Shape modelling & object detection

Iasonas Kokkinos
Center for Visual Computing
Ecole Centrale Paris / INRIA Saclay

Core problem: correspondence

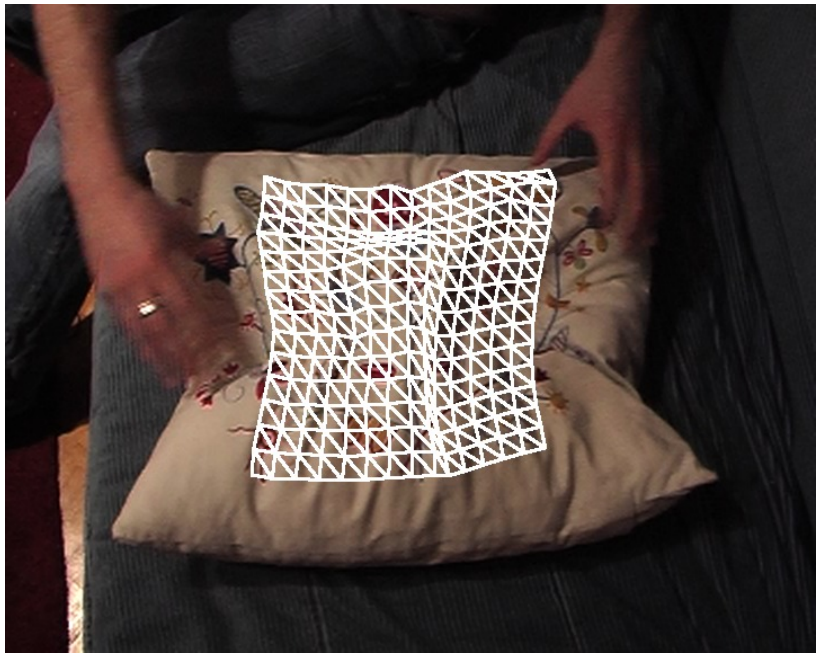
‘what is here comes from here’

Simple case: global rigid transformation (4-9 DoF)

Core problem: correspondence

‘what is here comes from here’

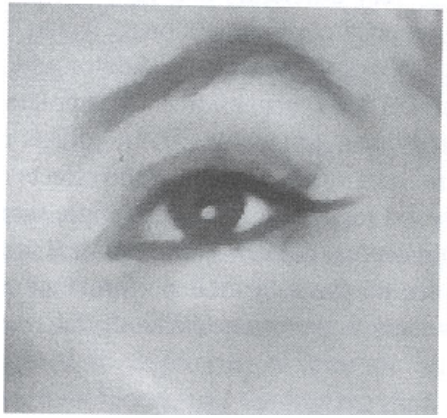
Complications: non-rigid transformation

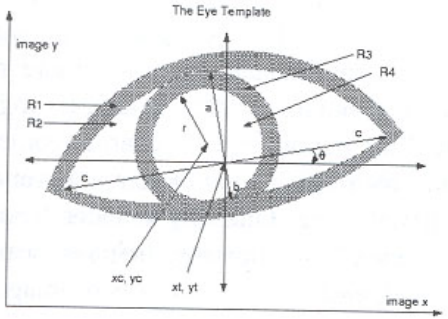


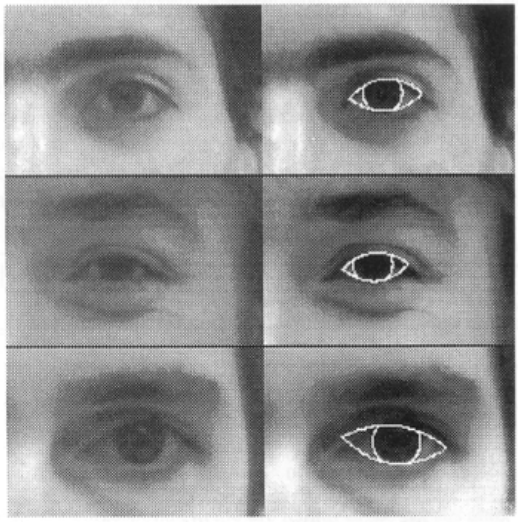
Core problem: correspondence

‘what is here comes from there’

Complications: intra-category variation







‘One shape fits all’

A.L. Yuille, D.S. Cohen and P.W. Hallinan. Feature extraction from faces using deformable templates. CVPR 1989.

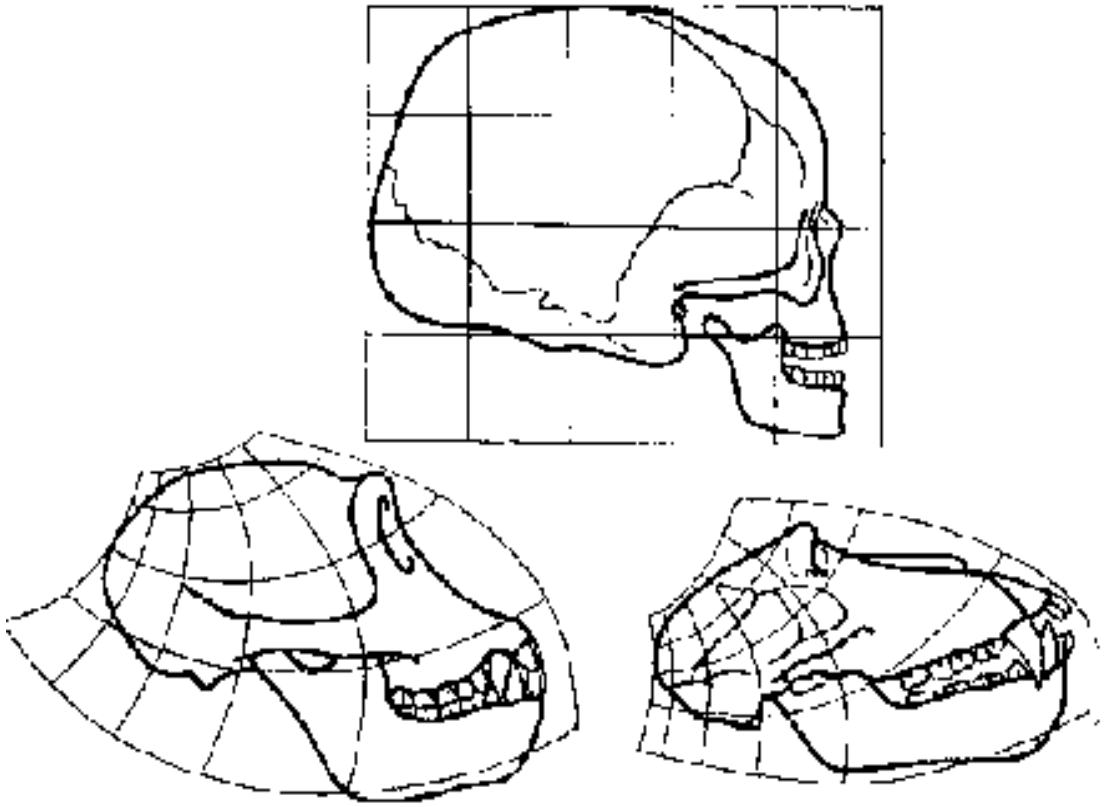
Core problem: correspondence

‘what is here comes from there’

Complications: 3D, pose variability

T. J. Cashman, A. W. Fitzgibbon: What Shape Are Dolphins? Building 3D Morphable Models from 2D Images, 2013
http://research.microsoft.com/en-us/um/people/awf/dolphins/

On growth and form (1917)



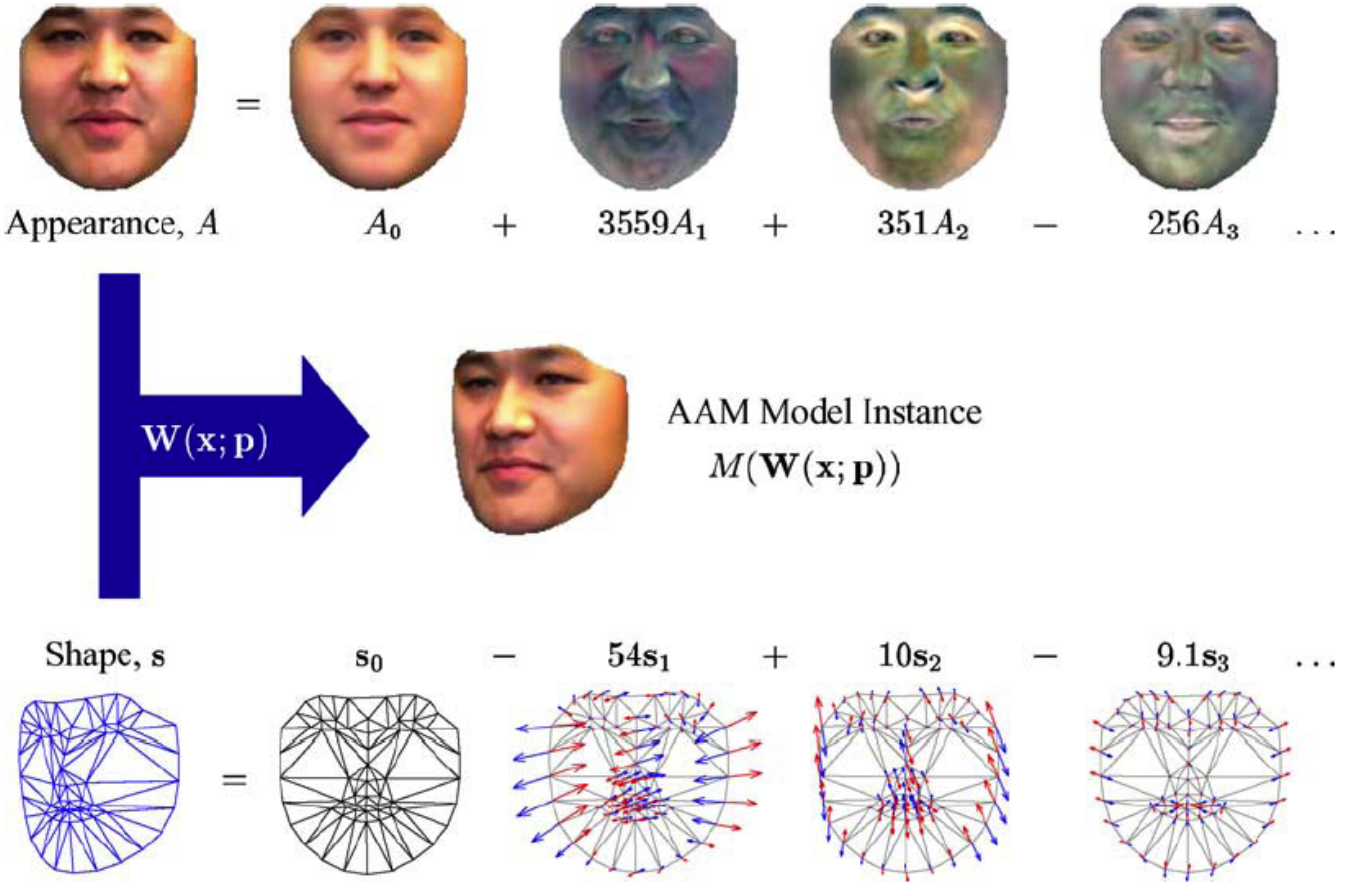
Skulls of a human, a chimpanzee and a baboon
and transformations between them

D’Arcy Thompson, “On Growth and Form” (1917)

Active Shape Models - 1992

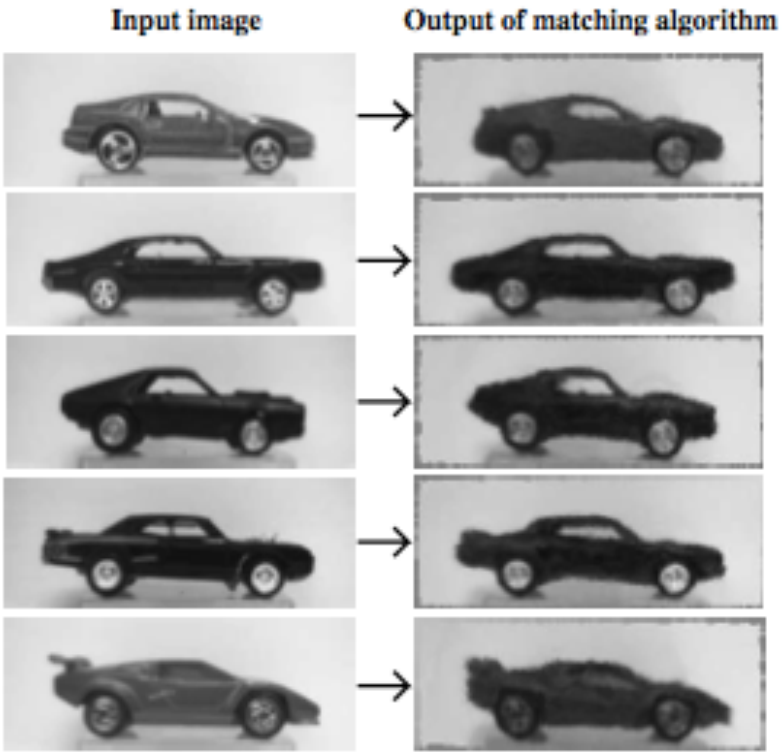
T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Training models of shape **from sets of examples.** BMVC 1992

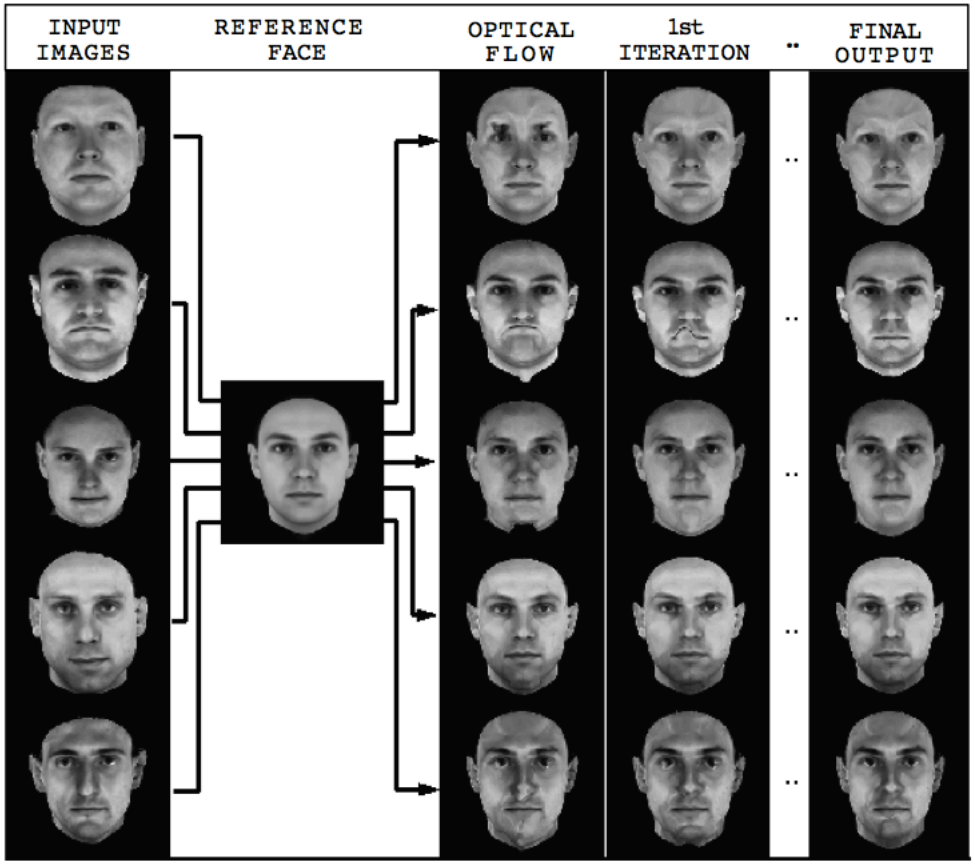
Active Appearance Models - 1998



T.F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models, 1998
T. F. Cootes, G. V. Wheeler, K. N. Walker, C. J. Taylor: View-based active appearance models.2002,
I. Matthews and S. Baker, "Active Appearance Models Revisited," 2004.

Morphable Models - 1996



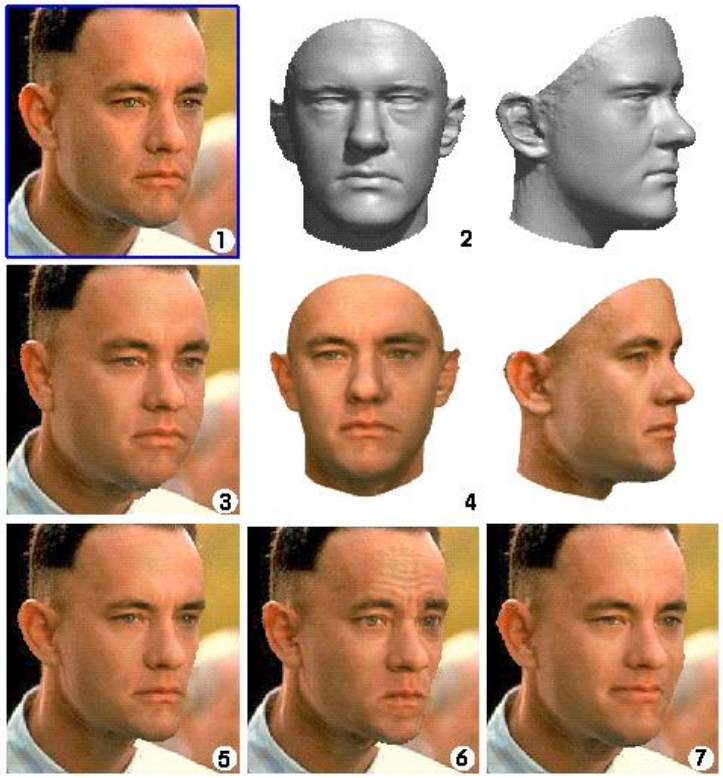


T. Vetter, T. Poggio: Linear Object Classes and Image Synthesis From a Single Example Image., 1997.

T. Vetter, M. J. Jones, T. Poggio: A bootstrapping algorithm for learning linear models of object classes. CVPR 1997

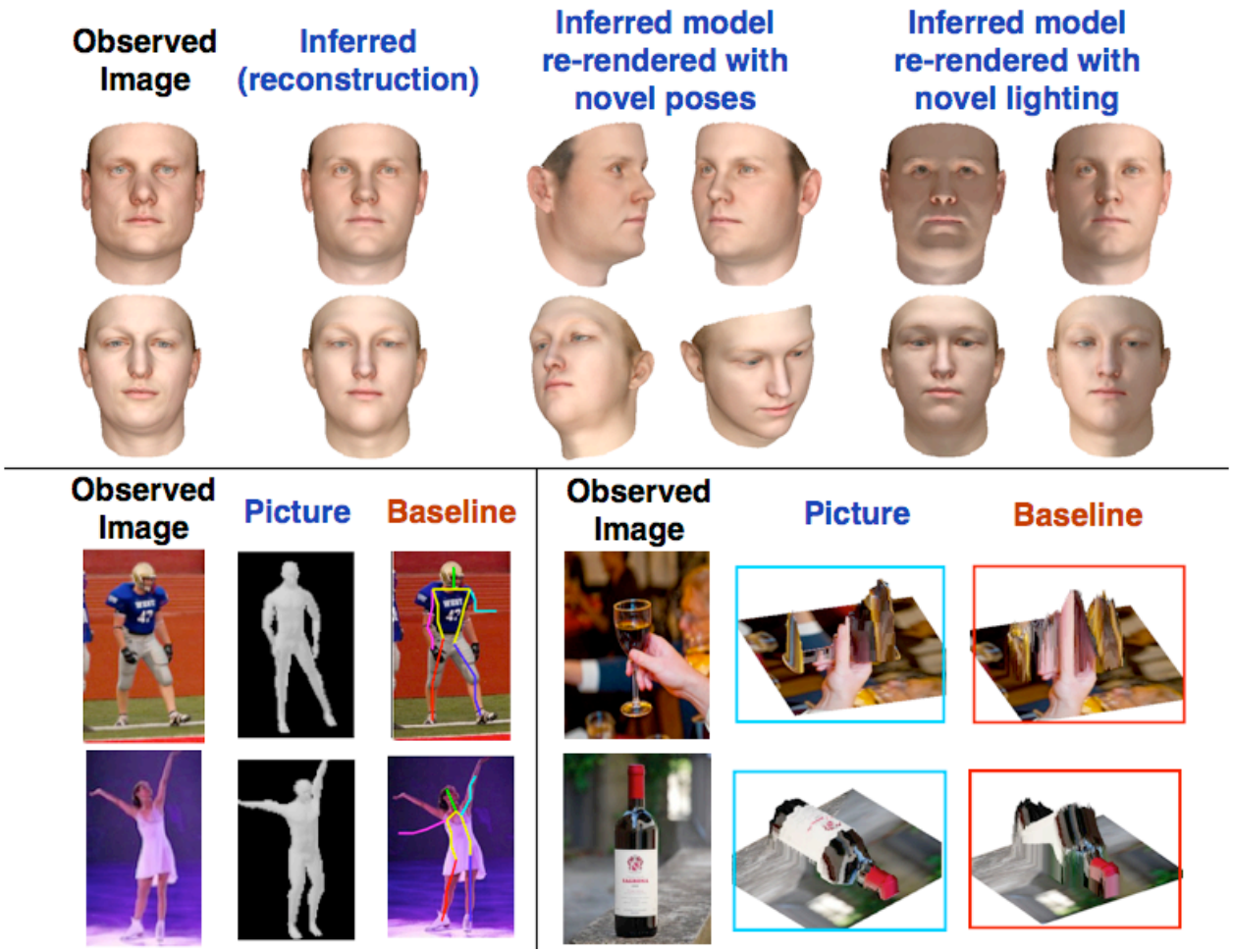
M. J. Jones, Y. Poggio: Multidimensional Morphable Models: A Framework for Representing and Matching Object Classes. IJCV 1998

