Image Parsing

Alan Yuille (UCLA)

Based on Tu, Chen, Yuille & Zhu

(ICCV 2003 – IJCV 2005).

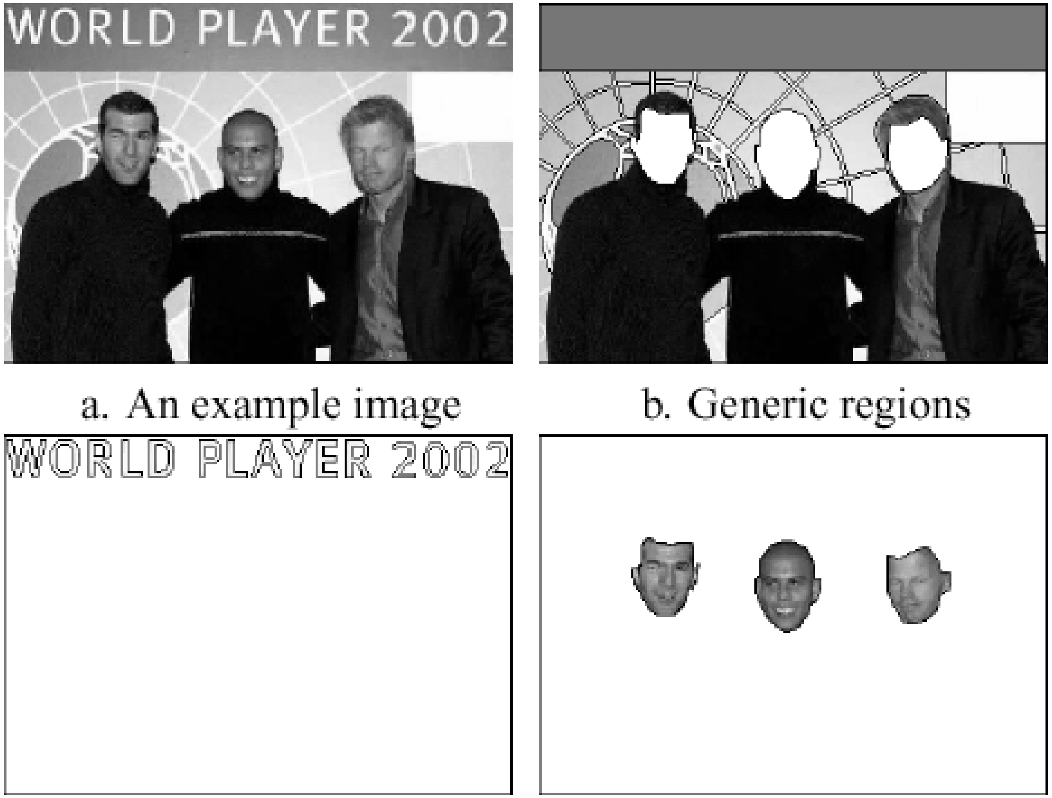
Image Parsing

- Parse Images into their constituent patterns – shaded regions, textures, and objects (faces & text).
- Detection and recognition of objects competes and cooperates with segmentation.
- Inference Algorithm: Bottom-up cues activate top-down generative models.

Parsing Example

Input Image (left)
Region Layer (right)

Text Object Layer (left)
Face Object Layer (right)



a. An example image

b. Generic regions

c. Text

d. Faces

Image Segmentation

- Goal: Decompose image domain R.

$\{R_i : i = 1, .., N\}$ s.t. $\cup_{i=1}^{N} R_i = R,$

$R_i \cap R_j = \emptyset \ \forall i \neq j \ \ \Gamma_i = \partial R_i$

- Cost Function Minimization (log prob):
- (Zhu & Yuille 1996)

$$E[\Gamma, \{\theta_i\}, N]$$

$$= \sum_{i=1}^{N} \{\frac{\mu}{2} \int_{\Gamma_i} ds - \log P(\{I(x) : x \in R_i\} | \theta_i) + \lambda\}$$

- Generative models of patterns $P(I(x)|\theta)$

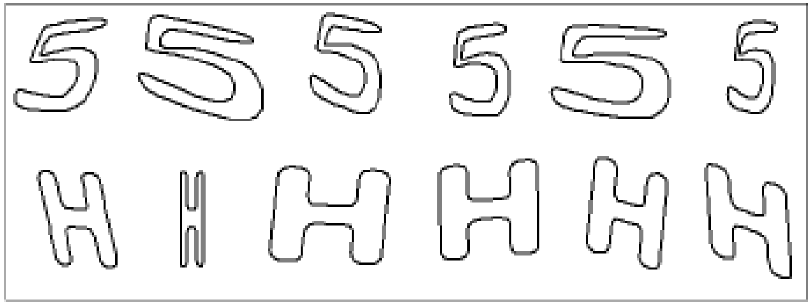
Generative Models: Objects

- Define Models P(I|Text), P(I|Face).
- Represent Text by deformable templates TP(i):i=1,...,62.
- Each template is represented geometrically by quadratic B-splines.
- Represent the image intensity of the Text by smooth shading

$$J(x,y;\theta) = ax^2 + bxy + cy^2 + dx + ey + f.$$

Generative Models: Text

- Text is described by L=(c,S,M), where c is template index, S control point positions, M is affine transformation.
- P(L)=P(c)P(S|c)P(M)
- Samples:

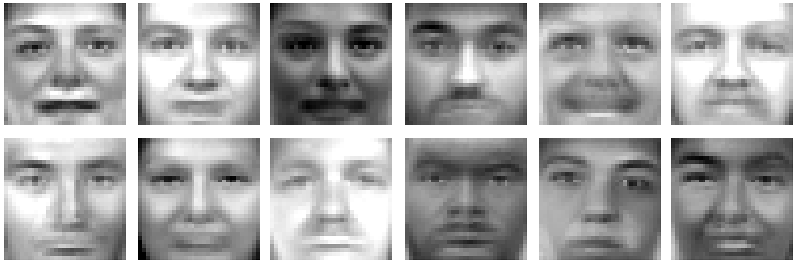


Generative Models: Faces.

- Principal Component Analysis of faces in the FERET Database.

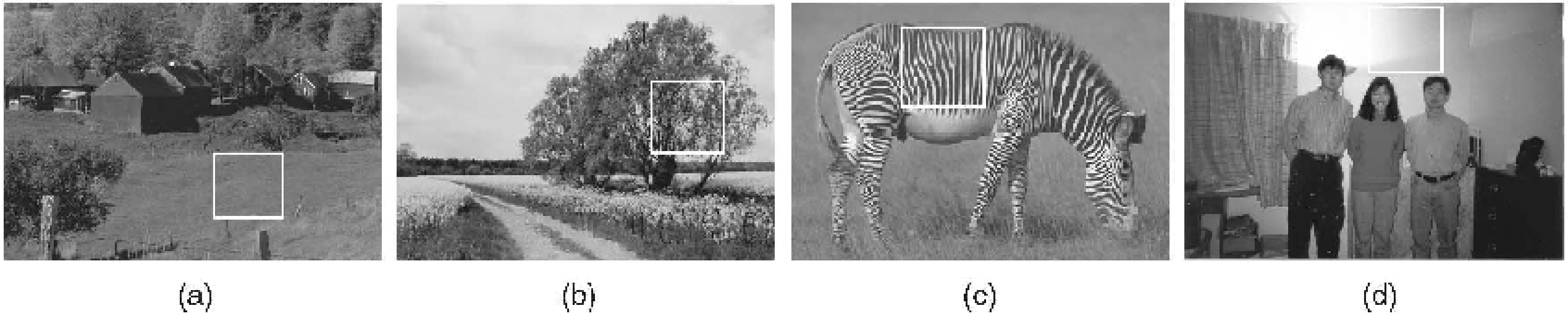
$$I(x) = \sum_{i=1}^{N} \alpha_i B_i(x)$$

- B(x) are principal components. Prior P(alpha) learnt from the dataset.
- Samples:



Generative Models: Generic.

