& \|\mathbf{k}_i\|_2 \leq 1, \quad i = 1, \ldots, K \\
& \mathbf{b}_b = \mathbf{k}_b^{v_b, h_b}, \quad b = 1, \ldots B
\end{aligned}
$$

Grosse et al. Shift-Invariant Sparse Coding for Audio Classification UAI, 2009

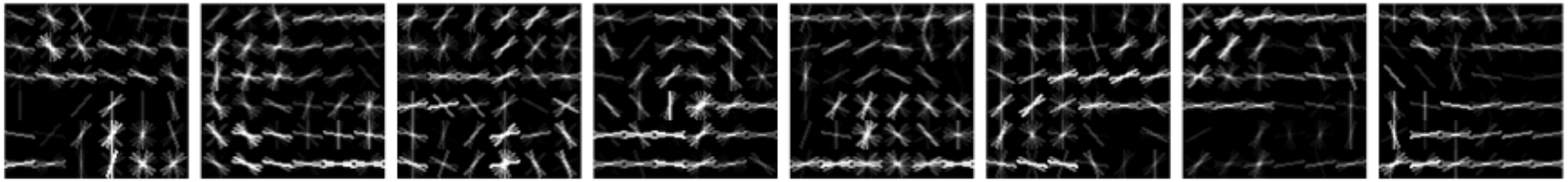
Shufflet-based reconstructions (SISC vs SC)



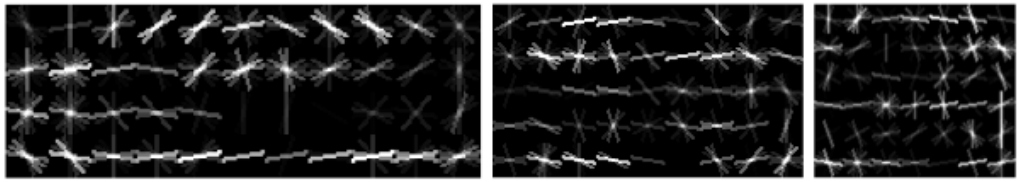
Original part filters for the first component of class 'car'



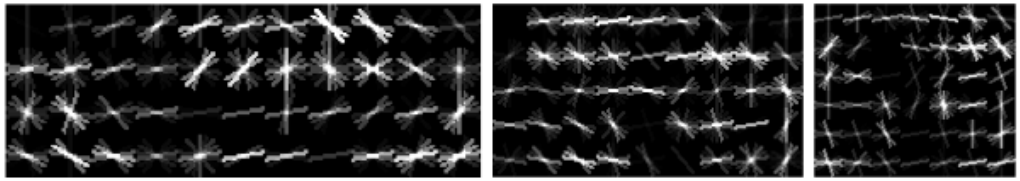
SISC-based reconstructions



Sparse Coding-based reconstructions



Original root filters for components 1, 3, 5 of class 'car'



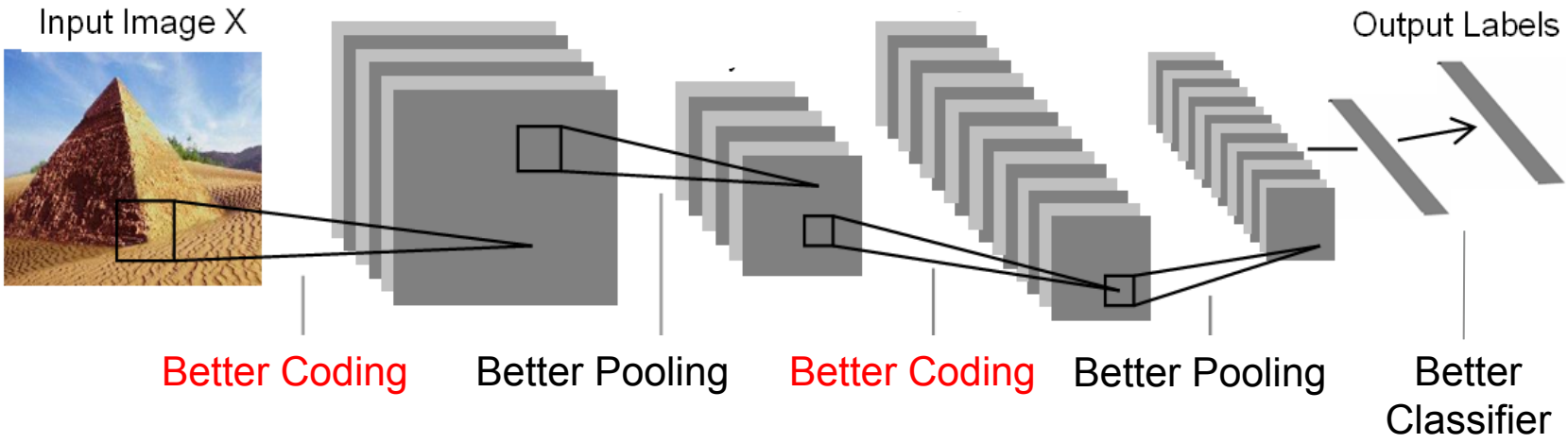
SISC-based approximations

Timing results

	Baseline	DTBB	Lookup-$p_e = 0.5$	Lookup-$p_e = 0.1$	Shufflets-$p_e = 0.5$	Shufflets $p_e = 0.1$
Part terms	2.26 ± 0.77	1.69 ± 0.18	0.69 ± 0.03	0.69 ± 0.06	0.28 ± 0.05	0.28 ± 0.05
$\theta = -0.5$	0.41 ± 0.06	0.21 ± 0.06	0.47 ± 0.11	1.04 ± 0.25	0.54 ± 0.22	0.64 ± 0.22
Sum	2.67 ± 0.83	1.90 ± 0.23	1.17 ± 0.12	1.74 ± 0.32	0.83 ± 0.27	0.93 ± 0.27
$\theta = -0.7$	0.41 ± 0.06	0.42 ± 0.10	1.00 ± 0.23	2.33 ± 0.65	0.74 ± 0.25	0.83 ± 0.26
Sum	2.67 ± 0.83	2.10 ± 0.24	1.70 ± 0.27	3.00 ± 0.71	1.02 ± 0.30	1.13 ± 0.31
$\theta = -1.0$	0.41 ± 0.06	1.31 ± 0.31	3.80 ± 0.90	9.40 ± 2.70	0.87 ± 0.24	1.01 ± 0.27
Sum	2.67 ± 0.83	3.00 ± 0.42	4.50 ± 1.02	10.01 ± 2.82	1.16 ± 0.29	1.29 ± 0.52

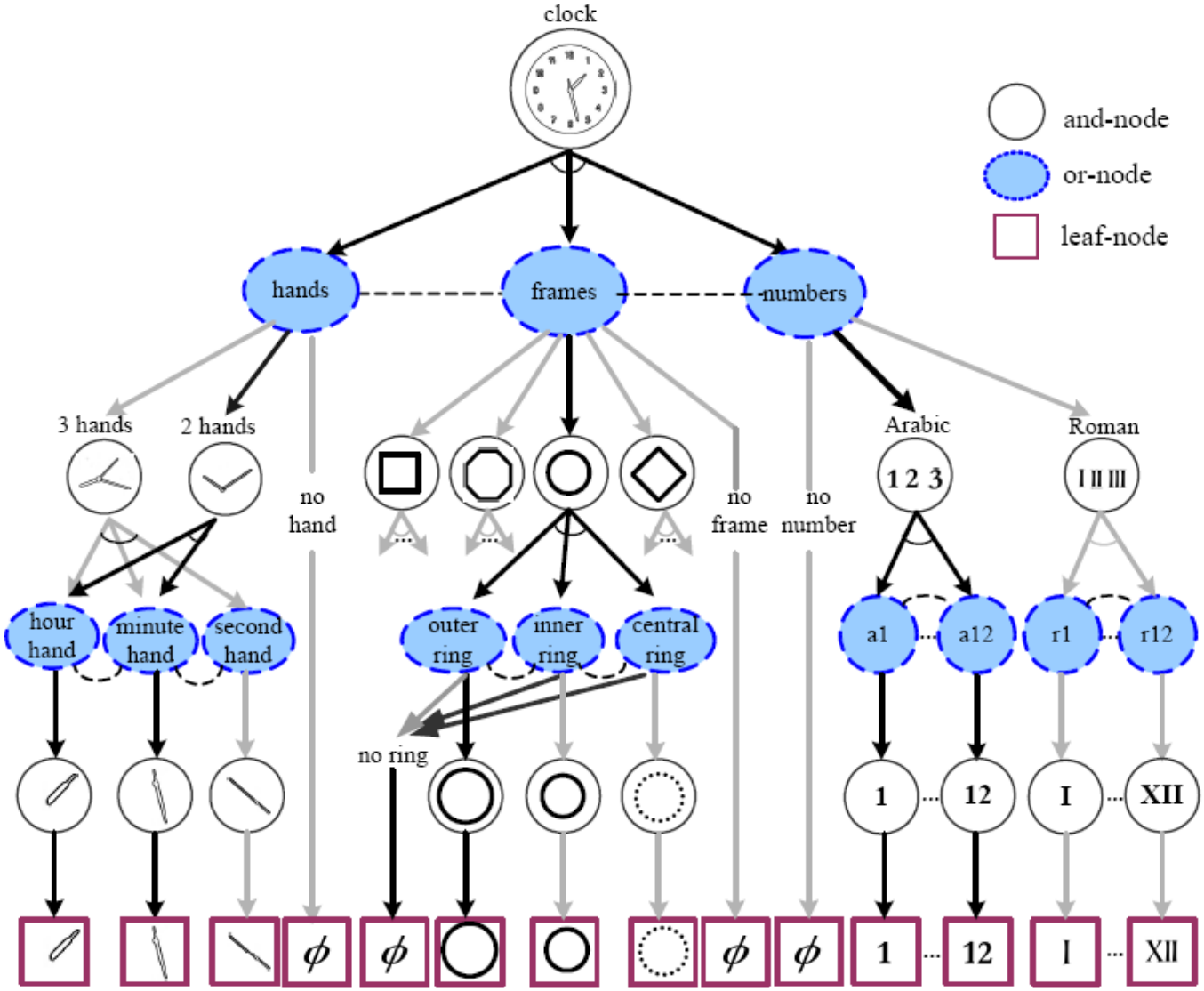
	Baseline	Cascade-DPM	Shufflet cascade, $p_e = 0.1$	Shufflet cascade, $p_e = 0.5$
$\theta = -0.5$	2.67 ± 0.82	0.56 ± 0.07	0.22 ± 0.03	0.17 ± 0.04
$\theta = -0.7$	2.67 ± 0.82	0.72 ± 0.09	0.23 ± 0.04	0.26 ± 0.06
$\theta = -1.0$	2.67 ± 0.82	1.04 ± 0.16	0.46 ± 0.10	0.57 ± 0.29

