Vision and Learning

CIS680 Review
Jianbo Shi
Spring 2004

Topics

Texture synthesis/analysis

Fixed body object detection/recognition

Flexible body object detection/recognition

Image segmentation
Image translation

Image Shape modeling

Human activity recognition

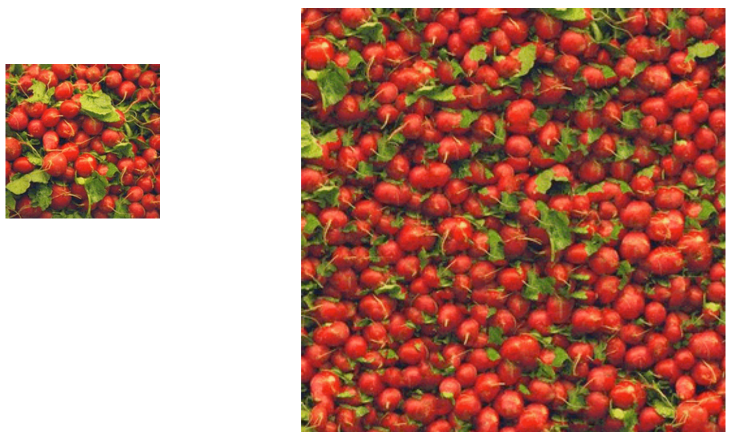
Texture synthesis/analysis

- Image features
  - Filter bank, filter histogram, nth order correlation
- Learning formulation
  - MRF, Max. Entropy, Max. Likelihood
- Techniques
  - Sampling, MCMC; Direct gradient decent; celver copying

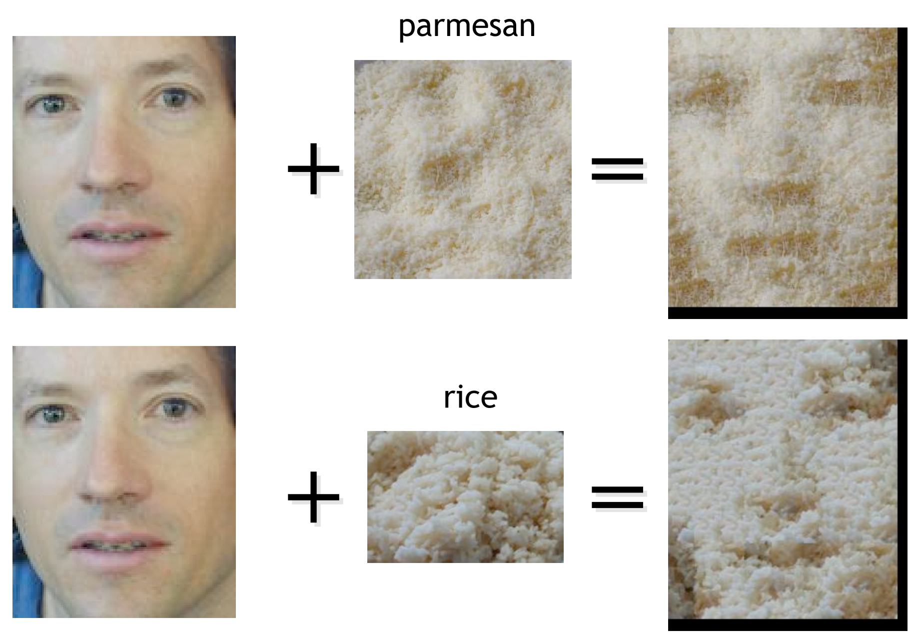
Texture synthesis/analysis

- An practical solution: Efros et. al.
- Texture similarity: Martin&Fowlkes&Malik, Rubner&Tomasi&Guibas, Puzicha et. al.
- Texture synthesis statistical: Zhu&Wu&Mumford
- Texture synthesis Energy: Portilla&Simoncelli

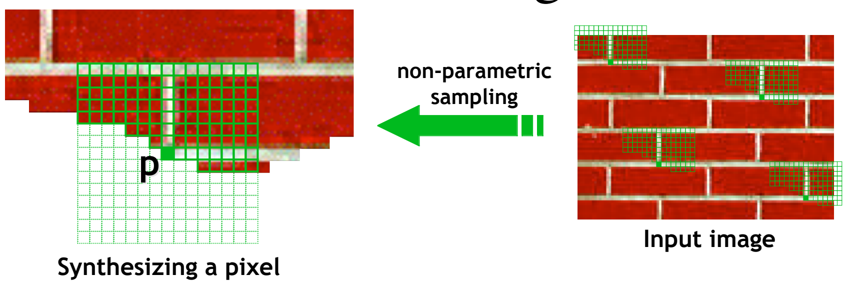
Texture Synthesis



[Efros,Leung ‘99]

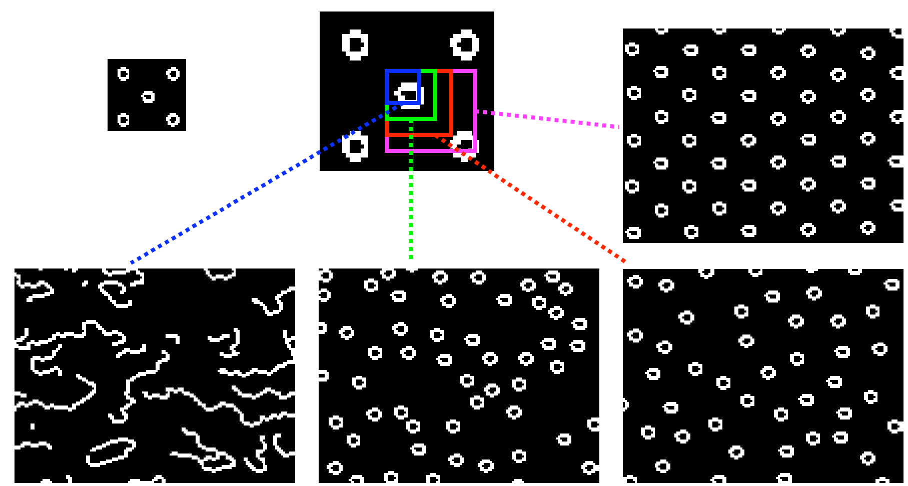


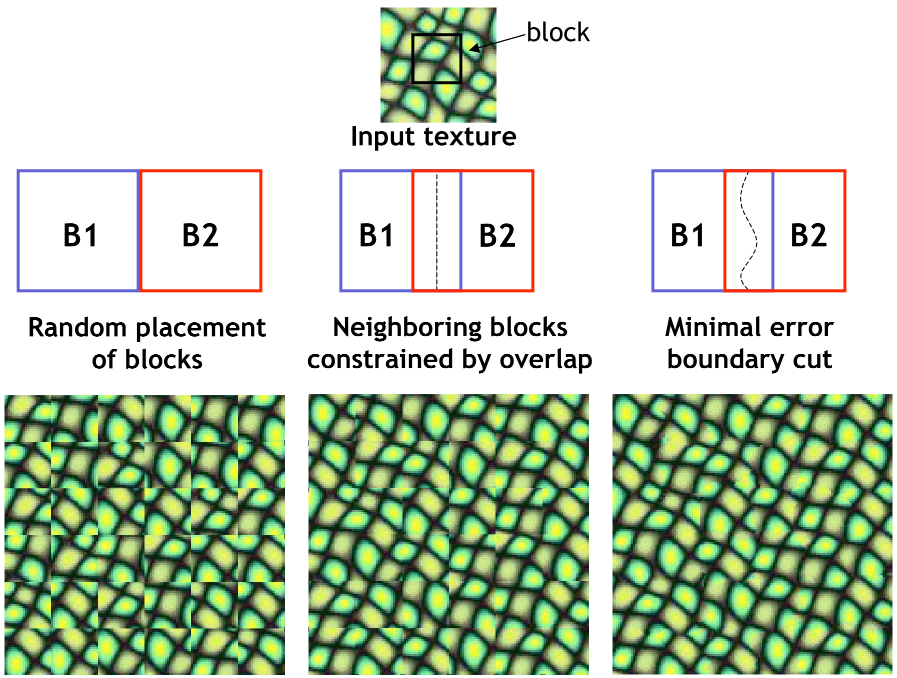
Efros & Leung ’99



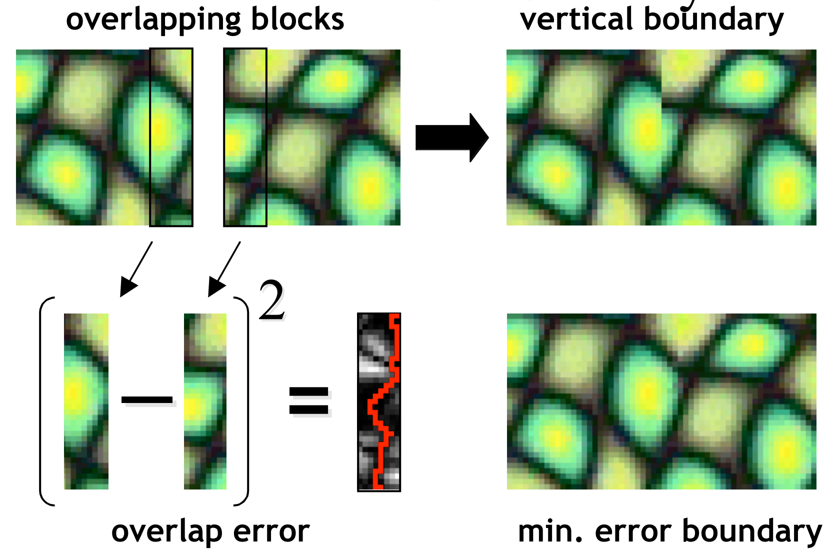
- Assuming Markov property, compute P(**p**|N(**p**))
  - Building explicit probability tables infeasible
  - Instead, let’s *search the input image* for all similar neighborhoods — that’s our histogram for **p**
- To synthesize **p**, just pick one match at random

Randomness Parameter

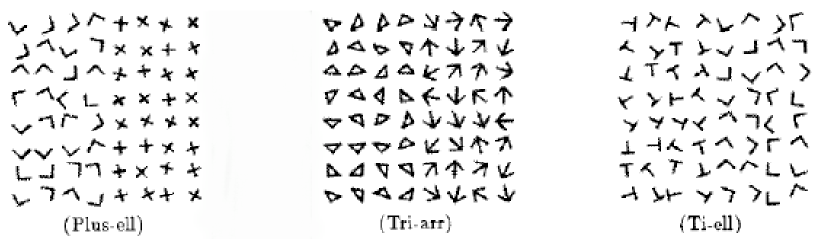




Minimal error boundary

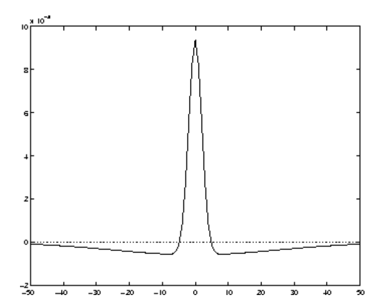


Texture similarity measure

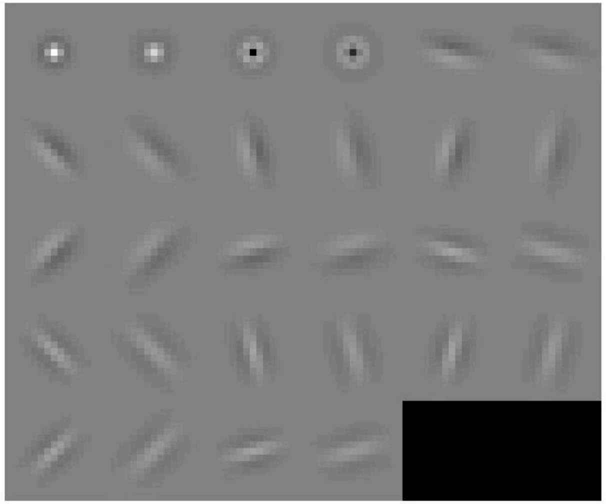


Difference of Gaussian (DOG)

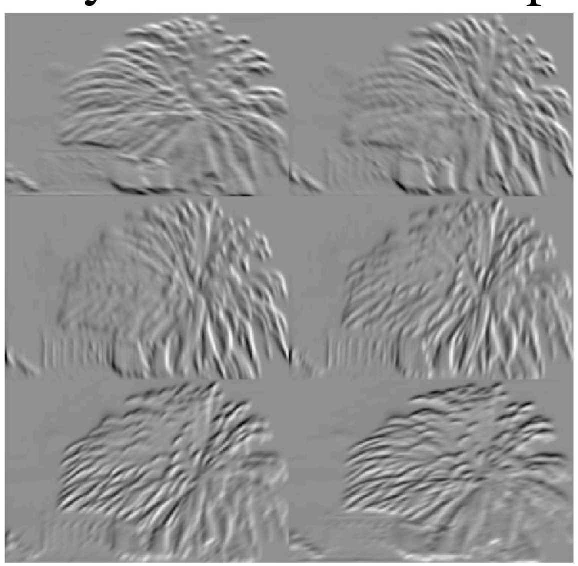
$$DoG(x,y) = \frac{c_{ex}^2}{2\pi\sigma_{ex}^2} e^{-\frac{x^2+y^2}{2\sigma_{ex}^2}} - \frac{c_{inh}^2}{2\pi\sigma_{inh}^2} e^{-\frac{x^2+y^2}{2\sigma_{inh}^2}}$$



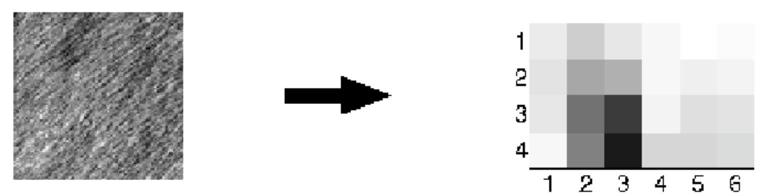
Filter Banks

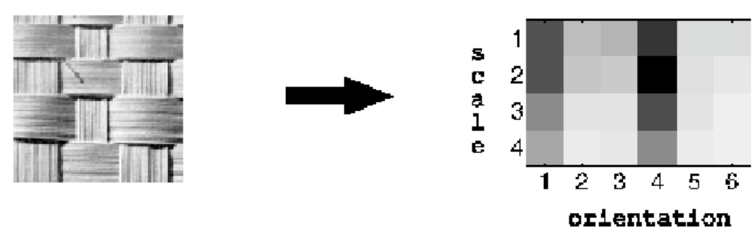


Odd symmetric fitler outputs



Similarity using ave. filter bank response





Texture histogram

$$f^r(i;I) \;=\; \left|\{\vec{x} : t^r_{i-1} < I^r(\vec{x}) \le t^r_i\}\right| \;.$$

(i) The *Minkowski-form distance* $\mathcal{L}_p$ is defined by:

$$D(I,J) = \left(\sum_i |f(i;I) - f(i;J)|^p\right)^{1/p} \;.$$

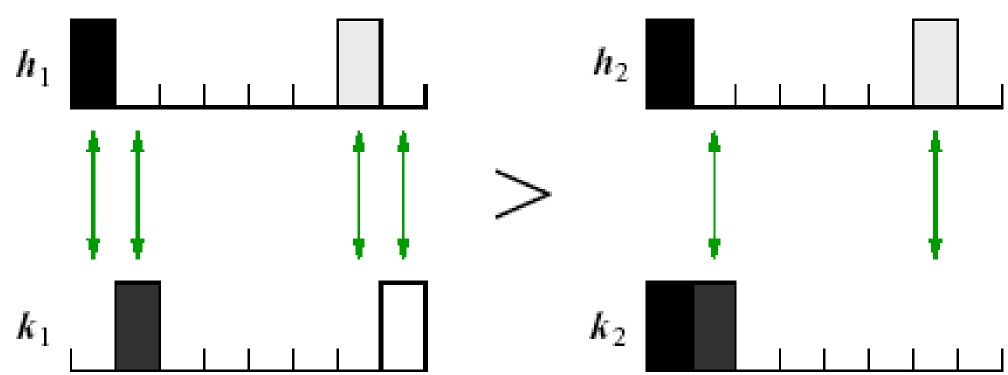
The χ^2–*statistic* is given by

$$D(I,J) = \sum_i \frac{\left(f(i;I) - \hat{f}(i)\right)^2}{\hat{f}(i)},$$

Perceptual similarity

$$D(I,J) = \sqrt{(\vec{f}_I - \vec{f}_J)^T \mathbf{A} (\vec{f}_I - \vec{f}_J)}\;,$$

Earth Moving Distance



(a)

Image similarity with L1 distance

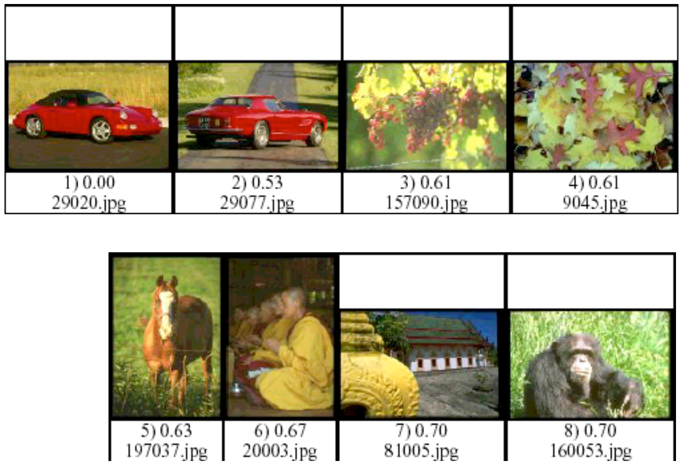


Image similarity w. chi-sqr statistics

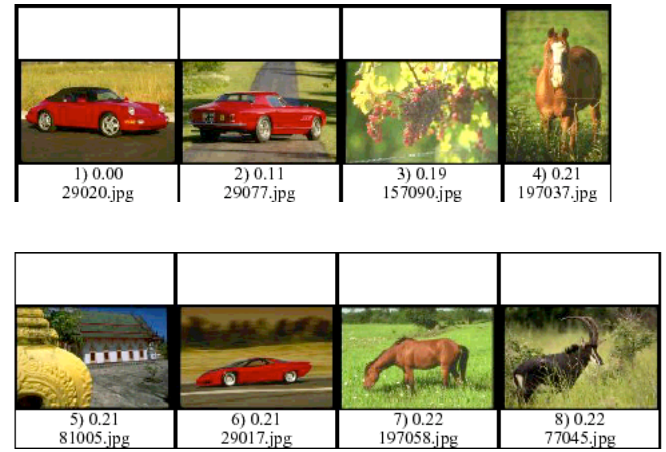


Image similarity w. quadratic-form

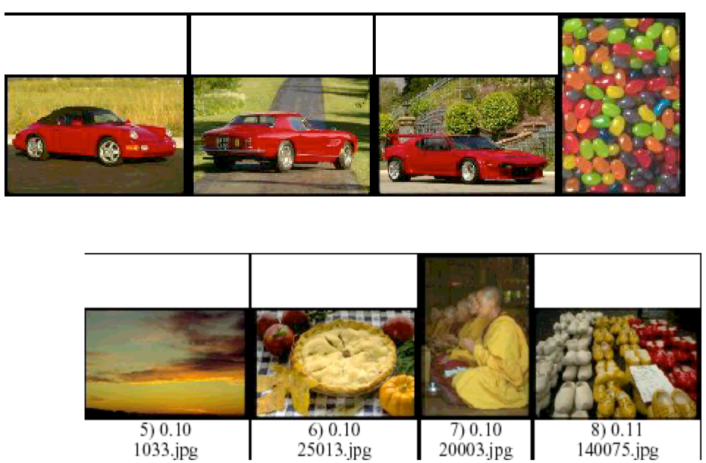
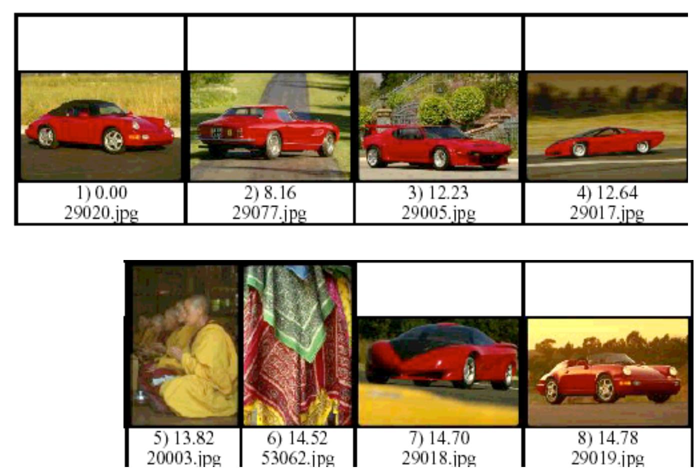
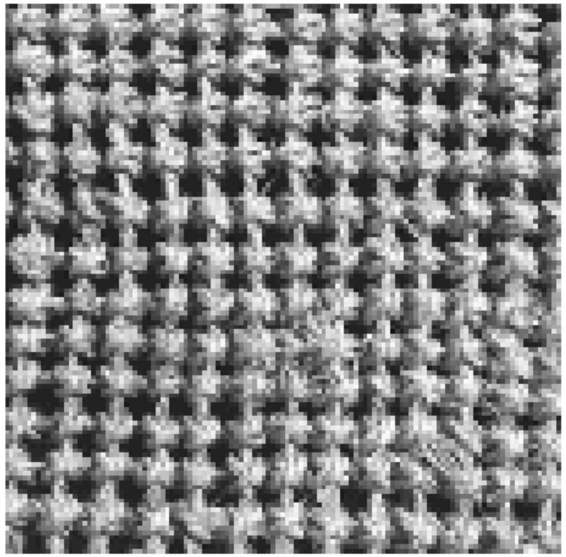
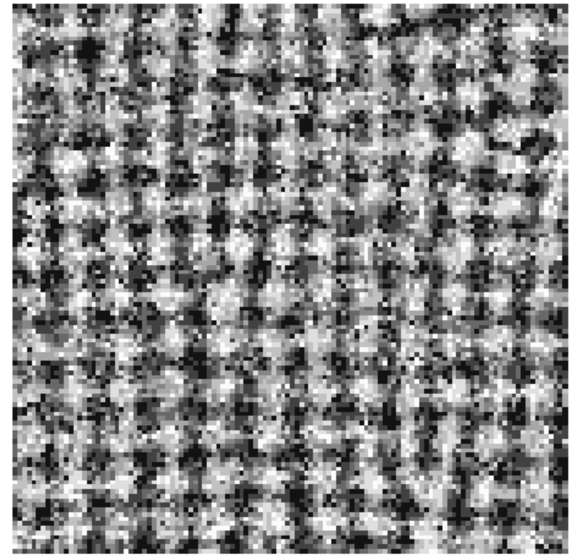