- Classify image regions as:
- 1. Grey Image Gaussian (i.i.d).
- 2. Grey Image Clutter. (Histogram).
- 3. Grey Image Texture (FRAME model)
- 4. Grey Image Shading. (Bezier surface)



Generic Models:

Gaussian i.i.d.$P(\mathbf{I}_R|\theta) = \prod_{\nu\in R} \frac{1}{\sqrt{2\pi\sigma}} e^{-(I_\nu-\mu)/(2\sigma^2)}$,

Histogram$P(\mathbf{I}_R|\theta) = \prod_{\nu\in R} h(I_\nu)$,

Texture$P(\mathbf{I}_R|\theta) = \prod_{\nu\in R} P(I_\nu|I_{\delta\nu}) = \prod_{\nu\in R} \frac{1}{Z_\nu} e^{-<\theta,h(I_\nu|I_{\partial\nu})>}$,

GlobalShading$P(\mathbf{I}_R|\theta) = \prod_{\nu\in R} \frac{1}{\sqrt{2\pi\sigma}} e^{-(I_\nu-B_\nu)^2/(2\sigma^2)}$

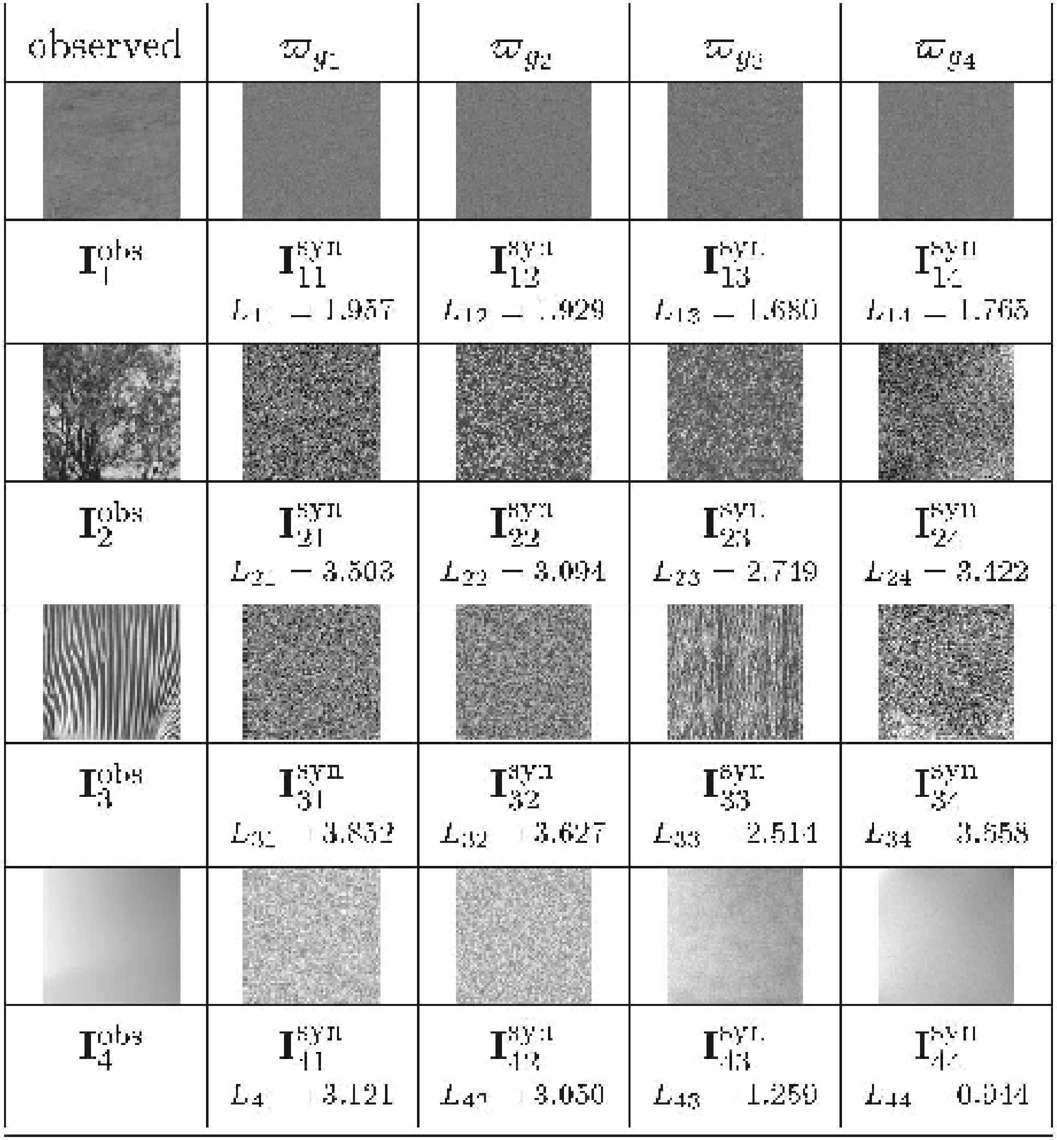
- h(.) denotes histogram.
- B(x,y) is a Bezier spline:

$$B(x,y) = ((1-x)^3, 3x(1-x)^2, 3x^2(1-x), x^3)^T \Sigma^{-1} ((1-y)^3, 3y(1-y)^2, 3y^2(1-y), y^3)$$

Samples

1st Column: Images

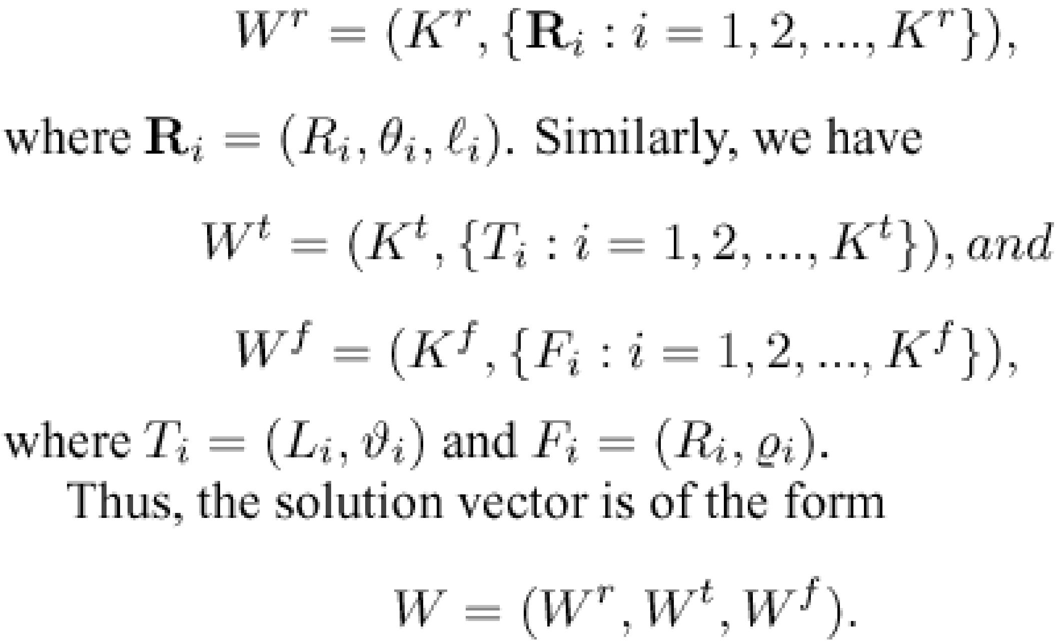
2-5 Columns: Images synthesized from the four families (after best parameter fit).