Fast detection with hierarchical models

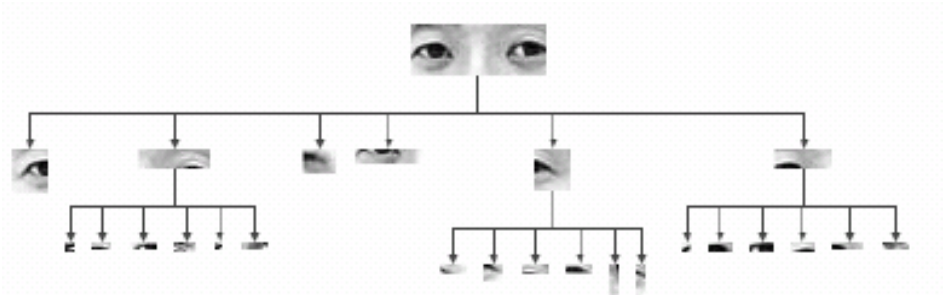


Stochastic Grammar of Images

Zhu and Mumford, `Quest for a Generic Grammar of Images', 2007



Hierarchies in vision and speech



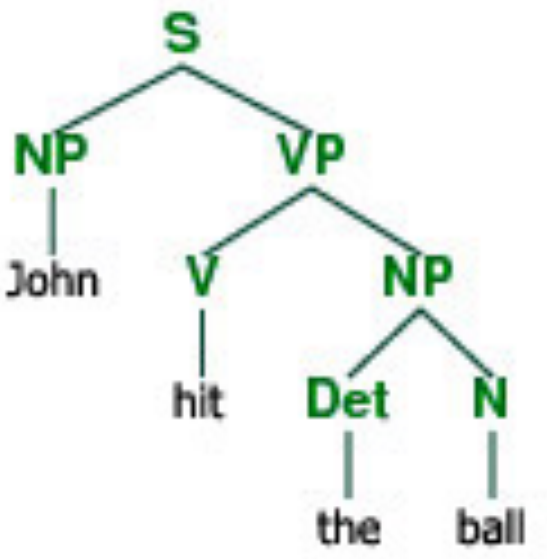
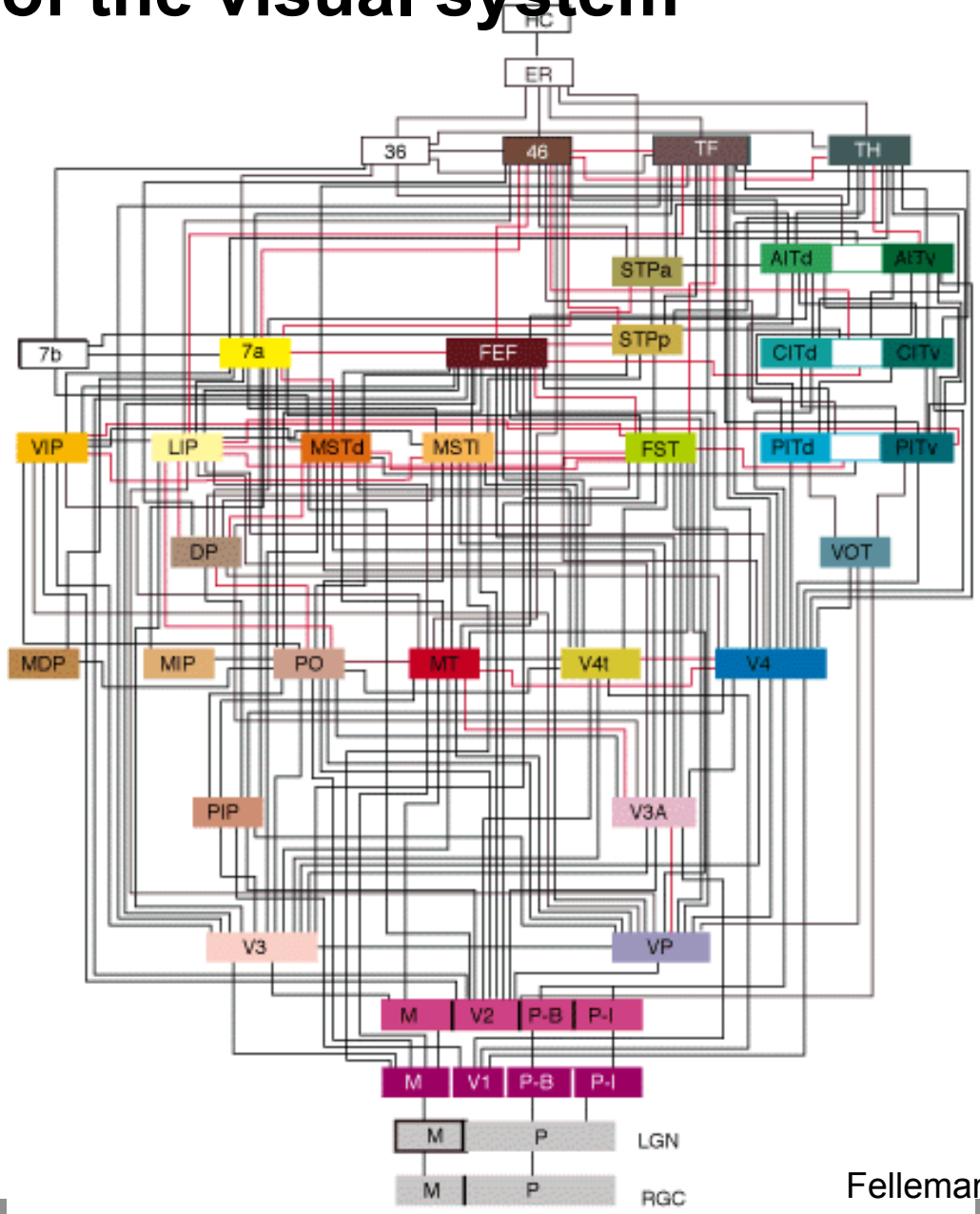
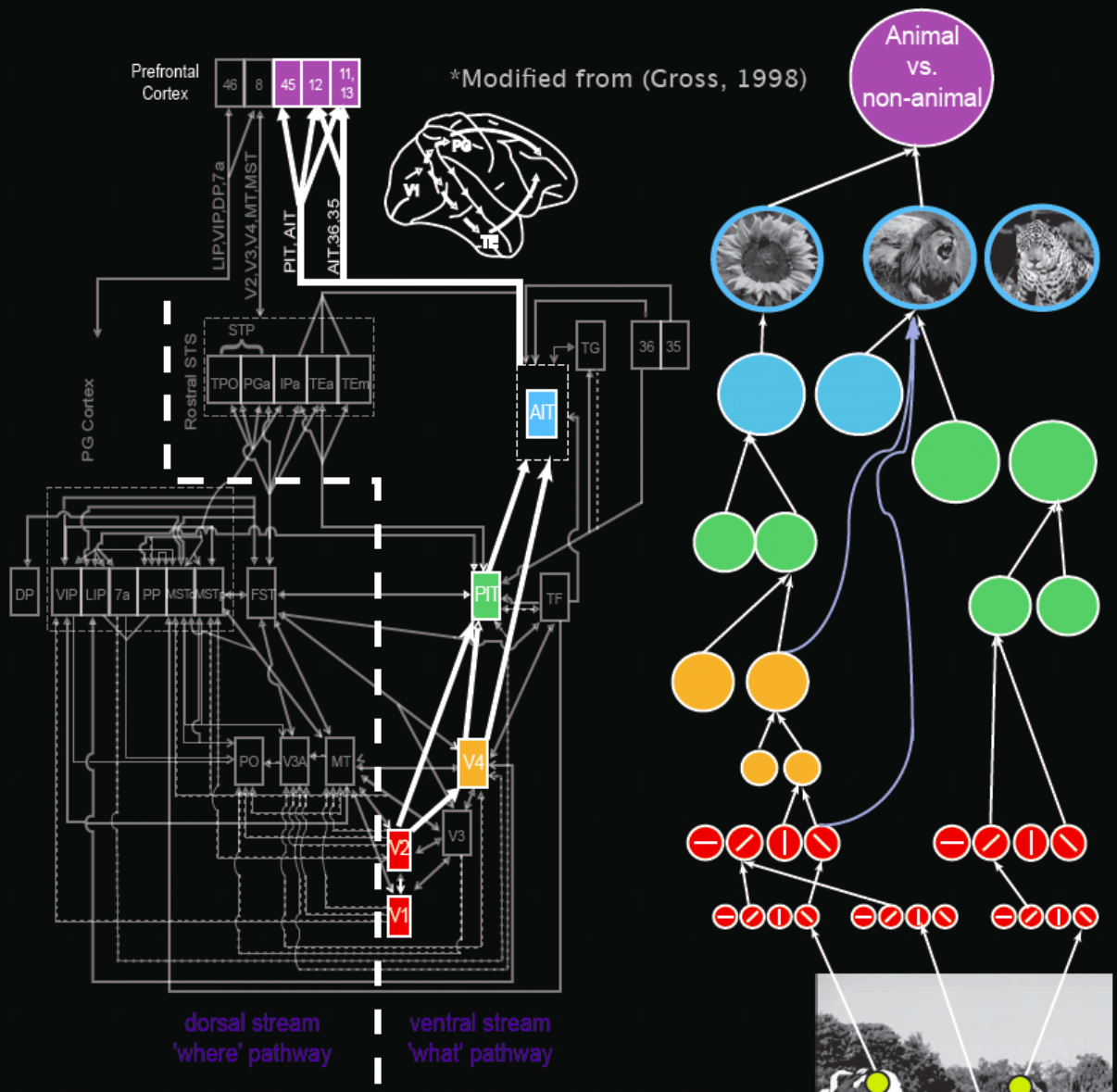


Diagram of the visual system



Felleman and Van Essen, 1991

Slide by T. Serre



Model layers	RF sizes	Num. units
classification units		10^0
S4	7°	10^2
C3	7°	10^3
C2b	7°	10^3
S3	1.2°- 3.2°	10^4
S2b	0.9°- 4.4°	10^7
C2	1.1°- 3.0°	10^5
S2	0.6°- 2.4°	10^7
C1	0.4°- 1.6°	10^4
S1	0.2°- 1.1°	10^6