Image similarity with Earth Moving Distance(EMD)



Models of P(I)

Markov Random Field(MRF)

- Sites, v
- Neighbourbood, N(v)
- MRF distribution:

$$p(\mathbf{I}(v)|\mathbf{I}(-v)) = p(\mathbf{I}(v)|\mathbf{I}(N_v))$$

- Gibbs distribution:

$$p(\mathbf{I}) = \frac{1}{Z} e^{-\sum_{c \in C} \lambda_C(\mathbf{I}(C))}$$

Hammersley-Clifford Theorem

- For a given N, p(I) is an MRF distribution iff p(I) is a Gibbs distribution

- This equivalence allows us to specify MRF through the definition of clique energy

$$p(\mathbf{I}) = \frac{1}{Z} \exp\{\sum_{\vec{v}} g(\mathbf{I}(\vec{v})) + \sum_{\vec{u},\vec{v}} \beta_{\vec{u}-\vec{v}} \mathbf{I}(\vec{u})\mathbf{I}(\vec{v})\},$$

Maximum Entropy Formulation

- Find a probability distribution p(x) s.t.

$$p^*(x) = \arg\max\{-\int p(x)\log p(x)dx\},$$

$$E_p[\phi_n(x)] = \int \phi_n(x)p(x)dx = \mu_n, \quad n = 1, ..., N,$$
$$\int p(x)dx = 1.$$

M.E. solution

$$p(x;\Lambda) = \frac{1}{Z(\Lambda)}\exp\{-\sum_{n=1}^{N}\lambda_n\phi_n(x)\},$$

$$Z(\Lambda) = \int \exp\{-\sum_{n=1}^{N}\lambda_n\phi_n(x)\}dx$$

$$\frac{d\lambda_n}{dt} = E_{p(I;\Lambda)}[\phi_n(x)] - \mu_n,$$

Monte Carlo Samplings

- *P(x) = a exp(- E(x)/T);*
- Metropolis
  - A proposed update distribution *q(y|x) = q(x|y)*
  - Take a update y, accept with *min(1,p(y)/p(x))*
- Metropolis Hasting
  - Accept with min(1, *[q(x|y)p(y)] / [q(y|x)p(x)])*
- Gibbs sampler:
  - Sample from the distribution itself, and sample one component at time
  - *q(y(k)|x(k), x(-k)) = p(y(k) | x(-k)*

Generating images from P(I)

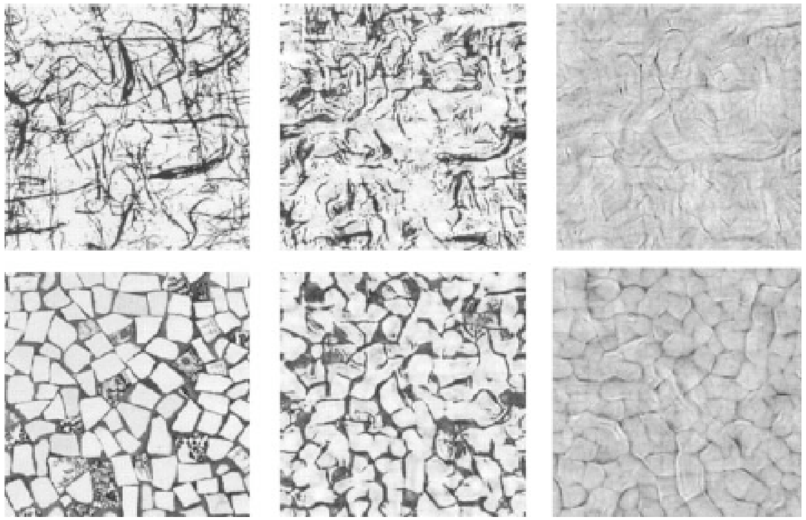
- Basic ideas on generating the right images:
  - Starting from a random image
  - Compute the filter output, and filter histogram(H)
  - Use H as an est. of $E_{p(\mathbf{I};\Lambda_K,S_K)}(H^{(\alpha)})$

$$\frac{d\lambda^{(\alpha)}}{dt} = E_{p(\mathbf{I};\Lambda_K,S_K)}[H^{(\alpha)}] - H^{obs(\alpha)}$$

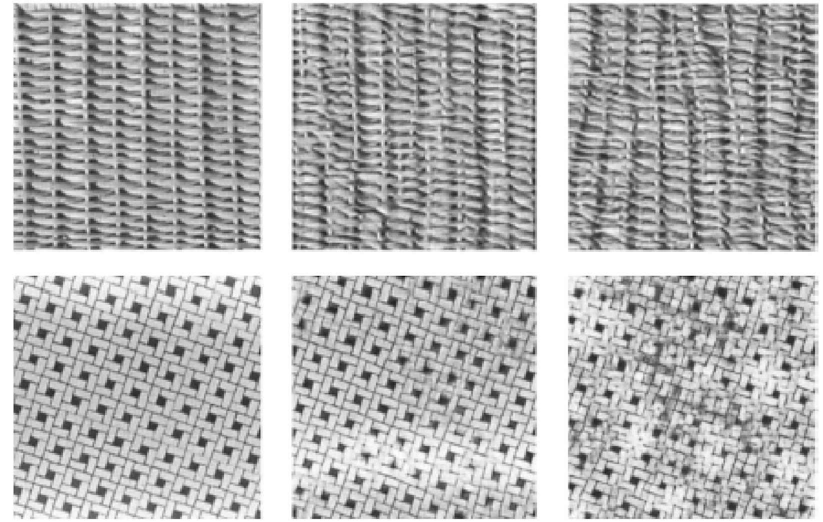
  - Use Gibbs sampling to flip I according to the new P(I)

$$= \frac{1}{Z(\Lambda_K)}\exp\{-\sum_{\alpha=1}^{K} < \lambda^{(\alpha)},\ H^{(\alpha)} >\}.$$

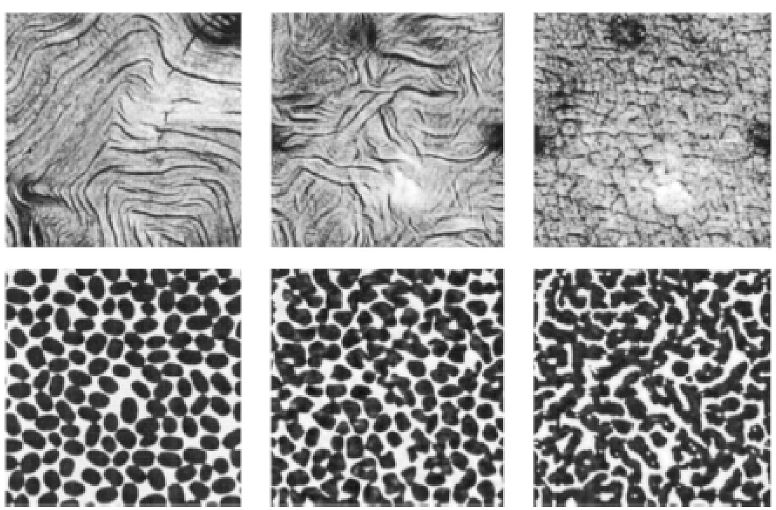
Image Intensity histogram



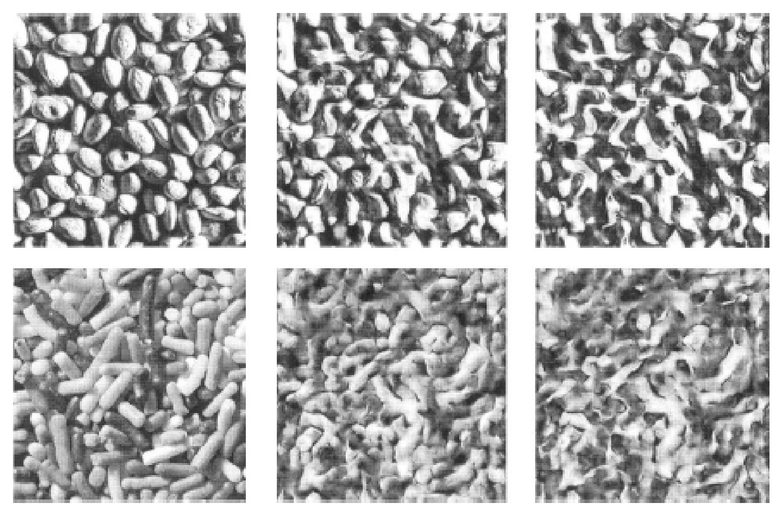
Spatial correlation



Orientation-Scale correlation



Filter phase correlation

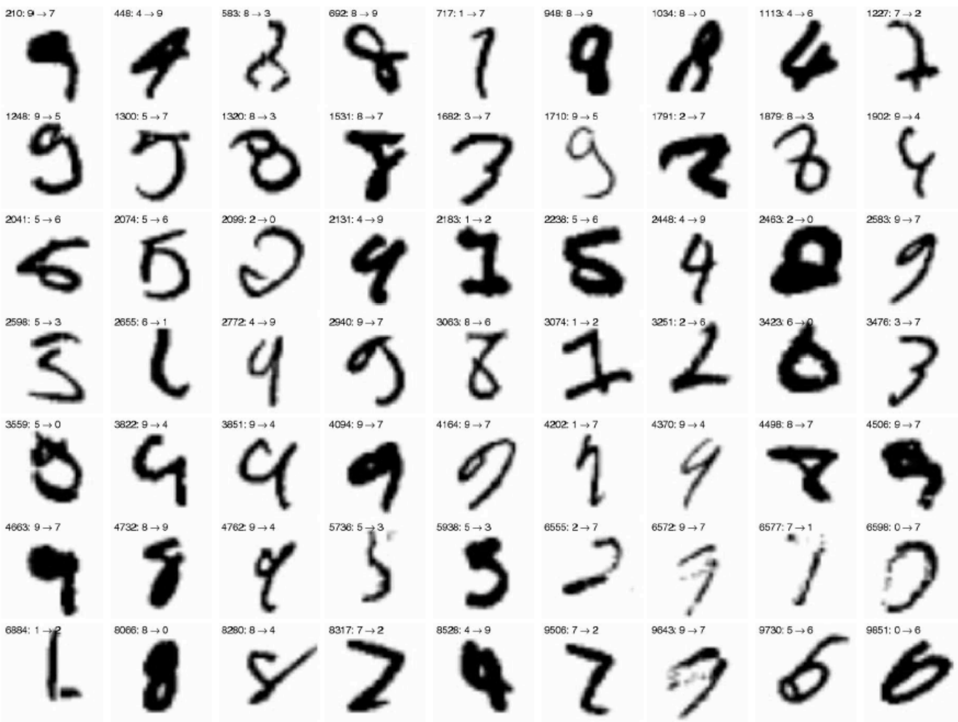


Fixed Body object detection

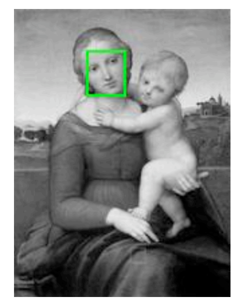
- Image features:
  - Pixel value, Edge map, Filter-bank, Viola-Jones facial parts, Shape-Context
- Formulation:
  - Naive Bayesian, discriminative classifier, function approximation
- Technique for training,
  - Boosting, SVM, Neural Network, Nearest neighbour

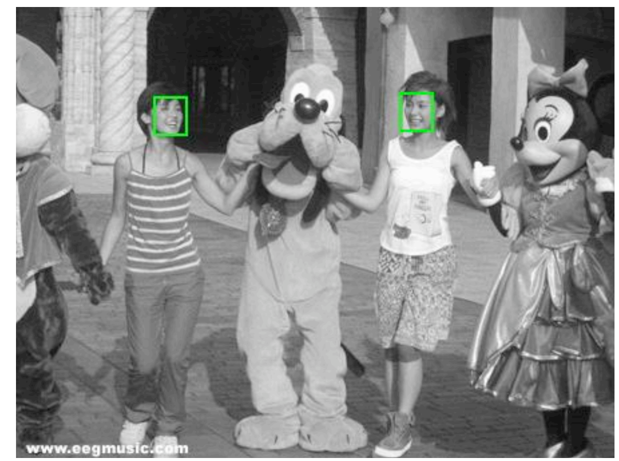
Fixed Body object detection

- Face detection: Scheinderman&Kanade, Viola&Jones
- Digit Recognition: LeCun et.al.
- SVM: Vapnik et. al.



Microsoft WDM Image Capture (Win32)
NetView
PC Camera
Video Capture.





www.eegmusic.com

The infeasible or ideal classification table

(1,1)	(1,2)	. . .	(20,20)	Classification
0	0	. . .	0	Non-object
0	0	. . .	1	Non-object
. . .	. . .	. . .	. . .	. . .
35	45	. . .	28	Object
. . .	. . .	. . .	. . .	. . .
255	255	. . .	255	Non-object

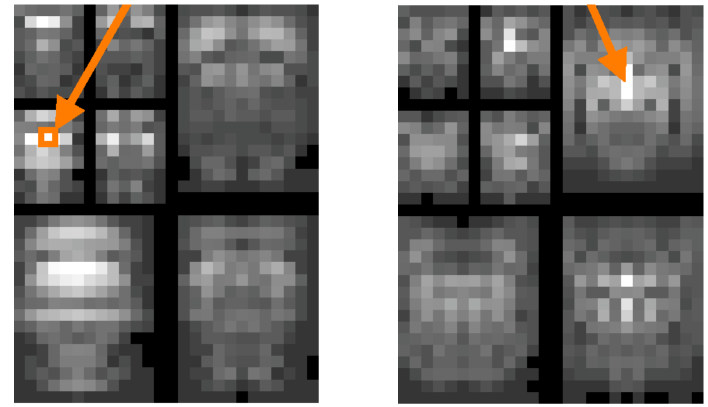
$256^{400} = 10^{964}$ entries for 20 x 20 images

p = 1e10; y = 100; d = 356; h = 24; m = 60; s = 60; f = 30;

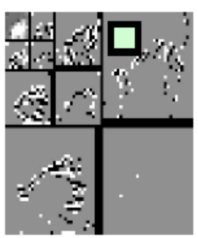
#images the entire population see in our life time = 10^19

Deposition to parts

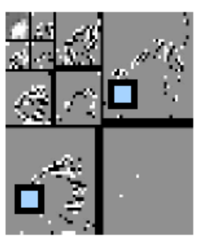
- pixel values on the face are correlated



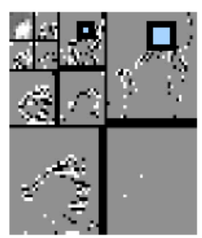
Scheinderman&Kanade



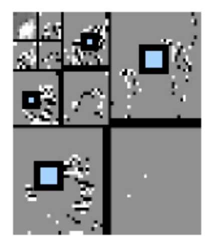
Intra-subband



Inter-orientation

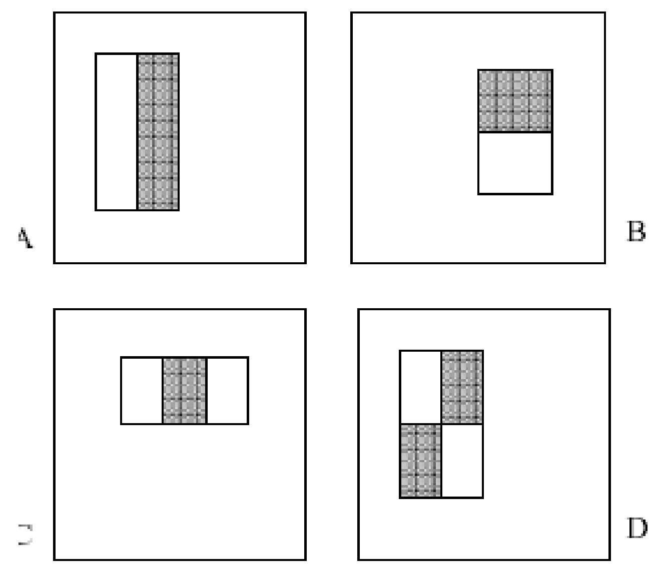


Inter-frequency

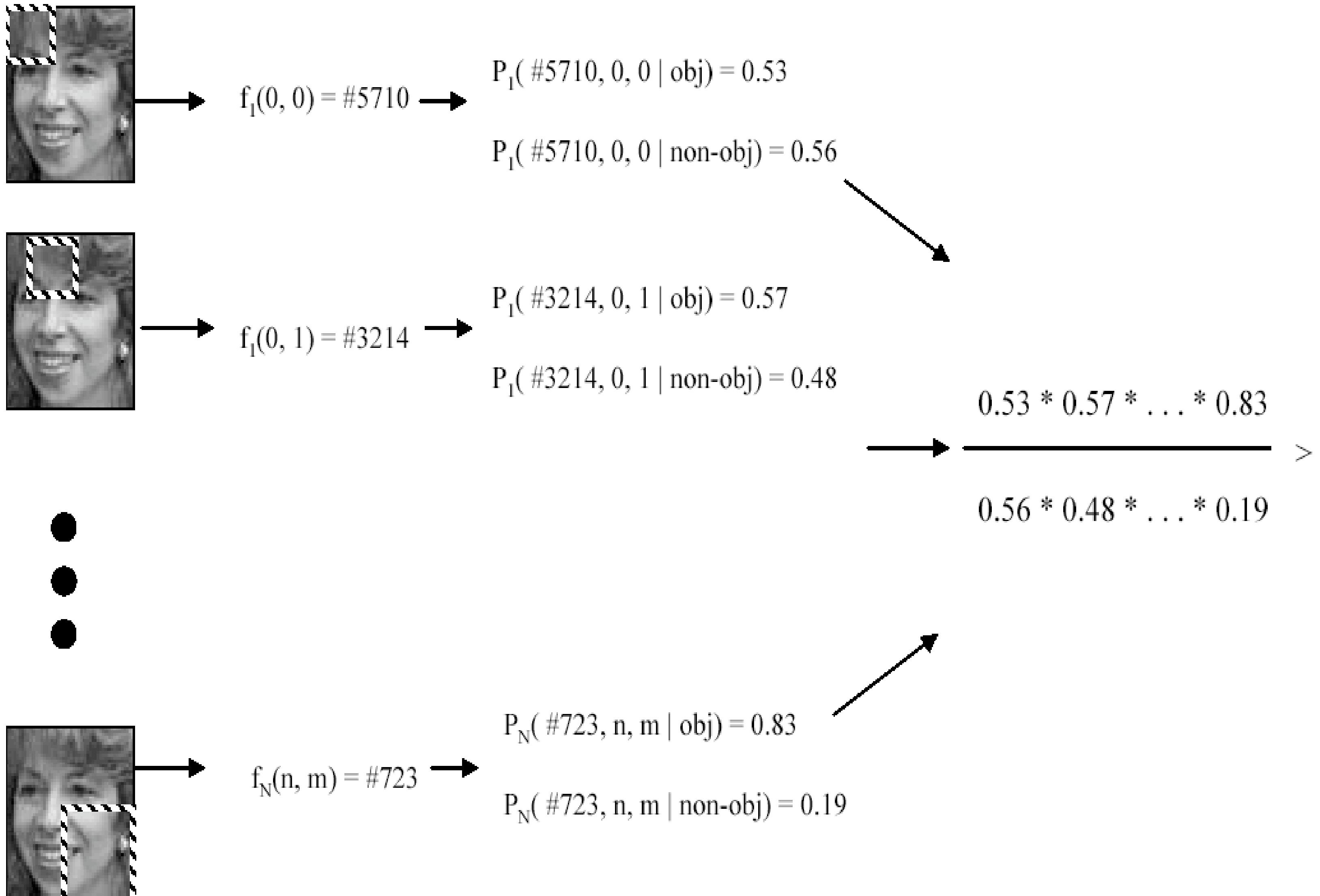


Inter-frequency/
Inter-orientation

Viola&Jones

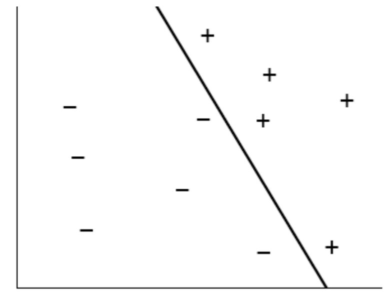


Naive Bayes



$f_1(0, 0) = \#5710$
$P_1(\#5710, 0, 0 \mid obj) = 0.53$
$P_1(\#5710, 0, 0 \mid non\text{-}obj) = 0.56$
$f_1(0, 1) = \#3214$
$P_1(\#3214, 0, 1 \mid obj) = 0.57$
$P_1(\#3214, 0, 1 \mid non\text{-}obj) = 0.48$
$f_N(n, m) = \#723$
$P_N(\#723, n, m \mid obj) = 0.83$
$P_N(\#723, n, m \mid non\text{-}obj) = 0.19$
$\frac{0.53 * 0.57 * \ldots * 0.83}{0.56 * 0.48 * \ldots * 0.19} > \lambda$

Classification

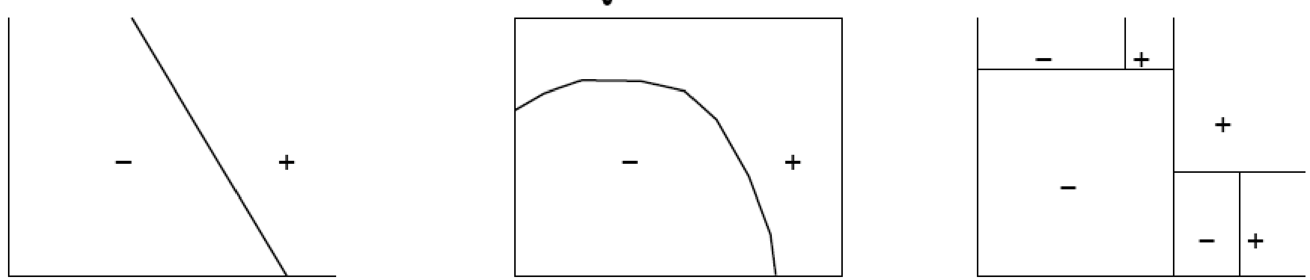


Input $x \in X$
Label $y = \pm 1$
Classifier $f : X -; \longrightarrow \{+1, -1\}$
Data set (training set)
$\mathcal{D} = \{(x_1, y_1), (x_2, y_2), \ldots (x_N, y_N)\}$
Learning algorithm $\mathcal{D}$ learning classifier $f \in \mathcal{F}$
We will be averaging several $f \in \mathcal{F}$. Hence, we call $\mathcal{F}$ the *base classifier* family.