observed	ϖ_{g_1}	ϖ_{g_2}	ϖ_{g_3}	ϖ_{g_4}
$\mathbf{I}^{\mathrm{obs}}_1$	$\mathbf{I}^{\mathrm{syn}}_{11}$ $L_{11} = 1.957$	$\mathbf{I}^{\mathrm{syn}}_{12}$ $L_{12} = 1.929$	$\mathbf{I}^{\mathrm{syn}}_{13}$ $L_{13} = 1.680$	$\mathbf{I}^{\mathrm{syn}}_{14}$ $L_{14} = 1.765$
$\mathbf{I}^{\mathrm{obs}}_2$	$\mathbf{I}^{\mathrm{syn}}_{21}$ $L_{21} = 3.503$	$\mathbf{I}^{\mathrm{syn}}_{22}$ $L_{22} = 3.094$	$\mathbf{I}^{\mathrm{syn}}_{23}$ $L_{23} = 2.719$	$\mathbf{I}^{\mathrm{syn}}_{24}$ $L_{24} = 3.422$
$\mathbf{I}^{\mathrm{obs}}_3$	$\mathbf{I}^{\mathrm{syn}}_{31}$ $L_{31} = 3.852$	$\mathbf{I}^{\mathrm{syn}}_{32}$ $L_{32} = 3.627$	$\mathbf{I}^{\mathrm{syn}}_{33}$ $L_{33} = 2.514$	$\mathbf{I}^{\mathrm{syn}}_{34}$ $L_{34} = 3.558$
$\mathbf{I}^{\mathrm{obs}}_4$	$\mathbf{I}^{\mathrm{syn}}_{41}$ $L_{41} = 3.121$	$\mathbf{I}^{\mathrm{syn}}_{42}$ $L_{42} = 3.050$	$\mathbf{I}^{\mathrm{syn}}_{43}$ $L_{43} = 1.259$	$\mathbf{I}^{\mathrm{syn}}_{44}$ $L_{44} = 0.914$

Scene Representation

- Representation W has region types l -- generic (r), faces (f), & text (t).



$$W^r = (K^r, \{\mathbf{R}_i : i = 1, 2, ..., K^r\}),$$

where $\mathbf{R}_i = (R_i, \theta_i, \ell_i)$. Similarly, we have

$$W^t = (K^t, \{T_i : i = 1, 2, ..., K^t\}), and$$

$$W^f = (K^f, \{F_i : i = 1, 2, ..., K^f\}),$$

where $T_i = (L_i, \vartheta_i)$ and $F_i = (R_i, \varrho_i)$.
Thus, the solution vector is of the form

$$W = (W^r, W^t, W^f).$$

Likelihood Function.

- The likelihood function is written as:

$$p(\mathbf{I}|W) = \prod_{i}^{K^r} p(\mathbf{I}_{R_i}; \theta_i, \ell_i) \prod_{i}^{K^t} p(\mathbf{I}_{L_i}; \vartheta_i) \prod_{i}^{K^f} p(\mathbf{I}_{R_i}; \varrho_i).$$

- Variable l_i labels the generic region (e.g. texture, shading, clutter).
- Variable θ_i labels model parameters.

Prior and Posterior

- The prior is given by:

$$p(W) = \Big(p(K^r)\prod_{i=1}^{K^r} p(\mathbf{R}_i)\Big)\Big(p(K^t)\prod_{i=1}^{K^t} p(T_i)\Big)\Big(p(K^f)\prod_{i=1}^{K^f} p(F_i)\Big).$$

- where $p(R_i) \propto exp\{-\gamma Area(R_i)^{0.9} - \lambda|\partial R_i|\}.$
- Similar area prior for faces & text.
- Additional prior for face & text parameters.

MAP Estimation by DDMCMC

- Formalize Inference as MAP estimation:

$$W^* = \arg\max_{W \in \Omega} p(W|\mathbf{I}) = \arg\max_{W \in \Omega} p(\mathbf{I}|W)p(W).$$

- Inference by Data-Driven Markov Chain Monte Carlo (Tu & Zhu 2002).
- Propose move with prob: $q(W \rightarrow W'|\mathbf{I})$
- Accept move with probability

$$\alpha(W \rightarrow W') = \min(1, \frac{p(W'|\mathbf{I})}{p(W|\mathbf{I})} \cdot \frac{q(W' \rightarrow W|\mathbf{I})}{q(W \rightarrow W'|\mathbf{I})}).$$

DDMCMC 0

- Intuition: The data driven proposal encourage search in the correct parts of the solution space. (not uniform/Gaussian).
- Hypothesis and Verification.
- If proposal is q(W’|I), then at zero temperature proposal is accepted if

$$\frac{P(W'|I)}{P(W|I)} \geq \frac{q(W'|I)}{q(W|I)}$$

- Only need to evaluate log-likelihood ratios.

DDMCMC 1.

- DDMCMC is guaranteed to converge to samples from posterior $W \sim P(W|\mathbf{I})$
- For rapid convergence, we need good proposals which are fast to compute.
- Best proposal is $q(W \rightarrow W'|\mathbf{I}) = P(W'|\mathbf{I})$ but this is impractical.
- Instead, factorized models (Talk I), AdaBoost (Talk II).