Supervised task-dependent learning

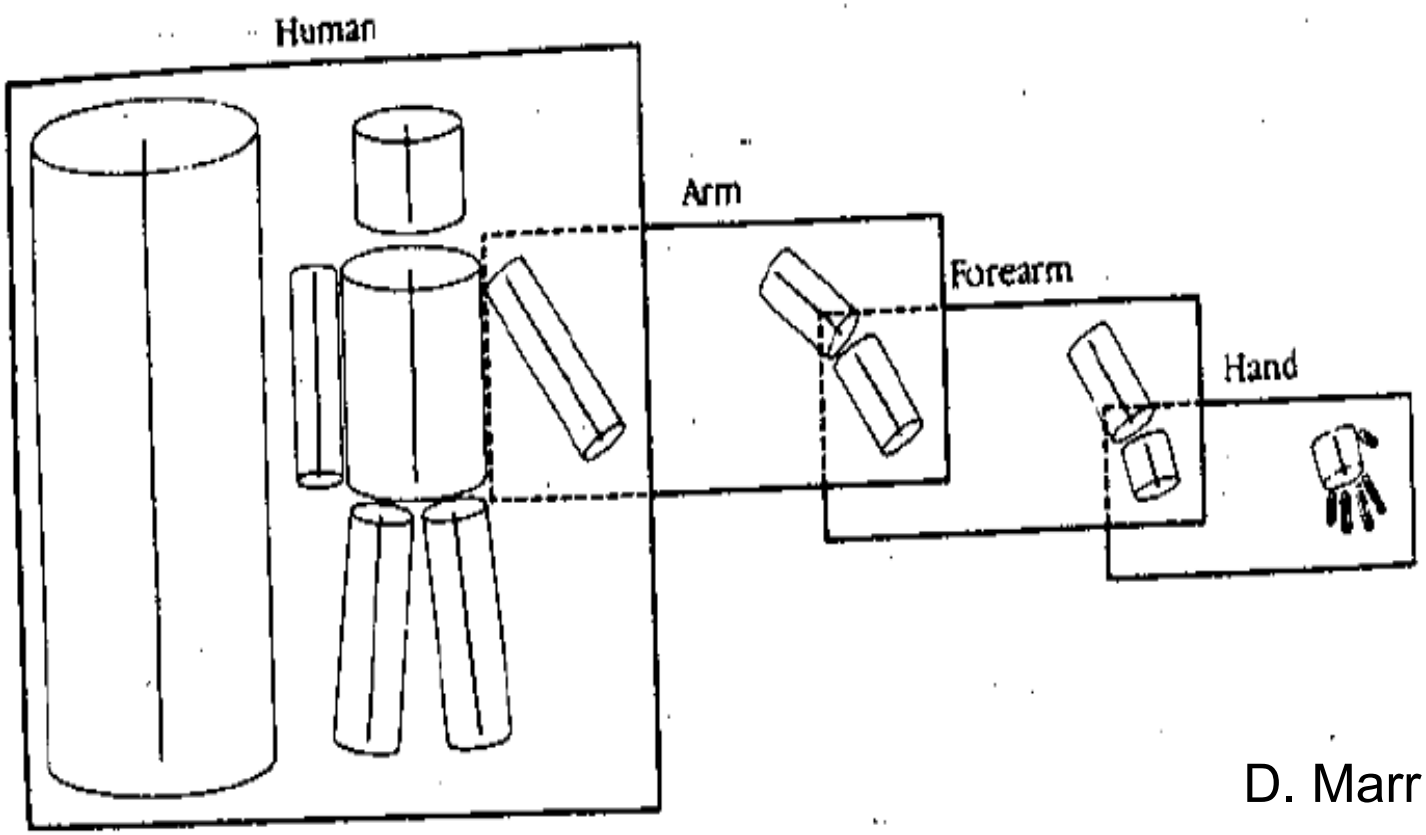
Unsupervised task-independent learning

Increase in complexity (number of subunits), RF size and invariance

Simple cells
Complex cells
Tuning
MAX
Main routes
Bypass routes

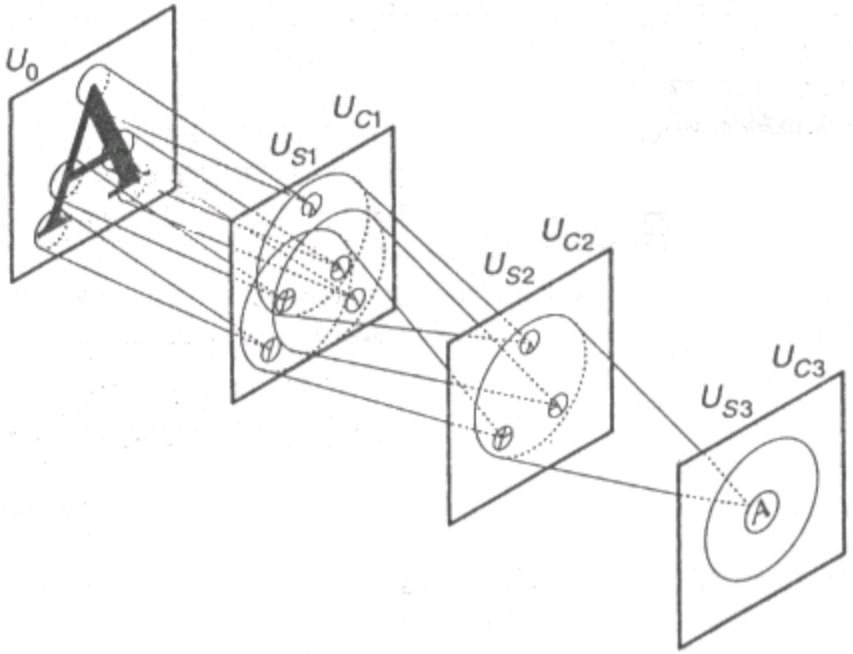
(Riesenhuber & Poggio 1999 2000;
Serre Kouh Cadieu Knoblich Kreiman & Poggio 2005;
Serre Oliva & Poggio 2007)