3D Morphable Models - 1999



V. Blanz, T. Vetter: A Morphable Model for the Synthesis of 3D Faces. SIGGRAPH 1999
V. Blanz, T. Vetter: Face Recognition Based on Fitting a 3D Morphable Model. IEEE PAMI, 2003
T. J. Cashman, A. W. Fitzgibbon: What Shape Are Dolphins? Building 3D Morphable Models from 2D Images, 2013

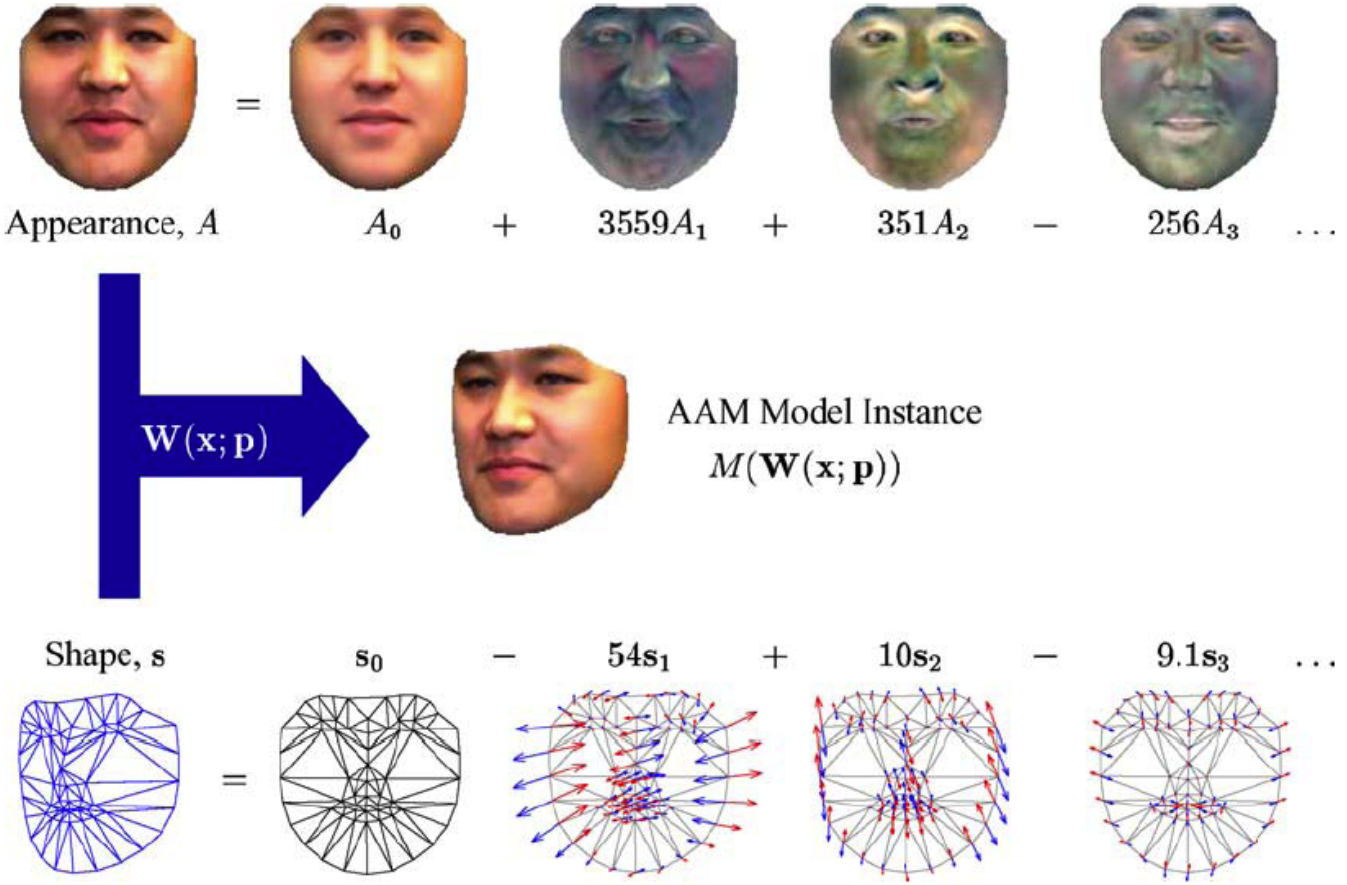
3D Models, 2015



Picture: A Probabilistic Programming Language for Scene Perception: T. D. Kulkarni, P. Kohli, J. B. Tenenbaum, V. Mansinghka, CVPR 2015

http://mrkulk.github.io/www_cvpr15/

Active Appearance Models - 1998



T.F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models, 1998
T. F. Cootes, G. V. Wheeler, K. N. Walker, C. J. Taylor: View-based active appearance models.2002,
I. Matthews and S. Baker, "Active Appearance Models Revisited," 2004.

AAM Search

Slide credits: Tim Cootes

From Lucas-Kanade to AAMs

Brightness constancy constraint

$$I(x_i + u, y + v, t) = I(x_i, y_i, t + 1)$$

$$I(\mathbf{x}_i + u \cdot (\mathbf{1}, \mathbf{0}) + v \cdot (\mathbf{0}, \mathbf{1}), t) = I(\mathbf{x}_i, t + 1)$$

AAM synthesis equation:

$$I_1(\mathbf{x}_i + \sum_k a_k \mathbf{b}_k(\mathbf{x}_i)) = I_2(\mathbf{x}_i)$$

I. Matthews and S. Baker, "Active Appearance Models Revisited," 2004.

AAM parameter estimation: shape

- Iterative scheme

$$E(\mathbf{s},\mathbf{t}) = \sum_{\mathbf{x}} \left[I(\mathcal{S}(\mathbf{x};\mathbf{s})) - \mathcal{T}(\mathbf{x};\mathbf{t}) \right]^2$$

$$E(\mathbf{s}+\mathbf{\Delta s},\mathbf{t}) = E(\mathbf{s},\mathbf{t}) + \mathcal{J}\mathbf{\Delta s} + \frac{1}{2}\mathbf{\Delta s}^T \mathcal{H} \mathbf{\Delta s}$$

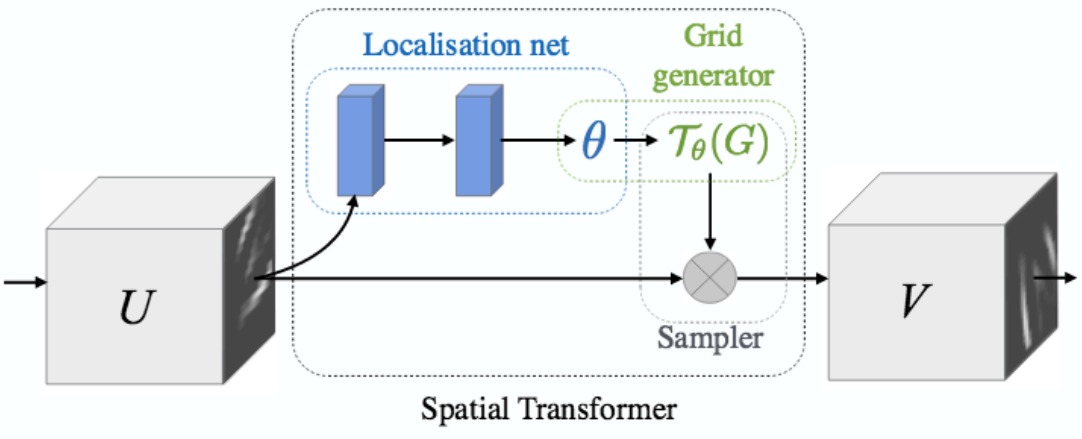
$$\Delta \mathbf{s}^* = -\mathcal{J}\mathcal{H}^{-1} \qquad \mathbf{s}' = \mathbf{s} - \mathcal{J}\mathcal{H}^{-1}$$

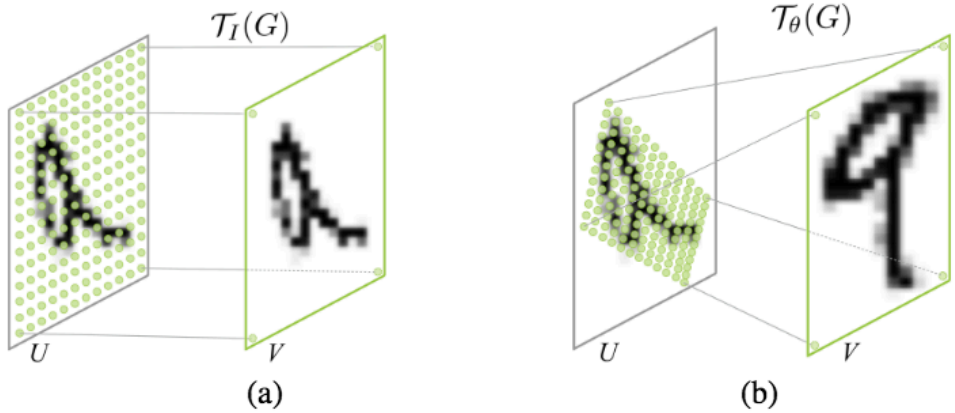
$$I(S(\mathbf{x};\mathbf{s}+\Delta\mathbf{s})) \simeq I(S(\mathbf{x};\mathbf{s})) + \sum_{i=1}^{Ns} \frac{dI}{d\mathbf{s}_i}(\mathbf{x};\mathbf{s})\Delta\mathbf{s}_i$$

$$\frac{dI}{d\mathbf{s}_i}(\mathbf{x};\mathbf{s}) = \frac{\partial I(S(\mathbf{x};\mathbf{s}))}{\partial x}\frac{\partial S_x}{\partial \mathbf{s}_i} + \frac{\partial I(S(\mathbf{x};\mathbf{s}))}{\partial y}\frac{\partial S_y}{\partial \mathbf{s}_i},$$

$$\mathcal{J}_i = \sum_x \left[I(S(\mathbf{x},\mathbf{s})) - T(\mathbf{x},\mathbf{t}) \right] \frac{\mathbf{dI}}{\mathbf{ds_i}}(x) \qquad \mathcal{H}_{i,j} = \sum_x \frac{\mathbf{dI}}{\mathbf{ds_i}}(x)\frac{\mathbf{dI}}{\mathbf{ds_j}}(x)$$

Deformable models, 2015 ('Spatial Transformer Networks')





ure 3: Two examples of applying the parameterised sampling grid to an image U producing the outpu
The sampling grid is the regular grid $G = \mathcal{T}_I(G)$, where I is the identity transformation parameters
sampling grid is the result of warping the regular grid with an affine transformation $\mathcal{T}_\theta(G)$.

STNs, M. Jaderberg, K. Simonyan, A. Zisserman, K. Kavukcuoglu, NIPS 2015

Deformable models, 2015 ('Spatial Transformer Networks')

For clarity of exposition, assume for the moment that $\mathcal{T}_\theta$ is a 2D affine transformation $\mathtt{A}_\theta$. We will discuss other transformations below. In this affine case, the pointwise transformation is

$$\begin{pmatrix} x_i^s \\ y_i^s \end{pmatrix} = \mathcal{T}_\theta(G_i) = \mathtt{A}_\theta \begin{pmatrix} x_i^t \\ y_i^t \\ 1 \end{pmatrix} = \begin{bmatrix} \theta_{11} & \theta_{12} & \theta_{13} \\ \theta_{21} & \theta_{22} & \theta_{23} \end{bmatrix} \begin{pmatrix} x_i^t \\ y_i^t \\ 1 \end{pmatrix} \tag{1}$$

The class of transformations $\mathcal{T}_\theta$ may be more constrained, such as that used for attention

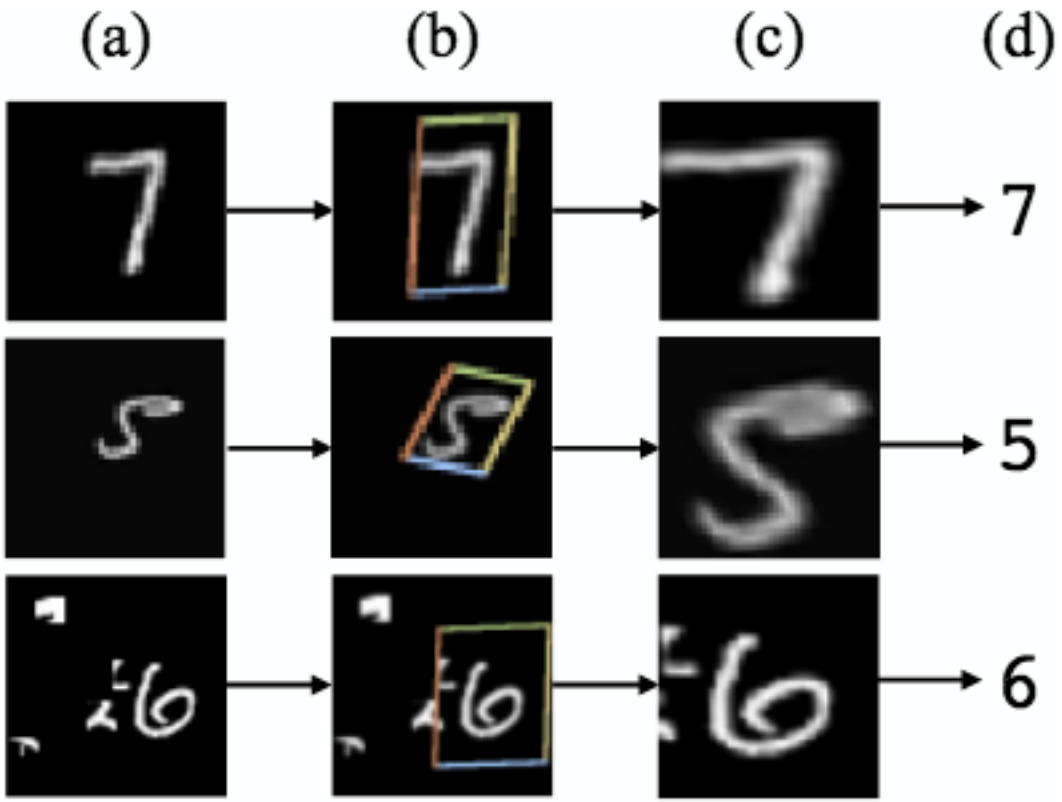
$$\mathtt{A}_\theta = \begin{bmatrix} s & 0 & t_x \\ 0 & s & t_y \end{bmatrix} \tag{2}$$

allowing cropping, translation, and isotropic scaling by varying s, t_x, and t_y. The transformation $\mathcal{T}_\theta$ can also be more general, such as a plane projective transformation with 8 parameters, piecewise affine, or a thin plate spline. Indeed, the transformation can have any parameterised form, provided that it is differentiable with respect to the parameters – this crucially allows gradients to be backpropagated through from the sample points $\mathcal{T}_\theta(G_i)$ to the localisation network output θ. If the

STNs, M. Jaderberg, K. Simonyan, A. Zisserman, K. Kavukcuoglu, NIPS 2015

Deformable models, 2015

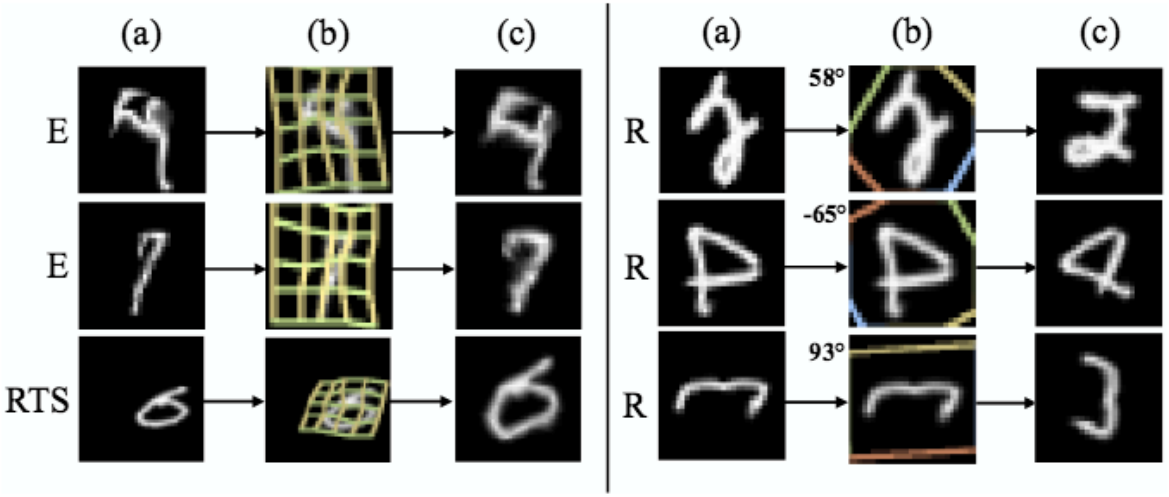
- https://drive.google.com/file/d/0B1nQa_sA3W2iN3RQLXVFRkNXN0k/view



STNs, M. Jaderberg, K. Simonyan, A. Zisserman, K. Kavukcuoglu, NIPS 2015

Deformable models, 2015

Model		MNIST Distortion R	RTS	P	E
FCN		2.1	5.2	3.1	3.2
CNN		1.2	0.8	1.5	1.4
ST-FCN	Aff	1.2	0.8	1.5	2.7
	Proj	1.3	0.9	1.4	2.6
	TPS	1.1	0.8	1.4	2.4
ST-CNN	Aff	0.7	0.5	0.8	1.2
	Proj	0.8	0.6	0.8	1.3
	TPS	0.7	0.5	0.8	1.1



STNs, M. Jaderberg, K. Simonyan, A. Zisserman, K. Kavukcuoglu, NIPS 2015

Deformable models, 2015

Model		
Cimpoi '15 [5]		66.7
Zhang '14 [40]		74.9
Branson '14 [3]		75.7
Lin '15 [23]		80.9
Simon '15 [30]		81.0
CNN (ours)	224px	82.3
2×ST-CNN	224px	83.1
2×ST-CNN	448px	83.9
4×ST-CNN	448px	**84.1**

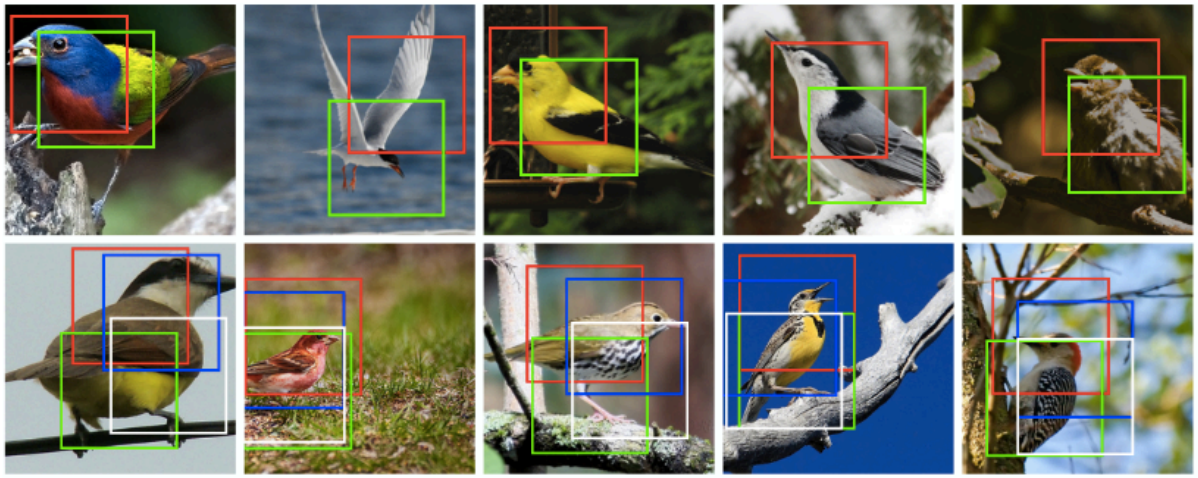


Table 3: *Left:* The accuracy on CUB-200-2011 bird classification dataset. Spatial transformer networks with two spatial transformers (2×ST-CNN) and four spatial transformers (4×ST-CNN) in parallel achieve higher accuracy. 448px resolution images can be used with the ST-CNN without an increase in computational cost due to downsampling to 224px *after* the transformers. *Right:* The transformation predicted by the spatial transformers of 2×ST-CNN (top row) and 4×ST-CNN (bottom row) on the input image. Notably for the 2×ST-CNN, one of the transformers (shown in red) learns to detect heads, while the other (shown in green) detects the body, and similarly for the 4×ST-CNN.