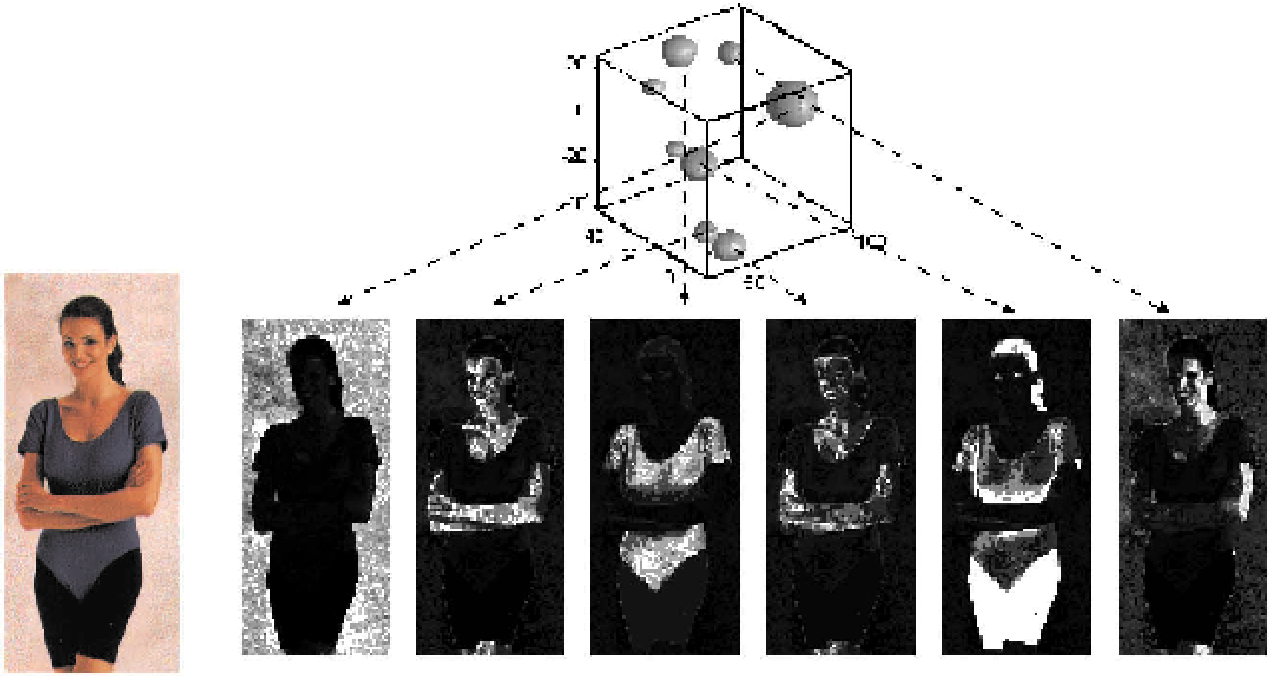
DDMCMC II:

- Proposals:
- Region boundaries are likely at edges.

$$P(x \in \{\Gamma_i : i = 1, ..., N\}) \sim P(f(I(x))|on)$$
$$P(x \notin \{\Gamma_i : i = 1, ..., N\}) \sim P(f(I(x))|off).$$

- Clustering to get region parameters θ
- Objects: Faces and Text – AdaBoost – conditional distributions(?) P(face|I) P(non-face|I)

Clustering for Colour.

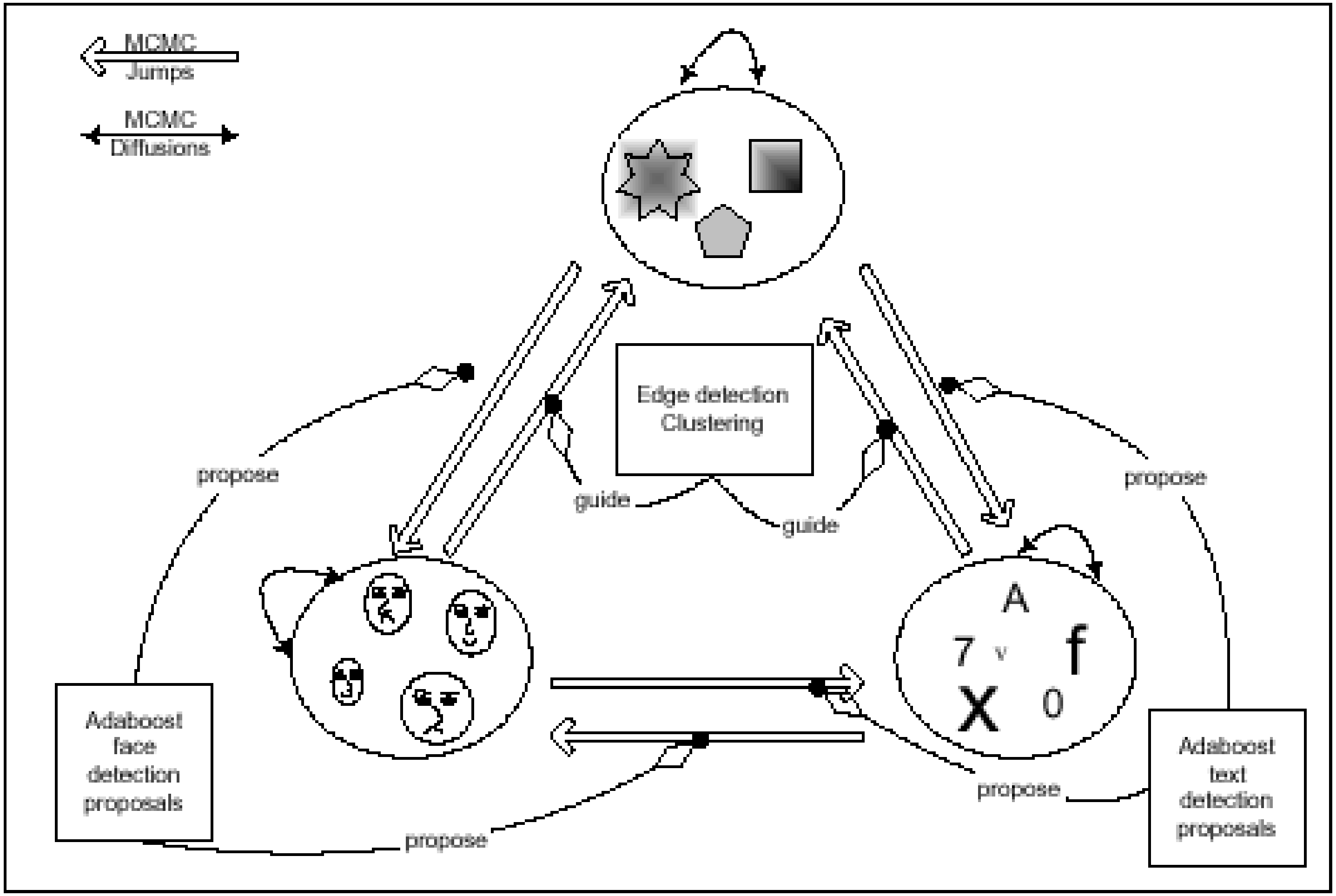


Input **I** Color clusters and their saliency maps $S_i^{c1}, i = 1, ..., 6$

Two Types of Moves.

- Discrete Moves: jumps -- create, merge, split region.
- Continuous Moves: diffusions -- move boundary, estimate model parameters.
- Low-level proposes moves that are evaluated by top-down generative models.
- Feedforward and Feedback in the Brain?

Bottom-Up & Top-Down


MCMC Jumps
MCMC Diffusions
Edge detection Clustering
guide
guide
propose
propose
propose
propose
A
7
v
f
X
0
Adaboost face detection proposals
Adaboost text detection proposals

Examples of Moves:

- Stochastic Region Competition moves the boundaries:

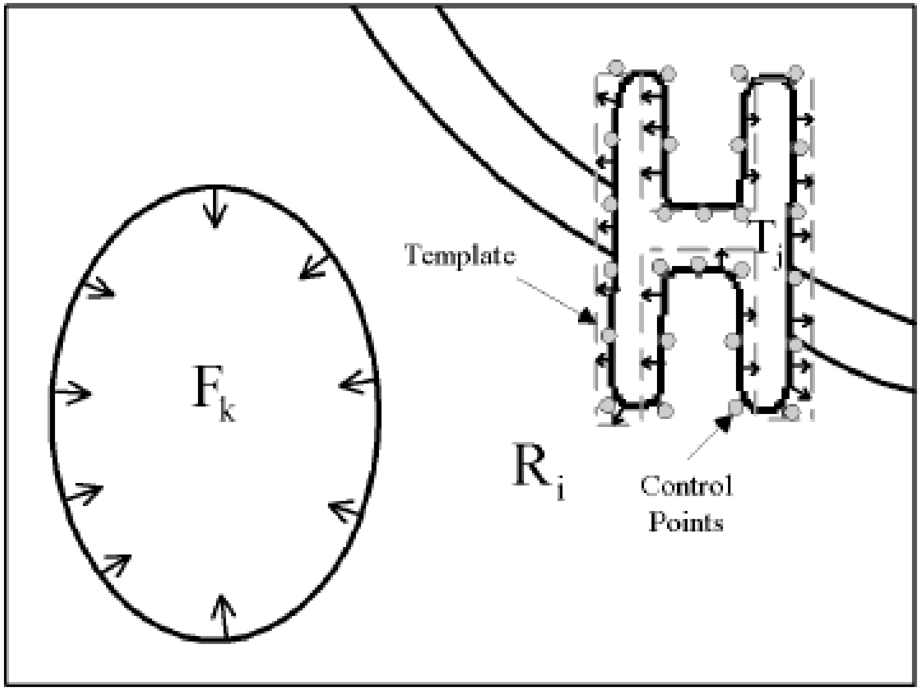
$$\frac{d\Gamma_{ij}(s)}{dt} = -\frac{\delta \log P(W|I)}{\partial \Gamma_{ij}(s)} + \sqrt{2T(t)}dw_t\vec{n}(s)$$

- reduces to:

$$\frac{d\Gamma_{ij}(s)}{dt} =$$
$$+ (\log \frac{p(I_{R_i};\theta_i)}{p(I_{R_j};\theta_j)} + c \cdot \kappa(s)) + \sqrt{2T(t)}dw_t\vec{n}(s)$$

Diffusion of Boundaries.

Diffusion of boundaries is driven by competition for boundary pixels by adjacent regions.



Example Move: Split Region:

- Jump dynamics splits a region

R_k into R_i, R_j $R_i \cup R_j = R_k,\ R_i \cap R_j = \emptyset$

- Generate proposal by:

$$G(W \mapsto W') = q_{split} q(R_k) q(\Gamma_{ij}|R_k) q(l_i) q(\theta_i|R_i, l_i) q(l_j) q(\theta_j|R_j, l_j)$$

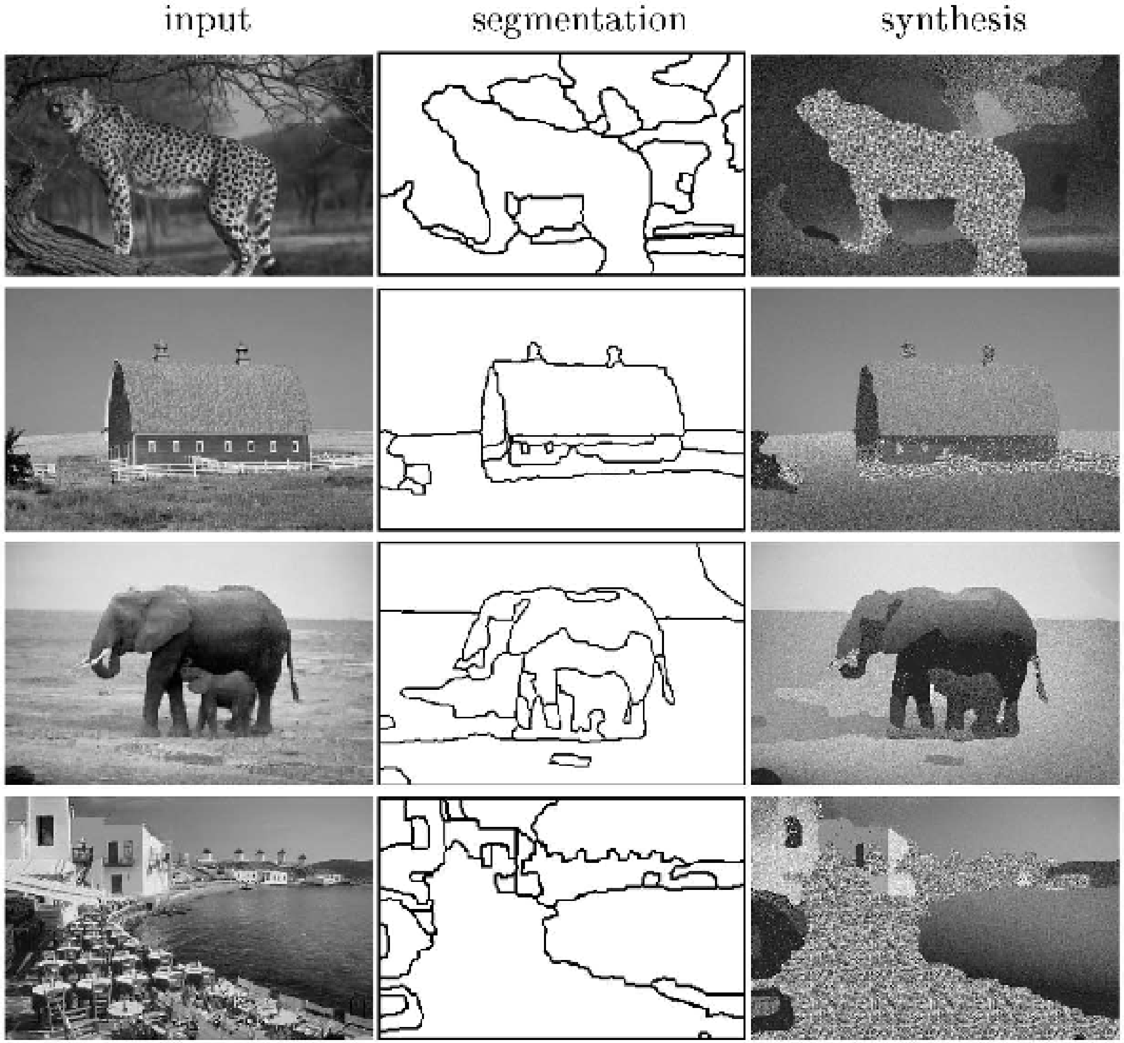
- This is chosen to approximate:

$$P(W'|W) \propto \frac{P(R_I|\theta_i)P(R_j|\theta_j)P(\Gamma_{ij})}{P(R_k|\theta_k)}$$

Results without faces & text.