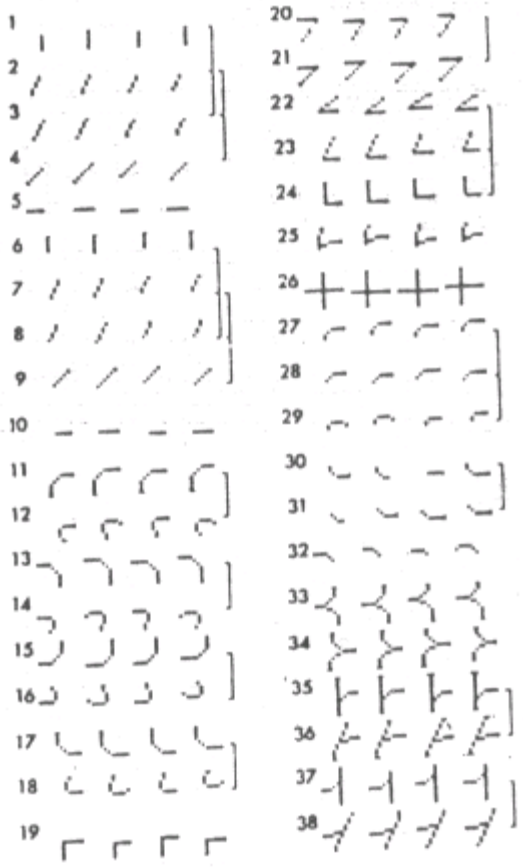
D. Marr's cylinders, 1978



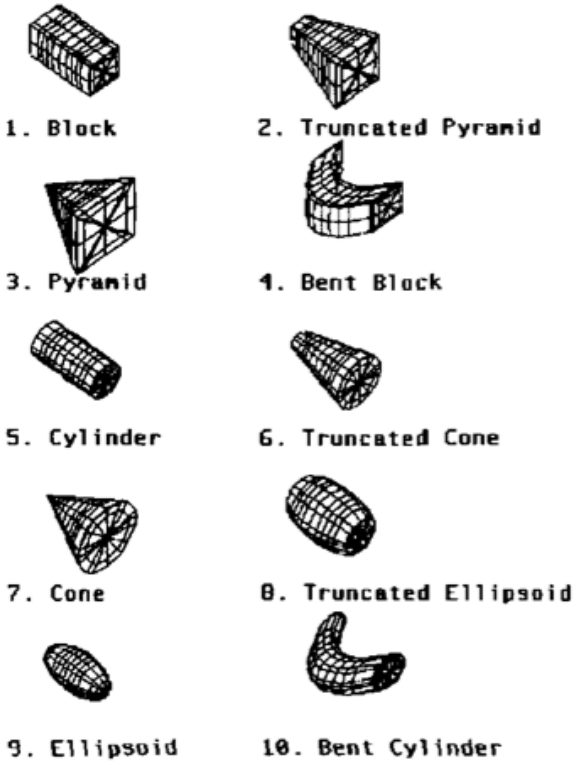
D. Marr

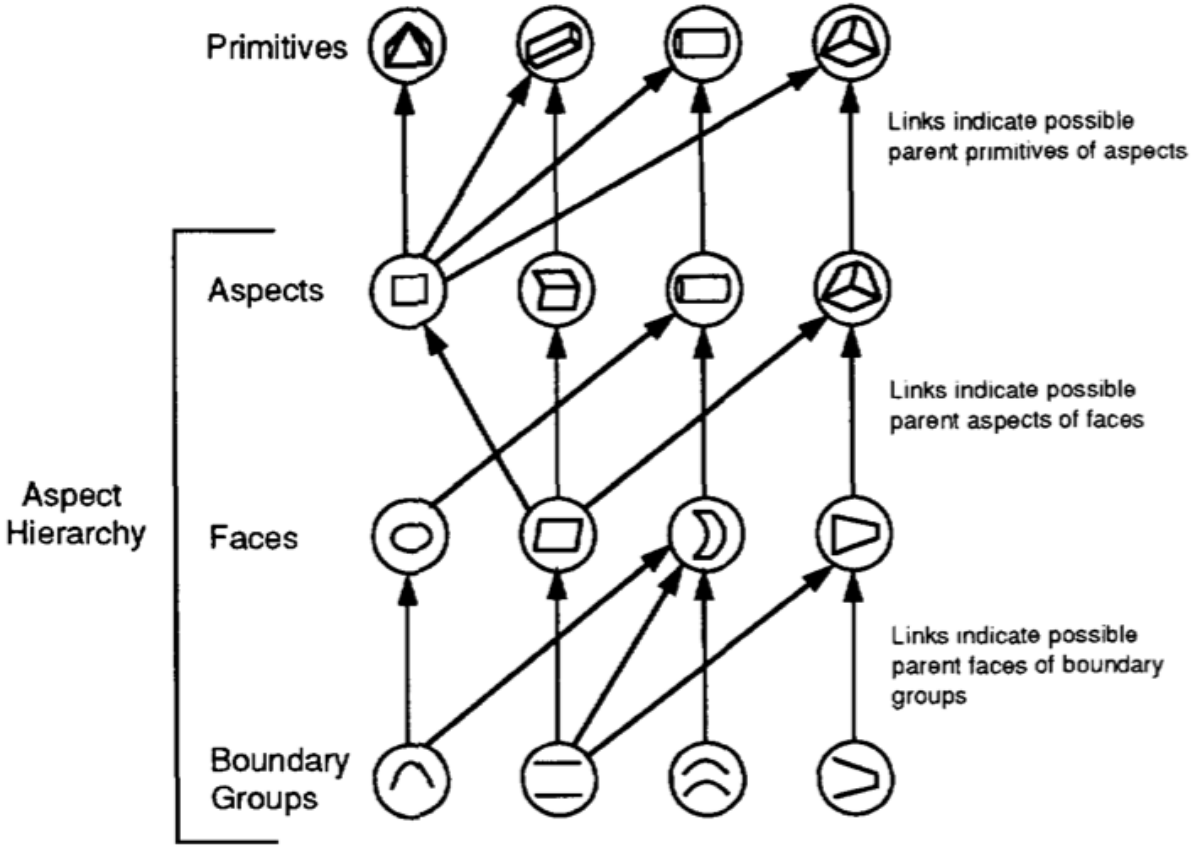
Fukshima's Neocognitron, 1983





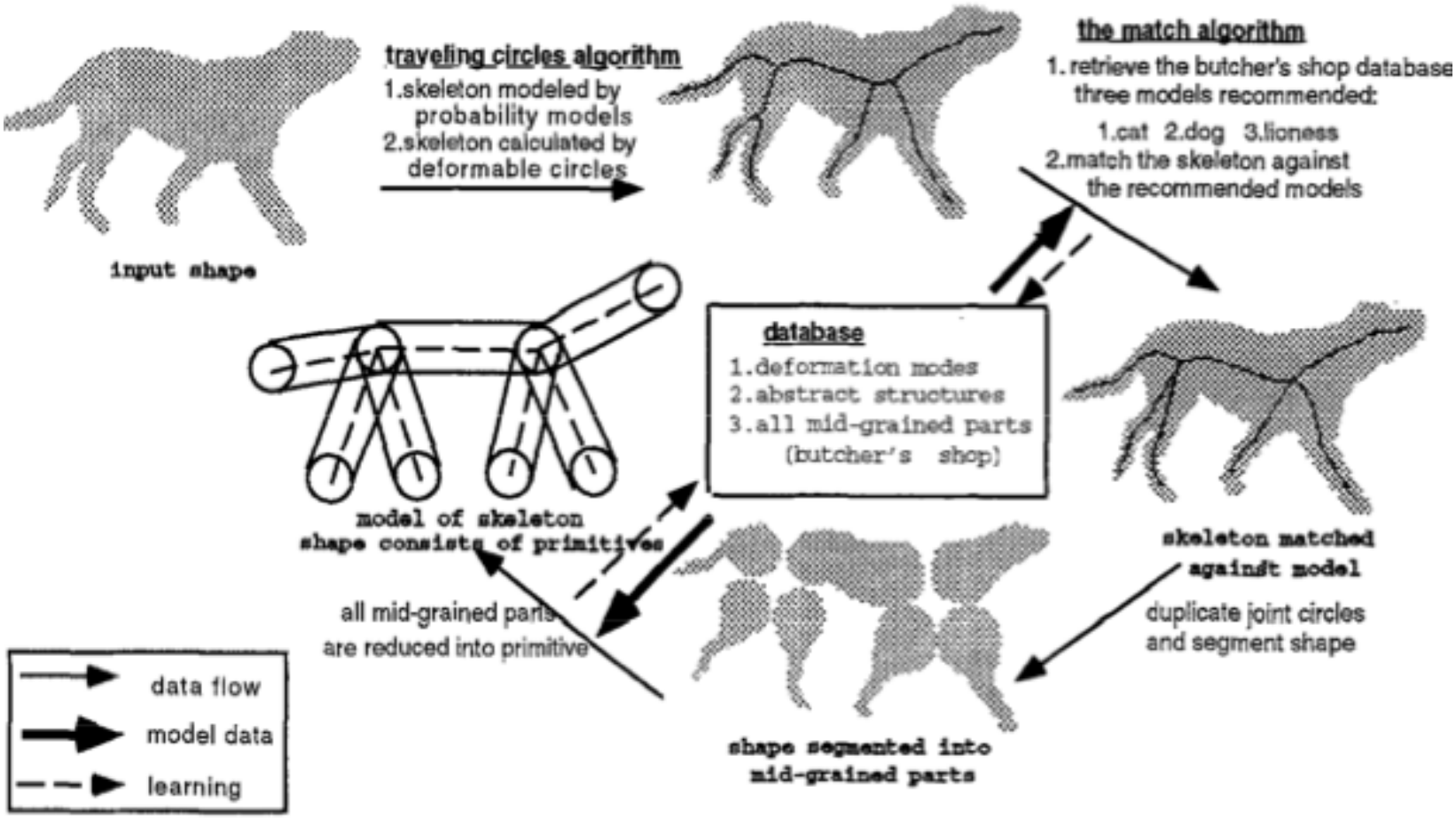
Perceptual Organization





I. Biederman, Recognition-by-components: a theory of human image understanding,1987
S. Dickinson et. al., 3-D Shape Recovery Using Distributed Aspect Matching, 1992

Perceptual Organization



S.C. Zhu and A. Yuille, FORMS: A Flexible Object Recognition and Modeling System, 1996.

Generalized A* Parsing

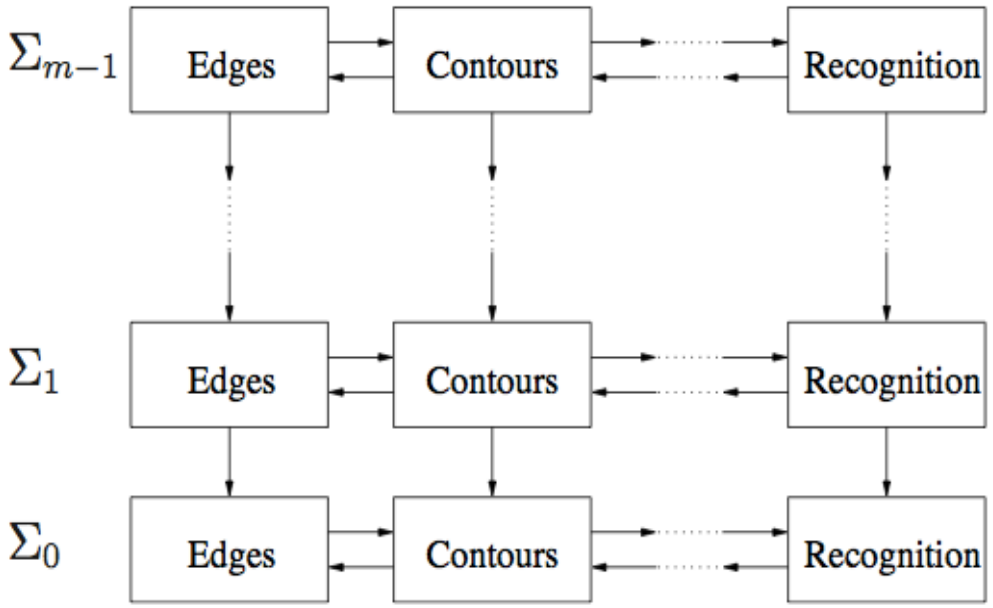
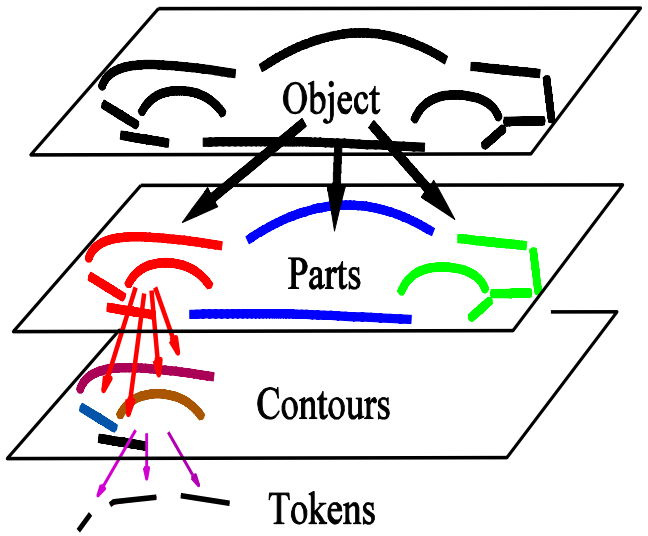


Figure 8: A vision system with several levels of processing. Forward arrows represent the normal flow of information from one stage of processing to the next. Backward arrows represent the computation of contexts. Downward arrows represent the influence of contexts.

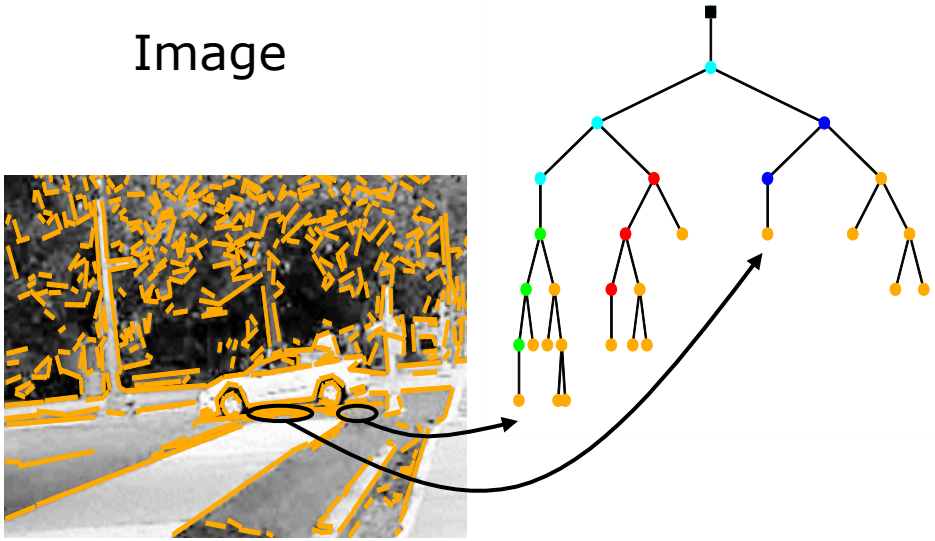
P. Felzenszwalb, D. McAllester: The Generalized A* Architecture. JAIR (2007).