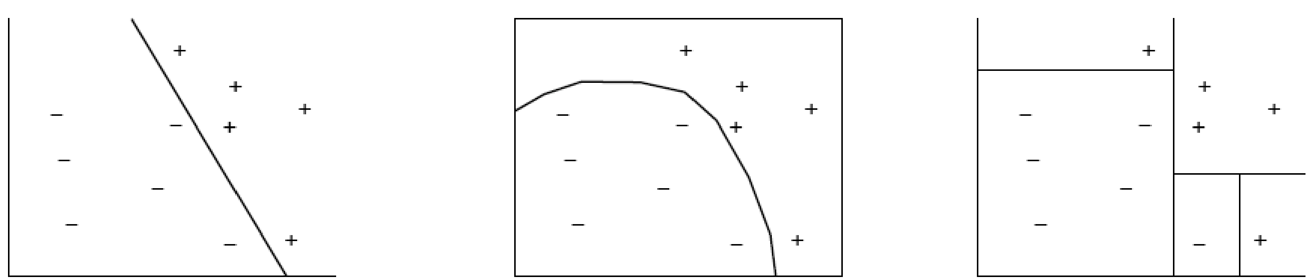
types of classifier

Bias vs. variance; margin

Linear **Quadratic** **Decision tree**



$f(x) = \mathrm{sign}\,(ax + b)$ $f(x) = \mathrm{sign}\,(ax^2 + bx + c)$



Cost of errors $C(y, f(x)) = 1$

Averaging: $F(x) = \text{sign}\ \Sigma_{k=1}^{M} c_k f_k(x)$

Averaging Classifiers

- can reduce variance
- can reduce bias
- can compensate for local optima (a form of bias)
- if $f_1, f_2, \ldots f_M$ make independent errors, averaging reduces error

Averaging is not always the same thing.

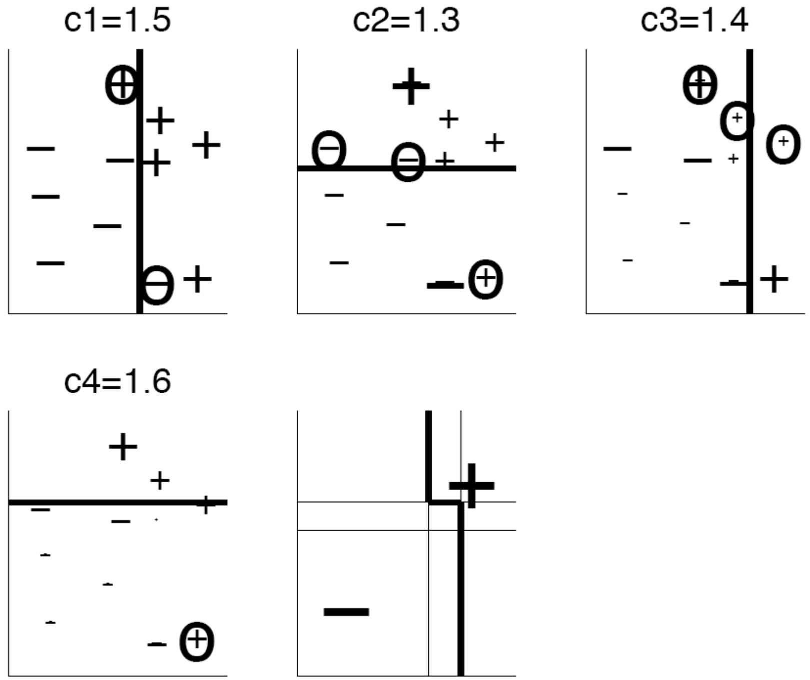
Depending how we choose $\mathcal{F}$, $f_1, f_2, \ldots f_M$ and $c_1, c_2, \ldots c_M$, we can obtain very different effects.

Bagging, Boosting, Bayes

Before seeing data	$P_0(f)$	prior distribution over f
After seeing $\mathcal{D}$	$P(f \mid \mathcal{D})$	posterior distribution over f
Bayes formula	$P(f \mid \mathcal{D}) = \frac{P_0(f)P(\mathcal{D} \mid f)}{\Sigma_{f' \in \mathcal{F}} P_0(f')P(\mathcal{D} \mid f')}$	

$$F(x) = \sum_{f \in \mathcal{F}} f(x) P(f \mid \mathcal{D})$$

Boosting



Input M, labeled training set $\mathcal{D}$

Initialize $F = 0$

$w_i^1 \;=\; \frac{1}{N}$ weight of datapoint x_i

for $k = 1, 2, \ldots M$

learn classifier for $\mathcal{D}$ with weights $w^k \;\Rightarrow\; f_k$

compute $e_k \;=\; \mathrm{Err}(f_k) \;=\; \Sigma_{i=1}^{N} w_i^k \mathbf{1}_{f(x_i) \neq y_i} \;<\; \frac{1}{2}$

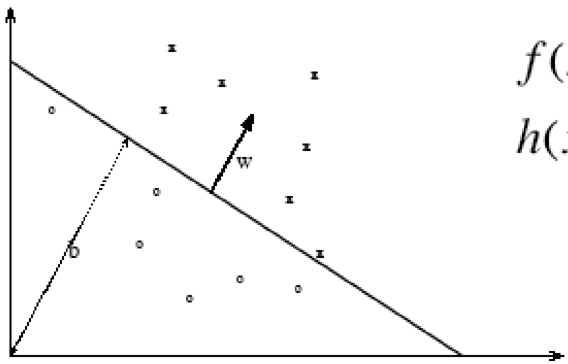
compute coefficient of f_k: $c_k \;=\; \log \frac{1-e_k}{e_k} \;>\; 0$

compute new weights $w_i^{k+1} \;=\; \begin{cases} w_i^k & \text{if } f_k(x_i) = y_i \\ w_i^k e^{c_k} & \text{if } f_k(x_i) \neq y_i \end{cases}$

Output $F(x) \;=\; \Sigma_{k=1}^{M} c_k f_k(x)$

Perceptron

- Linear Separation of the input space



$$f(x) = \langle w, x \rangle + b$$

$$h(x) = sign(f(x))$$

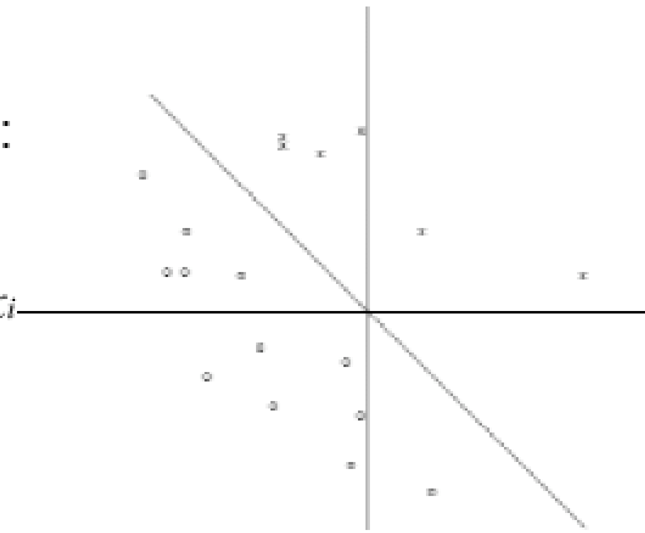
Perceptron Algorithm

Update rule (ignoring threshold):

- if $y_i(\langle w_k, x_i \rangle) \leq 0$

  then $w_{k+1} \leftarrow w_k + \eta y_i x_i$

  $k \leftarrow k+1$



Perceptron filter:

$$w = \sum \alpha_i y_i x_i$$

$$\alpha_i \geq 0$$

Dual representation:

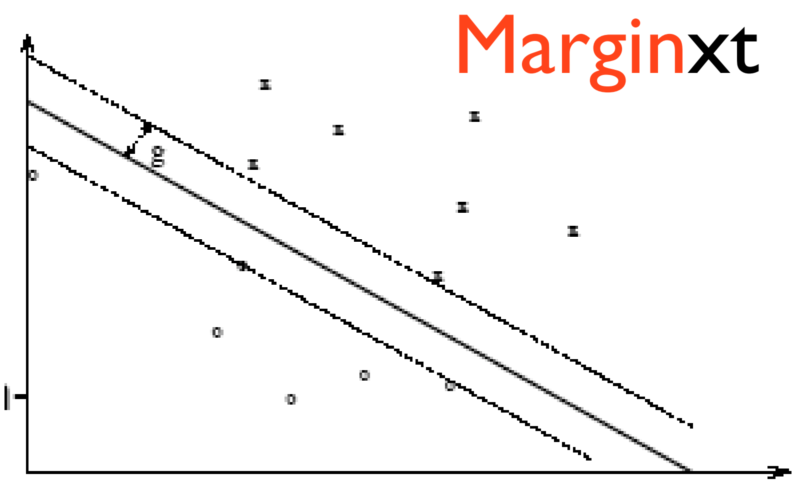
$$f(x) = \langle w, x \rangle + b = \sum \alpha_i y_i \langle x_i, x \rangle + b$$

$$w = \sum \alpha_i y_i x_i$$

Dual update rule:

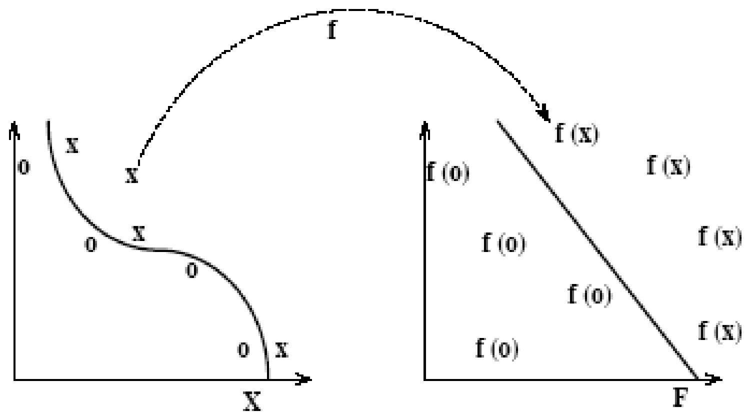
$$y_i \sum_j \alpha_j y_j \langle x_j, x_i \rangle + b \leq 0$$

$$\alpha_i \leftarrow \alpha_i + \eta$$



- Map data into a feature space where they are linearly separable

$$x \longrightarrow \phi(x)$$

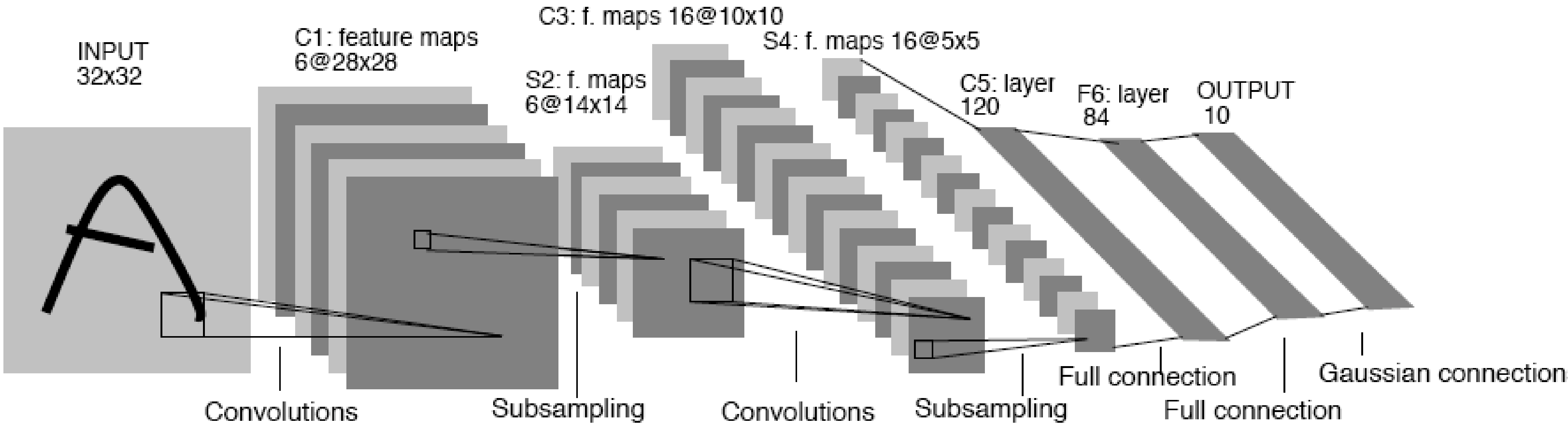


Kernels

- One can use LLMs in a feature space by simply rewriting it in dual representation and replacing dot products with kernels:

$$\langle x_1, x_2 \rangle \leftarrow K(x_1, x_2) = \langle \phi(x_1), \phi(x_2) \rangle$$

Convolutional Network:



$$X_n = F_n(W_n, X_{n-1}),$$

Back-propagation rule:

$$\frac{\partial E^p}{\partial W_n} = \frac{\partial F}{\partial W}(W_n, X_{n-1}) \frac{\partial E^p}{\partial X_n}$$

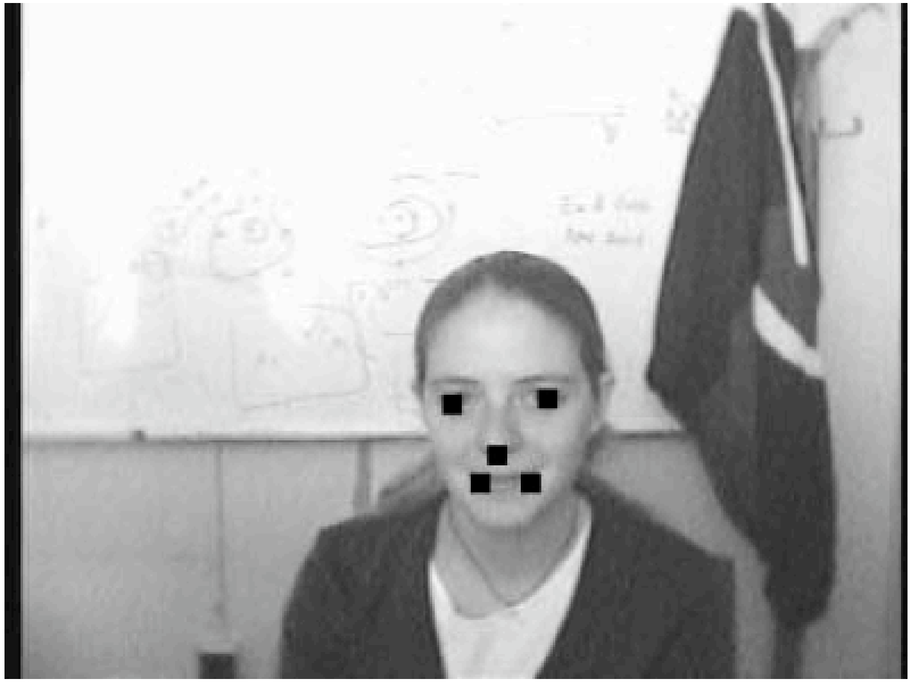
$$\frac{\partial E^p}{\partial X_{n-1}} = \frac{\partial F}{\partial X}(W_n, X_{n-1}) \frac{\partial E^p}{\partial X_n}$$

Flexible Multiple class object Recognition

- Image features:
  - Interests points, image patch (parts)
- Formulation:
  - Graphical model: HMM, (mixture) Tree, MRF; Decision (alternating) tree
- Learning Inference Technique:
  - Dynamic prog., EM, Variational Approach,

Flexible Multiple class object Recognition

- mixture of tree: Ioffe&Forsyth
- Tree structured MRF: Felzenszwalb&Huttenlocher
- General Graphical model: Fergus&Perona&Zisserman
- Multiple class: Mahamud&Hebert&Lafferty



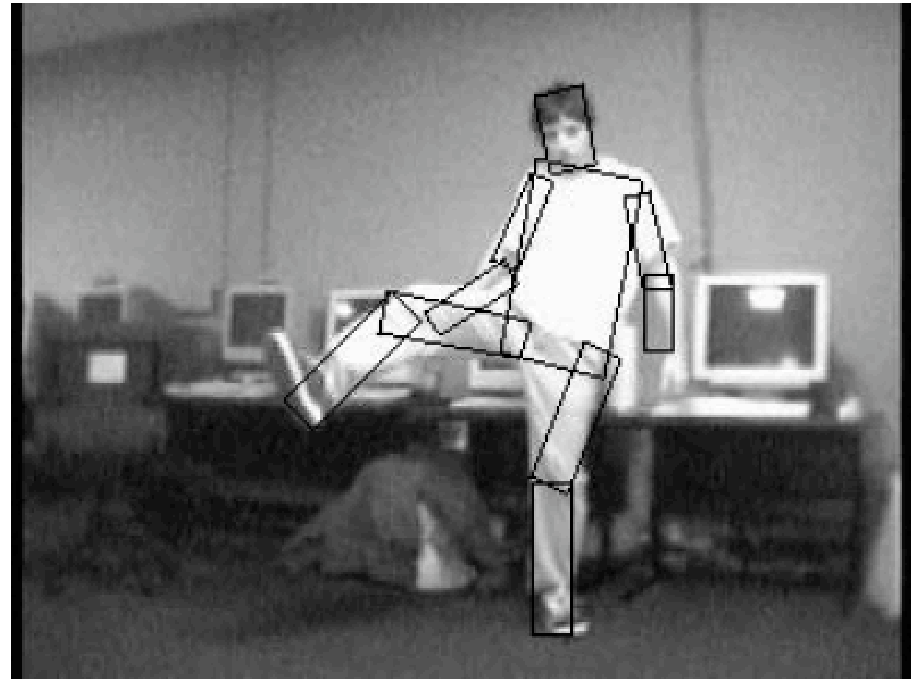
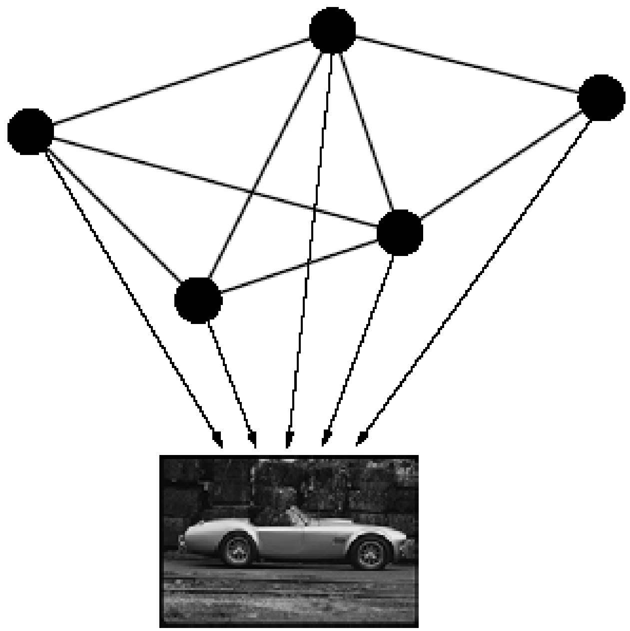


Image Likelihood $p(I|L,\theta)$ Posterior $p(L|I,\theta)$

MAP estimation: find the most likely L

Sampling: find other good matches

Model estimation: find the theta that fits



$$p(L|I,\theta) \propto p(I|L,\theta)p(L|\theta),$$

$$p(I|L,u) \propto \prod_{i=1}^{n} p(I|l_i,u_i).$$

$$p(L|E,c) = \prod_{(v_i,v_j)\in E} p(l_i,l_j|c_{ij}).$$

Model learning: $$p(I^1,\ldots,I^m,L^1,\ldots,L^m|\theta) = \prod_{k=1}^{m} p(I^k,L^k|\theta),$$

$$p(I,L|\theta) = p(I|L,\theta)p(L|\theta)$$

$$\theta^* = \arg\max_{\theta} \prod_{k=1}^{m} p(I^k|L^k,\theta) \prod_{k=1}^{m} p(L^k|\theta).$$