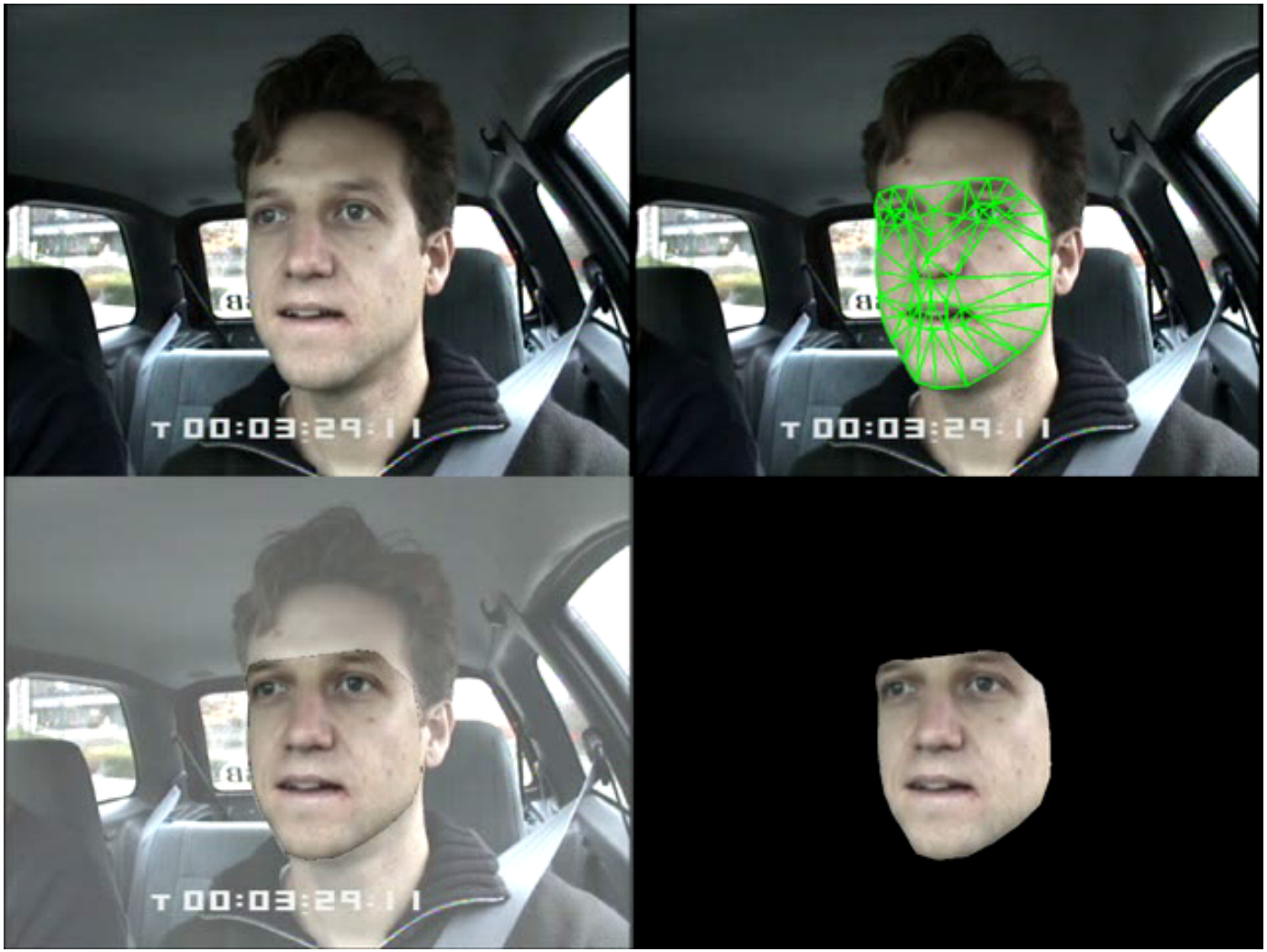
STNs, M. Jaderberg, K. Simonyan, A. Zisserman, K. Kavukcuoglu, NIPS 2015

Deformable models, 2015

- https://drive.google.com/file/d/0B1nQa_sA3W2iN3RQLXVFRkNXN0k/view

STNs, M. Jaderberg, K. Simonyan, A. Zisserman, K. Kavukcuoglu, NIPS 2015

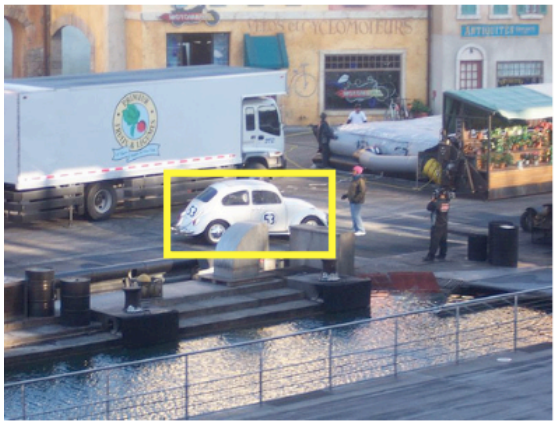
Up to now: template matching (single object)

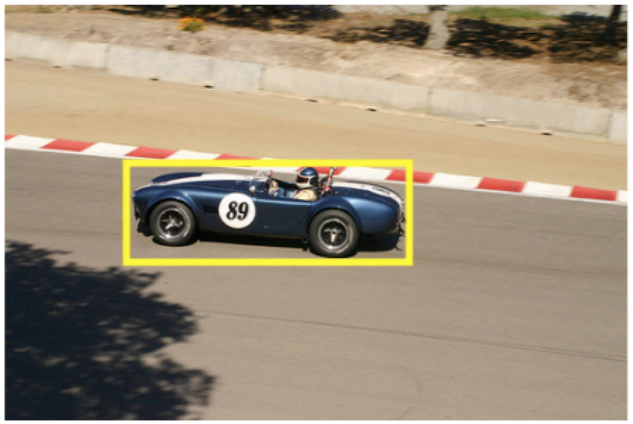


This lecture: detection

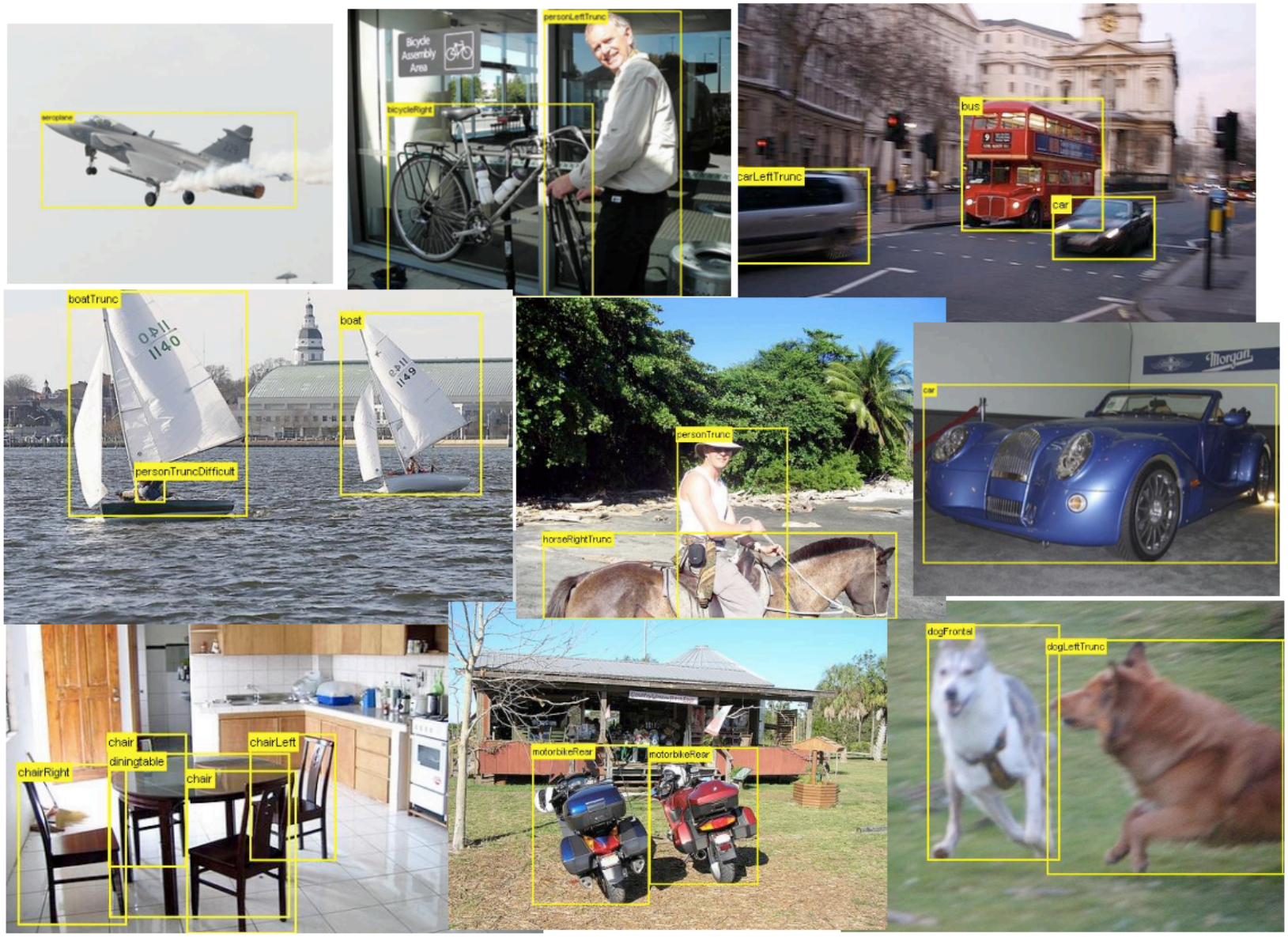
Face Detection, Pose Estimation, and Landmark Localization in the Wild, X. Zhu and D. Ramanan, CVPR 2012

This lecture: detection





PASCAL VOC: 20 classes, 20K images

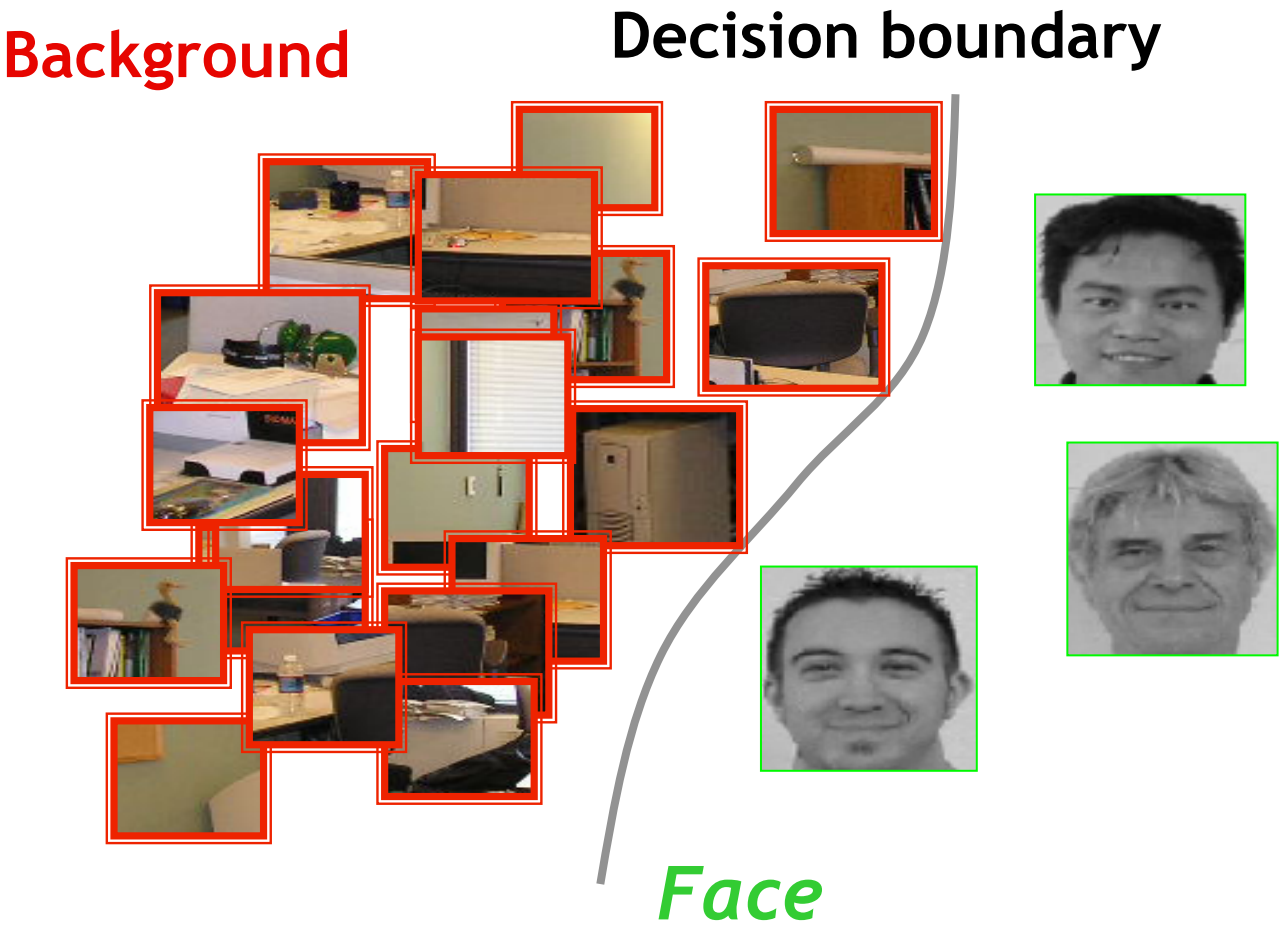


Sliding window approaches

- Scan window over image
  - Multiple scales
  - Multiple orientations
- Classify window as either:
  - Face
  - Non-face

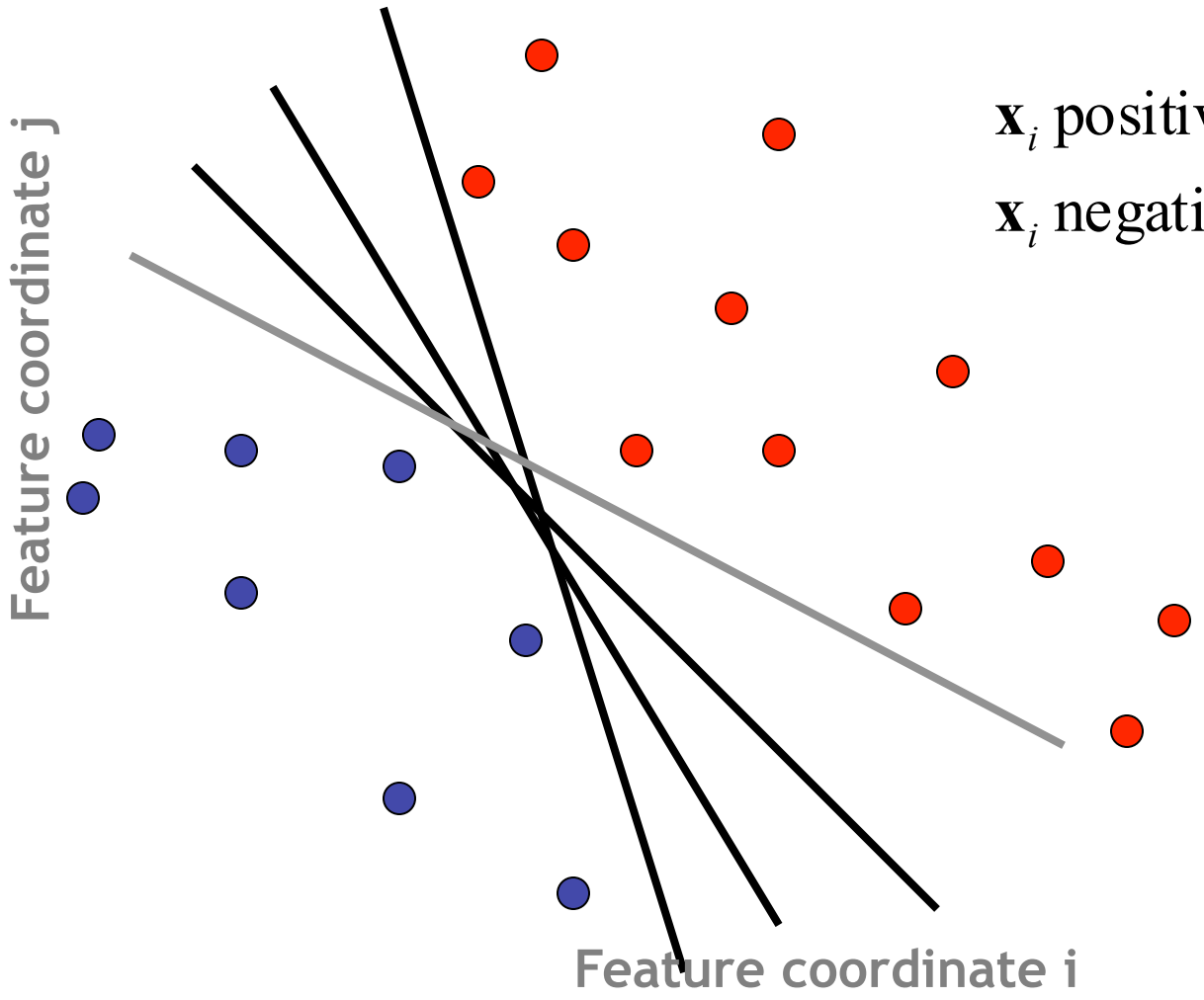
Slide credit: B. Leibe

`Faceness function': classifier



Linear Classifiers

- Find linear expression (*hyperplane*) to separate positive and negative examples



$$\mathbf{x}_i \text{ positive}: \quad \mathbf{x}_i \cdot \mathbf{w} + b \geq 0$$

$$\mathbf{x}_i \text{ negative}: \quad \mathbf{x}_i \cdot \mathbf{w} + b < 0$$

Each data point has a class label:

$$y_t = \begin{cases} +1 & (\bullet) \\ -1 & (\circ) \end{cases}$$

Loss function for linear regression

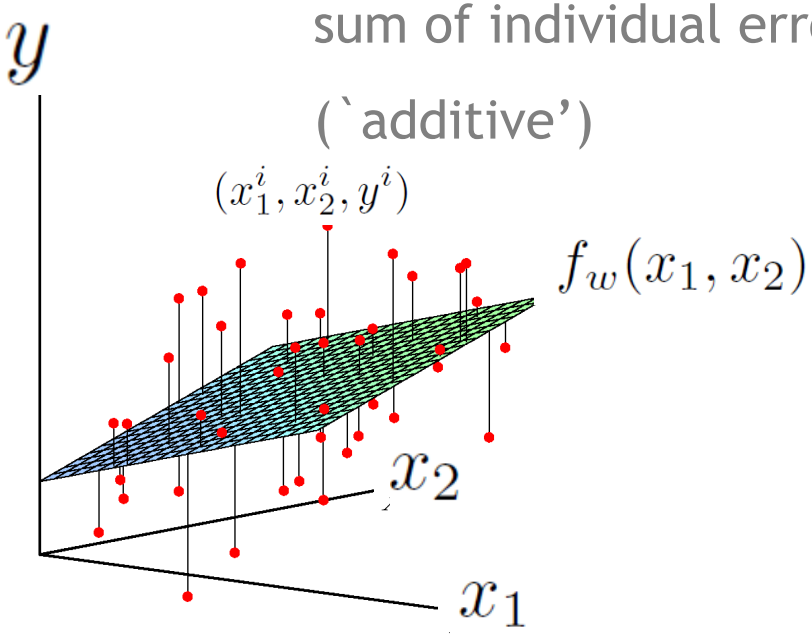
Training: given $S = \{(x^i, y^i)\}\ i = 1, \ldots, N$, estimate optima w

Loss function: quantify appropriateness of w

$$L(S, f_w) = \sum_{i=1}^{N} l(y^i, f_w(x^i)) = \sum_{i=1}^{N} (y^i - < x^i, w >)^2$$

sum of individual errors (`additive')

quadratic



Why this loss function?

Easy to optimize!