Hierarchical Shape Models



$$\log P(\mathcal{T}_{1:C}|\mathcal{S}_1, \mathcal{P}_{1:3})$$
$$= \log P_O(\mathbf{p}_O) + \sum_{p \in P} \log P_{p,O}(\mathbf{p}_{p|O})$$
$$+ \sum_{p \in P} \sum_{c \in S_p} \log P_{c,p}(\mathbf{p}_{c|p})$$
$$+ \sum_{c \in C} \log P(\mathcal{T}_{S_c}|\mathbf{p}_c) + c(\mathcal{T}_M)$$

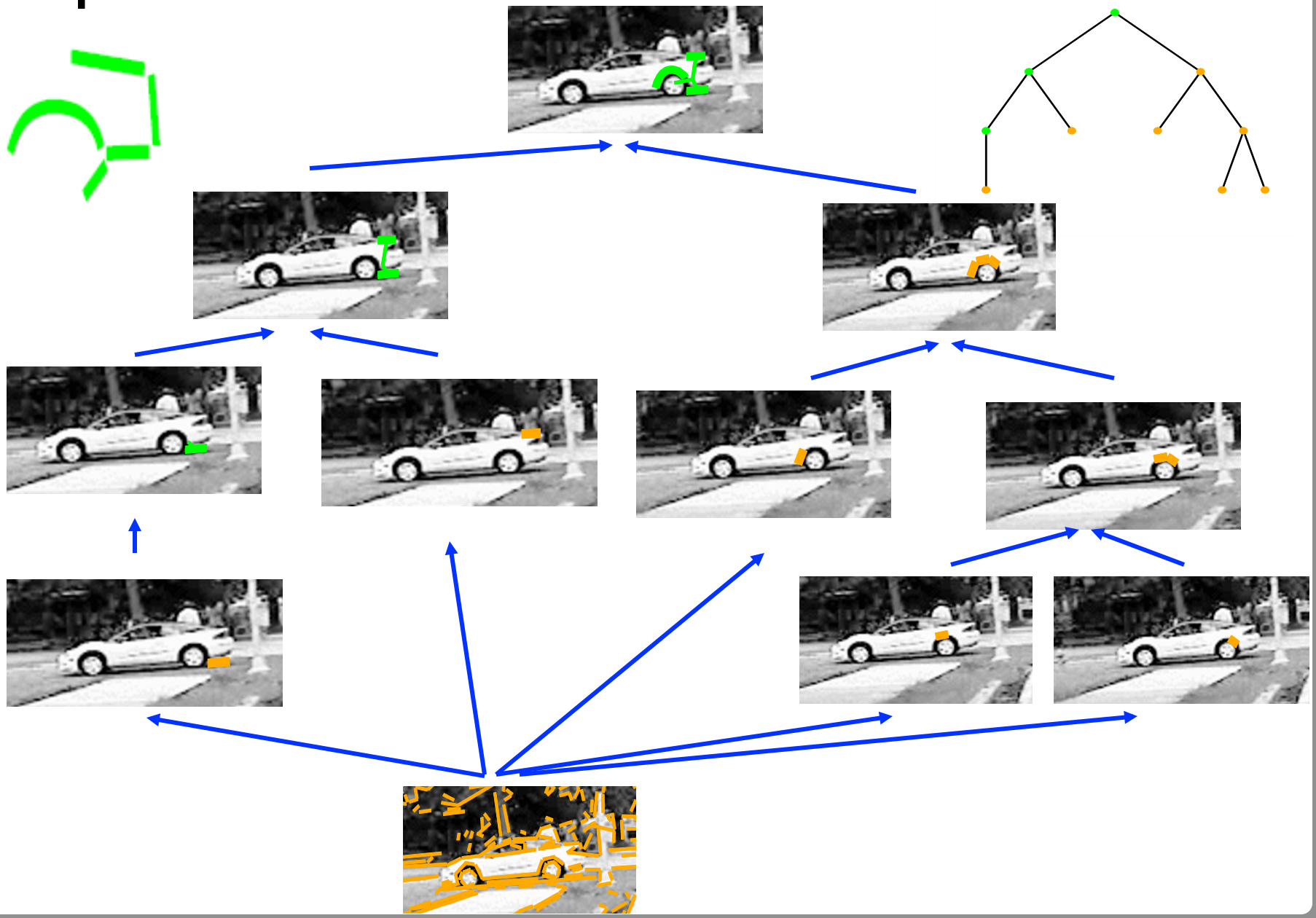
Image



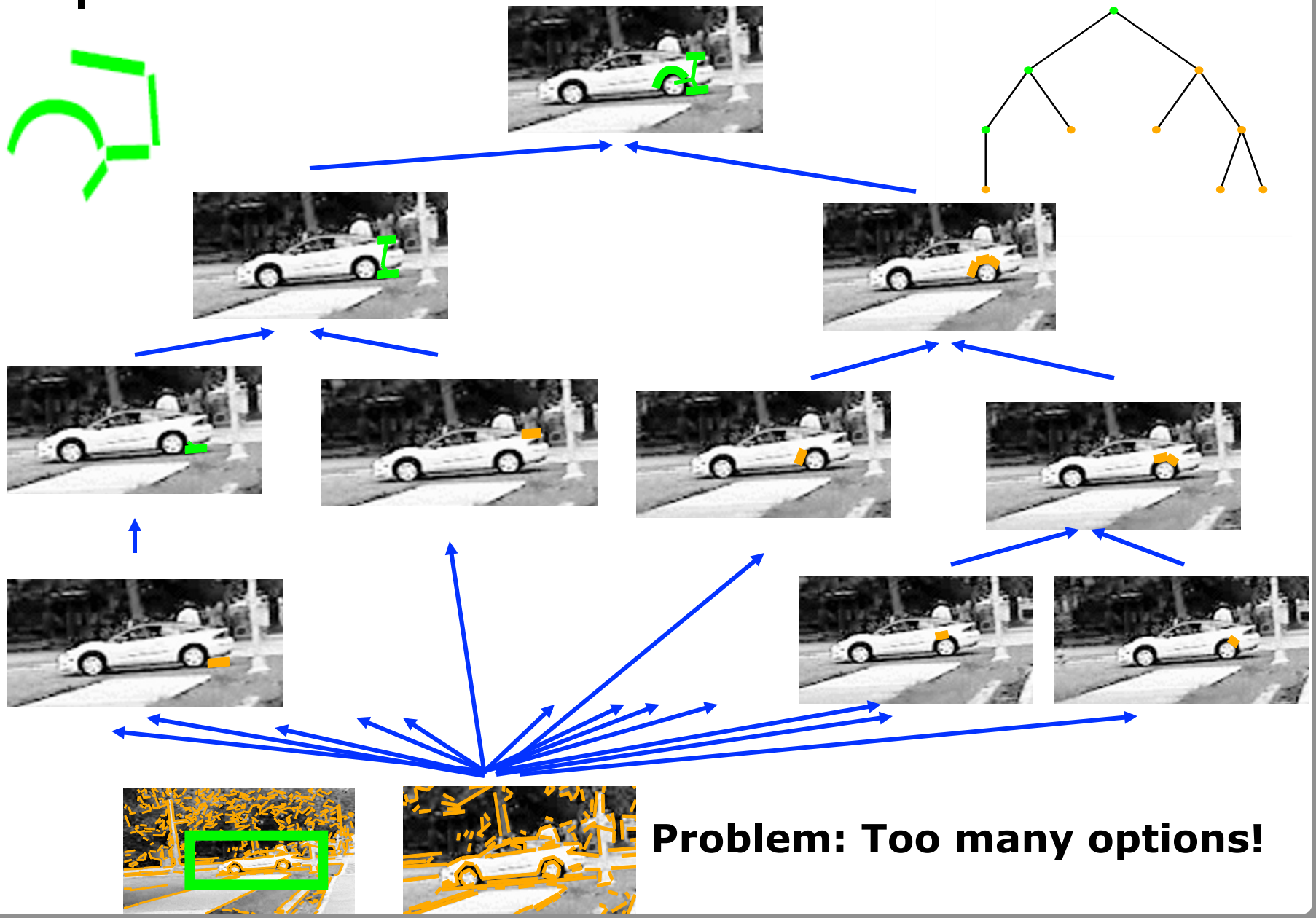
Object

I.K. and A. Yuille, **Inference and Learning with Hierarchical Shape Models, IJCV 2011**

Composition of ‘back’



Composition of 'back'



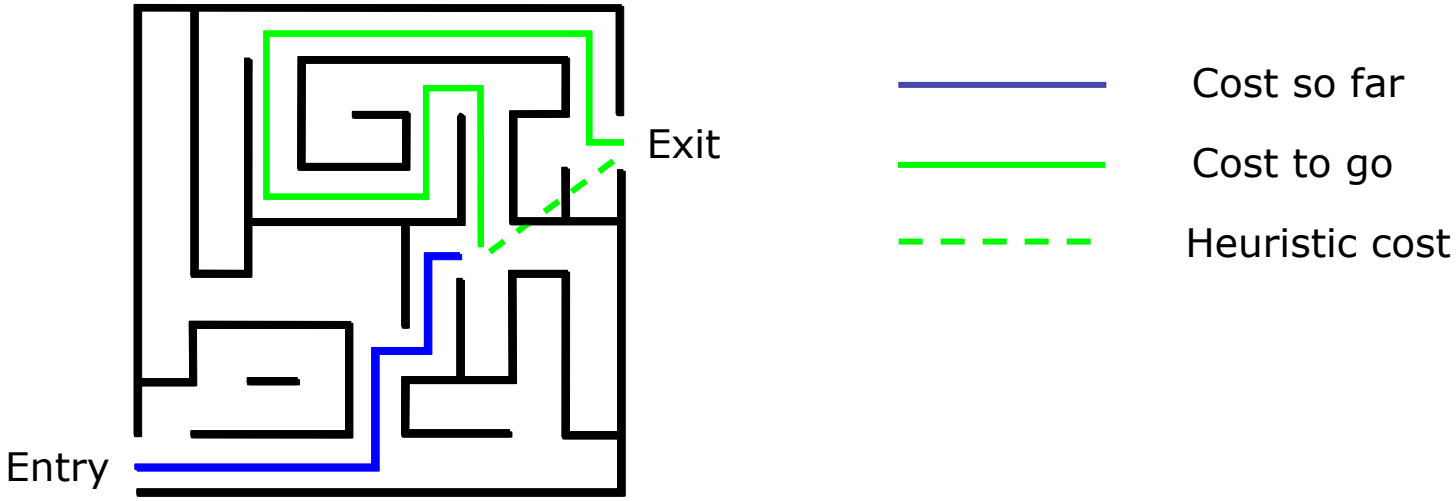
Problem: Too many options!