Finding max. likelihood:

$$L^* = \arg\max_L p(L|I,\theta) = \arg\max_L p(I|L,\theta)p(L|\theta).$$

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

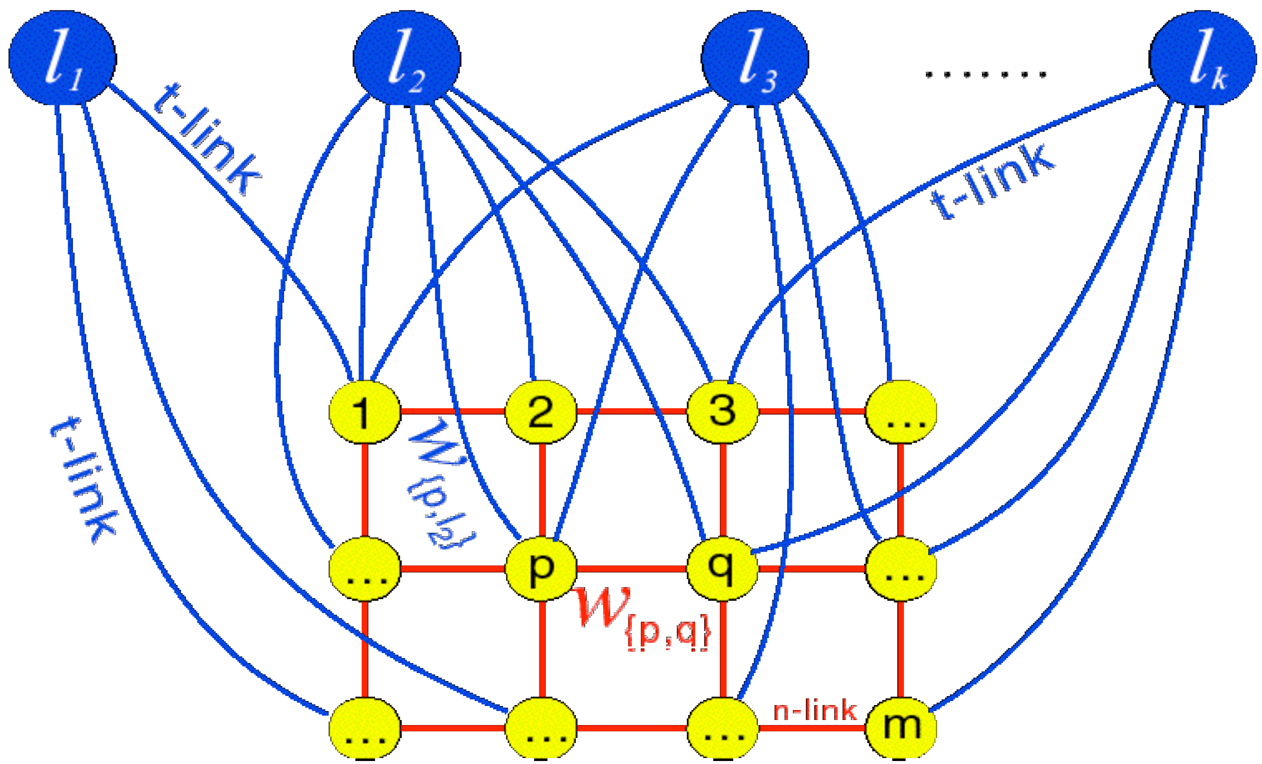
$$L^* = \arg\min_L \left(\sum_{i=1}^{n} m_i(l_i) + \sum_{(v_i,v_j)\in E} d_{ij}(l_i,l_j) \right)$$

Text

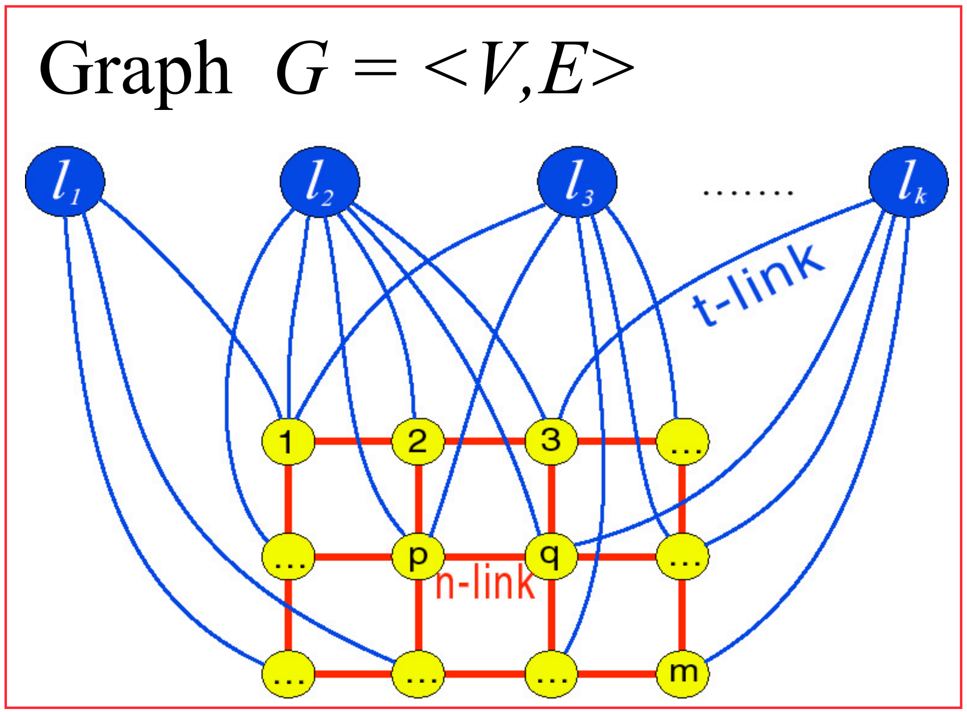
This is the a MRF, we are home!

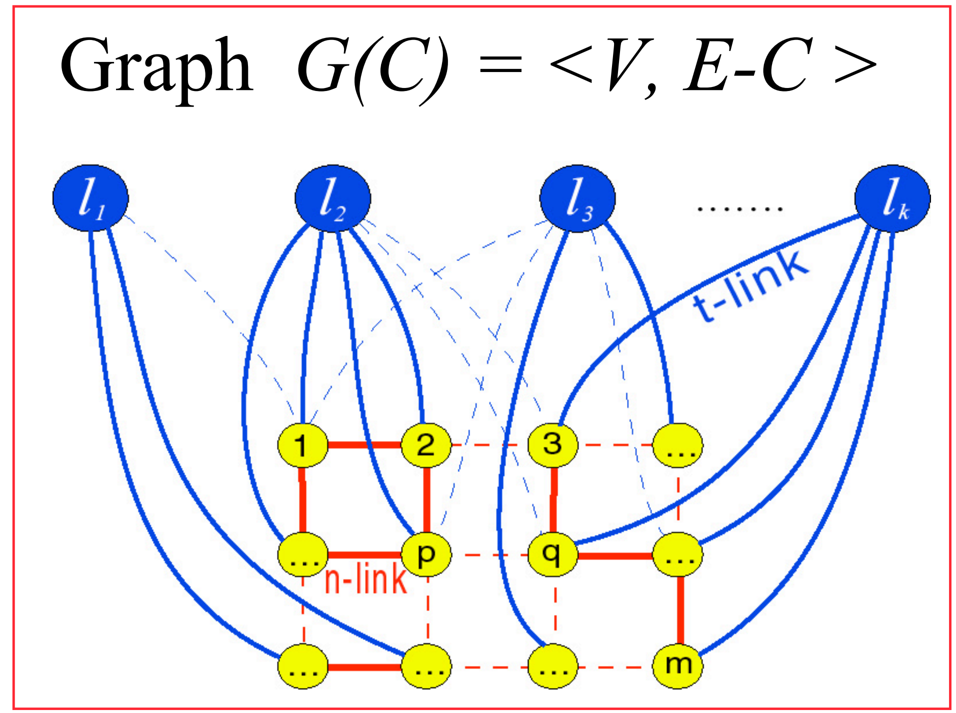
$$L^* = \arg\min_L \left(\sum_{i=1}^{n} m_i(l_i) + \sum_{(v_i,v_j)\in E} d_{ij}(l_i, l_j) \right)$$

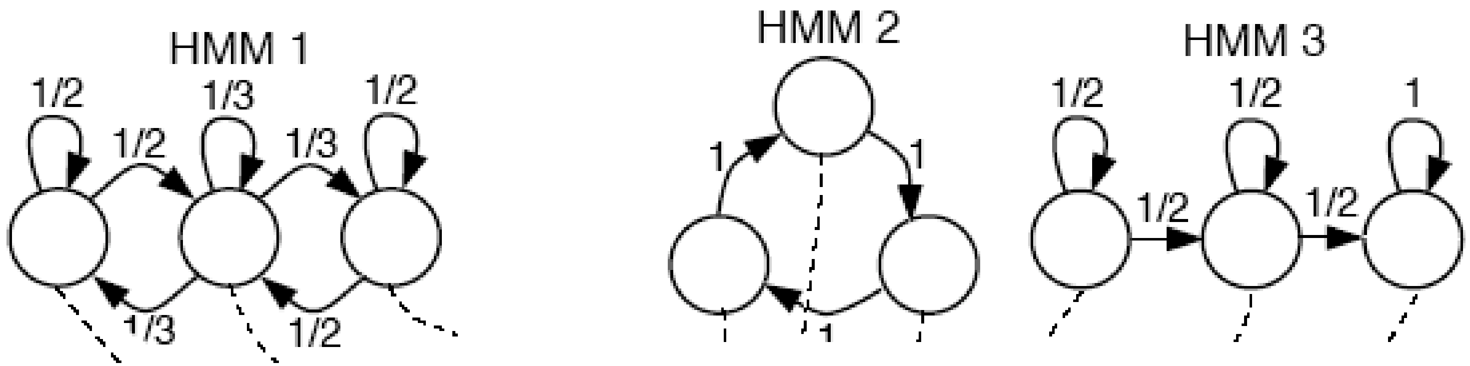
Terminals (possible disparity labels)



p-vertices
(pixels)


Graph $G = \langle V,E \rangle$
l_1
l_2
l_3
l_k
t-link
n-link
1
2
3
p
q
m


Graph $G(C) = \langle V, E\text{-}C \rangle$
l_1
l_2
l_3
l_k
t-link
n-link
1
2
3
p
q
m



Number of States,
number of observation
A set of State transition: $a_{ij} = p\{q_{t+1} = j | q_t = i\}, \quad 1 \leq i, j \leq N,$
A probability distribution $b_j(k) = p\{o_t = v_k | q_t = j\}, \quad 1 \leq j \leq N, \quad 1 \leq k \leq M$
Initial State distribution: $\pi_i = p\{q_1 = i\}, \quad 1 \leq i \leq N$

$$O = o_1, o_2, \ldots, o_T$$

(1) Evaluation Problem

Given an HMM λ and a sequence of observations O , what is the probability that the observations are generated by the model, $p\{O|\lambda\}$?

(2)The Decoding Problem

Given a model and a sequence of observations , what is the most likely state sequence in the model that produced the observations?

(3)The Learning Problem

Given a model and a sequence of observations , how should we adjust the model parameters $\{A, B, \pi\}$ in order to maximize $p\{O|\lambda\}$.

Evaluation:

$$\alpha_t(i) = p\{o_1, o_2, \ldots, o_t, q_t = i|\lambda\}$$

$$\alpha_{t+1}(j) = b_j(o_{t+1}) \sum_{i=1}^{N} \alpha_t(i) a_{ij}, \ 1 \leq j \leq N, \ 1 \leq t \leq T-1 \quad (1.2)$$

$$\alpha_1(j) = \pi_j b_j(o_1), \ 1 \leq j \leq N$$

Forward equation: $$p\{O|\lambda\} = \sum_{i=1}^{N} \alpha_T(i). \quad (1.3)$$

$$\beta_t(i) = p\{o_{t+1}, o_{t+2}, \ldots, o_T|q_t = i, \lambda\} \quad (1.4)$$

$$\beta_t(i) = \sum_{j=1}^{N} \beta_{t+1}(j) a_{ij} b_j(o_{t+1}), \ 1 \leq i \leq N, \ 1 \leq t \leq T-1 \quad (1.5)$$

$$\beta_T(i) = 1, \ 1 \leq i \leq N$$

Decoding

In this case We want to find the most likely state sequence for a given sequence of observations

$$\delta_t(i) = \max_{q_1 q_2 \cdots q_{t-1}} p\{q_1, q_2, \ldots, q_{t-1}, q_t = i, o_1, o_2, \ldots, o_{t-1} | \lambda\},$$

$$\delta_{t+1}(j) = b_j(o_{t+1}) \left[\max_{1 \le i \le N} \delta_t(i) a_{ij} \right], \;\; 1 \le i \le N, \;\; 1 \le t \le T-1 \qquad (1.8)$$

$$\delta_1(j) = \pi_j b_j(o_1), \;\; 1 \le j \le N$$

$$j^* = \arg \max_{1 \le j \le N} \delta_T(j),$$

Learning the parameters: Maximum Likelihood(ML)

In ML we try to maximize the probability of a given sequence of observations , belonging to a given class *w*, given the HMM of the class *w*, wrt the parameters of the model . This probability is the total likelihood of the observations and can be expressed mathematically as

$$L_{tot} = p\{O^w | \lambda_w\}$$

However since we consider only one class *w* at a time we can drop the subscript and superscript '*w*'s. Then the ML criterion can be given as,

$$L_{tot} = p\{O | \lambda\}$$

there is no known way to analytically solve for the model \lambda , which maximize the quantity Ltot . But we can choose model parameters such that it is locally maximized, using an iterative procedure, like Baum-Welch method

EM algorithm:

$$Q(\lambda, \bar{\lambda}) = \sum_{q} p\{q | O, \lambda\} \log[p\{O, q, \bar{\lambda}\}]$$

EM for HMM case:

$$Q(\lambda, \lambda') = \sum_{q \in \mathcal{Q}} \log P(O, q|\lambda) P(O, q|\lambda') \qquad \lambda = (A, B, \pi)$$

HMM pdf:

$$P(O, q|\lambda) = \pi_{q_0} \prod_{t=1}^{T} a_{q_{t-1} q_t} b_{q_t}(o_t)$$

$$Q(\lambda, \lambda') = \sum_{q \in \mathcal{Q}} \log \pi_{q_0} P(O, q|\lambda') +$$

$$\sum_{q \in \mathcal{Q}} \left(\sum_{t=1}^{T} \log a_{q_{t-1} q_t} \right) p(O, q|\lambda') +$$

$$\sum_{q \in \mathcal{Q}} \left(\sum_{t+1}^{T} \log b_{q_t}(o_t) \right) P(O, q|\lambda')$$

HMM case:

First Term: $$\sum_{q\in Q}\log \pi_{q_0} P(O,q|\lambda') = \sum_{i=1}^{N}\log \pi_i p(O,q_0=i|\lambda')$$

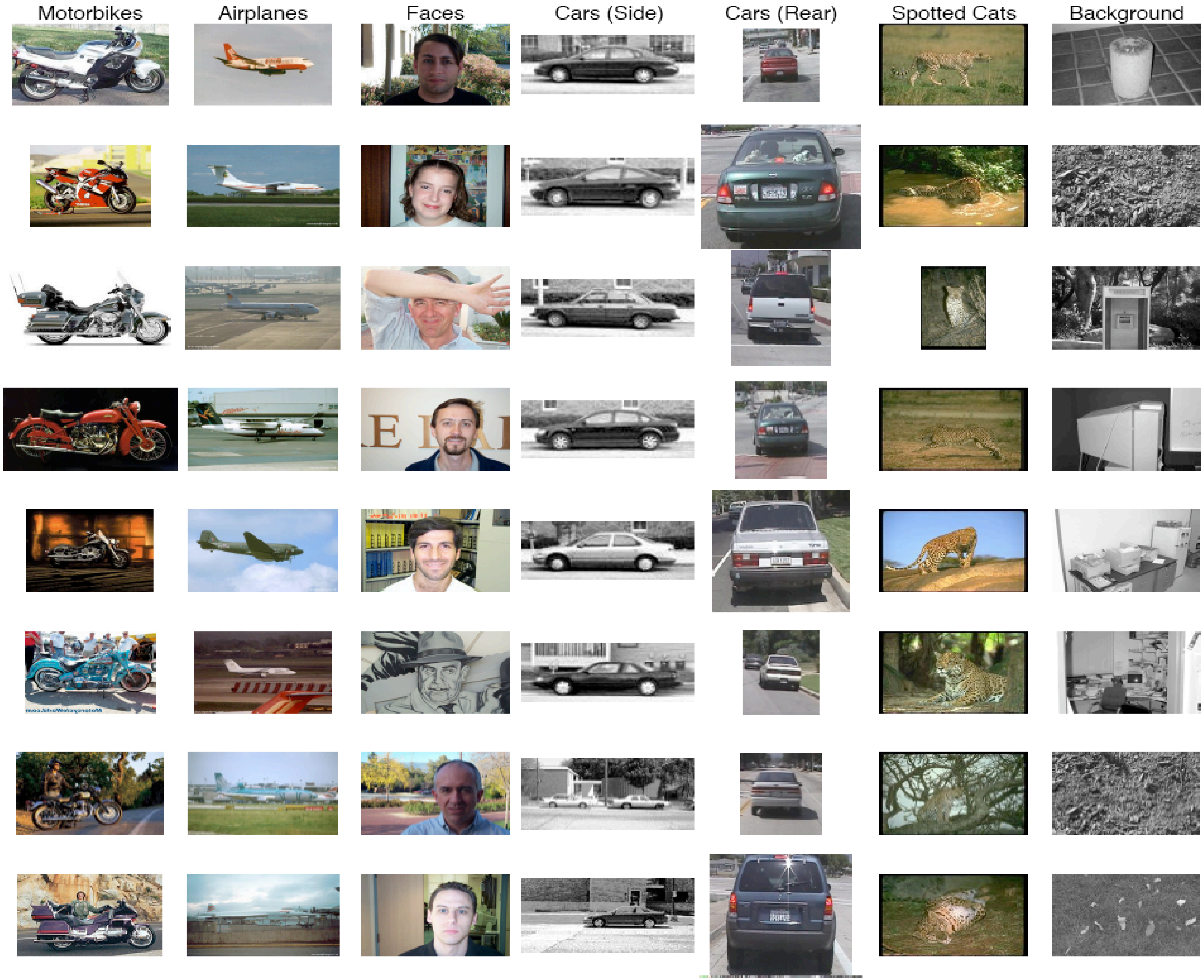
Optimal solution: $$\frac{\partial}{\partial \pi_i}\left(\sum_{i=1}^{N}\log \pi_i p(O,q_0=i|\lambda') + \gamma(\sum_{i=1}^{N}\pi_i - 1)\right) = 0$$

=> $$\pi_i = \frac{P(O,q_0=i|\lambda')}{P(O|\lambda')}$$

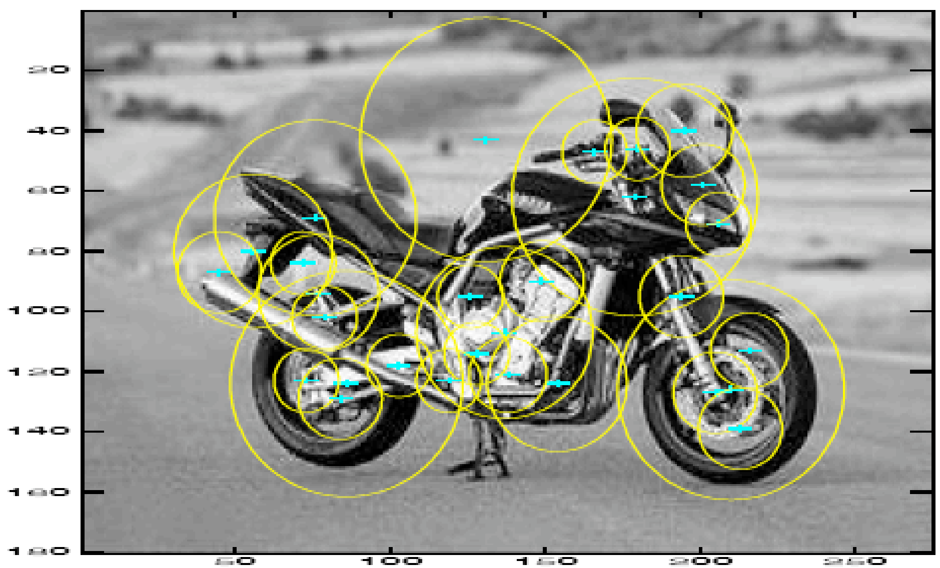
Second Term:

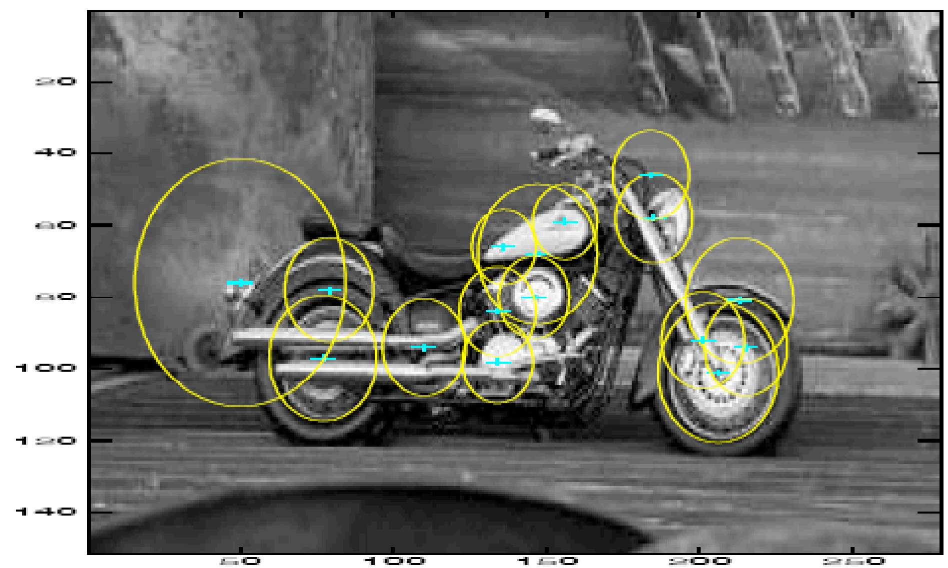
$$\sum_{q\in Q}\left(\sum_{t=1}^{T}\log a_{q_{t-1}q_t}\right) p(O,q|\lambda') = \sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{t=1}^{T}\log a_{ij} P(O,q_{t-1}=i,q_t=j|\lambda')$$

=> $$a_{ij} = \frac{\sum_{t=1}^{T} P(O,q_{t-1}=i,q_t=j|\lambda')}{\sum_{t=1}^{T} P(O,q_{t-1}=i|\lambda')}$$