Least squares solution for linear regression

Loss function: $L(w) = \sum_{i=1}^{N} (y^i - < x^i, w >)^2$

Introduce vectors and matrixes to rewrite as quadratic expression:

$$\mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix} \quad \mathbf{X} = \begin{bmatrix} x_1^1 & \dots & x_k^1 & \dots & x_K^1 \\ & & \vdots & & \\ x_1^N & \dots & x_k^N & \dots & x_K^N \end{bmatrix} \quad \mathbf{w} = \begin{bmatrix} w_1 \\ \vdots \\ w_k \\ \vdots \\ w_K \end{bmatrix}$$

Residual : $\mathbf{e} = \mathbf{y} - \mathbf{X}\mathbf{w}$

$$L(\mathbf{w}) = \mathbf{e}^T\mathbf{e} = \mathbf{y}^T\mathbf{y} - 2\mathbf{w}^T\mathbf{X}^T\mathbf{y} + \mathbf{w}^T\mathbf{X}^T\mathbf{X}\mathbf{w}$$

$$J(\mathbf{w}) = -2\mathbf{X}^T\mathbf{y} + 2\mathbf{X}^T\mathbf{X}\mathbf{w}$$

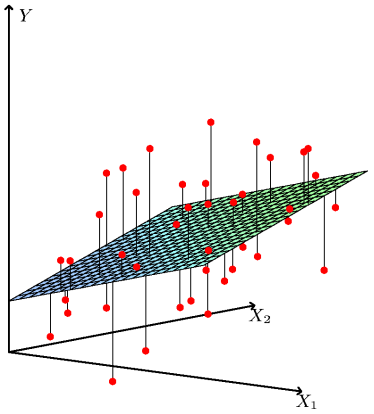
$$\mathbf{w}^* = \arg\min_{\mathbf{w}} L(\mathbf{w}) = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{y}$$

Probabilistic interpretation of least squares

Outputs: continuous random variables

$$y = \langle \mathbf{x}, \mathbf{w} \rangle + \epsilon, \quad \epsilon \sim N(0, \sigma)$$

linear function of inputs + zero mean Gaussian r.v.



$$P(y|\mathbf{x}, \mathbf{w}) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y - < \mathbf{x}, \mathbf{w} >)^2}{2\sigma^2}\right)$$

Optimal w: maximizes likelihood of observed outputs

$$C(\mathbf{w}) = \log P(\mathbf{y}|\mathbf{X}, \mathbf{w}) = \sum_{i=1}^{N} \log P(y^i|\mathbf{x}^i, \mathbf{w})$$

$\alpha = -\frac{1}{2\sigma^2}$ $c = -N/2 \log(2\pi\sigma^2)$

$$= c + a \sum_{i=1}^{N} (y^i - \mathbf{x}^i\mathbf{w})^2 = c + a\,(\mathbf{y} - \mathbf{X}\mathbf{w})^T (\mathbf{y} - \mathbf{X}\mathbf{w})$$

$$C(\mathbf{w}) = -|\alpha| L(\mathbf{w}) + c$$

But the labels are discrete!

A probabilistic criterion for training a classifier

Training set: $\left\{(\mathbf{x}^1, y^1), \ldots, (\mathbf{x}^N, y^N)\right\}, \quad \mathbf{x} \in R^M, \quad y \in \{0, 1\}$

y: discrete observations: model as samples from Bernoulli distribution

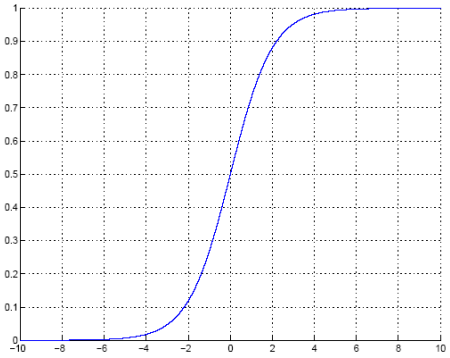
$$P(y = 1|\mathbf{x}, \mathbf{w}) = f(\mathbf{x}, \mathbf{w})$$
$$P(y = 0|\mathbf{x}, \mathbf{w}) = 1 - f(\mathbf{x}, \mathbf{w})$$
$$P(y|\mathbf{x}) = (f(\mathbf{x}, \mathbf{w}))^y \, (1 - f(\mathbf{x}, \mathbf{w}))^{1-y}$$

Find w that maximizes the likelihood of labels in the training set

$$-L(\mathbf{w}) = C(\mathbf{w}) = \log P(\mathbf{y}|\mathbf{X}, \mathbf{w})$$
$$= \sum_i y^i \log f(\mathbf{x}^i, \mathbf{w}) + (1 - y^i) \log(1 - f(\mathbf{x}^i, \mathbf{w}))$$

Sigmoidal function & logistic regression

$$P(y=1|\mathbf{x},\mathbf{w}) = f(\mathbf{x},\mathbf{w}) \;= g(<\mathbf{x},\mathbf{w}>) \;= \frac{1}{1+\exp(-<\mathbf{x},\mathbf{w}>)}$$



$$g(a) = \frac{1}{1+\exp(-a)}$$ sigmoidal

Training criterion from previous slide:

$$C(\mathbf{w}) = \sum_{i=1}^{N} y^i \log g(<\mathbf{x}^i,\mathbf{w}>) + (1-y^i)\log\left(1-g(<\mathbf{x}^i,\mathbf{w}>)\right)$$

A compact expression for the loss

Using $h_{\mathbf{w}}(\mathbf{x}) =< \mathbf{x}, \mathbf{w} >$,$y_{\pm} = 2y_b - 1$:

$$P(y = 1|h_{\mathbf{w}}(\mathbf{x})) = \frac{1}{1 + \exp(-h_{\mathbf{w}}(\mathbf{x}))}$$

$$P(y = -1|h_{\mathbf{w}}(\mathbf{x})) = 1 - P(y = 1|h_{\mathbf{w}}(\mathbf{x})) = 1 - \frac{1}{1 + \exp(-h_{\mathbf{w}}(\mathbf{x}))}$$

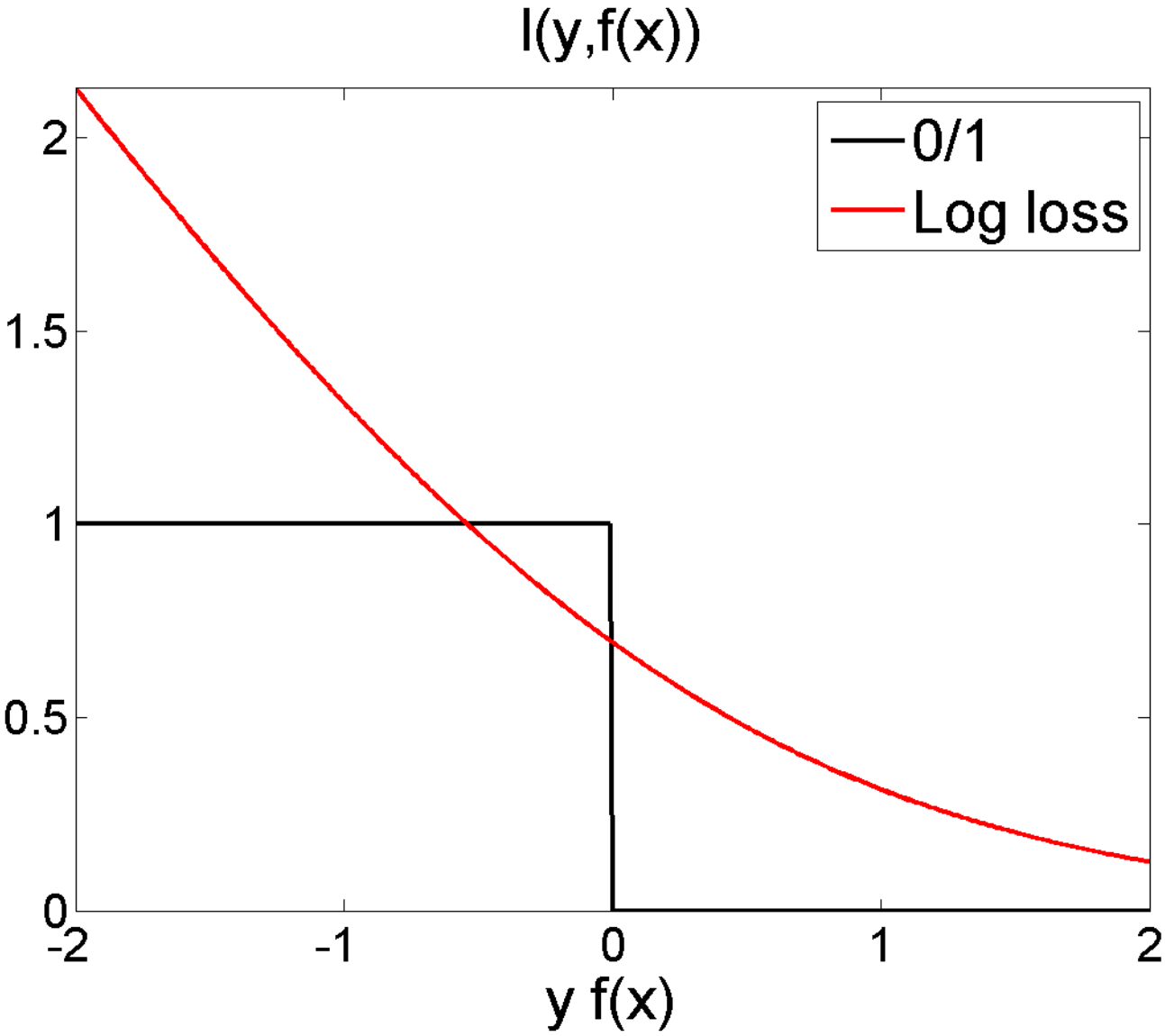
$$= \frac{\exp(-h_{\mathbf{w}}(\mathbf{x}))}{1 + \exp(-h_{\mathbf{w}}(\mathbf{x}))} = \frac{1}{\exp(h_{\mathbf{w}}(\mathbf{x})) + 1}$$

$$P(y = y^i|h_{\mathbf{w}}(\mathbf{x}^i)) = \frac{1}{1 + \exp(-y^i h_{\mathbf{w}}(\mathbf{x}^i))}$$

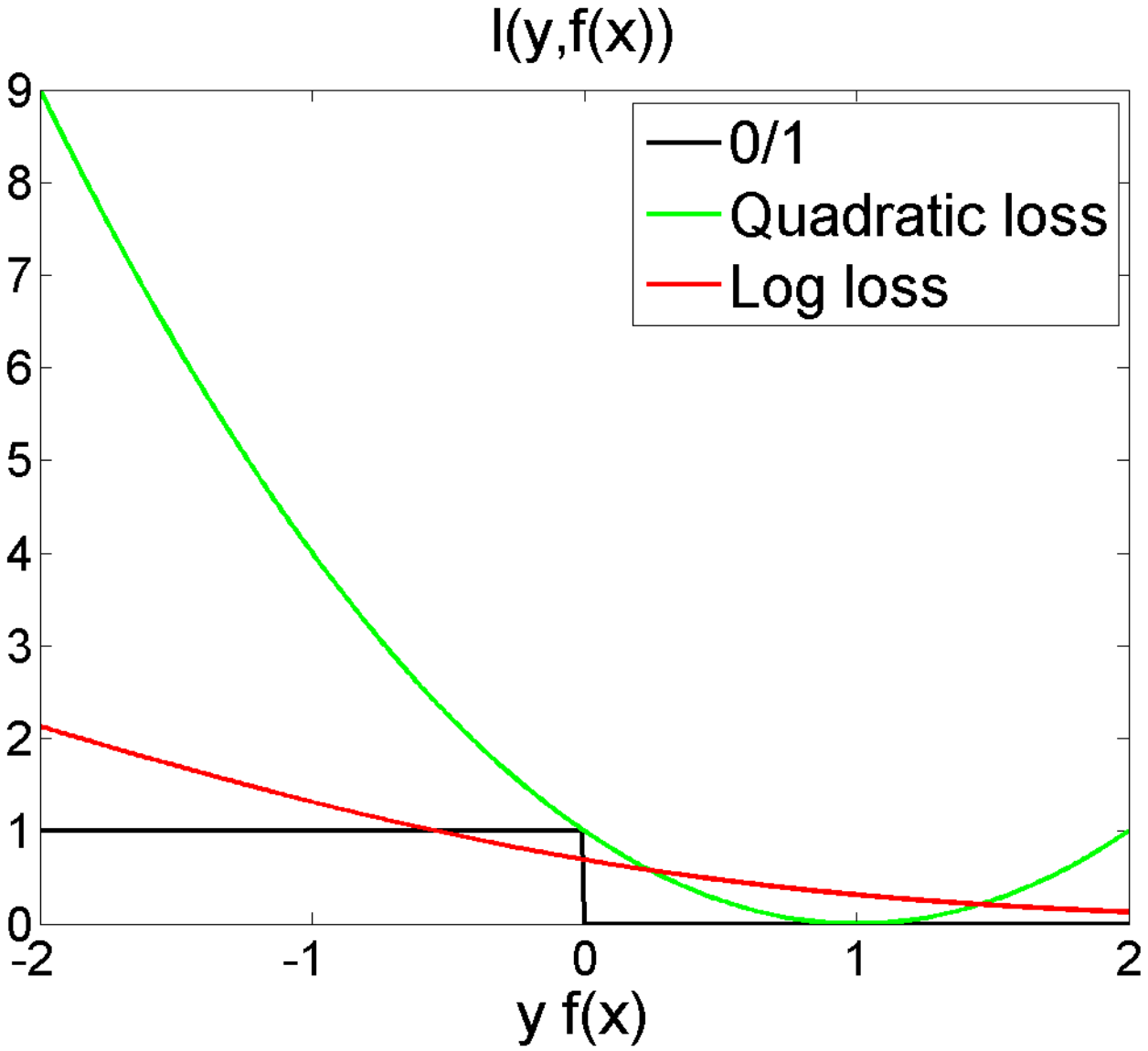
$$L(\mathbf{w}) = -C(\mathbf{w}) = \sum_{i=1}^{N} -\log P(y = y^i|h_{\mathbf{w}}(\mathbf{x}^i))$$

$$= \sum_{i=1}^{N} \log(1 + \exp(-y^i h_{\mathbf{w}}(\mathbf{x}^i)))$$

Log loss: $l(y, f(x)) = \log(1 + \exp(-yf(x)))$

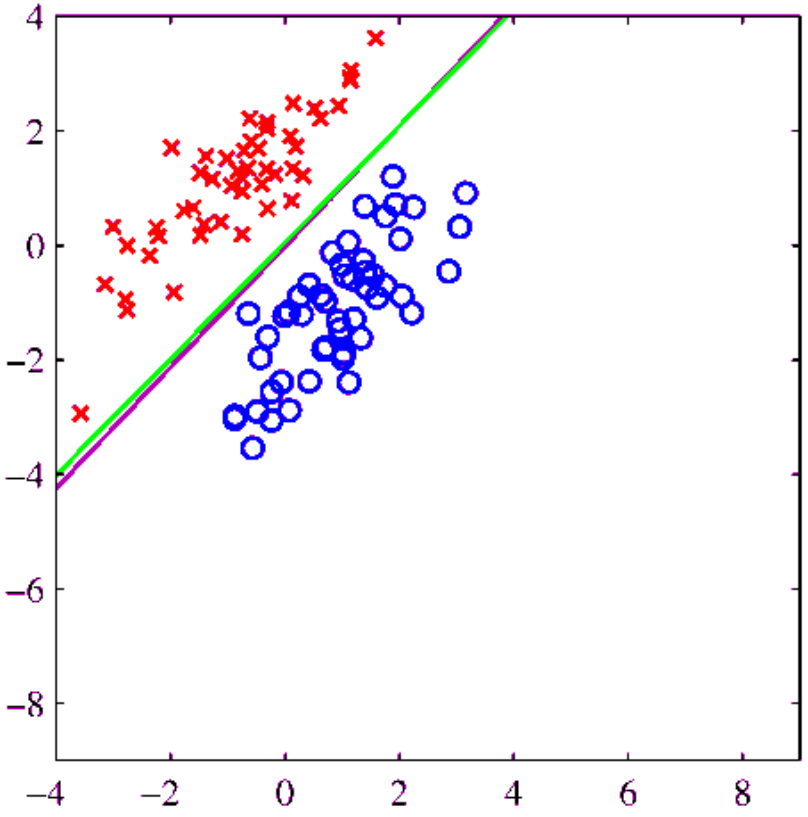


Log loss vs. quadratic loss



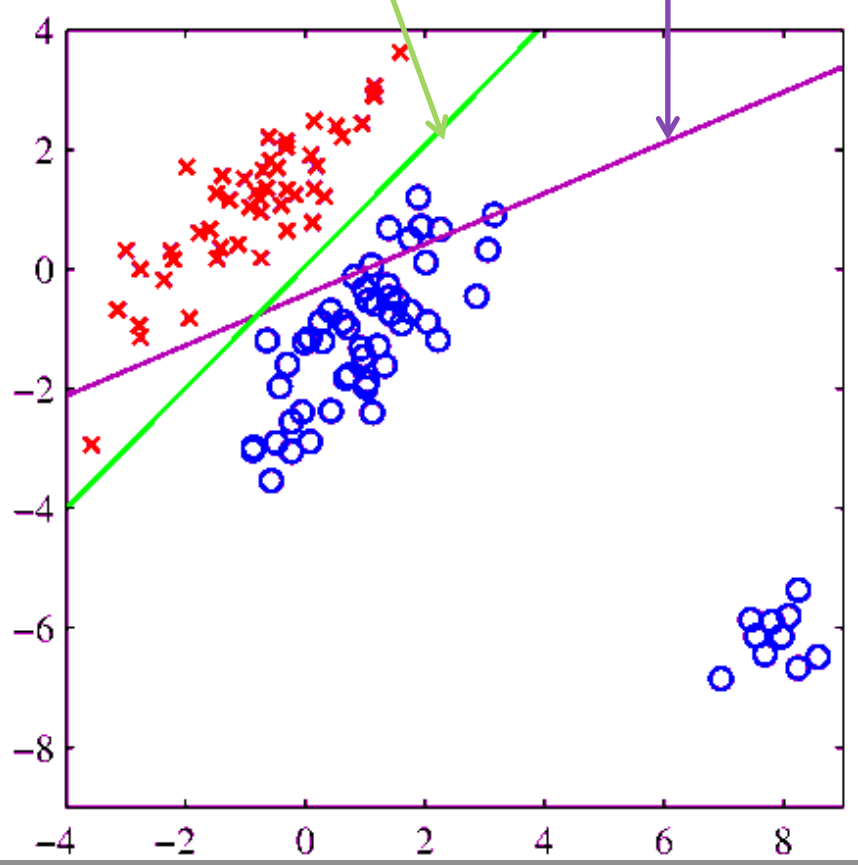
Logistic vs Linear Regression

Logistic regression is more robust

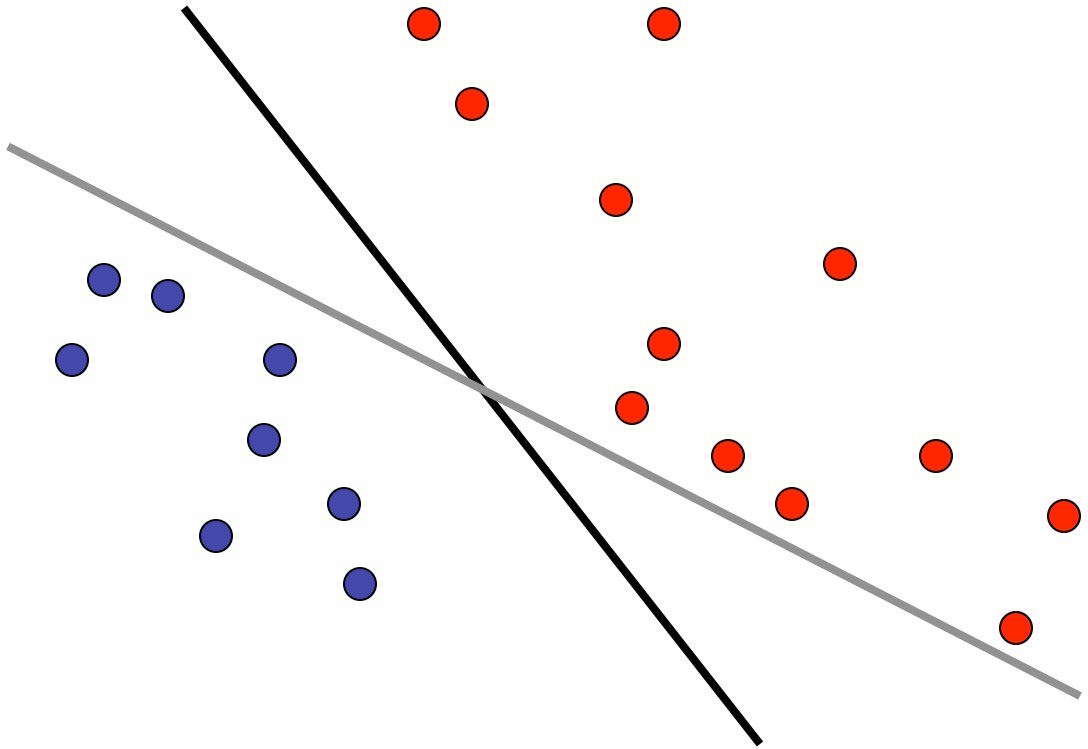


Linear Regression

Logistic Regression



Which classifier is best?



All points should lie clearly on the correct side of the boundary

How can we quantify this?

How can we enforce this?