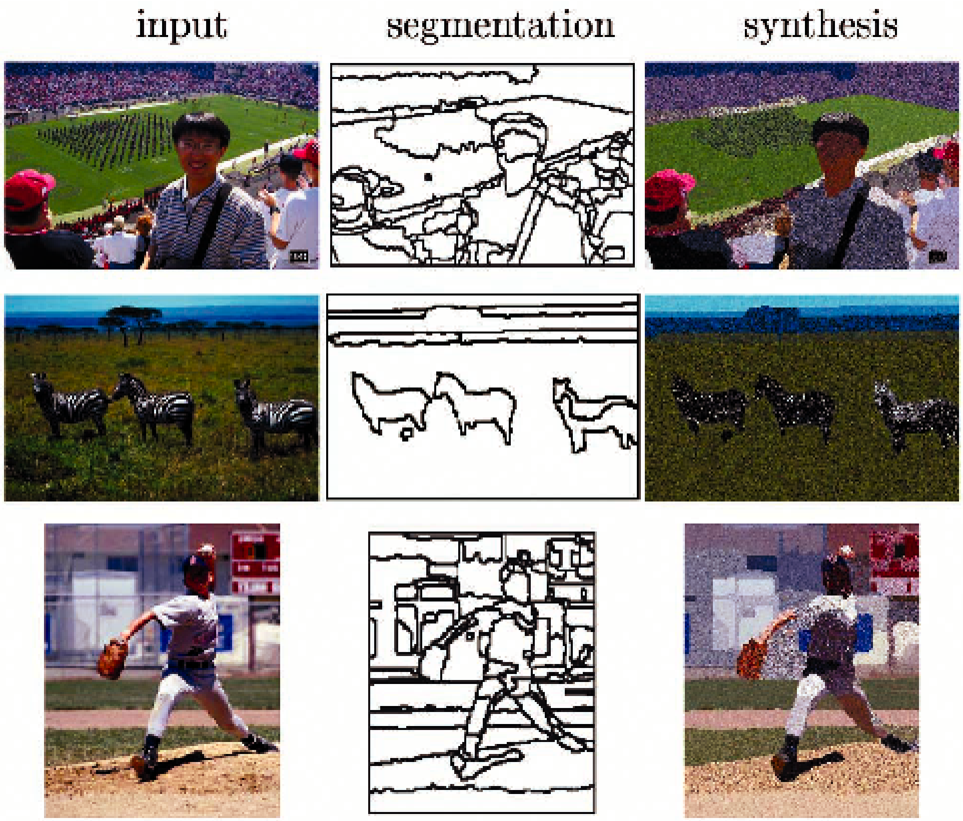
Input, Segmentation, and Synthesis from P(I|W*).

Tu & Zhu 2002.



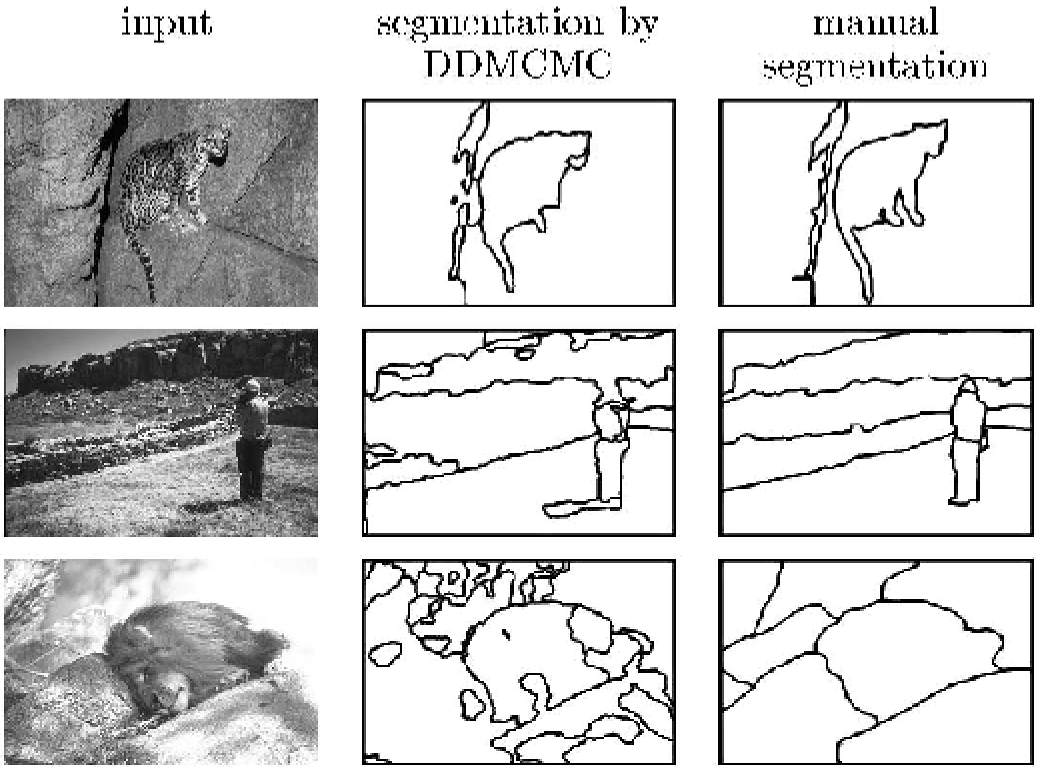
Colour without faces & text.

Faces and text are only segmented approximately.
(Tu & Zhu 2002).



Test on Berkeley Data.

Examples of results when algorithm run on Berkeley dataset.



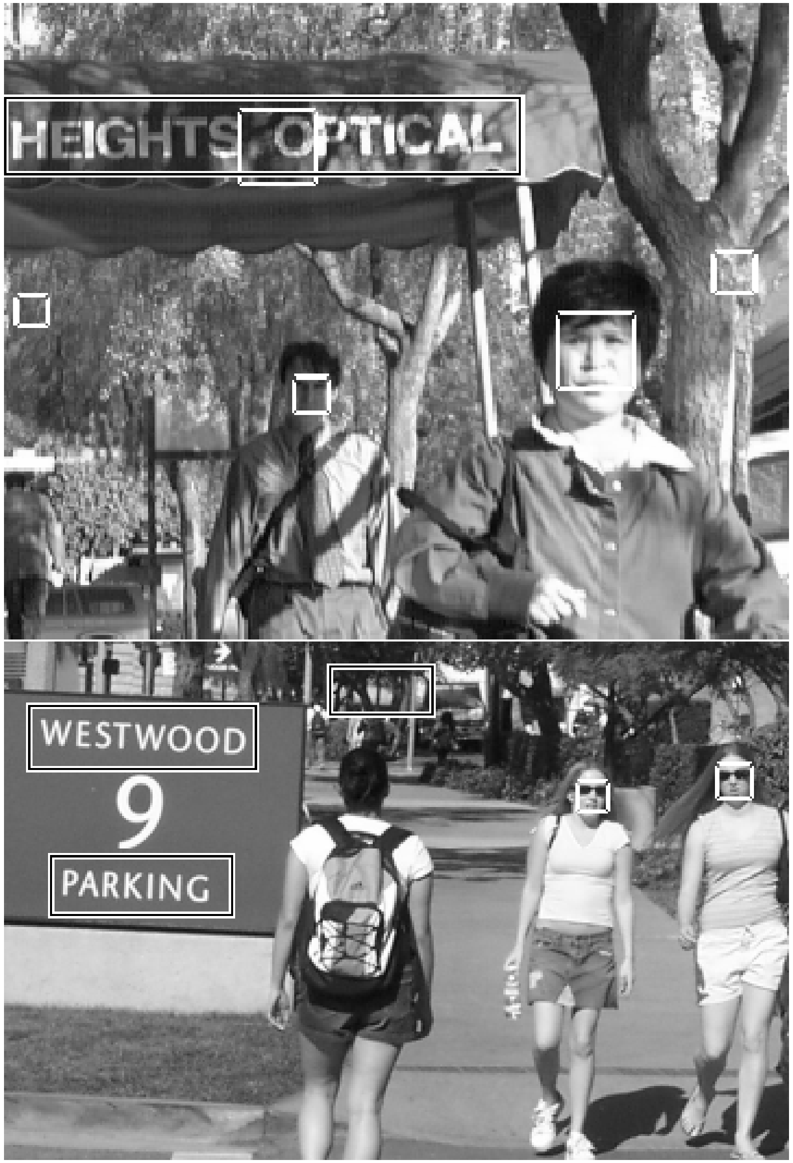
AdaBoost test.