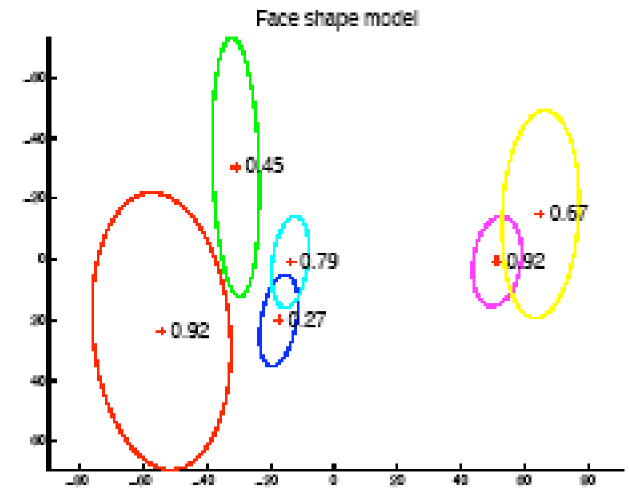

Motorbikes
Airplanes
Faces
Cars (Side)
Cars (Rear)
Spotted Cats
Background

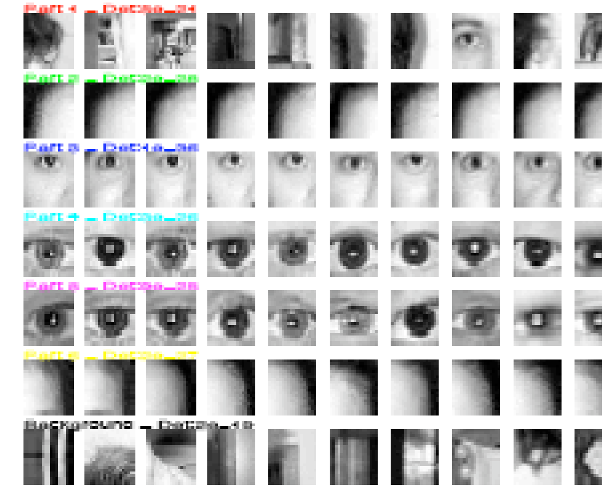
h: mapping parts to feature: P: -> N(x,s,a)

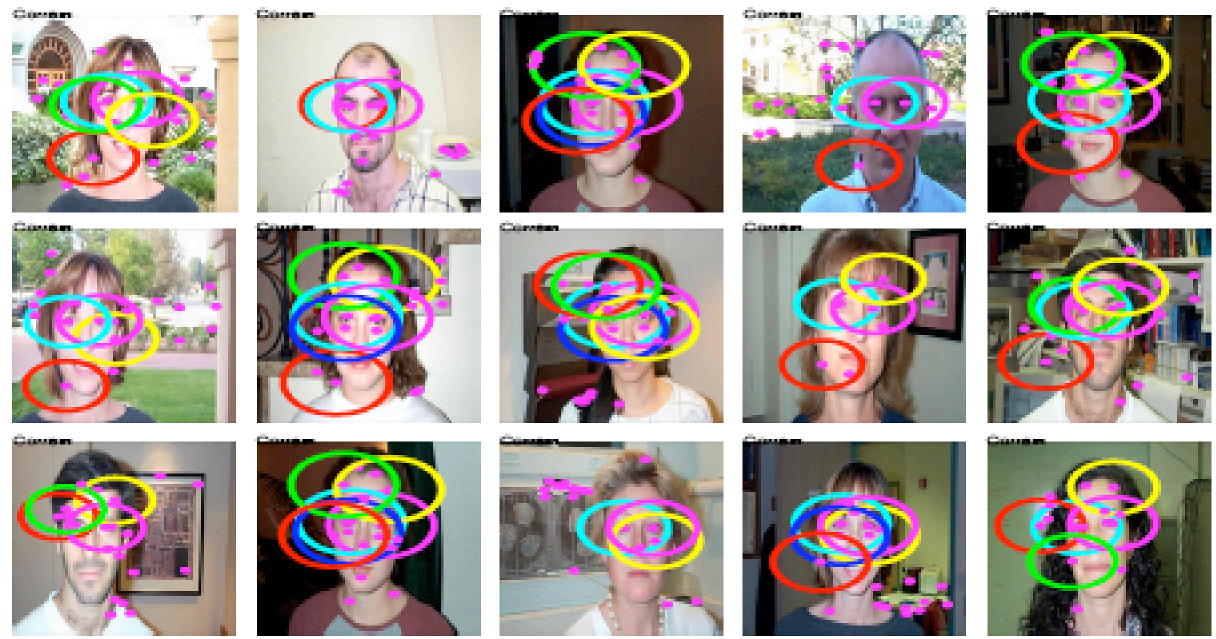




$$p(\mathbf{X},\mathbf{S},\mathbf{A}|\,\theta) = \sum_{\mathbf{h}\in H} p(\mathbf{X},\mathbf{S},\mathbf{A},\mathbf{h}|\,\theta) = \sum_{\mathbf{h}\in H} \underbrace{p(\mathbf{A}|\mathbf{X},\mathbf{S},\mathbf{h},\theta)}_{Appearance}\,\underbrace{p(\mathbf{X}|\mathbf{S},\mathbf{h},\theta)}_{Shape}\,\underbrace{p(\mathbf{S}|\mathbf{h},\theta)}_{Rel.\ Scale}\,\underbrace{p(\mathbf{h}|\theta)}_{Other}$$

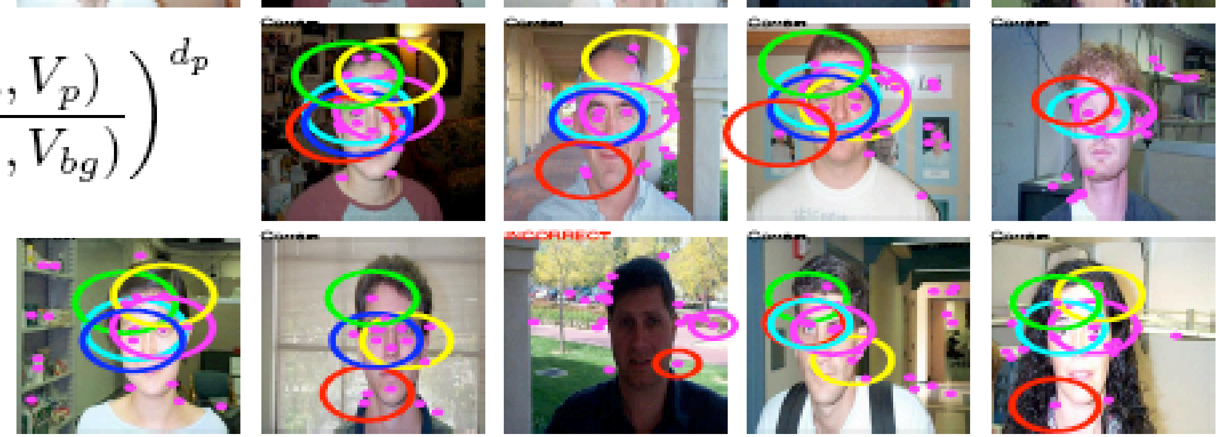






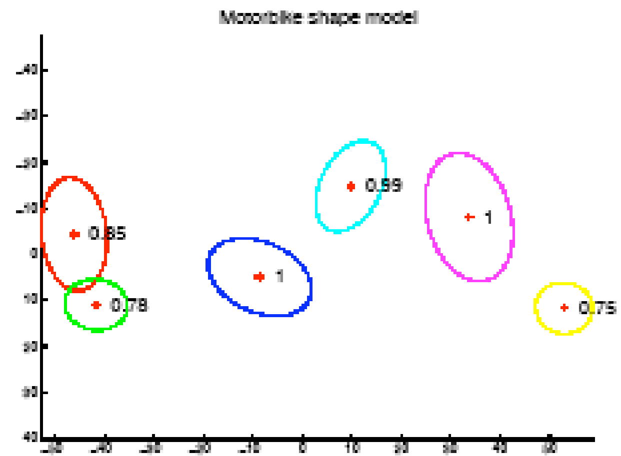
Appearance

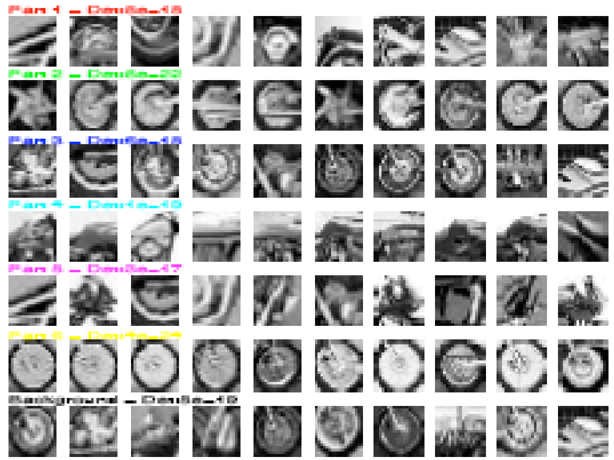
$$\frac{p(\mathbf{A}|\mathbf{X},\mathbf{S},\mathbf{h},\theta)}{p(\mathbf{A}|\mathbf{X},\mathbf{S},\mathbf{h},\theta_{bg})} = \prod_{p=1}^{P}\left(\frac{\mathrm{G}(\mathbf{A}(h_p)|\mathbf{c}_p,V_p)}{\mathrm{G}(\mathbf{A}(h_p)|\mathbf{c}_{bg},V_{bg})}\right)^{d_p}$$

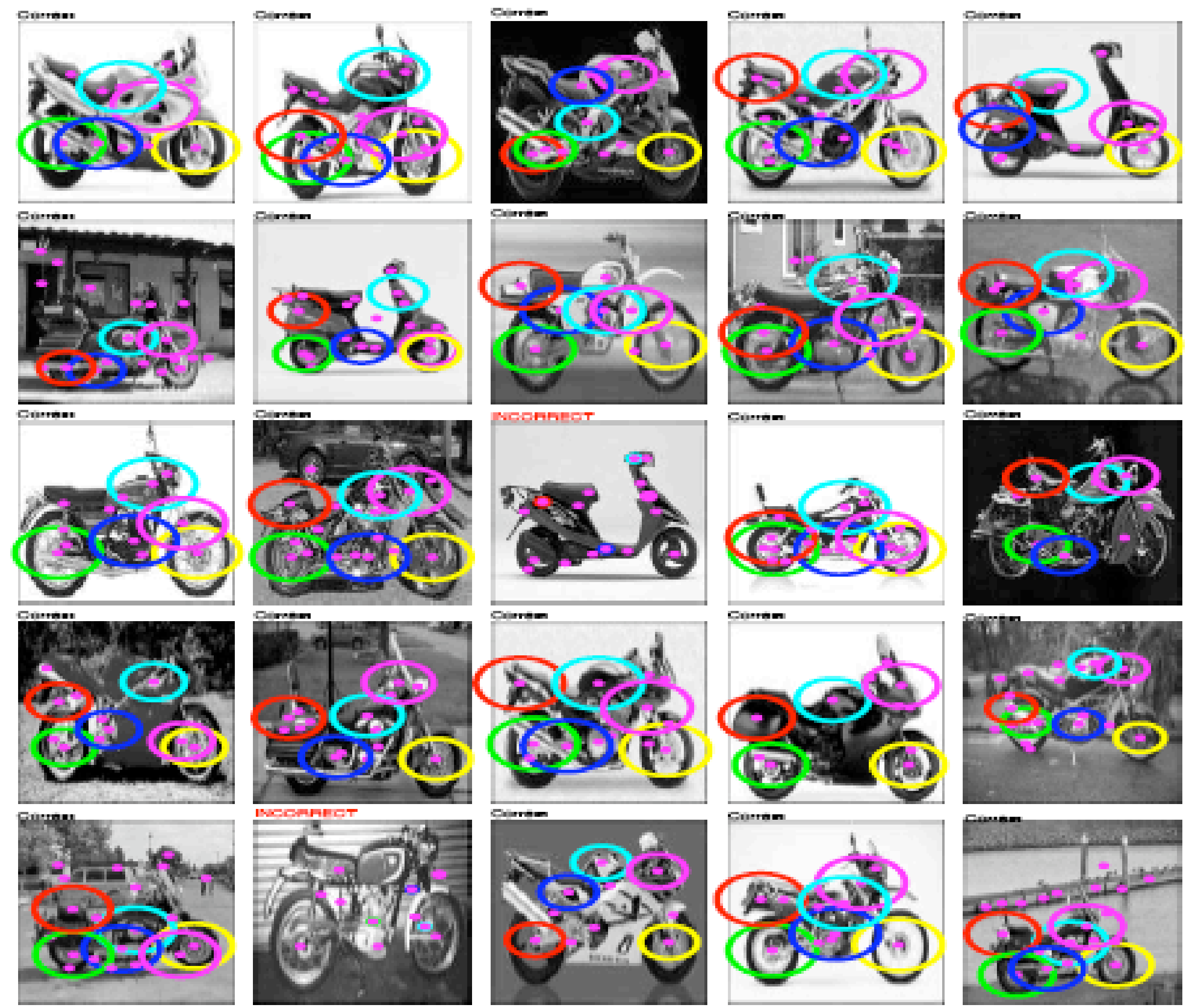


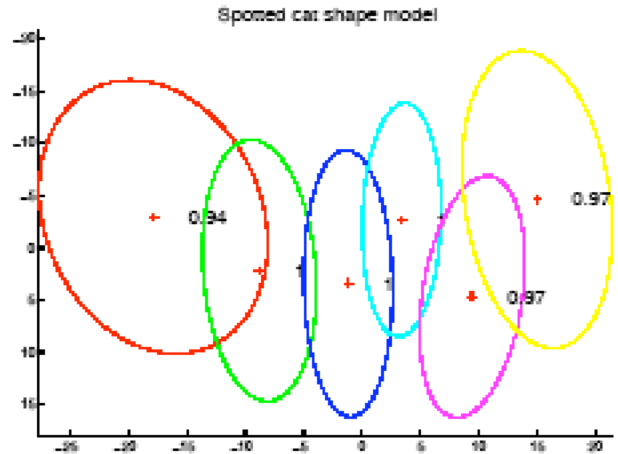
Shape

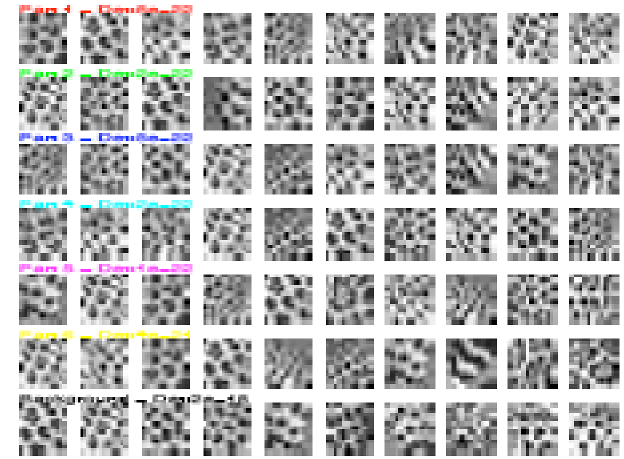
$$\frac{p(\mathbf{X}|\mathbf{S},\mathbf{h},\theta)}{p(\mathbf{X}|\mathbf{S},\mathbf{h},\theta_{bg})} = \mathrm{G}(\mathbf{X}(\mathbf{h})|\boldsymbol{\mu},\Sigma)\,\alpha^f$$











$$\theta = \{\boldsymbol{\mu}, \Sigma, \mathbf{c}, V, M, p(\mathbf{d}|\theta), t, U\}$$

$$\hat{\theta}_{ML} = \underset{\theta}{arg\,max}\;\; p(\mathbf{X}, \mathbf{S}, \mathbf{A}|\,\theta).$$

Need to use EM

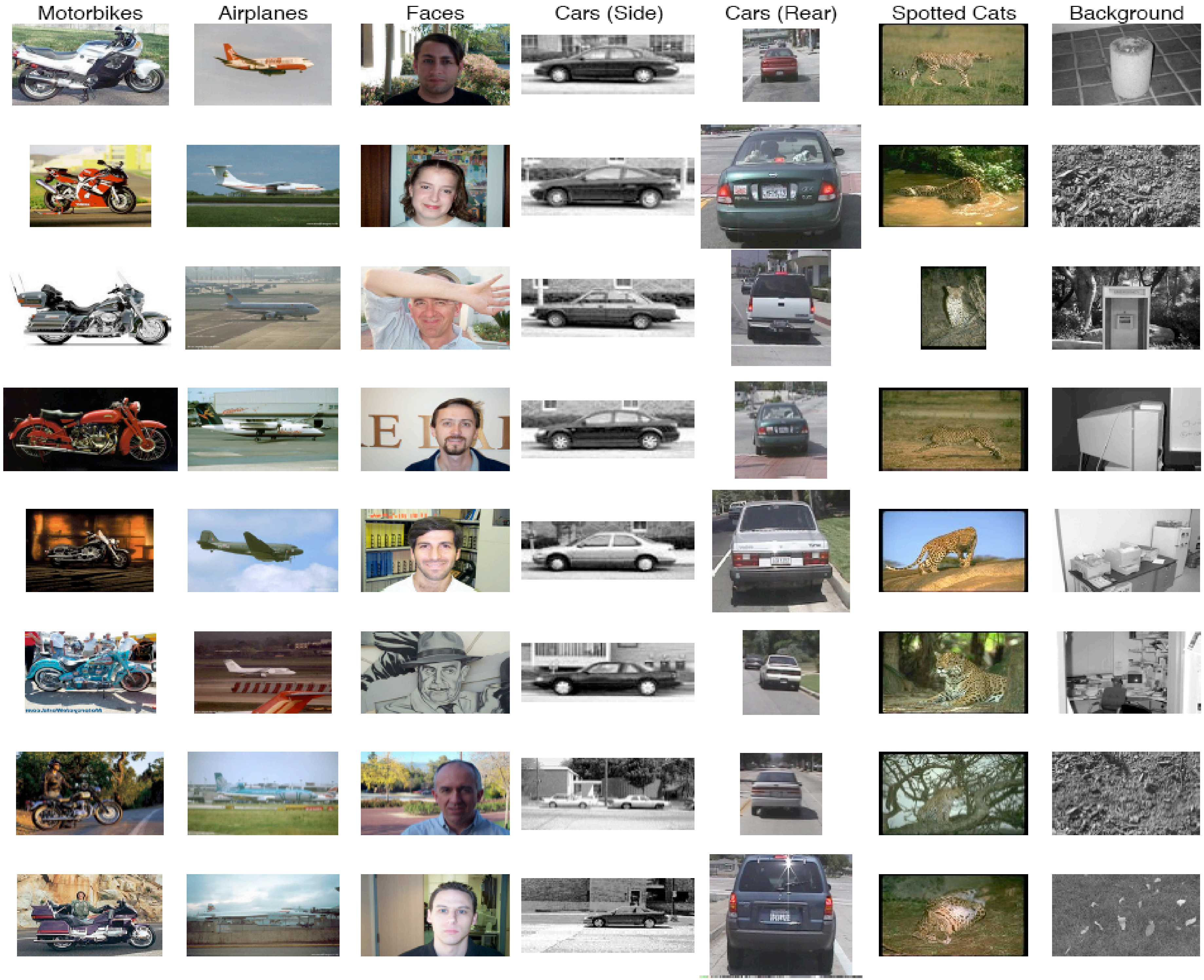

Motorbikes
Airplanes
Faces
Cars (Side)
Cars (Rear)
Spotted Cats
Background

Image Segmentation

- Image features
  - Pixel value, local filter/texture/motion features
- Formulation
  - Graph Cuts, MRF/min-cut, Random Walk, Information Bottleneck
- Inference Technique
  - Spectral graph method, Min-cut, deterministic annealing, variation approach

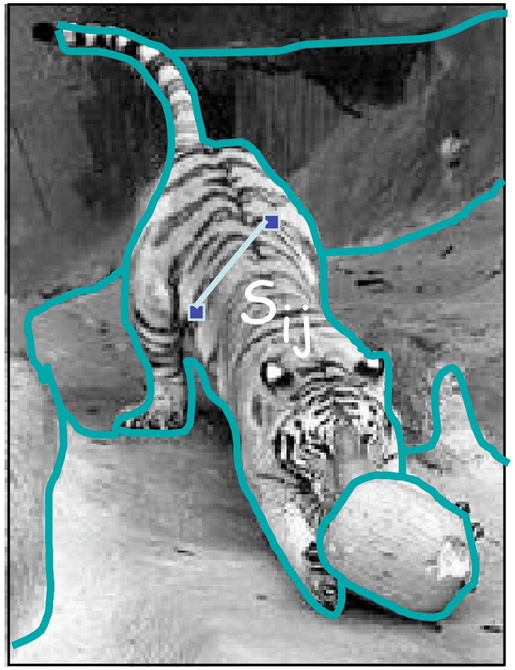
Image Segmentation

- Segmentation with graph cuts
- Segmentation and labeling with MRF, min-cut
- Segmentation with (semi) supervision
- clustering image and words
- Information Bottleneck