A* search & parsing

Dijkstra: prioritize based on 'cost so far' $c(s)$ $\{s : c(s) < c(S)\}$

A* : approximate 'cost to go' with heuristic $h(s)$ $\{s : c(s) + h(s) < c(S)\}$

prioritize using $c(s) + h(s)$



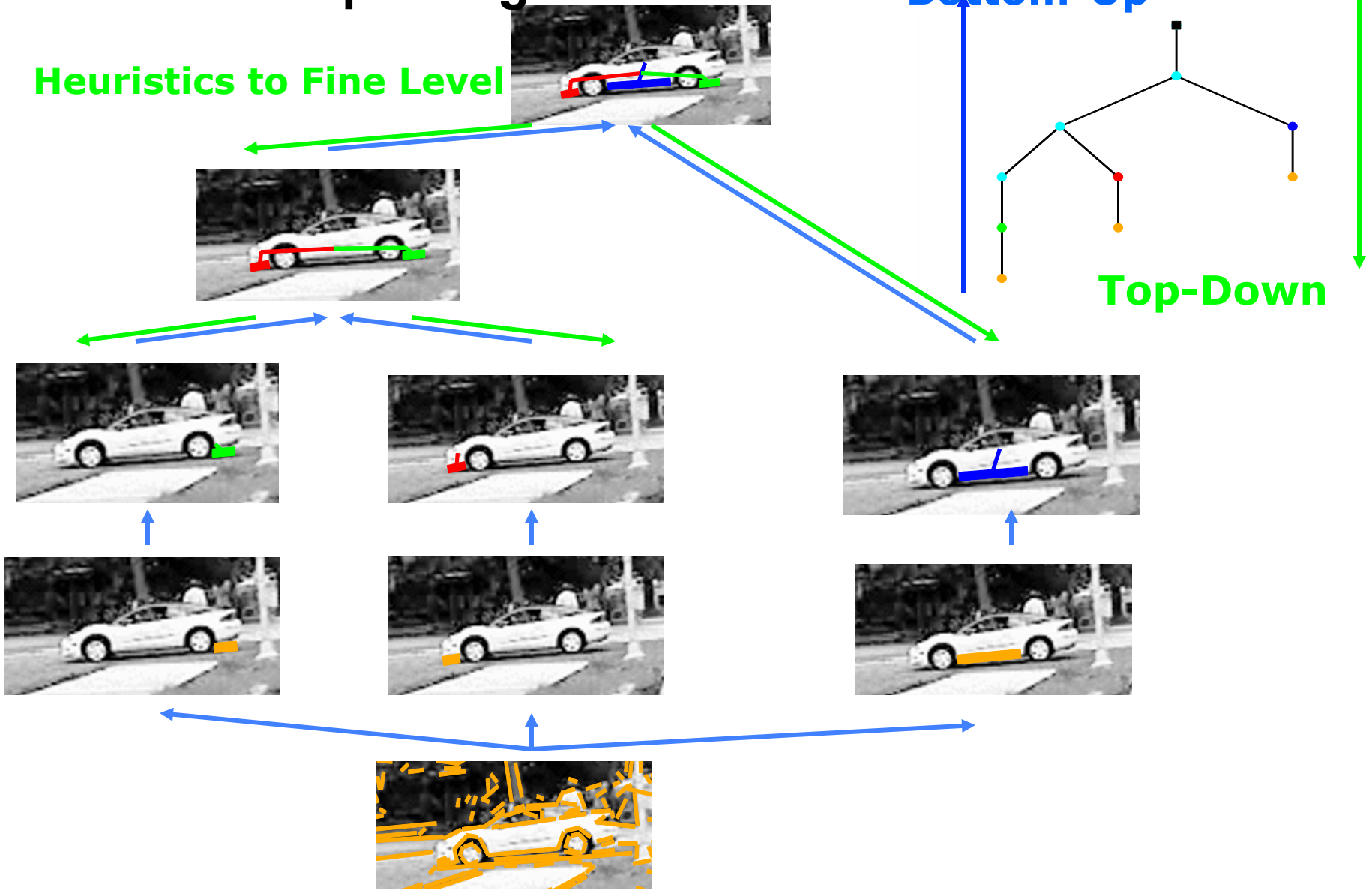
A* Parsing:

D. Klein and C. Manning, A* Parsing: Fast Exact Viterbi Parse Selection, (NAACL) 2003

P. Felzenszwalb, D. McAllester: The Generalized A* Architecture. (JAIR), 2007.

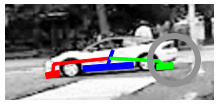
Our work: Object Parsing

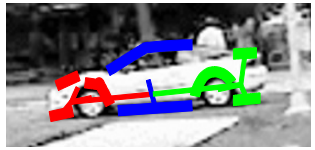
Coarse-level parsing

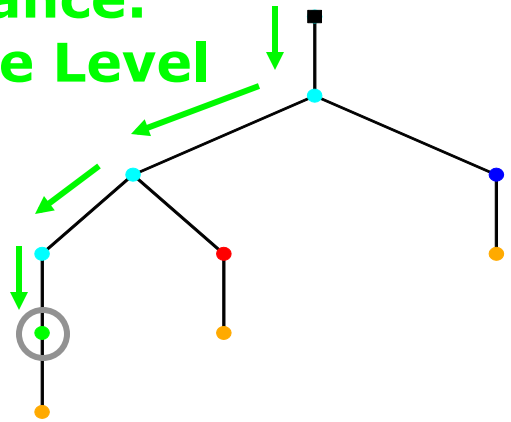


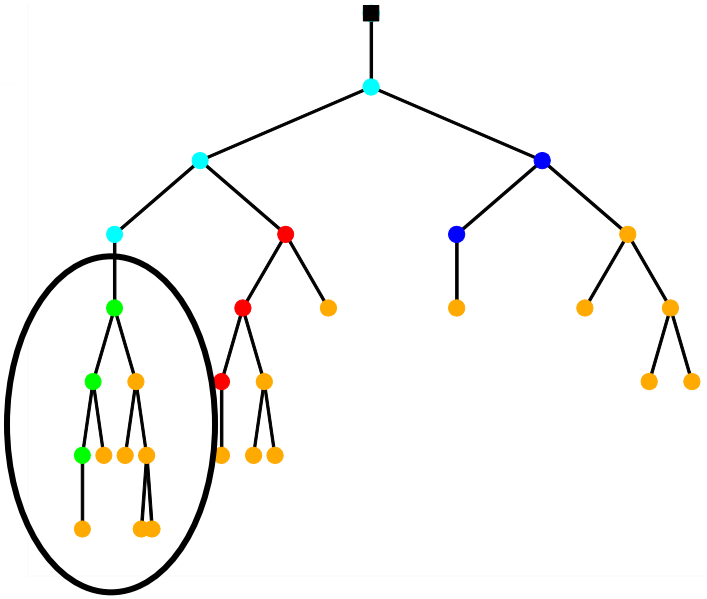
Fine-level parsing

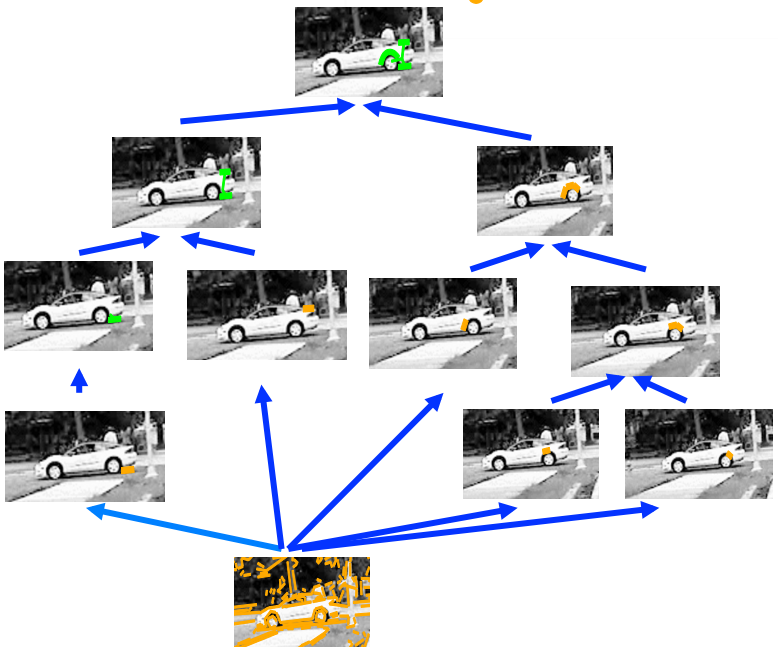
Top-Down Guidance: Heuristic, Coarse Level







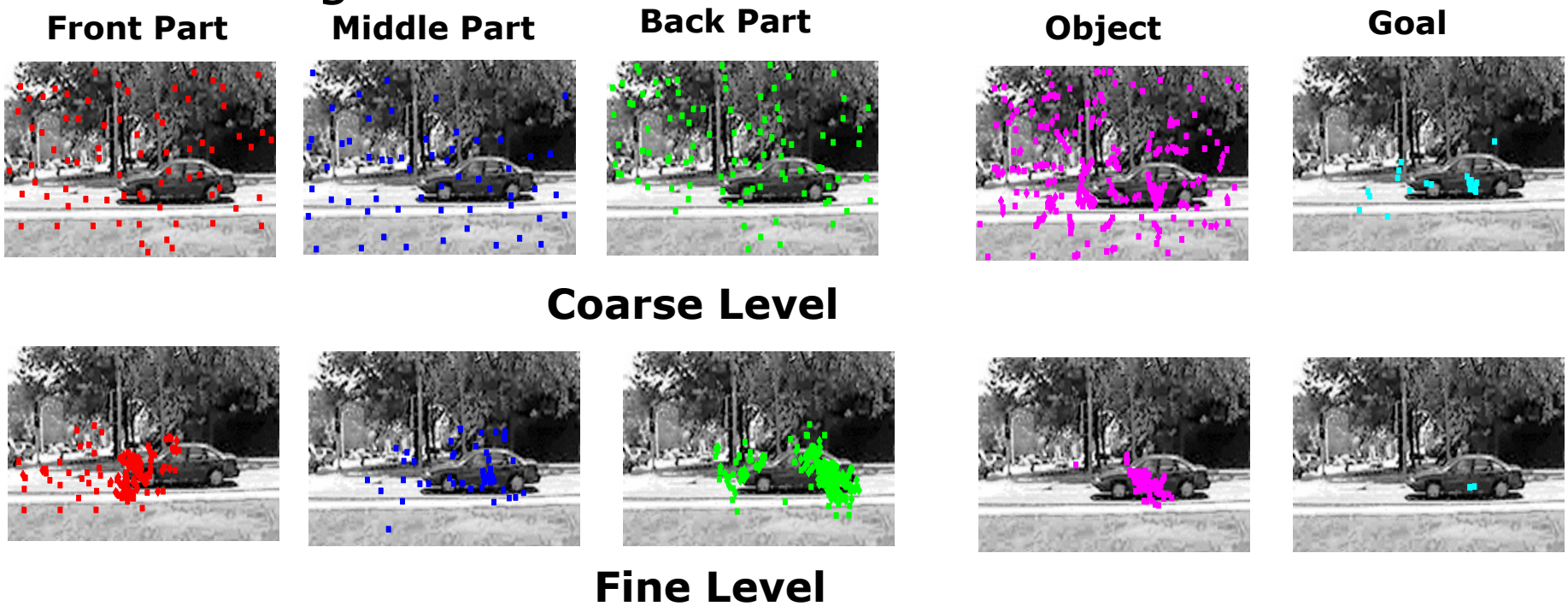




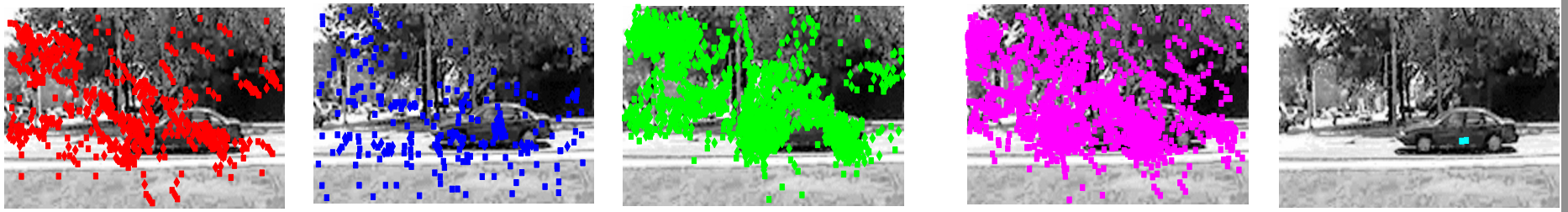
Bottom-Up Composition, Fine level

A* vs Knuth's Lightest Derivation (DP)