Functional Margins

Consider Logistic Regression:

$$P(y = 1|x; w) = g(\mathbf{w}^T x) = \frac{1}{1 + \exp(-\mathbf{w}^T x)}$$

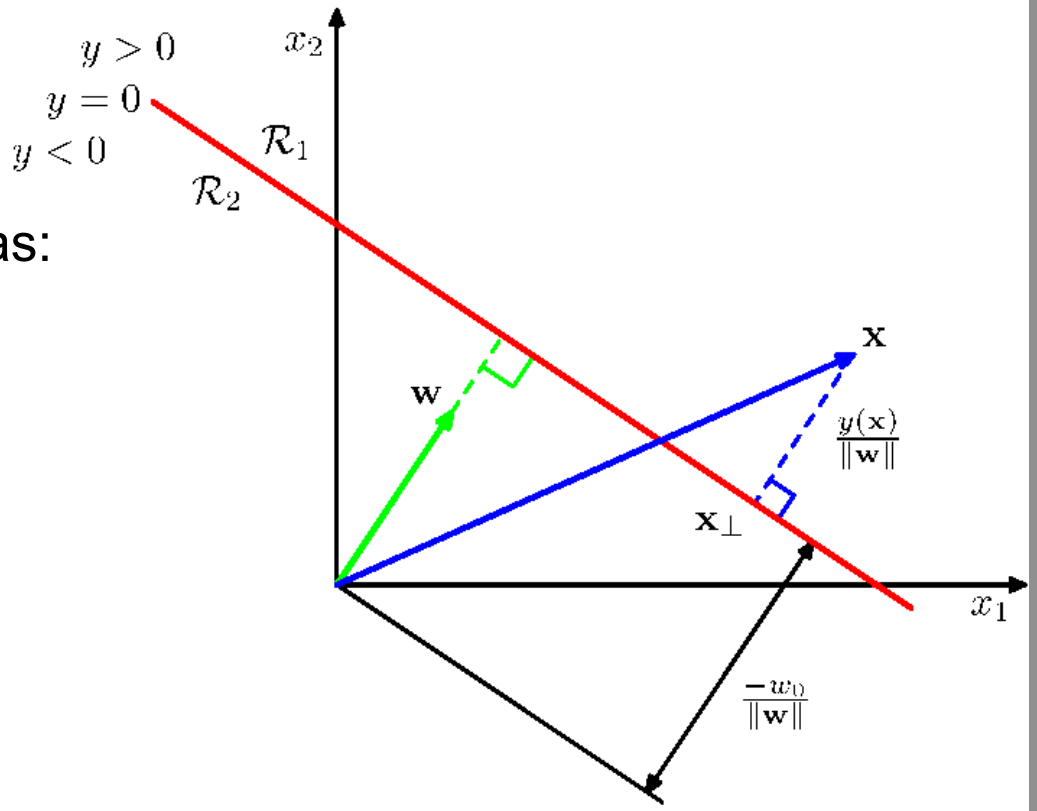
Ideally:
$$(\mathbf{w}^T x^i) \gg 0 \quad \text{if} \quad y^i = 1$$
$$(\mathbf{w}^T x^i) \ll 0 \quad \text{if} \quad y^i = -1$$

Put together: $y^i(\mathbf{w}^T x^i) \gg 0$

`functional margin'

Problem: scaling w changes functional margin, but not decision boundary

Geometric Margins



- Express point in feature space as:

$$\mathbf{x} = \mathbf{x}_\perp + \gamma \frac{\mathbf{w}}{|\mathbf{w}|}$$

- γ: distance from hyperplane

$$\mathbf{w}^T\mathbf{x} = \mathbf{w}^T\mathbf{x}_\perp + \mathbf{w}^T\gamma\frac{\mathbf{w}}{|\mathbf{w}|}$$

$$\mathbf{w}^T\mathbf{x} = -b + \gamma|\mathbf{w}|$$

$$\gamma = \frac{\mathbf{w}^T\mathbf{x} + b}{|\mathbf{w}|} = \frac{\mathbf{w}^T}{|\mathbf{w}|}\mathbf{x} + \frac{b}{|\mathbf{w}|}$$

- Make positive also for negative class:

$$\gamma^i = y^i\left(\frac{\mathbf{w}^T}{|\mathbf{w}|}\mathbf{x}^i + \frac{b}{|\mathbf{w}|}\right)$$

Margin-based Optimization

Least margin:

$$\gamma = \min_i \gamma^i$$

Optimization problem:

$$\begin{aligned} \max_{\gamma,\mathbf{w},b} \quad & \gamma \\ s.t. \quad & y^i(\mathbf{w}^T x^i + b) \geq \gamma, \quad i = 1 \ldots M \\ & |\mathbf{w}| = 1 \end{aligned}$$

Problem: non-convex constraint

Functional margin:

$$\begin{aligned} \max_{\gamma,\mathbf{w},b} \quad & \frac{\gamma'}{|\mathbf{w}|} \\ s.t. \quad & y^i(\mathbf{w}^T x^i + b) \geq \gamma', \quad i = 1 \ldots M \end{aligned}$$

Margin-Based Optimization (cont.d)

Rewrite last problem:

$$\max_{\gamma,\mathbf{w},b} \quad \frac{\gamma'}{|\mathbf{w}|}$$
$$s.t. \quad y^i(\mathbf{w}^T x^i + b) \geq \gamma', \quad i = 1 \ldots M$$

$$\max_{\mathbf{w},b} \quad \frac{1}{|\mathbf{w}|}$$
$$s.t. \quad y^i(\mathbf{w}^T x^i + b) \geq 1, \quad i = 1 \ldots M$$

Equivalent:

$$\min_{\mathbf{w},b} \quad \frac{1}{2}|\mathbf{w}|^2$$
$$s.t. \quad y^i(\mathbf{w}^T x^i + b) \geq 1, \quad i = 1 \ldots M$$

Quadratic programming problem

Quadratic cost

Linear constraints

Large Margin Classifiers

Goal: Maximize least margin: $\gamma = \min_i \gamma^i$

Optimization problem:

$$\max_{\gamma,\mathbf{w},b} \quad \gamma$$
$$s.t. \quad y^i(\mathbf{w}^T x^i + b) \geq \gamma, \quad i = 1 \ldots M$$
$$|\mathbf{w}| = 1$$

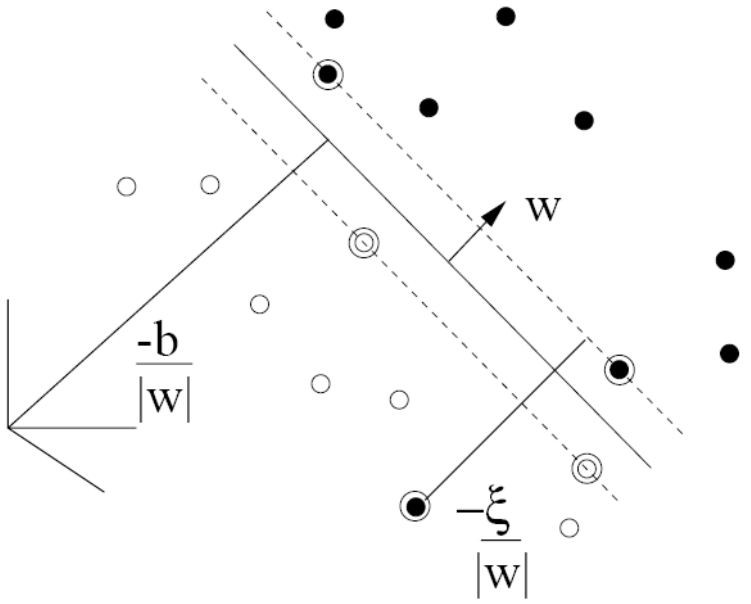
Equivalent problem:

$$\min_{\mathbf{w},b} \quad \frac{1}{2}|\mathbf{w}|^2$$
$$s.t. \quad y^i(\mathbf{w}^T x^i + b) \geq 1, \quad i = 1 \ldots M$$

Accounting for non-separable data

Introduce positive `slack' variables:

$$\begin{aligned} \mathbf{w}^T x^i + b \geq & \quad 1 - \xi^i, \qquad y_i = 1 \\ \mathbf{w}^T x^i + b \leq & \quad -1 + \xi^i, \qquad y_i = -1 \end{aligned}$$



Equivalently: $y^i(\mathbf{w}^T x^i + b) \geq 1 - \xi^i$

If $\xi^i > 1$ an error occurs

$\sum_i \xi^i$:upper bound on number of errors

New optimization problem

$$\begin{aligned} \min_{\mathbf{w},b} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{M} \xi^i \\ s.t. \quad & y^i(\mathbf{w}^T x^i + b) \geq 1 - \xi^i \\ & \xi^i \geq 0 \end{aligned}$$

Loss function

Optimization problem:

$$\begin{aligned} \min_{\mathbf{w},b} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N}\xi^i \\ s.t. \quad & y^i(\mathbf{w}^T x^i + b) \geq 1 - \xi^i \\ & \xi^i \geq 0 \end{aligned}$$

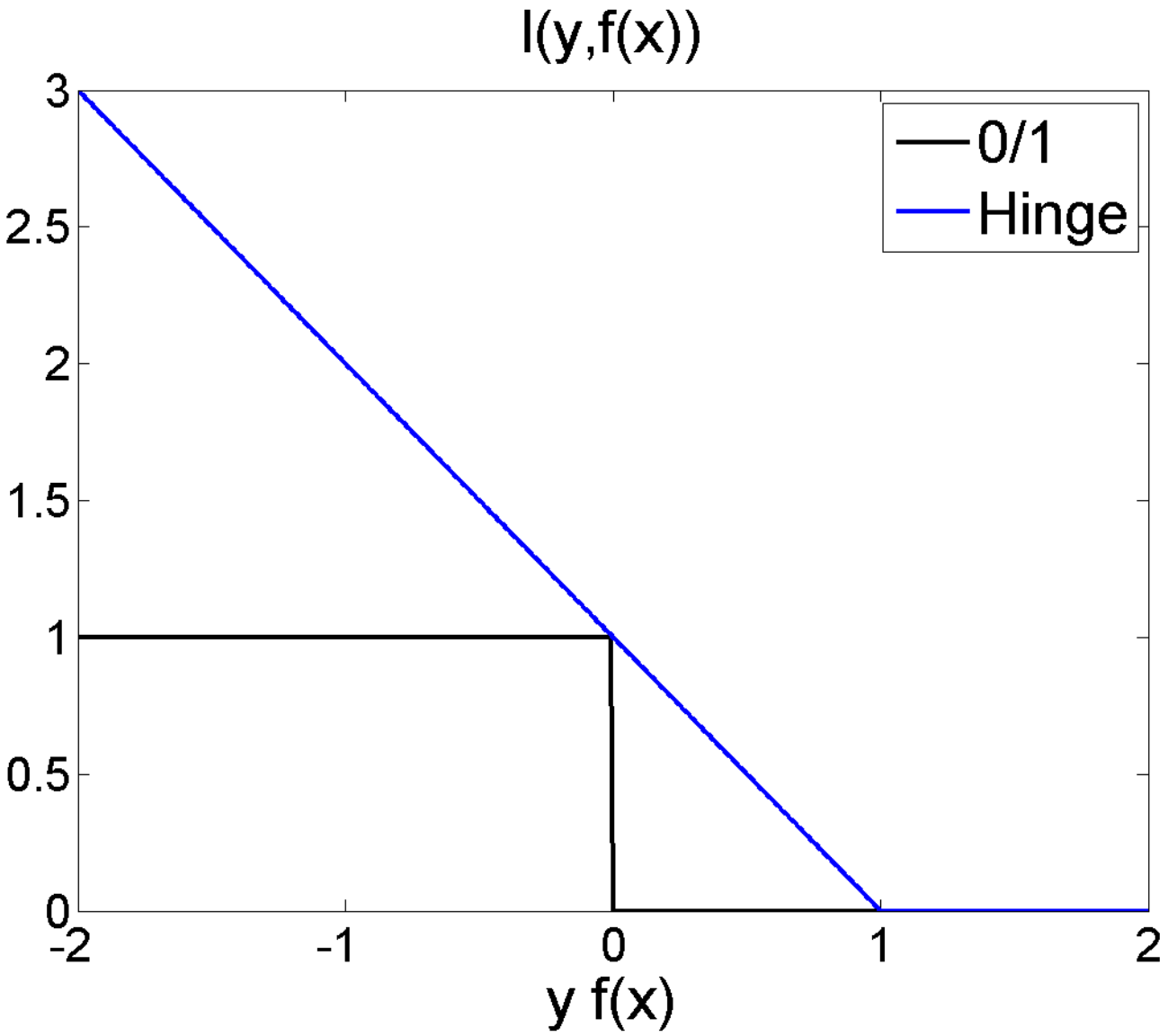
At optimum :

$$\begin{aligned} \xi^i &= \max(0, 1 - y^i(\mathbf{w}^T x^i + b)) \\ &= \max(0, 1 - y^i h_{\mathbf{w},b}(x^i)) \end{aligned}$$

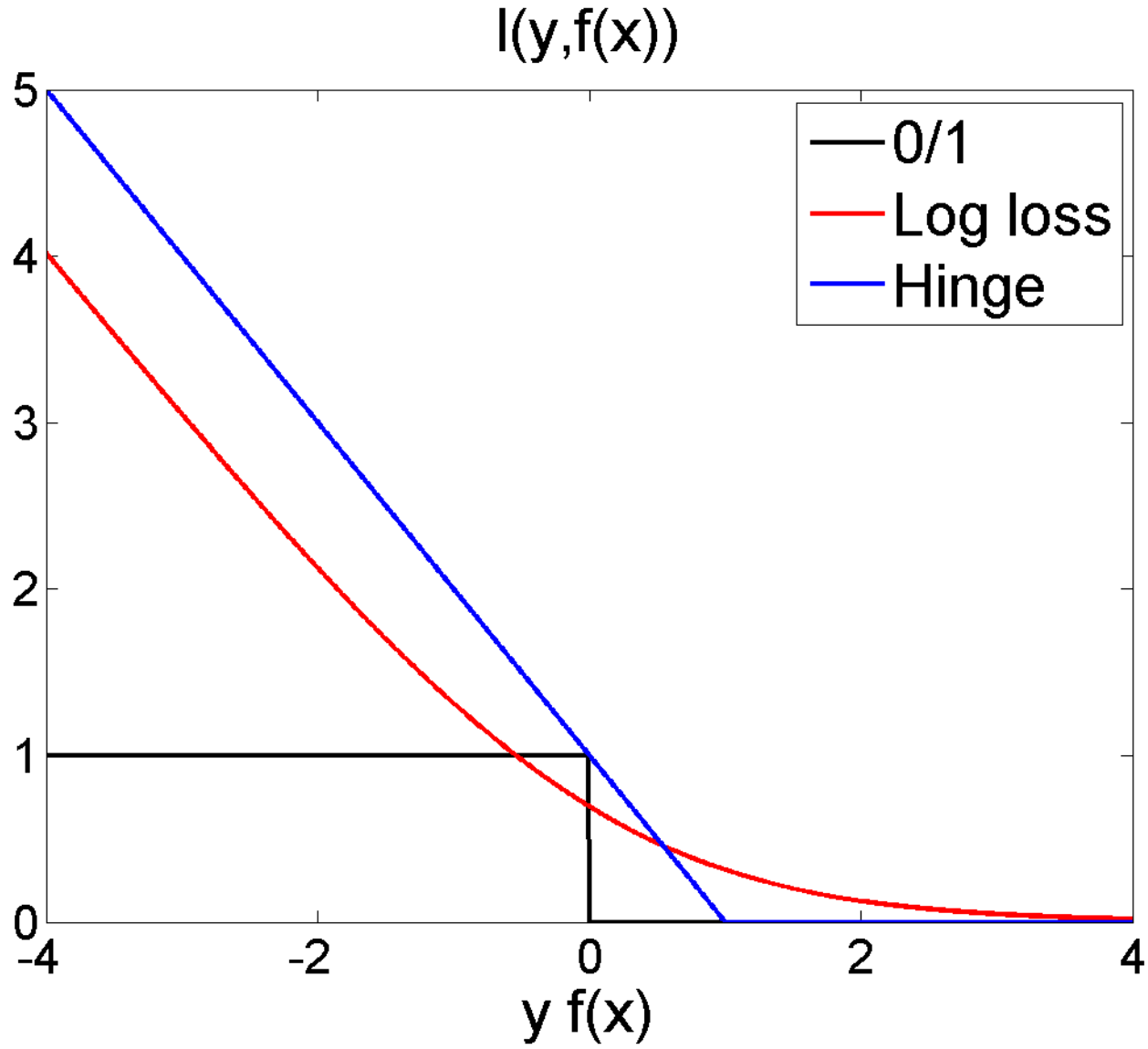
Training criterion:

$$\begin{aligned} L(\mathbf{w}) &= \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N}\max(0, 1 - y^i h_{\mathbf{w},b}(x^i)) \\ &\propto \lambda\|\mathbf{w}\|^2 + \sum_{i=1}^{N}\underbrace{\max(0, 1 - y^i h_{\mathbf{w},b}(x^i))}_{l(y^i, x^i)} \end{aligned}$$