Image segmentation by pairwise similarities

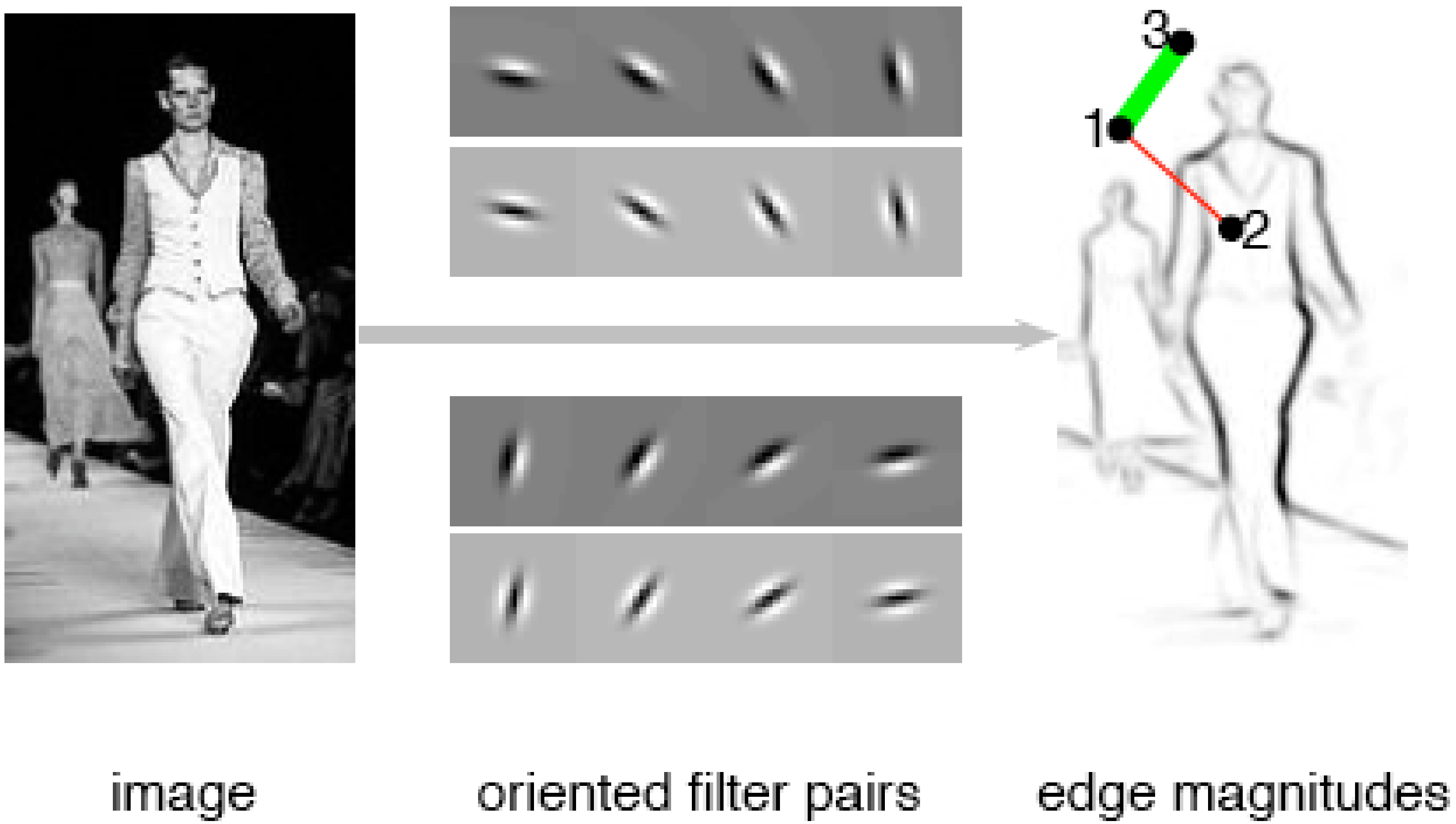
- Image = { pixels }
- Segmentation = partition of image into segments
- Similarity between pixels i and j

$$S_{ij} = S_{ji} \geq 0$$



- Objective: “similar pixels should be in the same segment, dissimilar pixels should be in different segments”

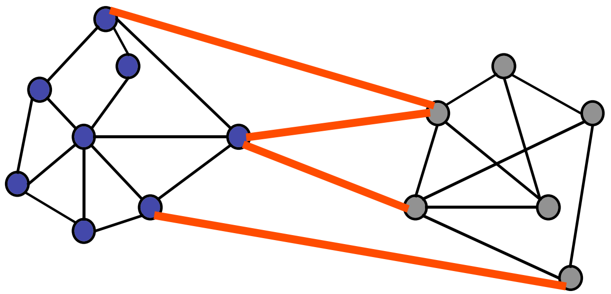
Pixel Similarity based on Intensity Edges



Cuts in a graph

- (edge) cut = set of edges whose removal makes a graph disconnected
- weight of a cut

$$cut(A, B) = \sum_{i \in A, j \in B} S_{ij}$$



- the normalized cut

$$NCut(A,B) = cut(A,B)\left(\frac{1}{vol\ A} + \frac{1}{vol\ B}\right)$$

Normalized Cut As Generalized Eigenvalue problem

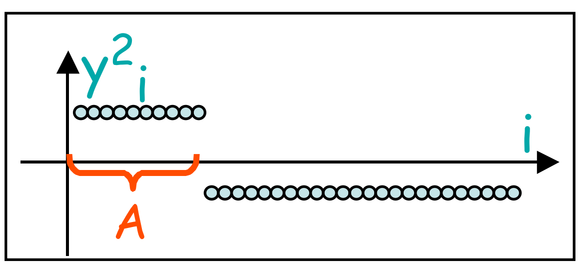
$$Ncut(\mathrm{A},\mathrm{B}) = \frac{cut(\mathrm{A},\mathrm{B})}{Vol(\mathrm{A})} + \frac{cut(\mathrm{A},\mathrm{B})}{Vol(\mathrm{B})}$$

$$= \frac{(1+x)^T (D-W)(1+x)}{k 1^T D 1} + \frac{(1-x)^T (D-W)(1-x)}{(1-k) 1^T D 1}; \quad k = \frac{\sum_{x_i > 0} D(i,i)}{\sum_i D(i,i)}$$

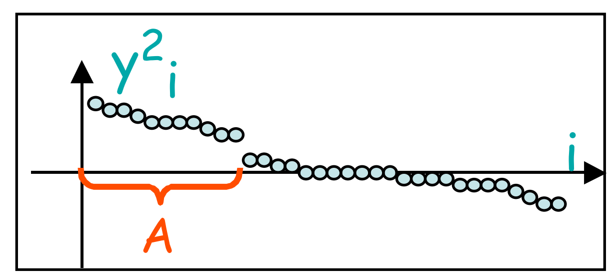
$$= \ldots$$

- after simplification, we get

$$Ncut(A,B) = \frac{y^T (D-W) y}{y^T D y}, \quad \text{with } y_i \in \{1, -b\},\ y^T D 1 = 0.$$



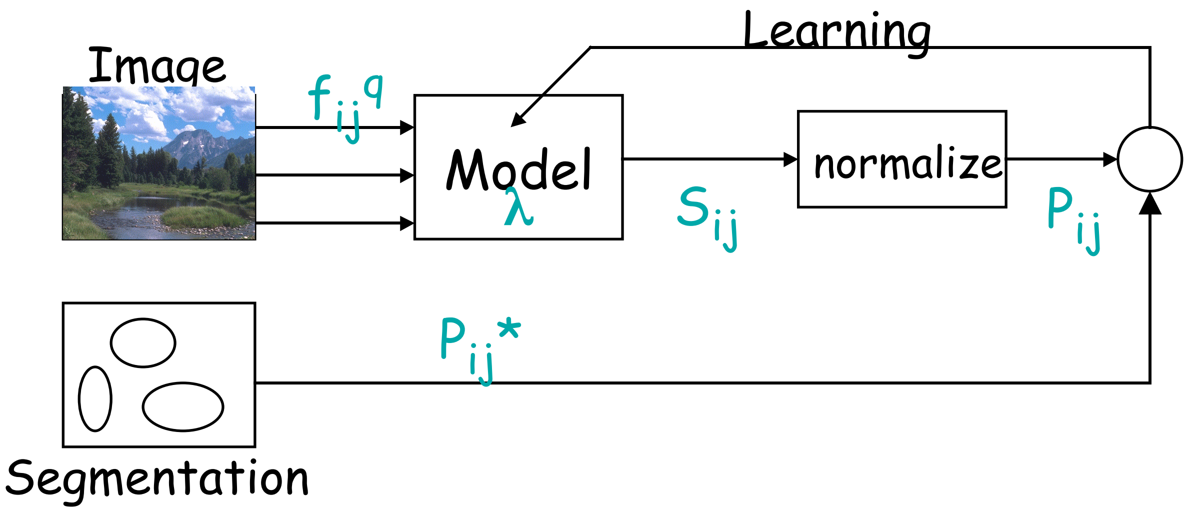
$$(D-W)x = \lambda D x$$



Current result

[Folkless et.al. 03]

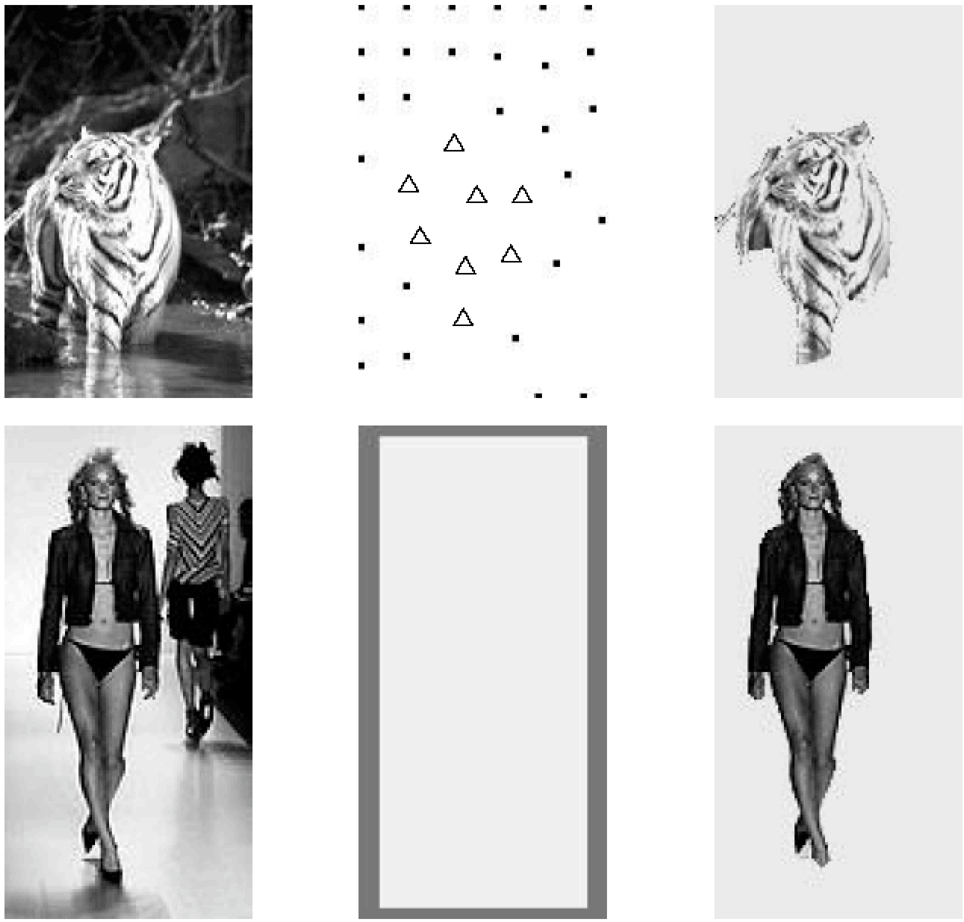
Learning image segmentation



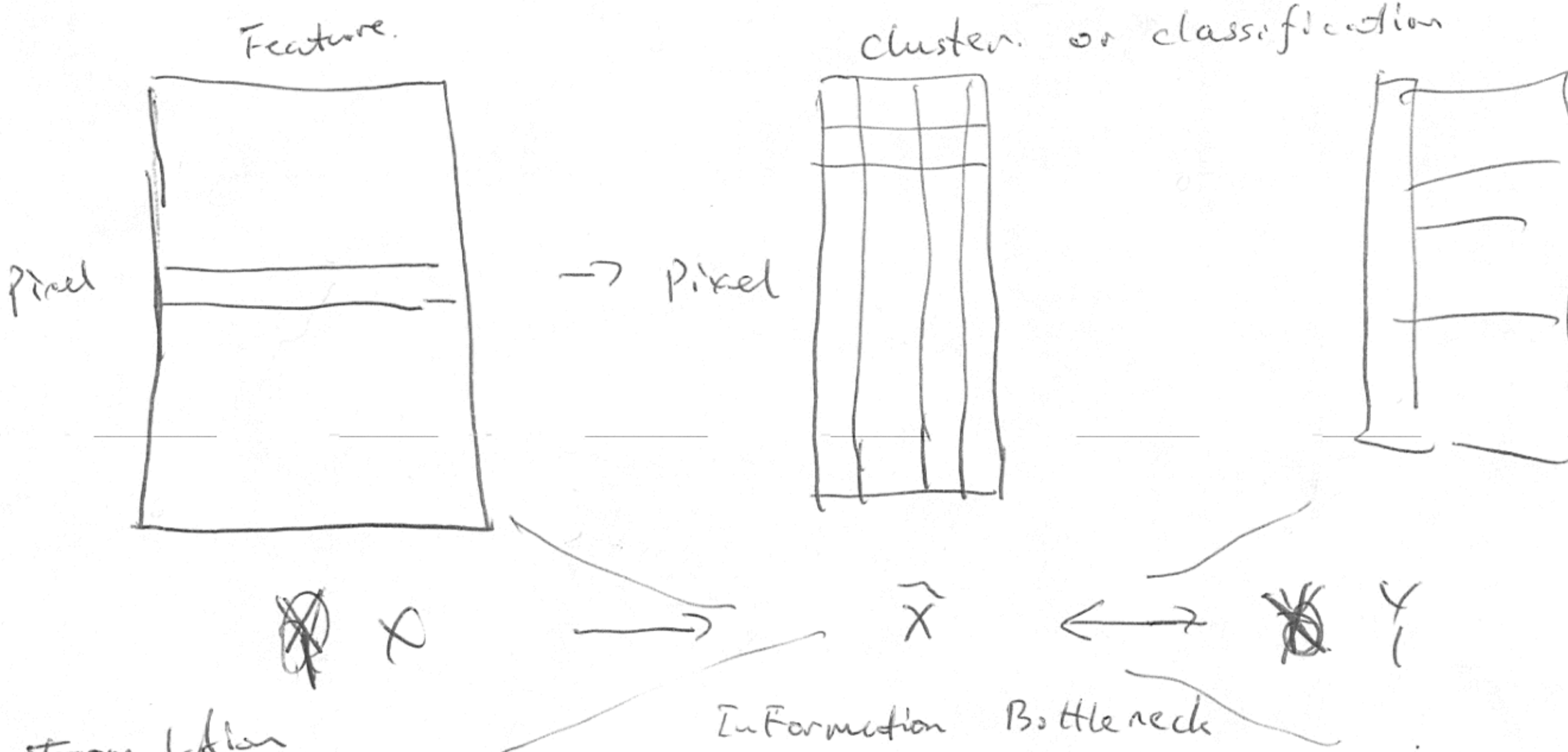
- Targets $P_{ij}^{*} = \begin{cases} 0, & j \notin A \\ \frac{1}{|A|}, & j \in A \end{cases}$ for i in segment A

- Model $S_{ij} = \exp(\sum_q \lambda_q f^q_{ij})$

Guide Grouping: Partial Grouping Cues



Information bottleneck



Formulation

$$F[p(\hat{x}|x)] = I(x;\hat{x}) + \beta \underline{\langle d(x,\hat{x})\rangle_{p(x,\hat{x})}}$$

$$\min \ I(X;\hat{X}) - \lambda I(\hat{X};Y)$$

$$I(\tilde{X}|Y) = H(Y) - H(Y|\hat{X})$$

$$H(Y|\hat{X}) = -\sum_{\hat{x},y} p(y,\hat{x}) \log p(y|\hat{x})$$

$$= -\sum_{\hat{x}\,y} \sum_{x} p(x,\hat{x},y) \log p(y|\hat{x})$$

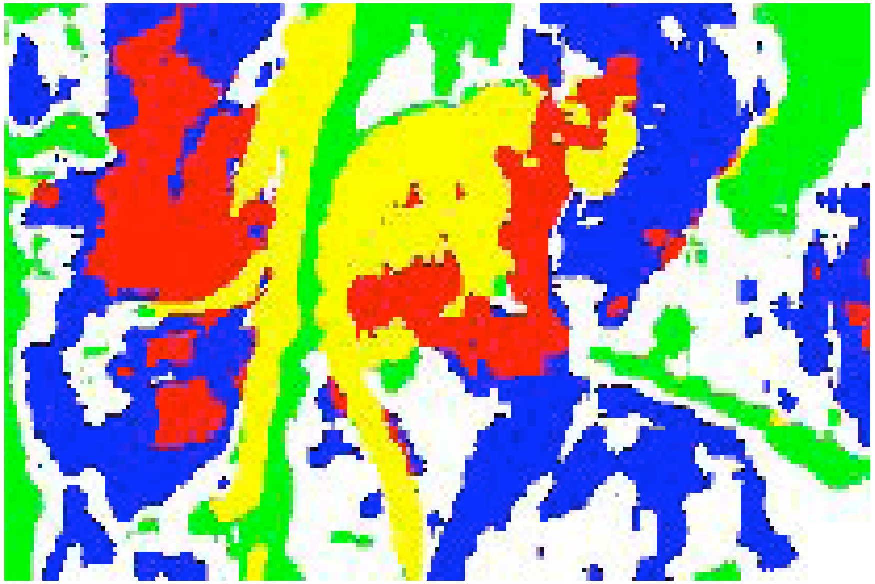
$$= -\sum_{x,y,\hat{x}} \underbrace{p(\hat{x}|x)}_{\boxed{p(\hat{x}|x,y) = p(\hat{x}|x)}} p(y|x) \underbrace{p(x)}_{\frac{1}{n}} \log p(y|\hat{x})$$

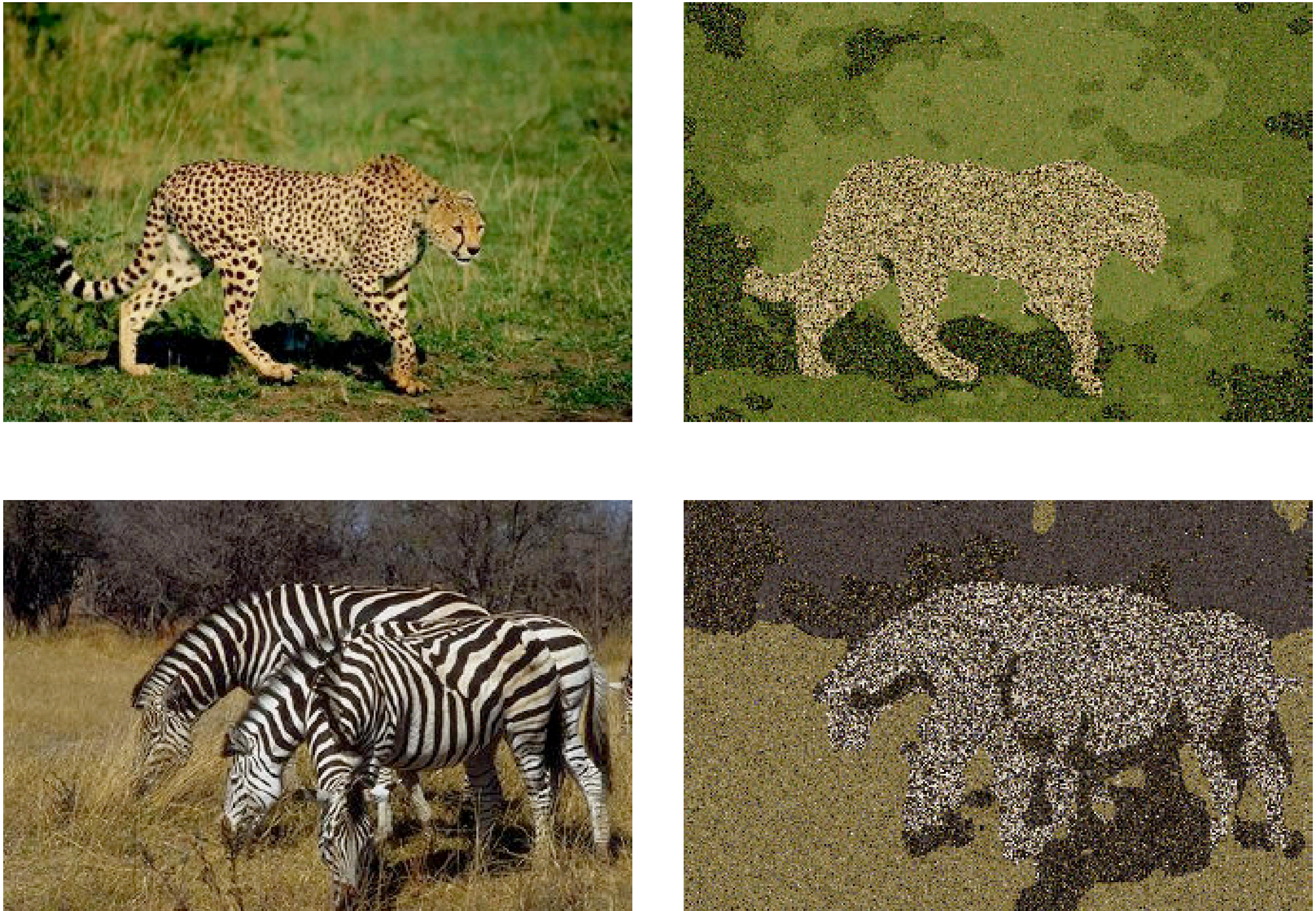
Total:

$$I(X;\hat{X}) - \lambda I(\hat{X};Y) =$$

$$= \frac{1}{n}\sum_{i=1}^{n}\sum_{v=1}^{k} \min_{v} \left[\log P_v + \lambda \sum_{j<m} \frac{n_{ij}}{n_i} \log\left(\sum_{\alpha=1} P_{\alpha|v} G_{\alpha}(j)\right)\right]$$