Boxes show faces & text detected by AdaBoost at fixed threshold.

Impossible to pick a threshold that gives no false positives/negatives on these two images.

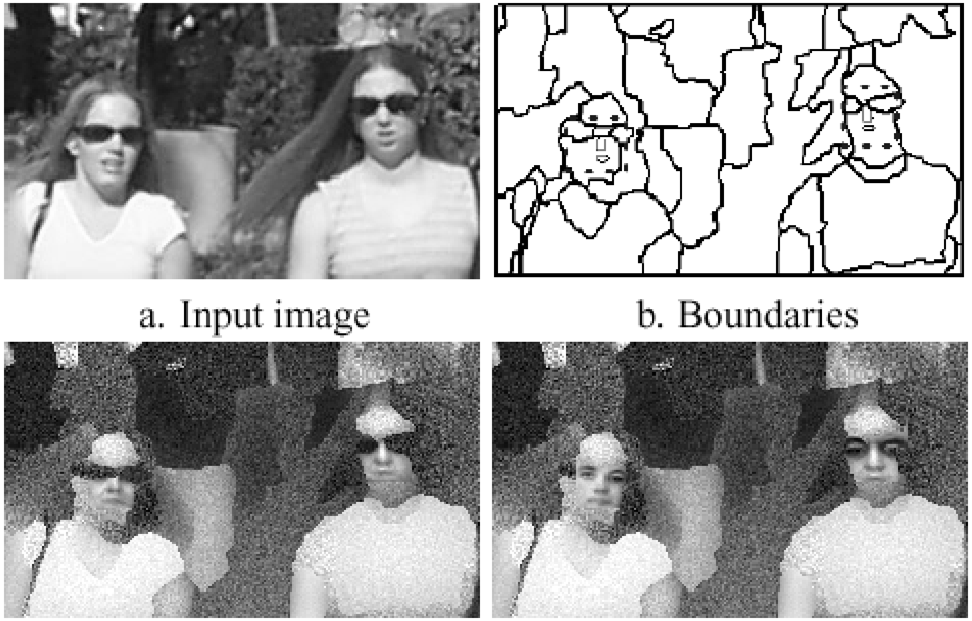
Boxes show high probability (?) proposals for faces & text.

(AdaBoost algorithm from 2nd day talk.)



Parking Image: cooperation/explain away

Generic “shaded region” processes detect the dark glasses, so the face model doesn’t need to “explain” that part of the data.



a. Input image

b. Boundaries

c. Synthesis 1

d. Synthesis 2

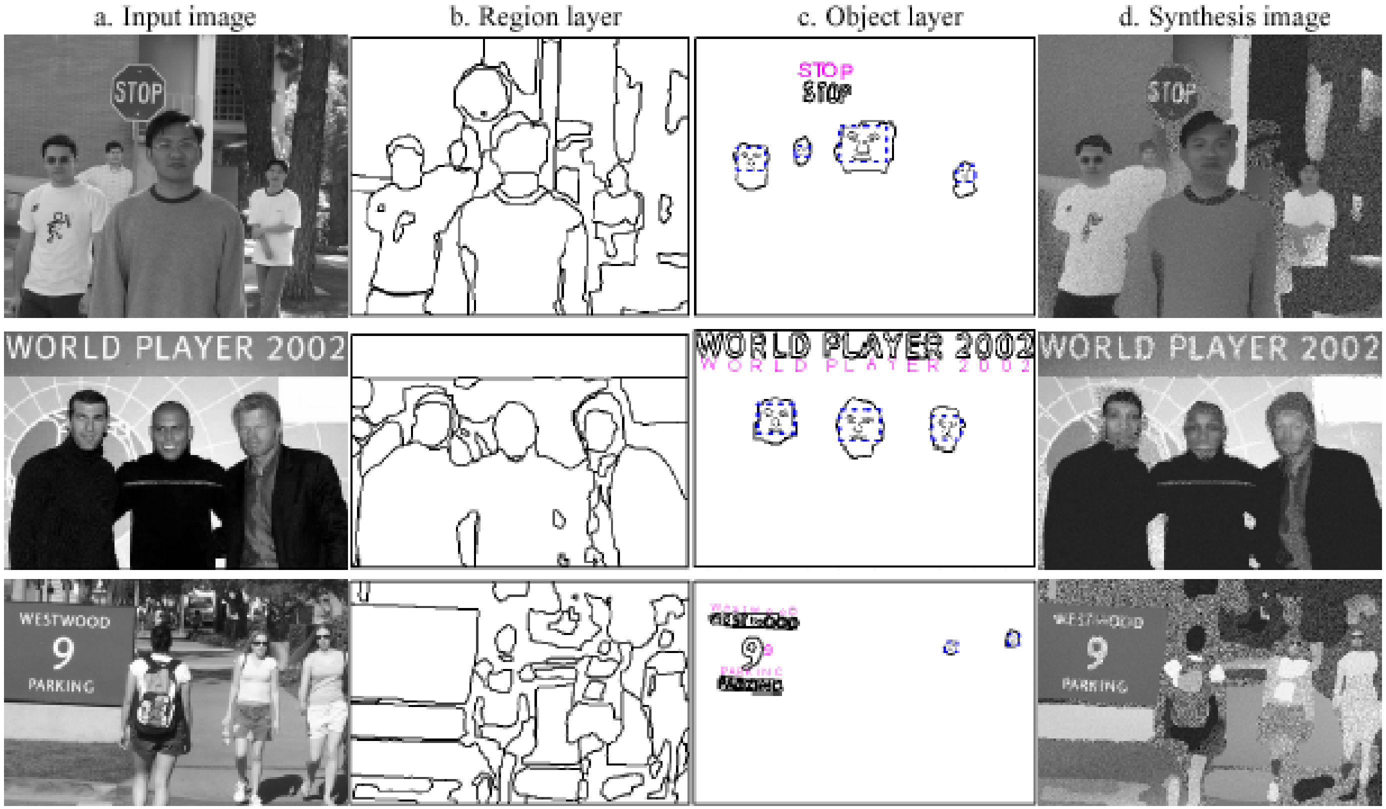
AdaBoost requires low threshold to detect these faces. (Cause false positives elsewhere.)

Stop Sign, Soccer, Parking.

Stop Sign.
Multiple scales.

Soccer Image.