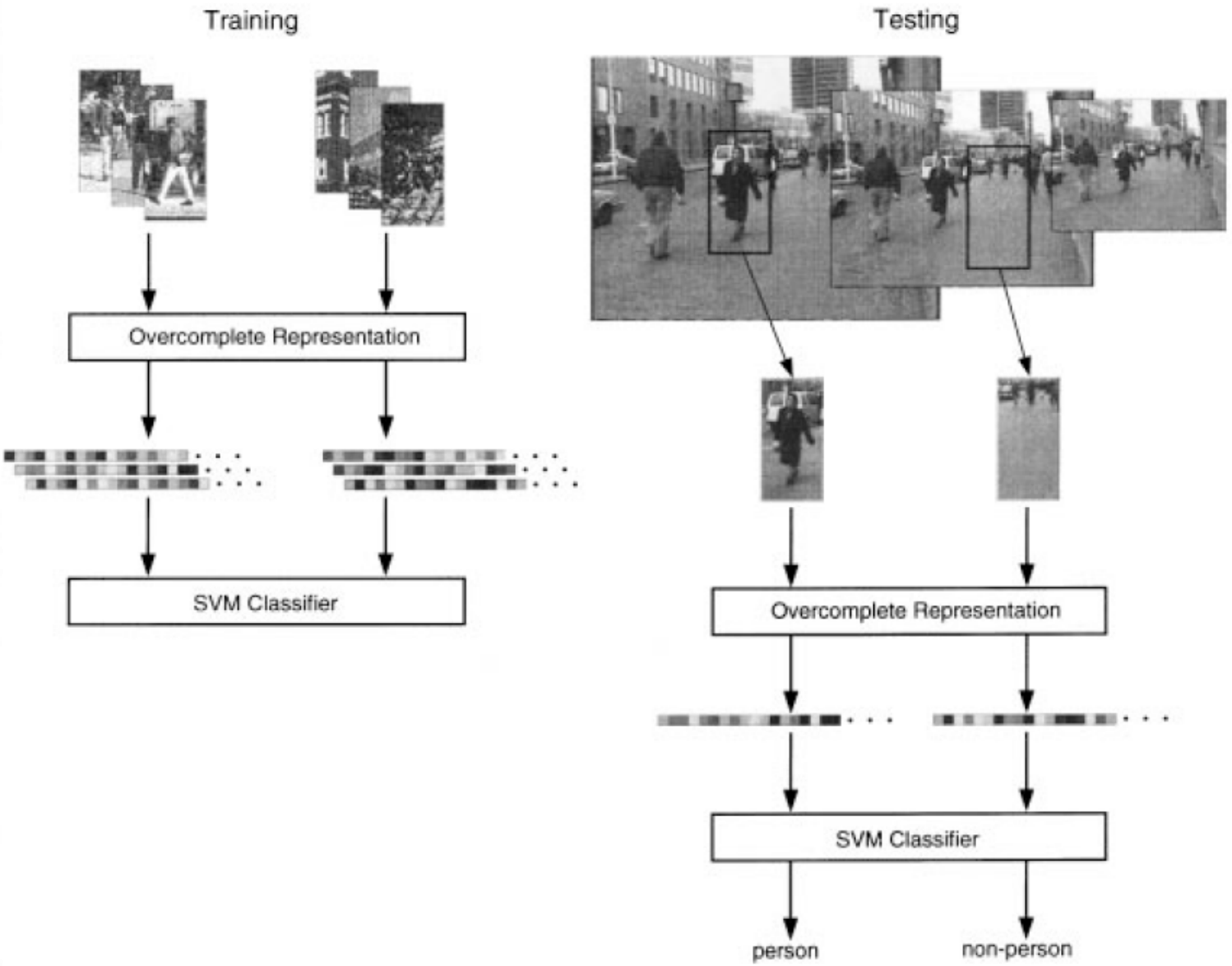
Hinge loss



Hinge loss vs log-loss

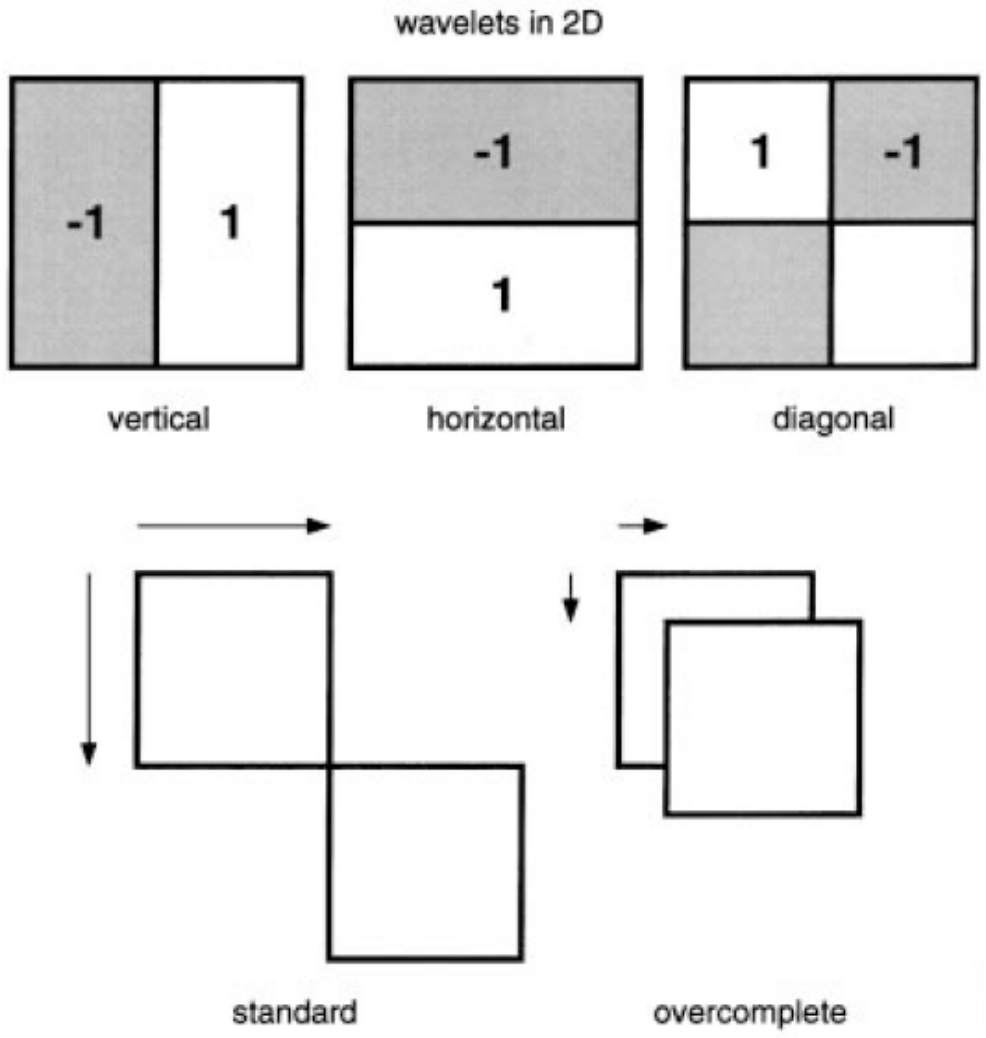


Papageorgiou & Poggio, CVPR 1998

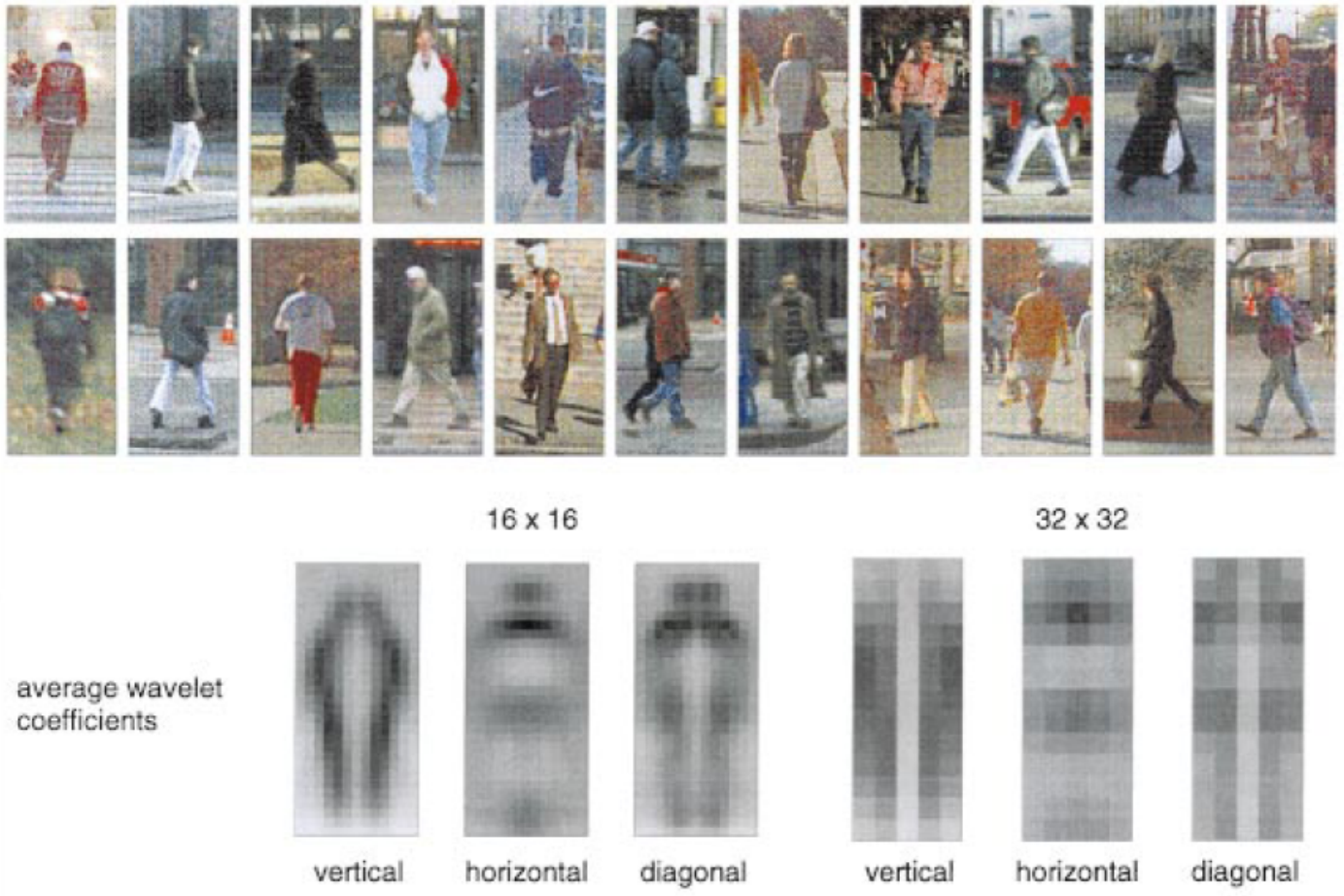


Papageorgiou & Poggio: Image Representation

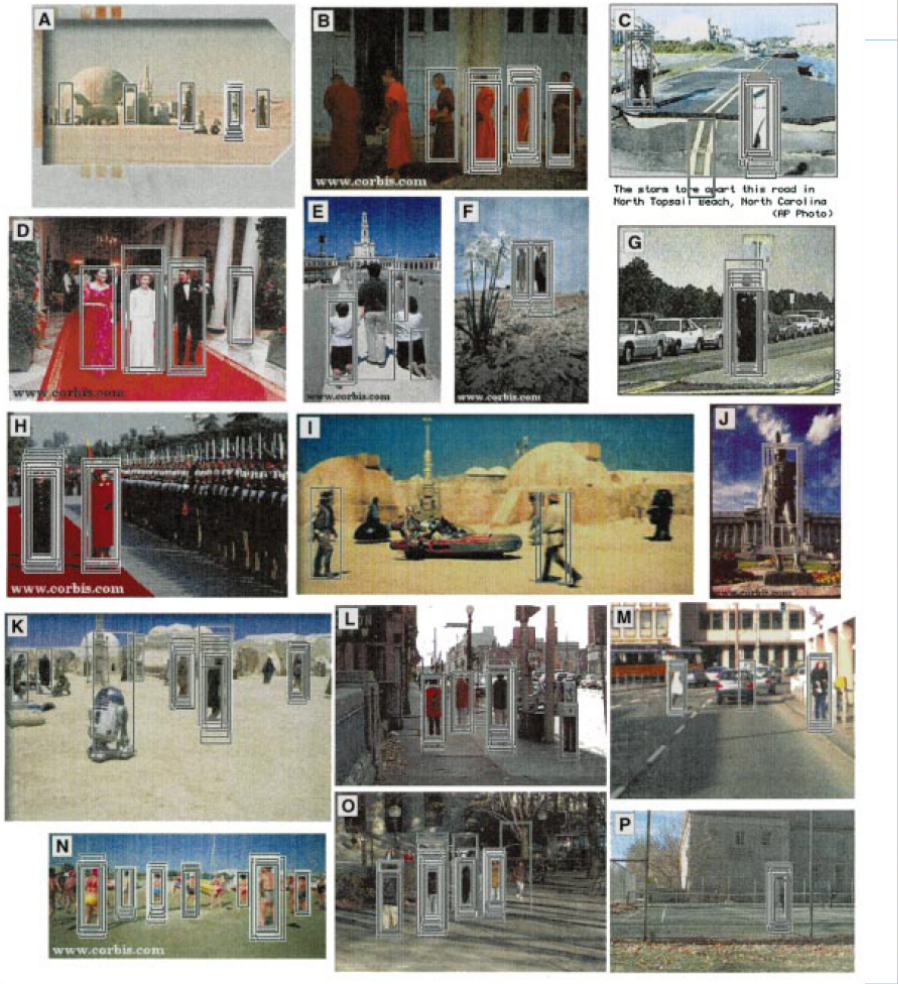
- Haar features

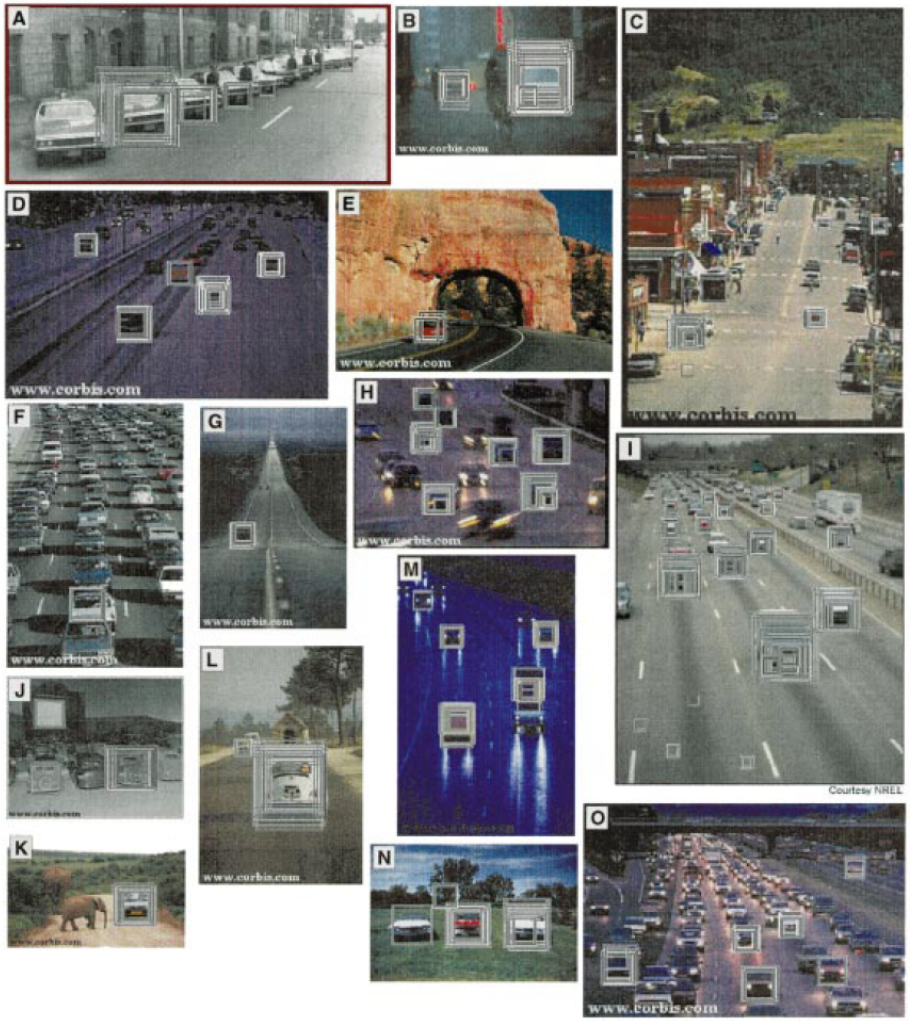


Papageorgiou & Poggio: training set

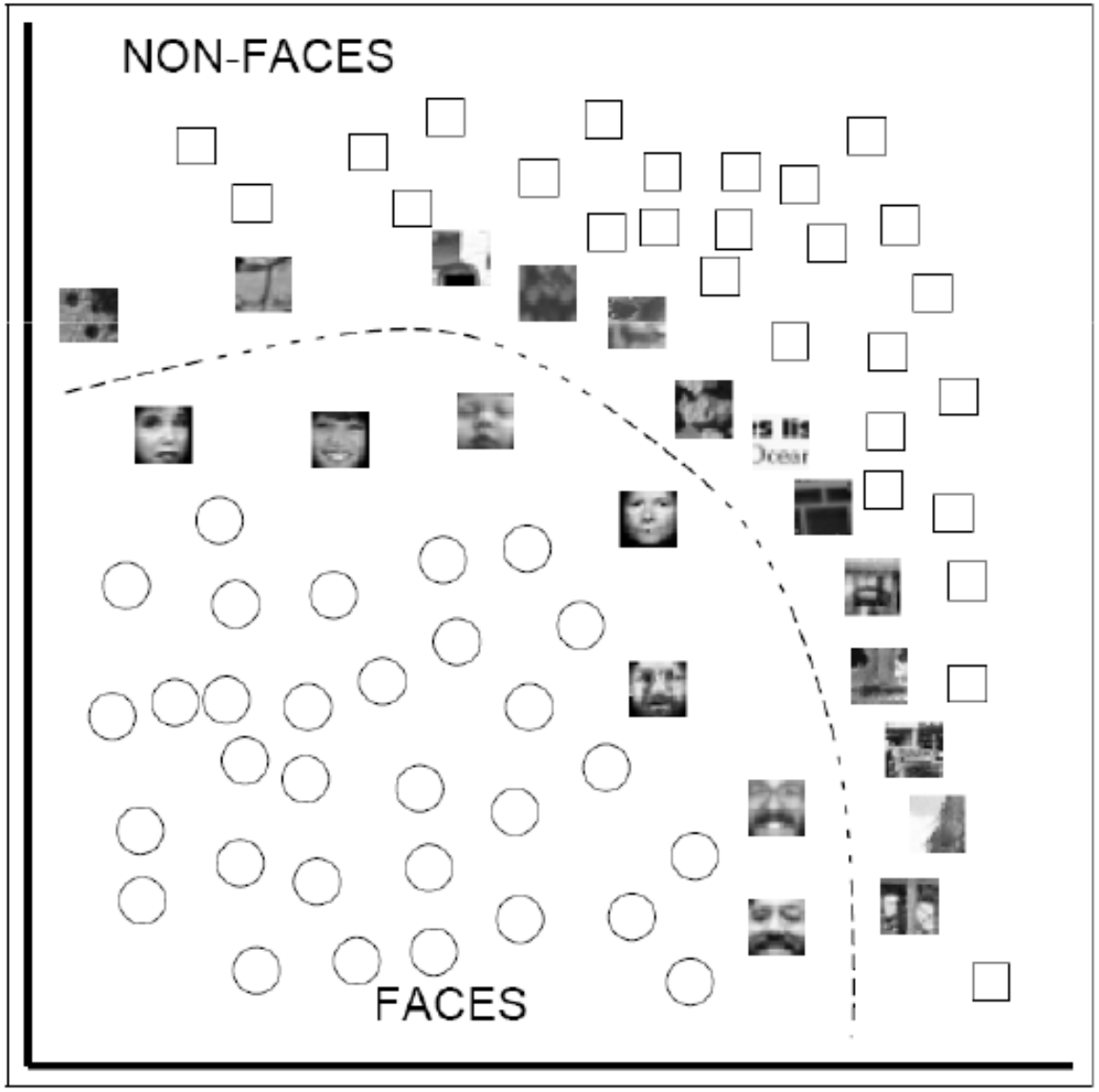


Sample Detections





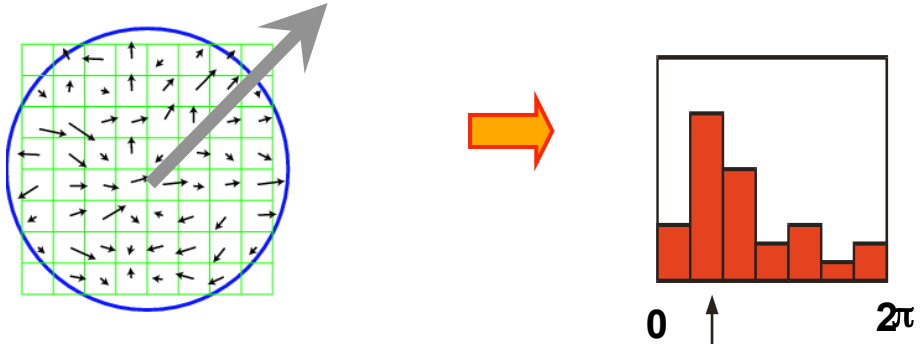
Support vectors for Faces (P&P 98)



Scale-Invariant Feature Transform (SIFT) descriptor

Use location and characteristic scale given by blob detector

Estimate orientation from orientation histogram

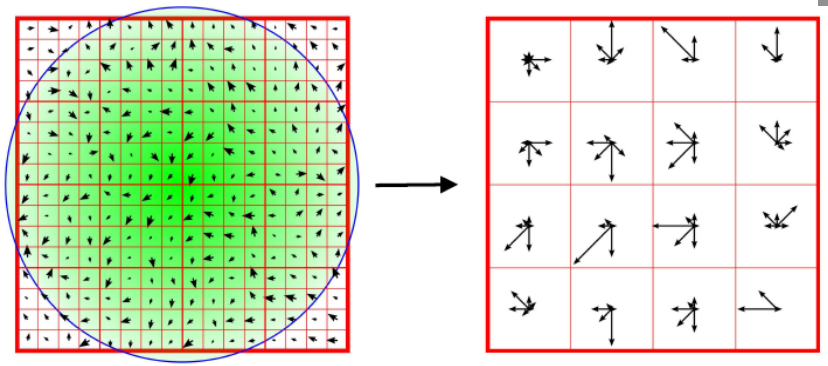


Break patch in 4x4 location blocks

8-bin orientation histogram per block

8x4x4 = 128-D descriptor

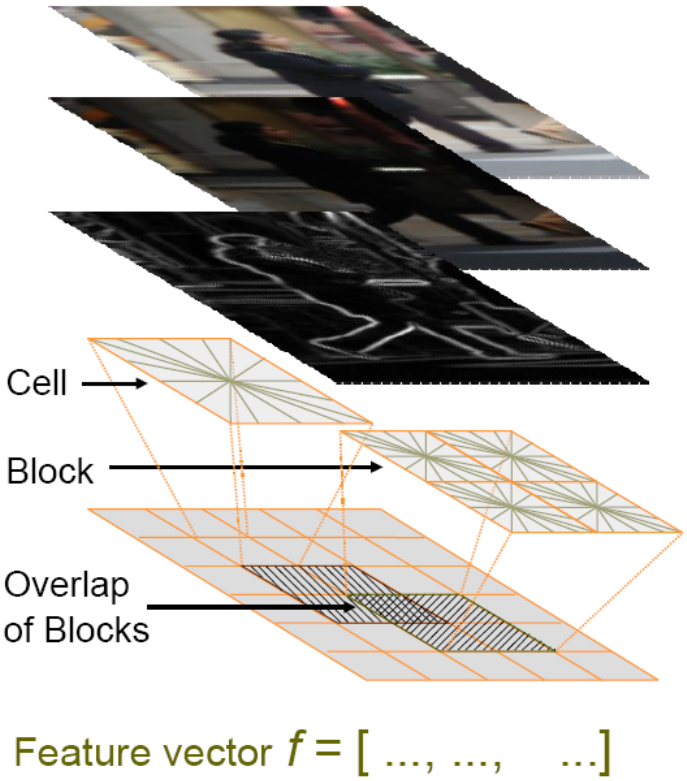
Normalize to unit norm



Invariance to: scale, orientation, multiplicative & additive changes

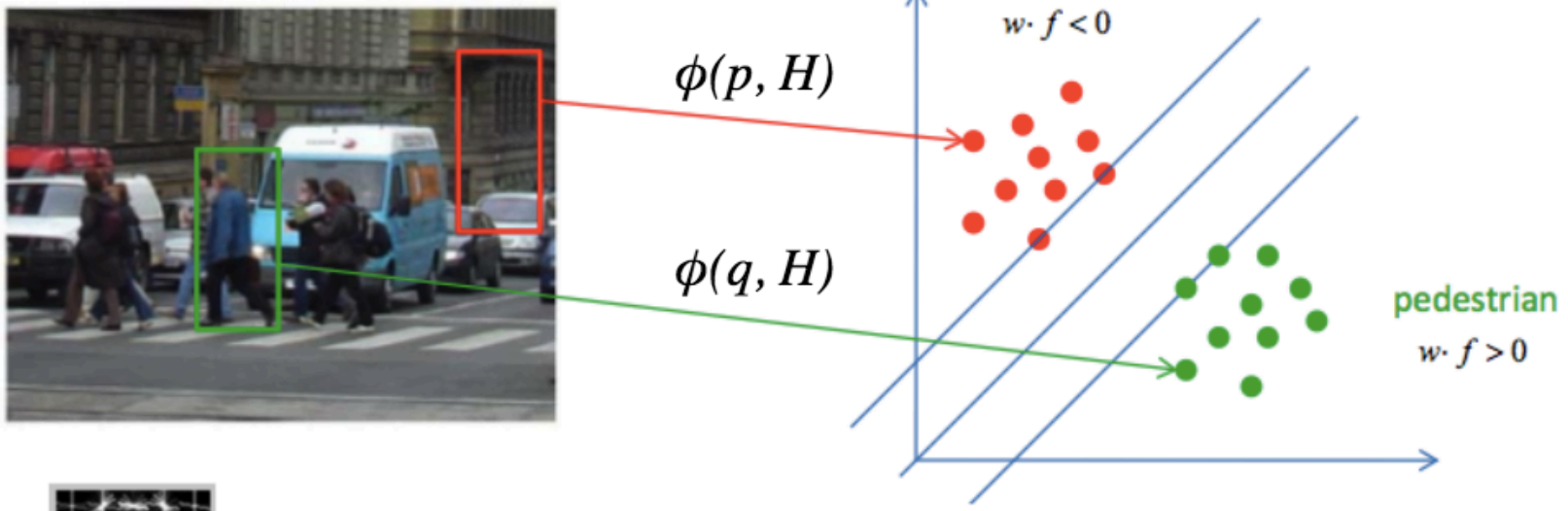
Dalal and Triggs, ICCV 2005

- Histogram of Oriented Gradient (HOG) features
- Highly accurate detection using linear SVM



Feature vector f = [..., ..., ...]