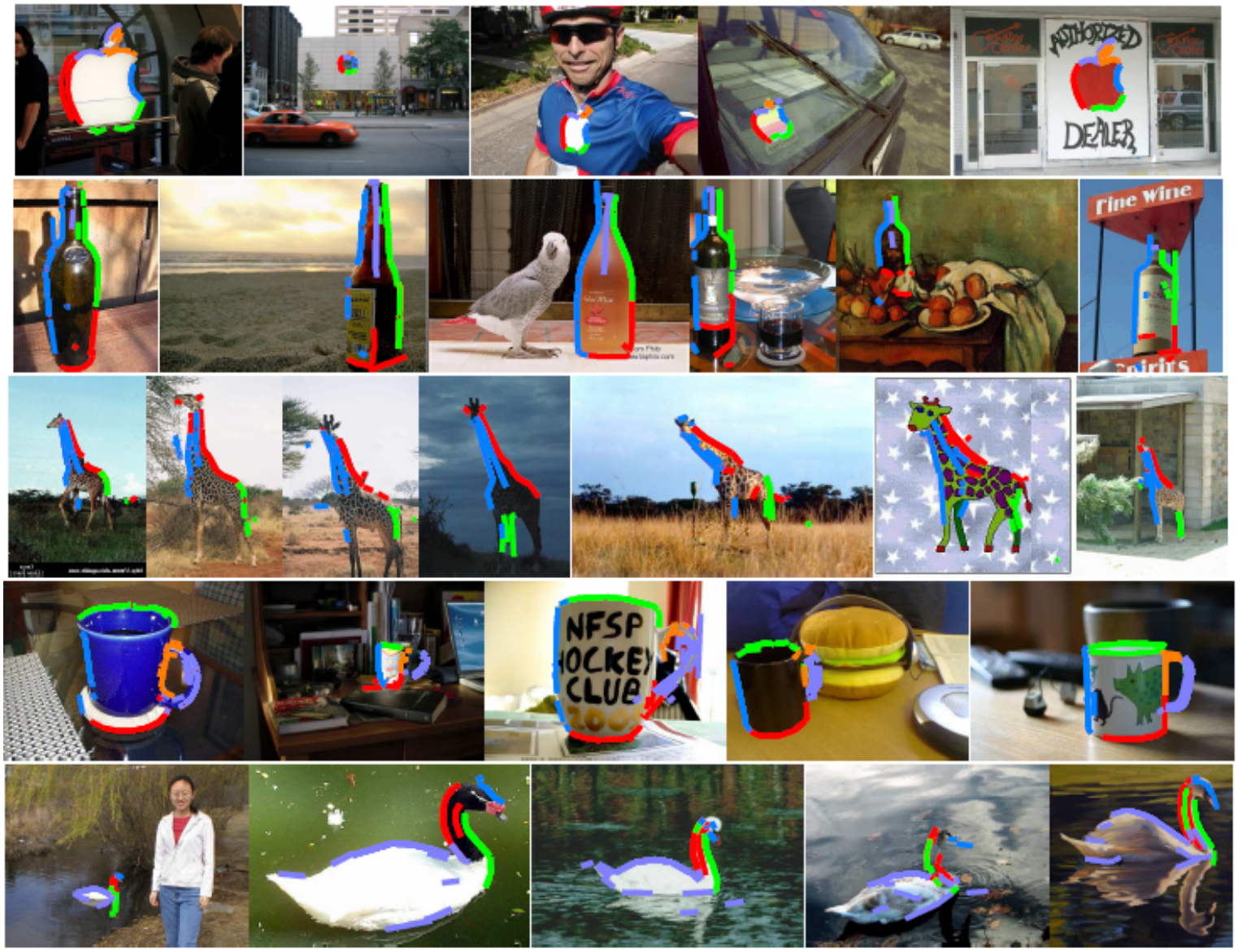
- A* Parsing



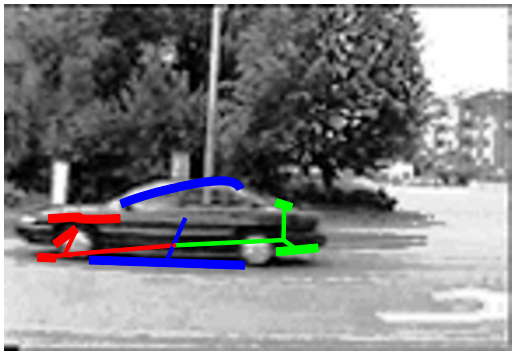
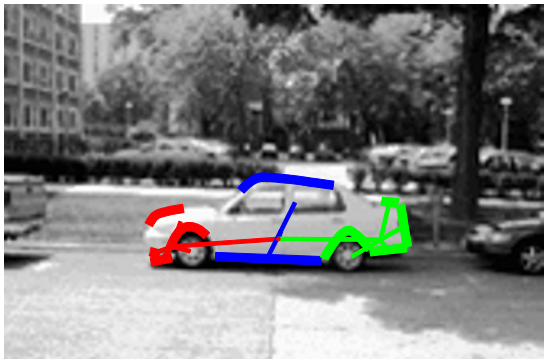
- KLD Parsing (only fine level)

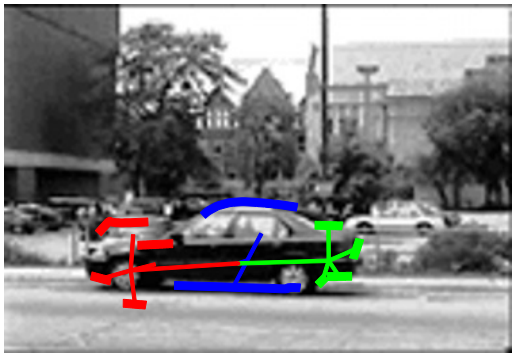
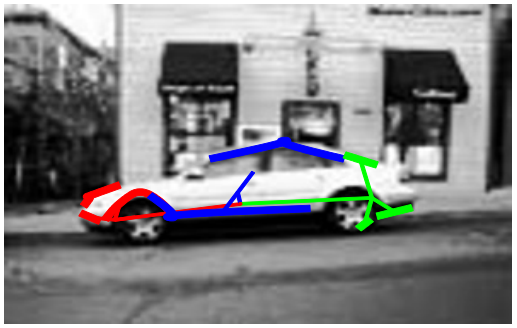
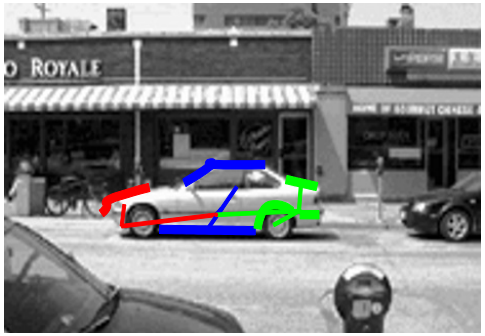


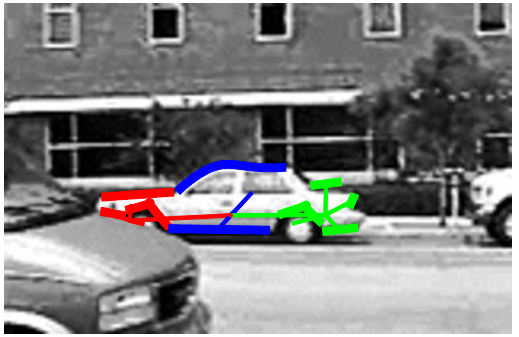
Parsing and localization results



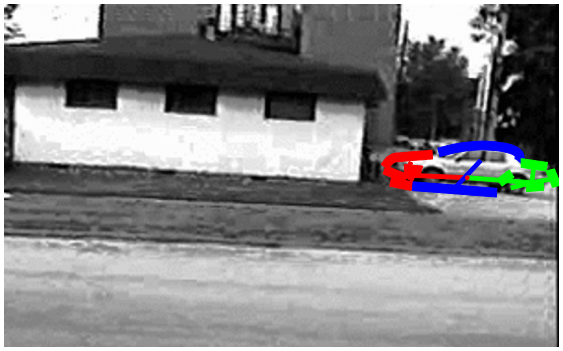
Parsing & Localization Results - I

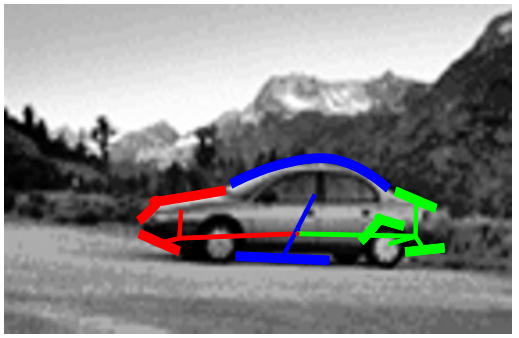
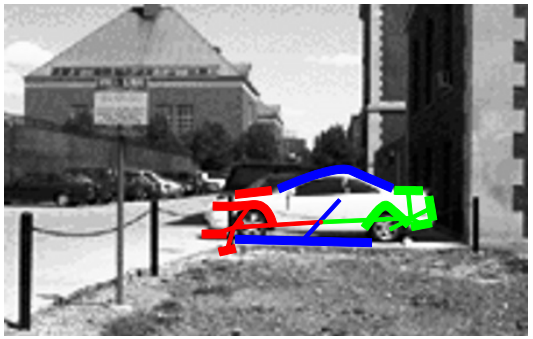
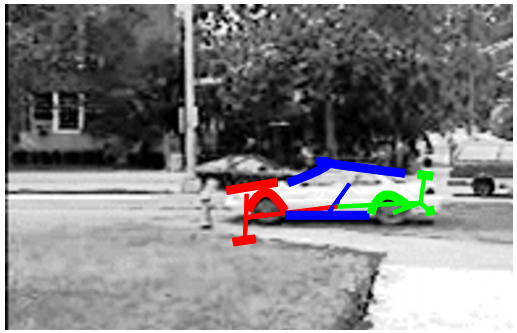