Parking Image.
Glasses/Shaded.
9 detected as a generic region.
(cooperative).



a. Input image

b. Region layer

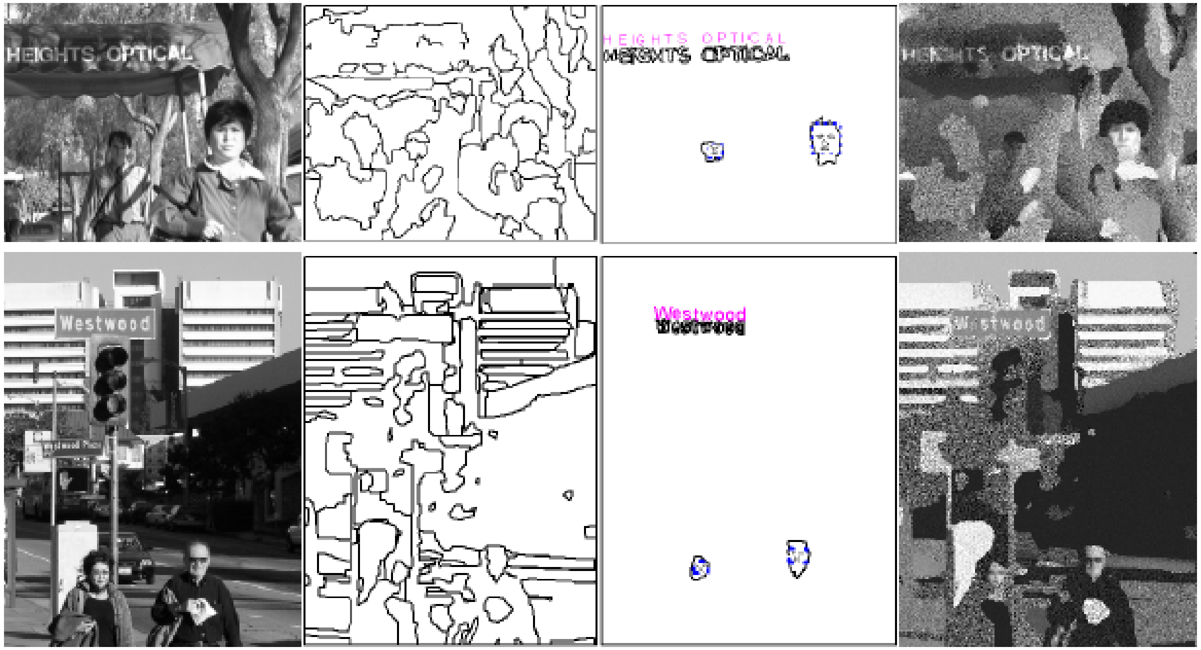
c. Object layer

d. Synthesis image

Street & Westwood.

Street: Face model is used to reject fake AdaBoost candidates. Cooperativity – shadows on text explained as shaded regions.

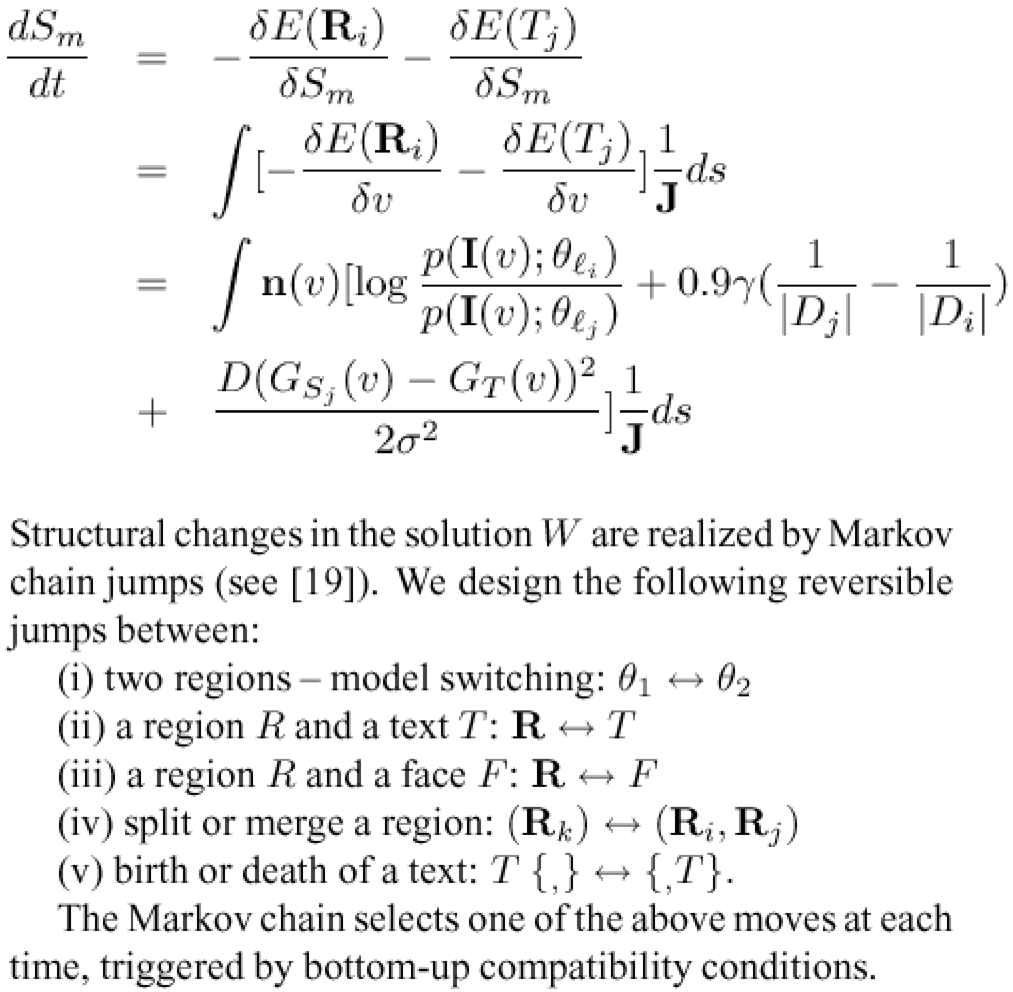
Westwood: shaded region models needed to explain away glasses.



Conclusion

- First Attempt at Image Parsing (Tu, Chen, Yuille, Zhu ICCV 2003. Nice, France).
- Cooperative/Competitive – explaining away.
- Is DDMCMC the best way to combine bottom-up and top-down (generative).
- When is top-down really needed?
- Better generative models (& multiscale).
- Scaling up to more objects –compositionality?
- *Thanks to Alain and Laurent for organization*.

Maths 3.



$$\begin{aligned}
\frac{dS_m}{dt} &= -\frac{\delta E(\mathbf{R}_i)}{\delta S_m} - \frac{\delta E(T_j)}{\delta S_m} \\
&= \int [-\frac{\delta E(\mathbf{R}_i)}{\delta v} - \frac{\delta E(T_j)}{\delta v}]\frac{1}{\mathbf{J}}ds \\
&= \int \mathbf{n}(v)[\log\frac{p(\mathbf{I}(v);\theta_{\ell_i})}{p(\mathbf{I}(v);\theta_{\ell_j})} + 0.9\gamma(\frac{1}{|D_j|} - \frac{1}{|D_i|}) \\
&+ \frac{D(G_{S_j}(v) - G_T(v))^2}{2\sigma^2}]\frac{1}{\mathbf{J}}ds
\end{aligned}$$

Structural changes in the solution W are realized by Markov chain jumps (see [19]). We design the following reversible jumps between:

(i) two regions – model switching: $\theta_1 \leftrightarrow \theta_2$
(ii) a region R and a text T: $\mathbf{R} \leftrightarrow T$
(iii) a region R and a face F: $\mathbf{R} \leftrightarrow F$
(iv) split or merge a region: $(\mathbf{R}_k) \leftrightarrow (\mathbf{R}_i, \mathbf{R}_j)$
(v) birth or death of a text: $T\,\{,\} \leftrightarrow \{,T\}$.

The Markov chain selects one of the above moves at each time, triggered by bottom-up compatibility conditions.