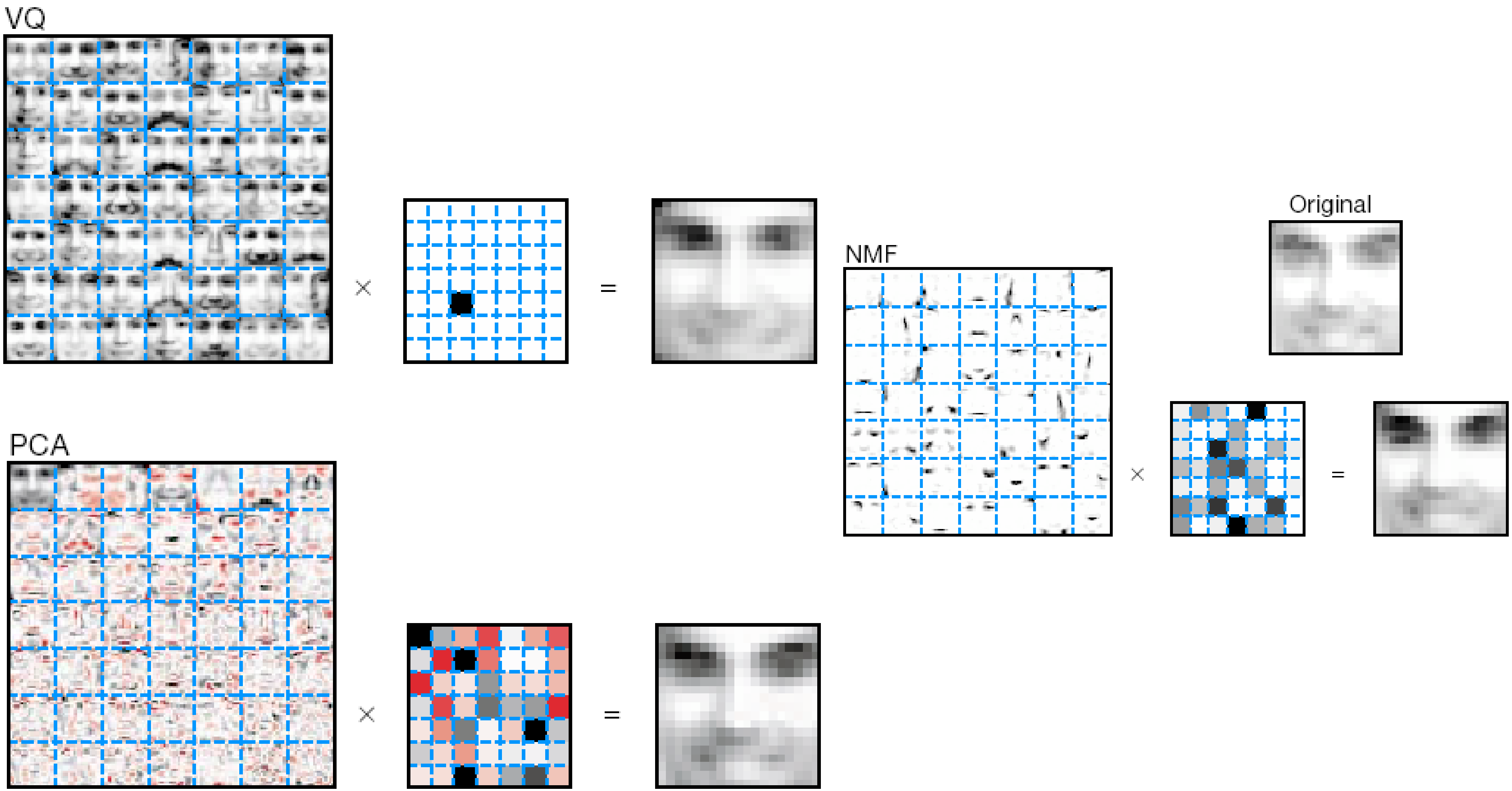



upcoming topics

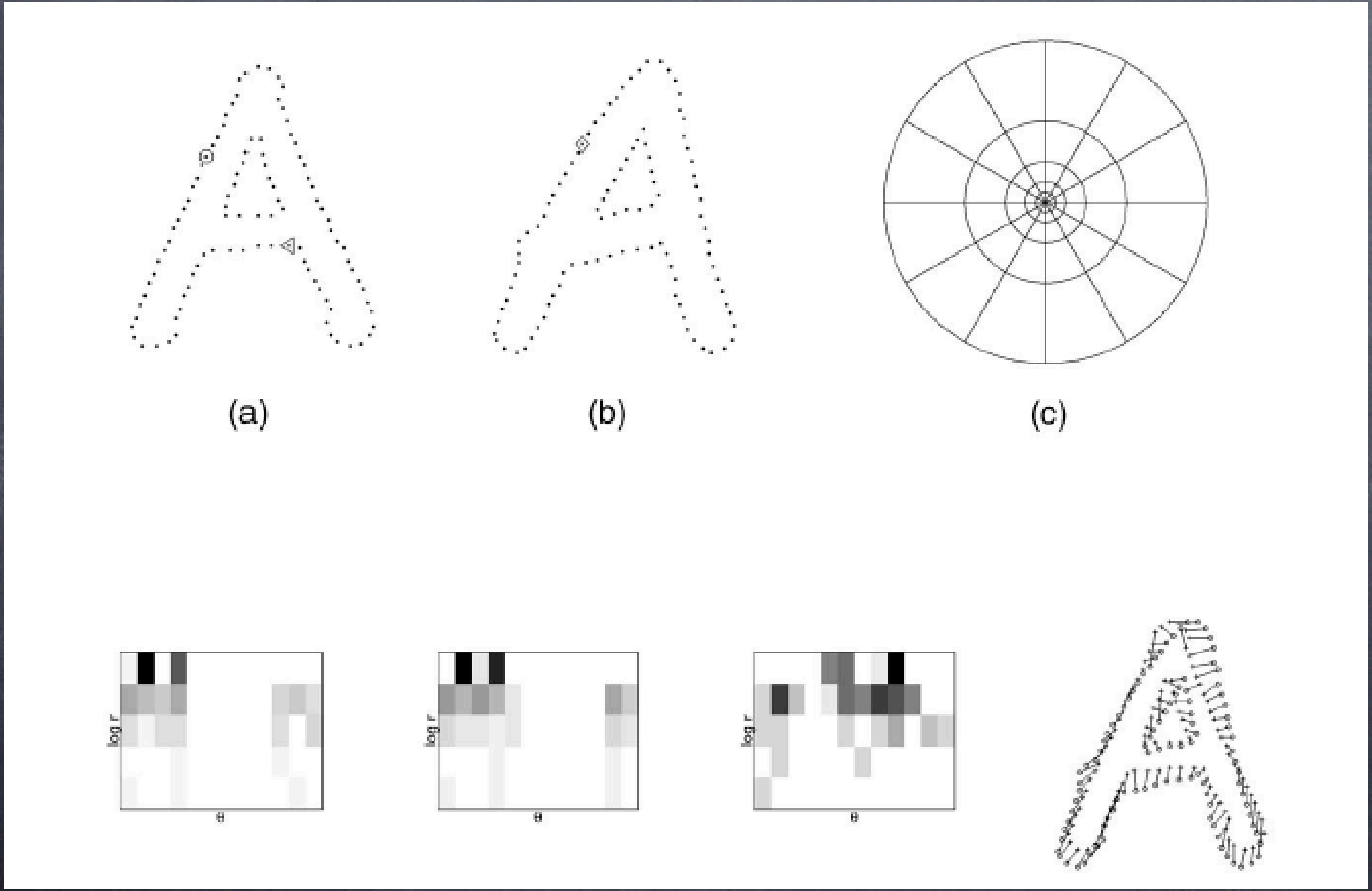
- Image Shape (geometric) features
- Human activity recognition

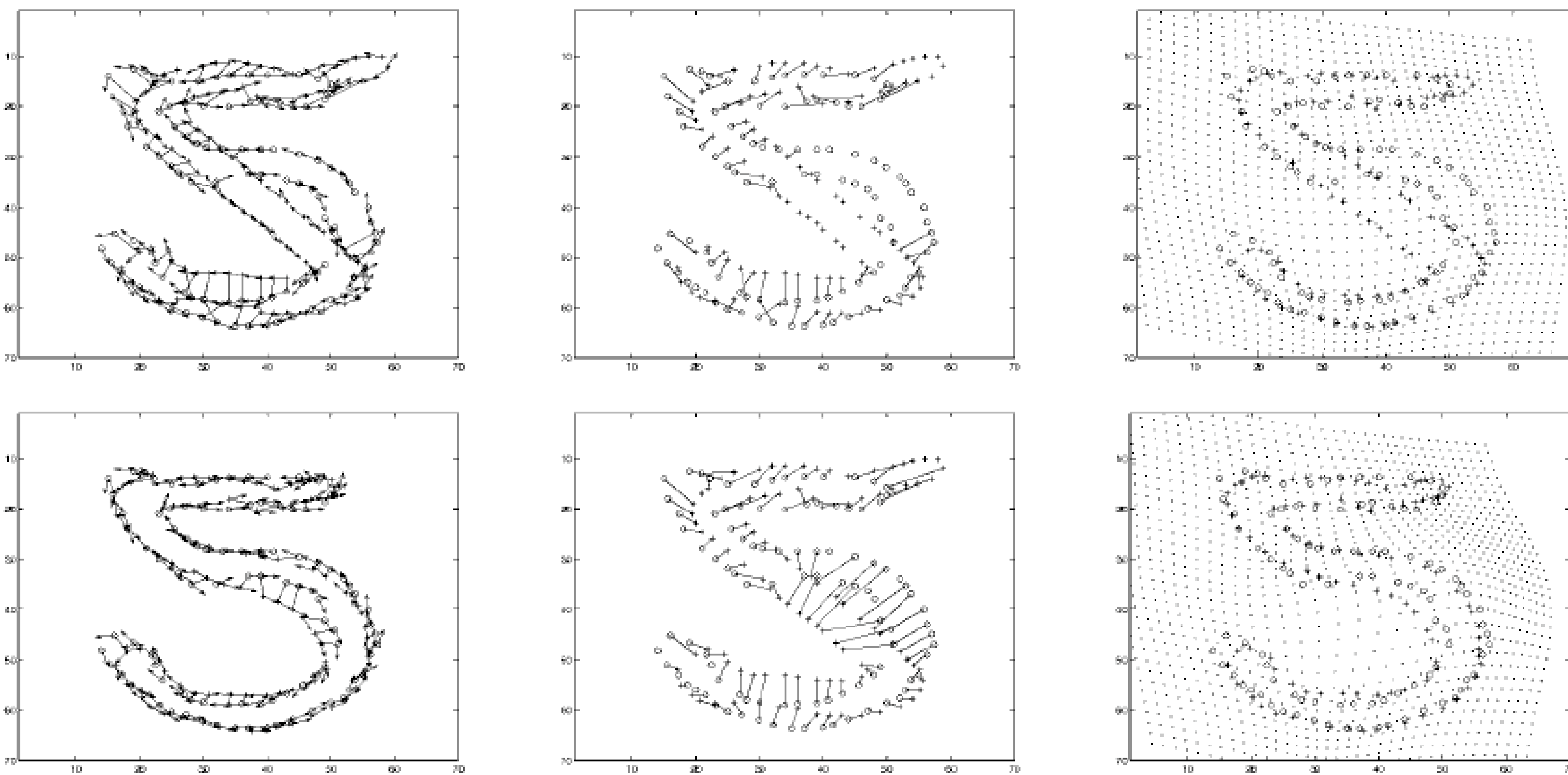
Image features/Shape

- Models of features
  - linear features/filters: PCA, ICA, Non-negative
- Geometrical:
  - Shape-context


VQ
×
=
NMF
Original
×
=
PCA
×
=

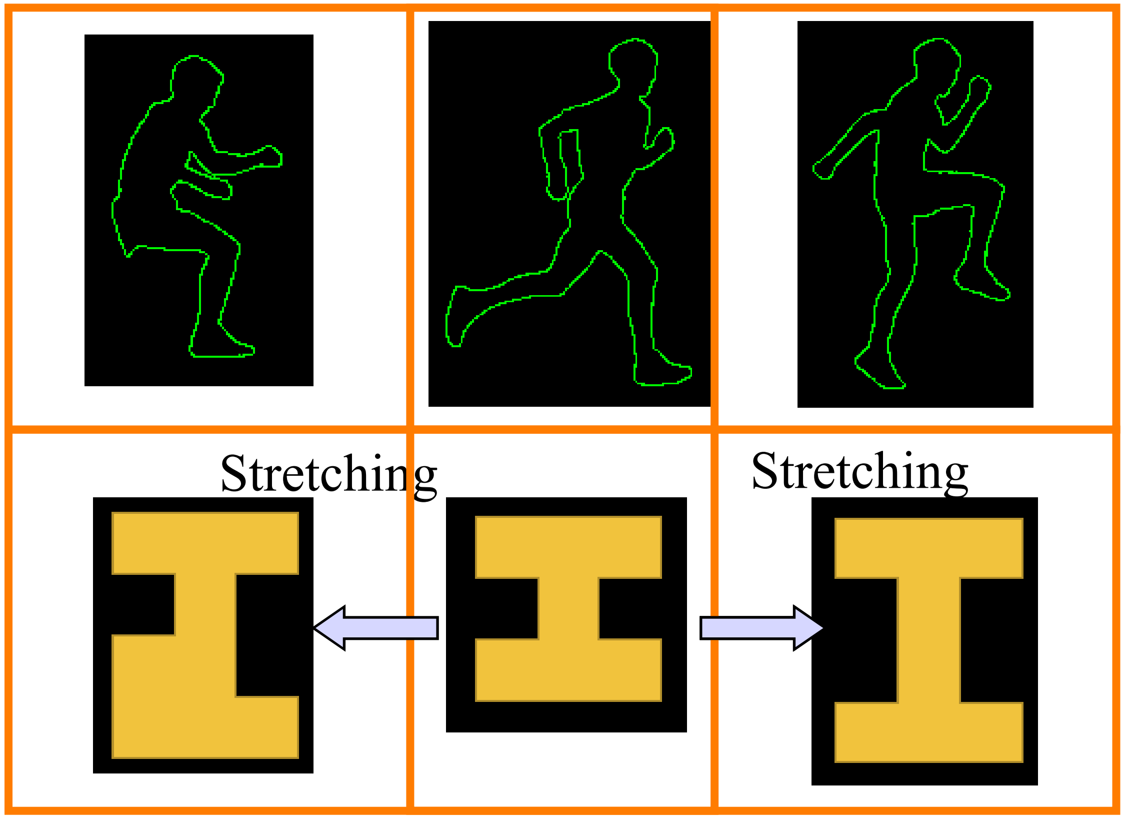
Shape context





Shape matching and similarity

Occlusion Articulation



3 tasks:

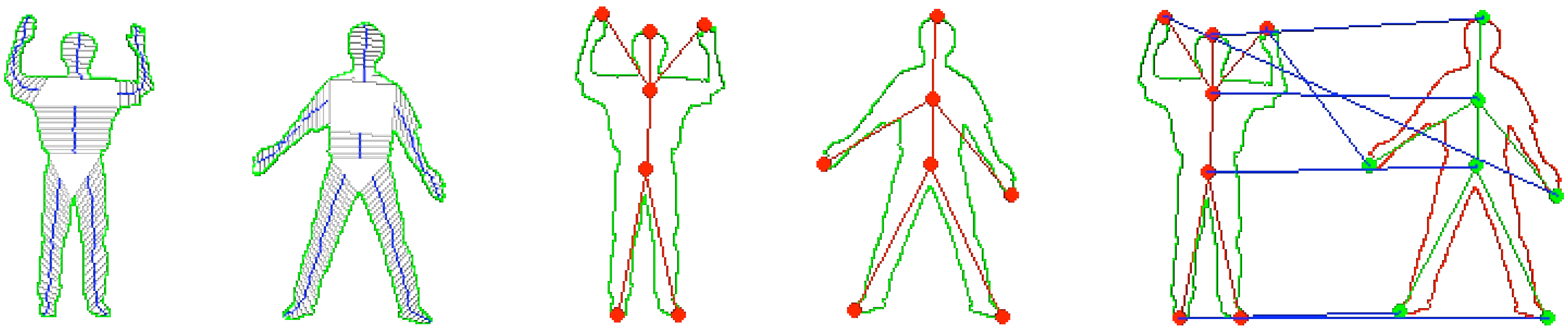
Correspondence

alignment transform

error + distortion magnitude

Shape representation

Silhouette



Edge marking:

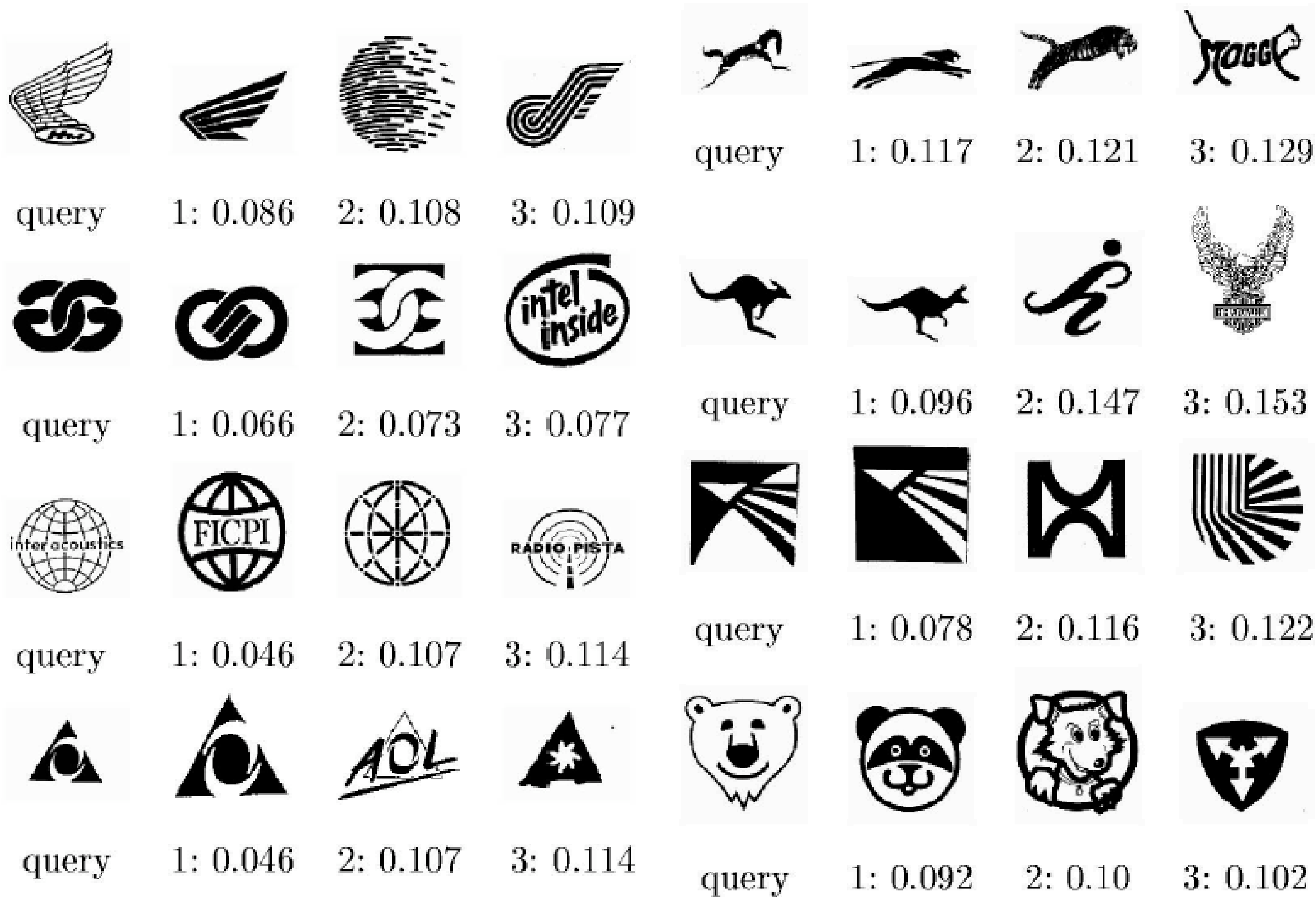
Hausdoff Distance

Distance transform

Alignment

Hash table

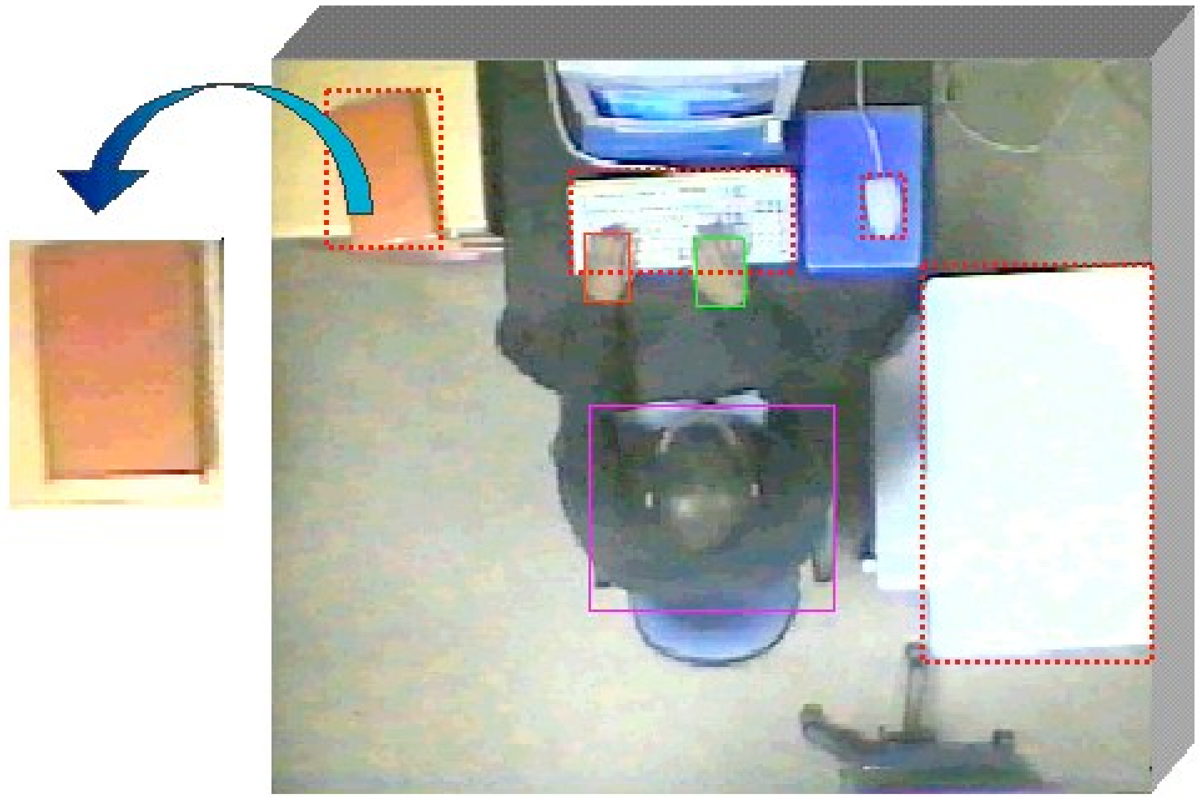
Decision tree

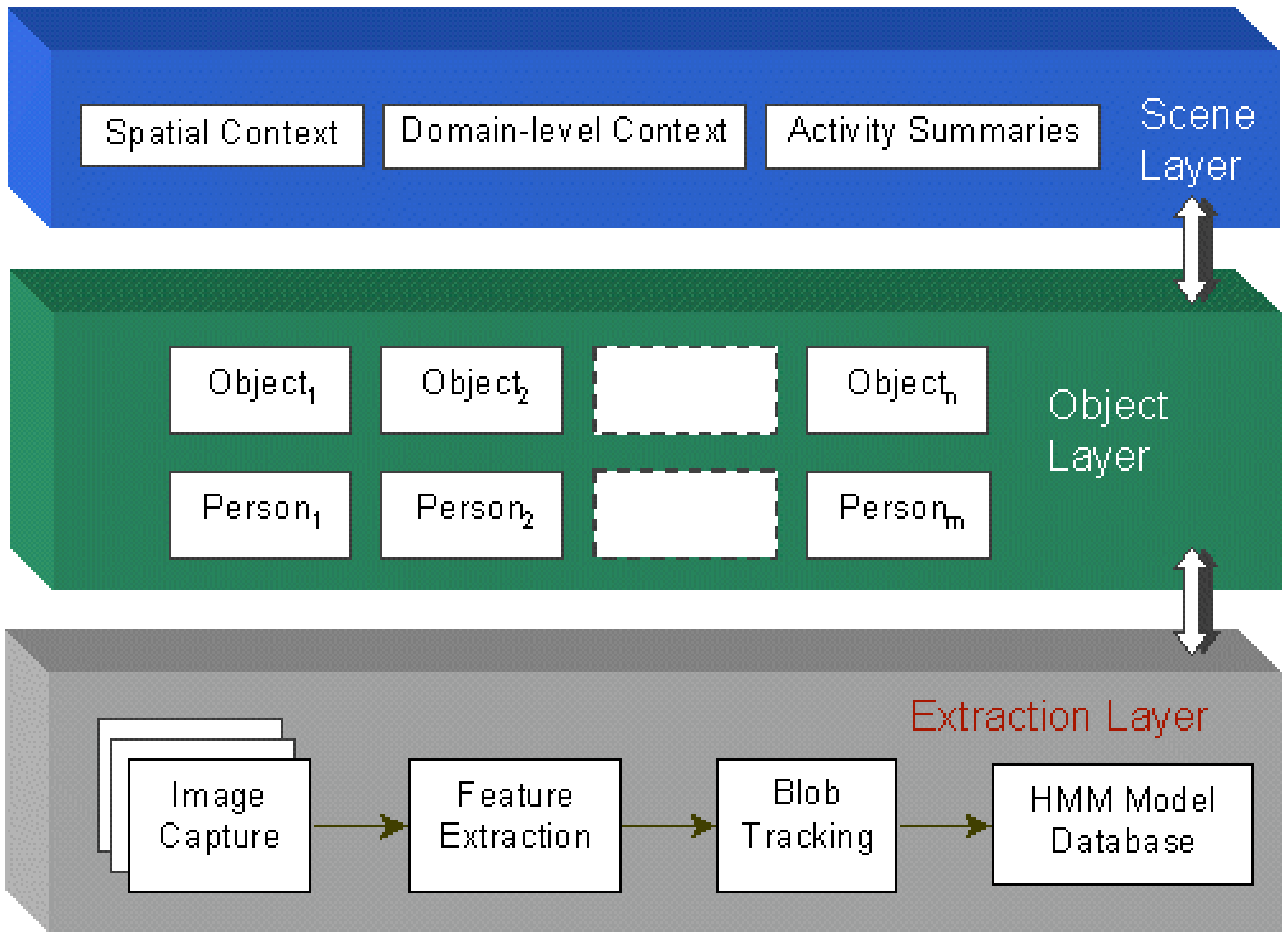


query
1: 0.086
2: 0.108
3: 0.109
query
1: 0.066
2: 0.073
3: 0.077
intel inside
query
1: 0.046
2: 0.107
3: 0.114
inter acoustics
FICPI
RADIO PISTA
query
1: 0.046
2: 0.107
3: 0.114
AOL
query
1: 0.117
2: 0.121
3: 0.129
query
1: 0.096
2: 0.147
3: 0.153
query
1: 0.078
2: 0.116
3: 0.122
query
1: 0.092
2: 0.10
3: 0.102

Activity Recognition

- Object tracking, and detection
  - blob detector/tracker, condensation tracking
- short time, single actor, action event classification
  - HMM, Dynamic models
- Extended time, multiple actor, activity
  - Context free grammar




Spatial Context
Domain-level Context
Activity Summaries
Scene Layer
Object$_1$
Object$_2$
Object$_n$
Object Layer
Person$_1$
Person$_2$
Person$_m$
Extraction Layer
Image Capture
Feature Extraction
Blob Tracking
HMM Model Database

Video->
Action

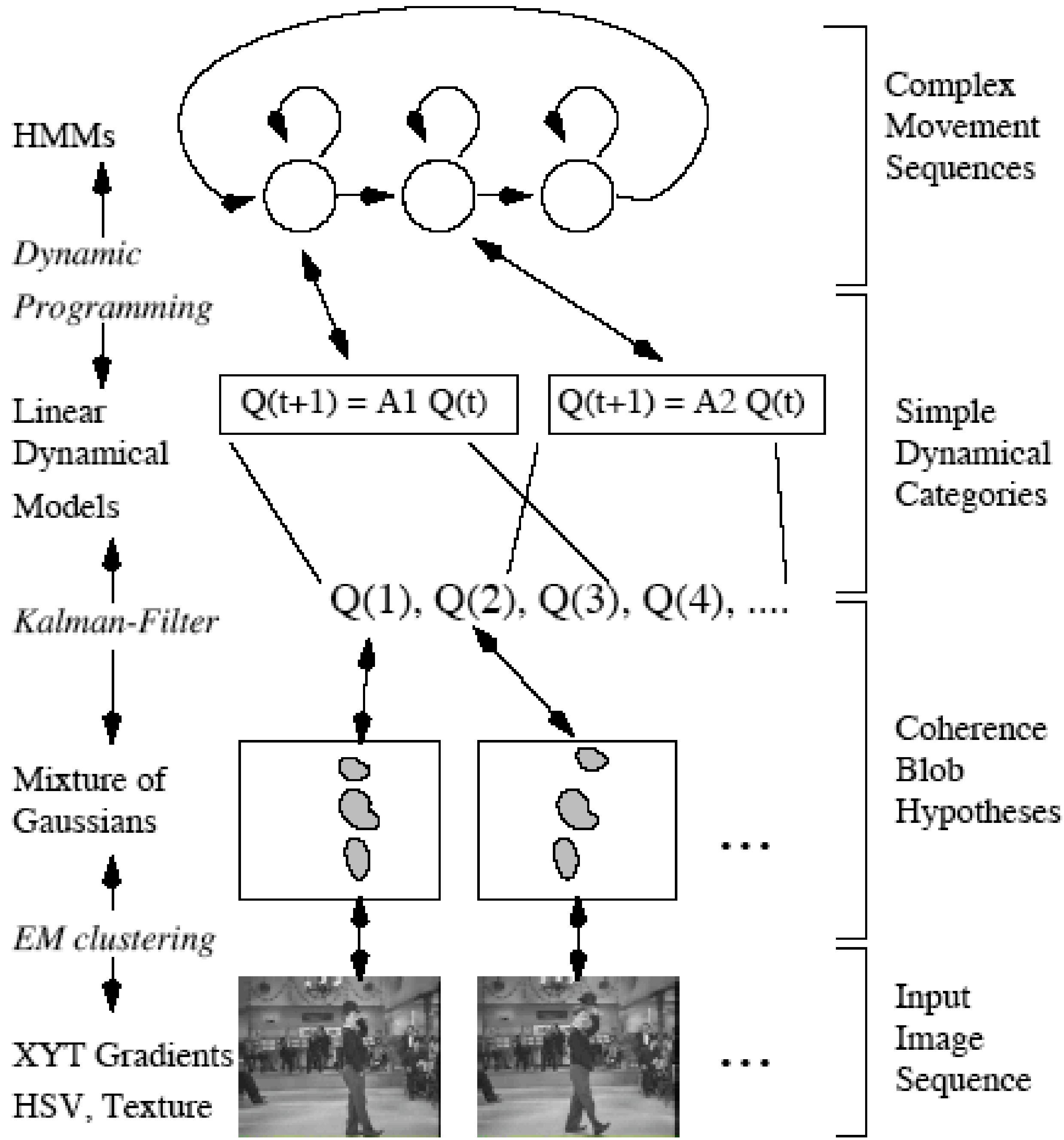
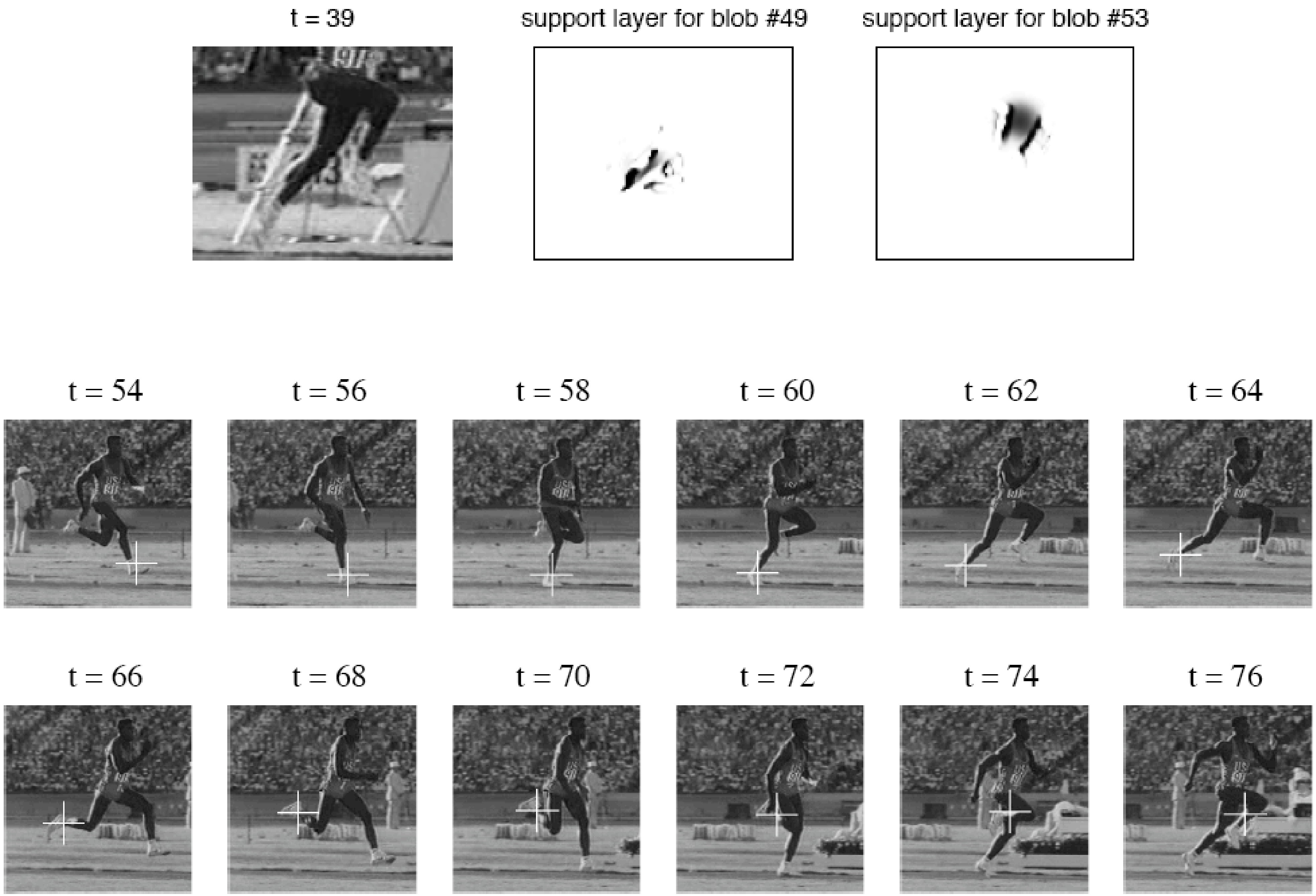
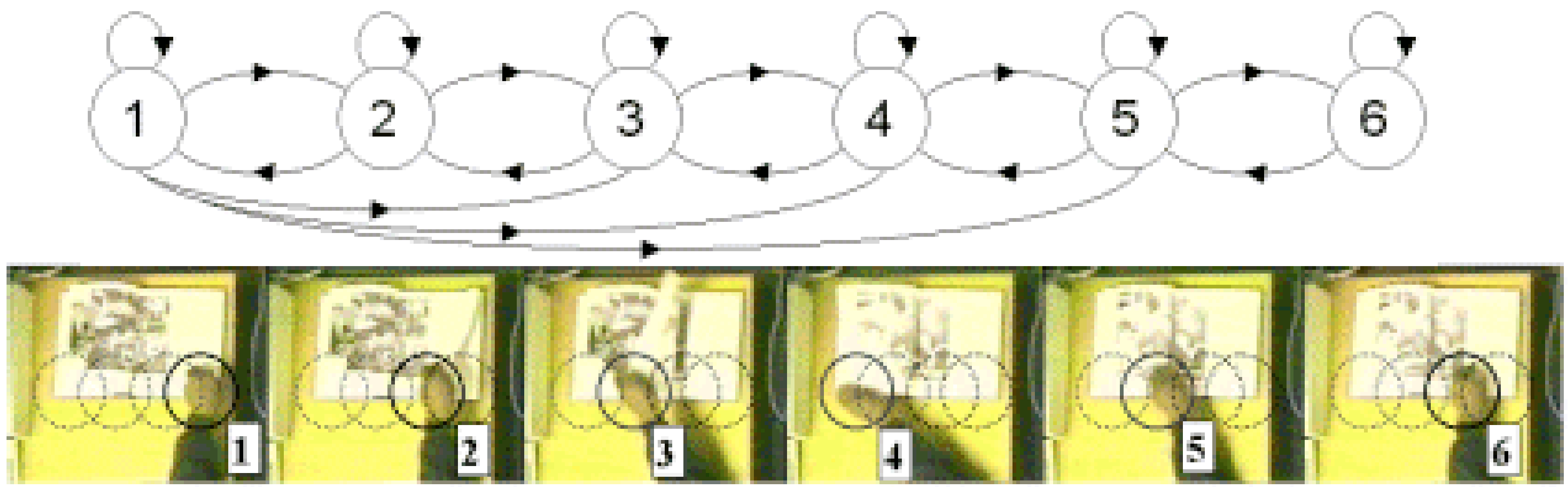


Figure 1: 4 level decomposition of human dynamics.


t = 39
support layer for blob #49
support layer for blob #53
t = 54
t = 56
t = 58
t = 60
t = 62
t = 64
t = 66
t = 68
t = 70
t = 72
t = 74
t = 76

Action Recognition

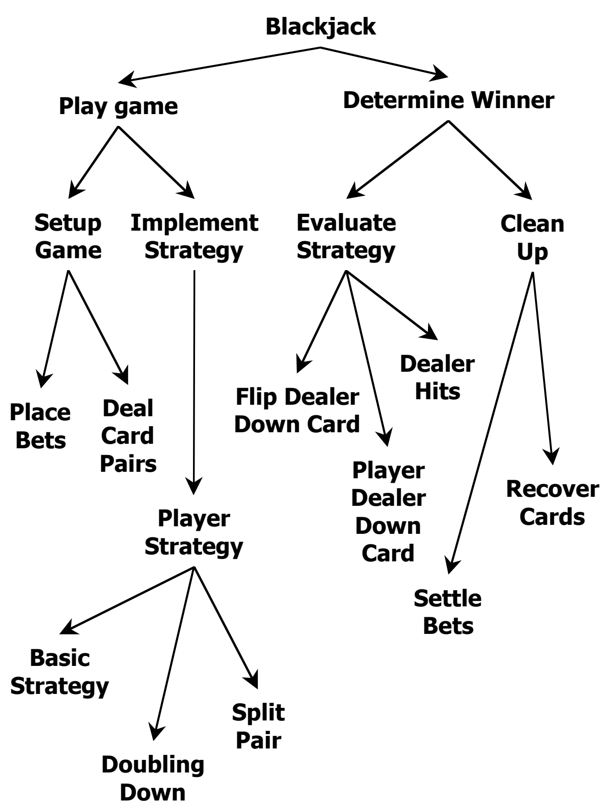


Action-> Activity

Specifying Grammar

Parsing Tree

Built from Production Rules



Production Rules taken from "rules" of Activity

Production Rules				Description
S	$\rightarrow$	AB	[1.0]	Blackjack $\rightarrow$ "play game" "determine winner"
A	$\rightarrow$	CD	[1.0]	play game $\rightarrow$ "setup game" "implement strategy"
B	$\rightarrow$	EF	[1.0]	determine winner $\rightarrow$ "evaluate strategy" "cleanup"
C	$\rightarrow$	HI	[1.0]	setup game $\rightarrow$ "place bets" "deal card pairs"
D	$\rightarrow$	GK	[1.0]	implement strategy $\rightarrow$ "player strategy"
E	$\rightarrow$	LKM	[0.6]	evaluate strategy $\rightarrow$ "flip dlr down-card" "dlr hits" "flip plyr down-card"
	$\rightarrow$	LM	[0.4]	$\rightarrow$ "flip dealer down-card" "flip player down-card"
F	$\rightarrow$	NO	[0.5]	cleanup $\rightarrow$ "settle bet" "recover card"
	$\rightarrow$	ON	[0.5]	$\rightarrow$ "recover card" "settle bet"
G	$\rightarrow$	J	[0.8]	player strategy $\rightarrow$ "Basic Strategy"
	$\rightarrow$	Hf	[0.1]	$\rightarrow$ "Splitting Pair"
	$\rightarrow$	$bfffH$	[0.1]	$\rightarrow$ "Doubling Down"
H	$\rightarrow$	l	[0.5]	place bets
	$\rightarrow$	lH	[0.5]	
I	$\rightarrow$	ffI	[0.5]	deal card pairs
	$\rightarrow$	ee	[0.5]	
J	$\rightarrow$	f	[0.8]	Basic strategy
	$\rightarrow$	fJ	[0.2]	
K	$\rightarrow$	e	[0.6]	house hits
	$\rightarrow$	eK	[0.4]	
L	$\rightarrow$	ae	[1.0]	Dealer downcard
M	$\rightarrow$	dh	[1.0]	Player downcard
N	$\rightarrow$	k	[0.16]	settle bet
	$\rightarrow$	kN	[0.16]	
	$\rightarrow$	j	[0.16]	
	$\rightarrow$	jN	[0.16]	
	$\rightarrow$	i	[0.18]	
	$\rightarrow$	iN	[0.18]	
O	$\rightarrow$	a	[0.25]	recover card
	$\rightarrow$	aO	[0.25]	
	$\rightarrow$	b	[0.25]	
	$\rightarrow$	bO	[0.25]	

Symbol	Domain-Specific Events
a	dealer removed card from house
b	dealer removed card from player
c	player removed card from house
d	player removed card from player
e	dealer added card to house
f	dealer dealt card to player
g	player added card to house
h	player added card to player
i	dealer removed chip
j	player removed chip
k	dealer pays player chip
l	player bets chip

Production Rules				Description
S	$\rightarrow$	AB	[1.0]	Blackjack $\rightarrow$ “play game” “determine winner”
A	$\rightarrow$	CD	[1.0]	play game $\rightarrow$ “setup game” “implement strategy”
B	$\rightarrow$	EF	[1.0]	determine winner $\rightarrow$ “evaluate strategy” “cleanup”
C	$\rightarrow$	HI	[1.0]	setup game $\rightarrow$ “place bets” “deal card pairs”
D	$\rightarrow$	GK	[1.0]	implement strategy $\rightarrow$ “player strategy”
E	$\rightarrow$	LKM	[0.6]	evaluate strategy $\rightarrow$ “flip dlr down-card” “dlr hits” “flip plyr down-car
	$\rightarrow$	LM	[0.4]	evaluate strategy $\rightarrow$ “flip dealer down-card” “flip player down-card”
F	$\rightarrow$	NO	[0.5]	cleanup $\rightarrow$ “settle bet” “recover card”
	$\rightarrow$	ON	[0.5]	$\rightarrow$ “recover card” “settle bet”
G	$\rightarrow$	J	[0.8]	player strategy $\rightarrow$ “Basic Strategy”
	$\rightarrow$	Hf	[0.1]	$\rightarrow$ “Splitting Pair”
	$\rightarrow$	$bfffH$	[0.1]	$\rightarrow$ “Doubling Down”
H	$\rightarrow$	l	[0.5]	place bets
	$\rightarrow$	lH	[0.5]	
I	$\rightarrow$	ffI	[0.5]	deal card pairs
	$\rightarrow$	ee	[0.5]	
J	$\rightarrow$	f	[0.8]	Basic strategy
	$\rightarrow$	fJ	[0.2]	
K	$\rightarrow$	e	[0.6]	house hits
	$\rightarrow$	eK	[0.4]	
L	$\rightarrow$	ae	[1.0]	Dealer downcard
M	$\rightarrow$	dh	[1.0]	Player downcard
N	$\rightarrow$	k	[0.16]	settle bet
	$\rightarrow$	kN	[0.16]	
	$\rightarrow$	j	[0.16]	
	$\rightarrow$	jN	[0.16]	
	$\rightarrow$	i	[0.18]	
	$\rightarrow$	iN	[0.18]	
O	$\rightarrow$	a	[0.25]	recover card
	$\rightarrow$	aO	[0.25]	
	$\rightarrow$	b	[0.25]	
	$\rightarrow$	bO	[0.25]	

Symbol	Domain-Specific Events
a	dealer removed card from house
b	dealer removed card from player
c	player removed card from house
d	player removed card from player
e	dealer added card to house
f	dealer dealt card to player
g	player added card to house
h	player added card to player
i	dealer removed chip
j	player removed chip
k	dealer pays player chip
l	player bets chip

Table 6.3: SCFG G_{21} for Blackjack/“21” card game: Production rules, probabilities, and descriptions. Detectable domain-specific events make up the terminal alphabet V_T of G_{21}.

Topics

Texture
synthesis/analysis

Fixed body object
detection/recognition

Flexible body object
detection/recognition

Image segmentation
Image translation

Image Shape modeling

Human activity recognition