Dalal & Triggs: HOG features + SVMs



Typical form of

There is much more background than objects

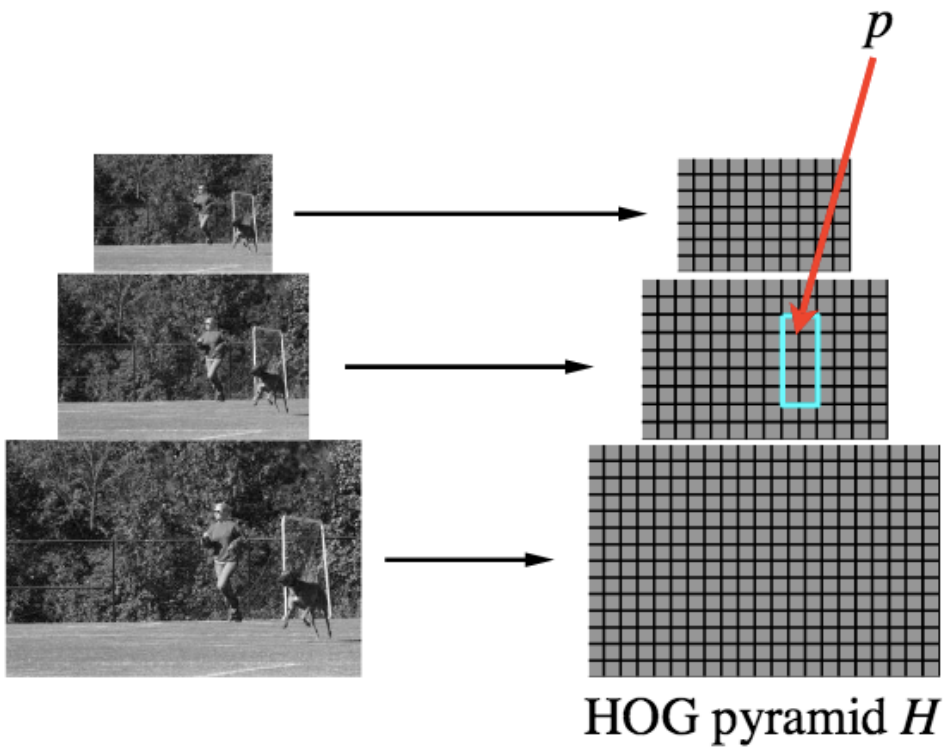
Start with random negatives and repeat:

1) Train a model

2) Harvest false positives to define “hard negatives”

Dallal & Triggs: multi-scale detector

- Array of weights for features in subwindow of HOG pyramid
- Score is dot product of filter and feature vector

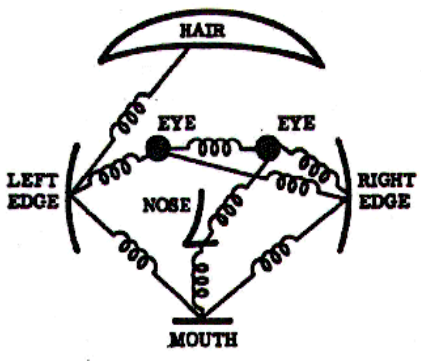


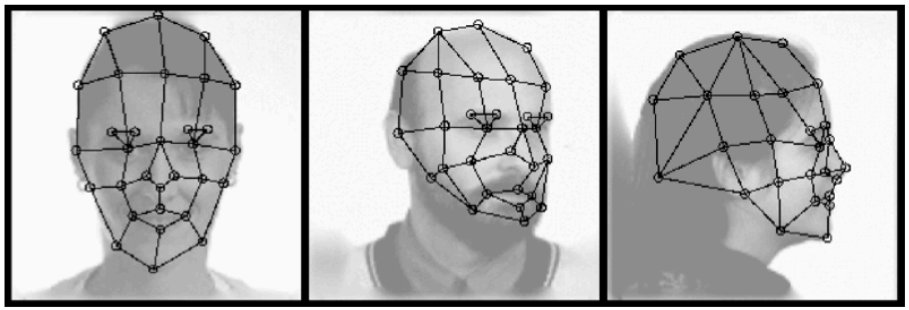
Filter F

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

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

Deformable Part Models (DPMs)



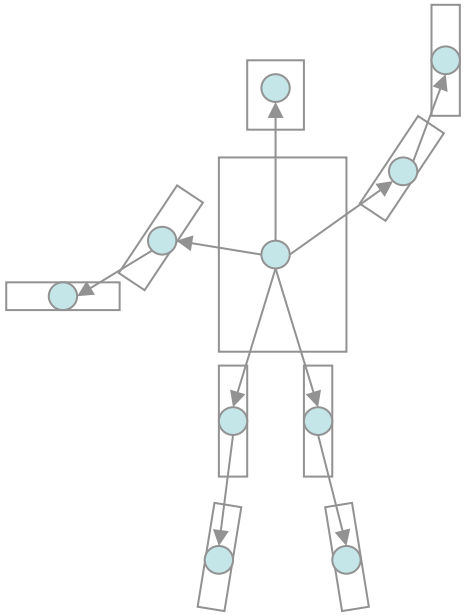


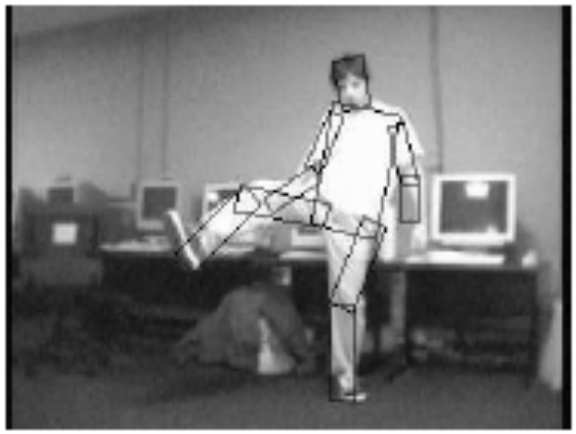
M. Fischler and R. Erschlanger. The Representation and Matching of Pictorial Structures '73.
M. Lades, et al: Distortion Invariant Object Recognition in the Dynamic Link Architecture. '93
Y. Amit, A. Kong: Graphical Templates for Model Registration. '96
A. L. Yuille, J. M. Coughlan: An A perspective on deterministic optimization for deformable templates. '00*
M. C. Burl, P. Perona: Recognition of Planar Object Classes. '96
M. C. Burl, M. Weber, P. Perona: A Probabilistic Approach to Object Recognition Using Local Photometry and Global Geometry. '98
M. Weber, M. Welling, P. Perona: Unsupervised Learning of Models for Recognition. '00
P. Felzenszwalb, and D. Huttenlocher, Pictorial Structures for Object Recognition, IJCV '05
P. Felzenszwalb, et. al., Object Detection with Discriminatively Trained DPMs,'10

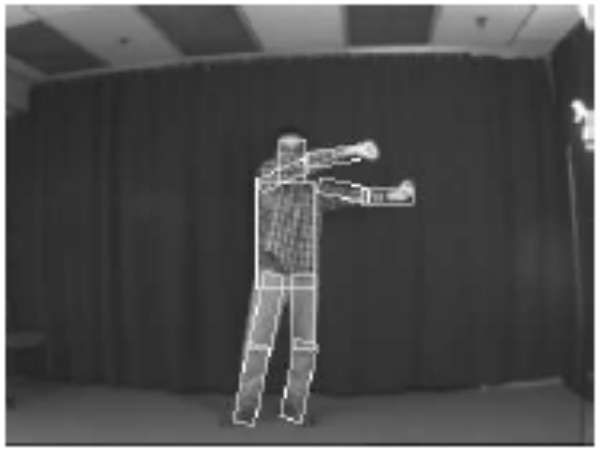
Deformable Part Models

‘what is here comes from there’

Complications: computation







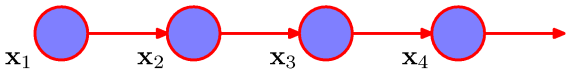
Is there a shape? Where?

Lecture outline

- Introduction
  - Graphs and computation
  - Graphical models
- Chain-structured graphical models
- Tree-structured graphical models

Problem 1: measuring the length of a queue

while only complaining to your closest neighbors

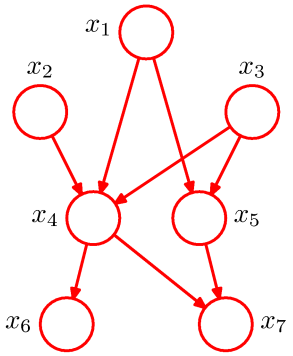


Problem 2: two queues

Problem N: N queues

Lecture outline

- Introduction
  - Computation on graphs
  - Graphical models
- Chain-structured graphical models
- Tree-structured graphical models



Graphical models

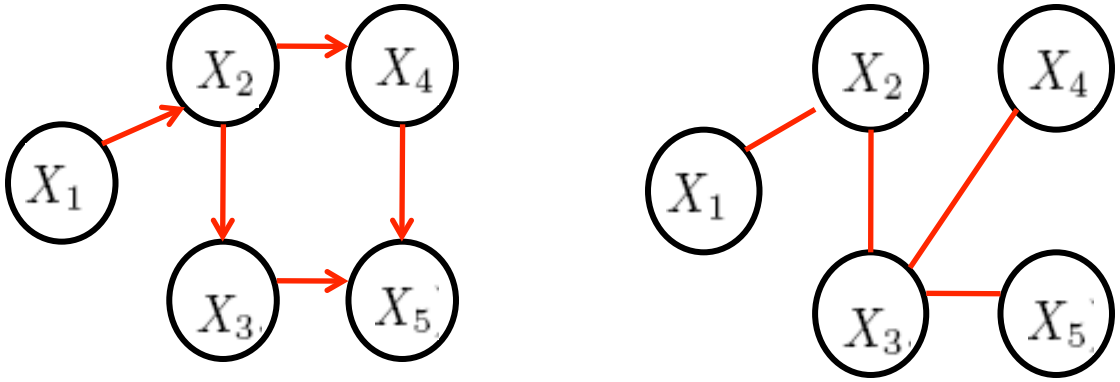
Blend of graph theory and probability

Graph nodes: random variables

Graph edges: relationships

Directed graphs: Bayesian Networks

Undirected graphs: Markov Random Fields



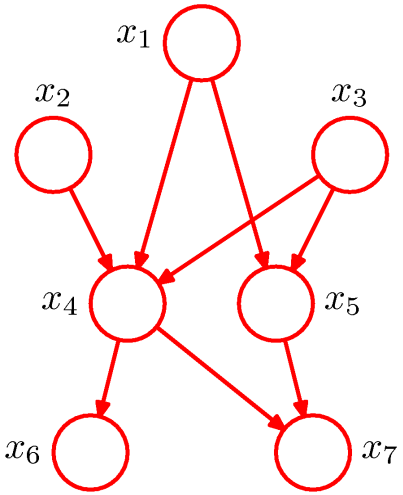
Bayesian Networks

Directed Acyclic Graph $G(V, E)$

Nodes - V : random variables

Edges - E: dependencies

Leave from `parents', arrive at children



Distribution of each child conditioned on parent: $P(X_v|X_{\pi_v})$

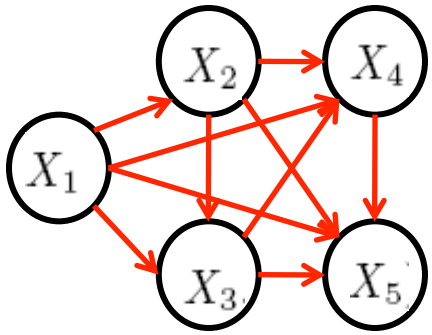
$$P(X_v|\emptyset) = P(X_v)$$

Joint probability distribution: $P(X) = \prod_v P(X_v|X_{\pi_v})$

Conditional independencies

We can trivially obtain a bayesian network for a probability distribution

$$P(X_1, X_2, X_3, X_4, X_5) = P(X_1)P(X_2|X_1)P(X_3|X_2, X_1)P(X_4|X_3, X_2, X_1)P(X_5|X_4, X_3, X_2, X_1)$$

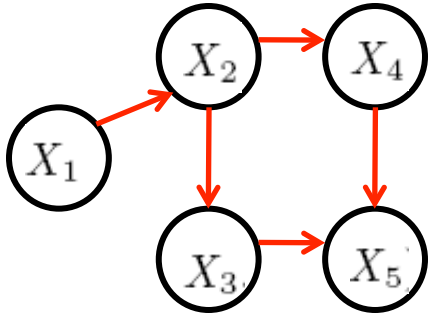


Fully connected graph!

Bayesian net

Non-trivial cases: some conditional independence

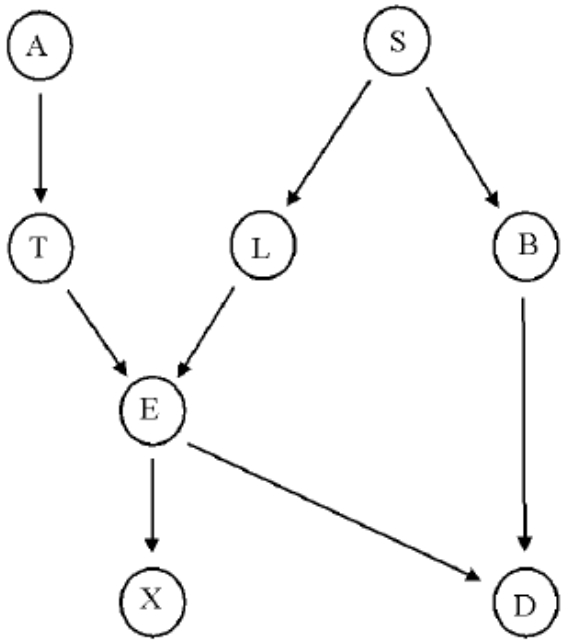
$$\begin{aligned} P(X_3|X_2, X_1) &= P(X_3|X_2) \\ P(X_4|X_3, X_2, X_1) &= P(X_4|X_2) \\ P(X_5|X_1, X_2, X_3, X_4, X_5) &= P(X_5|X_3, X_4) \end{aligned}$$



Main interest: exploit independencies for training & inference

A Bayesian Network-based Expert System

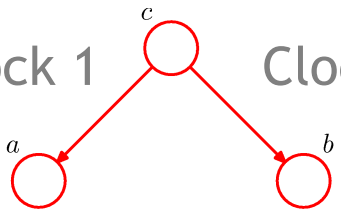
- `Asia Network'
  - A: trip to Asia
  - T: turbeculosis
  - S: smoking
  - L: Lung Cancer
  - B: Bronchitis
  - E: Turbeculosis/Lung Cancer
  - X-ray results
  - D: Dyspnea



- Given: X-rays, Dyspnea, patient went to Asia, patient smokes
- Wanted: posterior probability of Bronchitis

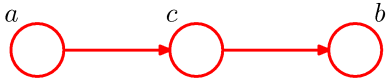
Reading Conditional Independence from a BN Graph

Big Ben



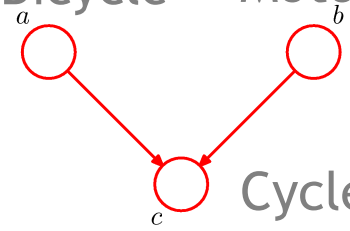
$$P(a,b) \neq P(a)P(b)$$

State 1 State 2 State 3

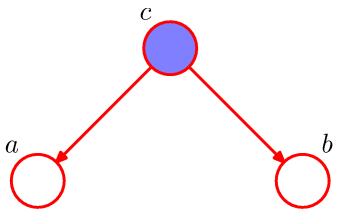


$$P(a,b) \neq P(a)P(b)$$

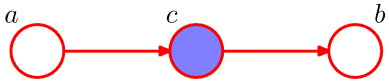
Bicycle Motorcycle



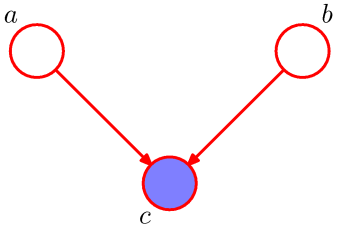
$$P(a,b) = P(a)P(b)$$



$$P(a,b|c) = P(a|c)P(b|c)$$



$$P(a,b|c) = P(a|c)P(b|c)$$

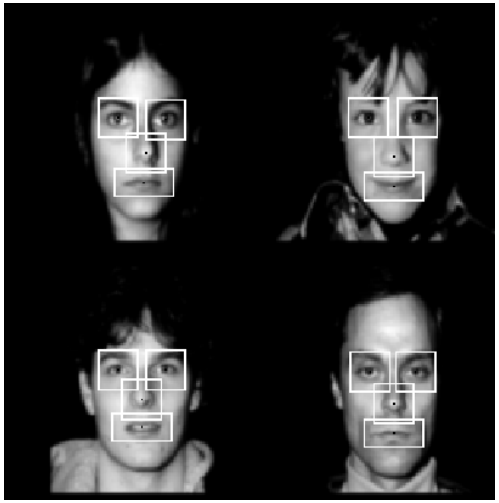


$$P(a,b|c) \neq P(a|c)P(b|c)$$

`Explaining away'

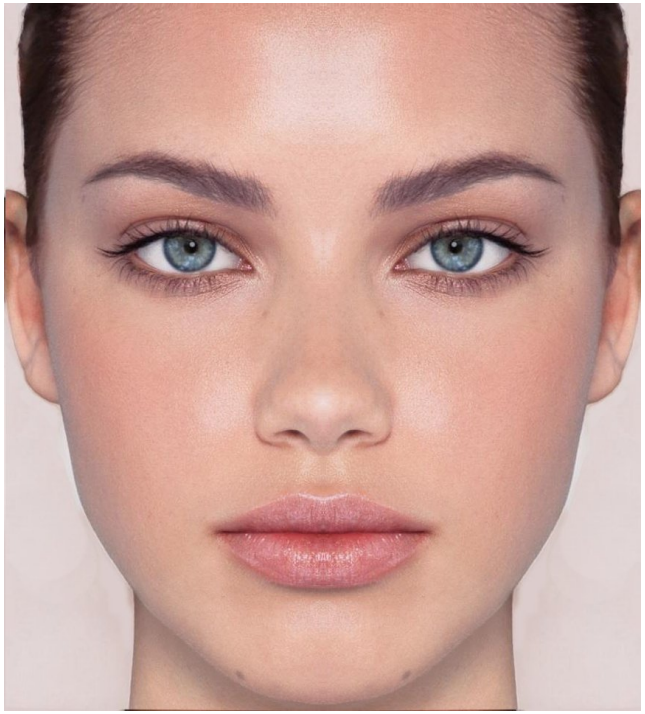
Lecture outline

- Introduction
- Chain-structured graphical models
- Tree-structured graphical models
  - Pictorial Structures
  - Max/Sum-Product algorithm