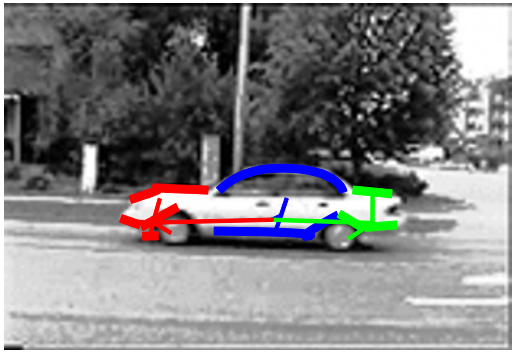
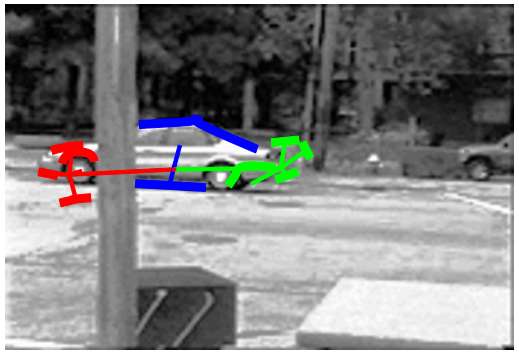
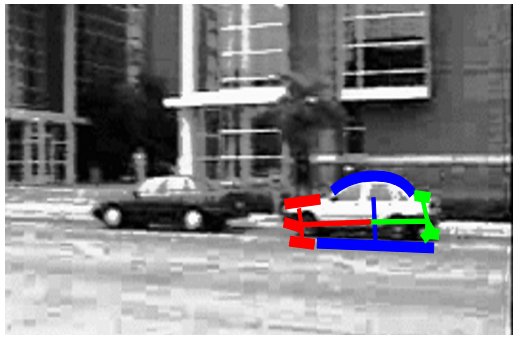




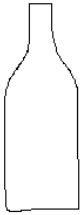
Parsing & Localization Results - II

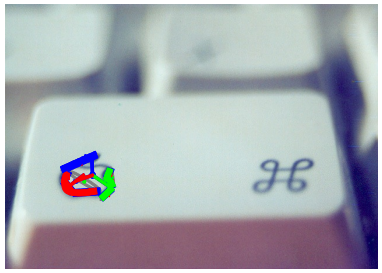


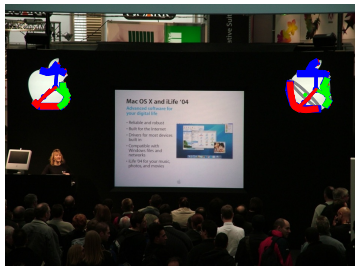




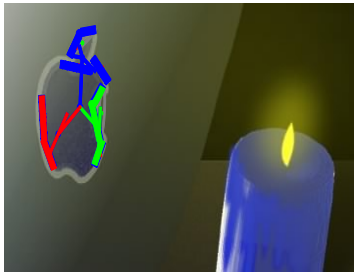




Parsing & Localization Results - III

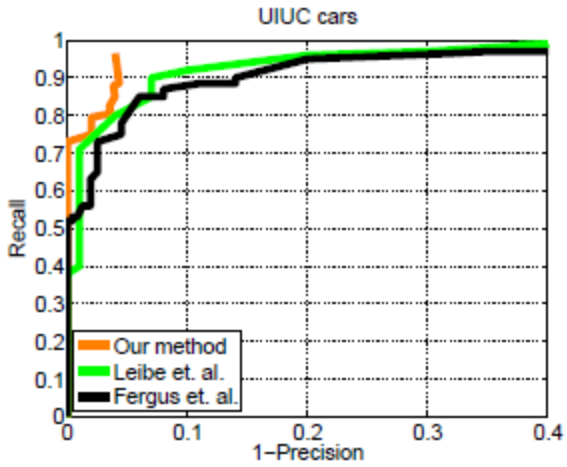


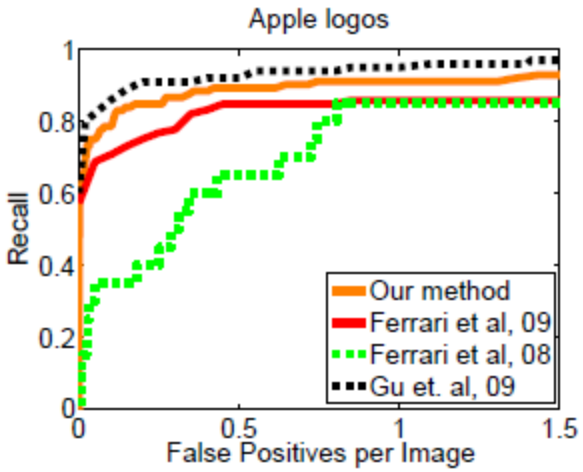


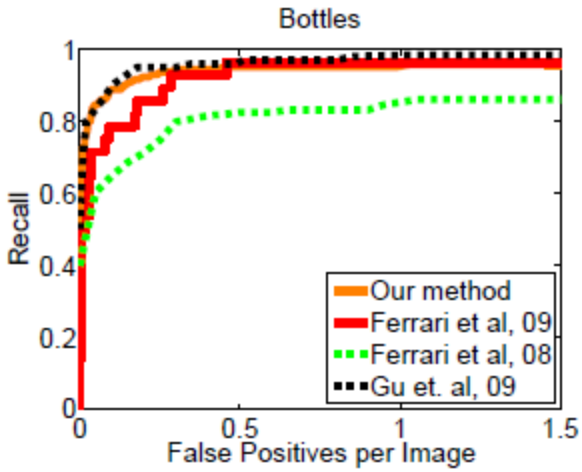


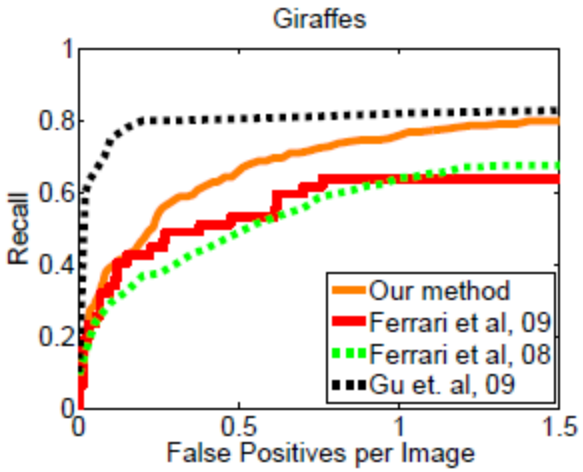


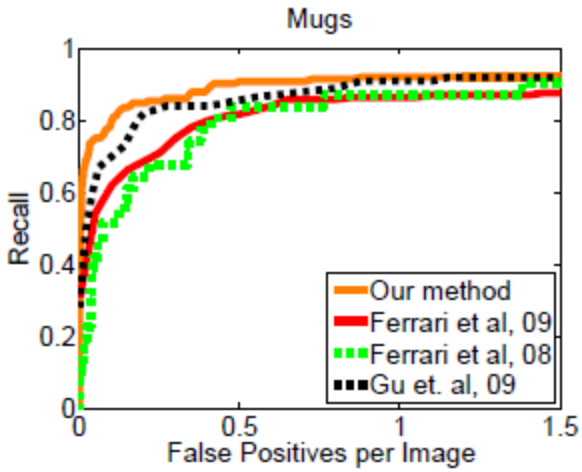
Benchmark results

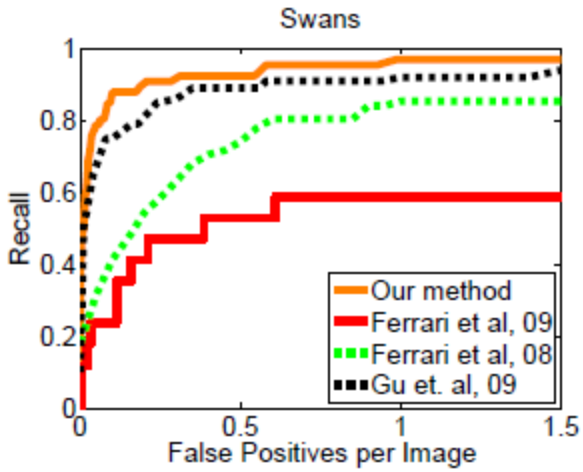






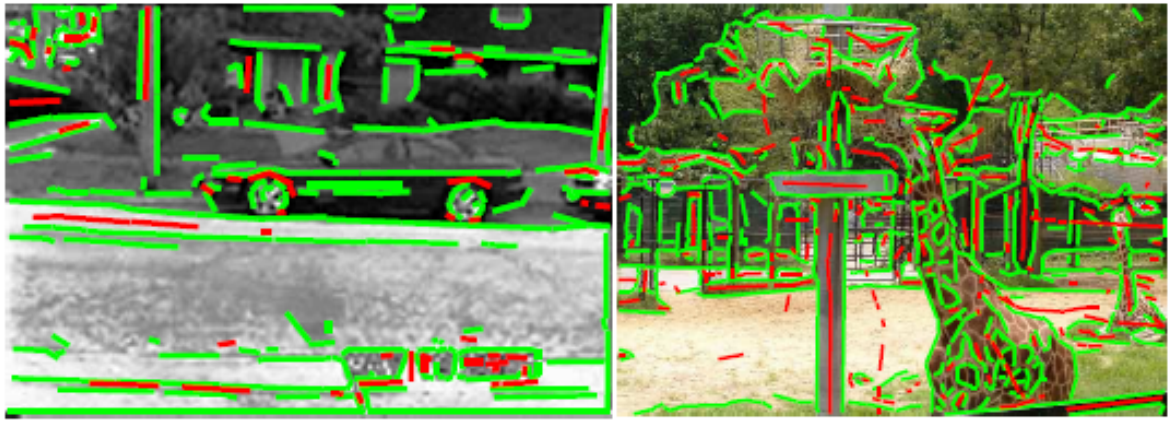






Failure cases

Front-end failures



Missing appearance information/poor shape model