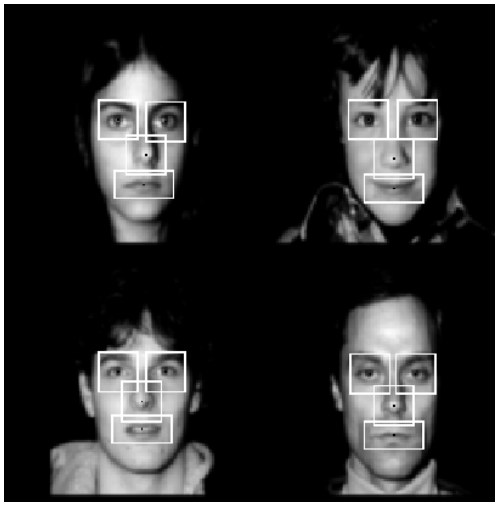
Problem N: N queues

Part-based Models

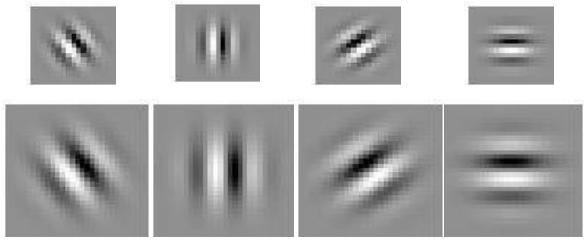


Unary terms (appearance model)

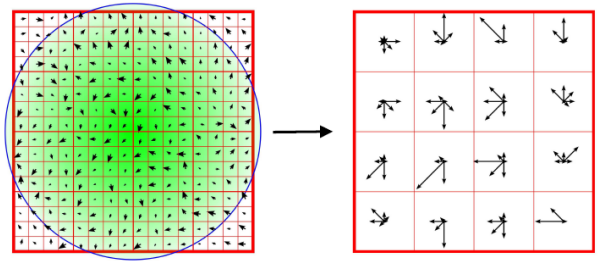
- Consider a patch of the image around the location of the part
- Construct statistical model for appearance at part location



Filterbank responses

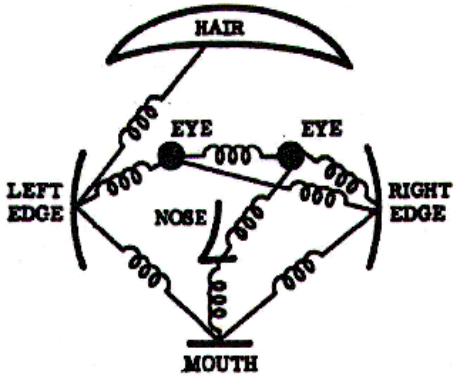


SIFT features

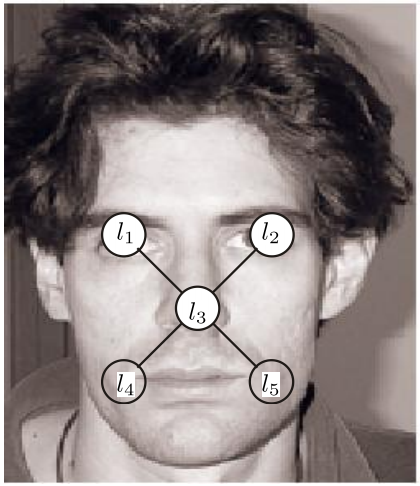


Pairwise terms (deformation model)

- Deformable objects as physical systems
  - Masses: driven towards good image locations
  - Springs: enforce relations among parts



- Star models (tree-structured models)
  - Pick single part as root (e.g. the nose)
  - Model other parts (eyes, mouth corners) w.r.t. root



- Probability of configuration $L = (l_1, l_2, l_3, l_4, l_5)$

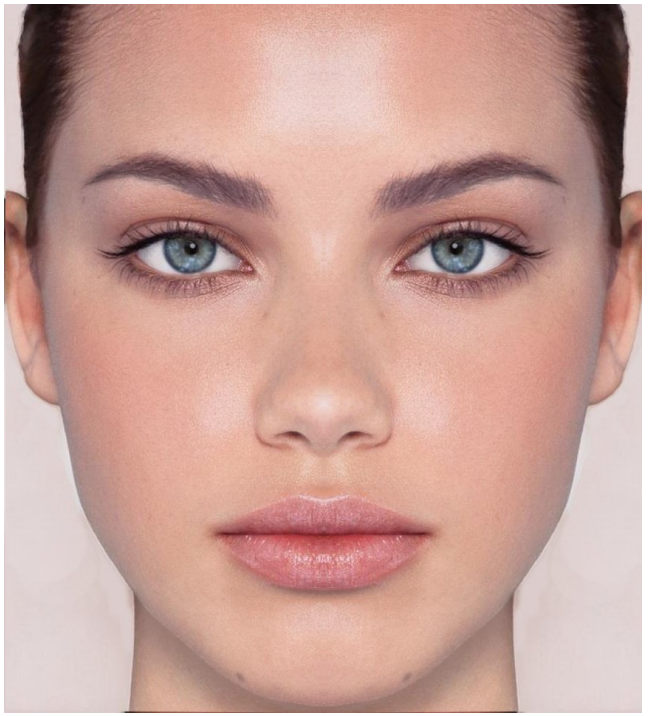
$$P(L|I) \propto P(I|L)P(L) = \prod_i \Phi_i(I_i|l_i) \prod_{(i,j)\in\mathcal{C}} \Psi_{i,j}(l_i, l_j)$$

Unary Terms — Pairwise Terms

- How can we maximize over L?

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

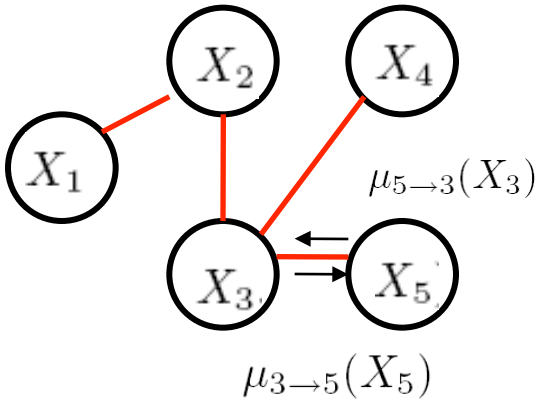
$$B_p(x, x_p) = -(h - h_p - \hat{h}_p)^2 \eta - (v - v_p - \hat{v}_p)^2 \nu$$

Object score

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

Lecture outline

- Introduction
- Chain-structured graphical models
- Tree-structured graphical models
  - Pictorial Structures
  - Max/Sum-Product algorithm

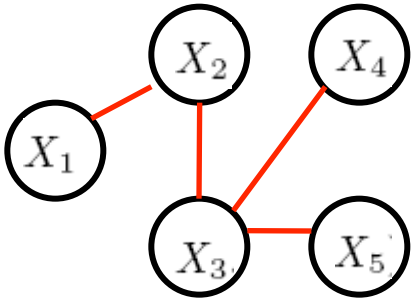


Inference on tree-structured Graphs

- Assume we want to find the most likely value of X_1

$$P(X) = \frac{1}{Z}\prod_{i=1}^{5}\Phi_i(X_i)\prod_{(i,j)\in\mathcal{C}}\Psi_{i,j}(X_i,X_j)$$

$$\mathcal{C} = \{(1,2),(2,3),(3,4),(3,5)\}$$



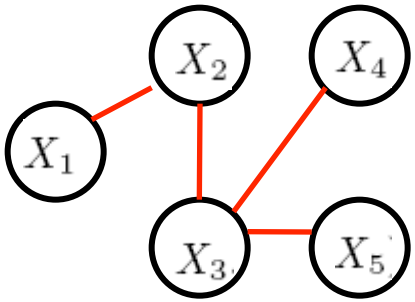
- Brute-force maximization:

$$\max P(X_1) = \max_{X_2,X_3,X_4,X_5} P(X_1,X_2,X_3,X_4,X_5)$$

$$|X_1| = K \rightarrow K^4$$

- Exploit factorization

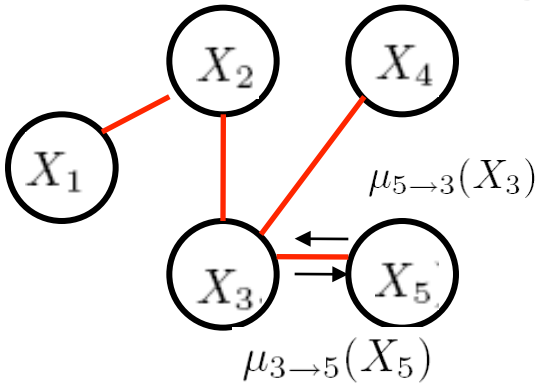
Exploiting the factorization



$$P(X) = \frac{1}{Z}\prod_{i=1}^{5}\Phi_i(X_i)\prod_{(i,j)\in\mathcal{C}}\Psi_{i,j}(X_i,X_j)$$
$$\mathcal{C} = \{(1,2),(2,3),(3,4),(3,5)\}$$

$$\begin{aligned}
& \max P(X_1) \\
= \; & \max_{X_2,X_3,X_4,X_5}\Phi(X_1)\Phi(X_2)\Phi(X_3)\Phi(X_4)\Phi(X_5)\Psi(X_1,X_2)\Psi(X_2,X_3)\Psi(X_3,X_4)\Psi(X_3,X_5) \\
= \; & \Phi(X_1)\max_{X_2}\Phi(X_2)\Psi(X_1,X_2)\max_{X_3,X_4}\Phi(X_3)\Psi(X_2,X_3)\Phi(X_4)\Psi(X_3,X_4)\underbrace{\max_{X_5}\Phi(X_5)\Psi(X_3,X_5)}_{\mu_5(X_3)} \\
= \; & \Phi(X_1)\max_{X_2}\Phi(X_2)\Psi(X_1,X_2)\max_{X_3}\Phi(X_3)\Psi(X_2,X_3)\mu_5(X_3)\underbrace{\max_{X_4}\Phi(X_4)\Psi(X_3,X_4)}_{\mu_4(X_3)} \\
= \; & \Phi(X_1)\max_{X_2}\Phi(X_2)\Psi(X_1,X_2)\underbrace{\max_{X_3}\Phi(X_3)\Psi(X_2,X_3)\mu_5(X_3)\mu_4(X_3)}_{\mu_3(X_2)} \\
= \; & \Phi(X_1)\underbrace{\max_{X_2}\Phi(X_2)\Psi(X_1,X_2)\mu_3(X_2)}_{\mu_2(X_1)}
\end{aligned}$$

Max-Product algorithm



- Distributed computation on a graph
  - `message-passing'

- Message from node i to node j

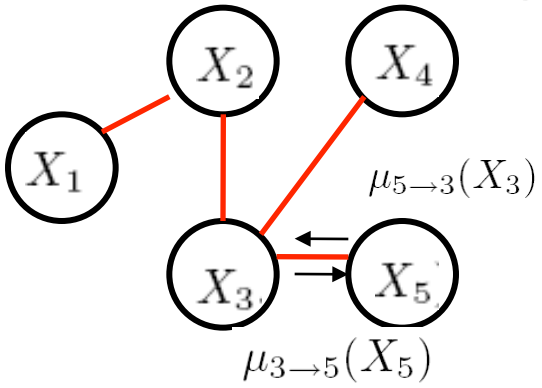
$$\mu_{i\to j}(X_j) \quad = \quad \max_{X_i} \Phi_i(X_i)\Psi_{i,j}(X_i,X_j) \prod_{k\in\mathcal{N}(i)\setminus j} \mu_{k\to j}(X_i)$$

  - `Given all I know from the rest, this is where you should be'

- Beliefs: $$B_j(X_j) \quad = \quad \Phi_j(X_j) \prod_{i\in\mathcal{N}(j)} \mu_{i\to j}(X_j)$$

- At convergence $$B_j(X_j) = \max P(X_j)$$

Sum-Product algorithm



- Distributed computation on a graph
  - `message-passing'

- Message from node i to node j

$$\mu_{i\to j}(X_j) = \sum_{X_i} \psi_i(X_i)\psi_{i,j}(X_i, X_j) \prod_{k\in\{\mathcal{N}(i)\setminus j\}} \mu_{k\to i}(X_i)$$

  - `Given all I know from the rest, this is where you should be'

- Beliefs: $$B_j(X_j) = \Phi_j(X_j) \prod_{i\in\mathcal{N}(j)} \mu_{i\to j}(X_j)$$

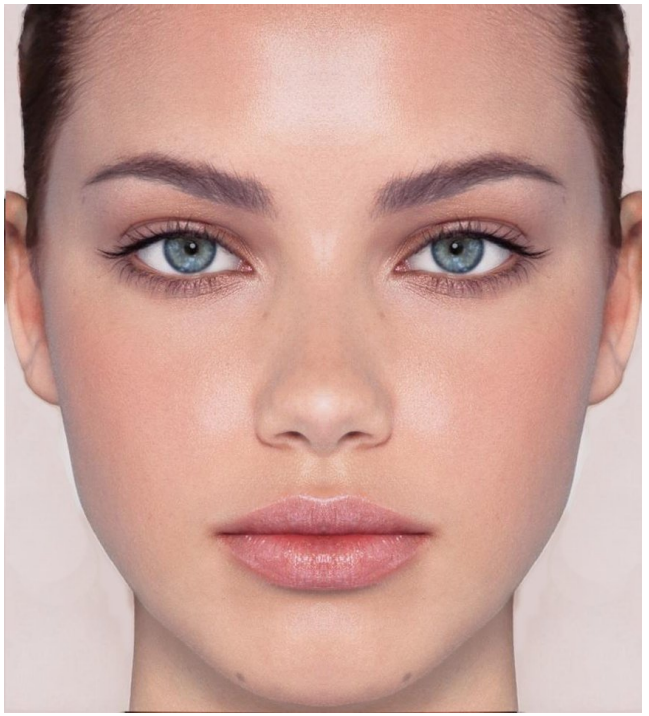
- At convergence $$P(X_i) = B(X_i)$$

Max/Sum-Product algorithms

- Asychronous inference
- Local computations
- Guaranteed optimality for tree-structured graphs
- Equivalent algorithms:
  - Tracking: Kalman Filtering (Sum-Product)
  - HMMs: Alpha-Beta (Sum-Product)
  - Coding/HMMs: Viterbi Algorithm (Max-Product)
  - Bayesian Nets: Belief Propagation (Sum-Product)
- Max Product: Dynamic Programming

Deformable Part Models (DPMs)

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



Local appearance

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

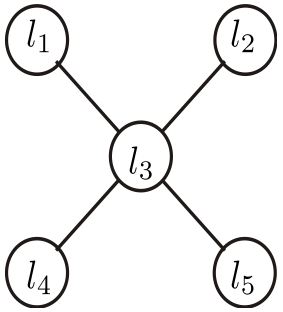
Pairwise compatibility

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

Object score

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

Pictorial Structures



- Graphical model
  - Nodes: part locations
  - Edges: dependencies - relative locations

- Object localization: maximization with respect to $L = (l_1, l_2, l_3, l_4, l_5)$

$$P(L|I) \propto P(I|L)P(L) = \prod_i \Phi_i(I_i|l_i) \prod_{(i,j)\in\mathcal{C}} \Psi_{i,j}(l_i, l_j)$$

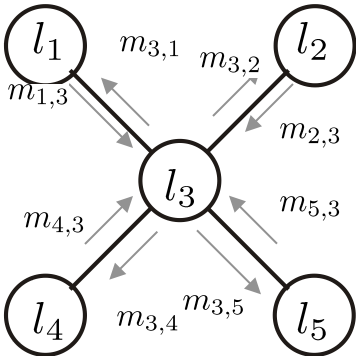
Unary terms **Pairwise terms**

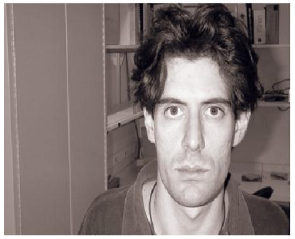
- Message Passing: $\mu_{i\to j}(X_j) = \max_{X_i} \Phi_i(X_i)\Psi_{i,j}(X_i, X_j) \prod_{k\in\mathcal{N}(i)\setminus j} \mu_{k\to j}(X_i)$

- Efficient Implementation: Felzenszwalb & Huttenlocher, CVPR 2001/IJCV 05

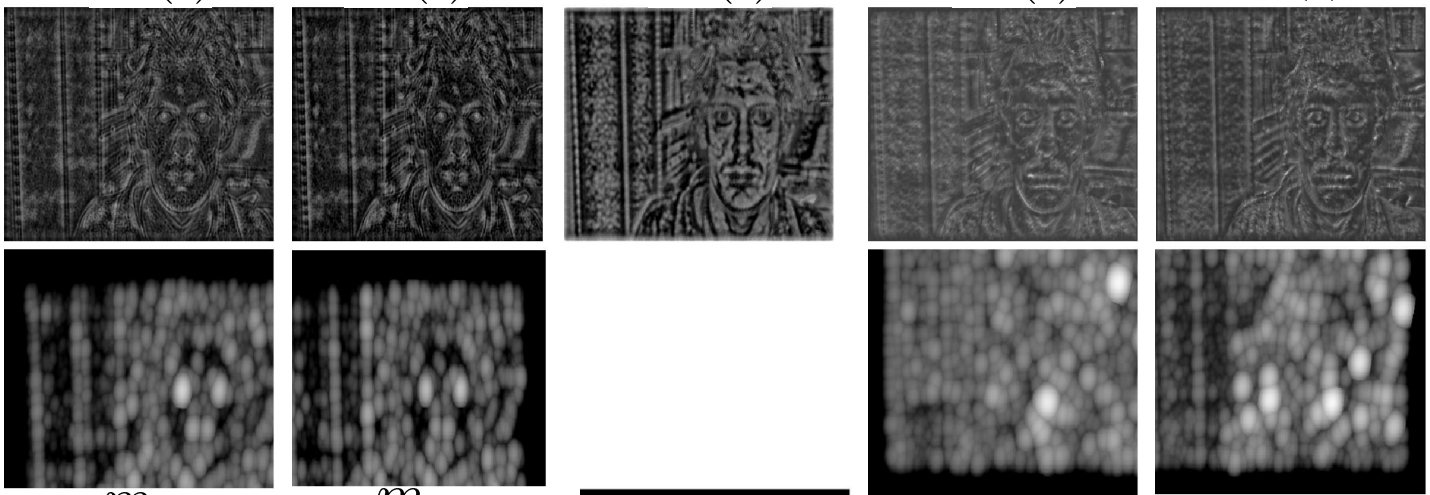
$$\mu_{i\to j}(X_j) = \max_{X_i} \Phi_i(X_i)\Psi_{i,j}(X_i, X_j) \prod_{k\in\mathcal{N}(i)\setminus j} \mu_{k\to j}(X_i)$$

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

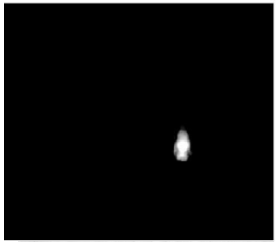




$\Phi_1(l)$ $\Phi_2(l)$ $\Phi_3(l)$ $\Phi_4(l)$ $\Phi_5(l)$

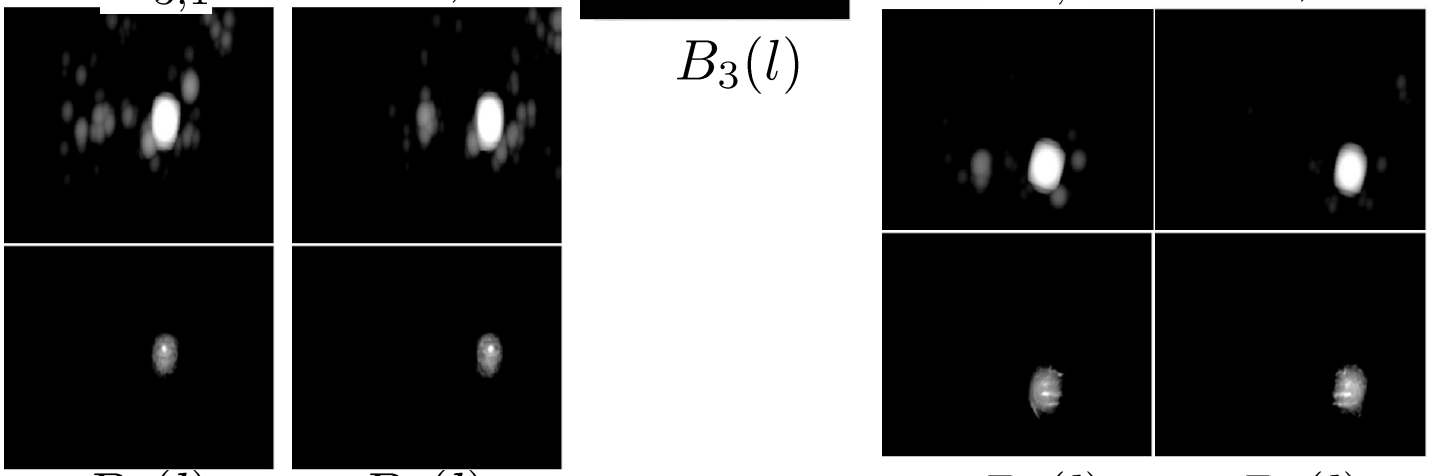


$m_{1,3}$ $m_{2,3}$ $m_{4,3}$ $m_{5,3}$



$B_3(l)$

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



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